Large Scale Learning in Gaussian Processes

Neil D. Lawrence

11th July 2015
Large Scale Kernel Learning
Workshop@ICML

Outline

Gaussian Processes

Variational Methods

Variational Compression

Extensions

Outline

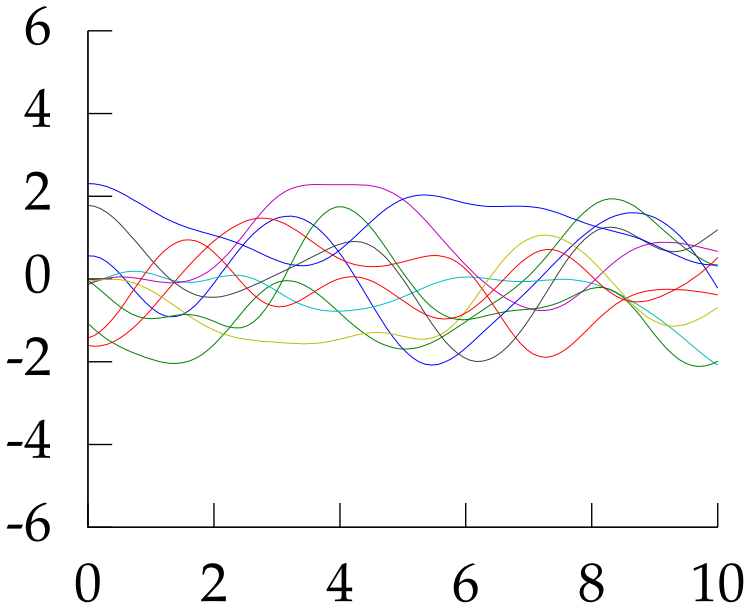
Gaussian Processes

Variational Methods

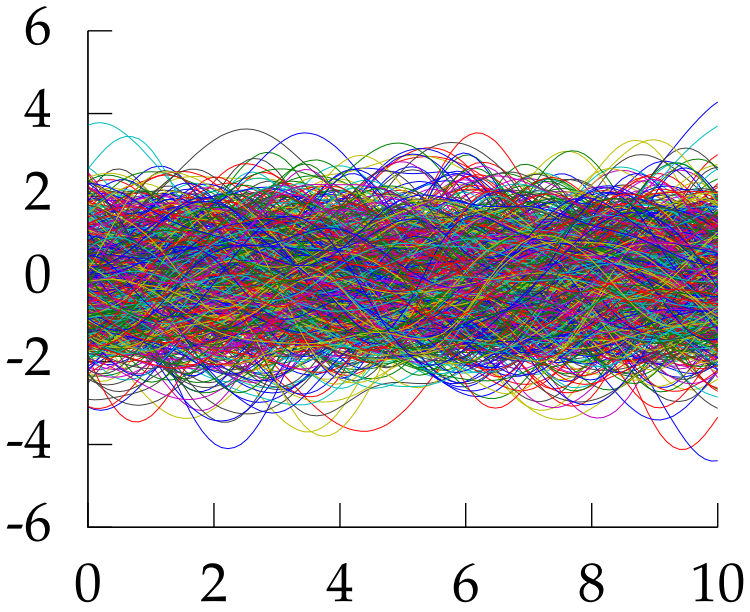
Variational Compression

Extensions

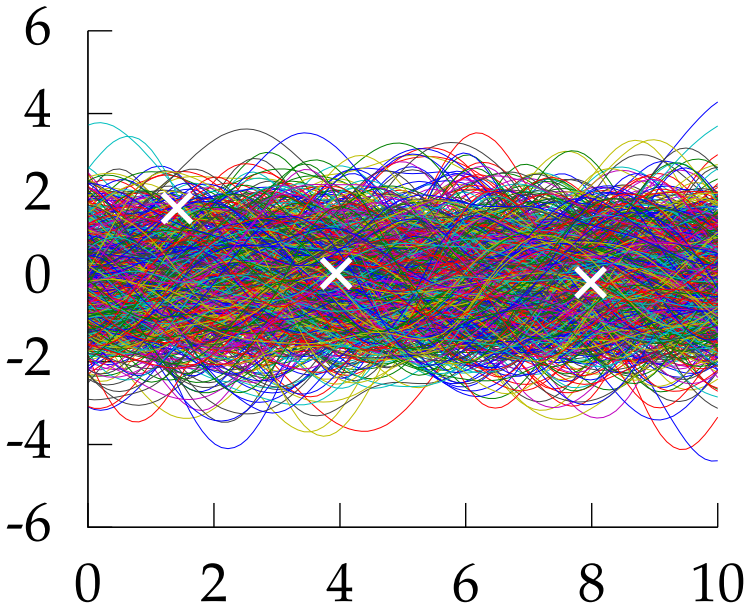
Gaussian Processes: Extremely Short Overview



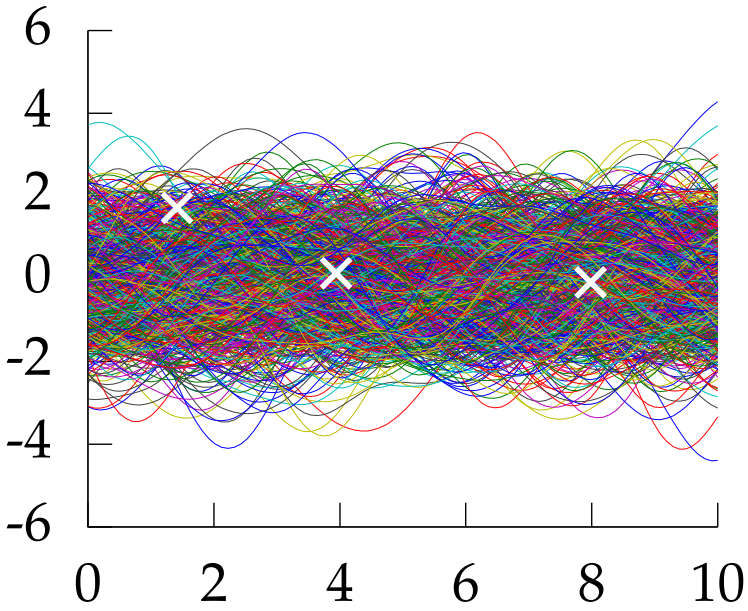
Gaussian Processes: Extremely Short Overview

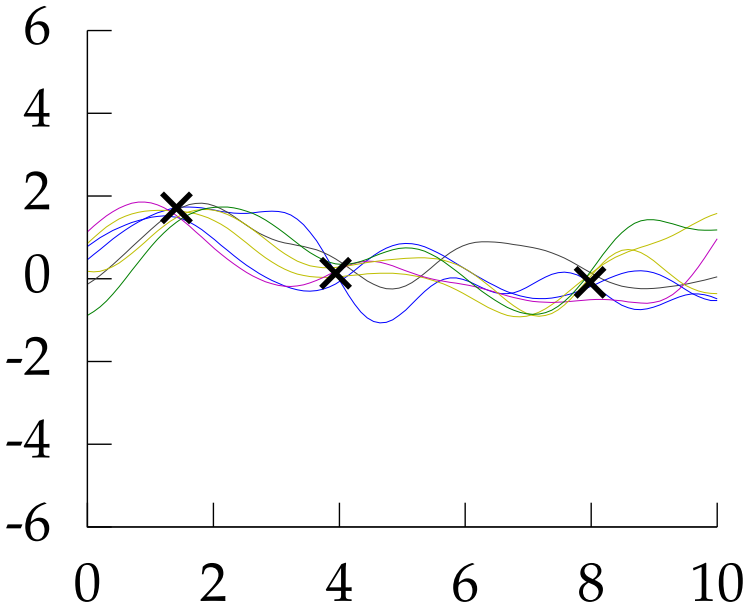


Gaussian Processes: Extremely Short Overview



Gaussian Processes: Extremely Short Overview





Gaussian Processes and Kernels

- The kernel, $\mathbf{K}$ acts as the *covariance* of the Gaussian.
- Predictions in Gaussian processes consist of the posterior *mean function* and *covariance*.
- For non degenerate covariance matrix, $\mathbf{K}$, the model is non parametric.

Gaussian Process Summer School

http://gpss.cc

14th-17th September 2015

Sheffield, UK

Covariance

- By convention we write covariance matrix as,

$$\begin{bmatrix} \mathbf{K}_{\mathbf{f},\mathbf{f}} & \mathbf{K}_{*,\mathbf{f}}^\top \\ \mathbf{K}_{*,\mathbf{f}} & \mathbf{K}_{*,*} \end{bmatrix},$$

  where
  - $\mathbf{K}_{\mathbf{f},\mathbf{f}}$ from kernel function computed at the *training data*
  - $\mathbf{K}_{*,\mathbf{f}}$ is the cross covariance between training and test.
  - $\mathbf{K}_{*,*}$ is the covariance for the test data.
- This encodes all the relationships between points in our function $f(\mathbf{x})$.

Prediction in Gaussian Processes

- If we make noise corrupted observations, $y_i = f(\mathbf{x}_i) + \epsilon$, $\epsilon \sim \mathcal{N}\left(0, \sigma^2\right)$
- We can estimate,

$$\mathbf{y}_* | \mathbf{y} \sim \mathcal{N}\left(\boldsymbol{\mu}, \mathbf{C}\right)$$

with

$$\boldsymbol{\mu} = \mathbf{K}_{*,\mathbf{f}} \left(\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2 \mathbf{I}\right)^{-1} \mathbf{y}$$

$$\mathbf{C} = \mathbf{K}_{*,*} - \mathbf{K}_{*,\mathbf{f}} \left(\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma^2 \mathbf{I}\right)^{-1} \mathbf{K}_{*,\mathbf{f}}^\top$$

- Model is non parametric. To make predictions we have to retain all training data $\{\mathbf{y}, \mathbf{X}\}$.

Variational Compression

(Snelson and Ghahramani, 2006; Quiñonero Candela and Rasmussen, 2005; Lawrence, 2007; Titsias, 2009)

- Complexity of standard GP:
  - $\mathcal{O}(n^3)$ in computation.
  - $\mathcal{O}(n^2)$ in storage.

Variational Compression

(Snelson and Ghahramani, 2006; Quiñonero Candela and Rasmussen, 2005; Lawrence, 2007; Titsias, 2009)

- Complexity of standard GP:
  - $\mathcal{O}(n^3)$ in computation.
  - $\mathcal{O}(n^2)$ in storage.
- Via low rank representations of covariance:
  - $\mathcal{O}(nm^2)$ in computation.
  - $\mathcal{O}(nm)$ in storage.
- Where m is user chosen number of *inducing* variables. They give the rank of the resulting covariance.

Variational Compression

(Snelson and Ghahramani, 2006; Quiñonero Candela and Rasmussen, 2005; Lawrence, 2007; Titsias, 2009)

- Complexity of standard GP:
  - $\mathcal{O}(n^3)$ in computation.
  - $\mathcal{O}(n^2)$ in storage.
- Via low rank representations of covariance:
  - $\mathcal{O}(nm^2)$ in computation.
  - $\mathcal{O}(nm)$ in storage.
- Where m is user chosen number of *inducing* variables. They give the rank of the resulting covariance.

Variational Compression

- Inducing variables are a compression of the real observations.
- They are like pseudo-data. They can be in space of $\mathbf{f}$ or a space that is related through a linear operator (Álvarez et al., 2010) — e.g. a gradient or convolution.

Variational Compression II

- **Importantly** conditioning on inducing variables renders the likelihood independent across the data.
- It turns out that this allows us to variationally handle uncertainty on the kernel (including the inputs to the kernel).
- It also allows standard scaling approaches: stochastic variational inference Hensman et al. (2013), parallelization Gal et al. (2014) and work by Zhenwen Dai on GPUs to be applied: an *engineering* challenge?

Outline

Gaussian Processes

Variational Methods

Variational Compression

Extensions

What is Variational Inference?

- Convert an integral into an optimization.
- Entered machine learning via statistical physics in 1990s.
- *But* there's a classic example from statistics: expectation maximization.

Variational Bound

Latent variable model: marginal likelihood computed by integrating latent variables.

$$p(\mathbf{y}|\boldsymbol{\theta}) = \int p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) p(\mathbf{x}) \mathrm{d}\mathbf{x}$$

Variational Bound

Log marginal likelihood computed by integrating latent variables.

$$\log p(\mathbf{y}|\boldsymbol{\theta}) = \log \int p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) p(\mathbf{x}) \mathrm{d}\mathbf{x}$$

Variational Bound

Jensen's inequality allows us to obtain a lower bound.

$$\log p(\mathbf{y}) = \log \int p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) p(\mathbf{x}) \mathrm{d}\mathbf{x}$$

Variational Bound

Jensen's inequality allows us to obtain a lower bound.

$$\log p(\mathbf{y}) \geq \int \log p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) p(\mathbf{x}) \mathrm{d}\mathbf{x}$$

But the bound can be very loose.

Variational Bound

Modify Jensens by introducing variational distribution, $q(\mathbf{x})$.

$$\log p(\mathbf{y}) = \log \int q(\mathbf{x}) \frac{p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) p(\mathbf{x})}{q(\mathbf{x})} \mathrm{d}\mathbf{x}$$

Variational Bound

Modify Jensens by introducing variational distribution, $q(\mathbf{x})$.

$$\log p(\mathbf{y}) \geq \int q(\mathbf{x}) \log \frac{p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) p(\mathbf{x})}{q(\mathbf{x})} \mathrm{d}\mathbf{x}$$

Bound is tightened through changing $q(\mathbf{x})$.

Variational Bound

This is the bound behind EM, in E-step set $q(\mathbf{x}) = p(\mathbf{x}|\mathbf{y})$.

$$\log p(\mathbf{y}) \geq \int q(\mathbf{x}) \log \frac{p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})p(\mathbf{x})}{q(\mathbf{x})} \mathrm{d}\mathbf{x}$$

Replace variational distribution with ...

Variational Bound

This is the bound behind EM, in E-step set $q(\mathbf{x}) = p(\mathbf{x}|\mathbf{y})$.

$$\log p(\mathbf{y}|\boldsymbol{\theta}) = \int p(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}) \log \frac{p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})p(\mathbf{x})}{p(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})} \mathrm{d}\mathbf{x}$$

... true posterior which allows for ...

Variational Bound

This is the bound behind EM, in E-step set $q(\mathbf{x}) = p(\mathbf{x}|\mathbf{y})$.

$$\log p(\mathbf{y}|\boldsymbol{\theta}) = \int p(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}) \log p(\mathbf{y}|\boldsymbol{\theta}) \mathrm{d}\mathbf{x}$$

... a reorganisation via product rule ...

Variational Bound

This is the bound behind EM, in E-step set $q(\mathbf{x}) = p(\mathbf{x}|\mathbf{y})$.

$$\log p(\mathbf{y}|\boldsymbol{\theta}) = \log p(\mathbf{y}|\boldsymbol{\theta})$$

... to recover equality (bound is tight).

Variational Bound

This is the bound behind EM, in M-step ignore fact that $q(\mathbf{x}) = p(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$ should depend on parameters and maximize bound.

$$\log p(\mathbf{y}) \geq \int q(\mathbf{x}) \log \frac{p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) p(\mathbf{x})}{q(\mathbf{x})} \mathrm{d}\mathbf{x}$$

Variational Bound

This is the bound behind EM, in M-step ignore fact that $q(\mathbf{x}) = p(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$ should depend on parameters and maximize bound.

$$\log p(\mathbf{y}) \geq \int q(\mathbf{x}) \log p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) \mathrm{d}\mathbf{x} + \int q(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} \mathrm{d}\mathbf{x}$$

Split into expected log likelihood ...

Variational Bound

This is the bound behind EM, in M-step ignore fact that $q(\mathbf{x}) = p(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$ should depend on parameters and maximize bound.

$$\log p(\mathbf{y}) \geq \langle \log p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) \rangle_{q(\mathbf{x})} - \mathrm{KL}\left(q(\mathbf{x}) \,||\, p(\mathbf{x})\right)$$

... and Kullback Leibler divergence term.

Variational Bound

This gives the variational lower bound ...

$$\mathcal{L} = \langle \log p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) \rangle_{q(\mathbf{x})} - \mathrm{KL}\left(q(\mathbf{x}) \,||\, p(\mathbf{x})\right)$$

... which is an information theoretic interpretation of marginalization.

How is this a Variational Method?

- To apply EM we need to compute $p(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$
- Often this is intractable, in this case we note that:

$$\log p(\mathbf{y}) = \int q(\mathbf{x}) \log \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{q(\mathbf{x})} \mathrm{d}\mathbf{x} + \int q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p(\mathbf{x}|\mathbf{y})} \mathrm{d}\mathbf{x}$$

(dropping conditioning on $\boldsymbol{\theta}$)
the difference between the bound and the log likelihood is the Kullback Leibler divergence between the true posterior and the variational distribution $q(\mathbf{x})$.

How is this a Variational Method?

- To apply EM we need to compute $p(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$
- Often this is intractable, in this case we note that:

$$\log p(\mathbf{y}) = \mathcal{L} + \mathrm{KL}\left(q(\mathbf{x}) \,\|\, p(\mathbf{x}|\mathbf{y})\right)$$

(dropping conditioning on $\boldsymbol{\theta}$)
the difference between the bound and the log likelihood is the Kullback Leibler divergence between the true posterior and the variational distribution $q(\mathbf{x})$.

Outline

Gaussian Processes

Variational Methods

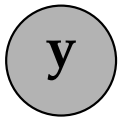
Variational Compression

Extensions

Variational Compression

Model for our data, $\mathbf{y}$.

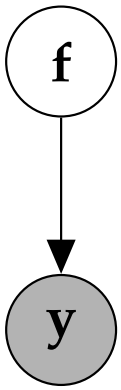
$$p(\mathbf{y})$$



Variational Compression

Prior density over $\mathbf{f}$. Likelihood relates data, $\mathbf{y}$, to $\mathbf{f}$.

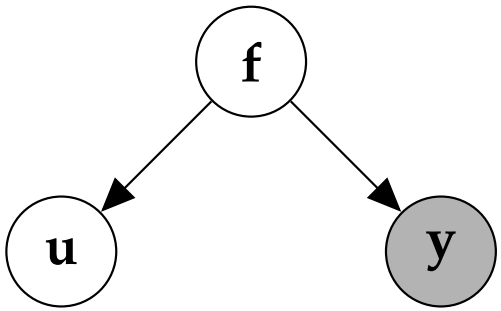
$$p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f})\mathrm{d}\mathbf{f}$$



Variational Compression

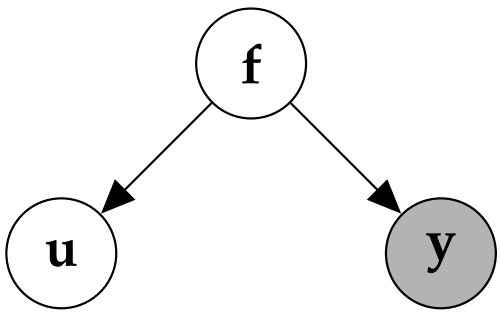
Augment standard model with a set of m new inducing variables, $\mathbf{u}$.

$$p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{f})p(\mathbf{u}|\mathbf{f})p(\mathbf{f})\mathrm{d}\mathbf{f}\mathrm{d}\mathbf{u}$$



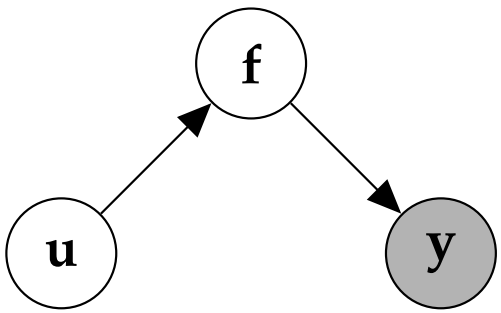
Variational Compression

$$p(\mathbf{y}) = \int \int p(\mathbf{y}|\mathbf{f})p(\mathbf{u}|\mathbf{f})p(\mathbf{f})\mathrm{d}\mathbf{f}\mathrm{d}\mathbf{u}$$



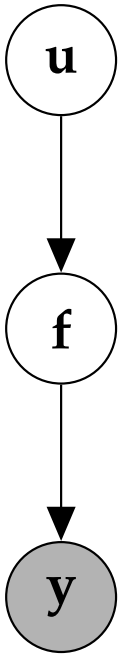
Variational Compression

$$p(\mathbf{y}) = \int \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{u})\mathrm{d}\mathbf{f}p(\mathbf{u})\mathrm{d}\mathbf{u}$$



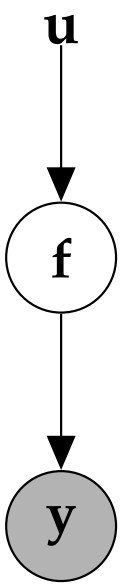
Variational Compression

$$p(\mathbf{y}) = \int \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{u})\mathrm{d}\mathbf{f}p(\mathbf{u})\mathrm{d}\mathbf{u}$$



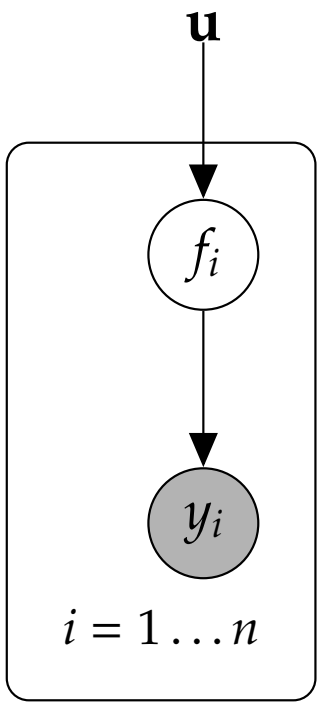
Variational Compression

$$p(\mathbf{y}|\mathbf{u}) = \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{u})\mathrm{d}\mathbf{f}$$



Variational Compression

$$p(\mathbf{y}|\mathbf{u}) = \int \prod_{i=1}^{n} p(y_i|f_i)p(\mathbf{f}|\mathbf{u})\mathrm{d}\mathbf{f}$$



Variational Compression

Consider the conditional likelihood.

$$p(\mathbf{y}|\mathbf{u}) = \int \prod_{i=1}^{n} p(y_i|f_i) p(\mathbf{f}|\mathbf{u}) \mathrm{d}\mathbf{f}$$

Variational Compression

Consider the conditional log likelihood.

$$\log p(\mathbf{y}|\mathbf{u}) = \log \int \prod_{i=1}^{n} p(y_i|f_i) p(\mathbf{f}|\mathbf{u}) \mathrm{d}\mathbf{f}$$

Variational Compression

Introduce variational lower bound

$$\log p(\mathbf{y}|\mathbf{u}) \geq \int q(\mathbf{f}) \log \frac{\prod_{i=1}^{n} p(y_i|f_i) p(\mathbf{f}|\mathbf{u})}{q(\mathbf{f})} \mathrm{d}\mathbf{f}$$

Variational Compression

Set $q(\mathbf{f}) = p(\mathbf{f}|\mathbf{u})$

$$\log p(\mathbf{y}|\mathbf{u}) \geq \int p(\mathbf{f}|\mathbf{u}) \sum_{i=1}^{n} \log p(y_i|f_i) \mathrm{d}\mathbf{f}$$

Variational Compression

Set $q(\mathbf{f}) = p(\mathbf{f}|\mathbf{u})$

$$\log p(\mathbf{y}|\mathbf{u}) \geq \sum_{i=1}^{n} \langle \log p(y_i|f_i) \rangle_{p(f_i|\mathbf{u})}$$

Variational Compression

Difference between bound and truth is KL divergence:

$$\mathrm{KL}\left(p(\mathbf{f}|\mathbf{u}) \,||\, p(\mathbf{f}|\mathbf{u},\mathbf{y})\right) = \int p(\mathbf{f}|\mathbf{u}) \log \frac{p(\mathbf{f}|\mathbf{u})}{p(\mathbf{f}|\mathbf{u},\mathbf{y})} \mathrm{d}\mathbf{f}$$

This is why we call it variational compression, information in $\mathbf{y}$ is compressed into $\mathbf{u}$

Augmented Covariance

- For GPs assume augmented covariance

$$\begin{bmatrix} \mathbf{K}_{\mathbf{f},\mathbf{f}} & \mathbf{K}_{\mathbf{u},\mathbf{f}}^\top & \mathbf{K}_{*,\mathbf{f}}^\top \\ \mathbf{K}_{\mathbf{u},\mathbf{f}} & \mathbf{K}_{\mathbf{u},\mathbf{u}} & \mathbf{K}_{*,\mathbf{u}}^\top \\ \mathbf{K}_{*,\mathbf{f}} & \mathbf{K}_{*,\mathbf{u}} & \mathbf{K}_{*,*} \end{bmatrix},$$

  where
  - $\mathbf{K}_{\mathbf{u},\mathbf{f}}$ is the cross covariance between the original function and inducing vectors
  - $\mathbf{K}_{*,\mathbf{u}}$ is the cross covariance between inducing vectors and test data.
  - $\mathbf{K}_{\mathbf{u},\mathbf{u}}$ is the covariance for the inducing vectors data.

Conditional Distribution

- For variational compression we need to estimate,

$$\mathbf{f}|\mathbf{u} \sim \mathcal{N}\left(\boldsymbol{\mu}, \mathbf{C}\right)$$

with

$$\boldsymbol{\mu} = \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u}$$

$$\mathbf{C} = \mathbf{K}_{\mathbf{f},\mathbf{f}} - \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}}$$

Gaussian $p(y_i|f_i)$

For Gaussian likelihoods:

$$\langle \log p(y_i|f_i) \rangle_{p(f_i|\mathbf{u})} = -\frac{1}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}\left(y_i - \langle f_i \rangle\right)^2 - \frac{1}{2\sigma^2}\left(\left\langle f_i^2 \right\rangle - \langle f_i \rangle^2\right)$$

Gaussian $p(y_i|f_i)$

For Gaussian likelihoods:

$$\langle \log p(y_i|f_i) \rangle_{p(f_i|\mathbf{u})} = -\frac{1}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}\left(y_i - \langle f_i \rangle\right)^2 - \frac{1}{2\sigma^2}\left(\left\langle f_i^2 \right\rangle - \langle f_i \rangle^2\right)$$

Implying:

$$p(y_i|\mathbf{u}) \geq \exp\langle \log c_i \rangle \, \mathcal{N}\left(y_i | \langle f_i \rangle, \sigma^2\right)$$

Gaussian Process Over $\mathbf{f}$ and $\mathbf{u}$

Define:

$$q_{i,i} = \text{var}_{p(f_i|\mathbf{u})}(f_i) = \left\langle f_i^2 \right\rangle_{p(f_i|\mathbf{u})} - \langle f_i \rangle^2_{p(f_i|\mathbf{u})}$$

We can write:

$$c_i = \exp\left(-\frac{q_{i,i}}{2\sigma^2}\right)$$

If joint distribution of $p(\mathbf{f}, \mathbf{u})$ is Gaussian then:

$$q_{i,i} = k_{i,i} - \mathbf{k}_{i,\mathbf{u}}^\top \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} \mathbf{k}_{i,\mathbf{u}}$$

c_i is not a function of $\mathbf{u}$ but *is* a function of $\mathbf{X}_\mathbf{u}$.

Lower Bound on Likelihood

Substitute variational bound into marginal likelihood:

$$p(\mathbf{y}) \geq \prod_{i=1}^{n} c_i \int \mathcal{N}\left(\mathbf{y} | \langle \mathbf{f} \rangle, \sigma^2 \mathbf{I}\right) p(\mathbf{u}) \mathrm{d}\mathbf{u}$$

Note that:

$$\langle \mathbf{f} \rangle_{p(\mathbf{f}|\mathbf{u})} = \mathbf{K}_{\mathbf{f},\mathbf{u}} \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} \mathbf{u}$$

is *linearly* dependent on $\mathbf{u}$.

Deterministic Training Conditional

Making the marginalization of $\mathbf{u}$ straightforward. In the Gaussian case:

$$p(\mathbf{u}) = \mathcal{N}\left(\mathbf{u}|\mathbf{0}, \mathbf{K}_{\mathbf{u},\mathbf{u}}\right)$$

$$\int p(\mathbf{y}|\mathbf{u})p(\mathbf{u})\mathrm{d}\mathbf{u} \geq \prod_{i=1}^{n} c_i \int \mathcal{N}\left(\mathbf{y}|\mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u}, \sigma^2\right) \mathcal{N}\left(\mathbf{u}|\mathbf{0}, \mathbf{K}_{\mathbf{u},\mathbf{u}}\right) \mathrm{d}\mathbf{u}$$

Deterministic Training Conditional

Making the marginalization of $\mathbf{u}$ straightforward. In the Gaussian case:

$$p(\mathbf{u}) = \mathcal{N}\left(\mathbf{u}|\mathbf{0}, \mathbf{K}_{\mathbf{u},\mathbf{u}}\right)$$

$$\int p(\mathbf{y}|\mathbf{u})p(\mathbf{u})\mathrm{d}\mathbf{u} \geq \prod_{i=1}^{n} c_i \mathcal{N}\left(\mathbf{y}|\mathbf{0}, \sigma^2\mathbf{I} + \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}}\right)$$

Deterministic Training Conditional

Making the marginalization of $\mathbf{u}$ straightforward. In the Gaussian case:

$$p(\mathbf{u}) = \mathcal{N}\left(\mathbf{u}|\mathbf{0}, \mathbf{K}_{\mathbf{u},\mathbf{u}}\right)$$

$$\int p(\mathbf{y}|\mathbf{u})p(\mathbf{u})\mathrm{d}\mathbf{u} \geq \prod_{i=1}^{n} c_i \mathcal{N}\left(\mathbf{y}|\mathbf{0}, \sigma^2\mathbf{I} + \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}}\right)$$

Maximize log of the bound to find covariance function parameters,

$$L \geq \sum_{i=1}^{n} \log c_i + \log \mathcal{N}\left(\mathbf{y}|\mathbf{0}, \sigma^2\mathbf{I} + \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f},}\right)$$

Deterministic Training Conditional

Making the marginalization of $\mathbf{u}$ straightforward. In the Gaussian case:

$$p(\mathbf{u}) = \mathcal{N}\left(\mathbf{u}|\mathbf{0}, \mathbf{K}_{\mathbf{u},\mathbf{u}}\right)$$

$$\int p(\mathbf{y}|\mathbf{u})p(\mathbf{u})\mathrm{d}\mathbf{u} \geq \prod_{i=1}^{n} c_i \mathcal{N}\left(\mathbf{y}|\mathbf{0}, \sigma^2\mathbf{I} + \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}}\right)$$

Maximize log of the bound to find covariance function parameters,

$$L \geq \sum_{i=1}^{n} \log c_i + \log \mathcal{N}\left(\mathbf{y}|\mathbf{0}, \sigma^2\mathbf{I} + \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f},}\right)$$

Deterministic Training Conditional

Making the marginalization of $\mathbf{u}$ straightforward. In the Gaussian case:

$$p(\mathbf{u}) = \mathcal{N}\left(\mathbf{u}|\mathbf{0}, \mathbf{K}_{\mathbf{u},\mathbf{u}}\right)$$

$$\int p(\mathbf{y}|\mathbf{u})p(\mathbf{u})\mathrm{d}\mathbf{u} \geq \prod_{i=1}^{n} c_i \mathcal{N}\left(\mathbf{y}|\mathbf{0}, \sigma^2\mathbf{I} + \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}}\right)$$

Maximize log of the bound to find covariance function parameters,

$$L \approx \log \mathcal{N}\left(\mathbf{y}|\mathbf{0}, \sigma^2\mathbf{I} + \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f},}\right)$$

- If the bound is normalized, the c_i terms are removed.

Deterministic Training Conditional

Making the marginalization of $\mathbf{u}$ straightforward. In the Gaussian case:

$$p(\mathbf{u}) = \mathcal{N}\left(\mathbf{u}|\mathbf{0}, \mathbf{K}_{\mathbf{u},\mathbf{u}}\right)$$

$$\int p(\mathbf{y}|\mathbf{u})p(\mathbf{u})\mathrm{d}\mathbf{u} \geq \prod_{i=1}^{n} c_i \mathcal{N}\left(\mathbf{y}|\mathbf{0}, \sigma^2\mathbf{I} + \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}}\right)$$

Maximize log of the bound to find covariance function parameters,

- If the bound is normalized, the c_i terms are removed.
- This results in the projected process approximation (Rasmussen and Williams, 2006) or DTC (Quiñonero Candela and Rasmussen, 2005). Proposed by (Smola and Bartlett, 2001; Seeger et al., 2003; Csató and Opper, 2002; Csató, 2002).

Relationship to Nyström Approximation

- Variational lower bound leads to Nyström style approximation (Williams and Seeger, 2001; Seeger et al., 2003). Relations to subset of regressors (Poggio and Girosi, 1990; Williams et al., 2002).

$$\mathbf{K} \approx \sigma^2\mathbf{I} + \mathbf{K}_{\mathbf{fu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{uf}}$$

- Has probabilistic interpretation of

$$\mathbf{u} \sim \mathcal{N}\left(0, \mathbf{K}_{\mathbf{uu}}\right)$$
$$\mathbf{y}|\mathbf{u} \sim \mathcal{N}\left(\mathbf{K}_{\mathbf{fu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{u}, \sigma^2\mathbf{I}\right)$$

cf

$$\mathbf{w} \sim \mathcal{N}\left(0, \alpha\mathbf{I}\right)$$
$$\mathbf{y}|\mathbf{w} \sim \mathcal{N}\left(\mathbf{\Phi}\mathbf{w}, \sigma^2\mathbf{I}\right)$$
$$\mathbf{y} \sim \mathcal{N}\left(0, \alpha\mathbf{\Phi}\mathbf{\Phi}^\top + \sigma^2\mathbf{I}\right)$$

Outline

Gaussian Processes

Variational Methods

Variational Compression

Extensions

Marginalising Latent Variables

- Integrating out $\mathbf{X}$ becomes possible variationally, because Gaussian expectations of

$$\log \mathcal{N}\left(\mathbf{f}|\mathbf{K}_{\mathbf{fu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{u}, \sigma^2\mathbf{I}\right)$$

  are now *tractable*
- Relies on computing expectations of $\mathbf{K}_{\mathbf{fu}}$ and $\mathbf{K}_{\mathbf{uf}}\mathbf{K}_{\mathbf{fu}}$ under Gaussian density over $\mathbf{X}$.

Apply Variational Inference Before Integration of $\mathbf{u}$

$$\int p(\mathbf{y}|\mathbf{u})p(\mathbf{u})\mathrm{d}\mathbf{u} \geq \prod_{i=1}^{n} c_i \int \mathcal{N}\left(\mathbf{y}|\mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u}, \sigma^2\right) \mathcal{N}\left(\mathbf{u}|\mathbf{0}, \mathbf{K}_{\mathbf{u},\mathbf{u}}\right) \mathrm{d}\mathbf{u}$$

Apply Variational Inference Before Integration of $\mathbf{u}$

$$\int p(\mathbf{y}|\mathbf{u})p(\mathbf{u})p(\mathbf{x})\mathrm{d}\mathbf{u}\mathrm{d}\mathbf{x} \geq$$

$$\int q(\mathbf{x}) \log \frac{\prod_{i=1}^{n} c_i \mathcal{N}\left(\mathbf{y}|\mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u}, \sigma^2\right) \mathcal{N}\left(\mathbf{u}|\mathbf{0}, \mathbf{K}_{\mathbf{u},\mathbf{u}}\right)}{q(\mathbf{x})} \mathrm{d}\mathbf{u}$$

Titsias and Lawrence (2010)

Bayesian Gaussian Process Latent Variable Model

Michalis K. Titsias
School of Computer Science
University of Manchester

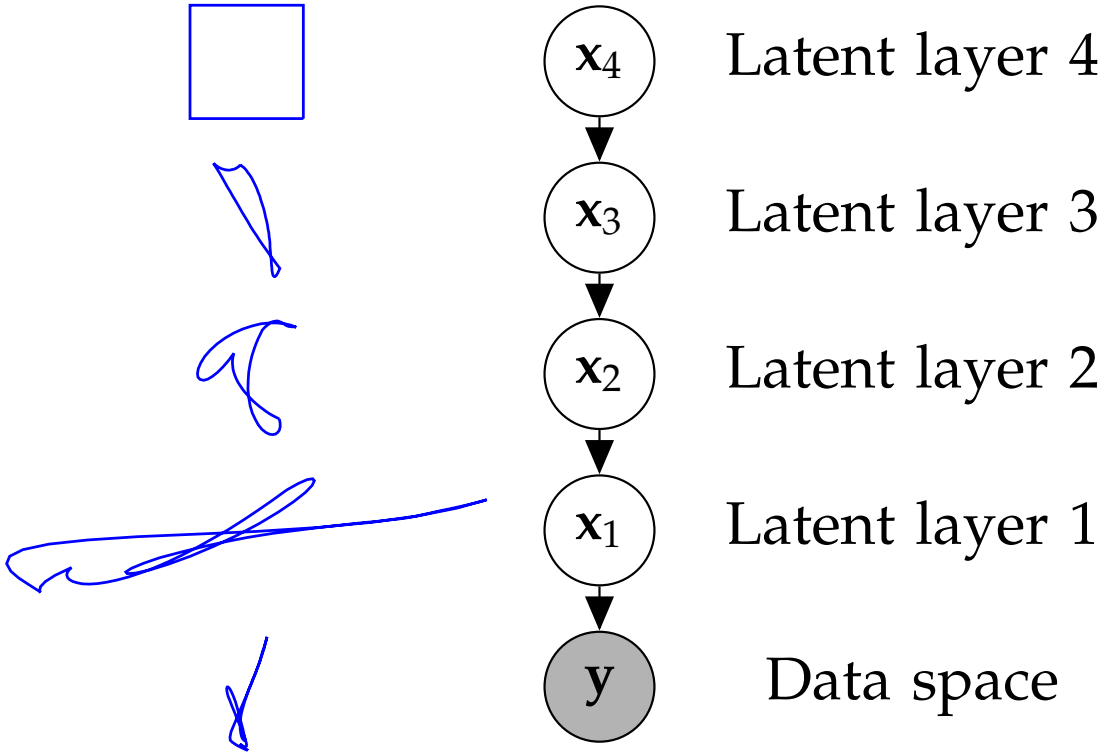
Neil D. Lawrence
School of Computer Science
University of Manchester

Abstract

We introduce a variational inference framework for training the Gaussian process latent variable model and thus performing Bayesian nonlinear dimensionality reduction. This method allows us to variationally integrate out the input variables of the Gaussian process and compute a lower bound on the exact marginal likelihood of the nonlinear latent variable model. The maxi-

Lawrence (2004, 2005). GP-LVM can be considered as a multiple-output GP regression model where only the output data are given. The inputs are unobserved and are treated as latent variables, however instead of integrating out the latent variables, they are optimized. This trick makes the model tractable and some theoretical grounding for the approach is given by the fact that the model can be seen as a nonlinear extension of the linear probabilistic PCA (PPCA). In PPCA (and in factor analysis (FA)) Bayesian extensions of the model are straightforward

Structures for Extracting Information from Data



Damianou and Lawrence (2013)

Deep Gaussian Processes

Andreas C. Damianou **Neil D. Lawrence**
Dept. of Computer Science & Sheffield Institute for Translational Neuroscience,
University of Sheffield, UK

Abstract

In this paper we introduce deep Gaussian process (GP) models. Deep GPs are a deep belief network based on Gaussian process mappings. The data is modeled as the output of a multivariate GP. The inputs to that Gaussian process are then governed by another GP. A single layer model is equivalent to a standard GP or the GP latent variable model (GP-LVM). We perform inference in

the question as to whether deep structures and the learning of abstract structure can be undertaken in *smaller* data sets. For smaller data sets, questions of generalization arise: to demonstrate such structures are justified it is useful to have an objective measure of the model's applicability.

The traditional approach to deep learning is based around binary latent variables and the restricted Boltzmann machine (RBM) [Hinton, 2010]. Deep hierarchies are constructed by stacking these models and various approximate inference techniques (such as contrastive divergence)

Leads to Other Approximations ...

- Let's be explicity about storing approximate posterior of $\mathbf{u}$, $q(\mathbf{u})$.
- Now we have

$$p(\mathbf{y}^*|\mathbf{y}) = \int p(\mathbf{y}^*|\mathbf{u})q(\mathbf{u}|\mathbf{y})\mathbf{u}$$

- Inducing variables look a lot like regular parameters.
- *But*: their dimensionality does not need to be set at design time.
- They can be modified arbitrarily at run time without effecting the model likelihood.
- They only effect the quality of compression and the lower bound.

In GPs for Big Data

- Exploit the resulting factorization ...

$$p(\mathbf{y}^*|\mathbf{y}) = \int p(\mathbf{y}^*|\mathbf{u})q(\mathbf{u}|\mathbf{y})\mathbf{u}$$

In GPs for Big Data

- Exploit the resulting factorization ...

$$p(\mathbf{y}^*|\mathbf{y}) = \int p(\mathbf{y}^*|\mathbf{u})q(\mathbf{u}|\mathbf{y})\mathbf{u}$$

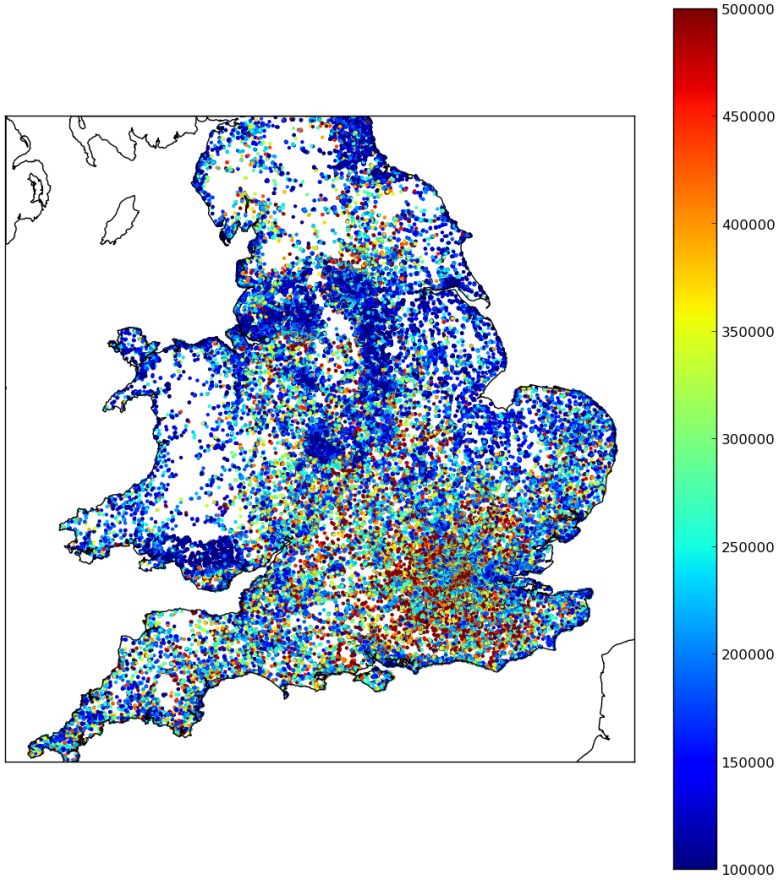
- The distribution now *factorizes*:

$$p(\mathbf{y}^*|\mathbf{y}) = \int \prod_{i=1}^{n^*} p(y_i^*|\mathbf{u})q(\mathbf{u}|\mathbf{y})\mathbf{u}$$

- This factorization can be exploited for stochastic variational inference (Hoffman et al., 2012).

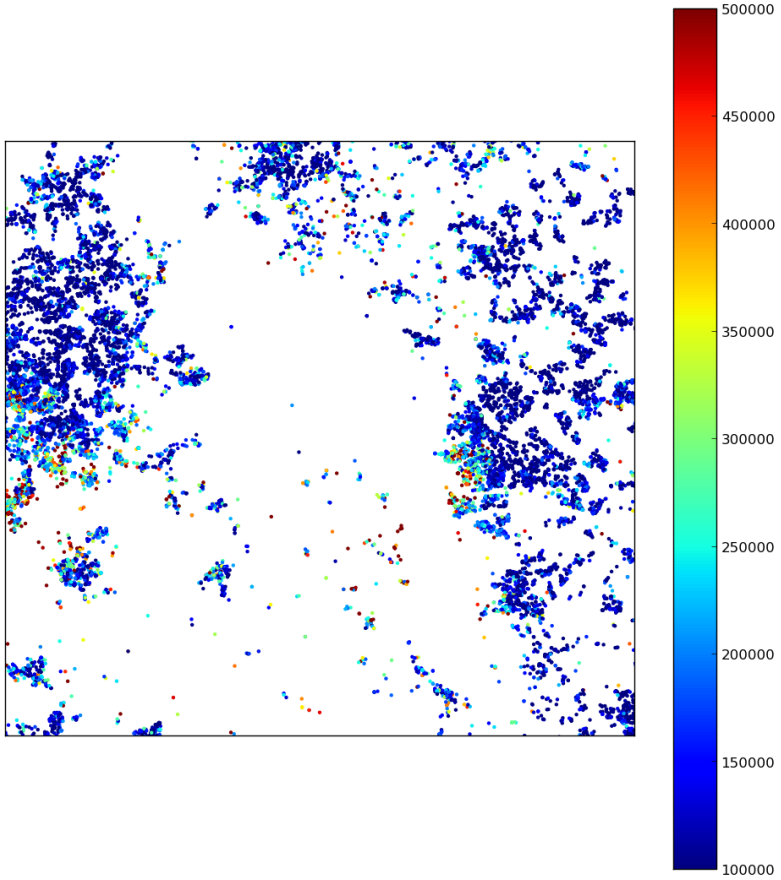
Nonparametrics for Very Large Data Sets

Modern data availability



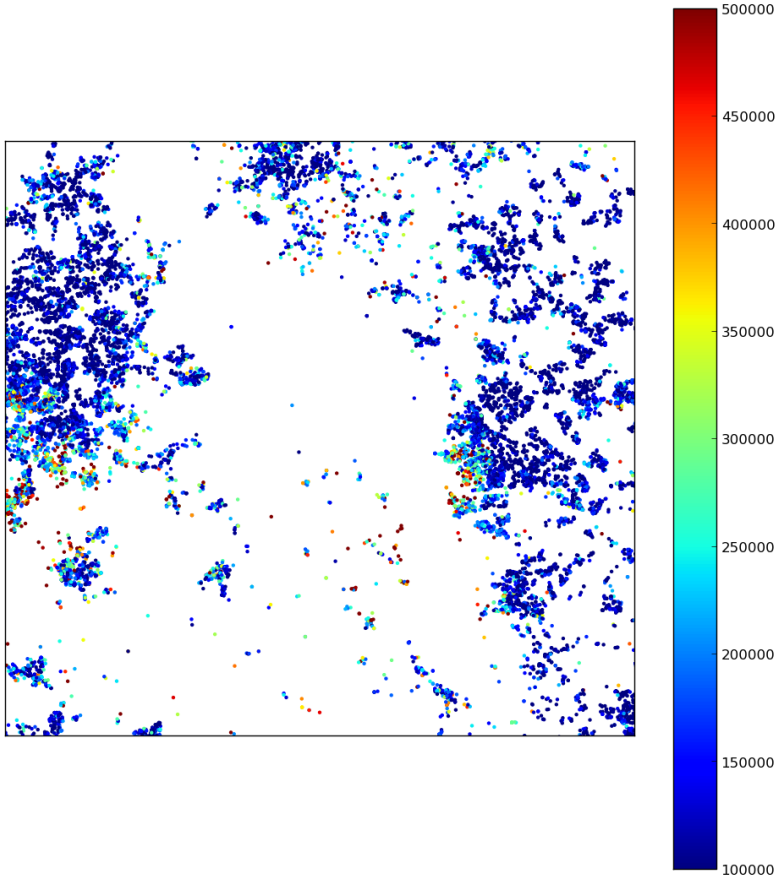
Nonparametrics for Very Large Data Sets

Proxy for index of deprivation?



Nonparametrics for Very Large Data Sets

Actually index of deprivation is a proxy for this ...



Hensman et al. (2013)

Gaussian Processes for Big Data

James Hensman*
Dept. Computer Science
The University of Sheffield
Sheffield, UK

Nicolò Fusi*
Dept. Computer Science
The University of Sheffield
Sheffield, UK

Neil D. Lawrence*
Dept. Computer Science
The University of Sheffield
Sheffield, UK

Abstract

We introduce stochastic variational inference for Gaussian process models. This enables the application of Gaussian process (GP) models to data sets containing millions of data points. We show how GPs can be variationally decomposed to depend on a set

Even to accommodate these data sets, various approximate techniques are required. One approach is to partition the data set into separate groups [e.g. Snelson and Ghahramani, 2007, Urtasun and Darrell, 2008]. An alternative is to build a low rank approximation to the covariance matrix based around 'inducing variables' [see e.g. Csató and Opper, 2002, Seeger et al., 2003, Quiñonero Candela and Rasmussen, 2005, Tit-

Hensman et al. (2013)

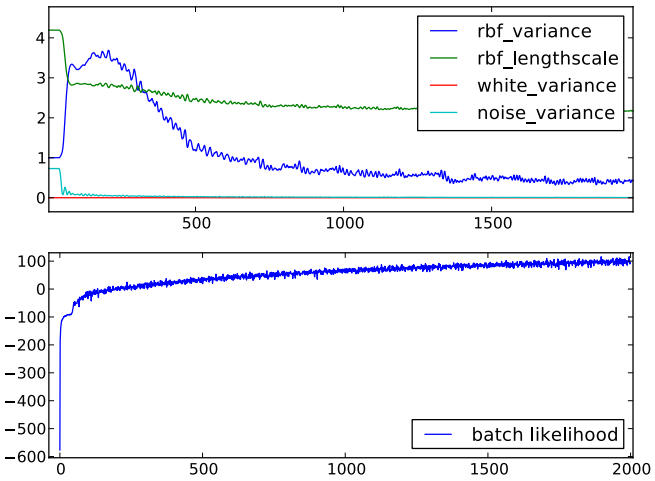


Figure 4: Convergence of the SVIGP algorithm on the two dimensional toy data

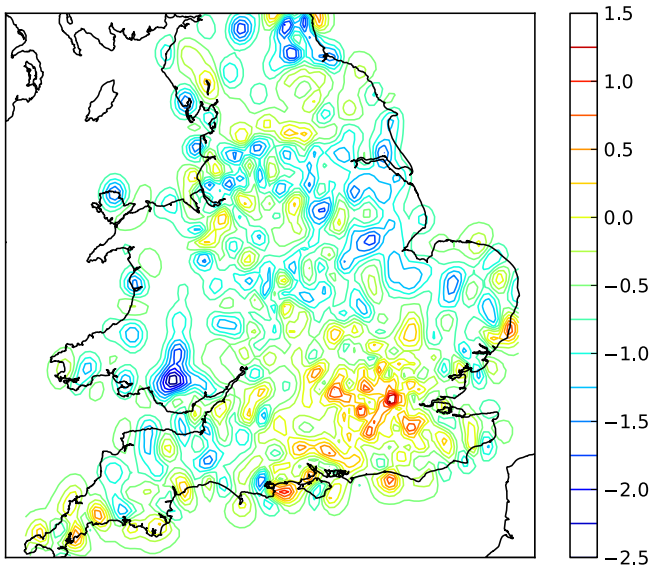


Figure 5: Variability of apartment price (logarithmically!) throughout England and Wales.

`land-registry-monthly-price-paid-data/`, which covers England and Wales, and filtered for apartments. This resulted in a data set with 75,000 entries,

ted a GP with the same covariance function as our

Practical Nonparametric Modelling

- Dealing with modern data requires non parametrics
- Non parametrics by their nature have implications on storage and computation.
- Variational approximations provide an optimal compression of the non parametric model to parametric.
- The quality of the approximation can be varied at *run time* according to particular modelling needs.
- The variational compression formalism leads to a raft of extensions.

References I

M. A. Álvarez, D. Luengo, M. K. Titsias, and N. D. Lawrence. Efficient multioutput Gaussian processes through variational inducing kernels. In Teh and Titterington (2010), pages 25–32. [PDF].

L. Csató. *Gaussian Processes — Iterative Sparse Approximations*. PhD thesis, Aston University, 2002.

L. Csató and M. Opper. Sparse on-line Gaussian processes. *Neural Computation*, 14(3):641–668, 2002.

A. Damianou and N. D. Lawrence. Deep Gaussian processes. In C. Carvalho and P. Ravikumar, editors, *Proceedings of the Sixteenth International Workshop on Artificial Intelligence and Statistics*, volume 31, AZ, USA, 2013. JMLR W&CP 31. [PDF].

Y. Gal, M. van der Wilk, and C. E. Rasmussen. Distributed variational inference in sparse Gaussian process regression and latent variable models. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27, Cambridge, MA, 2014.

J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. In A. Nicholson and P. Smyth, editors, *Uncertainty in Artificial Intelligence*, volume 29. AUAI Press, 2013. [PDF].

M. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. Technical report,

N. D. Lawrence. Learning for larger datasets with the Gaussian process latent variable model. In M. Meila and X. Shen, editors, *Proceedings of the Eleventh International Workshop on Artificial Intelligence and Statistics*, pages 243–250, San Juan, Puerto Rico, 21-24 March 2007. Omnipress. [PDF].

T. K. Leen, T. G. Dietterich, and V. Tresp, editors. *Advances in Neural Information Processing Systems*, volume 13, Cambridge, MA, 2001. MIT Press.

T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.

J. Quiñonero Candela and C. E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, 2005.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006. [Google Books] .

M. Seeger, C. K. I. Williams, and N. D. Lawrence. Fast forward selection to speed up sparse Gaussian process regression. In C. M. Bishop and B. J. Frey, editors, *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, Key West, FL, 3–6 Jan 2003.

References II

A. J. Smola and P. L. Bartlett. Sparse greedy Gaussian process regression. In Leen et al. (2001), pages 619–625.

E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18, Cambridge, MA, 2006. MIT Press.

Y. W. Teh and D. M. Titterington, editors. *Artificial Intelligence and Statistics*, volume 9, Chia Laguna Resort, Sardinia, Italy, 13-16 May 2010. JMLR W&CP 9.

M. K. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In D. van Dyk and M. Welling, editors, *Proceedings of the Twelfth International Workshop on Artificial Intelligence and Statistics*, volume 5, pages 567–574, Clearwater Beach, FL, 16-18 April 2009. JMLR W&CP 5.

M. K. Titsias and N. D. Lawrence. Bayesian Gaussian process latent variable model. In Teh and Titterington (2010), pages 844–851. [PDF].

C. K. I. Williams, C. E. Rasmussen, A. Schwaighofer, and V. Tresp. Observations of the Nyström method for Gaussian process prediction. Technical report, University of Edinburgh,

C. K. I. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In Leen et al. (2001), pages 682–688.