Modelling with Massively Missing Data

Neil D. Lawrence
Sheffield Institute of Translational Neuroscience and
Department of Computer Science, University of Sheffield,
U.K.

Facebook, Menlo Park

20th March 2014

Outline

Box Quote

Nonparametrics

Flexible Parametric Approximation

Conclusions

Box Quote

All models are wrong, but some are useful. *(Box, 1976)*

Box Quote

All models are wrong, but some are useful. *(Box, 1976)*

- Useful quote, but overused.

Box Quote

All models are wrong, but some are useful. *(Box, 1976)*

- Useful quote, but overused.
- Almost become an excuse, my model is wrong so it *might* be useful.

Box Quote

All models are wrong, but some are useful. *(Box, 1976)*

- Useful quote, but overused.
- Almost become an excuse, my model is wrong so it *might* be useful.

... the scientist must be alert to what is importantly wrong. It is inappropriate to worry about mice when there are tigers abroad. *(Box, 1976)*

An Incorrect Model

- Write down our data ...

$$\mathbf{Y} \in \mathfrak{R}^{n \times p}$$

An Incorrect Model

- Write down our data ...

$$\mathbf{Y} \in \mathfrak{R}^{n \times p}$$

... this is WRONG!

- A presumption: there is something special and separate about indices over n and p.
- The subtle difference between features and data points.
- In practice both n and p could be uncountably large!
- Standard approach seems to assume that p is fixed.
- A historic anachronism from the days of collating statistical information?

There is nothing special about p ...

- Rather ... let's assume each data is indexed by the type of data, as well as location, time, etc.
- So $y_{17,234}$ is price of a hamburger from McDonald's in Leicester square on 13th April 1984 at 13:34 and $y_{239,201}$ is the price of a chicken wrap from Pret a Manger in Cambridge on 27th December 2001 at 14:34.
- Further $y_{734,124}$ might be the brand of car my mother currently drives.

Prediction

The answer to any prediction problem is a probability distribution. *(Peter McCullogh via Peter Diggle)*

- We assume that we are interested in predicting something about our variables (the likely cost of a burger given the cost of a chicken wrap).

Factorizations

- Often researchers write down the resulting factorization without a second thought:

$$p(\mathbf{Y}|\boldsymbol{\theta}) = \prod_{i=1}^{n} p(\mathbf{y}_{i,:}|\boldsymbol{\theta})$$

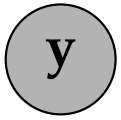
- This means that all our information about different data is stored in the parameters.
- If model is complex, and number of parameters is large, then they will be badly determined when data is few.
- For me, by *definition* all interesting problems have complex models.

Not Wrong ... Just Useless

- Here's a model that's not wrong ...

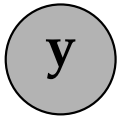
Not Wrong ... Just Useless

- Here's a (graphical) model that's not wrong ...



Not Wrong ... Just Useless

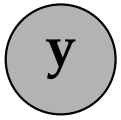
- Here's a model that's not wrong ...



... it's just useless.

Not Wrong ... Just Useless

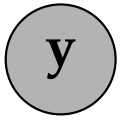
- Here's a model that's not wrong ...



  ... it's just useless.
- Does that imply all models that are not wrong are useless?

Not Wrong ... Just Useless

- Here's a model that's not wrong ...



  ... it's just useless.
- Does that imply all models that are not wrong are useless?
- What is the minimum we can say about our data to get something useful?

Outline

Box Quote

Nonparametrics

Flexible Parametric Approximation

Conclusions

The TT Channel

- Objective: predict test data, $\mathbf{y}^*$, given training data, $\mathbf{y}$.
- Parametric models assume

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

  for some fixed dimensional vector parameters $\boldsymbol{\theta}$.
- This looks like a communication channel between training and test data (TT Channel).
- Capacity of channel given by dimensionality of $\boldsymbol{\theta}$.

Massively Missing Data

- Michael Goldstein's Maid (via Tony O'Hagan).
- Let me tell you something unusual about myself ...
- Large amounts of weak information can give a strong picture.
- But we must deal with uncertainty when this info isn't present.
- In real life almost all data is missing almost always.

Kolmogorov Consistency

- **Claim:** To be 'not wrong' my model must be 'Kolmogorov Consistent'.

Kolmogorov Consistency

- **Claim:** To be 'not wrong' my model must be 'Kolmogorov Consistent'.
- Kolmogorov consistency says regardless of future observations, my current marginal model of the data is correct. If $\mathbf{y}^* \in \Re^{n^* \times 1}$ then

$$p(\mathbf{y}|n^*) = \int p(\mathbf{y}, \mathbf{y}^*) \mathrm{d}\mathbf{y}^*$$

But if the model is Kolmogorov consistent, $p(\mathbf{y}|n^*) = p(\mathbf{y})$.

Kolmogorov Consistency

- **Claim:** To be 'not wrong' my model must be 'Kolmogorov Consistent'.
- Kolmogorov consistency says regardless of future observations, my current marginal model of the data is correct. If $\mathbf{y}^* \in \Re^{n^* \times 1}$ then

$$p(\mathbf{y}|n^*) = \int p(\mathbf{y}, \mathbf{y}^*) \mathrm{d}\mathbf{y}^*$$

  But if the model is Kolmogorov consistent, $p(\mathbf{y}|n^*) = p(\mathbf{y})$.
- Here: $\mathbf{y}$ is past observations, $\mathbf{y}^*$ is all possible *future* observations (in either p or n).
- Models of this type allow us to deal with *massive* missing data because $\mathbf{y}^*$ can even be infinite dimensional.
- To these models missing data is equivalent to test data.

Nonparametric TT Channel

- In a non parametric model:

$$p(\mathbf{y}^*|\mathbf{y})$$

Cannot be written as

$$\int p(\mathbf{y}^*|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{y})\mathrm{d}\boldsymbol{\theta}$$

for fixed dimensional $\boldsymbol{\theta}$.

The TT Channel

- Objective: predict test data, $\mathbf{y}^*$, given training data, $\mathbf{y}$.
- Parametric models assume

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

  for some fixed dimensional vector parameters $\boldsymbol{\theta}$.
- This looks like a communication channel between training and test data (TT Channel).
- Capacity of channel given by dimensionality of $\boldsymbol{\theta}$.

Open Data

- Automatic data curation: from curated data to curation of publicly available data.
- Open Data: `http://www.openstreetmap.org/?lat=53.38086&lon=-1.48545&zoom=17&layers=M`.

Open Data

- Automatic data curation: from curated data to curation of publicly available data.
- Open Data: `http://www.openstreetmap.org/?lat=53.38086&lon=-1.48545&zoom=17&layers=M`.

Open Data

- Automatic data curation: from curated data to curation of publicly available data.
- Open Data: `http://www.openstreetmap.org/?lat=53.38086&lon=-1.48545&zoom=17&layers=M`.

- Social network data, music information (Spotify), exercise.

Outline

Box Quote

Nonparametrics

Flexible Parametric Approximation

Conclusions

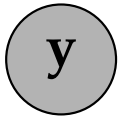
Inducing Variable Approximations

- Date back to (Williams and Seeger, 2001; Smola and Bartlett, 2001; Csató and Opper, 2002; Seeger et al., 2003; Snelson and Ghahramani, 2006). See Quiñonero Candela and Rasmussen (2005) for a review.
- We follow variational perspective of (Titsias, 2009).
- This is an augmented variable method, followed by a collapsed variational approximation (King and Lawrence, 2006; Hensman et al., 2012).

Augmented Variable Model: Not Wrong but Useful?

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

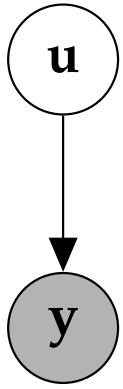
$$p(\mathbf{y}) = \int p(\mathbf{y}, \mathbf{u}) \mathrm{d}\mathbf{u}$$



Augmented Variable Model: Not Wrong but Useful?

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

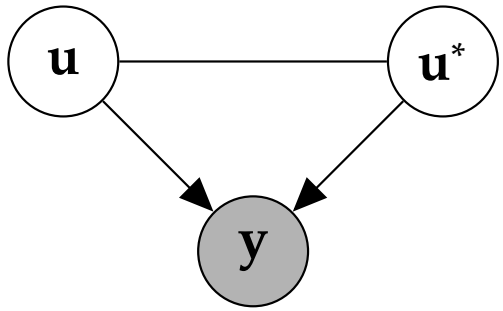
$$p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{u})p(\mathbf{u})\mathrm{d}\mathbf{u}$$



Augmented Variable Model: Not Wrong but Useful?

Important: Ensure inducing variables are *also* Kolmogorov consistent (we have m^* other inducing variables we are not *yet* using.)

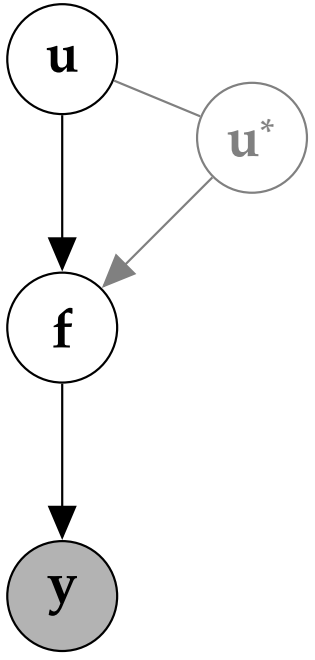
$$p(\mathbf{u}) = \int p(\mathbf{u}, \mathbf{u}^*) \mathrm{d}\mathbf{u}^*$$



Augmented Variable Model: Not Wrong but Useful?

Assume that relationship is through **f** (represents 'fundamentals'—push Kolmogorov consistency up to here).

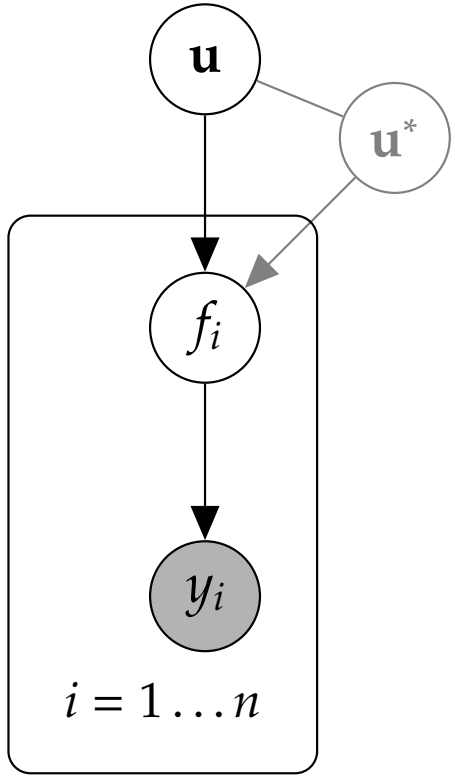
$$p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{u})p(\mathbf{u})\mathrm{d}\mathbf{f}\mathrm{d}\mathbf{u}$$



Augmented Variable Model: Not Wrong but Useful?

Convenient to assume factorization (*doesn't* invalidate model—think delta function as worst case).

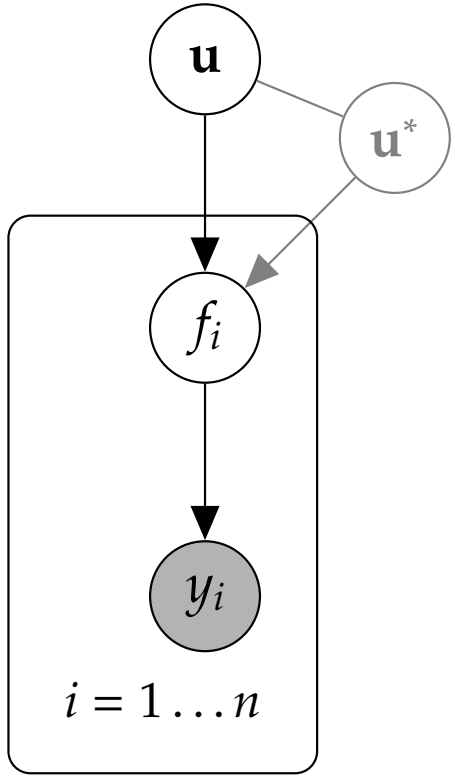
$$p(\mathbf{y}) = \int \prod_{i=1}^{n} p(y_i|f_i)p(\mathbf{f}|\mathbf{u})p(\mathbf{u})\mathrm{d}\mathbf{f}\mathrm{d}\mathbf{u}$$



Augmented Variable Model: Not Wrong but Useful?

Focus on integral over $\mathbf{f}$.

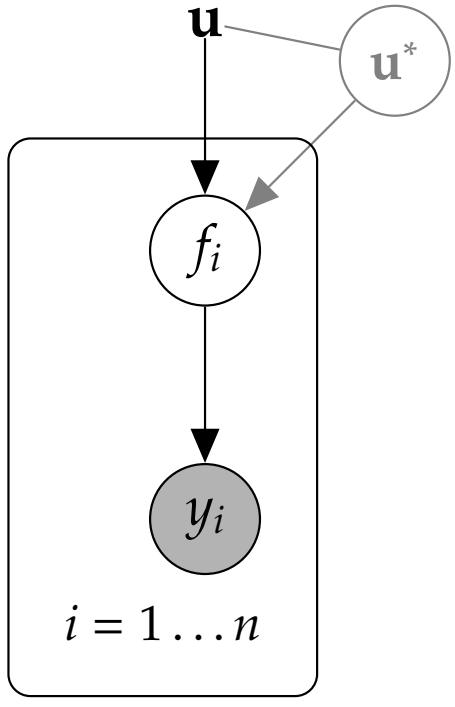
$$p(\mathbf{y}) = \int \int \prod_{i=1}^{n} p(y_i|f_i)p(\mathbf{f}|\mathbf{u})\mathrm{d}\mathbf{f}p(\mathbf{u})\mathrm{d}\mathbf{u}$$



Augmented Variable Model: Not Wrong but Useful?

Focus on integral over $\mathbf{f}$.

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



Variational Bound on $p(\mathbf{y}|\mathbf{u})$

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

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

Variational Bound on $p(\mathbf{y}|\mathbf{u})$

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

(Titsias, 2009)

- Example, set $q(\mathbf{f}) = p(\mathbf{f}|\mathbf{u})$,

$$\log p(\mathbf{y}|\mathbf{u}) \geq \log \int p(\mathbf{f}|\mathbf{u}) \log p(\mathbf{y}|\mathbf{f})\mathrm{d}\mathbf{f}.$$

$$p(\mathbf{y}|\mathbf{u}) \geq \exp \int p(\mathbf{f}|\mathbf{u}) \log p(\mathbf{y}|\mathbf{f})\mathrm{d}\mathbf{f}.$$

Optimal Compression in Inducing Variables

- Maximizing lower bound minimizes the KL divergence (information gain):

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

- This is minimized when the information stored about $\mathbf{y}$ is stored already in $\mathbf{u}$.
- The bound seeks an *optimal compression* from the *information gain* perspective.
- If $\mathbf{u} = \mathbf{f}$ bound is exact ($\mathbf{f}$ d-separates $\mathbf{y}$ from $\mathbf{u}$).

Choice of Inducing Variables

- Optimizing the bound directly not always practical.
- Free to choose whatever heuristics for the inducing variables.
- Can quantify which heuristics perform better through checking lower bound.

Factorizing Likelihoods

- If the likelihood, $p(\mathbf{y}|\mathbf{f})$, factorizes

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

Factorizing Likelihoods

- If the likelihood, $p(\mathbf{y}|\mathbf{f})$, factorizes

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

Factorizing Likelihoods

- If the likelihood, $p(\mathbf{y}|\mathbf{f})$, factorizes

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

Factorizing Likelihoods

- If the likelihood, $p(\mathbf{y}|\mathbf{f})$, factorizes

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

Factorizing Likelihoods

- If the likelihood, $p(\mathbf{y}|\mathbf{f})$, factorizes

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

Factorizing Likelihoods

- If the likelihood, $p(\mathbf{y}|\mathbf{f})$, factorizes

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

Factorizing Likelihoods

- If the likelihood, $p(\mathbf{y}|\mathbf{f})$, factorizes

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

Factorizing Likelihoods

- If the likelihood, $p(\mathbf{y}|\mathbf{f})$, factorizes

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

- Then the bound factorizes.

Factorizing Likelihoods

- If the likelihood, $p(\mathbf{y}|\mathbf{f})$, factorizes

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

- Then the bound factorizes.

Factorizing Likelihoods

- If the likelihood, $p(\mathbf{y}|\mathbf{f})$, factorizes

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

- Then the bound factorizes.
- Now need a choice of distributions for $\mathbf{f}$ and $\mathbf{y}|\mathbf{f}$ ...

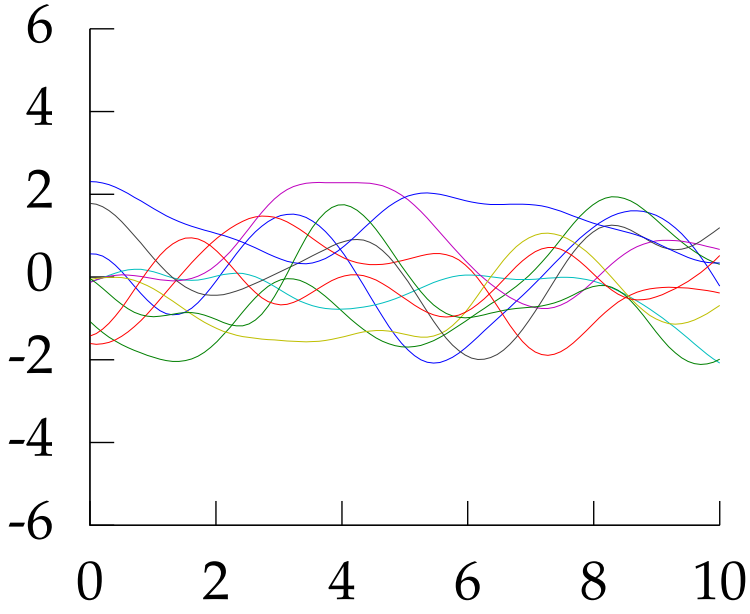
Potential Implications

- Distributed Models
  - Can we distribute inducing variables across different machines (in different continents!?).
  - Approximate the model appropriately according to where we are and what we are predicting.
- Resource Allocation
  - Be able to improve the resolution of the predictive model at *run time*.
  - Allocate resources *heuristically*.
- Retain the principled probabilistic framework for the global model.

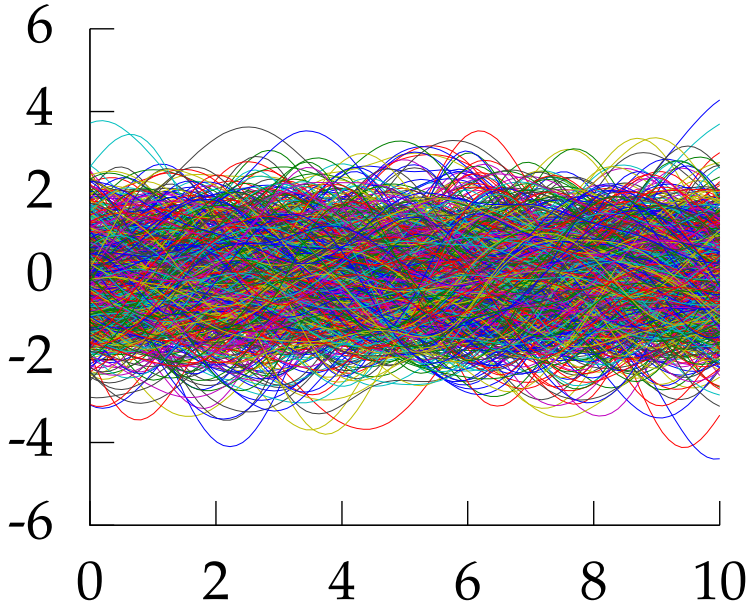
Prior and Likelihood Choice

- Choose a Gaussian process prior for $\mathbf{f}$.
  - This is not always correct, have a need for more flexible priors ... see Deep GPs (Damianou and Lawrence, 2013).
- Choose a factorized Gaussian likelihood for $\mathbf{y}|\mathbf{f}$.
  - Gaussian assumption can also be relaxed (Hensman et al., 2014).

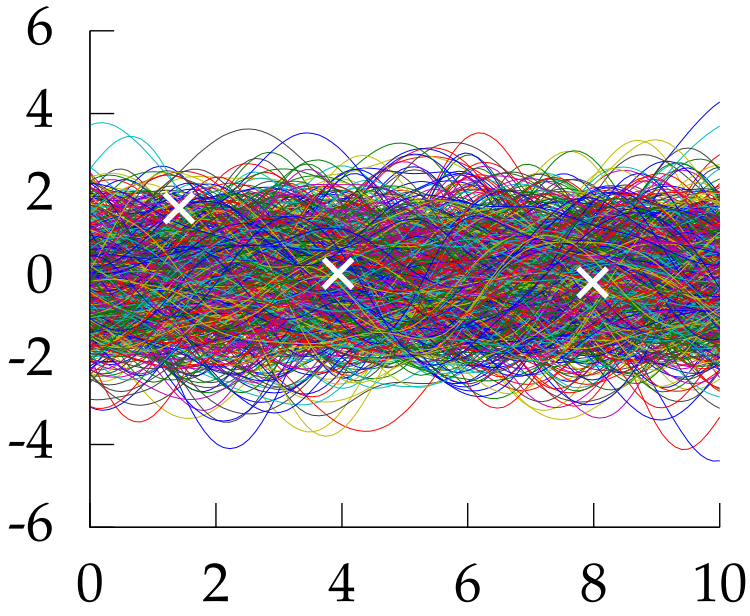
Gaussian Processes: Extremely Short Overview



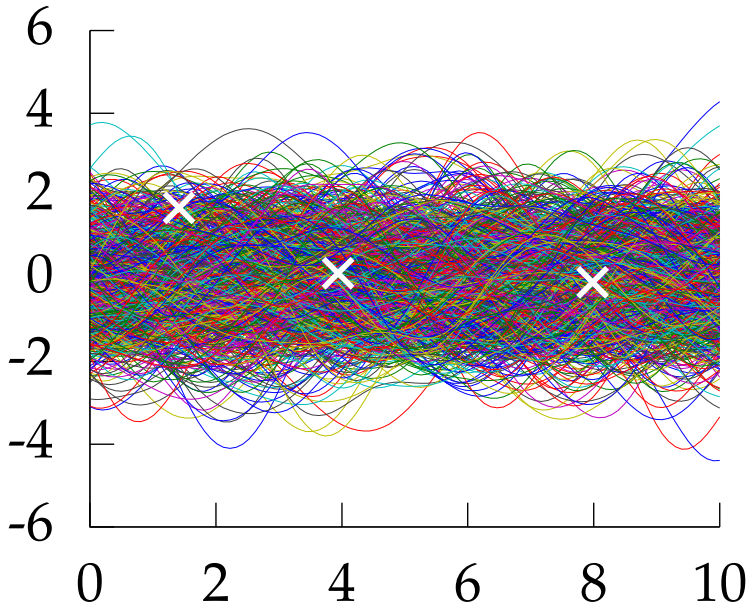
Gaussian Processes: Extremely Short Overview

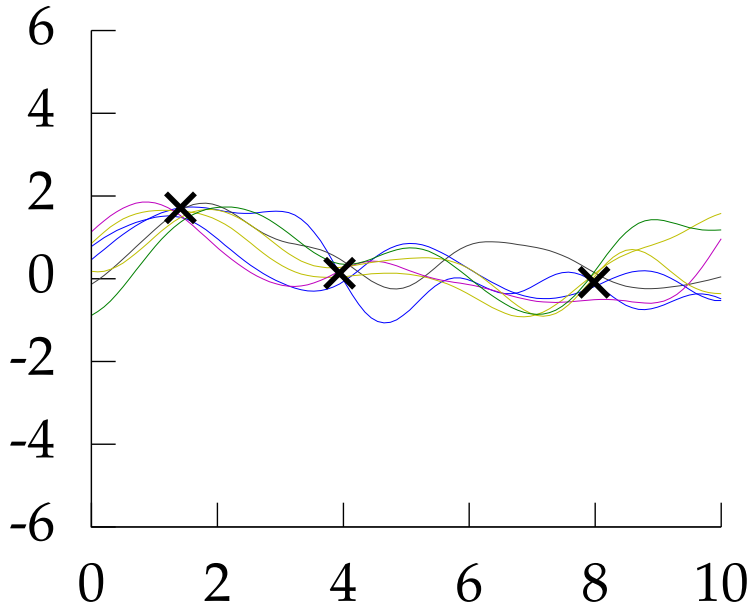


Gaussian Processes: Extremely Short Overview



Gaussian Processes: Extremely Short Overview





Reducing GP Complexity

(Lawrence, 2007; Titsias, 2009)

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

Reducing GP Complexity

(Lawrence, 2007; Titsias, 2009)

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

Reducing GP Complexity

(Lawrence, 2007; Titsias, 2009)

- Complexity of standard GP:
  - $O(n^3)$ in computation.
  - $O(n^2)$ in storage.
- Via low rank representations of covariance:
  - $O(nm^2)$ in computation.
  - $O(nm)$ in storage.
- Where m is user chosen number of *inducing* variables. They give the rank of the resulting covariance.
- Inducing variables live either in space of $\mathbf{f}$ or a space that is related through a linear operator (Álvarez et al., 2010) — could be gradient or convolution.

Gaussian $p(y_i|f_i)$

For Gaussian likelihoods:

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

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:

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

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).

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 ...

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

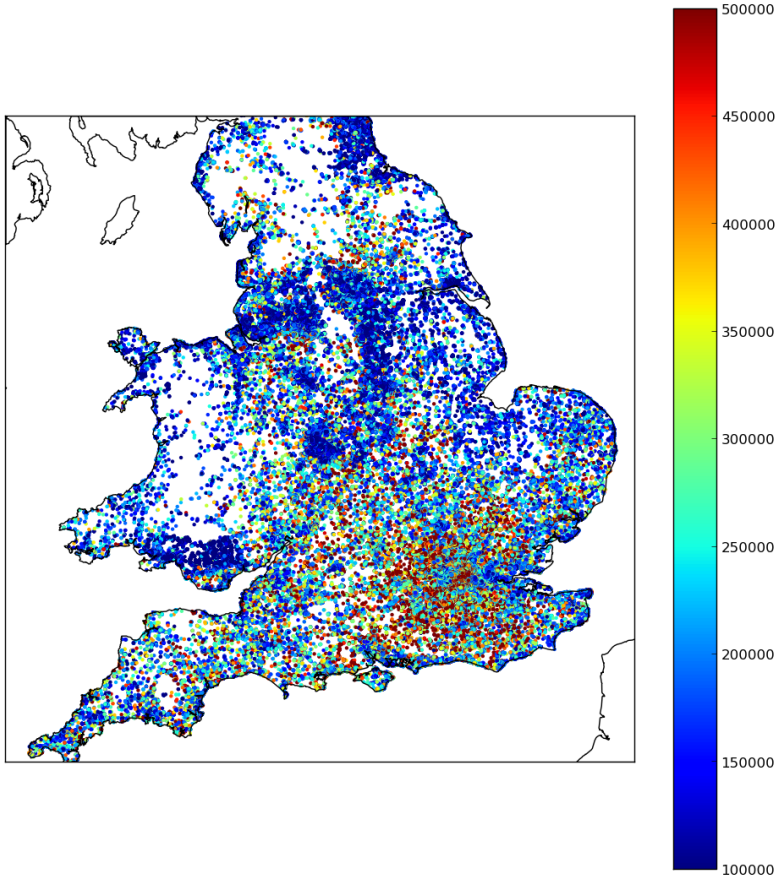
- The distribution now *factorizes*:

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

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

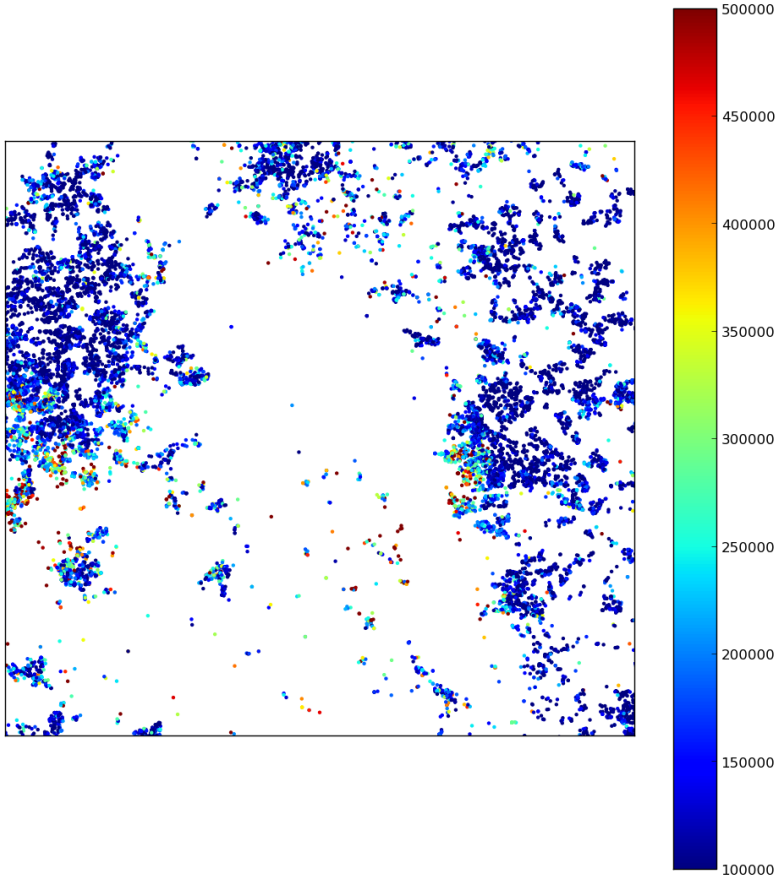
Nonparametrics for Very Large Data Sets

Modern data availability



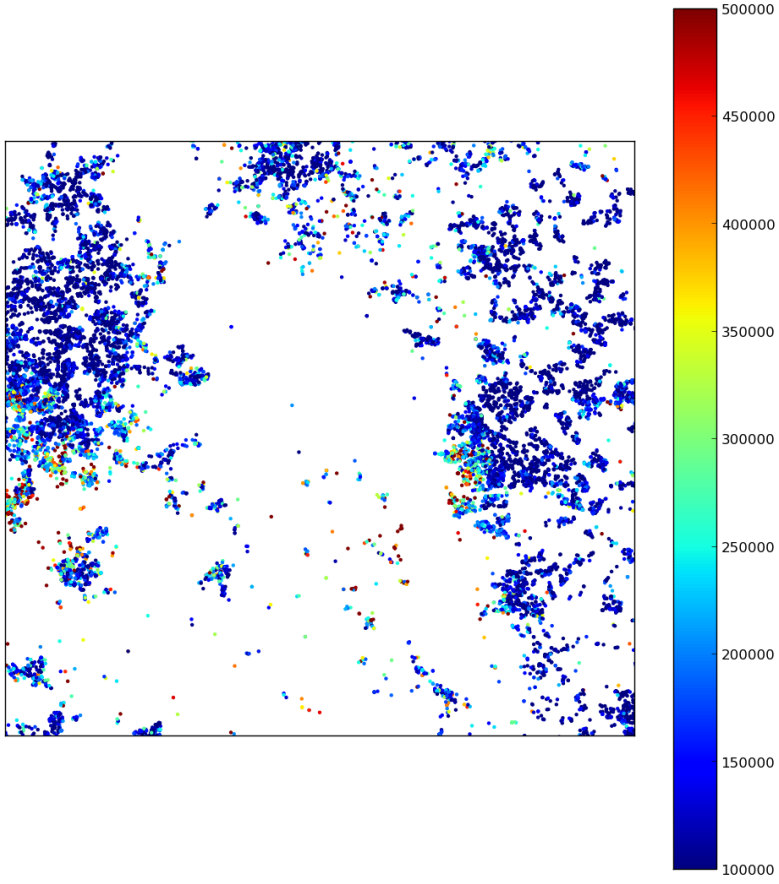
Nonparametrics for Very Large Data Sets

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)

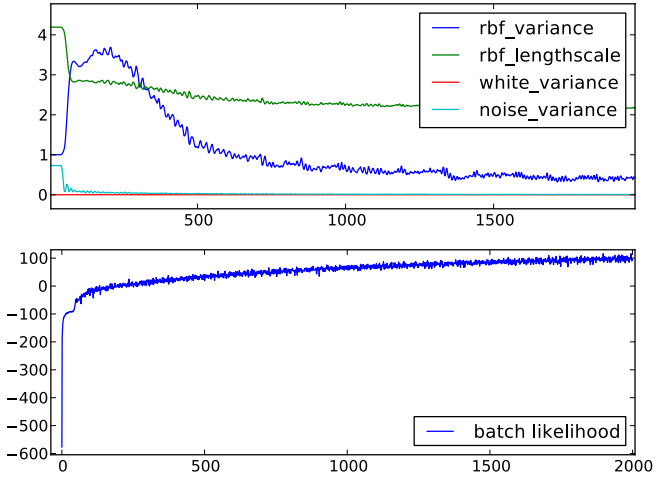


Figure 4: Convergence of the SVIGP algorithm on the two dimensional toy data

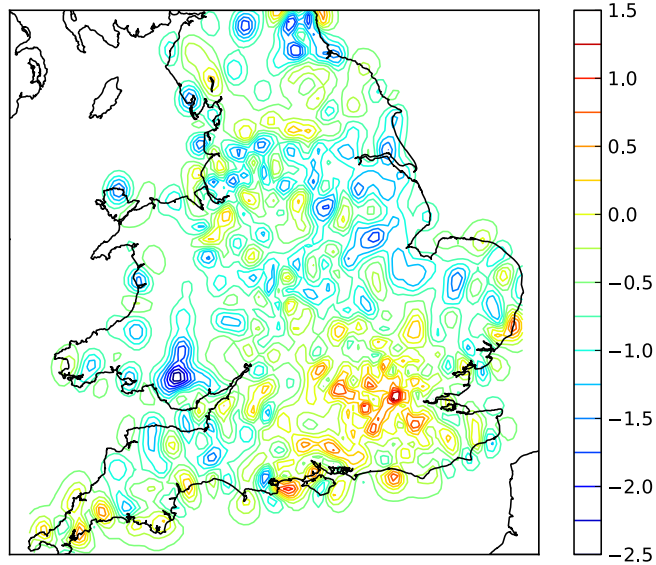
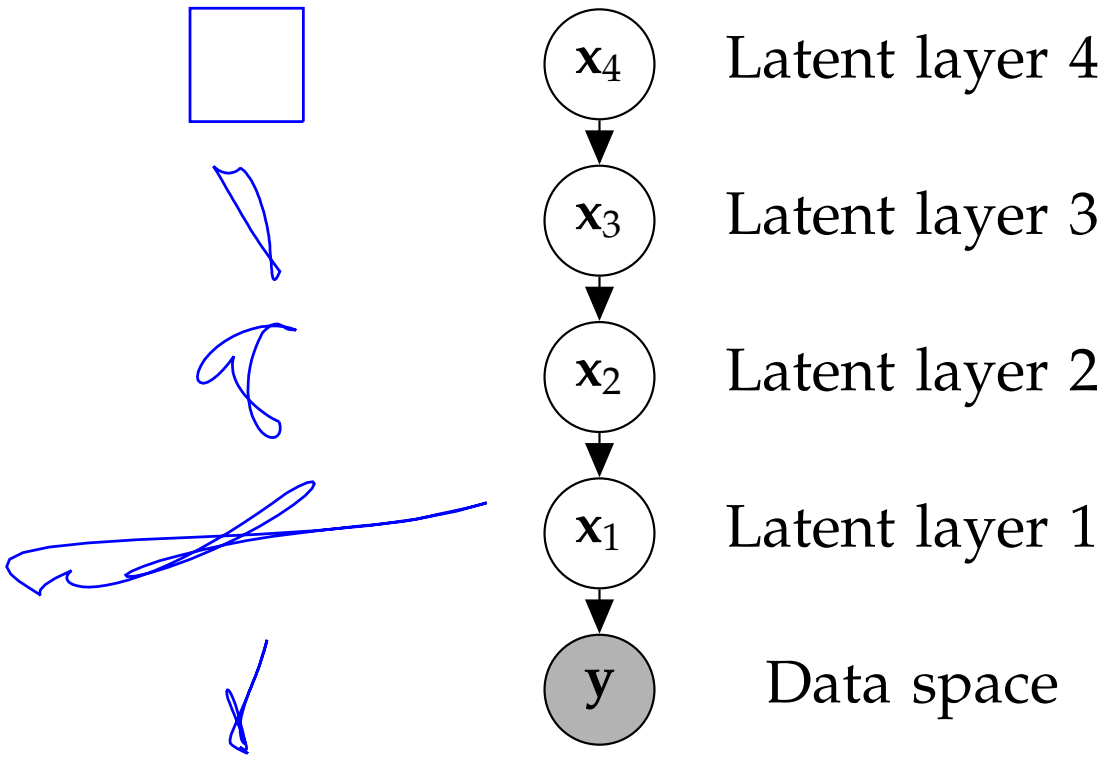


Figure 5: Variability of apartment price (logarithmically!) throughout England and Wales.

`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

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

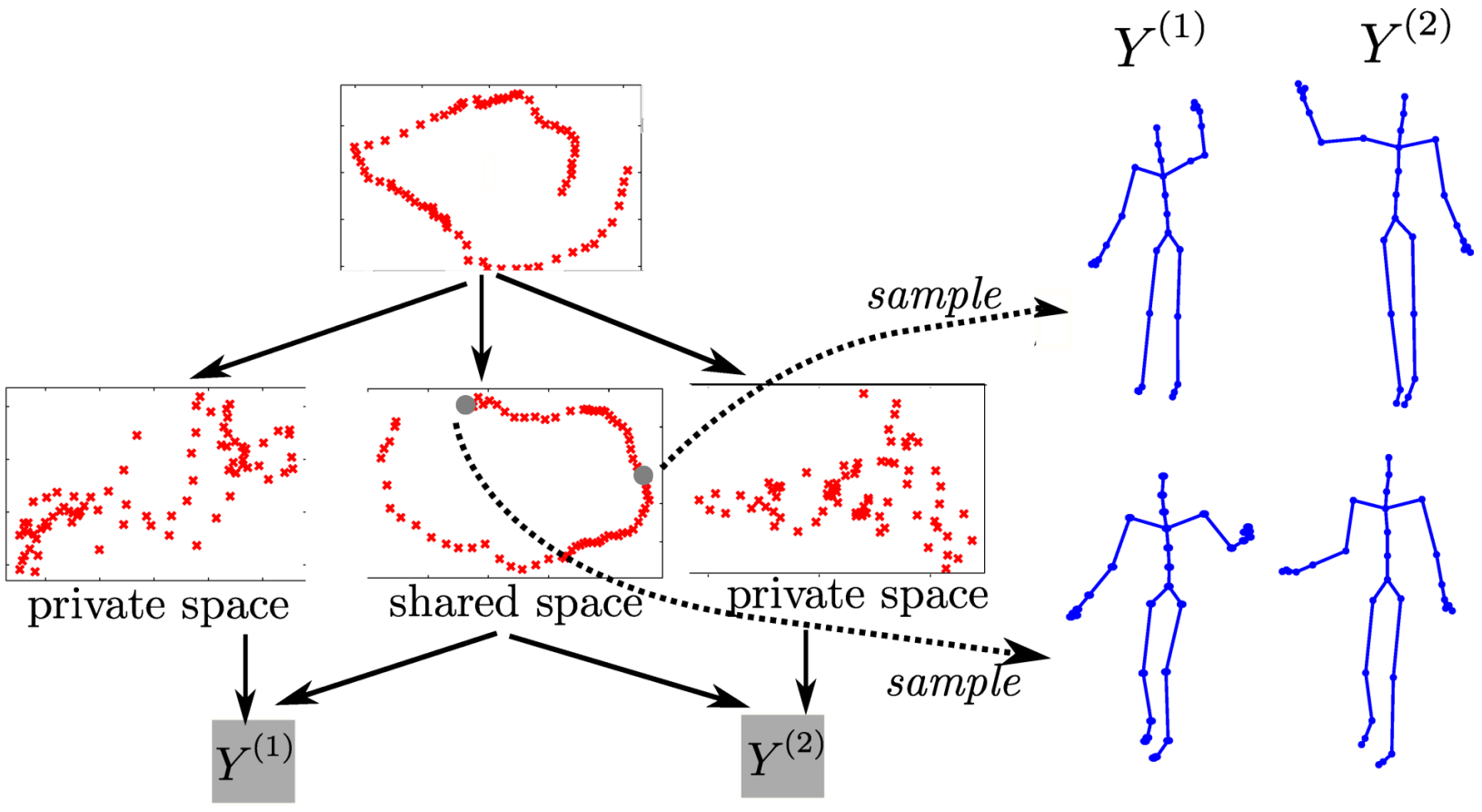
the question as to whether deep structures and the learning of abstract structure can be undertaken in *smaller* data sets. For smaller data sets, questions of generalization arise: to demonstrate such structures are justified it is useful to have an objective measure of the model's applicability.

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)

Motion Capture

- ‘High five’ data.
- Model learns structure between two interacting subjects.

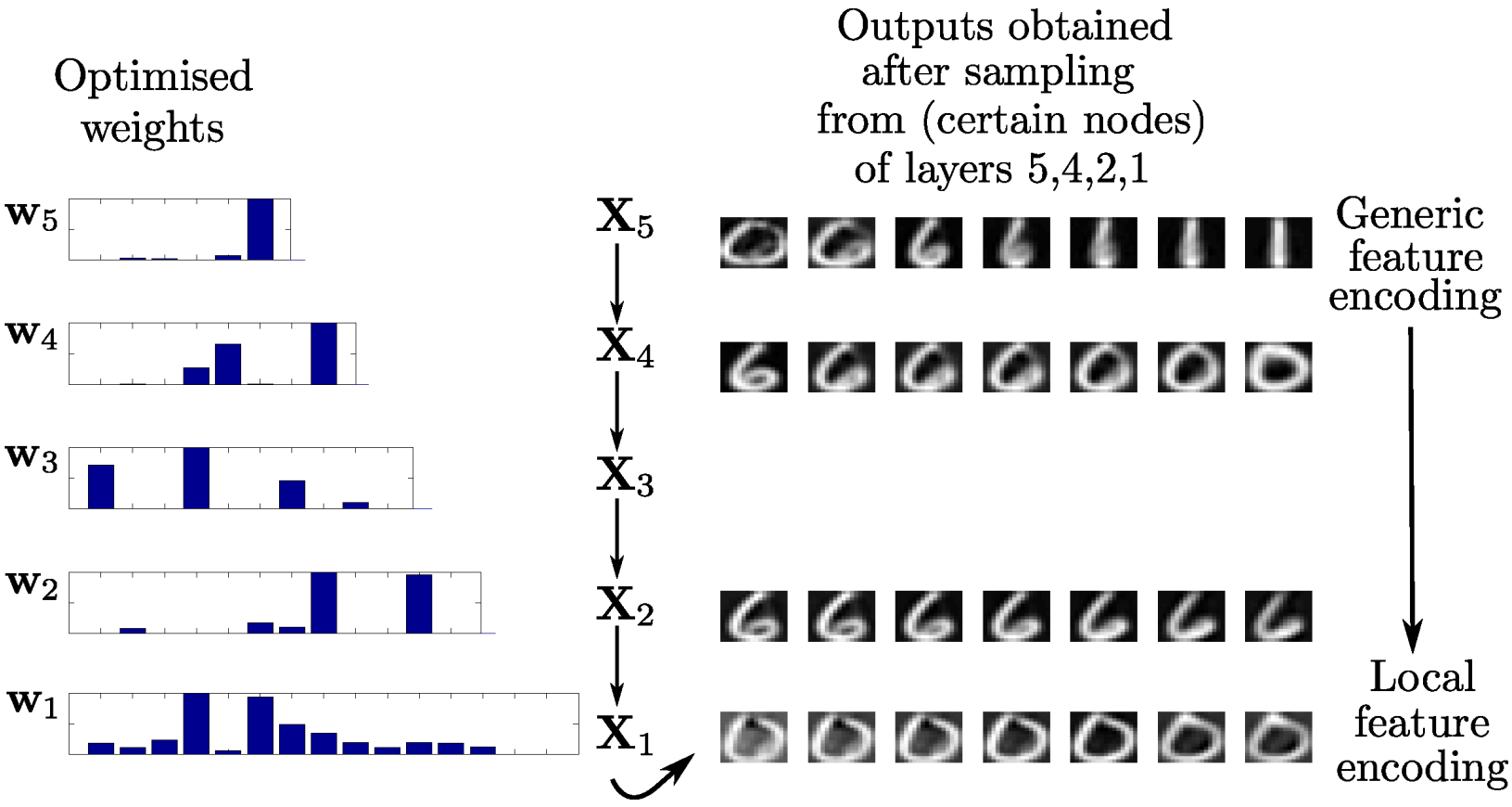
Deep hierarchies – motion capture



Digits Data Set

- Are deep hierarchies justified for small data sets?
- We can lower bound the evidence for different depths.
- For 150 6s, 0s and 1s from MNIST we found at least 5 layers are required.

Deep hierarchies – MNIST



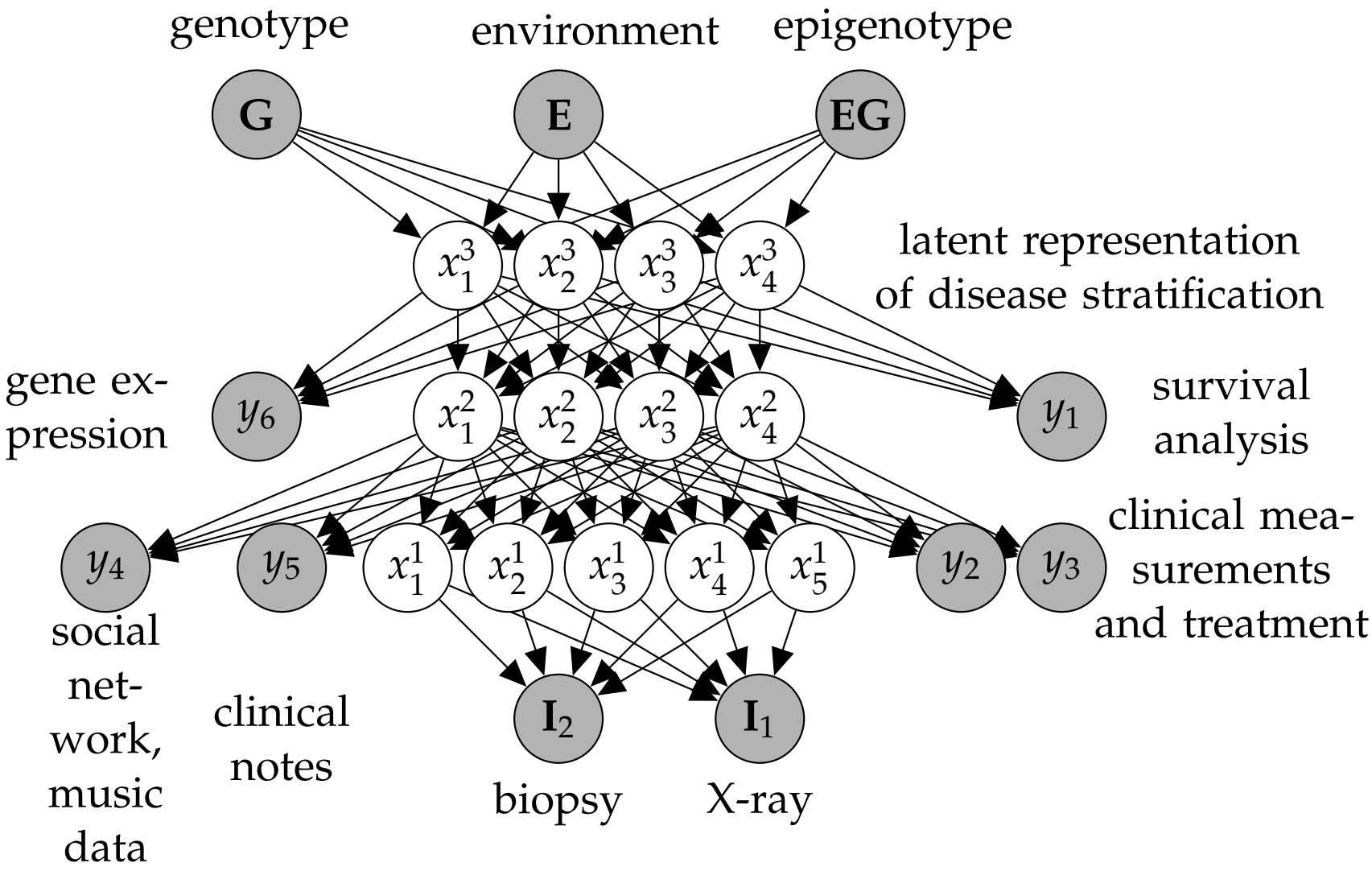
Practical Nonparametric Modelling

- 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.

Important Concepts

- Kolmogorov consistency.
- The bandwidth of the TT channel.
- Models which are complex enough (non parametrics).
- But have parametric approximations that can be adapted at runtime.

Deep Health


genotype
environment
epigenotype
G
E
EG
x_1^3 x_2^3 x_3^3 x_4^3
latent representation of disease stratification
gene expression
y_6
x_1^2 x_2^2 x_3^2 x_4^2
y_1
survival analysis
y_4
y_5
x_1^1 x_2^1 x_3^1 x_4^1 x_5^1
y_2
y_3
clinical measurements and treatment
social network, music data
clinical notes
I_2
I_1
biopsy
X-ray

Deep Health: Power Ranger Model of Research

Thanks to Alan Saul for creating the image.

Summary

- Deep models allow abstract representation of data sets at higher levels.
- Deep GPs allow structure learning.
- Current limitation is on data set size.
- Addressing this through work by James Hensman on Stochastic Variational Inference for GPs (Hensman et al., 2013).
- Intention is to deploy these models for assimilating a wide range of data types in personalized health (text, survival times, images, genotype, phenotype).
- Requires population scale models with millions of features.

References I

M. A. Álvarez, D. Luengo, M. K. Titsias, and N. D. Lawrence. Efficient multioutput Gaussian processes through variational inducing kernels. In Y. W. Teh and D. M. Titterington, editors, *Proceedings of the Thirteenth International Workshop on Artificial Intelligence and Statistics*, volume 9, pages 25–32, Chia Laguna Resort, Sardinia, Italy, 13-16 May 2010. JMLR W&CP 9. [PDF].

G. E. P. Box. Science and statistics. *Journal of the American Statistical Association*, 71(365), 1976.

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].

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].

J. Hensman, M. Rattray, and N. D. Lawrence. Fast variational inference in the conjugate exponential family. In P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25, Cambridge, MA, 2012. [PDF].

J. Hensman, M. Zwiessele, and N. D. Lawrence. Tilted variational Bayes. In S. Kaski and J. Corander, editors, *Proceedings of the Seventeenth International Workshop on Artificial Intelligence and Statistics*, volume 33, Iceland, 2014. JMLR W&CP 33.

M. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *arXiv preprint arXiv:1206.7051*, 2012.

N. J. King and N. D. Lawrence. Fast variational inference for Gaussian Process models through KL-correction. In *ECML, Berlin, 2006*, Lecture Notes in Computer Science, pages 270–281, Berlin, 2006. Springer-Verlag. [PDF].

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.

References II

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.

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.

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.

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