Variational Inference for Gaussian and Determinantal Point Processes

Michalis K. Titsias
Department of Informatics,
Athens University of Economics and Business, Greece

Motivation

The general setting: We are interested in applying variational inference to Bayesian non-parametric models where the number of parameters grows with the number of data

Motivation

The general setting: We are interested in applying variational inference to Bayesian non-parametric models where the number of parameters grows with the number of data

The challenge: How can we variationally approximate/represent infinite posteriors?

Motivation

The general setting: We are interested in applying variational inference to Bayesian non-parametric models where the number of parameters grows with the number of data

The challenge: How can we variationally approximate/represent infinite posteriors?

In this talk: We will present a variational method that has been developed for Gaussian process models and then extend it to determinantal point processes

Gaussian process regression

Inputs $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ and outputs $\mathbf{y} = (y_1, \ldots, y_n)$ such that

$$y_i = f(\mathbf{x}_i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

Gaussian process regression

Inputs $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ and outputs $\mathbf{y} = (y_1, \ldots, y_n)$ such that

$$y_i = f(\mathbf{x}_i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

Place GP prior on latent function $f(\mathbf{x})$:

$$f(\mathbf{x}) \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}')),$$

Gaussian process regression

Inputs $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ and outputs $\mathbf{y} = (y_1, \ldots, y_n)$ such that

$$y_i = f(\mathbf{x}_i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

Place GP prior on latent function $f(\mathbf{x})$:

$$f(\mathbf{x}) \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}')),$$

Given that we have n data our current "marginal model" is

$$p(\mathbf{y}|\mathbf{f})p(\mathbf{f}) = \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma^2 I)\mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K}_{ff}), \quad [\mathbf{K}_{ff}]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$$

where $\mathbf{f} = (f_1, \ldots, f_n)$ are the parameters

Gaussian process regression

and now the problem appears: as we keep collecting more data the size of $\mathbf{f} = (f_1, f_2, f_3, \ldots)$ increases and the kernel matrix gets bigger and bigger

$$\mathbf{K}_{ff} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & k(\mathbf{x}_1, \mathbf{x}_3) & \ldots \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & k(\mathbf{x}_2, \mathbf{x}_3) & \ldots \\ k(\mathbf{x}_3, \mathbf{x}_1) & k(\mathbf{x}_3, \mathbf{x}_2) & k(\mathbf{x}_3, \mathbf{x}_3) & \ldots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Gaussian process regression

and now the problem appears: as we keep collecting more data the size of $\mathbf{f} = (f_1, f_2, f_3, \ldots)$ increases and the kernel matrix gets bigger and bigger

$$\mathbf{K}_{ff} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & k(\mathbf{x}_1, \mathbf{x}_3) & \ldots \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & k(\mathbf{x}_2, \mathbf{x}_3) & \ldots \\ k(\mathbf{x}_3, \mathbf{x}_1) & k(\mathbf{x}_3, \mathbf{x}_2) & k(\mathbf{x}_3, \mathbf{x}_3) & \ldots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Space scales as $O(n^2)$ and time as $O(n^3)$

Gaussian process regression

and now the problem appears: as we keep collecting more data the size of $\mathbf{f} = (f_1, f_2, f_3, \ldots)$ increases and the kernel matrix gets bigger and bigger

$$\mathbf{K}_{ff} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & k(\mathbf{x}_1, \mathbf{x}_3) & \ldots \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & k(\mathbf{x}_2, \mathbf{x}_3) & \ldots \\ k(\mathbf{x}_3, \mathbf{x}_1) & k(\mathbf{x}_3, \mathbf{x}_2) & k(\mathbf{x}_3, \mathbf{x}_3) & \ldots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Space scales as $O(n^2)$ and time as $O(n^3)$

Thus GP computations, e.g. learning by maximizing the marginal likelihood

$$p(\mathbf{y}) = \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f})d\mathbf{f} = \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_{ff} + \sigma^2 I)$$

are not feasible for very large n

Inducing variables

The problem is that $\mathbf{f}$ grows as we collect more data

Inducing variables

The problem is that $\mathbf{f}$ grows as we collect more data

Idea: Summarize/replace $\mathbf{f}$ by a smaller parameter vector $\mathbf{u}$

Inducing variables

The problem is that $\mathbf{f}$ grows as we collect more data

Idea: Summarize/replace $\mathbf{f}$ by a smaller parameter vector $\mathbf{u}$

The size of $\mathbf{u}$ must be user-controllable based on current computational resources

- it could grow if the computational capacity increase in future

Inducing variables

The problem is that $\mathbf{f}$ grows as we collect more data

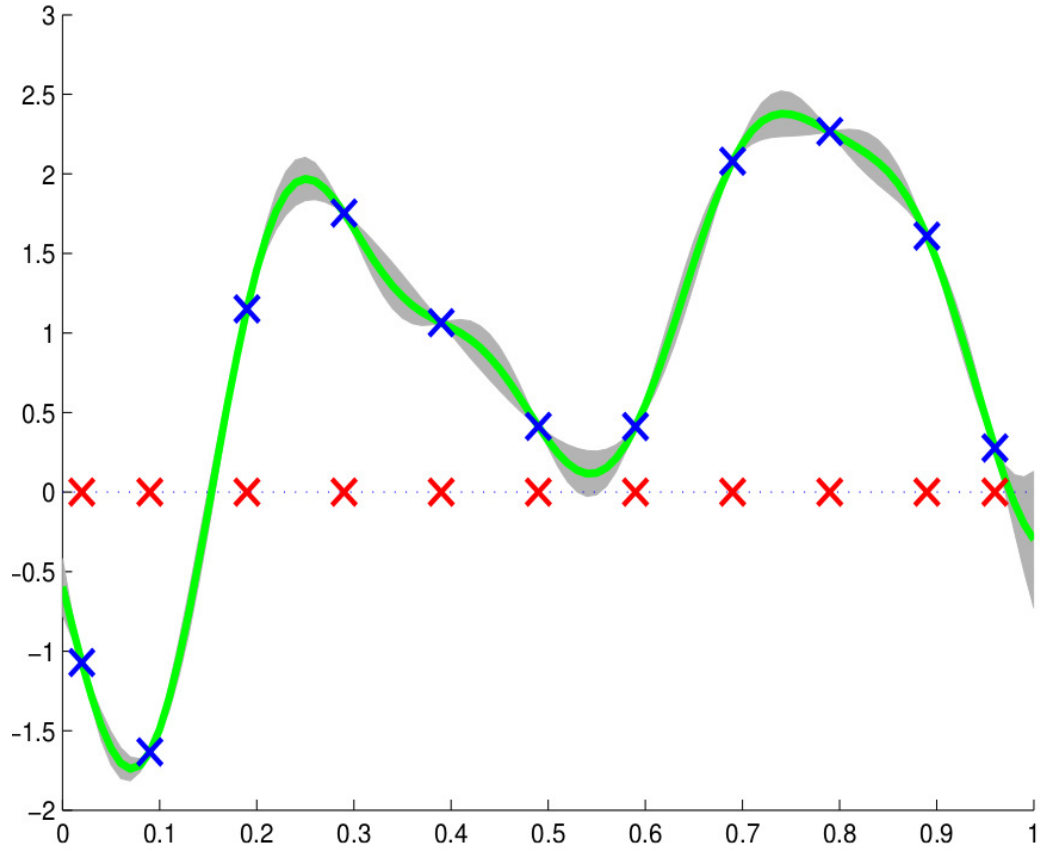
Idea: Summarize/replace $\mathbf{f}$ by a smaller parameter vector $\mathbf{u}$

The size of $\mathbf{u}$ must be user-controllable based on current computational resources

- it could grow if the computational capacity increase in future

Obviously how $\mathbf{u}$ is going to be defined and optimized is crucial

Inducing variables



A realization of a full (infinite) GP function/sample path

Summarize with a discrete set of function values $\mathbf{u} = (u_1, \ldots, u_m)$

and some uncertainty for the intermediate points

Inducing variables

Inducing variables $\mathbf{u}$ form a vector of user-controllable size that augments the GP prior:

$$p(\mathbf{f}, \mathbf{u}) = \mathcal{N}\left(\begin{bmatrix} \mathbf{f} \\ \mathbf{u} \end{bmatrix} \middle| \mathbf{0}, \begin{bmatrix} \mathbf{K}_{ff} & \mathbf{K}_{fu} \\ \mathbf{K}_{uf} & \mathbf{K}_{uu} \end{bmatrix} \right), \mathbf{K}_{fu} = \mathbb{E}[\mathbf{f}\mathbf{u}^T], \mathbf{K}_{uu} = \mathbb{E}[\mathbf{u}\mathbf{u}^T]$$

.gr/

Inducing variables

Inducing variables $\mathbf{u}$ form a vector of user-controllable size that augments the GP prior:

$$p(\mathbf{f}, \mathbf{u}) = \mathcal{N}\left(\begin{bmatrix} \mathbf{f} \\ \mathbf{u} \end{bmatrix} \middle| \mathbf{0}, \begin{bmatrix} \mathbf{K}_{ff} & \mathbf{K}_{fu} \\ \mathbf{K}_{uf} & \mathbf{K}_{uu} \end{bmatrix} \right), \mathbf{K}_{fu} = \mathbb{E}[\mathbf{f}\mathbf{u}^T], \mathbf{K}_{uu} = \mathbb{E}[\mathbf{u}\mathbf{u}^T]$$

.gr/

$\mathbf{u}$ can be:

- a subset of $\mathbf{f}$
- values of $f(\mathbf{x})$ at arbitrary "pseudo-inputs"
- arbitrary linear functionals, e.g. $u = z_i f(\mathbf{x}_i) + z_j f(\mathbf{x}_j)$

Inducing variables

Inducing variables $\mathbf{u}$ form a vector of user-controllable size that augments the GP prior:

$$p(\mathbf{f}, \mathbf{u}) = \mathcal{N}\left(\begin{bmatrix} \mathbf{f} \\ \mathbf{u} \end{bmatrix} \middle| \mathbf{0}, \begin{bmatrix} \mathbf{K}_{ff} & \mathbf{K}_{fu} \\ \mathbf{K}_{uf} & \mathbf{K}_{uu} \end{bmatrix} \right), \mathbf{K}_{fu} = \mathbb{E}[\mathbf{f}\mathbf{u}^T], \mathbf{K}_{uu} = \mathbb{E}[\mathbf{u}\mathbf{u}^T]$$

.gr/

$\mathbf{u}$ can be:

- a subset of $\mathbf{f}$
- values of $f(\mathbf{x})$ at arbitrary "pseudo-inputs"
- arbitrary linear functionals, e.g. $u = z_i f(\mathbf{x}_i) + z_j f(\mathbf{x}_j)$

The augmentation with $\mathbf{u}$ adds some parameters Z

- indices that specify the subset in $\mathbf{f}$, pseudo-inputs, weights etc
- $\mathbf{K}_{fu}$ and $\mathbf{K}_{uu}$ depend on those parameters

Inducing variables

We have

$$p(\mathbf{f}, \mathbf{u}) = \mathcal{N}\left(\left[\begin{array}{c} \mathbf{f} \\ \mathbf{u} \end{array}\right] \middle| \mathbf{0}, \left[\begin{array}{cc} \mathbf{K}_{ff} & \mathbf{K}_{fu} \\ \mathbf{K}_{uf} & \mathbf{K}_{uu} \end{array}\right]\right) = p(\mathbf{f}|\mathbf{u})p(\mathbf{u})$$

Inducing variables

We have

$$p(\mathbf{f}, \mathbf{u}) = \mathcal{N}\left(\begin{bmatrix} \mathbf{f} \\ \mathbf{u} \end{bmatrix} \middle| \mathbf{0}, \begin{bmatrix} \mathbf{K}_{ff} & \mathbf{K}_{fu} \\ \mathbf{K}_{uf} & \mathbf{K}_{uu} \end{bmatrix}\right) = p(\mathbf{f}|\mathbf{u})p(\mathbf{u})$$

where

$$p(\mathbf{f}|\mathbf{u}) = \mathcal{N}(\mathbf{f}|\mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{u}, \mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}) \quad \text{conditional GP prior}$$

$$p(\mathbf{u}) = \mathcal{N}(\mathbf{u}|\mathbf{0}, \mathbf{K}_{uu}) \quad \text{marginal over } \mathbf{u}$$

Inducing variables

We have

$$p(\mathbf{f}, \mathbf{u}) = \mathcal{N}\left(\begin{bmatrix} \mathbf{f} \\ \mathbf{u} \end{bmatrix} \middle| \mathbf{0}, \begin{bmatrix} \mathbf{K}_{ff} & \mathbf{K}_{fu} \\ \mathbf{K}_{uf} & \mathbf{K}_{uu} \end{bmatrix}\right) = p(\mathbf{f}|\mathbf{u})p(\mathbf{u})$$

where

$$p(\mathbf{f}|\mathbf{u}) = \mathcal{N}(\mathbf{f}|\mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{u}, \mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}) \quad \text{conditional GP prior}$$

$$p(\mathbf{u}) = \mathcal{N}(\mathbf{u}|\mathbf{0}, \mathbf{K}_{uu}) \quad \text{marginal over } \mathbf{u}$$

We can marginalize out $\mathbf{u}$ and recover back $p(\mathbf{f})$:

$$\int p(\mathbf{f}|\mathbf{u})p(\mathbf{u})d\mathbf{u} = p(\mathbf{f}) \quad \textit{consistency}$$

Inducing variables

We have

$$p(\mathbf{f}, \mathbf{u}) = \mathcal{N}\left(\left[\begin{array}{c} \mathbf{f} \\ \mathbf{u} \end{array}\right] \middle| \mathbf{0}, \left[\begin{array}{cc} \mathbf{K}_{ff} & \mathbf{K}_{fu} \\ \mathbf{K}_{uf} & \mathbf{K}_{uu} \end{array}\right]\right) = p(\mathbf{f}|\mathbf{u})p(\mathbf{u})$$

where

$$\begin{aligned} p(\mathbf{f}|\mathbf{u}) &= \mathcal{N}(\mathbf{f}|\mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{u}, \mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}) && \text{conditional GP prior} \\ p(\mathbf{u}) &= \mathcal{N}(\mathbf{u}|\mathbf{0}, \mathbf{K}_{uu}) && \text{marginal over } \mathbf{u} \end{aligned}$$

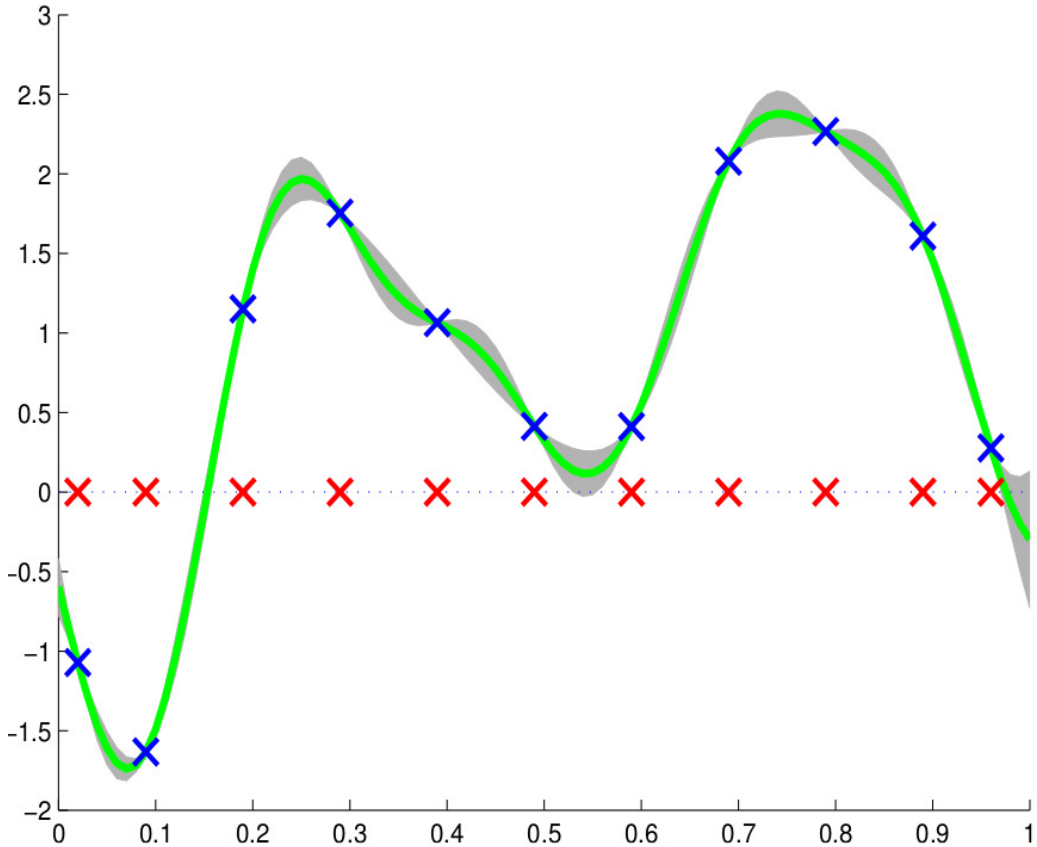
We can marginalize out $\mathbf{u}$ and recover back $p(\mathbf{f})$:

$$\int p(\mathbf{f}|\mathbf{u})p(\mathbf{u})d\mathbf{u} = p(\mathbf{f}) \qquad \textit{consistency}$$

An efficient construction of $\mathbf{u}$ and selection of values for Z should be such that $\mathbf{u}$ correlates strongly with $\mathbf{f}$

- i.e. $p(\mathbf{f}|\mathbf{u})$ is sharply picked

Inducing variables



A realized value for $\mathbf{f}$

Realized values for the inducing variables $\mathbf{u} = (u_1, \ldots, u_m)$, $u_i = f(\mathbf{z}_i)$

The augmentation parameters are the **inducing inputs** $Z = (\mathbf{z}_1, \ldots, \mathbf{z}_m)$

Conditional prior $p(\mathbf{f}|\mathbf{u})$

Inducing variables

The whole purpose of adding **u** is to help us obtain an approximation to our Bayesian non-parametric model (**without changing its non-parametric nature... as will be discussed shortly**) that will scale better computationally

Inducing variables

The whole purpose of adding $\mathbf{u}$ is to help us obtain an approximation to our Bayesian non-parametric model (**without changing its non-parametric nature... as will be discussed shortly**) that will scale better computationally

The big question now is how do we "turn around" $\mathbf{u}$ in order to make it the basis of our approximation? Further, how do we learn the augmentation parameters Z?

Variational learning of inducing variables

Augmented joint

$$p(\mathbf{y}, \mathbf{f}, \mathbf{u}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{u})p(\mathbf{u})$$

Variational learning of inducing variables

Augmented joint

$$p(\mathbf{y}, \mathbf{f}, \mathbf{u}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{u})p(\mathbf{u})$$

Augmented exact posterior

$$p(\mathbf{f}, \mathbf{u}|\mathbf{y}) = p(\mathbf{f}|\mathbf{u}, \mathbf{y})p(\mathbf{u}|\mathbf{y})$$

Variational learning of inducing variables

Augmented joint

$$p(\mathbf{y}, \mathbf{f}, \mathbf{u}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{u})p(\mathbf{u})$$

Augmented exact posterior

$$p(\mathbf{f}, \mathbf{u}|\mathbf{y}) = p(\mathbf{f}|\mathbf{u}, \mathbf{y})p(\mathbf{u}|\mathbf{y})$$

Marginal likelihood is invariant to the augmentation parameters Z

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

and the marginal posterior $p(\mathbf{f}|\mathbf{y})$ is also invariant to Z

- **I.e. Z is not model parameter**
- **$\Rightarrow$ we can turn it into variational parameter by lower bounding**

Variational learning of inducing variables

Joint

$$p(\mathbf{y}, \mathbf{f}, \mathbf{u}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{u})p(\mathbf{u})$$

Variational learning of inducing variables

Joint

$$p(\mathbf{y}, \mathbf{f}, \mathbf{u}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{u})p(\mathbf{u})$$

Exact posterior distribution

$$p(\mathbf{f}, \mathbf{u}|\mathbf{y}) = p(\mathbf{f}|\mathbf{u}, \mathbf{y})p(\mathbf{u}|\mathbf{y})$$

Variational learning of inducing variables

Joint

$$p(\mathbf{y}, \mathbf{f}, \mathbf{u}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{u})p(\mathbf{u})$$

Exact posterior distribution

$$p(\mathbf{f}, \mathbf{u}|\mathbf{y}) = p(\mathbf{f}|\mathbf{u}, \mathbf{y})p(\mathbf{u}|\mathbf{y})$$

Variational distribution

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

Variational learning of inducing variables

Joint

$$p(\mathbf{y}, \mathbf{f}, \mathbf{u}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{u})p(\mathbf{u})$$

Exact posterior distribution

$$p(\mathbf{f}, \mathbf{u}|\mathbf{y}) = p(\mathbf{f}|\mathbf{u}, \mathbf{y})p(\mathbf{u}|\mathbf{y})$$

Variational distribution

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

This choice encourages $\mathbf{u}$ to become approximate sufficient statistic

if $p(\mathbf{f}|\mathbf{u}) \approx p(\mathbf{f}|\mathbf{u}, \mathbf{y})$, then $\mathbf{u}$ summarizes well the data

Variational learning of inducing variables

Minimize KL $[q(\mathbf{f}, \mathbf{u})||p(\mathbf{f}, \mathbf{u}|\mathbf{y})]$ or equivalently maximize the bound on the log marginal likelihood

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

Variational learning of inducing variables

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

Variational learning of inducing variables

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

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

Variational learning of inducing variables

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

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

$$\log p(\mathbf{y}) \geq \int q(\mathbf{f},\mathbf{u}) \log \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{u})p(\mathbf{u})}{q(\mathbf{f},\mathbf{u})} d\mathbf{f}d\mathbf{u}$$

Variational learning of inducing variables

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

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

$$\log p(\mathbf{y}) \geq \int q(\mathbf{f},\mathbf{u}) \log \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{u})p(\mathbf{u})}{q(\mathbf{f},\mathbf{u})} d\mathbf{f}d\mathbf{u}$$

Substitute $q(\mathbf{f},\mathbf{u}) = p(\mathbf{f}|\mathbf{u})q(\mathbf{u})$:

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

Variational learning of inducing variables

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

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

$$\log p(\mathbf{y}) \geq \int q(\mathbf{f},\mathbf{u}) \log \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{u})p(\mathbf{u})}{q(\mathbf{f},\mathbf{u})} d\mathbf{f}d\mathbf{u}$$

Substitute $q(\mathbf{f},\mathbf{u}) = p(\mathbf{f}|\mathbf{u})q(\mathbf{u})$:

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

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

Variational learning of inducing variables

$$\log p(\mathbf{y}) \geq \int q(\mathbf{u}) \left[\int p(\mathbf{f}|\mathbf{u}) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f} + \log \frac{p(\mathbf{u})}{q(\mathbf{u})} \right] d\mathbf{u}$$

Variational learning of inducing variables

$$\log p(\mathbf{y}) \geq \int q(\mathbf{u}) \left[\int p(\mathbf{f}|\mathbf{u}) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f} + \log \frac{p(\mathbf{u})}{q(\mathbf{u})} \right] d\mathbf{u}$$

$$\log p(\mathbf{y}) \geq \int q(\mathbf{u}) \left[\log e^{\int p(\mathbf{f}|\mathbf{u}) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f}} + \log \frac{p(\mathbf{u})}{q(\mathbf{u})} \right] d\mathbf{u}$$

Variational learning of inducing variables

$$\log p(\mathbf{y}) \geq \int q(\mathbf{u}) \left[\int p(\mathbf{f}|\mathbf{u}) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f} + \log \frac{p(\mathbf{u})}{q(\mathbf{u})} \right] d\mathbf{u}$$

$$\log p(\mathbf{y}) \geq \int q(\mathbf{u}) \left[\log e^{\int p(\mathbf{f}|\mathbf{u}) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f}} + \log \frac{p(\mathbf{u})}{q(\mathbf{u})} \right] d\mathbf{u}$$

$$\log p(\mathbf{y}) \geq \int q(\mathbf{u}) \log \frac{e^{\int p(\mathbf{f}|\mathbf{u}) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f}} p(\mathbf{u})}{q(\mathbf{u})} d\mathbf{u}$$

Variational learning of inducing variables

$$\log p(\mathbf{y}) \geq \int q(\mathbf{u}) \left[\int p(\mathbf{f}|\mathbf{u}) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f} + \log \frac{p(\mathbf{u})}{q(\mathbf{u})} \right] d\mathbf{u}$$

$$\log p(\mathbf{y}) \geq \int q(\mathbf{u}) \left[\log e^{\int p(\mathbf{f}|\mathbf{u}) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f}} + \log \frac{p(\mathbf{u})}{q(\mathbf{u})} \right] d\mathbf{u}$$

$$\log p(\mathbf{y}) \geq \int q(\mathbf{u}) \log \frac{e^{\int p(\mathbf{f}|\mathbf{u}) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f}} p(\mathbf{u})}{q(\mathbf{u})} d\mathbf{u}$$

Maximize over $q(\mathbf{u})$:

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

Variational learning of inducing variables

Theorem 1 (bound).

For arbitrary GP model:

$$p(\mathbf{y}) \geq \int G(\mathbf{y}, \mathbf{u}) p(\mathbf{u}) d\mathbf{u}, \quad G(\mathbf{y}, \mathbf{u}) = e^{\int p(\mathbf{f}|\mathbf{u}) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f}}$$

For GP regression:

$$\mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_{ff} + \sigma^2 I) \geq \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf} + \sigma^2 I) e^{-\frac{1}{2\sigma^2}\mathrm{tr}\left(\mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}\right)}$$

Variational learning of inducing variables

Theorem 1 (bound).

For arbitrary GP model:

$$p(\mathbf{y}) \geq \int G(\mathbf{y}, \mathbf{u}) p(\mathbf{u}) d\mathbf{u}, \quad G(\mathbf{y}, \mathbf{u}) = e^{\int p(\mathbf{f}|\mathbf{u}) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f}}$$

For GP regression:

$$\mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_{ff} + \sigma^2 I) \geq \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf} + \sigma^2 I) e^{-\frac{1}{2\sigma^2}\mathrm{tr}\left(\mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}\right)}$$

Theorem 2 (monotonicity property). If we have inducing variables $\mathbf{u}$ and add an extra u_i the bound can only increase

$$\int G(\mathbf{y}, \mathbf{u}, u_i) p(\mathbf{u}, u_i) d\mathbf{u} \geq \int G(\mathbf{y}, \mathbf{u}) p(\mathbf{u}) d\mathbf{u}$$

Variational learning of inducing variables

Theorem 1 (bound).

For arbitrary GP model:

$$p(\mathbf{y}) \geq \int G(\mathbf{y}, \mathbf{u}) p(\mathbf{u}) d\mathbf{u}, \quad G(\mathbf{y}, \mathbf{u}) = e^{\int p(\mathbf{f}|\mathbf{u}) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f}}$$

For GP regression:

$$\mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_{ff} + \sigma^2 I) \geq \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf} + \sigma^2 I) e^{-\frac{1}{2\sigma^2}\mathrm{tr}\left(\mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}\right)}$$

Theorem 2 (monotonicity property). If we have inducing variables $\mathbf{u}$ and add an extra u_i the bound can only increase

$$\int G(\mathbf{y}, \mathbf{u}, u_i) p(\mathbf{u}, u_i) d\mathbf{u} \geq \int G(\mathbf{y}, \mathbf{u}) p(\mathbf{u}) d\mathbf{u}$$

Computation of the bound and the approximate GP prediction scale as $O(nm^2)$ where m is the number of inducing variables

Variational learning of inducing variables

For GP regression the bound has an interesting form:

$$\mathcal{F}(Z, \theta) = \log \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf} + \sigma^2 I) - \frac{1}{2\sigma^2}\mathrm{tr}\left(\mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}\right)$$

Variational learning of inducing variables

For GP regression the bound has an interesting form:

$$\mathcal{F}(Z, \boldsymbol{\theta}) = \log \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf} + \sigma^2 I) - \frac{1}{2\sigma^2}\mathrm{tr}\left(\mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}\right)$$

The first term is an approximation to the log marginal likelihood (*proposed by M. Seeger for learning kernel hyperparameters* $\boldsymbol{\theta}$)

Variational learning of inducing variables

For GP regression the bound has an interesting form:

$$\mathcal{F}(Z,\theta) = \log \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf} + \sigma^2 I) - \frac{1}{2\sigma^2}\text{tr}\left(\mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}\right)$$

The first term is an approximation to the log marginal likelihood (*proposed by M. Seeger for learning kernel hyperparameters* θ)

The second term is an extra regularization term which depends on the total variance of the conditional prior $p(\mathbf{f}|\mathbf{u})$:

$$\text{tr}\left(\mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}\right)$$

Variational learning of inducing variables

For GP regression the bound has an interesting form:

$$\mathcal{F}(Z,\boldsymbol{\theta}) = \log \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf} + \sigma^2 I) - \frac{1}{2\sigma^2}\mathrm{tr}\left(\mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}\right)$$

The first term is an approximation to the log marginal likelihood (*proposed by M. Seeger for learning kernel hyperparameters* $\boldsymbol{\theta}$)

The second term is an extra regularization term which depends on the total variance of the conditional prior $p(\mathbf{f}|\mathbf{u})$:

$$\mathrm{tr}\left(\mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}\right)$$

We maximize $\mathcal{F}(Z,\boldsymbol{\theta})$ over Z and kernel hyperparameters $\boldsymbol{\theta}$:

- Z is a variational parameter

Variational learning of inducing variables

The approximate posterior/predictive Gaussian process:

$$
\begin{aligned}
q(\mathbf{f}_*) &= \int p(\mathbf{f}_*|\mathbf{f},\mathbf{u}) q(\mathbf{f},\mathbf{u}) d\mathbf{f} d\mathbf{u} \\
&= \int p(\mathbf{f}_*|\mathbf{f},\mathbf{u}) p(\mathbf{f}|\mathbf{u}) q(\mathbf{u}) d\mathbf{f} d\mathbf{u} \\
&= \int p(\mathbf{f}_*|\mathbf{u}) q(\mathbf{u}) d\mathbf{u}
\end{aligned}
$$

where we used the consistency $\int p(\mathbf{f}_*|\mathbf{f},\mathbf{u}) p(\mathbf{f}|\mathbf{u}) d\mathbf{f} = p(\mathbf{f}|\mathbf{u})$

Variational learning of inducing variables

The approximate posterior/predictive Gaussian process:

$$
\begin{aligned}
q(\mathbf{f}_*) &= \int p(\mathbf{f}_*|\mathbf{f},\mathbf{u}) q(\mathbf{f},\mathbf{u}) d\mathbf{f} d\mathbf{u} \\
&= \int p(\mathbf{f}_*|\mathbf{f},\mathbf{u}) p(\mathbf{f}|\mathbf{u}) q(\mathbf{u}) d\mathbf{f} d\mathbf{u} \\
&= \int p(\mathbf{f}_*|\mathbf{u}) q(\mathbf{u}) d\mathbf{u}
\end{aligned}
$$

where we used the consistency $\int p(\mathbf{f}_*|\mathbf{f},\mathbf{u}) p(\mathbf{f}|\mathbf{u}) d\mathbf{f} = p(\mathbf{f}|\mathbf{u})$

There is a very important thing to be said here:

- This is not a discretized, truncated or low rank approximation (it is full rank **and NOT low rank as many people believe**)
- This is because the conditional GP $p(\mathbf{f}_*|\mathbf{u})$ is an infinite object

Variational learning of inducing variables

The approximate posterior/predictive Gaussian process:

$$
\begin{aligned}
q(\mathbf{f}_*) &= \int p(\mathbf{f}_*|\mathbf{f},\mathbf{u}) q(\mathbf{f},\mathbf{u}) d\mathbf{f} d\mathbf{u} \\
&= \int p(\mathbf{f}_*|\mathbf{f},\mathbf{u}) p(\mathbf{f}|\mathbf{u}) q(\mathbf{u}) d\mathbf{f} d\mathbf{u} \\
&= \int p(\mathbf{f}_*|\mathbf{u}) q(\mathbf{u}) d\mathbf{u}
\end{aligned}
$$

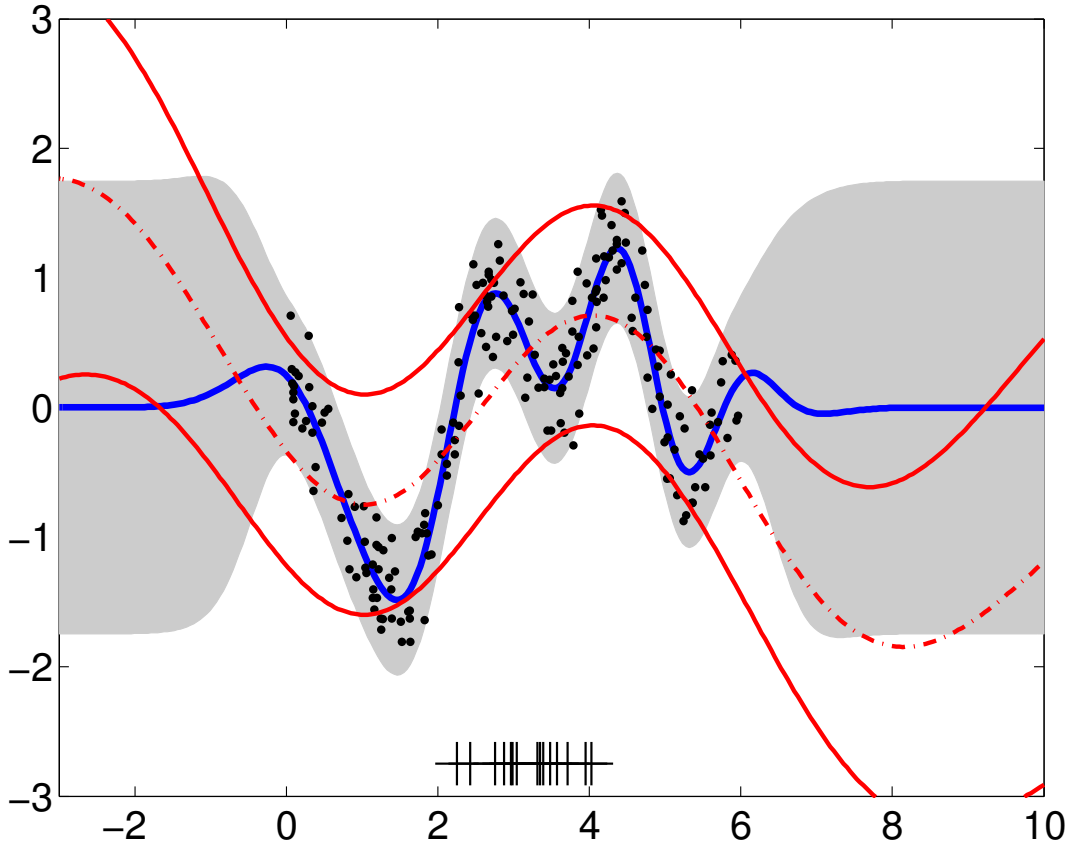
where we used the consistency $\int p(\mathbf{f}_*|\mathbf{f},\mathbf{u}) p(\mathbf{f}|\mathbf{u}) d\mathbf{f} = p(\mathbf{f}|\mathbf{u})$

There is a very important thing to be said here:

- This is not a discretized, truncated or low rank approximation (it is full rank **and NOT low rank as many people believe**)
- This is because the conditional GP $p(\mathbf{f}_*|\mathbf{u})$ is an infinite object

The approximation can be thought of been **restricted not to explore freely the information in the training data**. But it maintains fully the non-parametric nature of the model

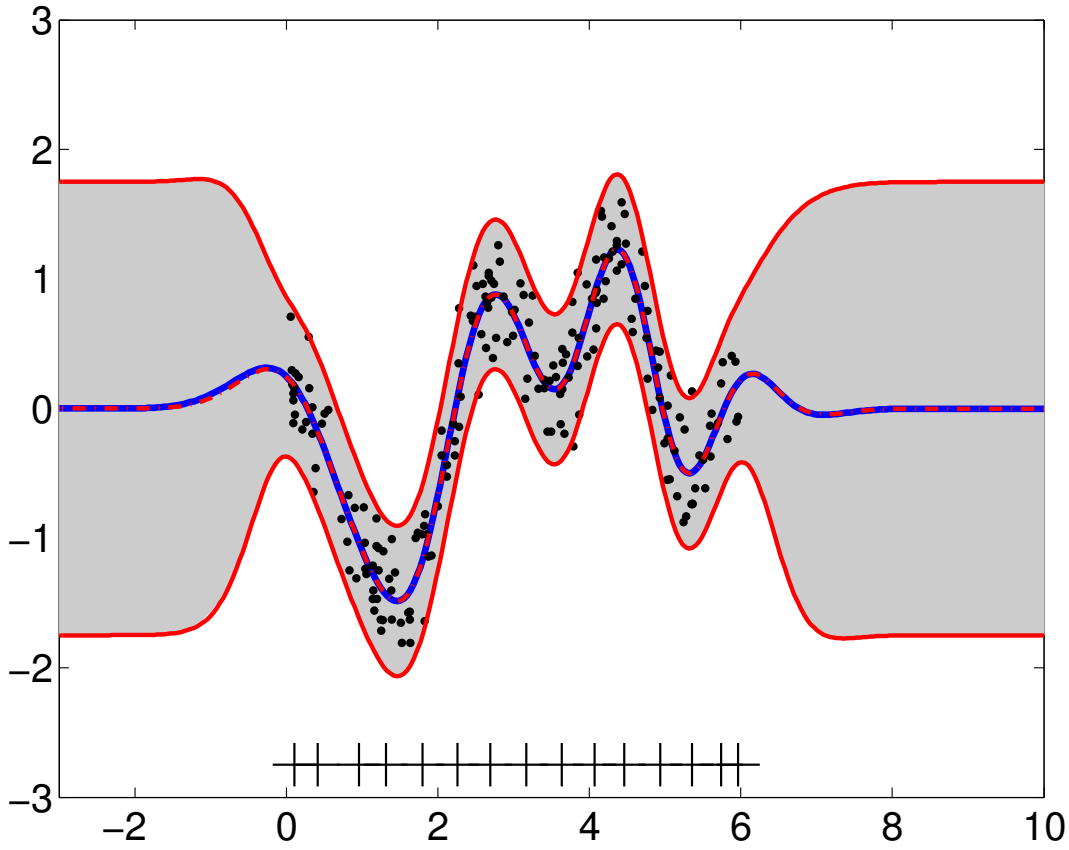
Variational learning of inducing variables



Full GP that scales as $O(n^3) = O(200^3)$
Variational approximation that scales as $O(nm^2) = O(200 \times 15^2)$ **at initialization**

- The crosses $(+)$ are the initial values of the inducing inputs Z

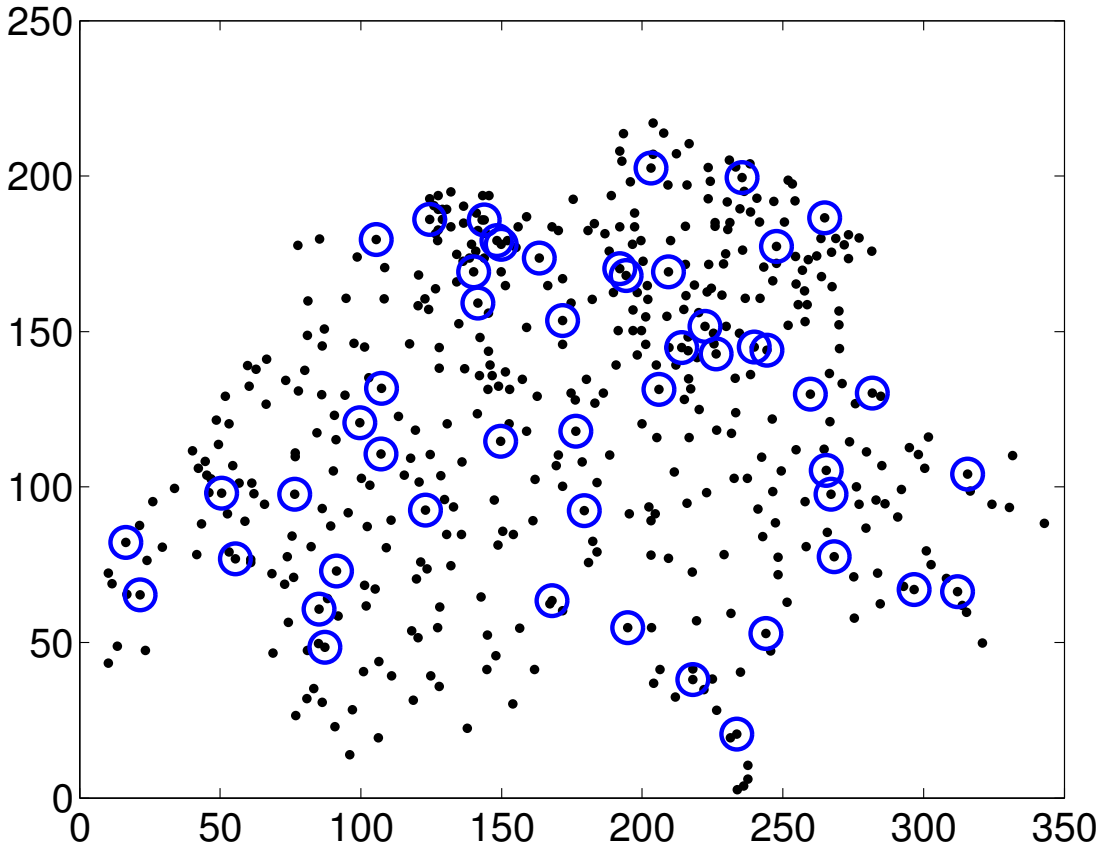
Variational learning of inducing variables



Full GP that scales as $O(n^3) = O(200^3)$

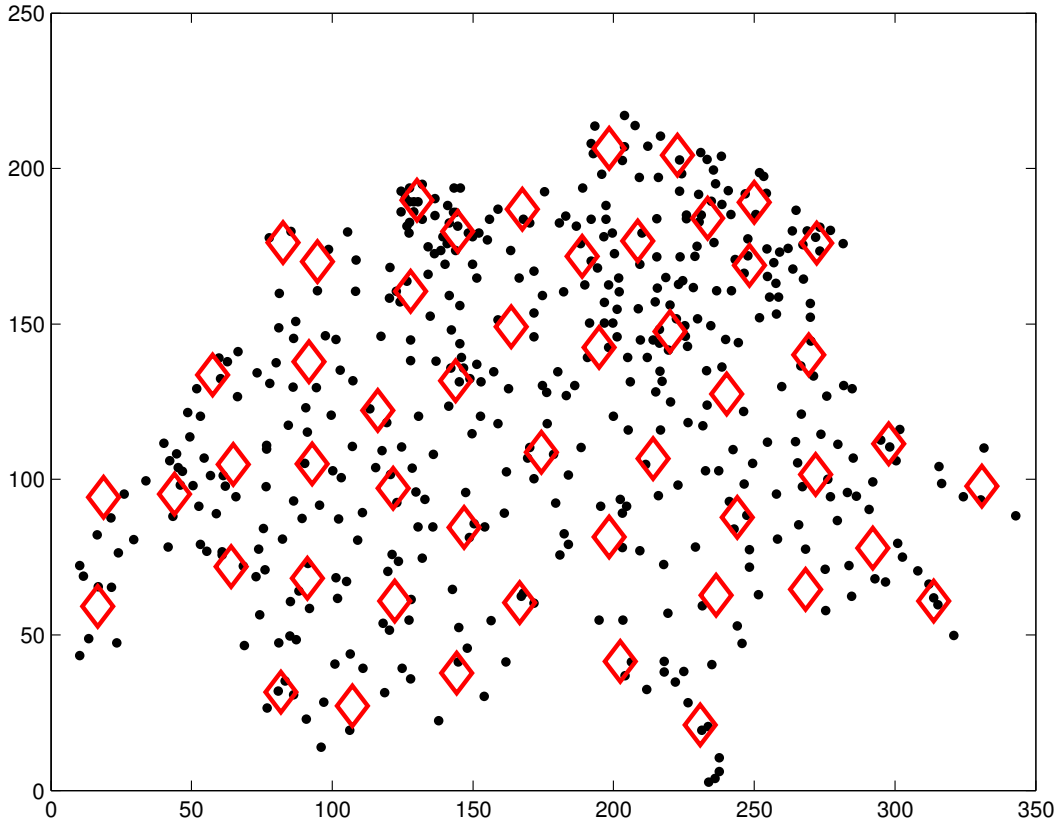
Variational approximation that scales as $O(nm^2) = O(200 \times 15^2)$ **after having maximized the bound**

Variational learning of inducing variables



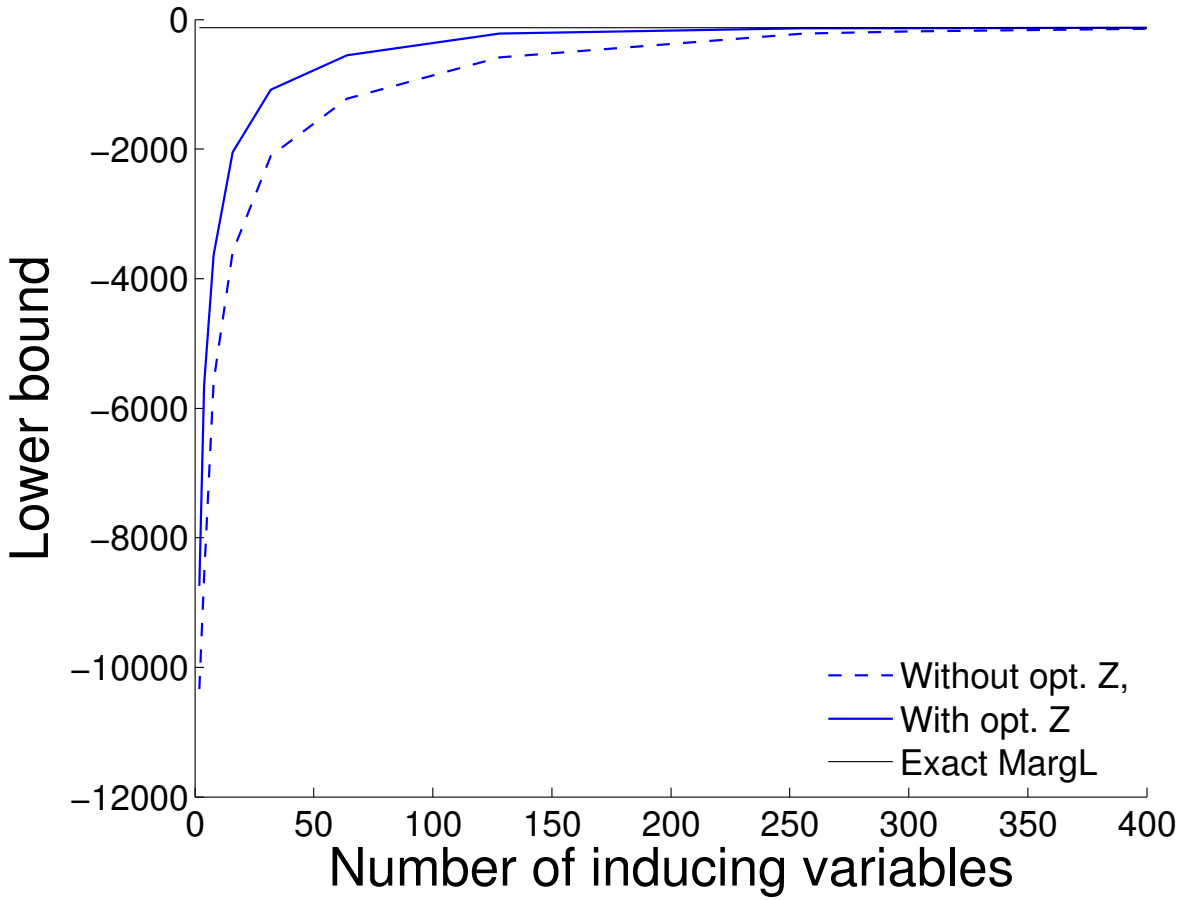
Initial locations of the inducing inputs Z

Variational learning of inducing variables



Locations of the inducing inputs after having maximized the bound

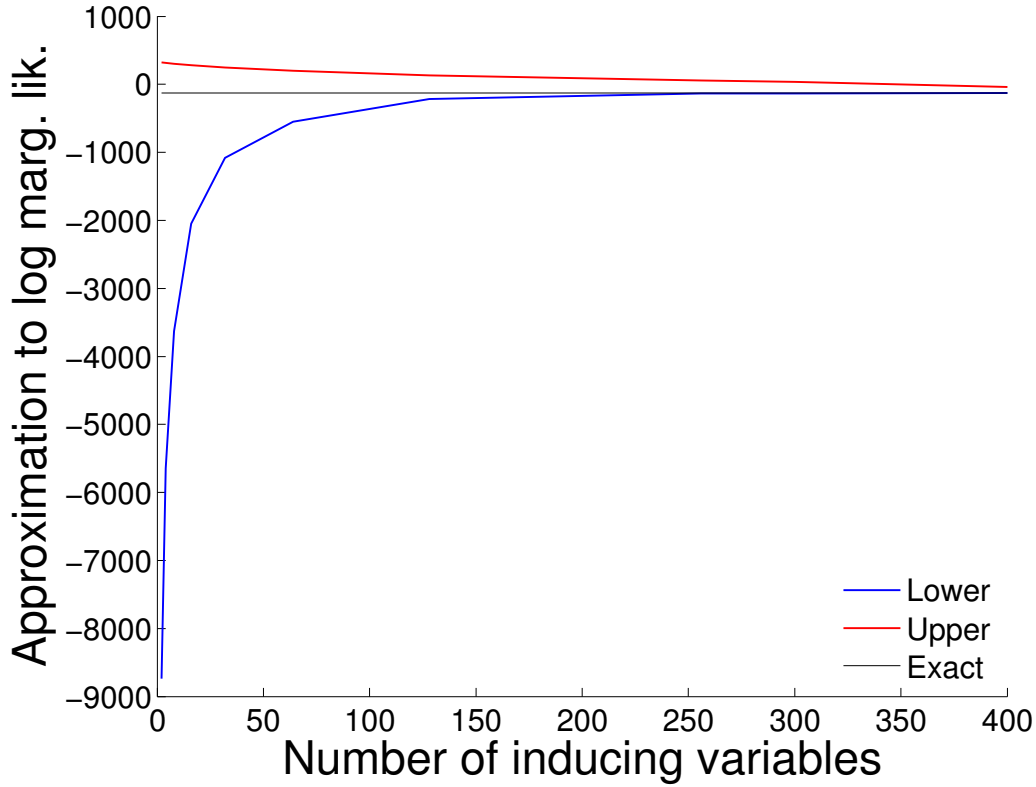
Variational learning of inducing variables



Maximization wrt the inducing inputs Z improves the approximation

(The example is based on the Boston housing data and Z is initialized to a random subset of training inputs)

Variational learning of inducing variables



If the bound flattens as we add more inducing variables we typically have reached full GP

To further assess the approximation we can consult an upper bound

$$p(\mathbf{y}) \leq \frac{1}{(2\pi)^{\frac{n}{2}} \left|\mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf} + \sigma^2 I\right|^{\frac{1}{2}}} e^{-\frac{1}{2}\mathbf{y}^T\left(\mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf} + cI + \sigma^2 I\right)^{-1}\mathbf{y}}$$

where $c = \text{tr}\left(\mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}\right)$ (the proof is given in the appendix)

Variational learning of inducing variables

When the bound becomes tight?

$$\mathcal{F}(Z,\theta) = \log \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf} + \sigma^2 I) - \frac{1}{2\sigma^2}\mathrm{tr}\left(\mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}\right)$$

Variational learning of inducing variables

When the bound becomes tight?

$$\mathcal{F}(Z,\theta) = \log \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf} + \sigma^2 I) - \frac{1}{2\sigma^2}\text{tr}\left(\mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}\right)$$

When the trace term is zero the bound becomes tight, i.e.

$$\text{If } \text{tr}\left(\mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}\right) = 0 \Rightarrow \mathbf{K}_{ff} = \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}, \ \mathcal{F} = \log p(\mathbf{y})$$

Variational learning of inducing variables

When the bound becomes tight?

$$\mathcal{F}(Z,\theta) = \log \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf} + \sigma^2 I) - \frac{1}{2\sigma^2}\text{tr}\left(\mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}\right)$$

When the trace term is zero the bound becomes tight, i.e.

$$\text{If } \text{tr}\left(\mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}\right) = 0 \Rightarrow \mathbf{K}_{ff} = \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}, \ \mathcal{F} = \log p(\mathbf{y})$$

This can be always achieved if we set $Z = X$ (so that $m = n$)

Variational learning of inducing variables

When the bound becomes tight?

$$\mathcal{F}(Z,\theta) = \log \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf} + \sigma^2 I) - \frac{1}{2\sigma^2}\text{tr}\left(\mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}\right)$$

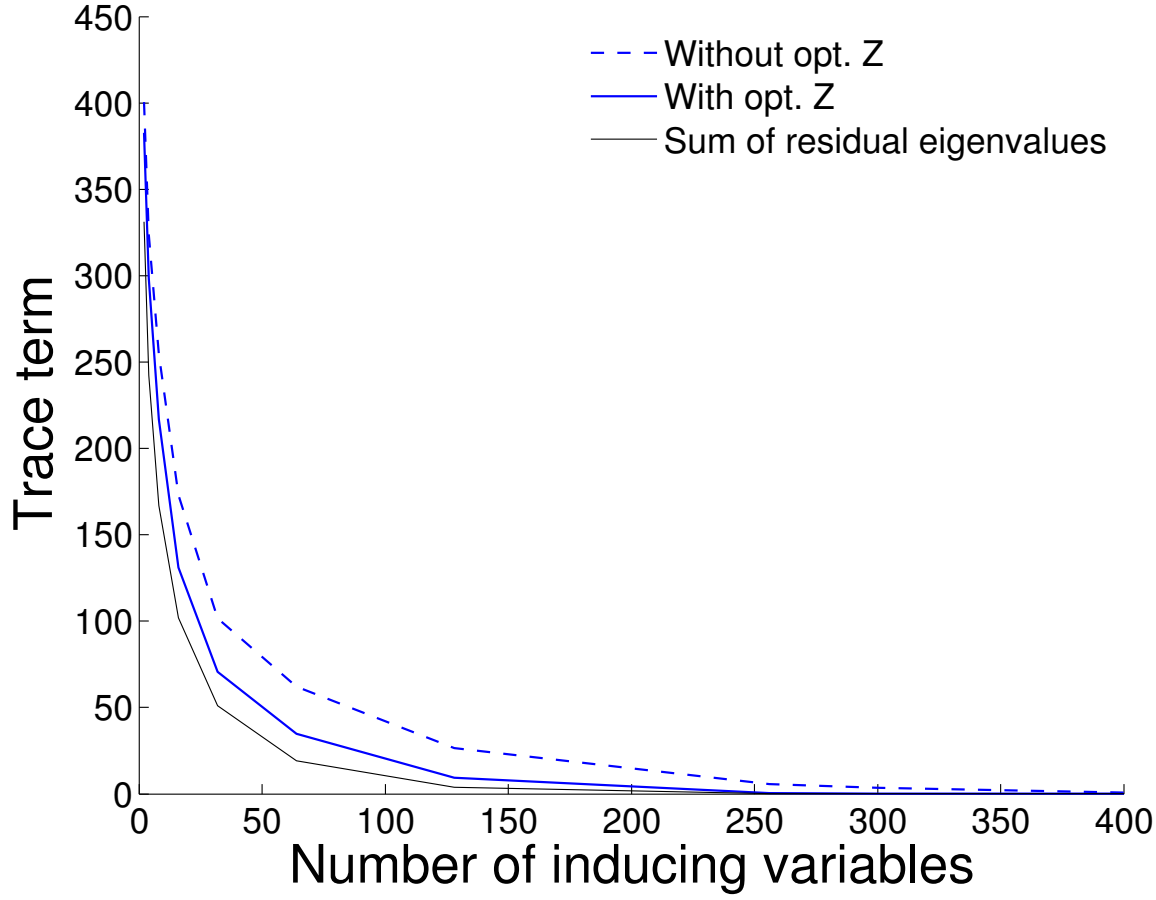
When the trace term is zero the bound becomes tight, i.e.

$$\text{If } \text{tr}\left(\mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}\right) = 0 \Rightarrow \mathbf{K}_{ff} = \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}, \ \mathcal{F} = \log p(\mathbf{y})$$

This can be always achieved if we set $Z = X$ (so that $m = n$)

Question: what is the best we can do if we use $m < n$ inducing variables?

Variational learning of inducing variables



It holds that $\text{tr}\left(\mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}\right) \geq \sum_{i=m+1}^{n} \lambda_i$

where $\lambda_1 \geq \lambda_2 \geq .... \geq \lambda_n$ are the eigenvalues of $\mathbf{K}_{ff}$

Some history

Early work (that set the foundation of these approximations) is: Csato and Opper (2002); Seeger (2003); Seeger, Williams and Lawrence (2003)

Some history

Early work (that set the foundation of these approximations) is: Csato and Opper (2002); Seeger (2003); Seeger, Williams and Lawrence (2003)

Snelson and Ghahramani (2006) (proposed pseudo-inputs and FITC) and the related unification of methods by Quinonero-Candela and Rasmussen (2005)

Some history

Early work (that set the foundation of these approximations) is: Csato and Opper (2002); Seeger (2003); Seeger, Williams and Lawrence (2003)

Snelson and Ghahramani (2006) (proposed pseudo-inputs and FITC) and the related unification of methods by Quinonero-Candela and Rasmussen (2005)

Titsias (2009) (derived the bound in regression and treated inducing variables as variational parameters). A continuation of this in Titsias and Lawrence (2010) extended the framework to variationally integrate out inputs in GP functions

Some history

Early work (that set the foundation of these approximations) is: Csato and Opper (2002); Seeger (2003); Seeger, Williams and Lawrence (2003)

Snelson and Ghahramani (2006) (proposed pseudo-inputs and FITC) and the related unification of methods by Quinonero-Candela and Rasmussen (2005)

Titsias (2009) (derived the bound in regression and treated inducing variables as variational parameters). A continuation of this in Titsias and Lawrence (2010) extended the framework to variationally integrate out inputs in GP functions

Hensman, Fusi and Lawrence (2013) (combined the framework with stochastic data sub-sampling variational inference)

Determinantal point processes

Lets now discuss how we can extend this approximation to determinantal point processes

Determinantal point processes

Given a discrete set of items $\mathcal{Y} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ a DPP defines a distribution over all 2^n possible subsets via a $n \times n$ kernel matrix $L_{\mathcal{Y}}$ such that $[L_{\mathcal{Y}}]_{ij} = L(\mathbf{x}_i, \mathbf{x}_j)$:

$$\Pr(\mathbf{Y} = Y) = \frac{\det(L_Y)}{\det(L_{\mathcal{Y}} + I)}$$

where L_Y is the kernel sub-matrix indexed by the elements of $Y \subseteq \mathcal{Y}$

Determinantal point processes

Given a discrete set of items $\mathcal{Y} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ a DPP defines a distribution over all 2^n possible subsets via a $n \times n$ kernel matrix $L_{\mathcal{Y}}$ such that $[L_{\mathcal{Y}}]_{ij} = L(\mathbf{x}_i, \mathbf{x}_j)$:

$$\Pr(\mathbf{Y} = Y) = \frac{\det(L_Y)}{\det(L_{\mathcal{Y}} + I)}$$

where L_Y is the kernel sub-matrix indexed by the elements of $Y \subseteq \mathcal{Y}$

A DPP is a point process that favors repulsion, i.e. items with very similar descriptors (**x**s) are unlikely to appear in the same realization

- for full details see e.g. Kulesza and Taskar, Foundations and Trends in Machine Learning (2012)

Determinantal point processes

Given a set of observed subsets of items $(Y_1, \ldots, Y_T)$ we can fit the model by ML:

$$\mathcal{L}(\boldsymbol{\theta}) = \log \prod_{t=1}^{T} \frac{\det(L_{Y_t})}{\det(L_{\mathcal{Y}} + I)}$$

Determinantal point processes

Given a set of observed subsets of items $(Y_1, \ldots, Y_T)$ we can fit the model by ML:

$$\mathcal{L}(\theta) = \log \prod_{t=1}^{T} \frac{\det(L_{Y_t})}{\det(L_{\mathcal{Y}} + I)}$$

When the set $\mathcal{Y}$ is very large we cannot store $L_{\mathcal{Y}}$ and compute $\det(L_{\mathcal{Y}} + I)$

- $\Rightarrow$ maximization of the exact likelihood becomes intractable

Determinantal point processes

Given a set of observed subsets of items $(Y_1, \ldots, Y_T)$ we can fit the model by ML:

$$\mathcal{L}(\boldsymbol{\theta}) = \log \prod_{t=1}^{T} \frac{\det(L_{Y_t})}{\det(L_{\mathcal{Y}} + I)}$$

When the set $\mathcal{Y}$ is very large we cannot store $L_{\mathcal{Y}}$ and compute $\det(L_{\mathcal{Y}} + I)$

- $\Rightarrow$ maximization of the exact likelihood becomes intractable

To apply variational inference we need to compute an upper bound on $\det(L_{\mathcal{Y}} + I)$ or equivalently a lower bound on

$$\frac{1}{\det(L_{\mathcal{Y}} + I)}$$

Determinantal point processes

We assume an inducing subset $Z \subseteq \mathcal{Y}$. From the bound in GP regression we know that:

$$\mathcal{N}(\mathbf{y}|\mathbf{0}, L_{\mathcal{Y}}+\sigma^2 I) \geq \mathcal{N}(\mathbf{y}|\mathbf{0}, L_{\mathcal{Y}Z}L_Z^{-1}L_{Z\mathcal{Y}}+\sigma^2 I)e^{-\frac{1}{2\sigma^2}\mathrm{tr}\left(L_{\mathcal{Y}}-L_{\mathcal{Y}Z}L_Z^{-1}L_{Z\mathcal{Y}}\right)}$$

Determinantal point processes

We assume an inducing subset $Z \subseteq \mathcal{Y}$. From the bound in GP regression we know that:

$$\mathcal{N}(\mathbf{y}|\mathbf{0}, L_{\mathcal{Y}}+\sigma^2 I) \geq \mathcal{N}(\mathbf{y}|\mathbf{0}, L_{\mathcal{Y}Z}L_Z^{-1}L_{Z\mathcal{Y}}+\sigma^2 I)e^{-\frac{1}{2\sigma^2}\mathrm{tr}\left(L_{\mathcal{Y}}-L_{\mathcal{Y}Z}L_Z^{-1}L_{Z\mathcal{Y}}\right)}$$

By setting $\mathbf{y} = \mathbf{0}$ and $\sigma^2 = 1$:

$$\frac{1}{\det(L_{\mathcal{Y}} + I)^{\frac{1}{2}}} \geq \frac{1}{\det(L_{\mathcal{Y}Z}L_Z^{-1}L_{Z\mathcal{Y}} + I)^{\frac{1}{2}}} e^{-\frac{1}{2}\mathrm{tr}\left(L_{\mathcal{Y}}-L_{\mathcal{Y}Z}L_Z^{-1}L_{Z\mathcal{Y}}\right)}$$

Determinantal point processes

We assume an inducing subset $Z \subseteq \mathcal{Y}$. From the bound in GP regression we know that:

$$\mathcal{N}(\mathbf{y}|\mathbf{0}, L_{\mathcal{Y}}+\sigma^2 I) \geq \mathcal{N}(\mathbf{y}|\mathbf{0}, L_{\mathcal{Y}Z}L_Z^{-1}L_{Z\mathcal{Y}}+\sigma^2 I)e^{-\frac{1}{2\sigma^2}\text{tr}\left(L_{\mathcal{Y}}-L_{\mathcal{Y}Z}L_Z^{-1}L_{Z\mathcal{Y}}\right)}$$

By setting $\mathbf{y} = \mathbf{0}$ and $\sigma^2 = 1$:

$$\frac{1}{\det(L_{\mathcal{Y}} + I)^{\frac{1}{2}}} \geq \frac{1}{\det(L_{\mathcal{Y}Z}L_Z^{-1}L_{Z\mathcal{Y}} + I)^{\frac{1}{2}}} e^{-\frac{1}{2}\text{tr}\left(L_{\mathcal{Y}}-L_{\mathcal{Y}Z}L_Z^{-1}L_{Z\mathcal{Y}}\right)}$$

By taking the square:

$$\frac{1}{\det(L_{\mathcal{Y}} + I)} \geq \frac{1}{\det(L_{\mathcal{Y}Z}L_Z^{-1}L_{Z\mathcal{Y}} + I)} e^{-\text{tr}\left(L_{\mathcal{Y}}-L_{\mathcal{Y}Z}L_Z^{-1}L_{Z\mathcal{Y}}\right)}$$

Determinantal point processes

$$\frac{1}{\det(L_{\mathcal{Y}} + I)} \geq \frac{1}{\det(L_{\mathcal{Y}Z} L_Z^{-1} L_{Z\mathcal{Y}} + I)} e^{-\mathrm{tr}\left(L_{\mathcal{Y}} - L_{\mathcal{Y}Z} L_Z^{-1} L_{Z\mathcal{Y}}\right)}$$

Determinantal point processes

$$\frac{1}{\det(L_{\mathcal{Y}} + I)} \geq \frac{1}{\det(L_{\mathcal{Y}Z} L_Z^{-1} L_{Z\mathcal{Y}} + I)} e^{-\mathrm{tr}\left(L_{\mathcal{Y}} - L_{\mathcal{Y}Z} L_Z^{-1} L_{Z\mathcal{Y}}\right)}$$

By applying the matrix determinant lemma and rearranging

$$\frac{1}{\det(L_{\mathcal{Y}} + I)} \geq \frac{\det(L_Z)}{\det(L_Z + L_{Z\mathcal{Y}} L_{\mathcal{Y}Z})} e^{-\mathrm{tr}(L_{\mathcal{Y}}) + \mathrm{tr}\left(L_Z^{-1} L_{Z\mathcal{Y}} L_{\mathcal{Y}Z}\right)}$$

which is computed in $O(nm^2)$ where m is the size of Z

Determinantal point processes

$$\frac{1}{\det(L_{\mathcal{Y}} + I)} \geq \frac{1}{\det(L_{\mathcal{Y}Z} L_Z^{-1} L_{Z\mathcal{Y}} + I)} e^{-\mathrm{tr}\left(L_{\mathcal{Y}} - L_{\mathcal{Y}Z} L_Z^{-1} L_{Z\mathcal{Y}}\right)}$$

By applying the matrix determinant lemma and rearranging

$$\frac{1}{\det(L_{\mathcal{Y}} + I)} \geq \frac{\det(L_Z)}{\det(L_Z + L_{Z\mathcal{Y}} L_{\mathcal{Y}Z})} e^{-\mathrm{tr}(L_{\mathcal{Y}}) + \mathrm{tr}\left(L_Z^{-1} L_{Z\mathcal{Y}} L_{\mathcal{Y}Z}\right)}$$

which is computed in $O(nm^2)$ where m is the size of Z

We can now substitute the bound on the likelihood and maximize the overall lower bound. Z is a variational parameter exactly as in the GP case

Determinantal point processes

If the space of items is continuous, i.e. $\mathcal{Y} = \mathbb{R}^D$, and assuming $\int L(\mathbf{x}, \mathbf{x}) d\mathbf{x} < \infty$ a DPP has density

$$\mathrm{P}(\mathbf{Y} = Y) = \frac{\det(L_Y)}{\prod_{i=1}^{\infty}(\lambda_i + 1)}$$

where λ_is are the eigenvalues of the kernel function. The model now is doubly intractable as typically we don't know the eigenvalues of the kernel

Determinantal point processes

If the space of items is continuous, i.e. $\mathcal{Y} = \mathbb{R}^D$, and assuming $\int L(\mathbf{x}, \mathbf{x}) d\mathbf{x} < \infty$ a DPP has density

$$\mathrm{P}(\mathbf{Y} = Y) = \frac{\det(L_Y)}{\prod_{i=1}^{\infty}(\lambda_i + 1)}$$

where λ_is are the eigenvalues of the kernel function. The model now is doubly intractable as typically we don't know the eigenvalues of the kernel

However, we can compute a lower bound:

$$\frac{1}{\prod_{i=1}^{\infty}(\lambda_i + 1)} \geq \frac{\det(L_Z)}{\det(L_Z + \Psi)} e^{-\int L(\mathbf{x},\mathbf{x}) d\mathbf{x} + \mathrm{tr}(L_Z^{-1}\Psi)}$$

where

$$[\Psi]_{ij} = \int L(\mathbf{z}_i, \mathbf{x}) L(\mathbf{x}, \mathbf{z}_j) d\mathbf{x}$$

Determinantal point processes

If the space of items is continuous, i.e. $\mathcal{Y} = \mathbb{R}^D$, and assuming $\int L(\mathbf{x}, \mathbf{x}) d\mathbf{x} < \infty$ a DPP has density

$$\mathrm{P}(\mathbf{Y} = Y) = \frac{\det(L_Y)}{\prod_{i=1}^{\infty}(\lambda_i + 1)}$$

where λ_is are the eigenvalues of the kernel function. The model now is doubly intractable as typically we don't know the eigenvalues of the kernel

However, we can compute a lower bound:

$$\frac{1}{\prod_{i=1}^{\infty}(\lambda_i + 1)} \geq \frac{\det(L_Z)}{\det(L_Z + \Psi)} e^{-\int L(\mathbf{x},\mathbf{x}) d\mathbf{x} + \mathrm{tr}(L_Z^{-1}\Psi)}$$

where

$$[\Psi]_{ij} = \int L(\mathbf{z}_i, \mathbf{x}) L(\mathbf{x}, \mathbf{z}_j) d\mathbf{x}$$

Z are again variational parameters and can be taken to be pseudo-inputs

Determinantal point processes

Discrete case:

$$\frac{1}{\det(L_{\mathcal{Y}} + I)} \geq \frac{\det(L_Z)}{\det(L_Z + L_{Z\mathcal{Y}} L_{\mathcal{Y}Z})} e^{-\mathrm{tr}(L_{\mathcal{Y}}) + \mathrm{tr}\left(L_Z^{-1} L_{Z\mathcal{Y}} L_{\mathcal{Y}Z}\right)}$$

Continuous case:

$$\frac{1}{\prod_{i=1}^{\infty}(\lambda_i + 1)} \geq \frac{\det(L_Z)}{\det(L_Z + \Psi)} e^{-\int L(\mathbf{x},\mathbf{x})d\mathbf{x} + \mathrm{tr}(L_Z^{-1}\Psi)}$$

Determinantal point processes

Discrete case:

$$\frac{1}{\det(L_{\mathcal{Y}}+I)} \geq \frac{\det(L_Z)}{\det(L_Z+L_{Z\mathcal{Y}}L_{\mathcal{Y}Z})} e^{-\mathrm{tr}(L_{\mathcal{Y}})+\mathrm{tr}\left(L_Z^{-1}L_{Z\mathcal{Y}}L_{\mathcal{Y}Z}\right)}$$

Continuous case:

$$\frac{1}{\prod_{i=1}^{\infty}(\lambda_i+1)} \geq \frac{\det(L_Z)}{\det(L_Z+\Psi)} e^{-\int L(\mathbf{x},\mathbf{x})d\mathbf{x}+\mathrm{tr}(L_Z^{-1}\Psi)}$$

These bounds have similar structure with the ones of Affandi, Fox, Adams and Taskar, ICML (2014)

- The important difference is that the new bounds do not depend on the difficult to compute eigenvalues of the kernel matrix or the unknown eigenvalues of the full kernel operator
- So the current variational framework should be applicable to a wider class of DPPs

Discussion

Summary: Variational inference based on inducing variables provides a rigorous mechanism to approximate GPs and DPPs

Some challenges:

- Can we further reduce the computational complexity of these methods?
- Can we use similar ideas in other Bayesian non-parametric models such as those based on Dirichlet processes?

Appendix (proof of the upper bound)

Theorem 3. The marginal likelihood in standard GP regression is upper bounded as follows:

$$p(\mathbf{y}) \leq \frac{1}{(2\pi)^{\frac{n}{2}} \left|\mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf} + \sigma^2 I\right|^{\frac{1}{2}}} e^{-\frac{1}{2}\mathbf{y}^T\left(\mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf} + cI + \sigma^2 I\right)^{-1}\mathbf{y}},$$

where $c = \mathrm{tr}\left(\mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}\right)$. To prove this we need to state the following known result (see Horn and johnson, Matrix analysis, (1985), Corollary 7.7.4).

Lemma. Assume two positive semi-definite matrices A and B such that the difference $A - B$ is also positive semi-definite, i.e. $A \geq B$ (meaning $\mathbf{y}^T A\mathbf{y} \geq \mathbf{y}^T B\mathbf{y} \geq 0,\ \forall \mathbf{y}$). It holds

1. $|A| \geq |B|$.
2. If A and B are invertible (strictly positive definite), then $B^{-1} \geq A^{-1}$.

Appendix (proof of the upper bound)

Based on point 1 and given that $\mathbf{K}_{ff} \geq \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf} \Rightarrow$ $\mathbf{K}_{ff} + \sigma^2 I \geq \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf} + \sigma^2 I$ we obtain

$$|\mathbf{K}_{ff} + \sigma^2 I| \geq |\mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf} + \sigma^2 I|,$$

from which we conclude that

$$\frac{1}{(2\pi)^{\frac{n}{2}}|\mathbf{K}_{ff} + \sigma^2 I|^{\frac{1}{2}}} \leq \frac{1}{(2\pi)^{\frac{n}{2}}|\mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf} + \sigma^2 I|^{\frac{1}{2}}}.$$

Thus, to prove the upper bound it remains to show that $e^{-\frac{1}{2}\mathbf{y}^T\left(\mathbf{K}_{ff}+\sigma^2 I\right)^{-1}\mathbf{y}} \leq e^{-\frac{1}{2}\mathbf{y}^T\left(\mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}+cI+\sigma^2 I\right)^{-1}\mathbf{y}}$ or

$$\mathbf{y}^T\left(\mathbf{K}_{ff} + \sigma^2 I\right)^{-1}\mathbf{y} \geq \mathbf{y}^T\left(\mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf} + cI + \sigma^2 I\right)^{-1}\mathbf{y}, \quad \forall \mathbf{y}$$

Equivalently we can show the following

$$\mathbf{y}^T\left(\mathbf{K}_{ff} + \sigma^2 I\right)\mathbf{y} \leq \mathbf{y}^T\left(\mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf} + cI + \sigma^2 I\right)\mathbf{y}, \quad \forall \mathbf{y}$$

and then apply point 2 in the Lemma.

Appendix (proof of the upper bound)

Thus it suffices to show that

$$\mathbf{y}^T \left(\mathbf{K}_{ff} + \sigma^2 I\right) \mathbf{y} \leq \mathbf{y}^T \left(\mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf} + cI + \sigma^2 I\right) \mathbf{y}, \quad \forall \mathbf{y}$$

By cancelling terms the above reduces to

$$\mathbf{y}^T \left(\mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}\right) \mathbf{y} \leq c||\mathbf{y}||^2, \quad \forall \mathbf{y}$$

Denote $Q = \mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}$ and let $Q = U\Lambda U^T$ be its eigen-decomposition. Then

$$\mathbf{y}^T U\Lambda U^T \mathbf{y} = \mathbf{z}^T \Lambda \mathbf{z} = \sum_{i=1}^{n} \lambda_i z_i^2 \leq \lambda_{\max} \sum_{i=1}^{n} z_i^2 = \lambda_{\max}||\mathbf{y}||^2 \leq c||\mathbf{y}||^2$$

which completes the proof. Here, we used that $\mathbf{z} = U^T\mathbf{y}$, $||\mathbf{z}|| = ||\mathbf{y}||$, $\lambda_{\max} = max(\lambda_1, \ldots, \lambda_n)$ and $c = \text{tr}\left(\mathbf{K}_{ff} - \mathbf{K}_{fu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uf}\right) = \sum_{i=1}^{n} \lambda_i$.