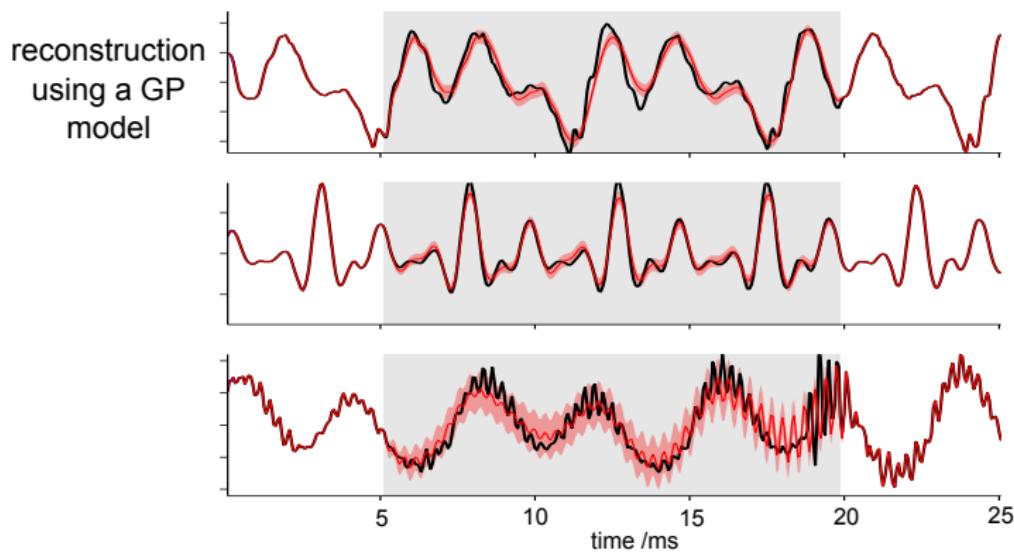
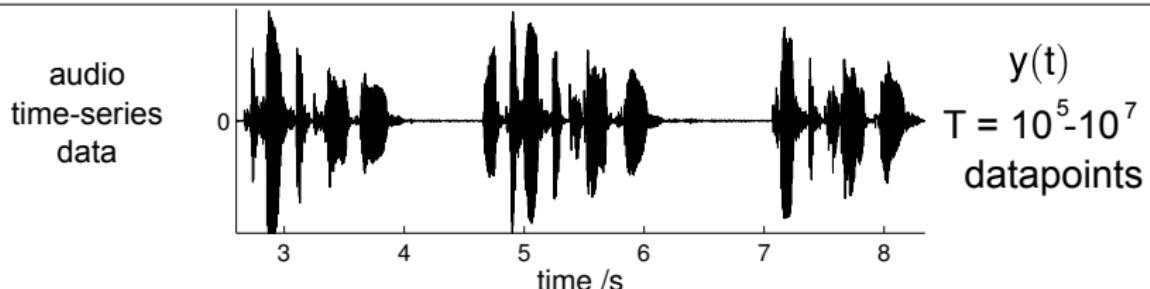


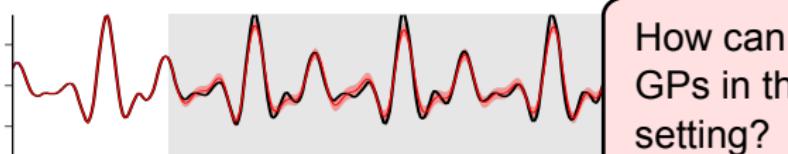
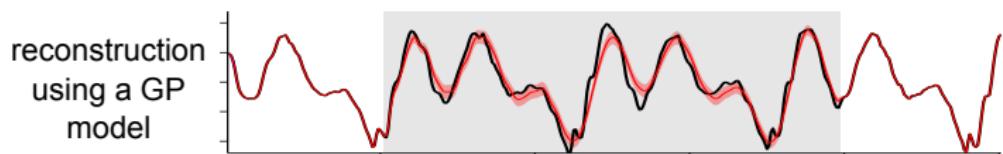
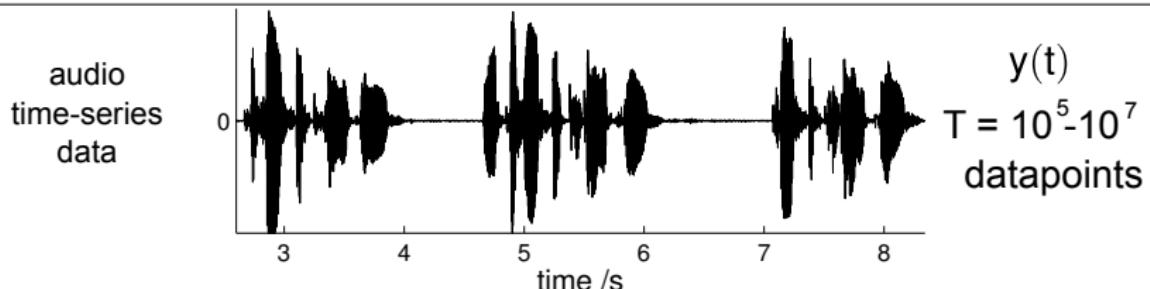
Approximate Inference for Scalable Non-linear Gaussian Process Models

Dr. Richard E. Turner (ret26@cam.ac.uk)
Computational and Biological Learning Lab, Department of
Engineering, University of Cambridge

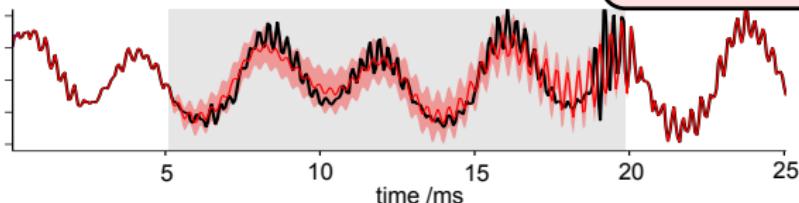
Motivating application 1: Audio modelling



Motivating application 1: Audio modelling

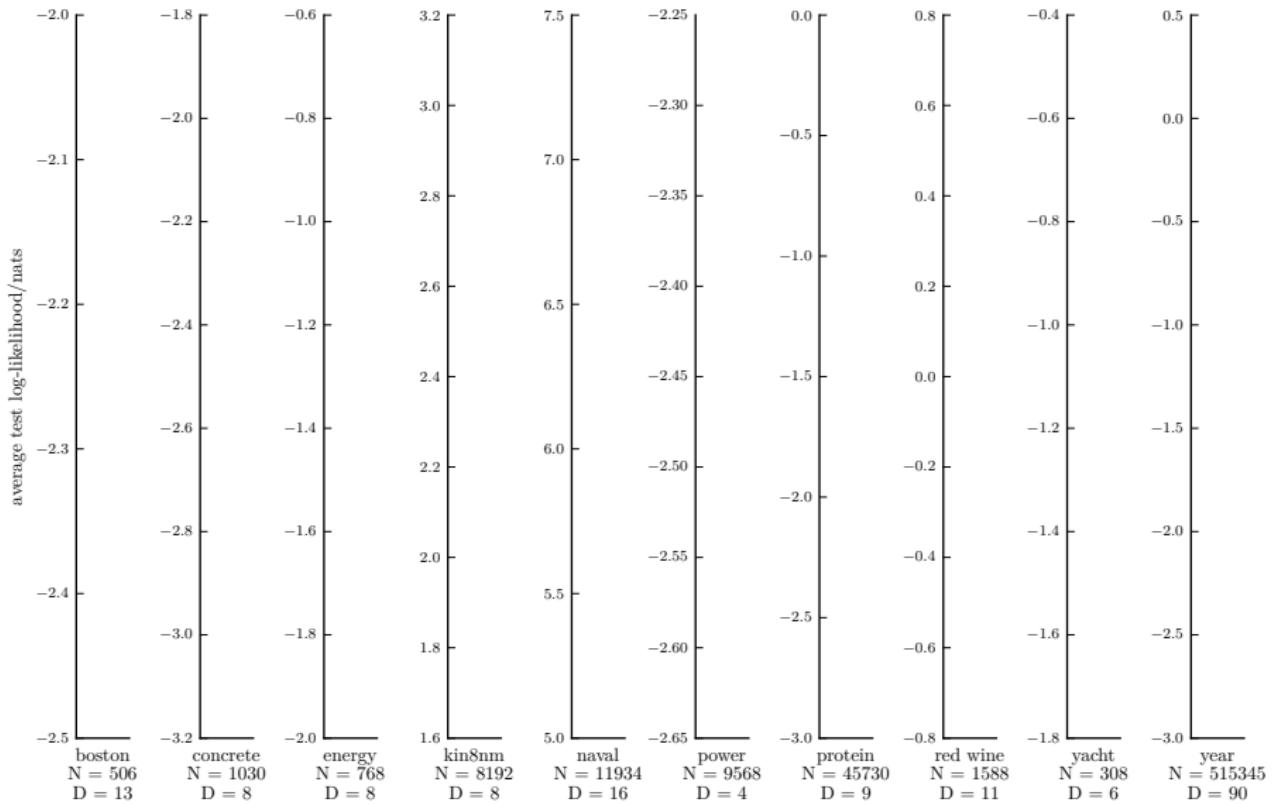


How can we use GPs in this setting?



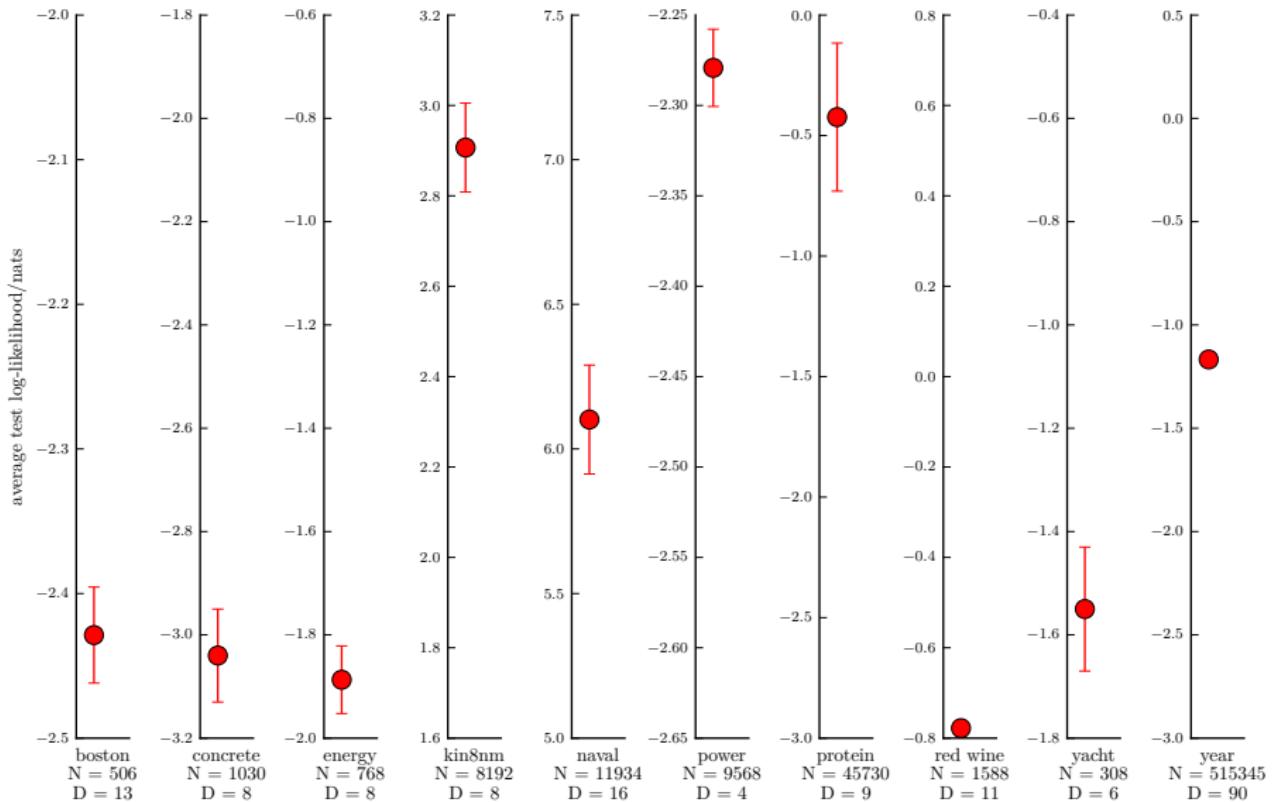
Motivating application 2: non-linear regression

BNN-deterministic BNN-sampling GP DGP



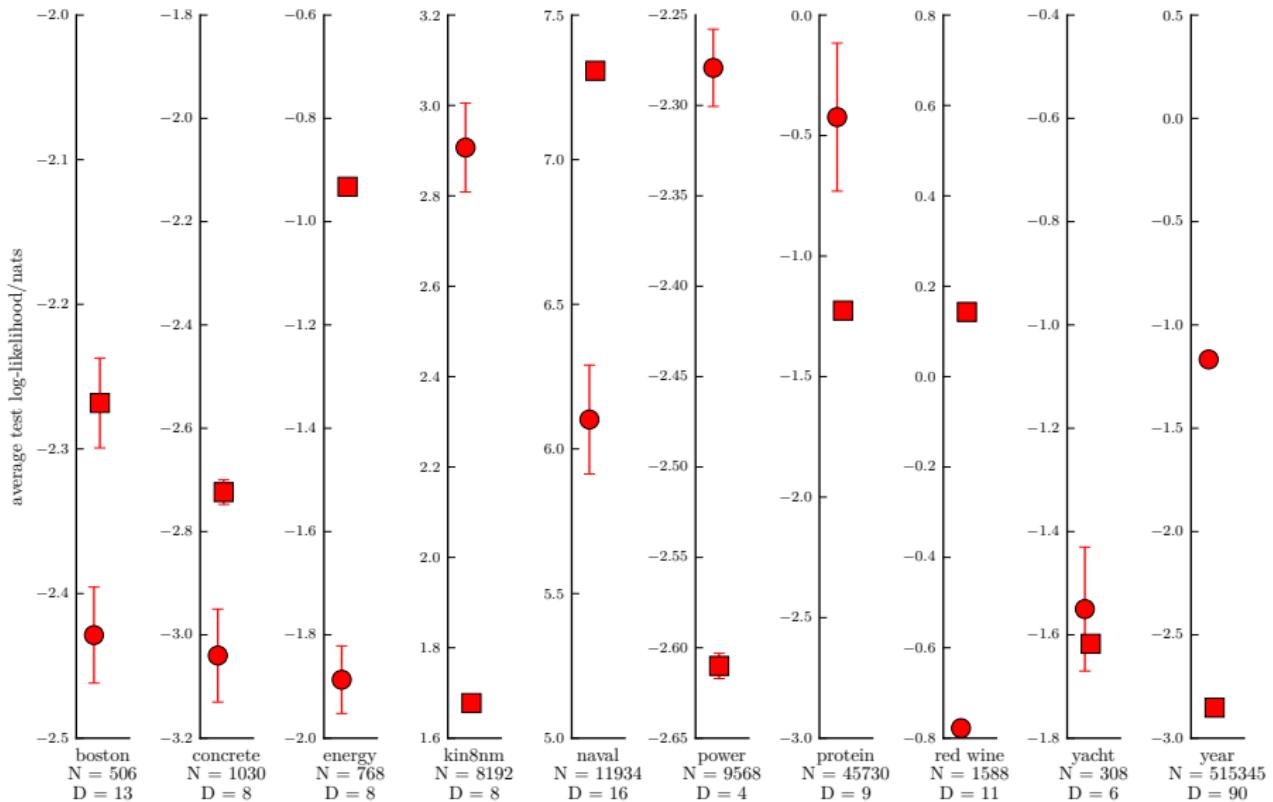
Motivating application 2: non-linear regression

BNN-deterministic BNN-sampling GP DGP



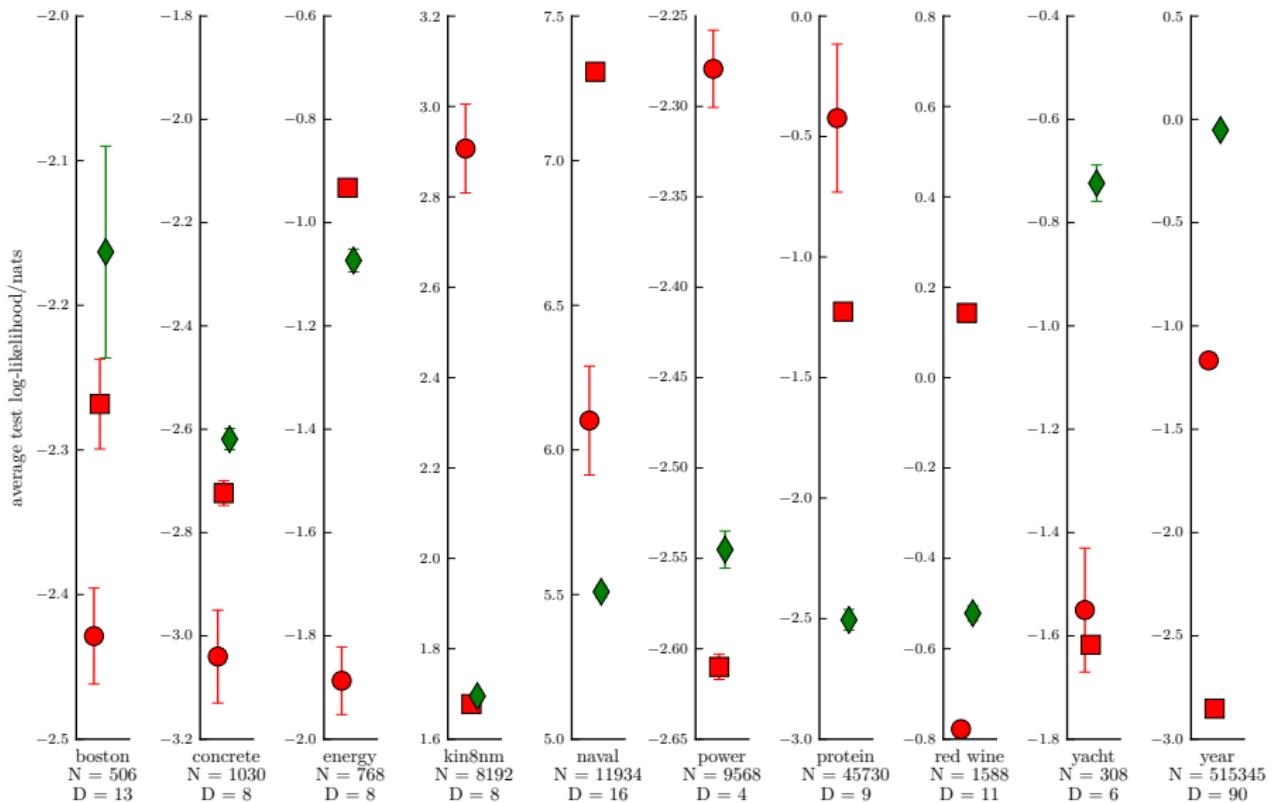
Motivating application 2: non-linear regression

BNN-deterministic BNN-sampling GP DGP



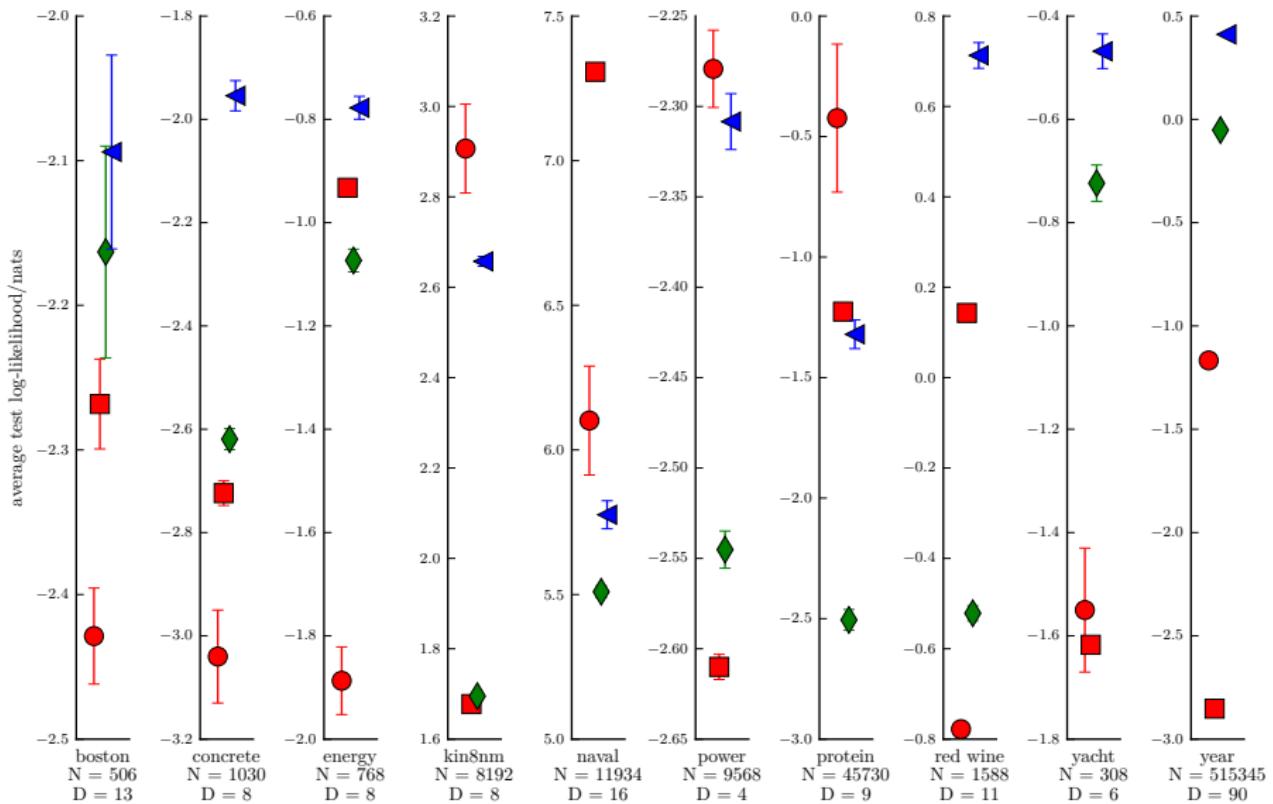
Motivating application 2: non-linear regression

BNN-deterministic BNN-sampling GP DGP



Motivating application 2: non-linear regression

BNN-deterministic BNN-sampling GP DGP



GP regression: introducing notation

Q1. What's the formal justification for how we were using GPs for regression?

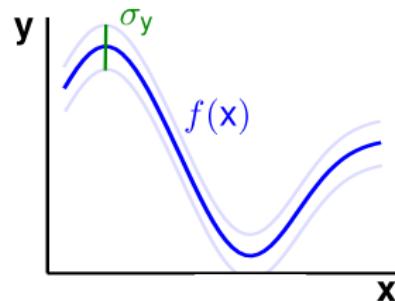
GP regression: introducing notation

Q1. What's the formal justification for how we were using GPs for regression?

generative model (like non-linear regression)

$$y(x) = f(x) + \epsilon \sigma_y$$

$$p(\epsilon) = \mathcal{N}(0, 1)$$



GP regression: introducing notation

Q1. What's the formal justification for how we were using GPs for regression?

generative model (like non-linear regression)

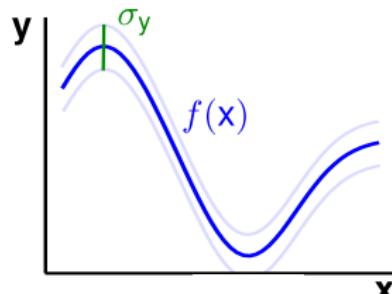
$$y(x) = f(x) + \epsilon \sigma_y$$

$$p(\epsilon) = \mathcal{N}(0, 1)$$

place GP prior over the non-linear function

$$p(f(x)|\theta) = \mathcal{GP}(0, K(x, x'))$$

$$K(x, x') = \sigma^2 \exp\left(-\frac{1}{2l^2}(x - x')^2\right) \quad (\text{smoothly wiggling functions expected})$$



GP regression: introducing notation

Q1. What's the formal justification for how we were using GPs for regression?

generative model (like non-linear regression)

$$y(x) = f(x) + \epsilon \sigma_y$$

$$p(\epsilon) = \mathcal{N}(0, 1)$$

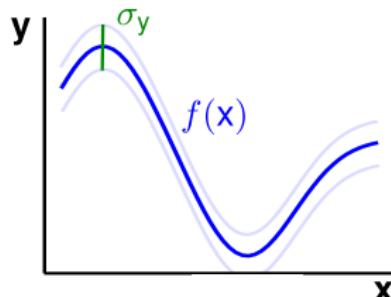
place GP prior over the non-linear function

$$p(f(x)|\theta) = \mathcal{GP}(0, K(x, x'))$$

$$K(x, x') = \sigma^2 \exp\left(-\frac{1}{2l^2}(x - x')^2\right) \quad (\text{smoothly wiggling functions expected})$$

sum of Gaussian variables = Gaussian: induces a GP over $y(x)$

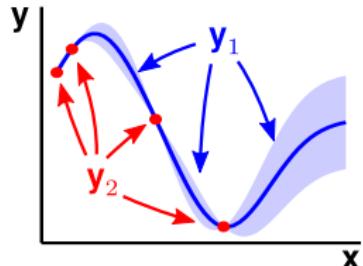
$$p(y(x)|\theta) = \mathcal{GP}(0, K(x, x') + I\sigma_y^2)$$



GP regression: introducing notation

Q3. How do we make predictions?

$$p(\mathbf{y}_1, \mathbf{y}_2) = \mathcal{N} \left(\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}, \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix} \right)$$
$$p(\mathbf{y}_1 | \mathbf{y}_2) = \frac{p(\mathbf{y}_1, \mathbf{y}_2)}{p(\mathbf{y}_2)}$$



$$\implies p(\mathbf{y}_1 | \mathbf{y}_2) = \mathcal{N} \left(\underbrace{\mathbf{a} + \mathbf{B}\mathbf{C}^{-1}(\mathbf{y}_2 - \mathbf{b})}_{\text{predictive mean}}, \mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^T \right)$$

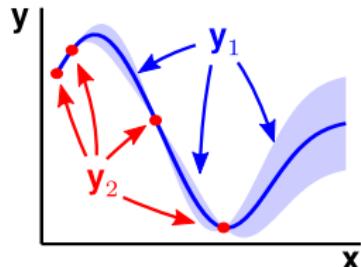
predictive mean

$$\mu_{\mathbf{y}_1 | \mathbf{y}_2} = \mathbf{a} + \mathbf{B}\mathbf{C}^{-1}(\mathbf{y}_2 - \mathbf{b})$$

GP regression: introducing notation

Q3. How do we make predictions?

$$p(\mathbf{y}_1, \mathbf{y}_2) = \mathcal{N} \left(\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}, \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix} \right)$$
$$p(\mathbf{y}_1 | \mathbf{y}_2) = \frac{p(\mathbf{y}_1, \mathbf{y}_2)}{p(\mathbf{y}_2)}$$



$$\implies p(\mathbf{y}_1 | \mathbf{y}_2) = \mathcal{N} \left(\underbrace{\mathbf{a} + \mathbf{B}\mathbf{C}^{-1}(\mathbf{y}_2 - \mathbf{b})}_{\text{predictive mean}}, \mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^T \right)$$

predictive mean

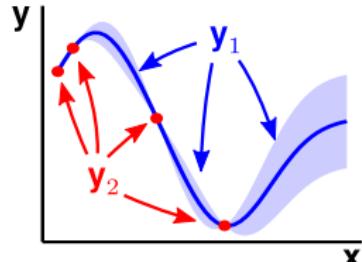
$$\begin{aligned} \mu_{\mathbf{y}_1 | \mathbf{y}_2} &= \mathbf{a} + \mathbf{B}\mathbf{C}^{-1}(\mathbf{y}_2 - \mathbf{b}) \\ &= \mathbf{B}\mathbf{C}^{-1}\mathbf{y}_2 \\ &= \mathbf{W}\mathbf{y}_2 \end{aligned}$$

linear in the data

GP regression: introducing notation

Q3. How do we make predictions?

$$p(\mathbf{y}_1, \mathbf{y}_2) = \mathcal{N} \left(\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}, \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix} \right)$$
$$p(\mathbf{y}_1 | \mathbf{y}_2) = \frac{p(\mathbf{y}_1, \mathbf{y}_2)}{p(\mathbf{y}_2)}$$



$$\Rightarrow p(\mathbf{y}_1 | \mathbf{y}_2) = \mathcal{N} \left(\underbrace{\mathbf{a} + \mathbf{B}\mathbf{C}^{-1}(\mathbf{y}_2 - \mathbf{b})}_{\text{predictive mean}}, \underbrace{\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^T}_{\text{predictive covariance}} \right)$$

predictive mean

$$\begin{aligned} \mu_{\mathbf{y}_1 | \mathbf{y}_2} &= \mathbf{a} + \mathbf{B}\mathbf{C}^{-1}(\mathbf{y}_2 - \mathbf{b}) \\ &= \mathbf{B}\mathbf{C}^{-1}\mathbf{y}_2 \\ &= \mathbf{W}\mathbf{y}_2 \end{aligned}$$

linear in the data

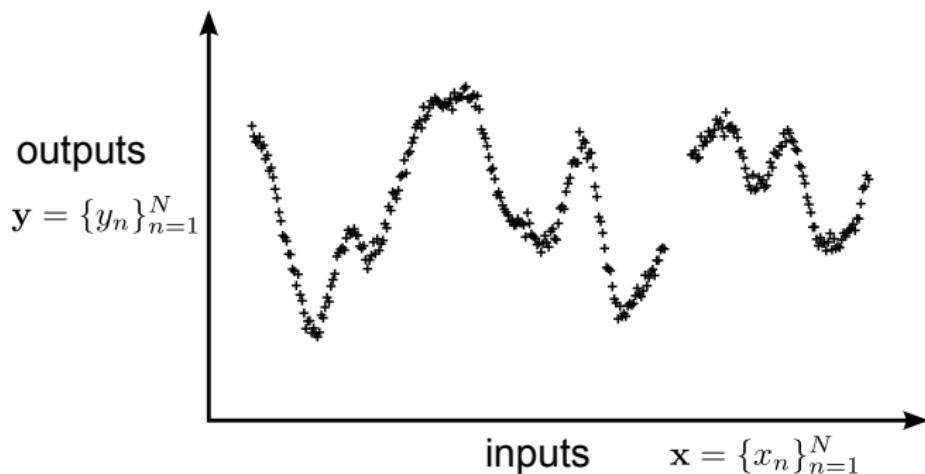
predictive covariance

$$\Sigma_{\mathbf{y}_1 | \mathbf{y}_2} = \mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^T$$

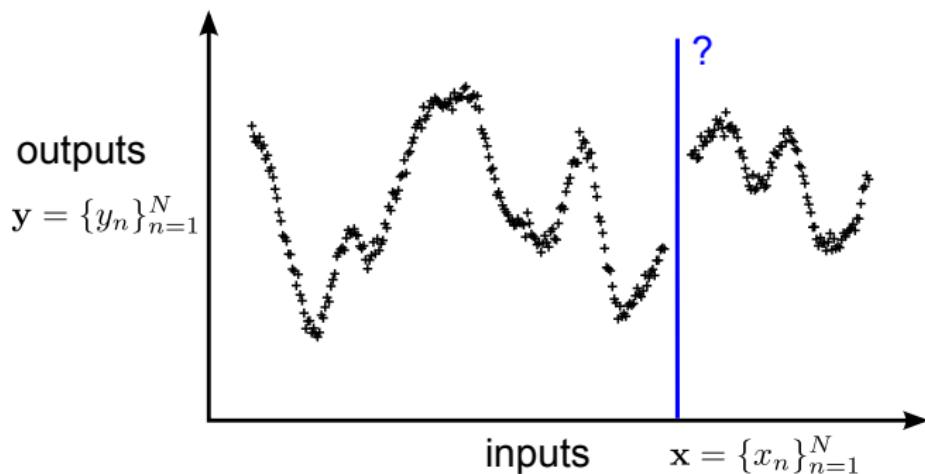
predictive uncertainty = prior uncertainty - reduction in uncertainty

predictions more confident than prior

Motivation: Gaussian Process Regression



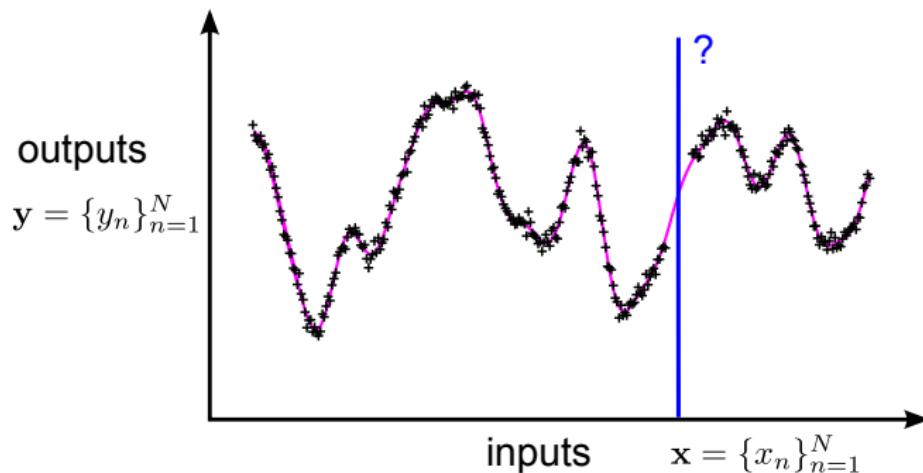
Motivation: Gaussian Process Regression



Motivation: Gaussian Process Regression

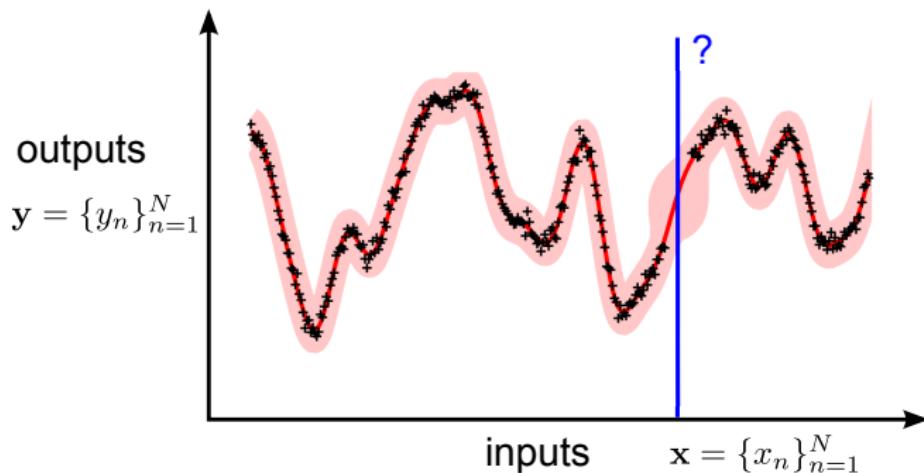
$$p(f|\theta) = \mathcal{GP}(f; 0, K_\theta)$$

$$p(y_n|f, x_n, \theta)$$



Motivation: Gaussian Process Regression

$$\begin{array}{ccc} p(f|\theta) = \mathcal{GP}(f; 0, K_\theta) & \xrightarrow{\text{inference \& learning}} & p(f|\mathbf{y}, \mathbf{x}, \theta) \\ p(y_n|f, x_n, \theta) & & p(\mathbf{y}|\mathbf{x}, \theta) \end{array}$$



Motivation: Gaussian Process Regression

$$p(f|\theta) = \mathcal{GP}(f; 0, K_\theta)$$

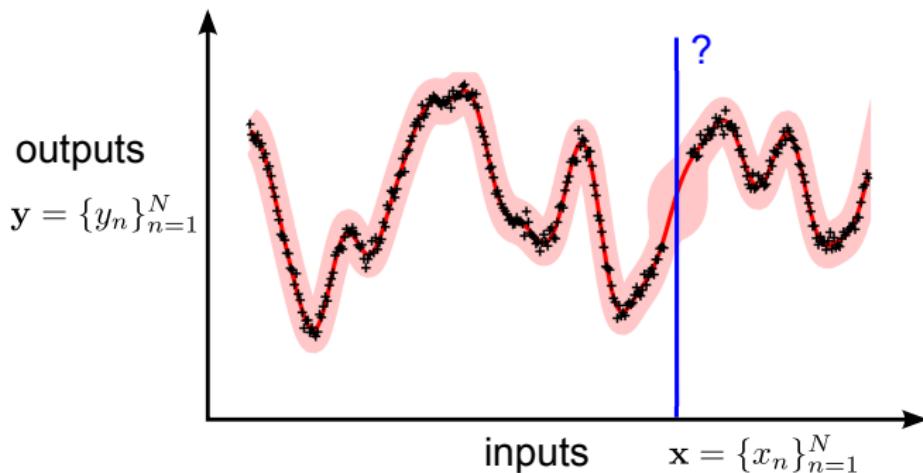
$$p(y_n|f, x_n, \theta)$$

inference & learning

intractabilities
computational $\mathcal{O}(N^3)$
analytic

$$p(f|\mathbf{y}, \mathbf{x}, \theta)$$

$$p(\mathbf{y}|\mathbf{x}, \theta)$$



Motivation: Gaussian Process Regression

$$p(f|\theta) = \mathcal{GP}(f; 0, K_\theta)$$

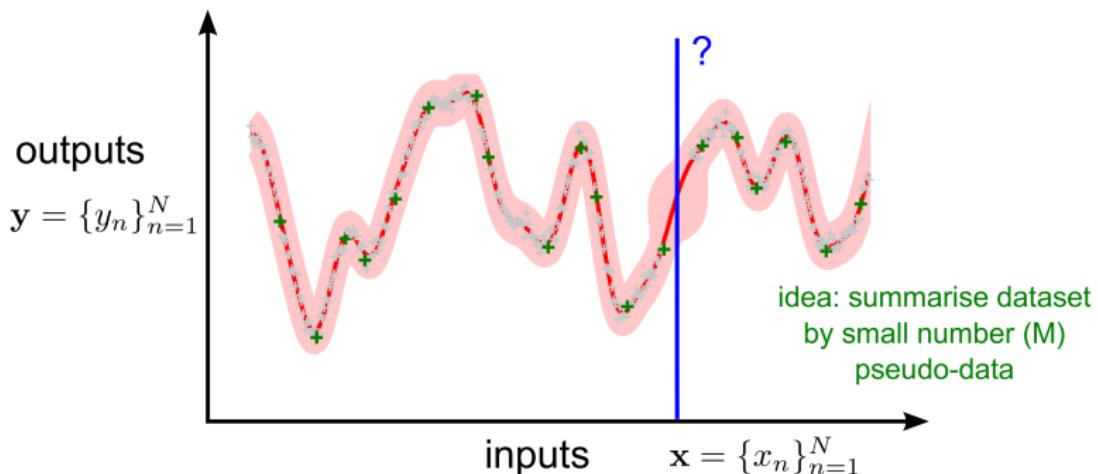
$$p(y_n|f, x_n, \theta)$$

inference & learning

intractabilities
computational $\mathcal{O}(N^3)$
analytic

$$p(f|\mathbf{y}, \mathbf{x}, \theta)$$

$$p(\mathbf{y}|\mathbf{x}, \theta)$$



A Brief History of Gaussian Process Approximations

FITC: Snelson et al. "Sparse Gaussian Processes using Pseudo-inputs"

PITC: Snelson et al. "Local and global sparse Gaussian process approximations"

EP: Csato and Opper 2002 / Qi et al. "Sparse-posterior Gaussian Processes for general likelihoods."

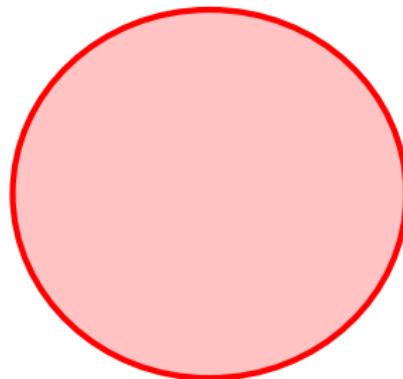
VFE: Titsias "Variational Learning of Inducing Variables in Sparse Gaussian Processes"

DTC / PP: Seeger et al. "Fast Forward Selection to Speed Up Sparse Gaussian Process Regression"

A Brief History of Gaussian Process Approximations

approximate generative model
exact inference

$\text{div}[p(\mathbf{f}, \mathbf{y}) || q(\mathbf{f}, \mathbf{y})]$



FITC: Snelson et al. "Sparse Gaussian Processes using Pseudo-inputs"

PITC: Snelson et al. "Local and global sparse Gaussian process approximations"

EP: Csato and Opper 2002 / Qi et al. "Sparse-posterior Gaussian Processes for general likelihoods."

VFE: Titsias "Variational Learning of Inducing Variables in Sparse Gaussian Processes"

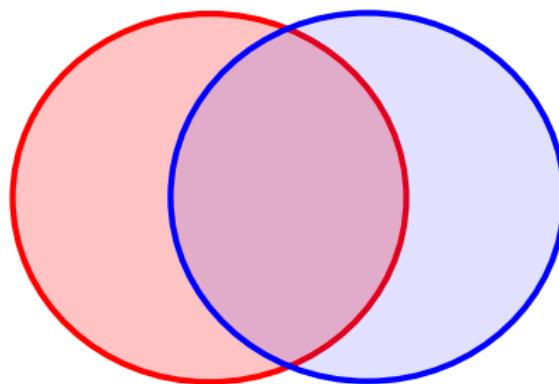
DTC / PP: Seeger et al. "Fast Forward Selection to Speed Up Sparse Gaussian Process Regression"

A Brief History of Gaussian Process Approximations

approximate generative model
exact inference

methods employing
pseudo-data

$$\text{div}[p(\mathbf{f}, \mathbf{y}) || q(\mathbf{f}, \mathbf{y})]$$



FITC: Snelson et al. "Sparse Gaussian Processes using Pseudo-inputs"

PITC: Snelson et al. "Local and global sparse Gaussian process approximations"

EP: Csato and Opper 2002 / Qi et al. "Sparse-posterior Gaussian Processes for general likelihoods."

VFE: Titsias "Variational Learning of Inducing Variables in Sparse Gaussian Processes"

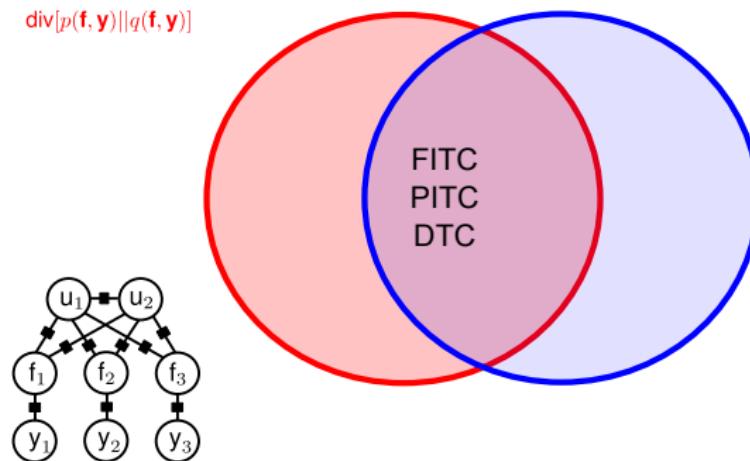
DTC / PP: Seeger et al. "Fast Forward Selection to Speed Up Sparse Gaussian Process Regression"

A Brief History of Gaussian Process Approximations

approximate generative model
exact inference

methods employing
pseudo-data

$$\text{div}[p(\mathbf{f}, \mathbf{y}) || q(\mathbf{f}, \mathbf{y})]$$



FITC: Snelson et al. "Sparse Gaussian Processes using Pseudo-inputs"

PITC: Snelson et al. "Local and global sparse Gaussian process approximations"

EP: Csato and Opper 2002 / Qi et al. "Sparse-posterior Gaussian Processes for general likelihoods."

VFE: Titsias "Variational Learning of Inducing Variables in Sparse Gaussian Processes"

DTC / PP: Seeger et al. "Fast Forward Selection to Speed Up Sparse Gaussian Process Regression"

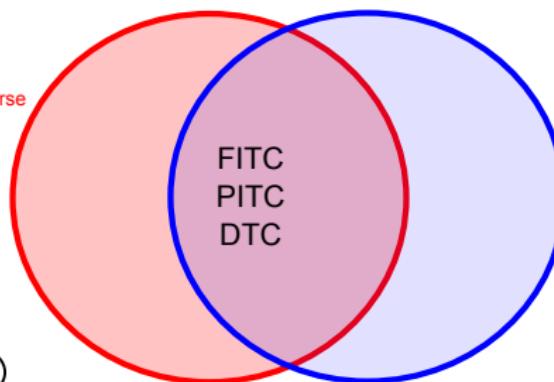
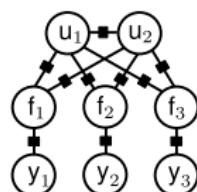
A Brief History of Gaussian Process Approximations

approximate generative model
exact inference

methods employing
pseudo-data

$$\text{div}[p(\mathbf{f}, \mathbf{y}) || q(\mathbf{f}, \mathbf{y})]$$

A Unifying View of Sparse
Approximate Gaussian
Process Regression
Quinonero-Candela &
Rasmussen, 2005
(FITC, PITC, DTC)



FITC: Snelson et al. "Sparse Gaussian Processes using Pseudo-inputs"

PITC: Snelson et al. "Local and global sparse Gaussian process approximations"

EP: Csato and Opper 2002 / Qi et al. "Sparse-posterior Gaussian Processes for general likelihoods."

VFE: Titsias "Variational Learning of Inducing Variables in Sparse Gaussian Processes"

DTC / PP: Seeger et al. "Fast Forward Selection to Speed Up Sparse Gaussian Process Regression"

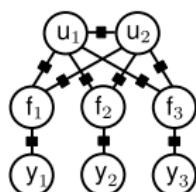
A Brief History of Gaussian Process Approximations

approximate generative model
exact inference methods employing
pseudo-data exact generative model
approximate inference

$$\text{div}[p(\mathbf{f}, \mathbf{y}) || q(\mathbf{f}, \mathbf{y})]$$

$$\text{div}[p(\mathbf{f} | \mathbf{y}) || q(\mathbf{f})]$$

A Unifying View of Sparse
Approximate Gaussian
Process Regression
Quinonero-Candela &
Rasmussen, 2005
(FITC, PITC, DTC)



FITC: Snelson et al. "Sparse Gaussian Processes using Pseudo-inputs"

PITC: Snelson et al. "Local and global sparse Gaussian process approximations"

EP: Csato and Opper 2002 / Qi et al. "Sparse-posterior Gaussian Processes for general likelihoods."

VFE: Titsias "Variational Learning of Inducing Variables in Sparse Gaussian Processes"

DTC / PP: Seeger et al. "Fast Forward Selection to Speed Up Sparse Gaussian Process Regression"

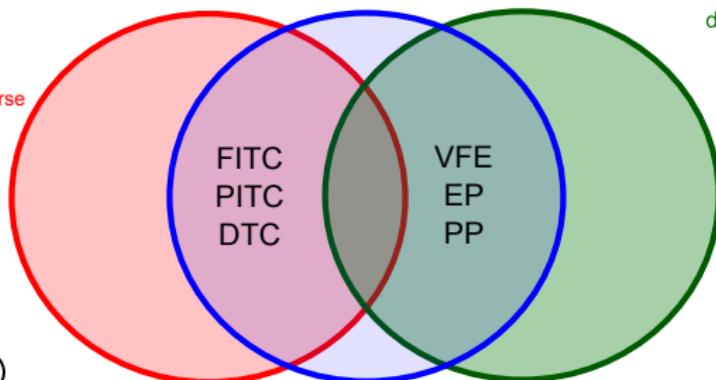
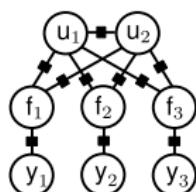
A Brief History of Gaussian Process Approximations

approximate generative model
exact inference methods employing
pseudo-data exact generative model
approximate inference

$$\text{div}[p(\mathbf{f}, \mathbf{y}) || q(\mathbf{f}, \mathbf{y})]$$

$$\text{div}[p(\mathbf{f} | \mathbf{y}) || q(\mathbf{f})]$$

A Unifying View of Sparse
Approximate Gaussian
Process Regression
Quinonero-Candela &
Rasmussen, 2005
(FITC, PITC, DTC)



FITC: Snelson et al. "Sparse Gaussian Processes using Pseudo-inputs"

PITC: Snelson et al. "Local and global sparse Gaussian process approximations"

EP: Csato and Opper 2002 / Qi et al. "Sparse-posterior Gaussian Processes for general likelihoods."

VFE: Titsias "Variational Learning of Inducing Variables in Sparse Gaussian Processes"

DTC / PP: Seeger et al. "Fast Forward Selection to Speed Up Sparse Gaussian Process Regression"

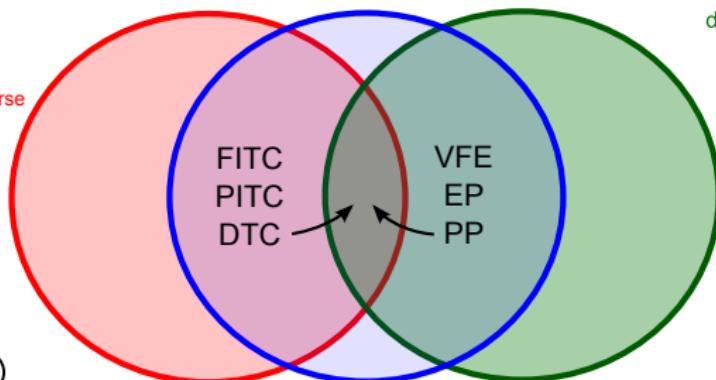
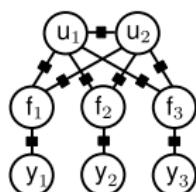
A Brief History of Gaussian Process Approximations

approximate generative model
exact inference methods employing
pseudo-data exact generative model
approximate inference

$$\text{div}[p(\mathbf{f}, \mathbf{y}) || q(\mathbf{f}, \mathbf{y})]$$

$$\text{div}[p(\mathbf{f} | \mathbf{y}) || q(\mathbf{f})]$$

A Unifying View of Sparse
Approximate Gaussian
Process Regression
Quinonero-Candela &
Rasmussen, 2005
(FITC, PITC, DTC)



FITC: Snelson et al. "Sparse Gaussian Processes using Pseudo-inputs"

PITC: Snelson et al. "Local and global sparse Gaussian process approximations"

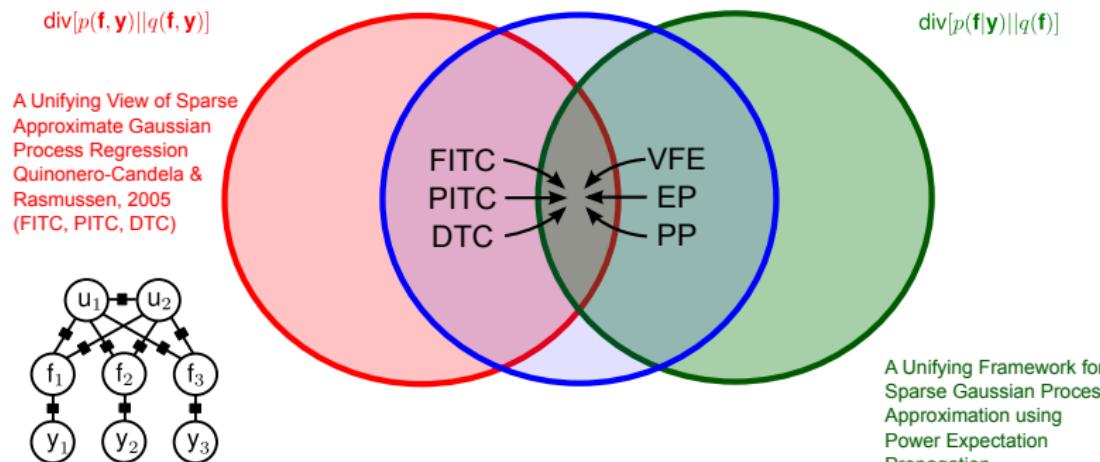
EP: Csato and Opper 2002 / Qi et al. "Sparse-posterior Gaussian Processes for general likelihoods."

VFE: Titsias "Variational Learning of Inducing Variables in Sparse Gaussian Processes"

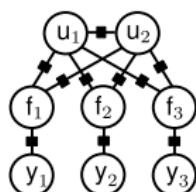
DTC / PP: Seeger et al. "Fast Forward Selection to Speed Up Sparse Gaussian Process Regression"

A Brief History of Gaussian Process Approximations

approximate generative model
exact inference methods employing
pseudo-data exact generative model
approximate inference



$\text{div}[p(\mathbf{f}, \mathbf{y}) || q(\mathbf{f}, \mathbf{y})]$
A Unifying View of Sparse
Approximate Gaussian
Process Regression
Quinonero-Candela &
Rasmussen, 2005
(FITC, PITC, DTC)



$\text{div}[p(\mathbf{f} | \mathbf{y}) || q(\mathbf{f})]$
A Unifying Framework for
Sparse Gaussian Process
Approximation using
Power Expectation
Propagation
Bui, Yan and Turner, 2016
(VFE, EP, FITC, PITC ...)

FITC: Snelson et al. "Sparse Gaussian Processes using Pseudo-inputs"

PITC: Snelson et al. "Local and global sparse Gaussian process approximations"

EP: Csato and Opper 2002 / Qi et al. "Sparse-posterior Gaussian Processes for general likelihoods."

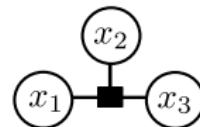
VFE: Titsias "Variational Learning of Inducing Variables in Sparse Gaussian Processes"

DTC / PP: Seeger et al. "Fast Forward Selection to Speed Up Sparse Gaussian Process Regression"

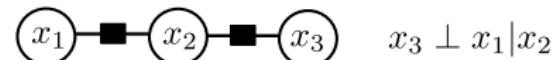
Factor Graphs: reminder

factor graph examples

$$p(x_1, x_2, x_3) = g(x_1, x_2, x_3)$$



$$p(x_1, x_2, x_3) = g_1(x_1, x_2)g_2(x_2, x_3)$$

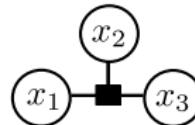


$$x_3 \perp x_1 | x_2$$

Factor Graphs: reminder

factor graph examples

$$p(x_1, x_2, x_3) = g(x_1, x_2, x_3)$$



$$p(x_1, x_2, x_3) = g_1(x_1, x_2)g_2(x_2, x_3)$$



$$x_3 \perp x_1 | x_2$$

what is the minimal factor graph for this multivariate Gaussian?

$$p(\mathbf{x}|\mu, \Sigma) = \mathcal{N}(\mathbf{x}; \mu, \Sigma)$$

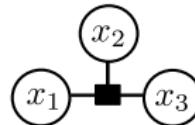
4 dimensional

$$\Sigma = \begin{bmatrix} 1 & 1/2 & 1/2 & 1/4 \\ 1/2 & 5/4 & 1/4 & 1/8 \\ 1/2 & 1/4 & 5/4 & 5/8 \\ 1/4 & 1/8 & 5/8 & 21/16 \end{bmatrix} \quad \Sigma^{-1} = \begin{bmatrix} 1.5 & -1/2 & -1/2 & 0 \\ -1/2 & 1 & 0 & 0 \\ -1/2 & 0 & 5/4 & -1/2 \\ 0 & 0 & -1/2 & 1 \end{bmatrix}$$

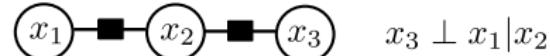
Factor Graphs: reminder

factor graph examples

$$p(x_1, x_2, x_3) = g(x_1, x_2, x_3)$$



$$p(x_1, x_2, x_3) = g_1(x_1, x_2)g_2(x_2, x_3)$$



$$x_3 \perp x_1 | x_2$$

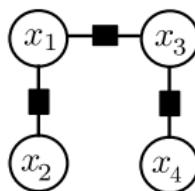
what is the minimal factor graph for this multivariate Gaussian?

$$p(\mathbf{x}|\mu, \Sigma) = \mathcal{N}(\mathbf{x}; \mu, \Sigma)$$

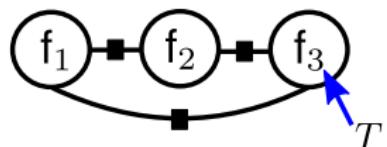
4 dimensional

$$\Sigma = \begin{bmatrix} 1 & 1/2 & 1/2 & 1/4 \\ 1/2 & 5/4 & 1/4 & 1/8 \\ 1/2 & 1/4 & 5/4 & 5/8 \\ 1/4 & 1/8 & 5/8 & 21/16 \end{bmatrix} \quad \Sigma^{-1} = \begin{bmatrix} 1.5 & -1/2 & -1/2 & 0 \\ -1/2 & 1 & 0 & 0 \\ -1/2 & 0 & 5/4 & -1/2 \\ 0 & 0 & -1/2 & 1 \end{bmatrix}$$

solution:



Fully independent training conditional (FITC) approximation

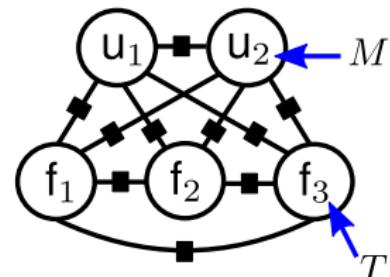


construct new generative model (with pseudo-data)
cheaper to perform exact learning and inference
calibrated to original

Fully independent training conditional (FITC) approximation

1. augment model with $M < T$ pseudo data

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

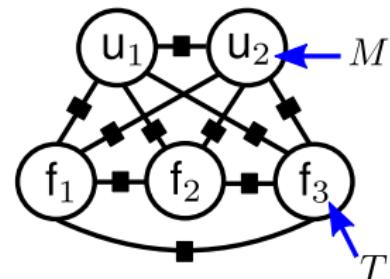


construct new generative model (with pseudo-data)
cheaper to perform exact learning and inference
calibrated to original

Fully independent training conditional (FITC) approximation

1. augment model with $M < T$ pseudo data

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



2. remove some of the dependencies

(results in simpler model)

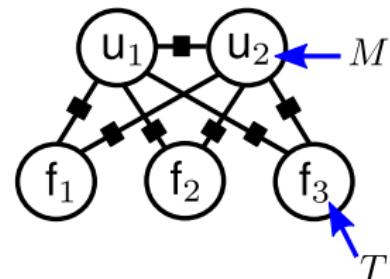
all factors

construct new generative model (with pseudo-data)
cheaper to perform exact learning and inference
calibrated to original

Fully independent training conditional (FITC) approximation

1. augment model with $M < T$ pseudo data

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



2. remove some of the dependencies

(results in simpler model)

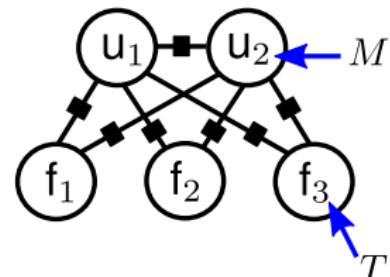


construct new generative model (with pseudo-data)
cheaper to perform exact learning and inference
calibrated to original

Fully independent training conditional (FITC) approximation

1. augment model with $M < T$ pseudo data

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



2. remove some of the dependencies
(results in simpler model)



3. calibrate model

(e.g. using KL divergence, many choices)

$$\arg \min_{q(\mathbf{u}), \{q(\mathbf{f}_t | \mathbf{u})\}_{t=1}^T} \text{KL}(p(\mathbf{f}, \mathbf{u}) || q(\mathbf{u}) \prod_{t=1}^T q(\mathbf{f}_t | \mathbf{u})) \implies \begin{aligned} q(\mathbf{u}) &= p(\mathbf{u}) \\ q(\mathbf{f}_t | \mathbf{u}) &= p(\mathbf{f}_t | \mathbf{u}) \end{aligned}$$

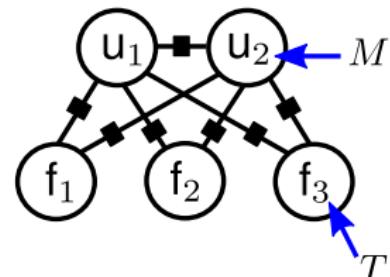
equal to exact conditionals

construct new generative model (with pseudo-data)
cheaper to perform exact learning and inference
calibrated to original

Fully independent training conditional (FITC) approximation

1. augment model with $M < T$ pseudo data

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



2. remove some of the dependencies
(results in simpler model)



3. calibrate model

(e.g. using KL divergence, many choices)

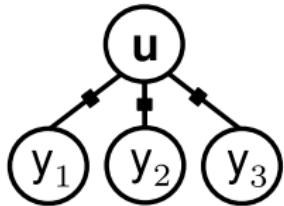
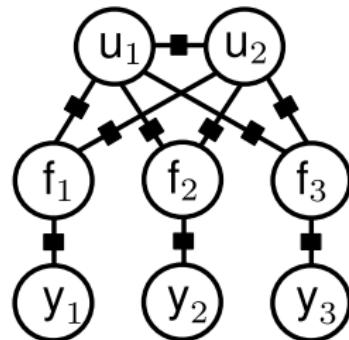
$$\arg \min_{q(\mathbf{u}), \{q(\mathbf{f}_t | \mathbf{u})\}_{t=1}^T} \text{KL}(p(\mathbf{f}, \mathbf{u}) || q(\mathbf{u}) \prod_{t=1}^T q(\mathbf{f}_t | \mathbf{u})) \implies \begin{aligned} q(\mathbf{u}) &= p(\mathbf{u}) \\ q(\mathbf{f}_t | \mathbf{u}) &= p(\mathbf{f}_t | \mathbf{u}) \end{aligned}$$

equal to exact conditionals

construct new generative model (with pseudo-data)
cheaper to perform exact learning and inference
calibrated to original

indirect
posterior
approximation

Fully independent training conditional (FITC) approximation

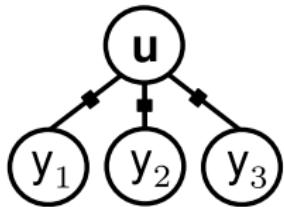
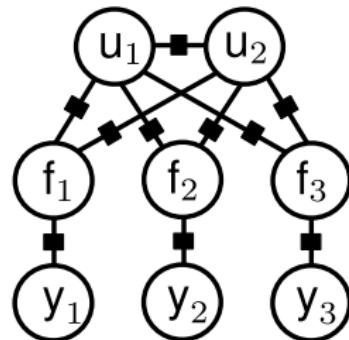


construct new generative model (with pseudo-data)
cheaper to perform exact learning and inference
calibrated to original

indirect
posterior
approximation

Fully independent training conditional (FITC) approximation

$$q(\mathbf{u}) = p(\mathbf{u}) = \mathcal{N}(\mathbf{u}; 0, K_{uu})$$



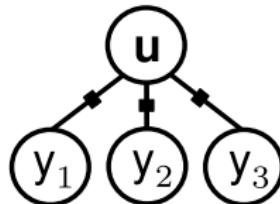
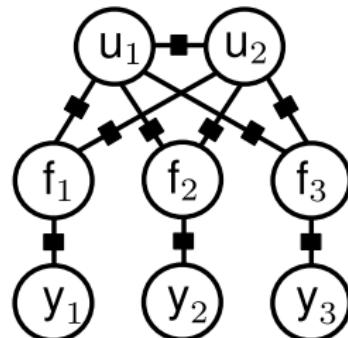
construct new generative model (with pseudo-data)
cheaper to perform exact learning and inference
calibrated to original

indirect
posterior
approximation

Fully independent training conditional (FITC) approximation

$$q(\mathbf{u}) = p(\mathbf{u}) = \mathcal{N}(\mathbf{u}; 0, K_{uu})$$

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



construct new generative model (with pseudo-data)
cheaper to perform exact learning and inference
calibrated to original

indirect
posterior
approximation

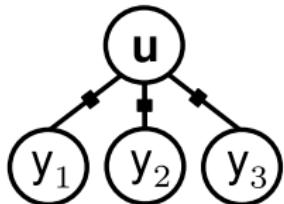
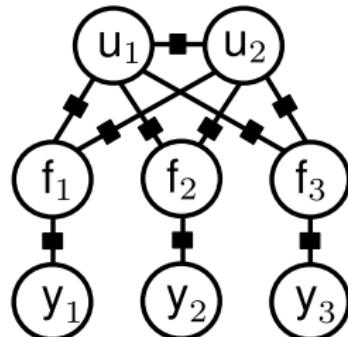
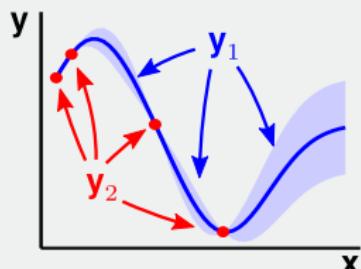
Fully independent training conditional (FITC) approximation

$$q(\mathbf{u}) = p(\mathbf{u}) = \mathcal{N}(\mathbf{u}; 0, K_{uu})$$

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

How do we make predictions?

$$p(\mathbf{y}_1 | \mathbf{y}_2) = \mathcal{N}(\mathbf{y}_1; \Sigma_{12}\Sigma_{22}^{-1}\mathbf{y}_2, \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{12}^\top)$$



construct new generative model (with pseudo-data)
cheaper to perform exact learning and inference
calibrated to original

indirect
posterior
approximation

Fully independent training conditional (FITC) approximation

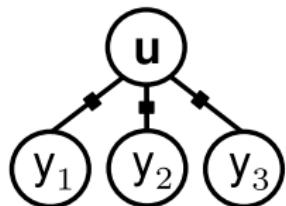
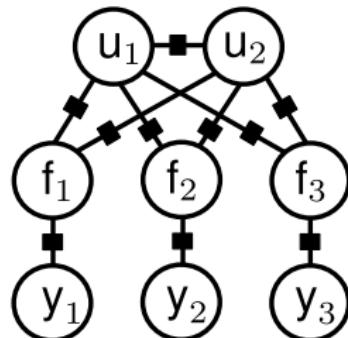
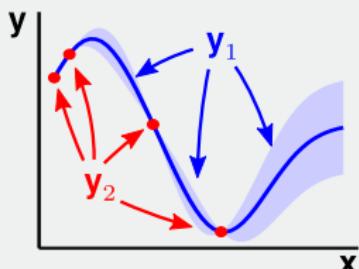
$$q(\mathbf{u}) = p(\mathbf{u}) = \mathcal{N}(\mathbf{u}; 0, K_{uu})$$

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

$$= \mathcal{N}(f_t; K_{f_t u} K_{uu}^{-1} \mathbf{u}, K_{f_t f_t} - K_{f_t u} K_{uu}^{-1} K_{u f_t})$$

How do we make predictions?

$$p(\mathbf{y}_1 | \mathbf{y}_2) = \mathcal{N}(\mathbf{y}_1; \Sigma_{12} \Sigma_{22}^{-1} \mathbf{y}_2, \Sigma_{11} - \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{12}^\top)$$



construct new generative model (with pseudo-data)
cheaper to perform exact learning and inference
calibrated to original

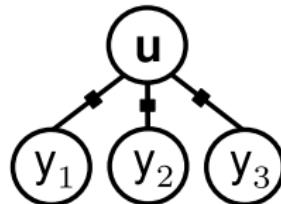
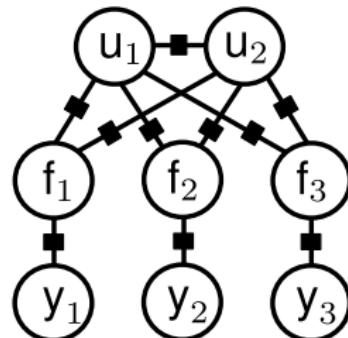
indirect
posterior
approximation

Fully independent training conditional (FITC) approximation

$$q(\mathbf{u}) = p(\mathbf{u}) = \mathcal{N}(\mathbf{u}; 0, K_{uu})$$

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

$$= \mathcal{N}(f_t; K_{f_t u} K_{uu}^{-1} \mathbf{u}, K_{f_t f_t} - K_{f_t u} K_{uu}^{-1} K_{u f_t})$$



construct new generative model (with pseudo-data)
cheaper to perform exact learning and inference
calibrated to original

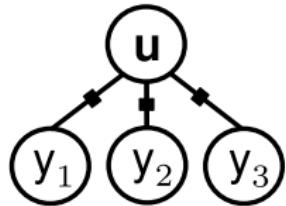
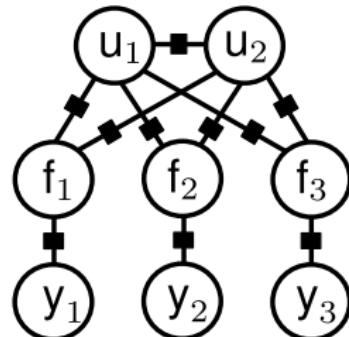
indirect
posterior
approximation

Fully independent training conditional (FITC) approximation

$$q(\mathbf{u}) = p(\mathbf{u}) = \mathcal{N}(\mathbf{u}; 0, K_{uu})$$

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

$$= \mathcal{N}(f_t; K_{f_t u} K_{uu}^{-1} \mathbf{u}, \underbrace{K_{f_t f_t} - K_{f_t u} K_{uu}^{-1} K_{uf_t}}_{D_{tt}})$$



construct new generative model (with pseudo-data)
cheaper to perform exact learning and inference
calibrated to original

indirect
posterior
approximation

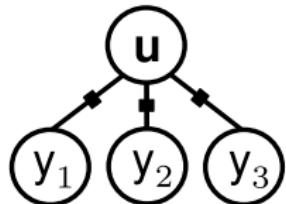
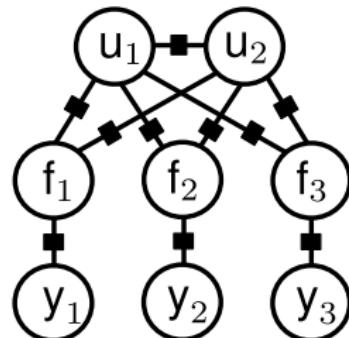
Fully independent training conditional (FITC) approximation

$$q(\mathbf{u}) = p(\mathbf{u}) = \mathcal{N}(\mathbf{u}; 0, K_{uu})$$

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

$$= \mathcal{N}(f_t; K_{f_t u} K_{uu}^{-1} \mathbf{u}, \underbrace{K_{f_t f_t} - K_{f_t u} K_{uu}^{-1} K_{uf_t}}_{D_{tt}})$$

$$q(y_t | f_t) = p(y_t | f_t) = \mathcal{N}(y_t; f_t, \sigma_y^2)$$



construct new generative model (with pseudo-data)
cheaper to perform exact learning and inference
calibrated to original

indirect
posterior
approximation

Fully independent training conditional (FITC) approximation

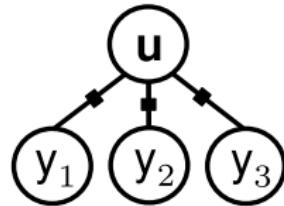
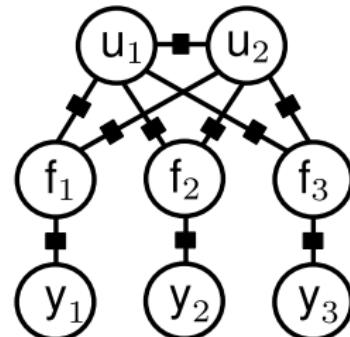
$$q(\mathbf{u}) = p(\mathbf{u}) = \mathcal{N}(\mathbf{u}; 0, K_{uu})$$

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

$$= \mathcal{N}(f_t; K_{f_t u} K_{uu}^{-1} \mathbf{u}, \underbrace{K_{f_t f_t} - K_{f_t u} K_{uu}^{-1} K_{uf_t}}_{D_{tt}})$$

$$q(y_t | f_t) = p(y_t | f_t) = \mathcal{N}(y_t; f_t, \sigma_y^2)$$

cost of computing likelihood is $\mathcal{O}(TM^2)$



construct new generative model (with pseudo-data)
cheaper to perform exact learning and inference
calibrated to original

indirect
posterior
approximation

Fully independent training conditional (FITC) approximation

$$q(\mathbf{u}) = p(\mathbf{u}) = \mathcal{N}(\mathbf{u}; 0, K_{uu})$$

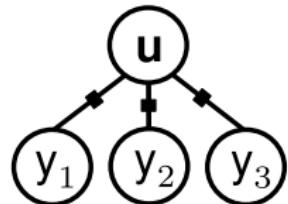
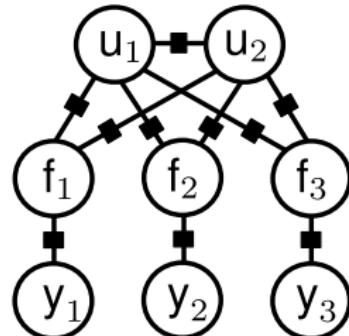
$$q(f_t | \mathbf{u}) = p(f_t | \mathbf{u})$$

$$= \mathcal{N}(f_t; K_{f_t u} K_{uu}^{-1} \mathbf{u}, \underbrace{K_{f_t f_t} - K_{f_t u} K_{uu}^{-1} K_{uf_t}}_{D_{tt}})$$

$$q(y_t | f_t) = p(y_t | f_t) = \mathcal{N}(y_t; f_t, \sigma_y^2)$$

cost of computing likelihood is $\mathcal{O}(TM^2)$

$$p(\mathbf{y}_t | \theta) = \mathcal{N}(\mathbf{y}; 0, K_{fu} K_{uu}^{-1} K_{uu} K_{uf}^{-1} + D + \sigma_y^2 I)$$



construct new generative model (with pseudo-data)

cheaper to perform exact learning and inference

calibrated to original

indirect
posterior
approximation

Fully independent training conditional (FITC) approximation

$$q(\mathbf{u}) = p(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mathbf{0}, K_{uu})$$

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

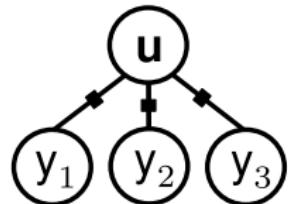
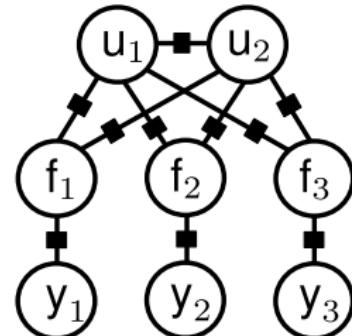
$$= \mathcal{N}(f_t; K_{f_t u} K_{uu}^{-1} \mathbf{u}, \underbrace{K_{f_t f_t} - K_{f_t u} K_{uu}^{-1} K_{uf_t}}_{D_{tt}})$$

$$q(y_t | f_t) = p(y_t | f_t) = \mathcal{N}(y_t; f_t, \sigma_y^2)$$

cost of computing likelihood is $\mathcal{O}(TM^2)$

$$p(\mathbf{y}_t | \theta) = \mathcal{N}(\mathbf{y}; \mathbf{0}, K_{fu} K_{uu}^{-1} K_{uu} K_{uf}^{-1} K_{uf} + D + \sigma_y^2 I)$$

$$= \mathcal{N}(\mathbf{y}; \mathbf{0}, K_{fu} K_{uu}^{-1} K_{uf} + D + \sigma_y^2 I)$$



construct new generative model (with pseudo-data)
cheaper to perform exact learning and inference
calibrated to original

indirect posterior approximation

Fully independent training conditional (FITC) approximation

$$q(\mathbf{u}) = p(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mathbf{0}, K_{uu})$$

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

$$= \mathcal{N}(f_t; K_{f_t u} K_{uu}^{-1} \mathbf{u}, \underbrace{K_{f_t f_t} - K_{f_t u} K_{uu}^{-1} K_{uf_t}}_{D_{tt}})$$

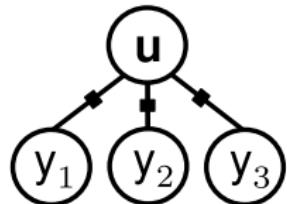
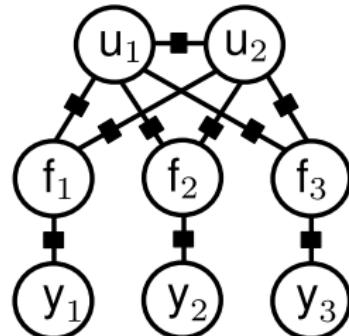
$$q(y_t | f_t) = p(y_t | f_t) = \mathcal{N}(y_t; f_t, \sigma_y^2)$$

cost of computing likelihood is $\mathcal{O}(TM^2)$

$$p(\mathbf{y}_t | \theta) = \mathcal{N}(\mathbf{y}; \mathbf{0}, K_{fu} K_{uu}^{-1} K_{uu} K_{uf}^{-1} K_{uf} + D + \sigma_y^2 I)$$

$$= \mathcal{N}(\mathbf{y}; \mathbf{0}, K_{fu} K_{uu}^{-1} K_{uf} + D + \sigma_y^2 I)$$

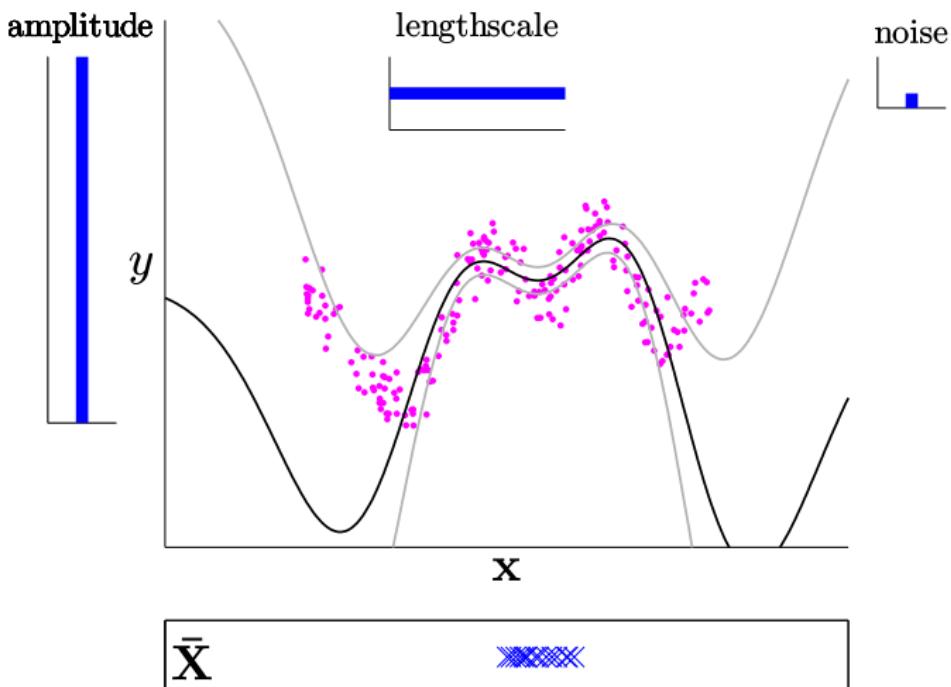
original variances along diagonal: stops variances collapsing



construct new generative model (with pseudo-data)
cheaper to perform exact learning and inference
calibrated to original

indirect
posterior
approximation

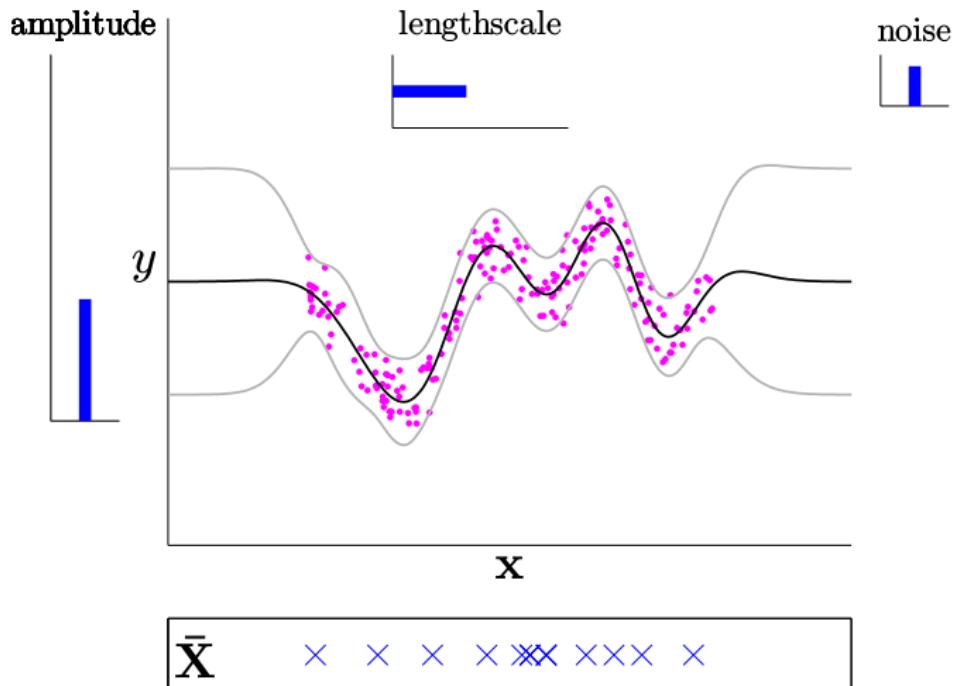
FITC: Demo (Snelson)



Initialize adversarially:

amplitude and lengthscale too big
noise too small
pseudo-inputs bunched up

FITC: Demo (Snelson)



Pseudo-inputs and hyperparameters optimized

Fully independent training conditional (FITC) approximation

- parametric (although cleverly so)
- if I see more data, should I add extra pseudo-data?
 - ▶ unnatural from a generative modelling perspective
 - ▶ natural from a prediction perspective (posterior gets more complex)
- ⇒ lost elegant separation of model, inference and approximation
- example of prior approximation

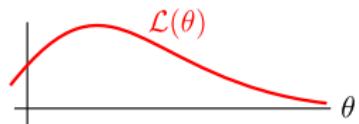
Extensions:

- methods for optimising pseudo-inputs (indirect approximations tend to over-fit)
- partially independent training conditional and tree-structured approximations (see extra slides)

Variational free-energy method (VFE)

lower bound the likelihood

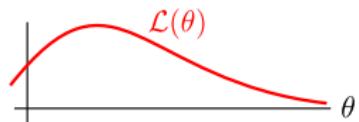
$$\mathcal{L}(\theta) = \log p(\mathbf{y}|\theta) = \log \int df \, p(\mathbf{y}, f|\theta)$$



Variational free-energy method (VFE)

lower bound the likelihood

$$\begin{aligned}\mathcal{L}(\theta) &= \log p(\mathbf{y}|\theta) = \log \int df \, p(\mathbf{y}, f|\theta) \\ &= \log \int df \, p(\mathbf{y}, f|\theta) \frac{q(f)}{q(f)}\end{aligned}$$

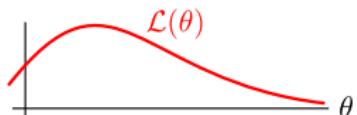


Variational free-energy method (VFE)

lower bound the likelihood

$$\mathcal{L}(\theta) = \log p(\mathbf{y}|\theta) = \log \int df \, p(\mathbf{y}, f|\theta)$$

$$= \log \int df \, p(\mathbf{y}, f|\theta) \frac{q(f)}{q(f)} \geq \int df \, q(f) \log \frac{p(\mathbf{y}, f|\theta)}{q(f)}$$

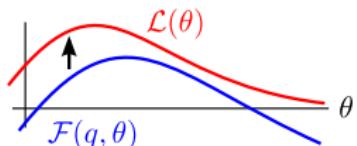


Variational free-energy method (VFE)

lower bound the likelihood

$$\mathcal{L}(\theta) = \log p(\mathbf{y}|\theta) = \log \int df \, p(\mathbf{y}, f|\theta)$$

$$= \log \int df \, p(\mathbf{y}, f|\theta) \frac{q(f)}{q(f)} \geq \int df \, q(f) \log \frac{p(\mathbf{y}, f|\theta)}{q(f)} = \mathcal{F}(q, \theta)$$



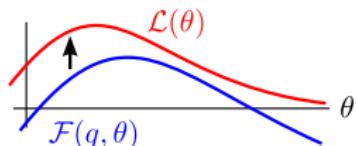
Variational free-energy method (VFE)

lower bound the likelihood

$$\mathcal{L}(\theta) = \log p(\mathbf{y}|\theta) = \log \int df p(\mathbf{y}, f|\theta)$$

$$= \log \int df p(\mathbf{y}, f|\theta) \frac{q(f)}{q(f)} \geq \int df q(f) \log \frac{p(\mathbf{y}, f|\theta)}{q(f)} = \mathcal{F}(\theta)$$

$$\mathcal{F}(\theta) = \int df q(f) \log \frac{p(f|\mathbf{y}, \theta)p(\mathbf{y}|\theta)}{q(f)}$$



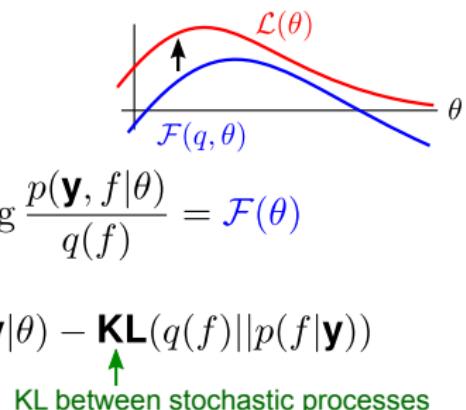
Variational free-energy method (VFE)

lower bound the likelihood

$$\mathcal{L}(\theta) = \log p(\mathbf{y}|\theta) = \log \int df p(\mathbf{y}, f|\theta)$$

$$= \log \int df p(\mathbf{y}, f|\theta) \frac{q(f)}{q(f)} \geq \int df q(f) \log \frac{p(\mathbf{y}, f|\theta)}{q(f)} = \mathcal{F}(\theta)$$

$$\mathcal{F}(\theta) = \int df q(f) \log \frac{p(f|\mathbf{y}, \theta)p(\mathbf{y}|\theta)}{q(f)} = \log p(\mathbf{y}|\theta) - \mathbf{KL}(q(f)||p(f|\mathbf{y}))$$



Variational free-energy method (VFE)

lower bound the likelihood

$$\mathcal{L}(\theta) = \log p(\mathbf{y}|\theta) = \log \int df p(\mathbf{y}, f|\theta)$$

$$= \log \int df p(\mathbf{y}, f|\theta) \frac{q(f)}{q(f)} \geq \int df q(f) \log \frac{p(\mathbf{y}, f|\theta)}{q(f)} = \mathcal{F}(\theta)$$

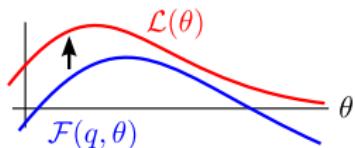
$$\mathcal{F}(\theta) = \int df q(f) \log \frac{p(f|\mathbf{y}, \theta)p(\mathbf{y}|\theta)}{q(f)} = \log p(\mathbf{y}|\theta) - \mathbf{KL}(q(f)||p(f|\mathbf{y}))$$

KL between stochastic processes

assume approximate posterior factorisation with special form

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

exact: $q(f_{\neq \mathbf{u}}|\mathbf{u}) = p(f_{\neq \mathbf{u}}|\mathbf{y}, \mathbf{u})$



Variational free-energy method (VFE)

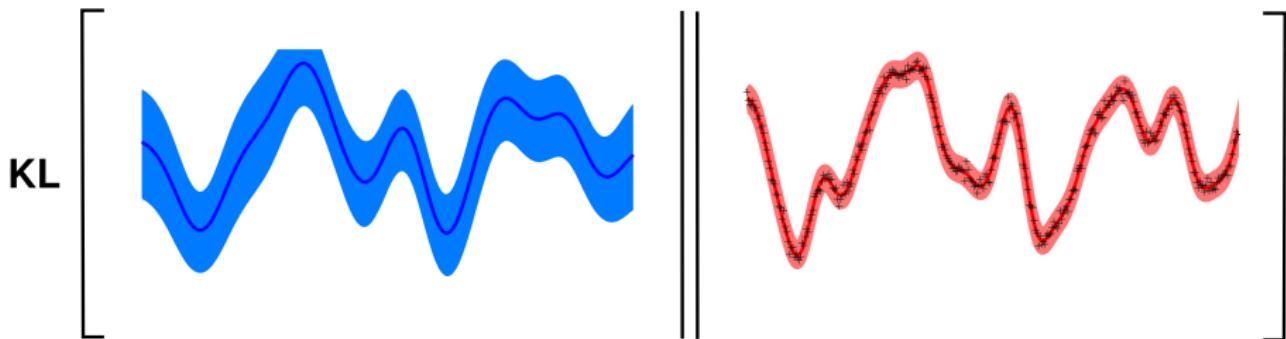
$$\mathcal{F}(\theta) = \log p(\mathbf{y}|\theta) - \mathbf{KL}(q(f)||p(f|\mathbf{y}))$$

approximate posterior

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

true posterior

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



Variational free-energy method (VFE)

$$\mathcal{F}(\theta) = \log p(\mathbf{y}|\theta) - \mathbf{KL}(q(f)||p(f|\mathbf{y}))$$

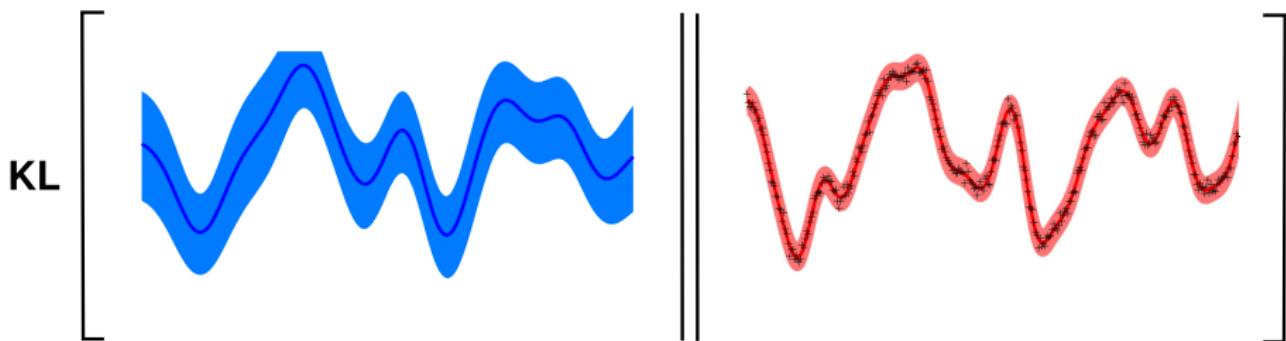
approximate posterior

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

same form as prediction
from GP-regression

true posterior

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



Variational free-energy method (VFE)

$$\mathcal{F}(\theta) = \log p(\mathbf{y}|\theta) - \mathbf{KL}(q(f)||p(f|\mathbf{y}))$$

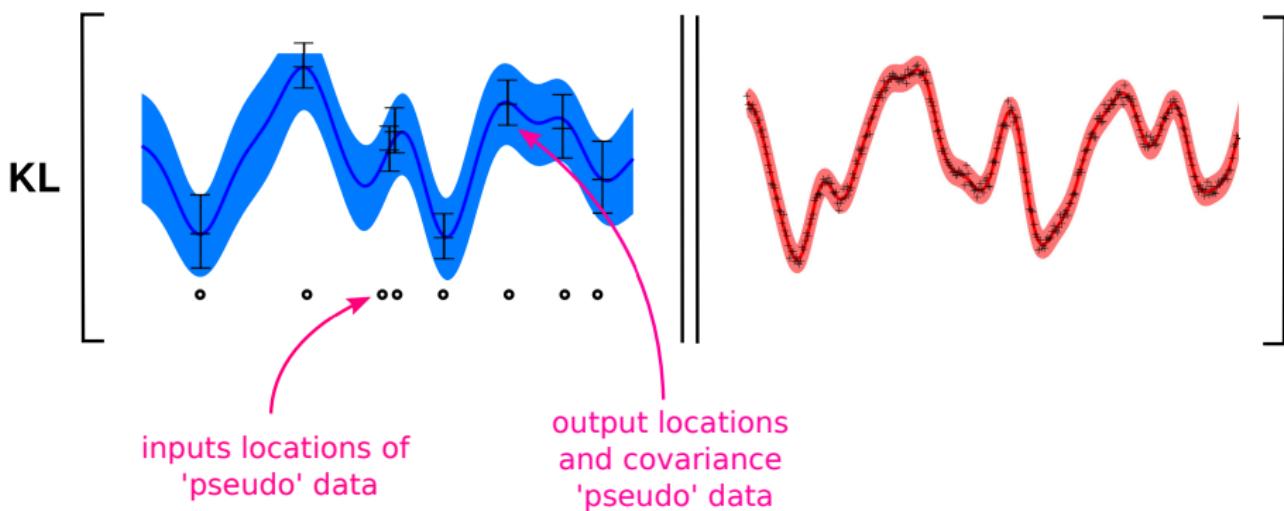
approximate posterior

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

same form as prediction
from GP-regression

true posterior

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



optimise variational free-energy wrt to these variational parameters

Variational free-energy method (VFE)

lower bound the likelihood

$$\mathcal{L}(\theta) = \log p(\mathbf{y}|\theta) = \log \int df p(\mathbf{y}, f|\theta)$$

$$= \log \int df p(\mathbf{y}, f|\theta) \frac{q(f)}{q(f)} \geq \int df q(f) \log \frac{p(\mathbf{y}, f|\theta)}{q(f)} = \mathcal{F}(\theta)$$

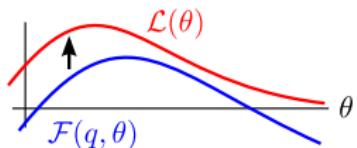
$$\mathcal{F}(\theta) = \int df q(f) \log \frac{p(f|\mathbf{y}, \theta)p(\mathbf{y}|\theta)}{q(f)} = \log p(\mathbf{y}|\theta) - \text{KL}(q(f)||p(f|\mathbf{y}))$$

KL between stochastic processes

assume approximate posterior factorisation with special form

$$q(f) = q(\mathbf{u}, f_{\neq \mathbf{u}}) = q(f_{\neq \mathbf{u}}|\mathbf{u})q(\mathbf{u}) = p(f_{\neq \mathbf{u}}|\mathbf{u})q(\mathbf{u}) \quad \leftarrow \text{predictive from GP regression}$$

exact: $q(f_{\neq \mathbf{u}}|\mathbf{u}) = p(f_{\neq \mathbf{u}}|\mathbf{y}, \mathbf{u})$



Variational free-energy method (VFE)

lower bound the likelihood

$$\mathcal{L}(\theta) = \log p(\mathbf{y}|\theta) = \log \int df p(\mathbf{y}, f|\theta)$$

$$= \log \int df p(\mathbf{y}, f|\theta) \frac{q(f)}{q(f)} \geq \int df q(f) \log \frac{p(\mathbf{y}, f|\theta)}{q(f)} = \mathcal{F}(\theta)$$

$$\mathcal{F}(\theta) = \int df q(f) \log \frac{p(f|\mathbf{y}, \theta)p(\mathbf{y}|\theta)}{q(f)} = \log p(\mathbf{y}|\theta) - \text{KL}(q(f)||p(f|\mathbf{y}))$$

KL between stochastic processes

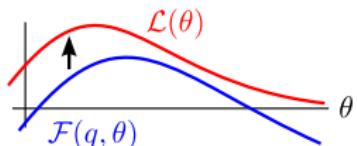
assume approximate posterior factorisation with special form

$$q(f) = q(\mathbf{u}, f_{\neq \mathbf{u}}) = q(f_{\neq \mathbf{u}}|\mathbf{u})q(\mathbf{u}) = p(f_{\neq \mathbf{u}}|\mathbf{u})q(\mathbf{u}) \quad \leftarrow \text{predictive from GP regression}$$

exact: $q(f_{\neq \mathbf{u}}|\mathbf{u}) = p(f_{\neq \mathbf{u}}|\mathbf{y}, \mathbf{u})$

plug into Free-energy:

$$\mathcal{F}(\theta) = \int df q(f) \log \frac{p(\mathbf{y}, f|\theta)}{p(f_{\neq \mathbf{u}}|\mathbf{u})q(\mathbf{u})}$$



Variational free-energy method (VFE)

lower bound the likelihood

$$\mathcal{L}(\theta) = \log p(\mathbf{y}|\theta) = \log \int df p(\mathbf{y}, f|\theta)$$

$$= \log \int df p(\mathbf{y}, f|\theta) \frac{q(f)}{q(f)} \geq \int df q(f) \log \frac{p(\mathbf{y}, f|\theta)}{q(f)} = \mathcal{F}(\theta)$$

$$\mathcal{F}(\theta) = \int df q(f) \log \frac{p(f|\mathbf{y}, \theta)p(\mathbf{y}|\theta)}{q(f)} = \log p(\mathbf{y}|\theta) - \text{KL}(q(f)||p(f|\mathbf{y}))$$

KL between stochastic processes

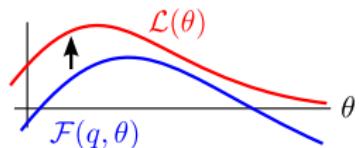
assume approximate posterior factorisation with special form

$$q(f) = q(\mathbf{u}, f_{\neq \mathbf{u}}) = q(f_{\neq \mathbf{u}}|\mathbf{u})q(\mathbf{u}) = p(f_{\neq \mathbf{u}}|\mathbf{u})q(\mathbf{u}) \quad \leftarrow \text{predictive from GP regression}$$

exact: $q(f_{\neq \mathbf{u}}|\mathbf{u}) = p(f_{\neq \mathbf{u}}|\mathbf{y}, \mathbf{u})$

plug into Free-energy:

$$\mathcal{F}(\theta) = \int df q(f) \log \frac{p(\mathbf{y}, f|\theta)}{p(f_{\neq \mathbf{u}}|\mathbf{u})q(\mathbf{u})} = \int df q(f) \log \frac{p(\mathbf{y}|\mathbf{f}, \theta)p(f_{\neq \mathbf{u}}|\mathbf{u})p(\mathbf{u})}{p(f_{\neq \mathbf{u}}|\mathbf{u})q(\mathbf{u})}$$



Variational free-energy method (VFE)

lower bound the likelihood

$$\mathcal{L}(\theta) = \log p(\mathbf{y}|\theta) = \log \int df p(\mathbf{y}, f|\theta)$$

$$= \log \int df p(\mathbf{y}, f|\theta) \frac{q(f)}{q(f)} \geq \int df q(f) \log \frac{p(\mathbf{y}, f|\theta)}{q(f)} = \mathcal{F}(\theta)$$

$$\mathcal{F}(\theta) = \int df q(f) \log \frac{p(f|\mathbf{y}, \theta)p(\mathbf{y}|\theta)}{q(f)} = \log p(\mathbf{y}|\theta) - \text{KL}(q(f)||p(f|\mathbf{y}))$$

KL between stochastic processes

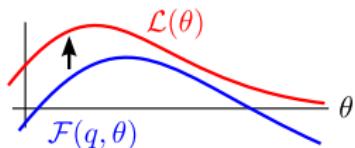
assume approximate posterior factorisation with special form

$$q(f) = q(\mathbf{u}, f_{\neq \mathbf{u}}) = q(f_{\neq \mathbf{u}}|\mathbf{u})q(\mathbf{u}) = p(f_{\neq \mathbf{u}}|\mathbf{u})q(\mathbf{u}) \quad \leftarrow \text{predictive from GP regression}$$

exact: $q(f_{\neq \mathbf{u}}|\mathbf{u}) = p(f_{\neq \mathbf{u}}|\mathbf{y}, \mathbf{u})$

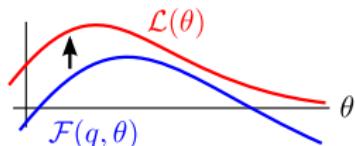
plug into Free-energy:

$$\mathcal{F}(\theta) = \int df q(f) \log \frac{p(\mathbf{y}, f|\theta)}{p(f_{\neq \mathbf{u}}|\mathbf{u})q(\mathbf{u})} = \int df q(f) \log \frac{p(\mathbf{y}|\mathbf{f}, \theta)p(f_{\neq \mathbf{u}}|\mathbf{u})p(\mathbf{u})}{p(f_{\neq \mathbf{u}}|\mathbf{u})q(\mathbf{u})}$$



Variational free-energy method (VFE)

lower bound the likelihood

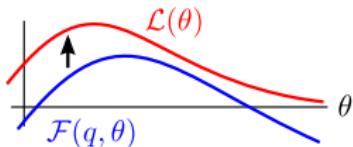


$$\mathcal{F}(\theta) = \int df q(f) \log \frac{p(\mathbf{y}, f | \theta)}{p(f_{\neq \mathbf{u}} | \mathbf{u}) q(\mathbf{u})} = \int df q(f) \log \frac{p(\mathbf{y} | \mathbf{f}, \theta) p(f_{\neq \mathbf{u}} | \mathbf{u}) p(\mathbf{u})}{p(f_{\neq \mathbf{u}} | \mathbf{u}) q(\mathbf{u})}$$

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

Variational free-energy method (VFE)

lower bound the likelihood



$$\mathcal{F}(\theta) = \int df q(f) \log \frac{p(\mathbf{y}, f | \theta)}{p(f_{\neq \mathbf{u}} | \mathbf{u}) q(\mathbf{u})} = \int df q(f) \log \frac{p(\mathbf{y} | \mathbf{f}, \theta) p(f_{\neq \mathbf{u}} | \mathbf{u}) p(\mathbf{u})}{p(f_{\neq \mathbf{u}} | \mathbf{u}) q(\mathbf{u})}$$

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

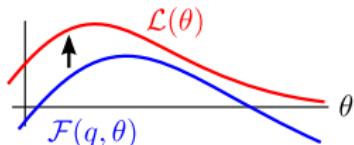
$$\mathcal{F}(\theta) = \langle \log p(\mathbf{y} | \mathbf{f}, \theta) \rangle_{q(f)} - \text{KL}(q(\mathbf{u}) || p(\mathbf{u}))$$

↑
average of
quadratic form

↑
KL between two
multivariate Gaussians

Variational free-energy method (VFE)

lower bound the likelihood



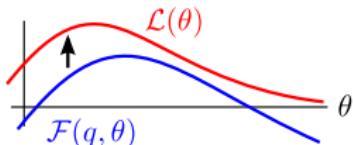
$$\mathcal{F}(\theta) = \int df \ q(f) \log \frac{p(\mathbf{y}, f | \theta)}{p(f_{\neq \mathbf{u}} | \mathbf{u})q(\mathbf{u})} = \int df \ q(f) \log \frac{p(\mathbf{y} | \mathbf{f}, \theta)p(f_{\neq \mathbf{u}} | \mathbf{u})p(\mathbf{u})}{p(f_{\neq \mathbf{u}} | \mathbf{u})q(\mathbf{u})}$$

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

make bound as tight as possible: $q^*(\mathbf{u}) = \arg \max_{q(\mathbf{u})} \mathcal{F}(q, \theta)$

Variational free-energy method (VFE)

lower bound the likelihood



$$\mathcal{F}(\theta) = \int df q(f) \log \frac{p(\mathbf{y}, f | \theta)}{p(f_{\neq \mathbf{u}} | \mathbf{u}) q(\mathbf{u})} = \int df q(f) \log \frac{p(\mathbf{y} | \mathbf{f}, \theta) p(f_{\neq \mathbf{u}} | \mathbf{u}) p(\mathbf{u})}{p(f_{\neq \mathbf{u}} | \mathbf{u}) q(\mathbf{u})}$$

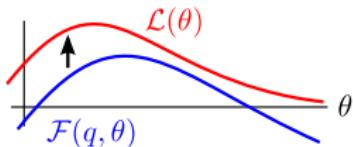
where $q(f) = q(\mathbf{u}, f_{\neq \mathbf{u}}) = q(f_{\neq \mathbf{u}} | \mathbf{u})q(\mathbf{u}) = p(f_{\neq \mathbf{u}} | \mathbf{u})q(\mathbf{u})$

make bound as tight as possible: $q^*(\mathbf{u}) = \arg \max_{q(\mathbf{u})} \mathcal{F}(q, \theta)$

$$q^*(\mathbf{u}) \propto p(\mathbf{u})\mathcal{N}(\mathbf{y}; \mathbf{K}_{\text{fu}}\mathbf{K}_{\text{uu}}^{-1}\mathbf{u}, \sigma_y^2\mathbf{I}) \quad (\text{DTC})$$

Variational free-energy method (VFE)

lower bound the likelihood



$$\mathcal{F}(\theta) = \int df q(f) \log \frac{p(\mathbf{y}, f | \theta)}{p(f_{\neq \mathbf{u}} | \mathbf{u}) q(\mathbf{u})} = \int df q(f) \log \frac{p(\mathbf{y} | \mathbf{f}, \theta) p(f_{\neq \mathbf{u}} | \mathbf{u}) p(\mathbf{u})}{p(f_{\neq \mathbf{u}} | \mathbf{u}) q(\mathbf{u})}$$

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

make bound as tight as possible: $q^*(\mathbf{u}) = \arg \max_{q(\mathbf{u})} \mathcal{F}(q, \theta)$

$$q^*(\mathbf{u}) \propto p(\mathbf{u})\mathcal{N}(\mathbf{y}; \mathbf{K}_{\text{fu}}\mathbf{K}_{\text{uu}}^{-1}\mathbf{u}, \sigma_y^2\mathbf{I}) \quad (\text{DTC})$$

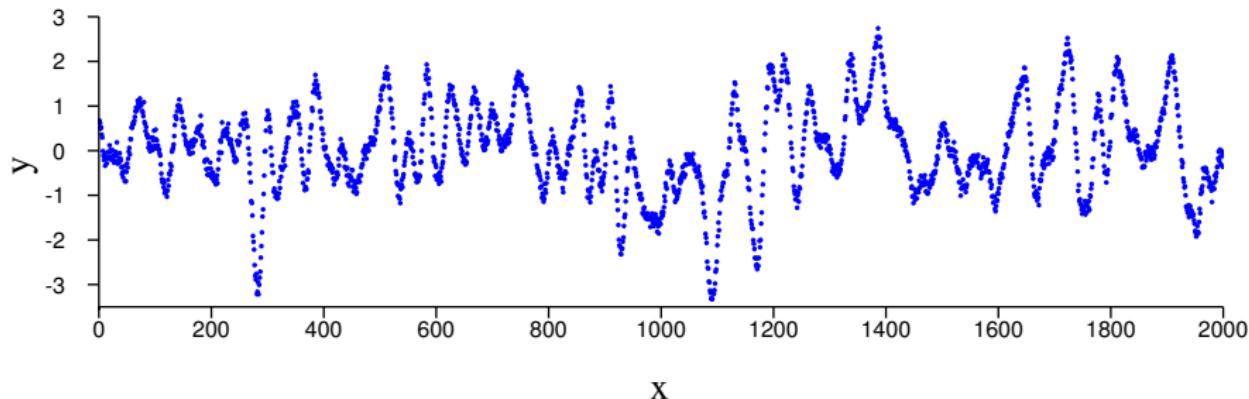
$$\mathcal{F}(q^*, \theta) = \log \mathcal{N}(\mathbf{y}; \mathbf{0}, K_{fu}K_{uu}^{-1}K_{uf}, \sigma_y^2 I) - \frac{1}{2\sigma_y^2} \text{trace}(K_{ff} - K_{fu}K_{uu}^{-1}K_{uf})$$

DTC like	uncertainty based correction
----------	------------------------------

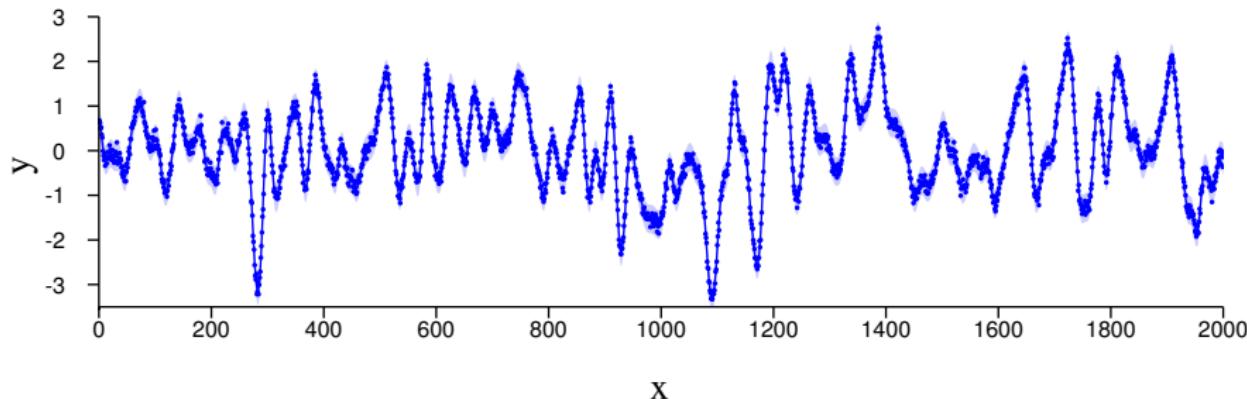
Summary of VFE method

- optimisation pseudo point inputs **better behaved** in VFE methods (direct posterior approximation)
- variational methods known to **underfit** (and have other **biases**)
- **no augmentation required: target is posterior over functions, which includes inducing variables**
 - ▶ pseudo-input locations are pure variational parameters (do not parameterise the generative model)
 - ▶ coherent way of adding pseudo-data: more complex posteriors require more computational resources (more pseudo-points)
- Curious observation:
VFE returns better mean estimates
FITC returns better error-bar estimates
- **how should we select $M = \text{number of pseudo-points?}$**

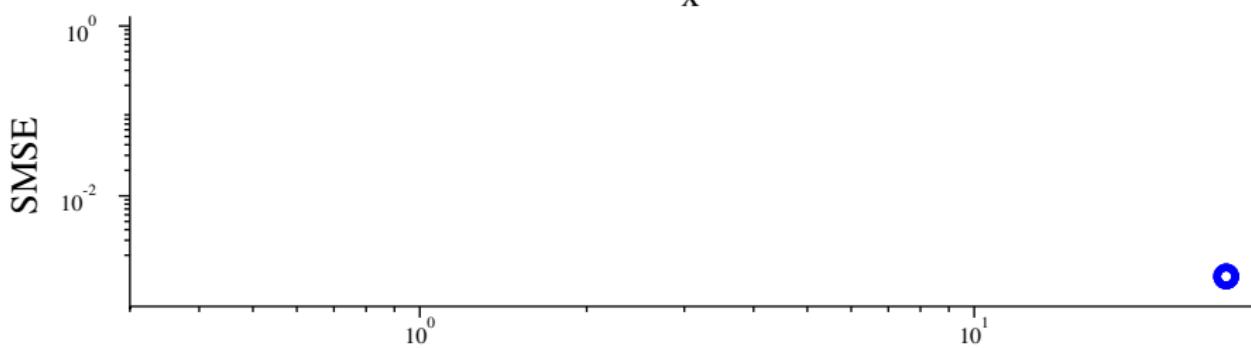
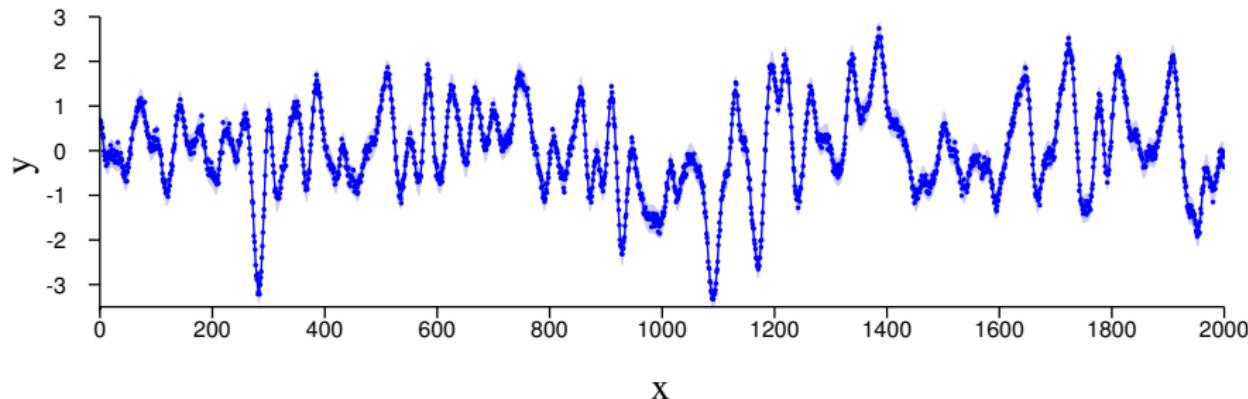
How do we select $M = \text{number of pseudo-data?}$



How do we select $M = \text{number of pseudo-data?}$

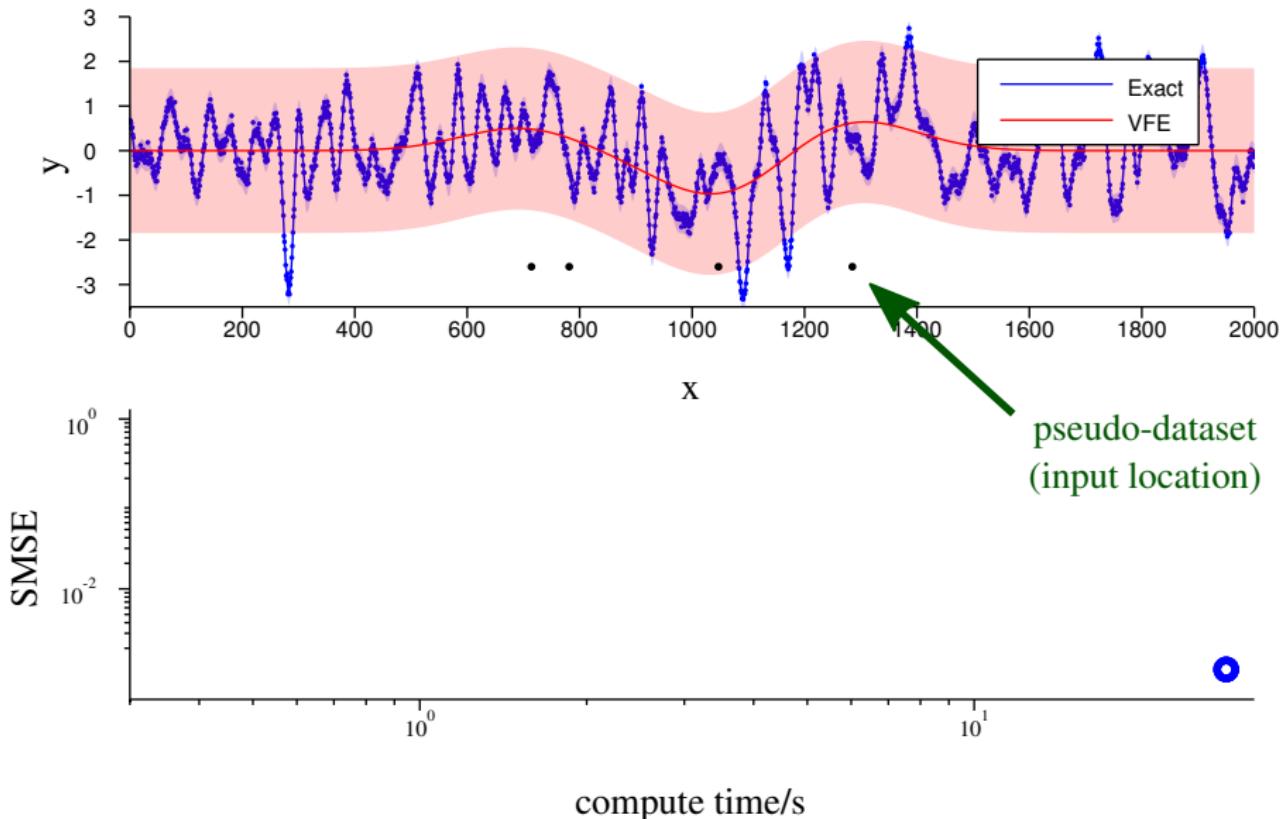


How do we select $M = \text{number of pseudo-data?}$

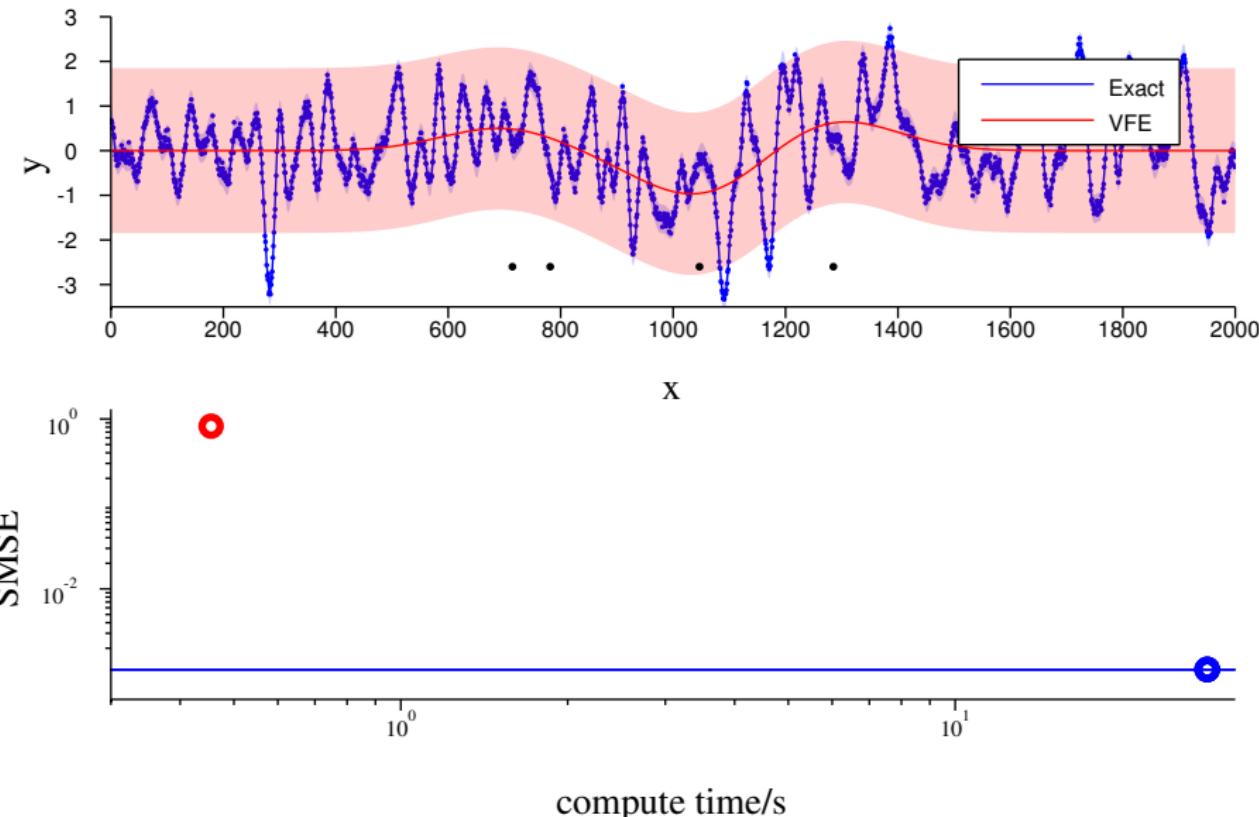


compute time/s

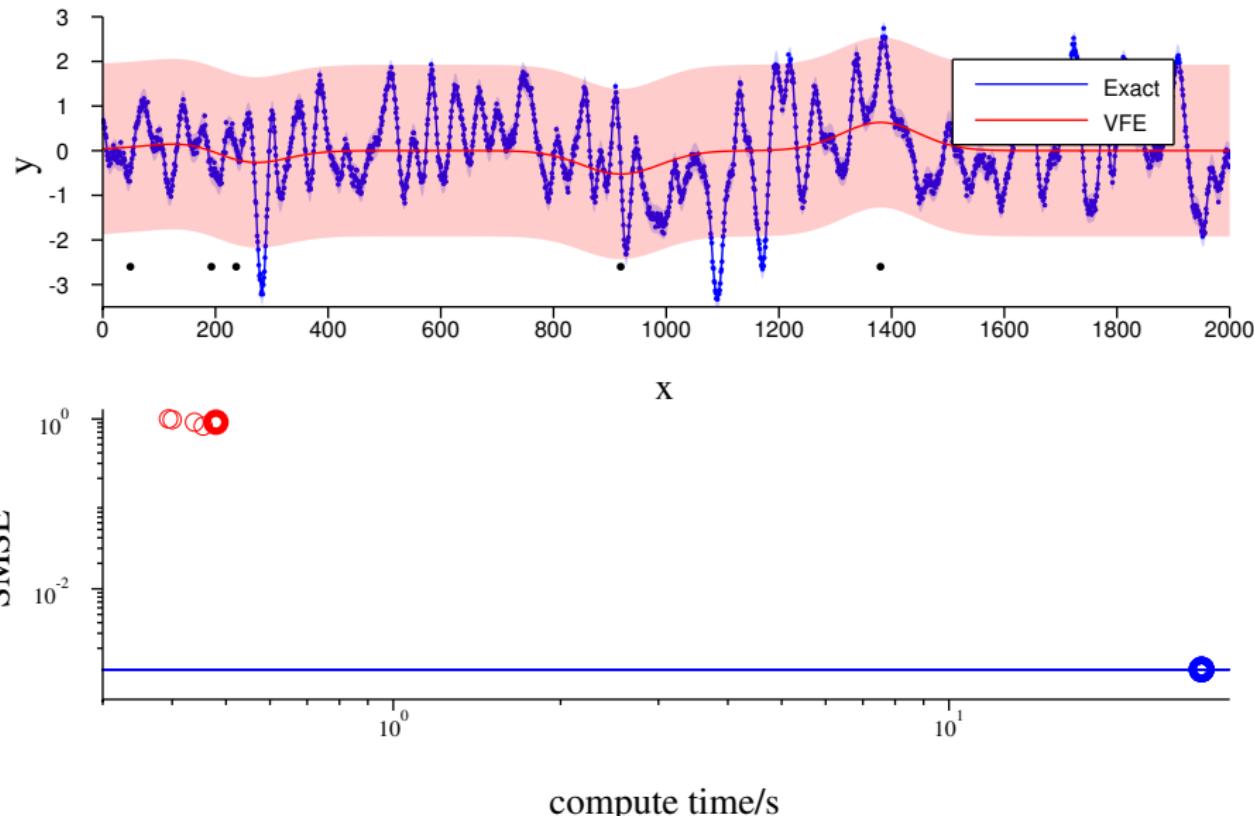
How do we select $M = \text{number of pseudo-data?}$



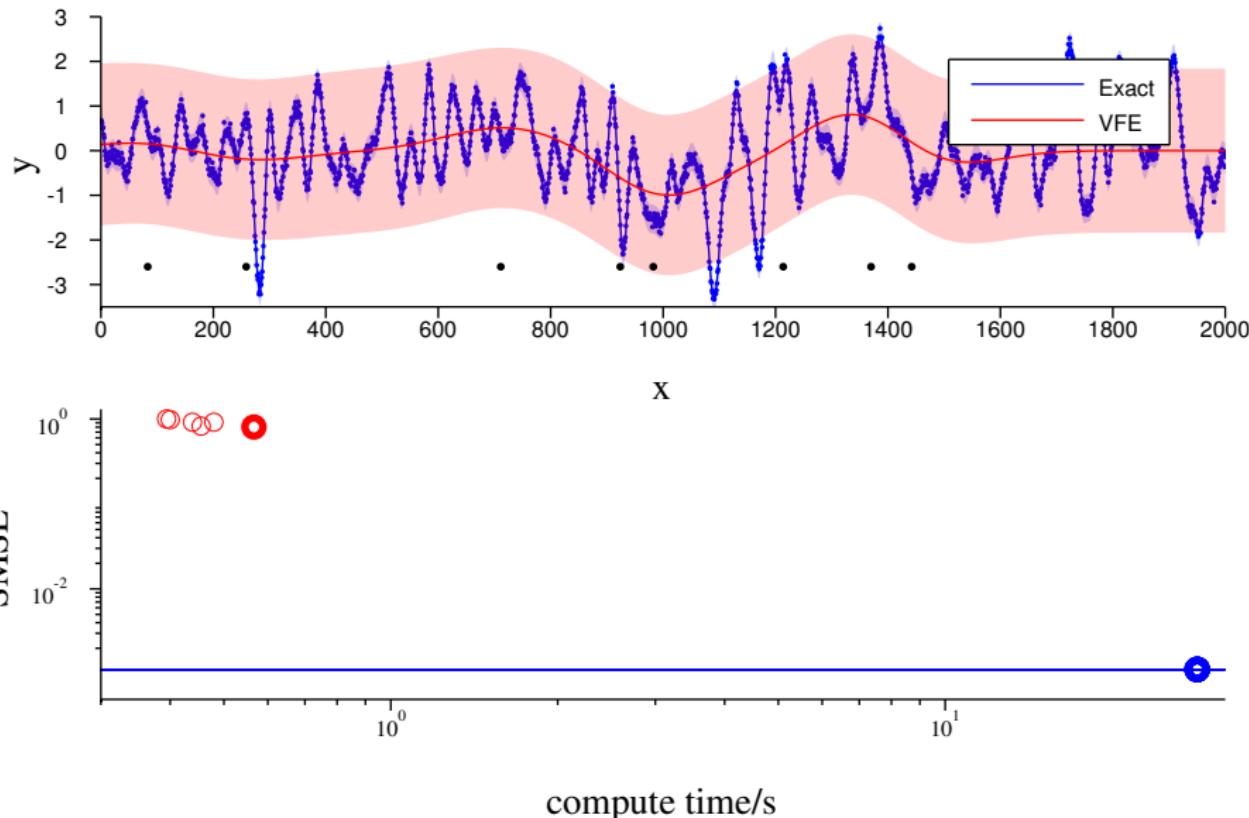
How do we select $M = \text{number of pseudo-data?}$



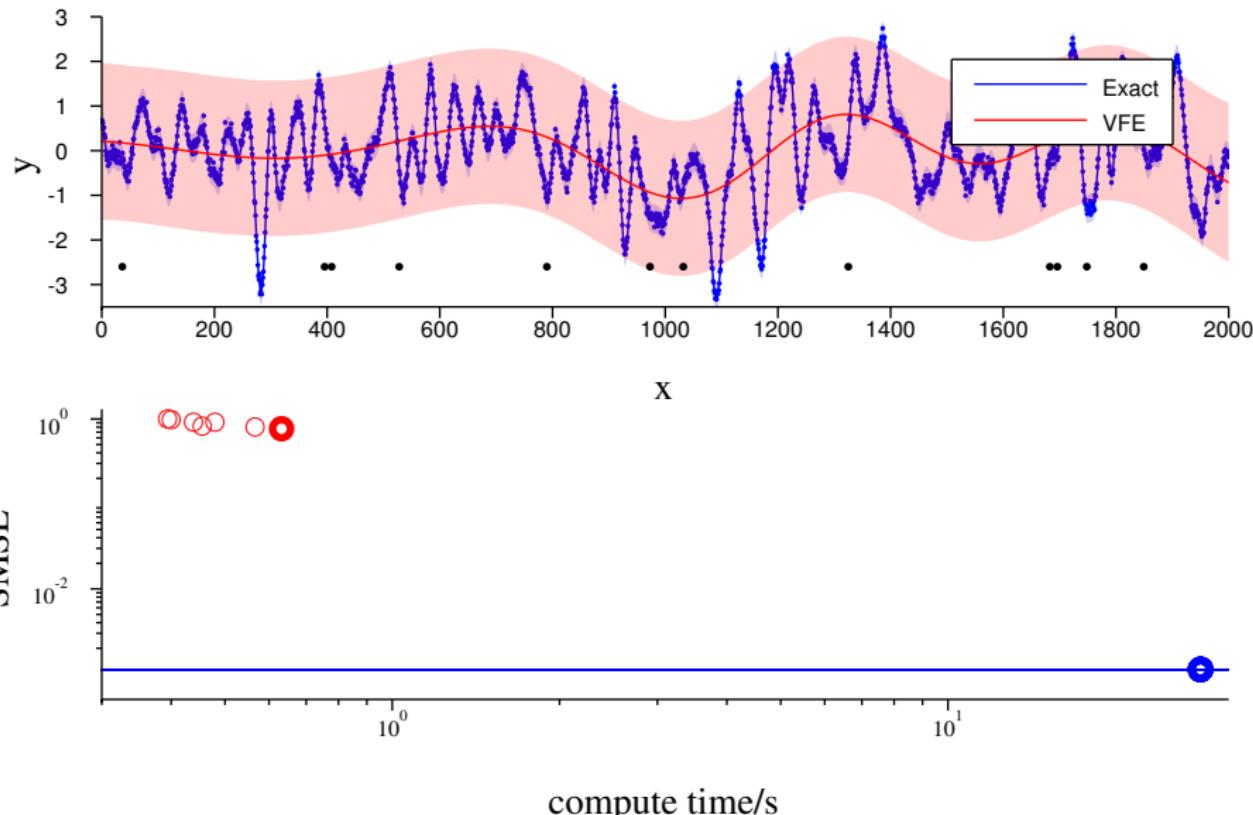
How do we select $M = \text{number of pseudo-data?}$



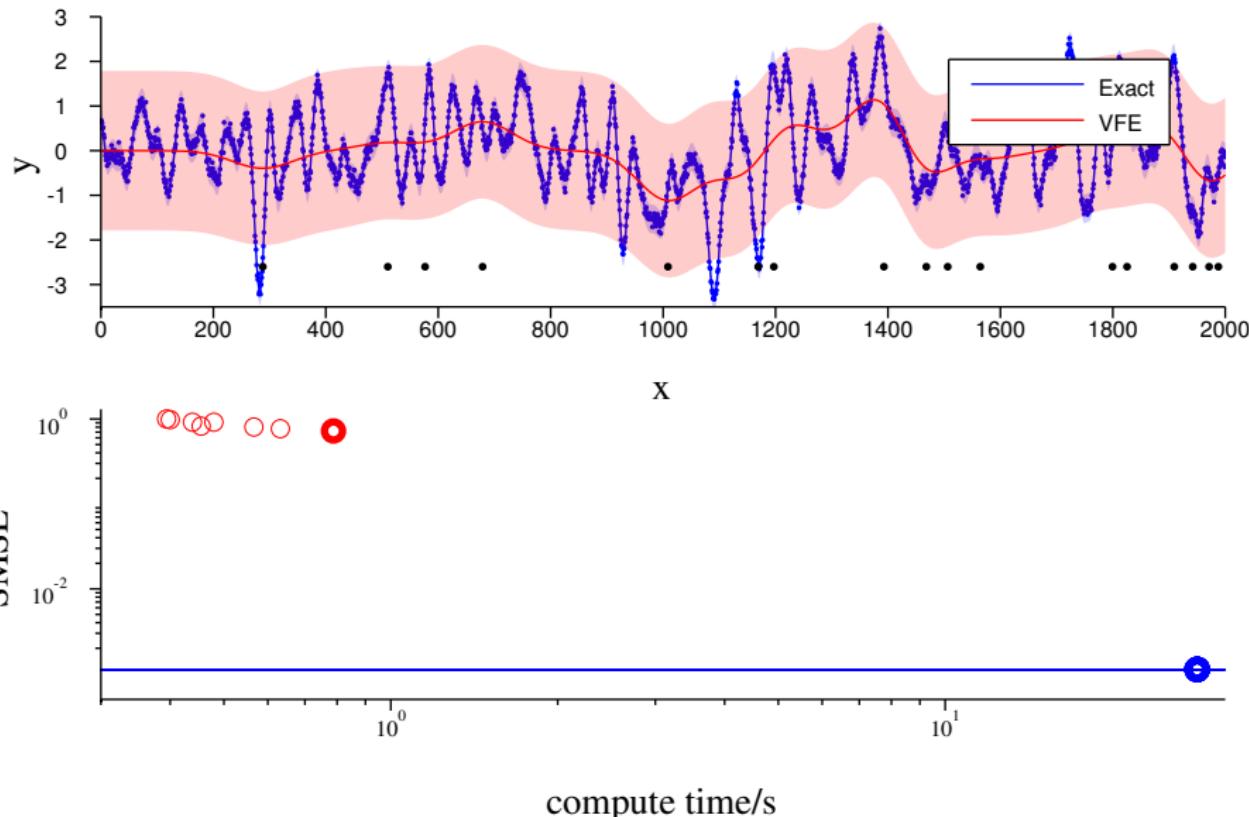
How do we select $M = \text{number of pseudo-data?}$



How do we select $M = \text{number of pseudo-data?}$

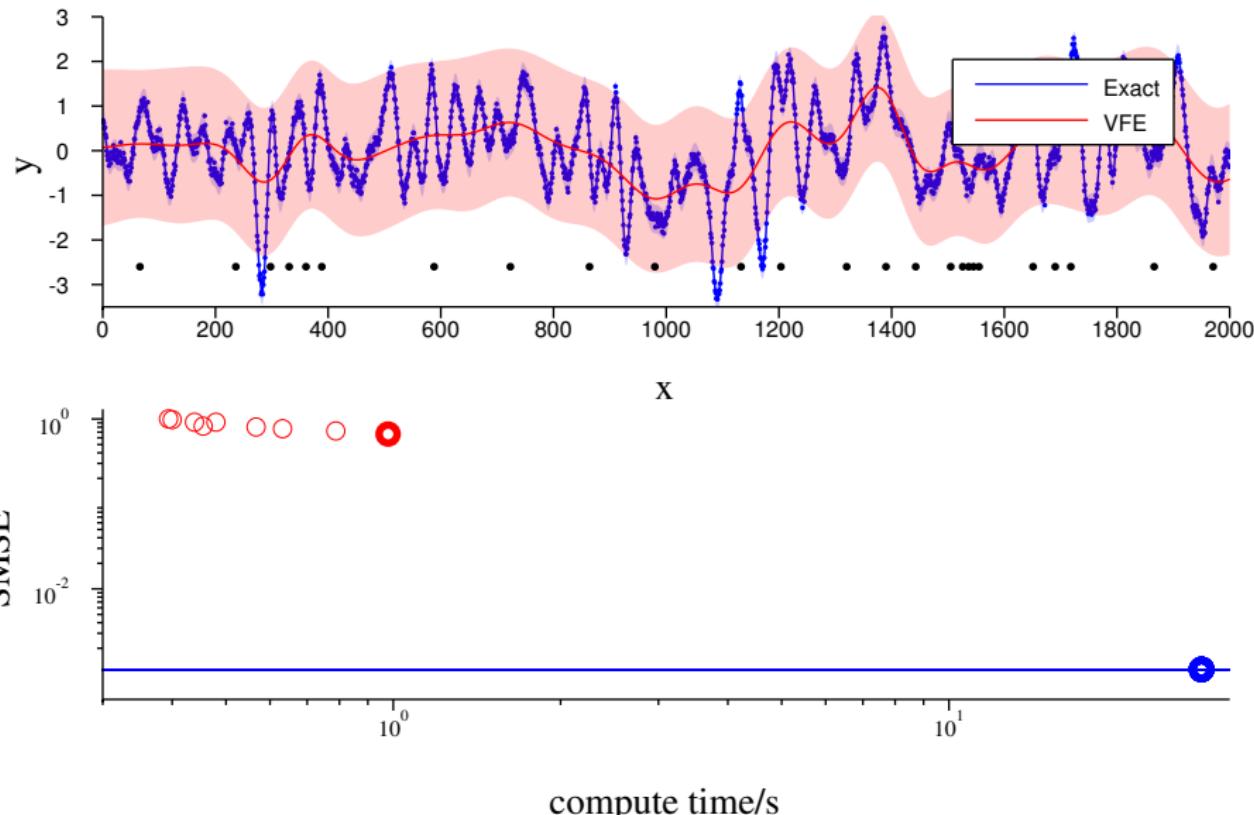


How do we select $M = \text{number of pseudo-data?}$

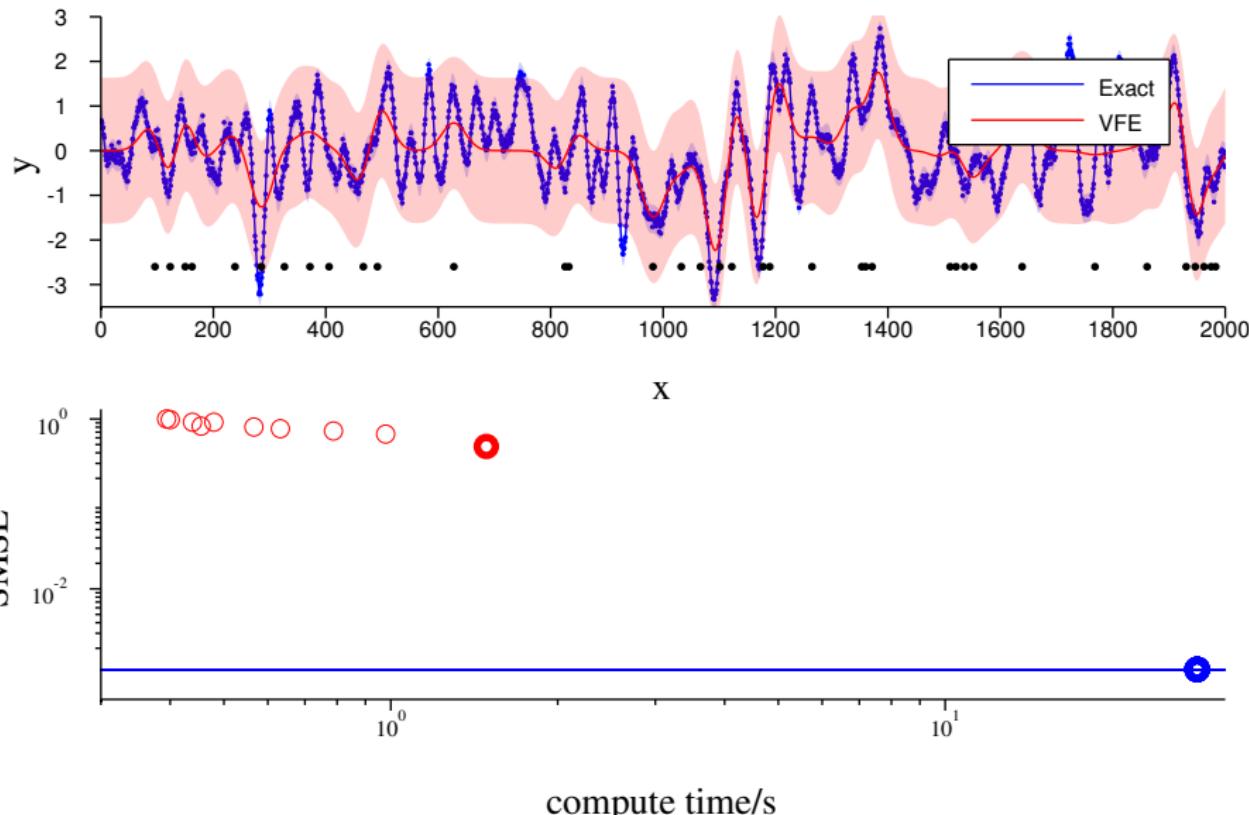


compute time/s

How do we select $M = \text{number of pseudo-data?}$

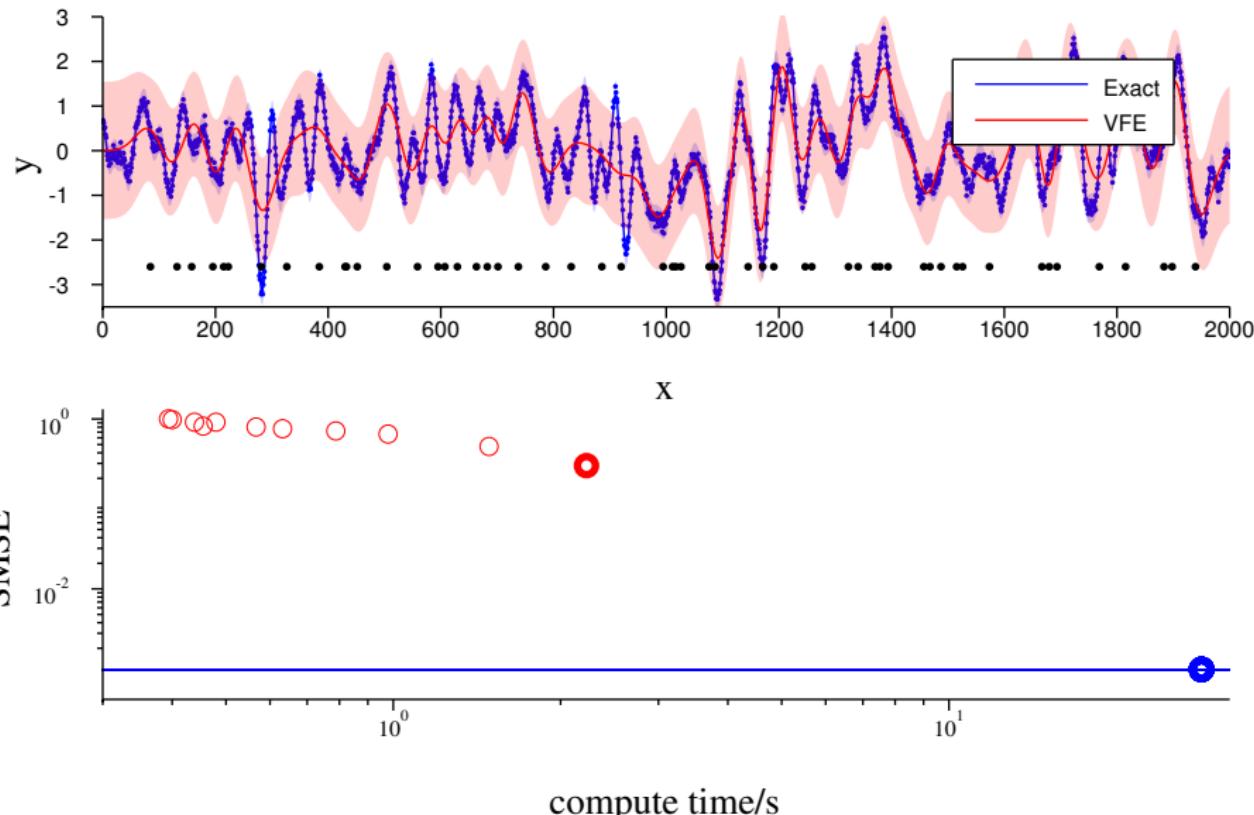


How do we select $M = \text{number of pseudo-data?}$

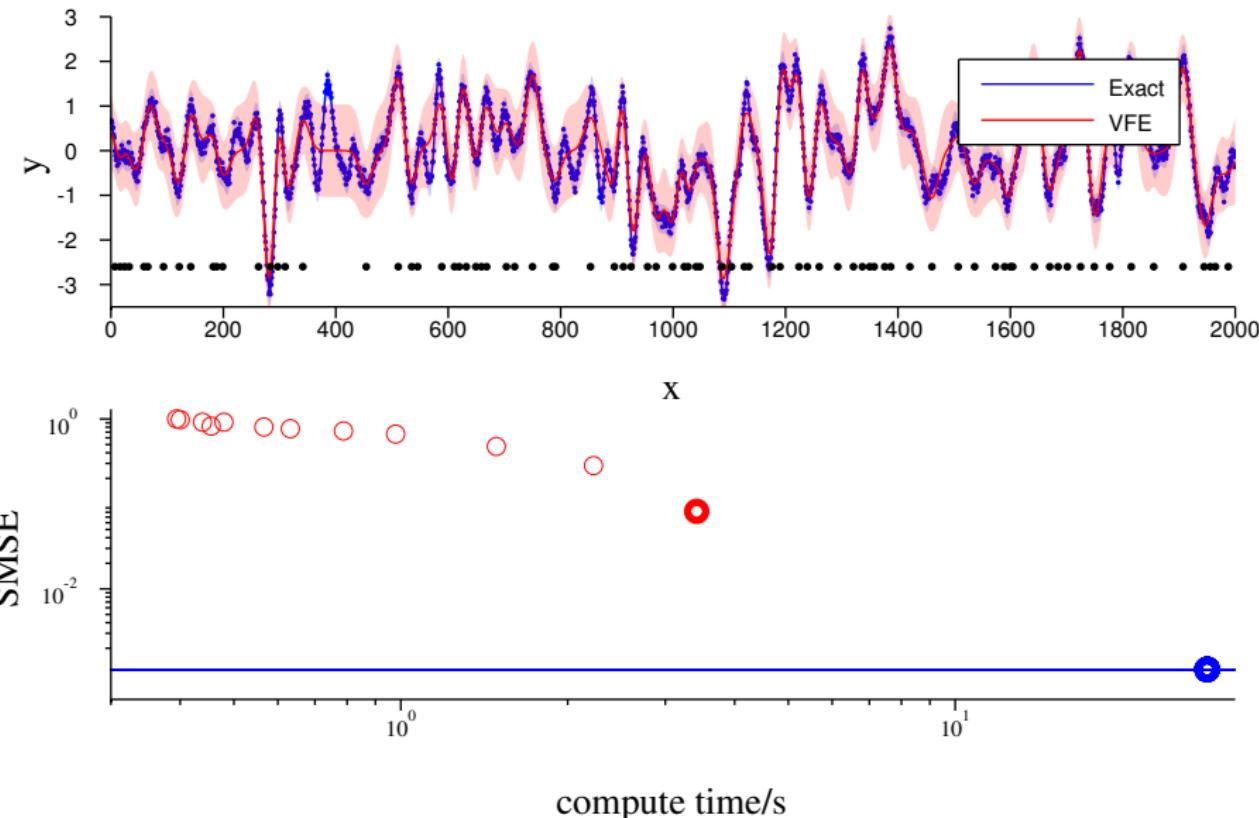


compute time/s

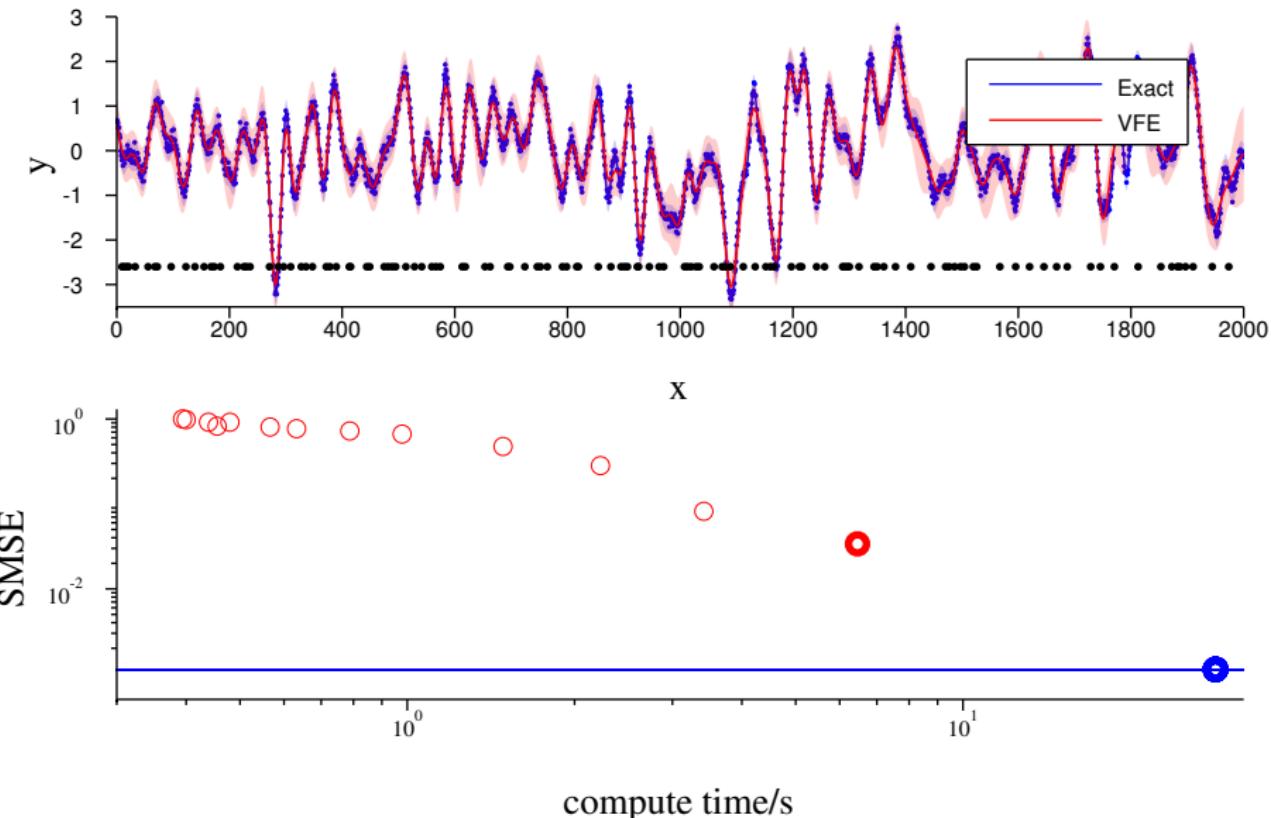
How do we select $M = \text{number of pseudo-data?}$



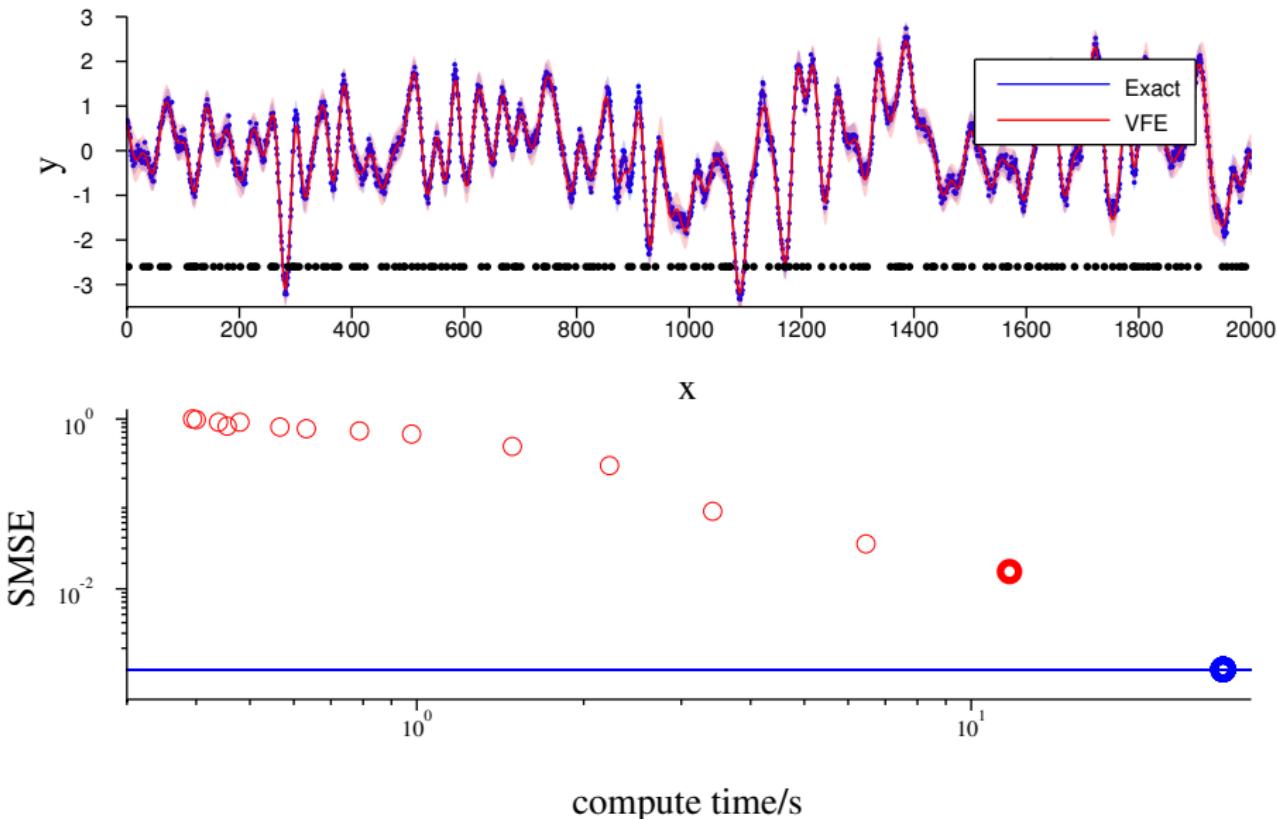
How do we select $M = \text{number of pseudo-data?}$



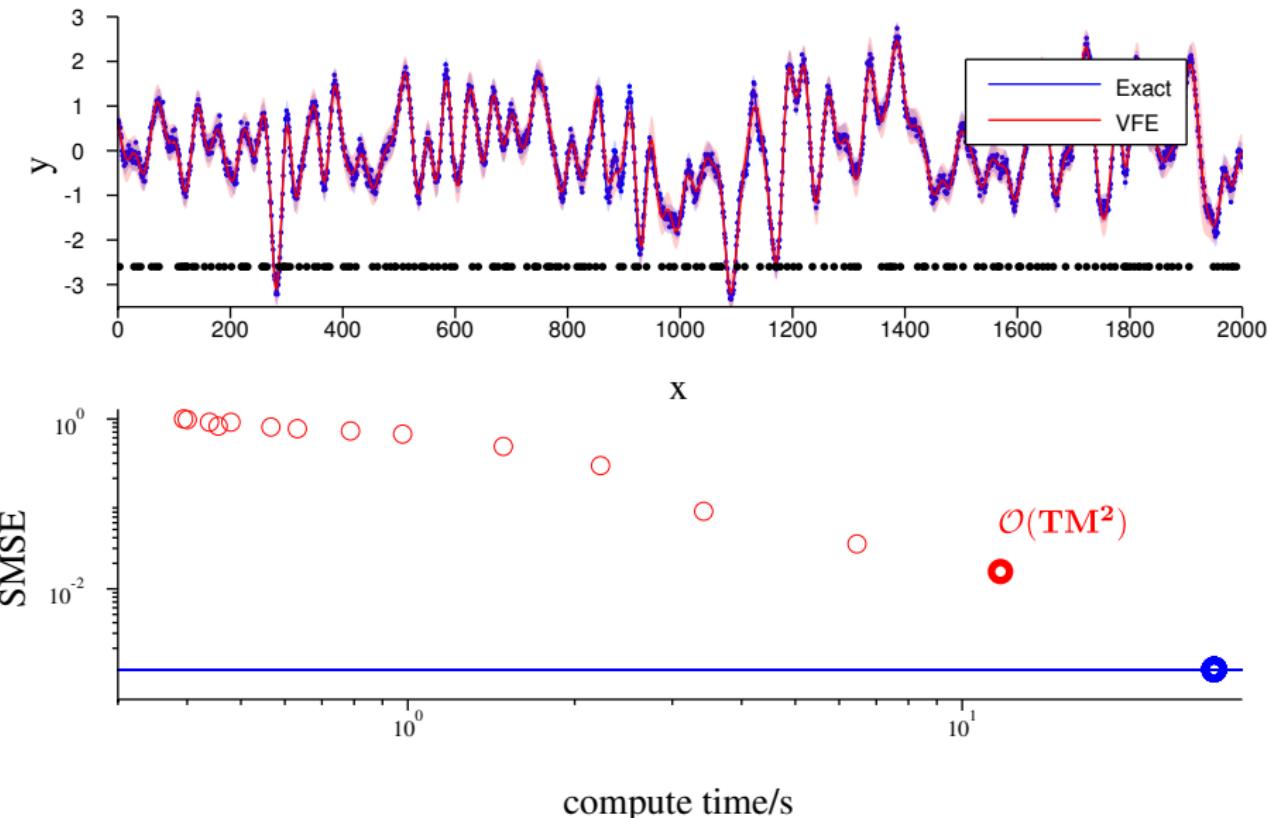
How do we select $M = \text{number of pseudo-data?}$



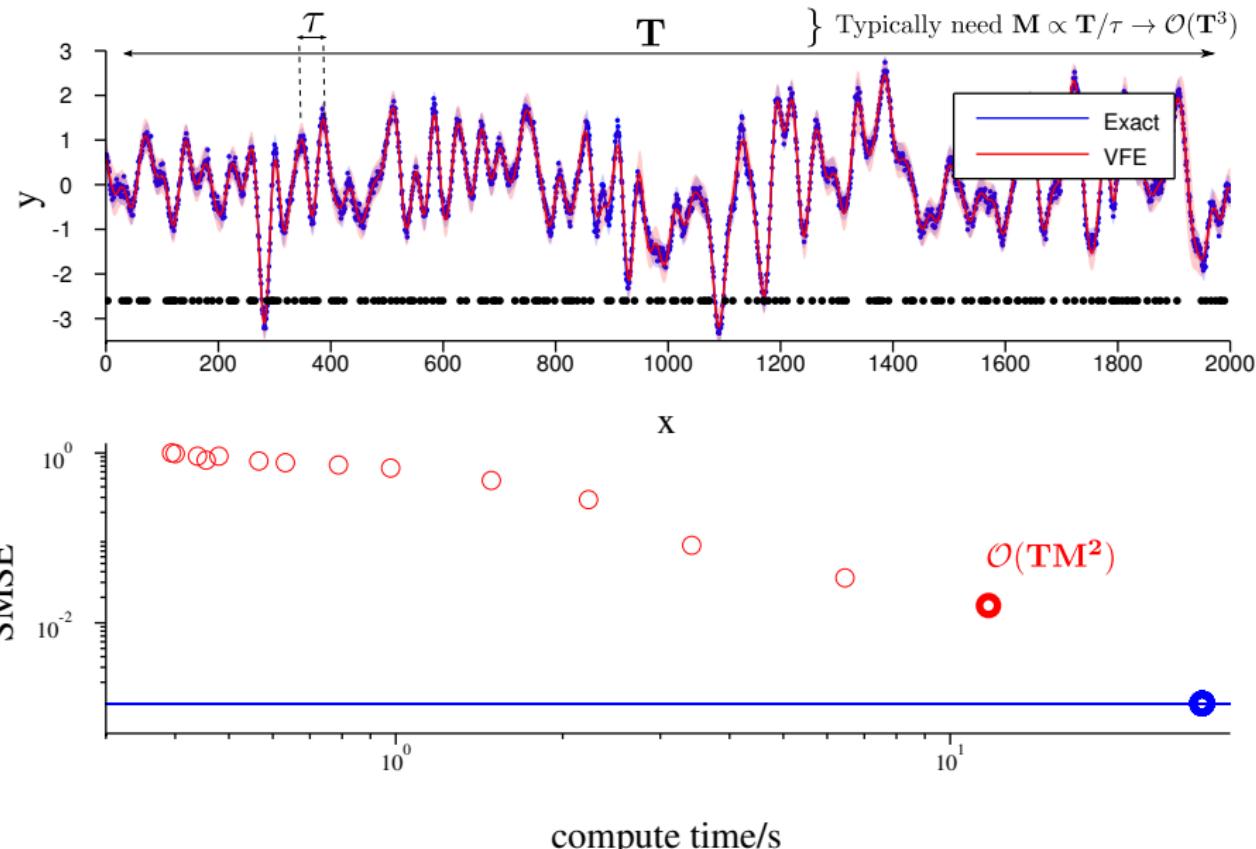
How do we select $M = \text{number of pseudo-data?}$



How do we select M = number of pseudo-data?



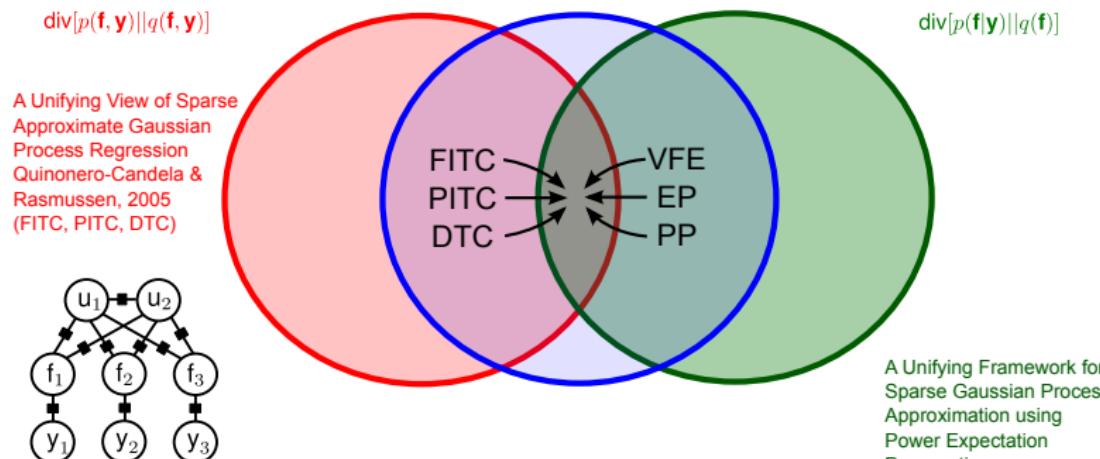
How do we select $M = \text{number of pseudo-data?}$



Power Expectation Propagation and Gaussian Processes

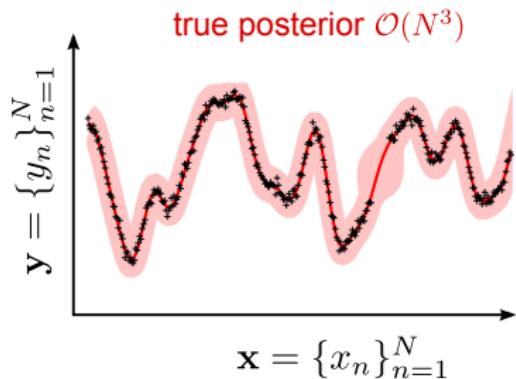
A Brief History of Gaussian Process Approximations

approximate generative model
exact inference methods employing
pseudo-data exact generative model
approximate inference



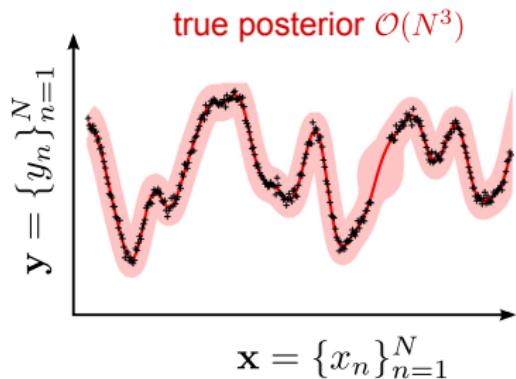
EP pseudo-point approximation

$$p^*(f) = p(f, \mathbf{y} | \mathbf{x}, \theta)$$



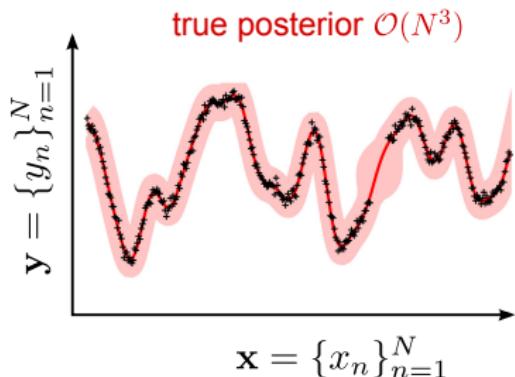
EP pseudo-point approximation

$$\begin{aligned} p^*(f) &= p(f, \mathbf{y} | \mathbf{x}, \theta) \\ &= p(f | \theta) \prod_{n=1}^N \underline{p(y_n | f, x_n, \theta)} \end{aligned}$$



EP pseudo-point approximation

$$\begin{aligned} p^*(f) &= p(f, \mathbf{y} | \mathbf{x}, \theta) \\ &= p(f | \theta) \prod_{n=1}^N \underline{p(y_n | f, x_n, \theta)} \\ &= \underbrace{p(\mathbf{y} | \mathbf{x}, \theta)}_{\text{marginal likelihood}} \underbrace{p(f | \mathbf{y}, \mathbf{x}, \theta)}_{\text{posterior}} \end{aligned}$$



EP pseudo-point approximation

$$p^*(f) = p(f, \mathbf{y} | \mathbf{x}, \theta)$$

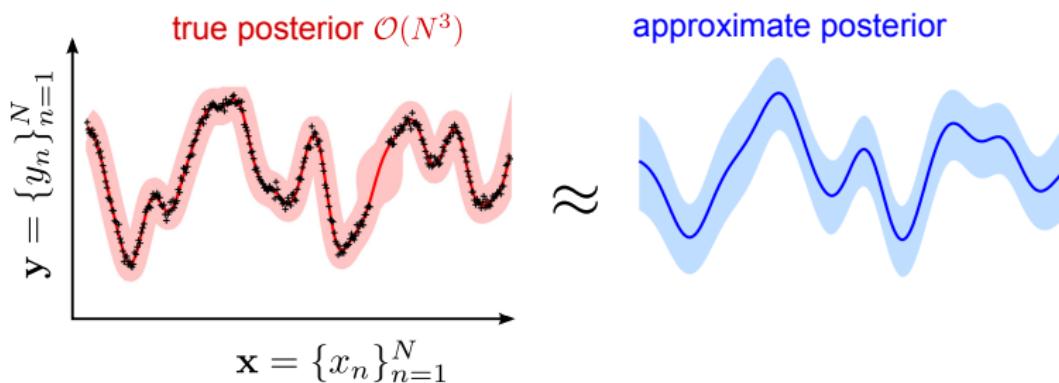
$$= p(f|\theta) \prod_{n=1}^N \underline{p(y_n|f, x_n, \theta)}$$

$$= \underline{p(\mathbf{y}|\mathbf{x}, \theta)} \underline{p(f|\mathbf{y}, \mathbf{x}, \theta)}$$

marginal likelihood

$$q^*(f) = p(f|\theta) \prod_{n=1}^N \underline{t_n(f)}$$

posterior



EP pseudo-point approximation

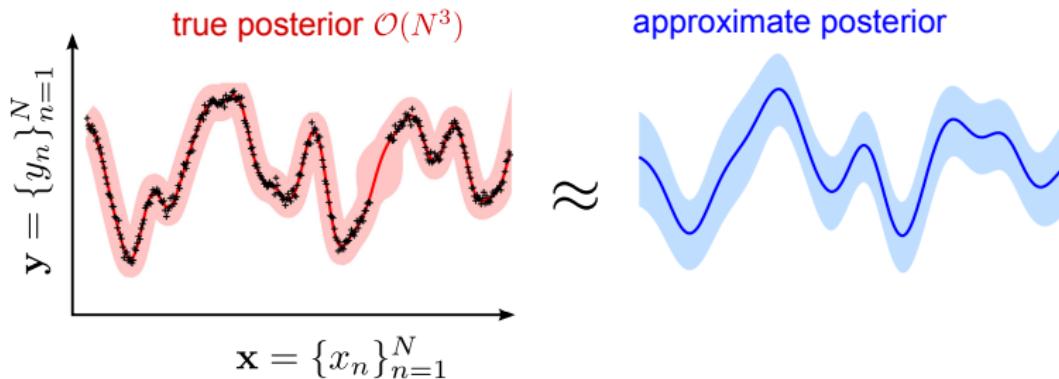
$$p^*(f) = p(f, \mathbf{y} | \mathbf{x}, \theta)$$

$$= p(f|\theta) \prod_{n=1}^N \underline{p(y_n|f, x_n, \theta)}$$

$$= \underbrace{p(\mathbf{y}|\mathbf{x}, \theta)}_{\text{marginal likelihood}} \underbrace{p(f|\mathbf{y}, \mathbf{x}, \theta)}_{\text{posterior}}$$

$$q^*(f) = p(f|\theta) \prod_{n=1}^N \underline{t_n(f)}$$

$$= \underbrace{Z_{\text{EP}}}_{\text{ }} \underbrace{q(f)}_{\text{ }}$$



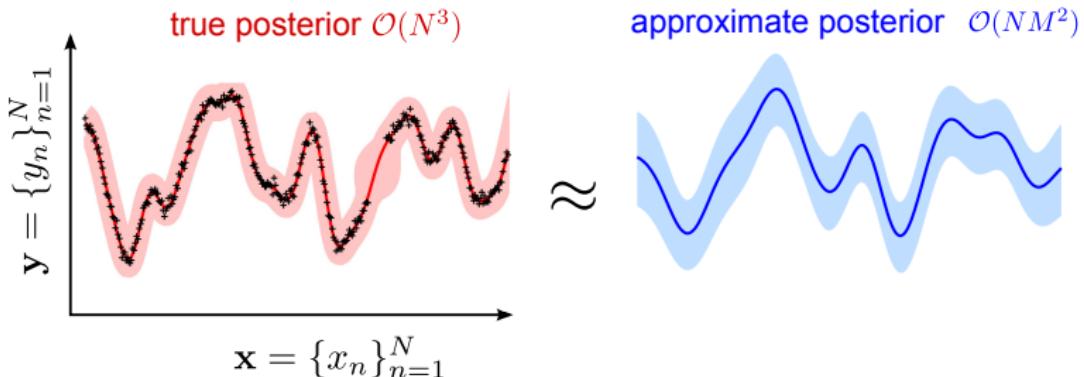
EP pseudo-point approximation

$$p^*(f) = p(f, \mathbf{y} | \mathbf{x}, \theta)$$

$$= p(f | \theta) \prod_{n=1}^N \underline{p(y_n | f, x_n, \theta)}$$

$$= \underbrace{p(\mathbf{y} | \mathbf{x}, \theta)}_{\text{marginal likelihood}} \underbrace{p(f | \mathbf{y}, \mathbf{x}, \theta)}_{\text{posterior}}$$

$$\begin{aligned} q^*(f) &= p(f | \theta) \prod_{n=1}^N \underline{t_n(f)} \\ &= \underline{Z_{\text{EP}}} \underline{q(f)} \\ t_n(f) &= \mathcal{N}(\mathbf{u}; \mu_n, \Sigma_n) \\ \dim(\mathbf{u}) &= M \quad f = \{\mathbf{u}, f_{\neq \mathbf{u}}\} \end{aligned}$$

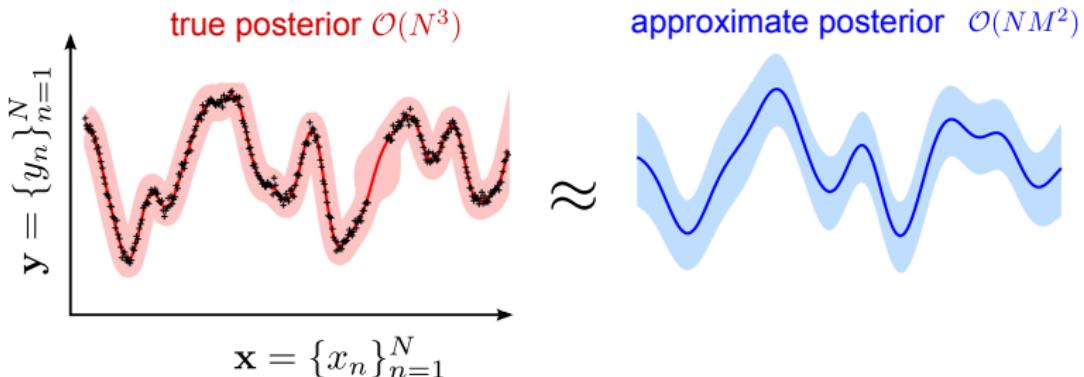


EP pseudo-point approximation

$$\begin{aligned} p^*(f) &= p(f, \mathbf{y} | \mathbf{x}, \theta) \\ &= p(f | \theta) \prod_{n=1}^N \underline{p(y_n | f, x_n, \theta)} \\ &= \underline{p(\mathbf{y} | \mathbf{x}, \theta)} \underline{p(f | \mathbf{y}, \mathbf{x}, \theta)} \end{aligned}$$

marginal likelihood posterior

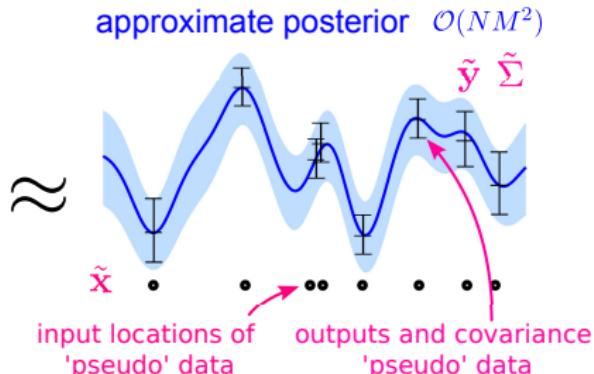
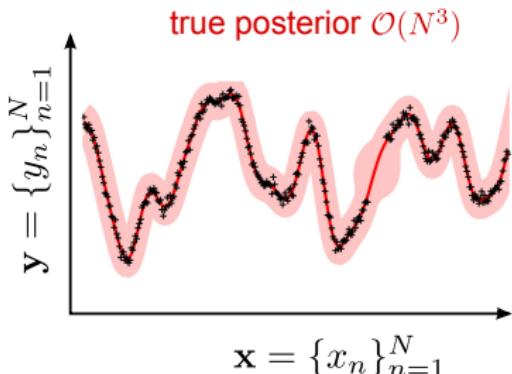
$$\begin{aligned} q^*(f) &= p(f | \theta) p(\tilde{\mathbf{y}} | \mathbf{u}, \tilde{\Sigma}) \\ &= p(f | \theta) \prod_{n=1}^N \underline{t_n(f)} \\ &= \underline{Z_{\text{EP}}} \underline{q(f)} \\ t_n(f) &= \mathcal{N}(\mathbf{u}; \mu_n, \Sigma_n) \\ \dim(\mathbf{u}) = M \quad f &= \{\mathbf{u}, f_{\neq \mathbf{u}}\} \end{aligned}$$



EP pseudo-point approximation

$$\begin{aligned} p^*(f) &= p(f, \mathbf{y} | \mathbf{x}, \theta) \\ &= p(f | \theta) \prod_{n=1}^N p(y_n | f, x_n, \theta) \\ &= \underbrace{p(\mathbf{y} | \mathbf{x}, \theta)}_{\text{marginal likelihood}} \underbrace{p(f | \mathbf{y}, \mathbf{x}, \theta)}_{\text{posterior}} \end{aligned}$$

$$\begin{aligned} q^*(f) &= p(f | \theta) p(\tilde{\mathbf{y}} | \mathbf{u}, \tilde{\Sigma}) \\ &= p(f | \theta) \prod_{n=1}^N t_n(f) \\ &= \underbrace{Z_{\text{EP}}}_{\text{exact joint of new GP regression model}} \underbrace{q(f)}_{t_n(f) = \mathcal{N}(\mathbf{u}; \mu_n, \Sigma_n)} \\ \dim(\mathbf{u}) &= M \quad f = \{\mathbf{u}, f_{\neq \mathbf{u}}\} \end{aligned}$$



EP algorithm

EP algorithm

1. remove

$$q^{\setminus n}(f) = \frac{q^*(f)}{t_n(\mathbf{u})}$$

cavity

take out one
pseudo-observation
likelihood

EP algorithm

1. remove

$$q^{\setminus n}(f) = \frac{q^*(f)}{t_n(\mathbf{u})}$$

cavity

take out one
pseudo-observation
likelihood

2. include

$$p_n^{\text{tilt}}(f) = q^{\setminus n}(f)p(y_n|f, x_n, \theta)$$

tilted

add in one
true observation
likelihood

EP algorithm

1. remove

$$q^{\setminus n}(f) = \frac{q^*(f)}{t_n(\mathbf{u})}$$

cavity

take out one
pseudo-observation
likelihood

2. include

$$p_n^{\text{tilt}}(f) = q^{\setminus n}(f)p(y_n|f, x_n, \theta)$$

tilted

KL between unnormalised
stochastic processes

add in one
true observation
likelihood

3. project

$$q^*(f) = \operatorname{argmin}_{q^*(f)} \text{KL} [p_n^{\text{tilt}}(f) || q^*(f)]$$

project onto
approximating
family

EP algorithm

1. remove

$$q^{\setminus n}(f) = \frac{q^*(f)}{t_n(\mathbf{u})}$$

cavity

take out one
pseudo-observation
likelihood

2. include

$$p_n^{\text{tilt}}(f) = q^{\setminus n}(f)p(y_n|f, x_n, \theta)$$

tilted

KL between unnormalised
stochastic processes

add in one
true observation
likelihood

3. project

$$q^*(f) = \operatorname{argmin}_{q^*(f)} \text{KL} [p_n^{\text{tilt}}(f) || q^*(f)]$$

project onto
approximating
family

4. update

$$t_n(\mathbf{u}) = \frac{q^*(f)}{q^{\setminus n}(f)}$$

update
pseudo-observation
likelihood

EP algorithm

1. remove

$$q^{\setminus n}(f) = \frac{q^*(f)}{t_n(\mathbf{u})}$$

cavity

take out one
pseudo-observation
likelihood

2. include

$$p_n^{\text{tilt}}(f) = q^{\setminus n}(f)p(y_n|f, x_n, \theta)$$

tilted

KL between unnormalised
stochastic processes

add in one
true observation
likelihood

3. project

$$q^*(f) = \underset{q^*(f)}{\operatorname{argmin}} \text{KL} [p_n^{\text{tilt}}(f) || q^*(f)]$$

project onto
approximating
family

1. minimum: moments matched at pseudo-inputs $\mathcal{O}(NM^2)$
2. Gaussian regression: matches moments everywhere

4. update

$$t_n(\mathbf{u}) = \frac{q^*(f)}{q^{\setminus n}(f)}$$

update
pseudo-observation
likelihood

EP algorithm

1. remove

$$q^{\setminus n}(f) = \frac{q^*(f)}{t_n(\mathbf{u})}$$

cavity

take out one
pseudo-observation
likelihood

2. include

$$p_n^{\text{tilt}}(f) = q^{\setminus n}(f)p(y_n|f, x_n, \theta)$$

tilted

KL between unnormalised
stochastic processes

add in one
true observation
likelihood

3. project

$$q^*(f) = \underset{q^*(f)}{\operatorname{argmin}} \text{KL} [p_n^{\text{tilt}}(f) || q^*(f)]$$

project onto
approximating
family

1. minimum: moments matched at pseudo-inputs $\mathcal{O}(NM^2)$
2. Gaussian regression: matches moments everywhere

4. update

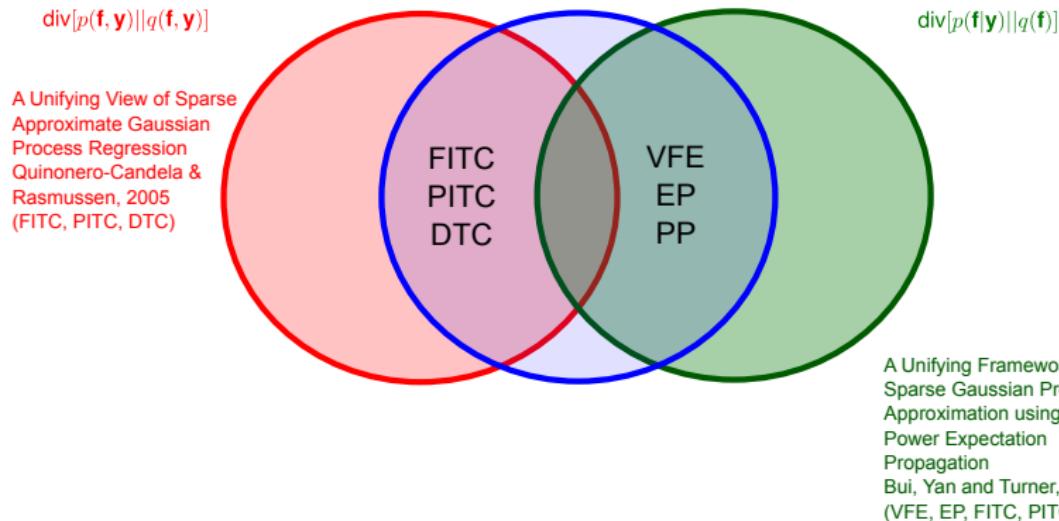
$$\begin{aligned} t_n(\mathbf{u}) &= \frac{q^*(f)}{q^{\setminus n}(f)} \\ &= z_n \mathcal{N}(\mathbf{K}_{f_n \mathbf{u}} \mathbf{K}_{\mathbf{u} \mathbf{u}}^{-1} \mathbf{u}; g_n, v_n) \end{aligned}$$

update
pseudo-observation
likelihood

rank 1

A Brief History of Gaussian Process Approximations

approximate generative model
exact inference methods employing
pseudo-data exact generative model
approximate inference



FITC: Snelson et al. "Sparse Gaussian Processes using Pseudo-inputs"

PITC: Snelson et al. "Local and global sparse Gaussian process approximations"

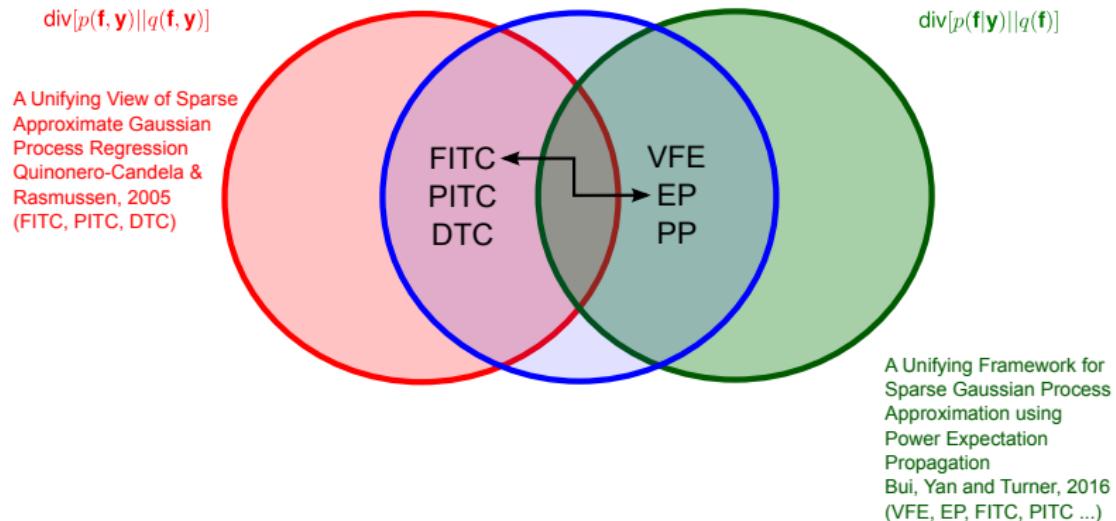
EP: Csato and Opper 2002 / Qi et al. "Sparse-posterior Gaussian Processes for general likelihoods."

VFE: Titsias "Variational Learning of Inducing Variables in Sparse Gaussian Processes"

DTC / PP: Seeger et al. "Fast Forward Selection to Speed Up Sparse Gaussian Process Regression"

Fixed points of EP = FITC approximation

approximate generative model
exact inference methods employing
pseudo-data exact generative model
approximate inference



FITC: Snelson et al. "Sparse Gaussian Processes using Pseudo-inputs"

PITC: Snelson et al. "Local and global sparse Gaussian process approximations"

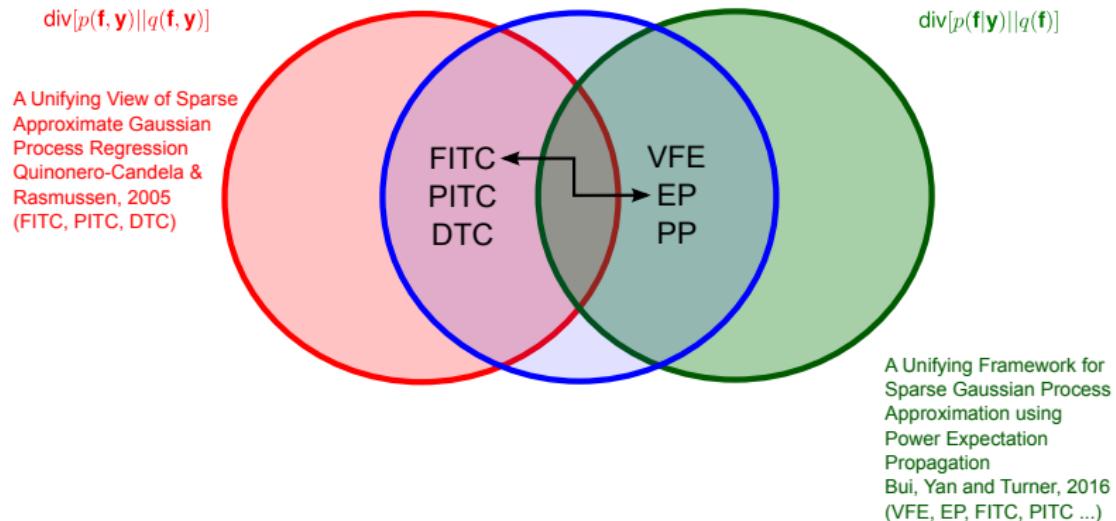
EP: Csato and Opper 2002 / Qi et al. "Sparse-posterior Gaussian Processes for general likelihoods."

VFE: Titsias "Variational Learning of Inducing Variables in Sparse Gaussian Processes"

DTC / PP: Seeger et al. "Fast Forward Selection to Speed Up Sparse Gaussian Process Regression"

Fixed points of EP = FITC approximation

approximate generative model
exact inference methods employing
pseudo-data exact generative model
approximate inference



FITC: Snelson et al. "Sparse Gaussian Processes using Pseudo-inputs"

PITC: Snelson et al. "Local and global sparse Gaussian process approximations"

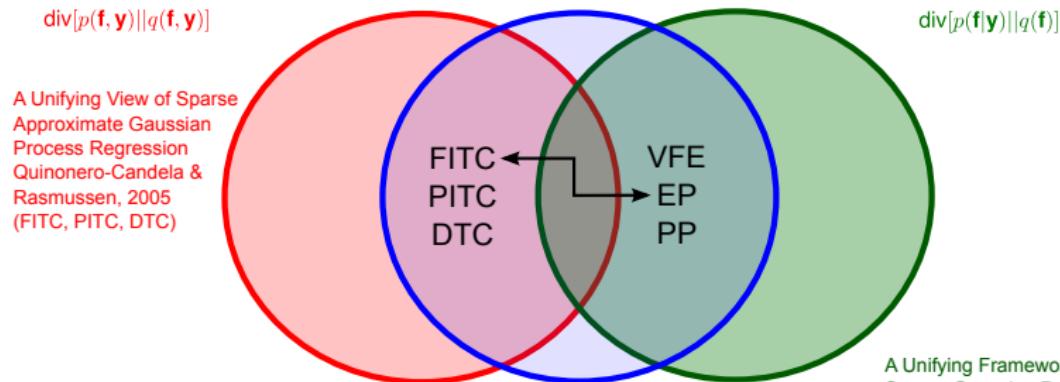
EP: Csato and Opper 2002 / Qi et al. "Sparse-posterior Gaussian Processes for general likelihoods."

VFE: Titsias "Variational Learning of Inducing Variables in Sparse Gaussian Processes"

DTC / PP: Seeger et al. "Fast Forward Selection to Speed Up Sparse Gaussian Process Regression"

Fixed points of EP = FITC approximation

approximate generative model
exact inference methods employing
pseudo-data exact generative model
approximate inference



interpretation resolves issues with FITC:
why does it work so well?
are we allowed to increase M with N

FITC: Snelson et al. "Sparse Gaussian Processes using Pseudo-inputs"

PITC: Snelson et al. "Local and global sparse Gaussian process approximations"

EP: Csato and Opper 2002 / Qi et al. "Sparse-posterior Gaussian Processes for general likelihoods."

VFE: Titsias "Variational Learning of Inducing Variables in Sparse Gaussian Processes"

DTC / PP: Seeger et al. "Fast Forward Selection to Speed Up Sparse Gaussian Process Regression"

EP algorithm

1. remove

$$q^{\setminus n}(f) = \frac{q^*(f)}{t_n(\mathbf{u})}$$

cavity

take out one
pseudo-observation
likelihood

2. include

$$p_n^{\text{tilt}}(f) = q^{\setminus n}(f)p(y_n|f, x_n, \theta)$$

↑
tilted

add in one
true observation
likelihood

3. project

$$q^*(f) = \underset{q^*(f)}{\operatorname{argmin}} \text{KL} [p_n^{\text{tilt}}(f) || q^*(f)]$$

project onto
approximating
family

1. minimum: moments matched at pseudo-inputs $\mathcal{O}(NM^2)$

2. Gaussian regression: matches moments everywhere

4. update

$$t_n(\mathbf{u}) = \frac{q^*(f)}{q^{\setminus n}(f)}$$

$$= z_n \mathcal{N}(\mathbf{K}_{f_n \mathbf{u}} \mathbf{K}_{\mathbf{u} \mathbf{u}}^{-1} \mathbf{u}; g_n, v_n)$$

update
pseudo-observation
likelihood

rank 1

Power EP algorithm (as tractable as EP)

1. remove

$$q^{\setminus n}(f) = \frac{q^*(f)}{t_n(\mathbf{u})^\alpha}$$

cavity

take out fraction of
pseudo-observation
likelihood

2. include

$$p_n^{\text{tilt}}(f) = q^{\setminus n}(f)p(y_n|f, x_n, \theta)^\alpha$$

↑
tilted

add in fraction of
true observation
likelihood

KL between unnormalised
stochastic processes

3. project

$$q^*(f) = \underset{q^*(f)}{\operatorname{argmin}} \text{KL} [p_n^{\text{tilt}}(f) || q^*(f)]$$

project onto
approximating
family

1. minimum: moments matched at pseudo-inputs $\mathcal{O}(NM^2)$

2. Gaussian regression: matches moments everywhere

4. update

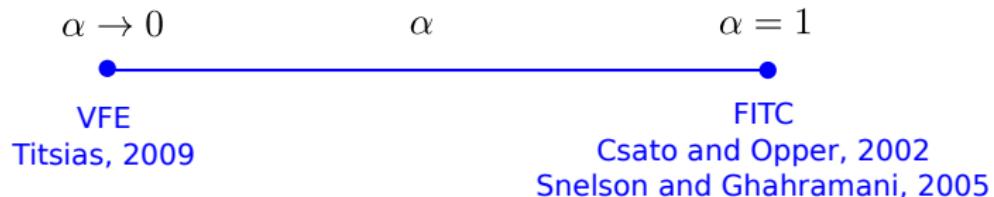
$$t_n(\mathbf{u})^\alpha = \frac{q^*(f)}{q^{\setminus n}(f)}$$

update
pseudo-observation
likelihood

$$t_n(\mathbf{u}) = z_n \mathcal{N}(\mathbf{K}_{f_n \mathbf{u}} \mathbf{K}_{\mathbf{u} \mathbf{u}}^{-1} \mathbf{u}; g_n, v_n)$$

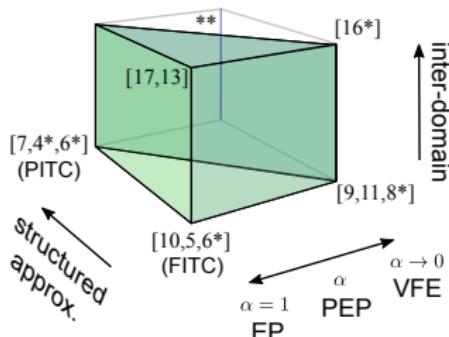
rank 1

Power EP: a unifying framework

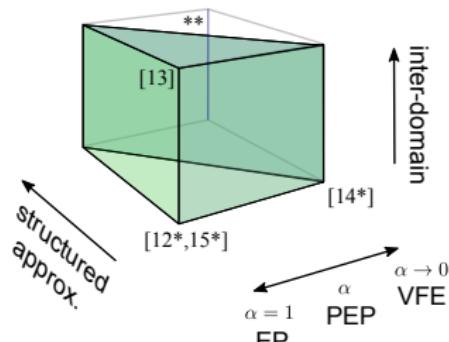


Power EP: a unifying framework

GP Regression



GP Classification



- [4] Quiñonero-Candela et al. 2005
- [5] Snelson et al., 2005
- [6] Snelson, 2006
- [7] Schwaighofer, 2002

* = optimised pseudo-inputs

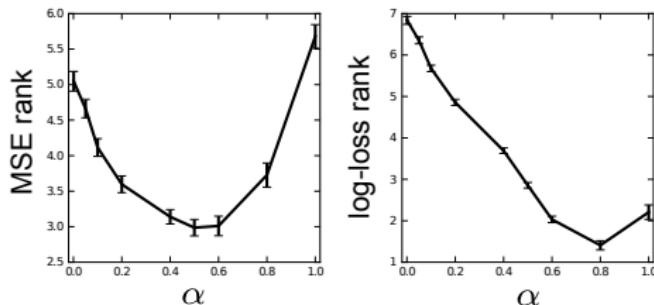
** = structured versions of VFE recover VFE

- [8] Titsias, 2009
- [9] Csató, 2002
- [10] Csató et al., 2002
- [11] Seeger et al., 2003

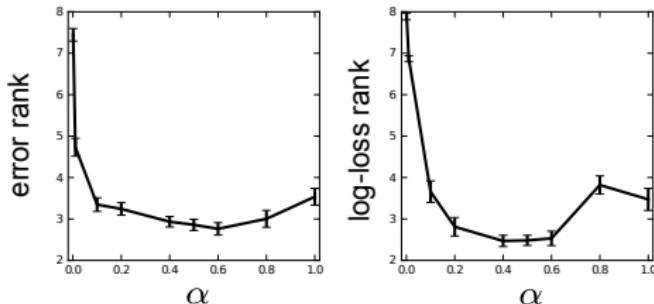
- [12] Naish-Guzman et al, 2007
- [13] Qi et al., 2010
- [14] Hensman et al., 2015
- [15] Hernández-Lobato et al., 2016
- [16] Matthews et al., 2016
- [17] Figueiras-Vidal et al., 2009

How should I set the power parameter α ?

8 UCI regression datasets
20 random splits
 $M = 0 - 200$
hypers and inducing
inputs optimised



6 UCI classification datasets
20 random splits
 $M = 10, 50, 100$
hypers and inducing
inputs optimised



$\alpha = 0.5$ does well on average

References (hyperlinked)

Approximate inference in GPs:

- A Unifying Framework for Sparse Gaussian Process Approximation using Power Expectation Propagation, arXiv preprint 2016

Scalable Approximate inference:

- Stochastic Expectation Propagation, NIPS 2015
- Black-box α -divergence Minimization, ICML 2016

Deep Gaussian Processes (incl. comparisons to Bayesian Neural Networks and GPs):

- Deep Gaussian Processes for Regression using Approximate Expectation Propagation, ICML 2016