DS-GA 1810.001 Modeling time series data

L11. (From last time) Fast GP approximations

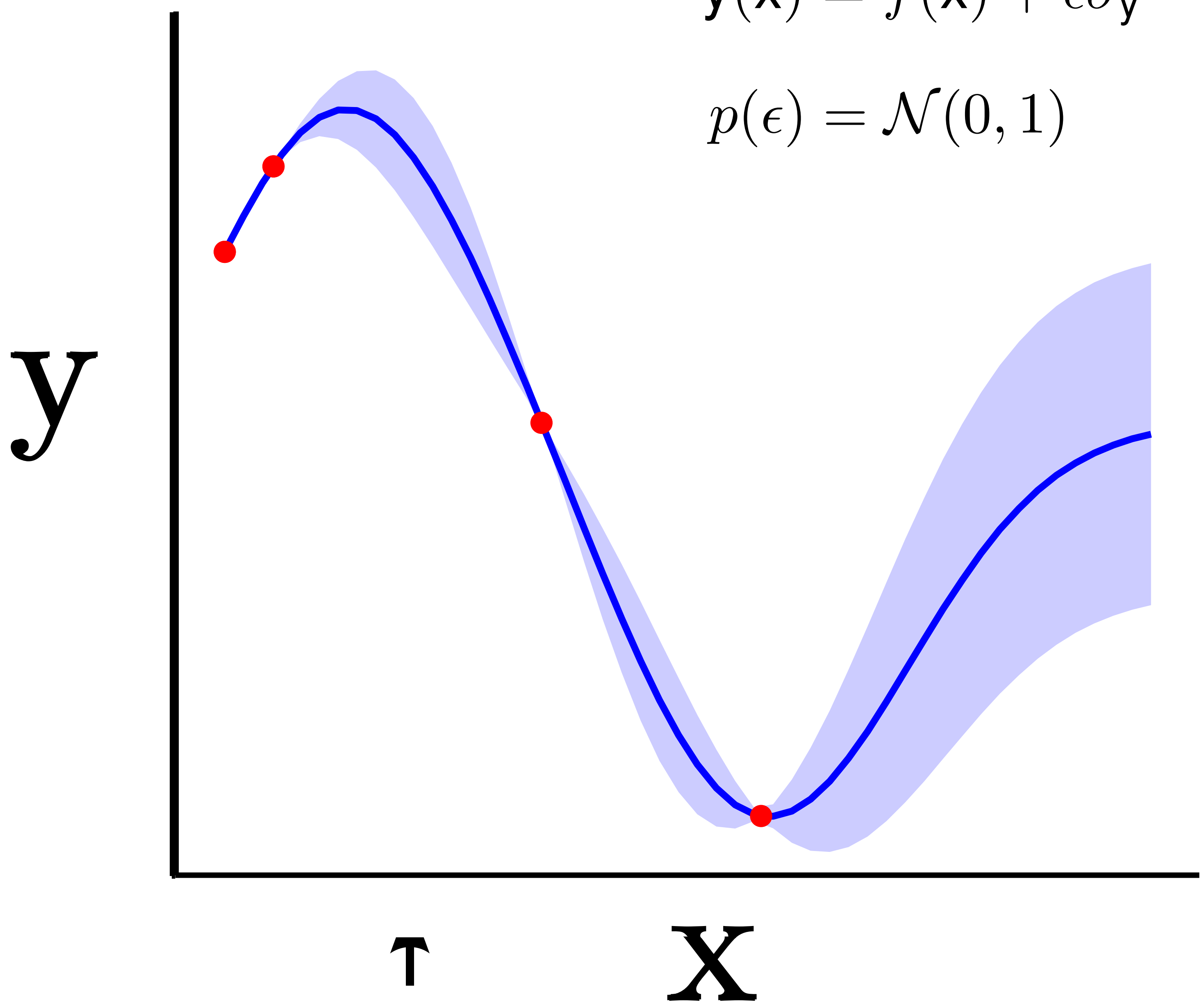
Instructor: Cristina Savin

NYU, CNS & CDS

Quick recap
1. Motivation: nonlinear regression

$$\mathsf{y}(\mathsf{x}) = f(\mathsf{x}) + \epsilon\sigma_{\mathsf{y}}$$

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

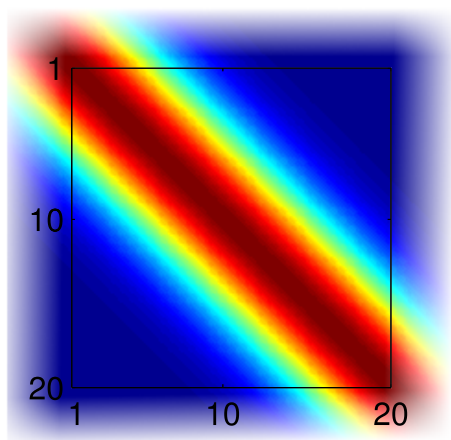


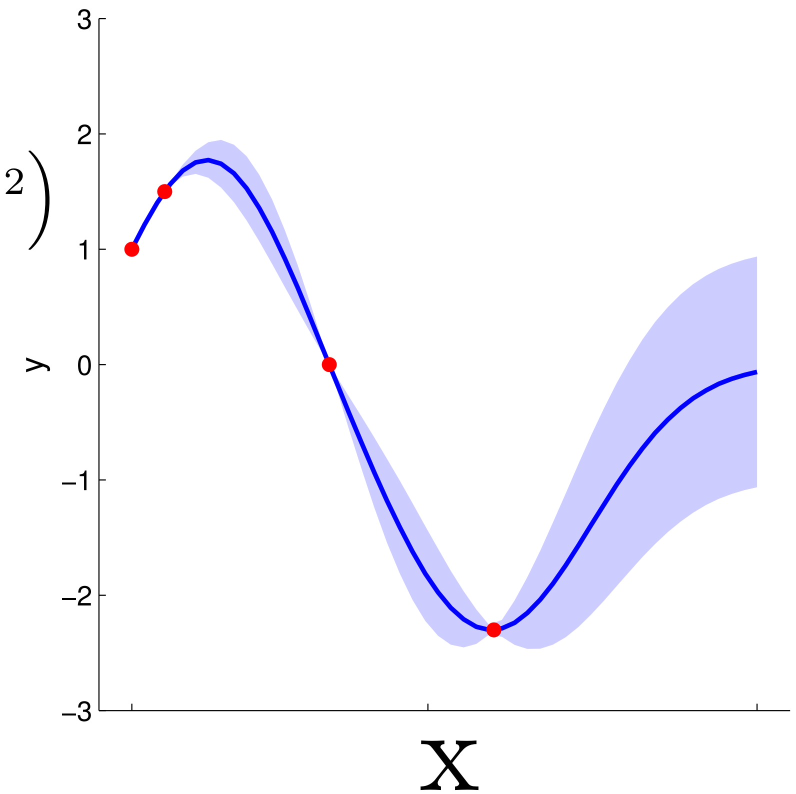
Quick recap
2.prior over functions: GP

$$\Sigma(\mathsf{x}_1, \mathsf{x}_2) = \mathsf{K}(\mathsf{x}_1, \mathsf{x}_2) + \mathsf{I}\sigma_{\mathsf{y}}^2$$

$$\mathsf{K}(\mathsf{x}_1, \mathsf{x}_2) = \sigma^2 \exp\left(-\frac{1}{2\mathsf{l}^2}(\mathsf{x}_1 - \mathsf{x}_2)^2\right)$$

$\Sigma =$





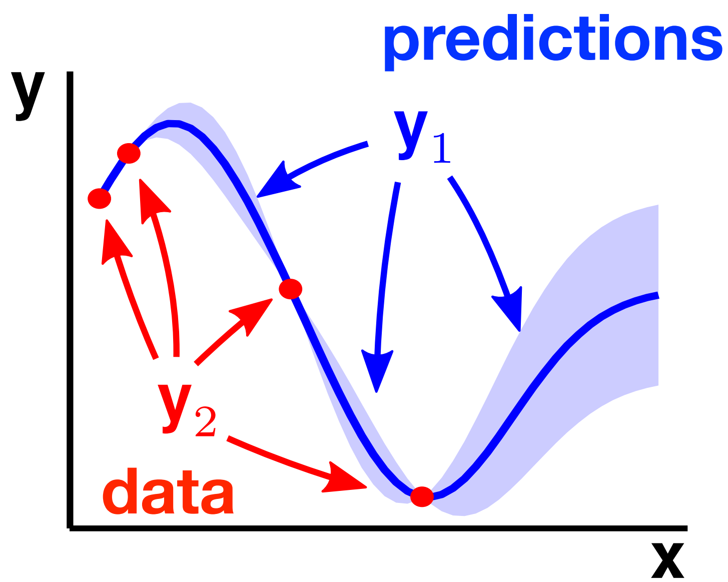
GP: generalization of multivariate gaussian

Definition: a Gaussian process is a collection of random variables, any finite number of which have (consistent) Gaussian distributions.

$$f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), \mathsf{K}(\mathbf{x}, \mathbf{x}')\right)$$ mean+covariance functions

Any function has nonzero probability, preference for certain structure

Quick recap
3. GP inference



$$p(\mathbf{y}_1|\mathbf{y}_2) = \frac{p(\mathbf{y}_1, \mathbf{y}_2)}{p(\mathbf{y}_2)}$$

Jointly gaussian:

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

$$\downarrow$$

$$p(\mathbf{y}_1|\mathbf{y}_2) = \mathcal{N}(\mathbf{a} + \mathsf{B}\mathsf{C}^{-1}(\mathbf{y}_2 - \mathbf{b}), \mathsf{A} - \mathsf{B}\mathsf{C}^{-1}\mathsf{B}^\intercal)$$

predicted mean

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

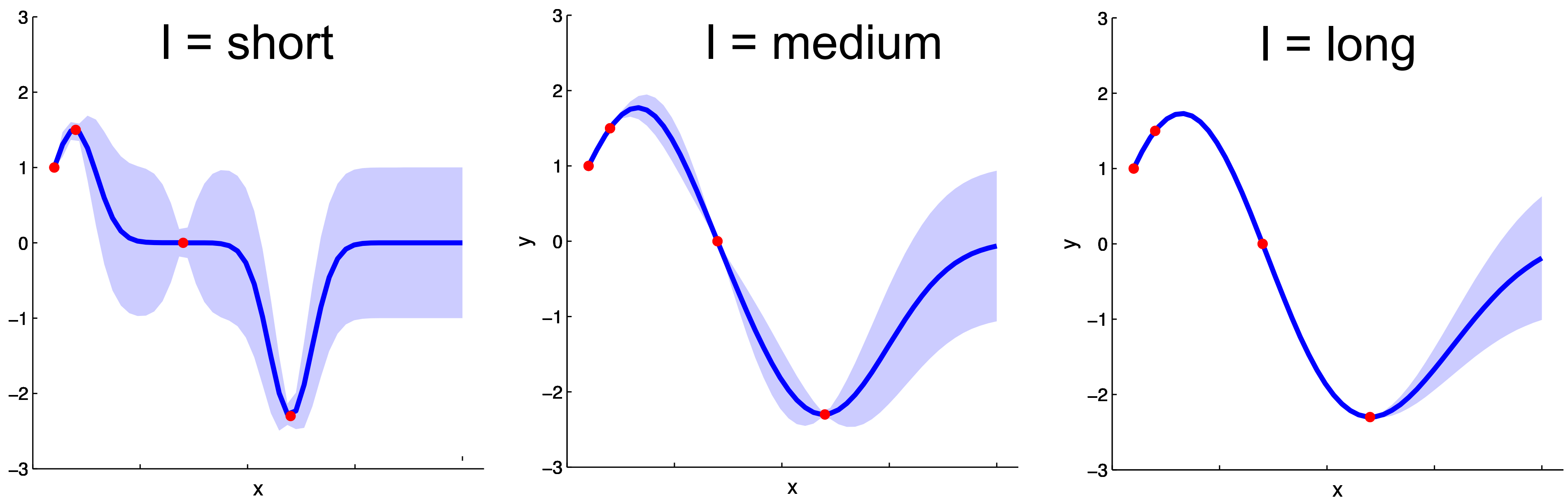
predicted covariance

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

Computational bottleneck!!!

Quick recap
4. GP hyperparameters

$$\mathsf{K}(\mathsf{x}_1, \mathsf{x}_2) = \sigma^2 \exp\left(-\frac{1}{2\mathsf{l}^2}(\mathsf{x}_1 - \mathsf{x}_2)^2\right)$$



Hyperparameters significantly influence outcome

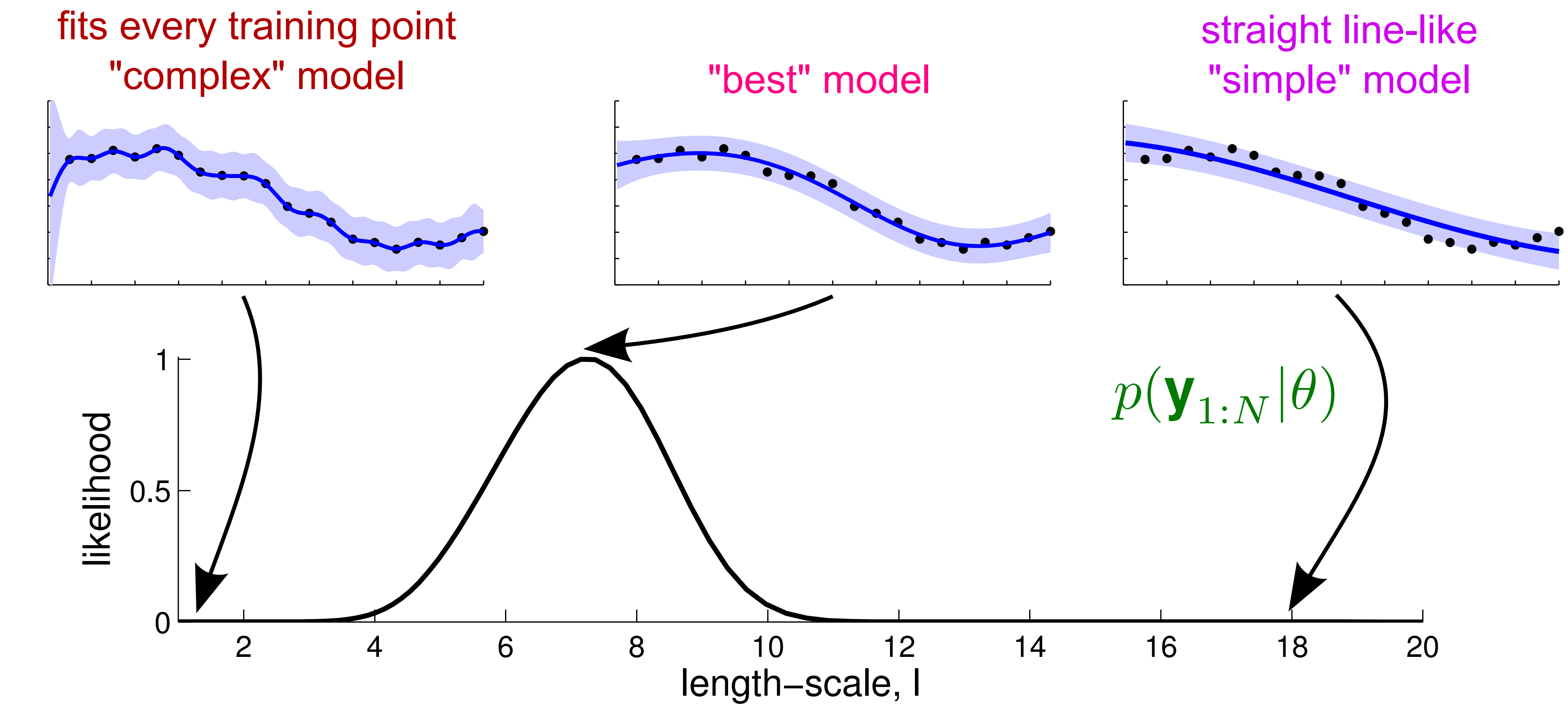
Learn values from data!

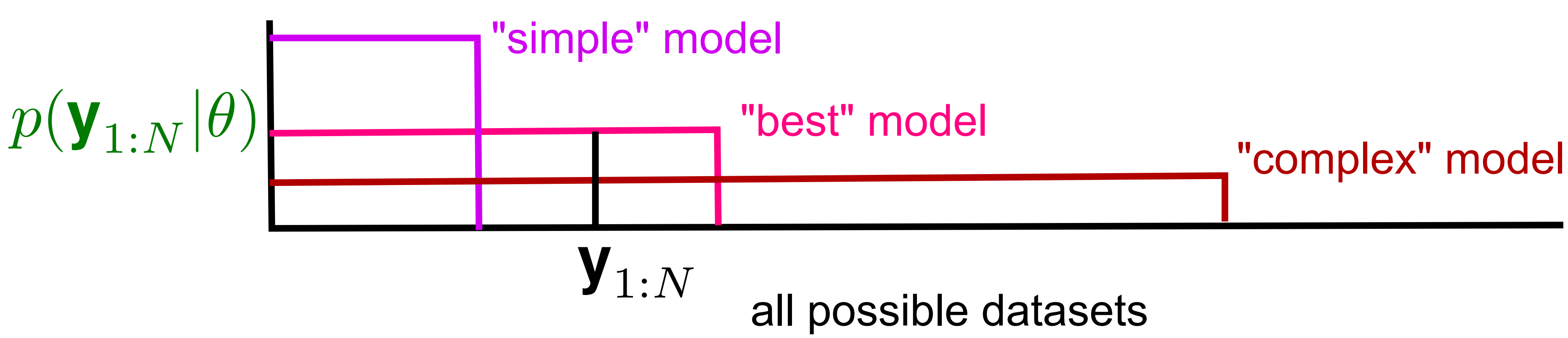
A. Maximum likelihood fit:

$$\operatorname{argmax}_{\theta} \mathrm{P}(\mathbf{y}|\theta)$$

B. Bayesian:

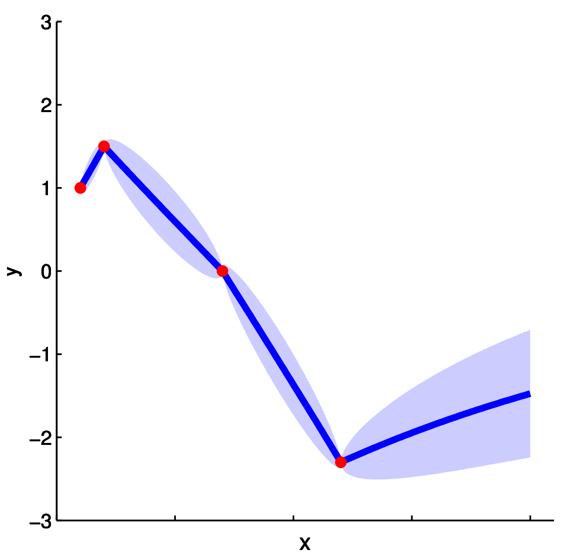
$$p(\theta|\mathbf{y}_{1:N}) = \frac{p(\mathbf{y}_{1:N}|\theta)p(\theta)}{p(\mathbf{y}_{1:N})}$$


fits every training point
"complex" model
"best" model
straight line-like
"simple" model
$p(\mathbf{y}_{1:N}|\theta)$
likelihood
1
0.5
0
2
4
6
8
10
12
14
16
18
20
length−scale, l

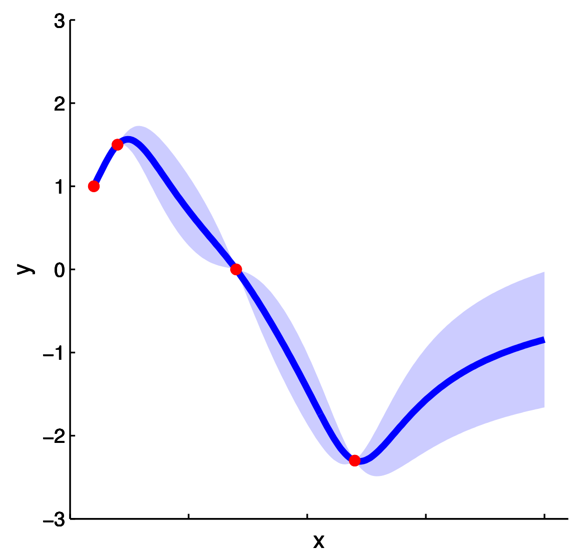

"simple" model
"best" model
"complex" model
$p(\mathbf{y}_{1:N}|\theta)$
$\mathbf{y}_{1:N}$
all possible datasets

Quick recap
5. GP kernels

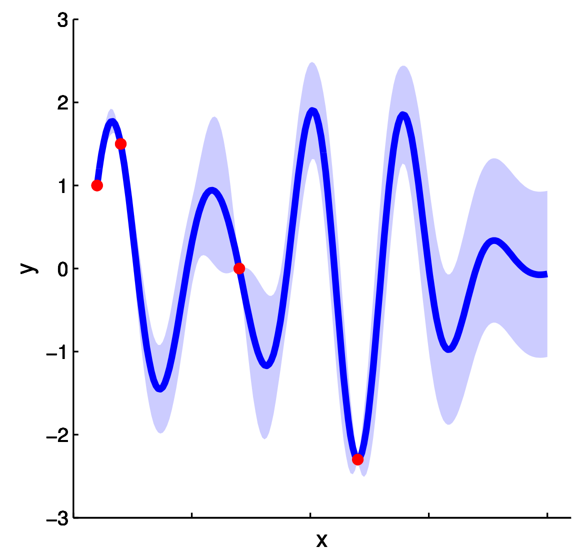
OU



RQ



periodic



Choice of covariance function influence outcome

Marginal likelihood Prior over models

$$p(M|\mathbf{y}_{1:N}) = \frac{p(\mathbf{y}_{1:N}|M)p(M)}{\sum_{M'} p(\mathbf{y}_{1:N}|M')p(M')}$$

$$p(\mathbf{y}_{1:N}|M) = \int \mathrm{d}\theta \; p(\mathbf{y}_{1:N}|\theta, M)p(\theta|M)$$

Usually unpleasant

What are Gaussian Processes good for?

Strengths

interpretable: covariance functions specify easy-to-explain high-level properties of functions

data-efficient: non-parametric + Bayesian ==⇒ lots of flexibility + avoid overfitting

optimal decision making: well-calibrated uncertainties: knows when it does not know

Weaknesses

large datasets: $N \leq 10^5$ unless there is special structure (invert and store covariance matrix)

high-dimensional inputs spaces: $D \leq 10^2$ unless there is special structure, e.g. Kronecker (compute pair-wise elements of covariance function)

A Brief History of Gaussian Process Approximations

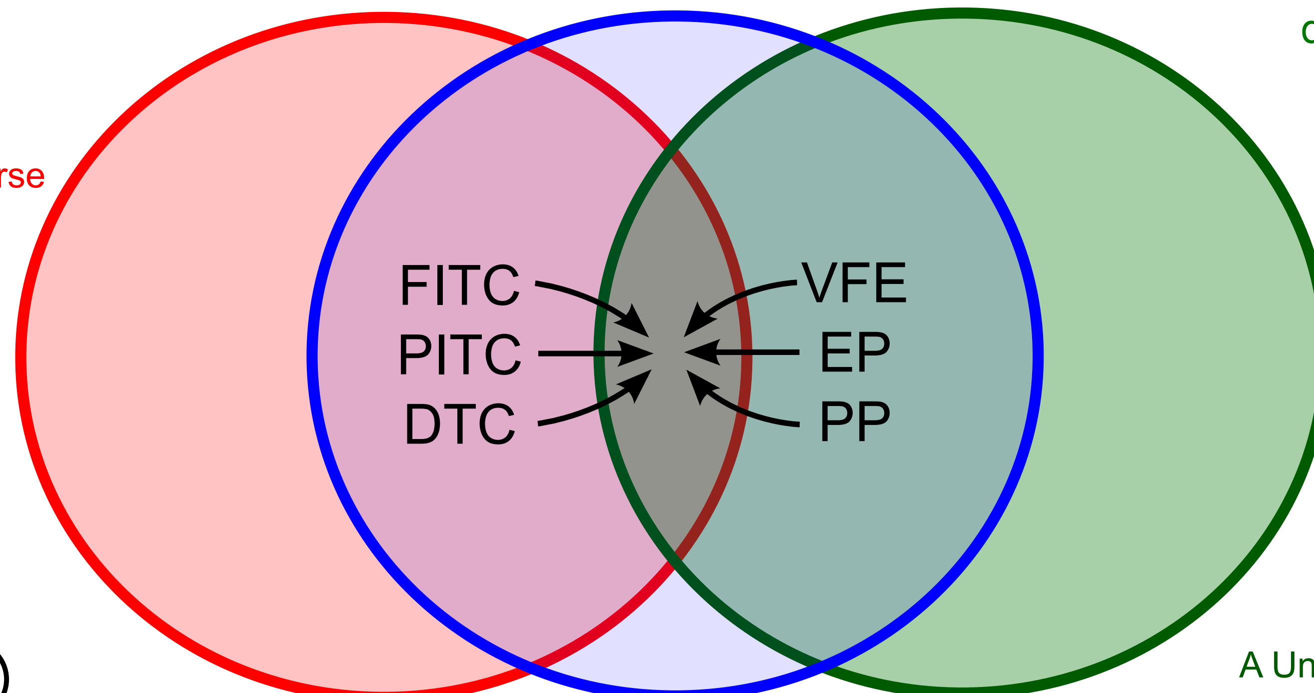
approximate generative model
exact inference

methods employing
pseudo-data

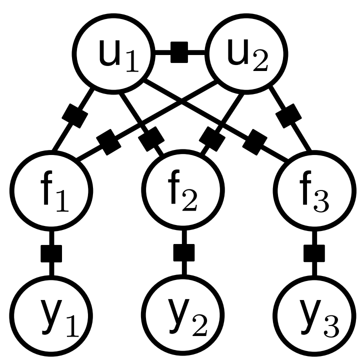
exact generative model
approximate inference

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)



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”

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

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

inference & learning

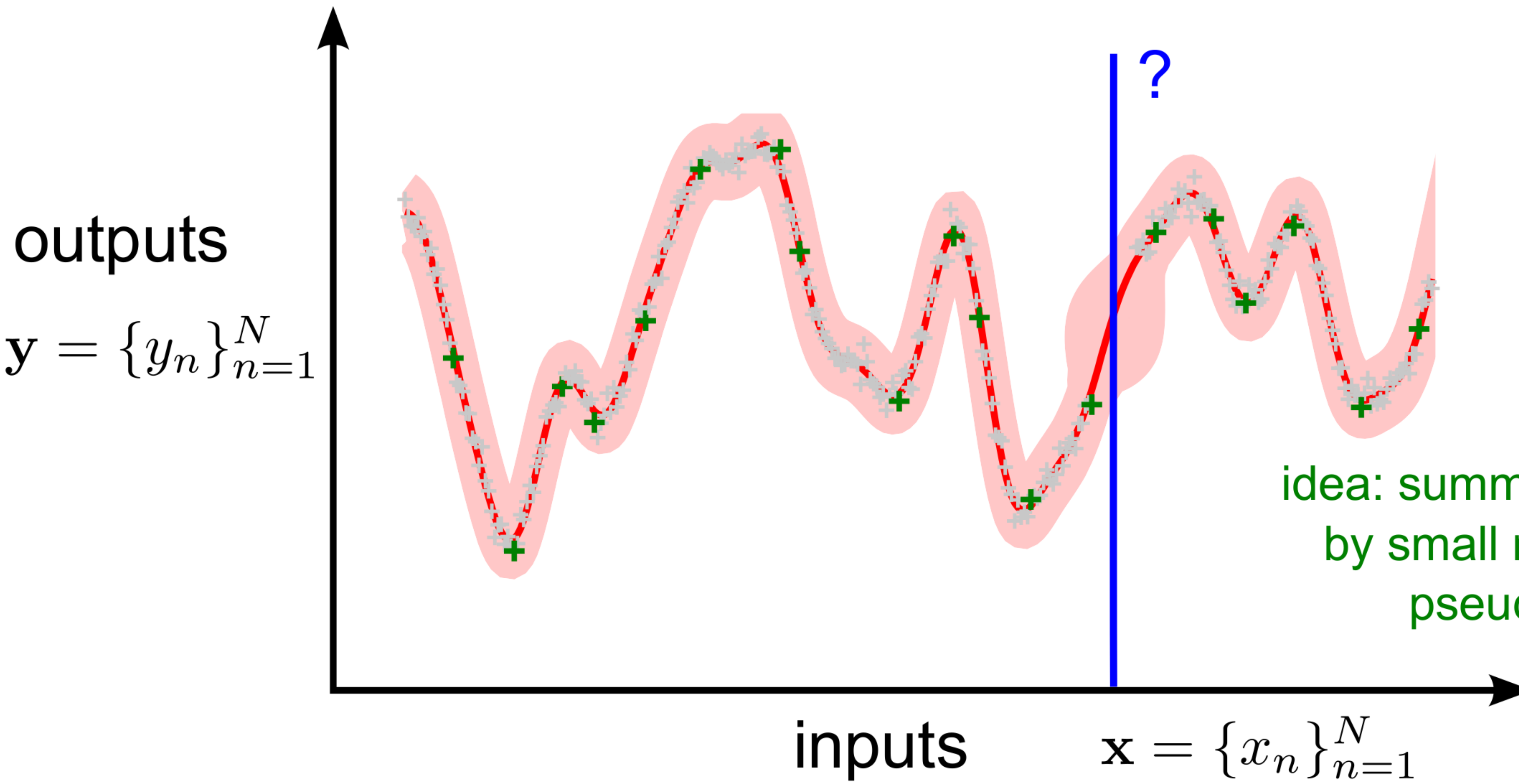
intractabilities

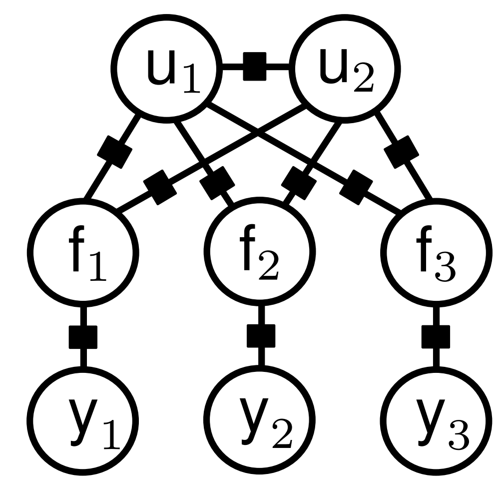
computational $\mathcal{O}(N^3)$

analytic

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

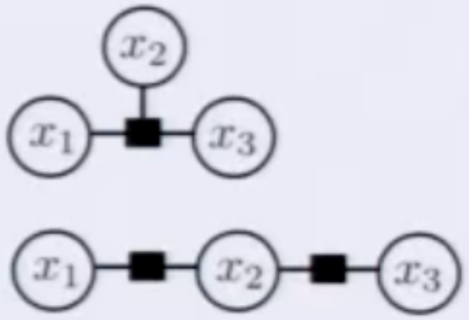
$$p(\mathbf{y}|\mathbf{x}, \theta)$$





idea: summarise dataset by small number (M) pseudo-data

An interlude: dependencies in multivariate gaussians and factor graphs



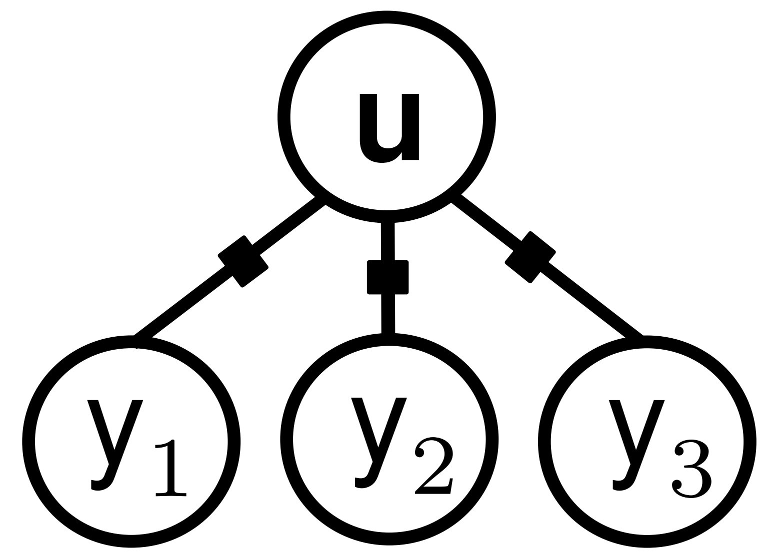
$x_3 \perp x_1 | x_2$

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

FITC

1. Add pseudo points

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



2. drop direct f-f dependencies

3. calibrate model

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

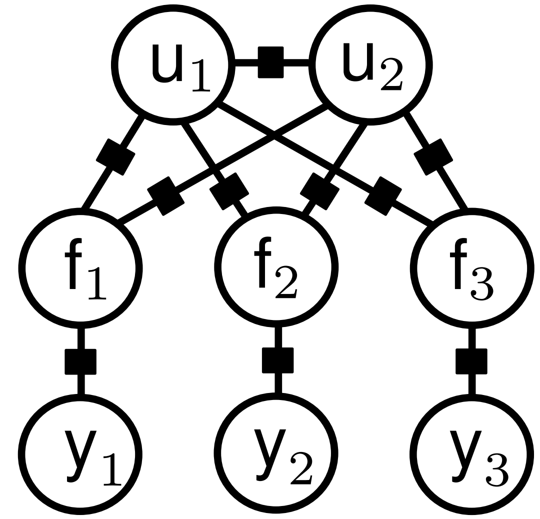
$$\underset{q(\mathbf{u}),\{q(\mathsf{f}_t|\mathbf{u})\}_{t=1}^{T}}{\arg\min} \mathsf{KL}\left(p(\mathbf{f},\mathbf{u})||q(\mathbf{u})\prod_{t=1}^{T} q(\mathsf{f}_t|\mathbf{u})\right) \implies \begin{array}{c} q(\mathbf{u}) = p(\mathbf{u}) \\ q(\mathsf{f}_t|\mathbf{u}) = p(\mathsf{f}_t|\mathbf{u}) \end{array}$$

equal to exact conditionals

4. Replace p with simpler model

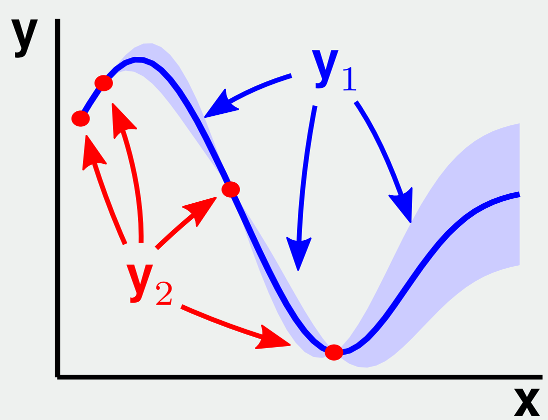
$$q(\mathbf{u}) = p(\mathbf{u}) = \mathcal{N}(\mathbf{u}; 0, \mathsf{K}_{\mathsf{uu}})$$

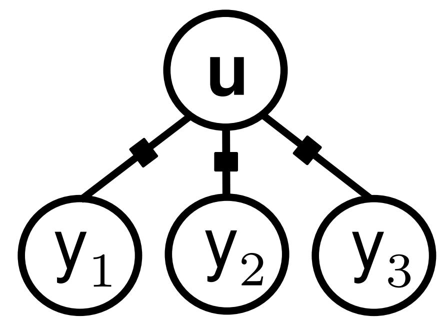
$$q(\mathsf{f}_t|\mathbf{u}) = p(\mathsf{f}_t|\mathbf{u})$$



How do we make predictions?

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





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

$$= \mathcal{N}(\mathsf{f}_t; \mathsf{K}_{\mathsf{f}_t\mathsf{u}}\mathsf{K}_{\mathsf{uu}}^{-1}\mathbf{u}, \underbrace{\mathsf{K}_{\mathsf{f}_t\mathsf{f}_t} - \mathsf{K}_{\mathsf{f}_t\mathsf{u}}\mathsf{K}_{\mathsf{uu}}^{-1}\mathsf{K}_{\mathsf{uf}_t}}_{\mathsf{D}_{tt}})$$

$$q(\mathsf{y}_t|\mathsf{f}_t) = p(\mathsf{y}_t|\mathsf{f}_t) = \mathcal{N}(\mathsf{y}_t; \mathsf{f}_t, \sigma_{\mathsf{y}}^2)$$

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

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

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