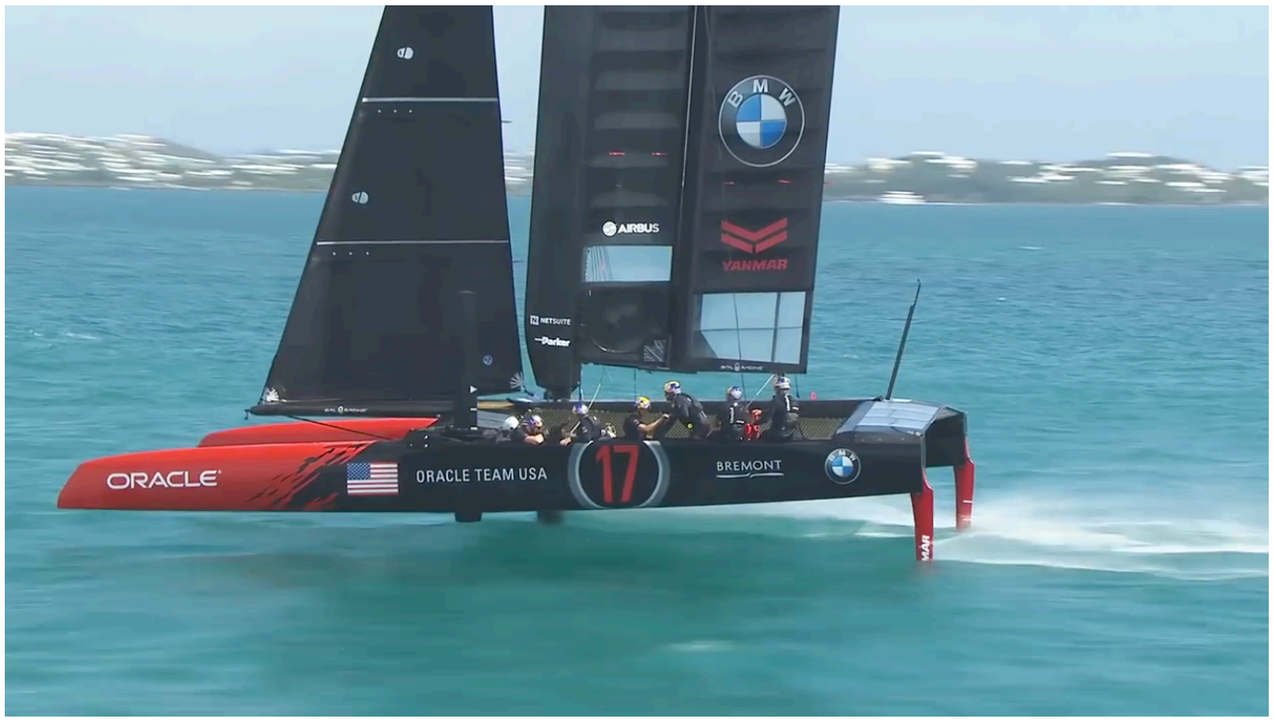
Risk-Assessment, learning and optimization using surrogate models

Paris Perdikaris
Massachusetts Institute of Technology, Department of Mechanical Engineering
Web: `http://web.mit.edu/parisp/www/`
Email: `parisp@mit.edu`

March 1, 2017
SIAM CSE
Atlanta, GA

Motivation



Optimization under uncertainty

$$\boldsymbol{x}, \boldsymbol{\xi} \cdots \blacktriangleright \blacksquare \cdots \blacktriangleright Y(\boldsymbol{x}, \boldsymbol{\xi})$$

1. $L(t) = \{\boldsymbol{x} : \mathcal{R}(Y(\boldsymbol{x}, \boldsymbol{\xi})) \leq t\}$

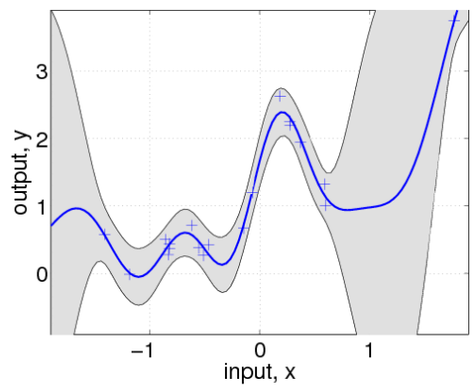
2. $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x} \in \mathbb{R}^d} \mathcal{R}(Y(\boldsymbol{x}, \boldsymbol{\xi}))$

- Assess the performance of the system
- Estimate the statistics of the response
- Data acquisition under limited budgets

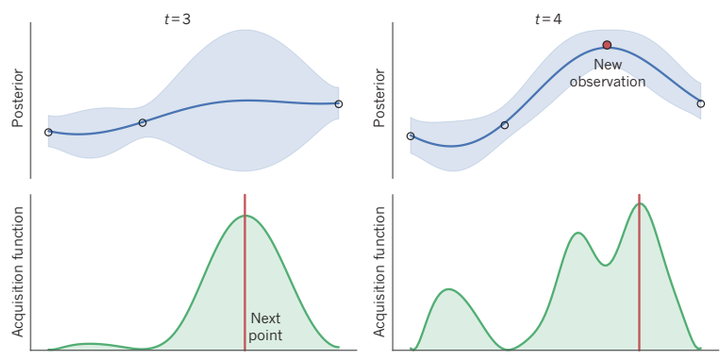
Model & parametric uncertainty

Surrogate uncertainty

Learning uncertainty



Gaussian process regression



Active learning & Bayesian optimization

Gaussian processes

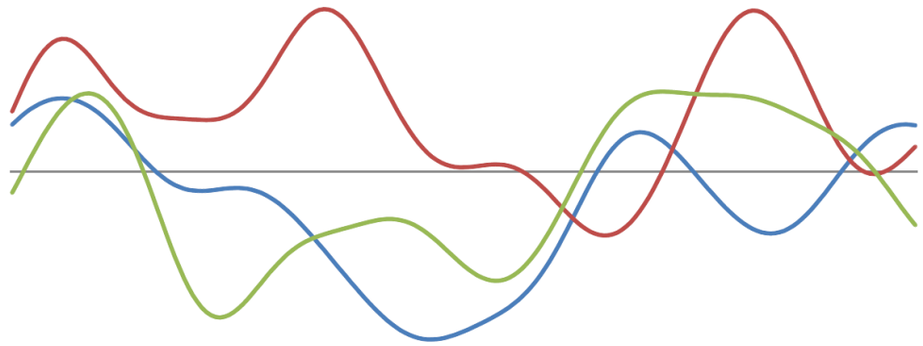
Starting point: The multivariate Gaussian distribution

$$p(\underbrace{f_1, f_2, \cdots, f_s}_{\mathbf{f}_A}, \underbrace{f_{s+1}, f_{s+2}, \cdots, f_N}_{\mathbf{f}_B}) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}) \qquad \boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_A \\ \boldsymbol{\mu}_B \end{bmatrix} \text{ and } \mathbf{K} = \begin{bmatrix} \mathbf{K}_{AA} & \mathbf{K}_{AB} \\ \mathbf{K}_{BA} & \mathbf{K}_{BB} \end{bmatrix}$$

Generalization: The Gaussian process

$$\boldsymbol{\mu}_\infty = \begin{bmatrix} \boldsymbol{\mu}_{\mathbf{f}} \\ \cdots \\ \cdots \end{bmatrix} \text{ and } \mathbf{K}_\infty = \begin{bmatrix} \mathbf{K}_{\mathbf{ff}} & \cdots \\ \cdots & \cdots \end{bmatrix} \qquad \mathbf{K}_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$$

mean function *covariance function*



Priors over functions: $f \sim \mathcal{GP}(\mu(x), K(\mathbf{x}, \mathbf{x}'; \theta))$

Samples from a GP prior

Infinite dimensional model, but finitely many observations: The marginalization property

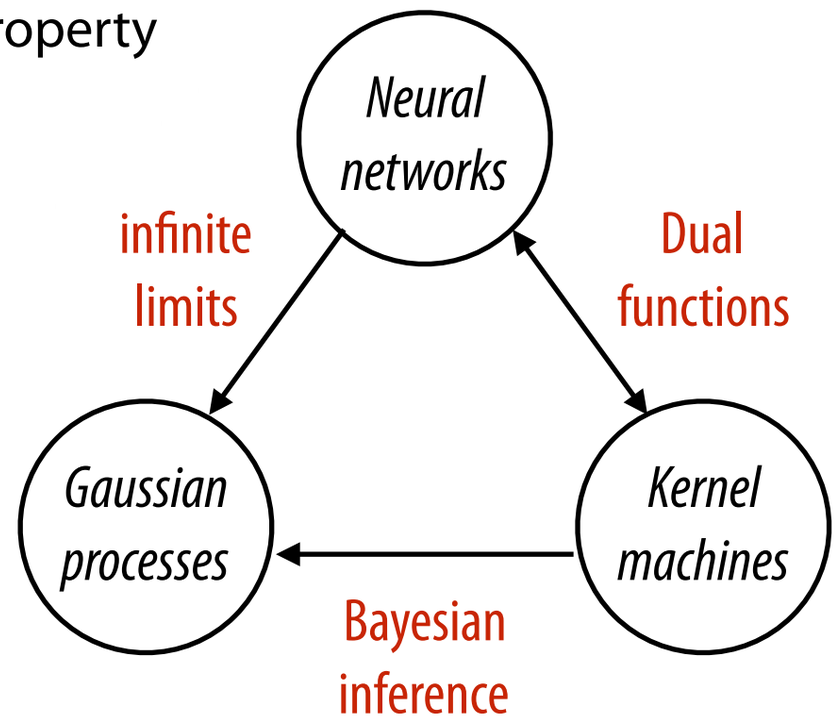
$$p(\mathbf{f}_A, \mathbf{f}_B) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}). \quad \text{Then:}$$

$$p(\mathbf{f}_A) = \int_{\mathbf{f}_B} p(\mathbf{f}_A, \mathbf{f}_B) \mathrm{d}\mathbf{f}_B = \mathcal{N}(\boldsymbol{\mu}_A, \mathbf{K}_{AA})$$

Posterior is also Gaussian:

$$p(\mathbf{f}_A, \mathbf{f}_B) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}). \quad \text{Then:}$$

$$p(\mathbf{f}_A | \mathbf{f}_B) = \mathcal{N}(\boldsymbol{\mu}_A + \mathbf{K}_{AB}\mathbf{K}_{BB}^{-1}(\mathbf{f}_B - \boldsymbol{\mu}_B), \mathbf{K}_{AA} - \mathbf{K}_{AB}\mathbf{K}_{BB}^{-1}\mathbf{K}_{BA})$$



Rasmussen, C. E. Gaussian processes for machine learning (2006)

Nonlinear regression with Gaussian processes

$$y = f(\mathbf{x}) + \epsilon, \qquad f \sim \mathcal{GP}(\mu(x), K(\mathbf{x}, \mathbf{x}'; \theta))$$

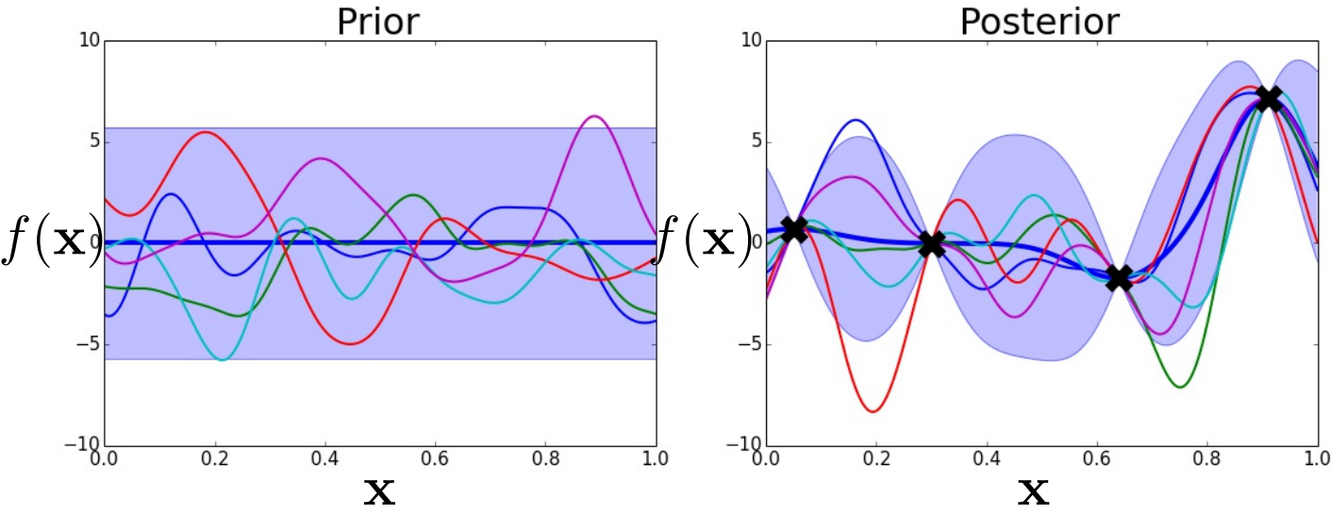
History:

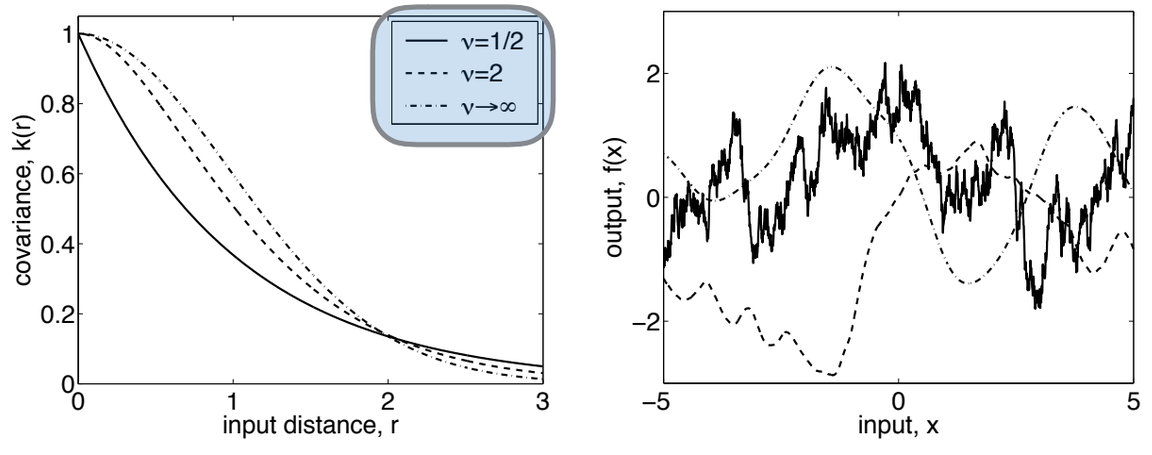
- Wiener–Kolmogorov filtering (1940)
- Kriging (spatial statistics, 1970)
- GP regression (machine learning, 1996)

Workflow:

- Assign a Gaussian process (GP) prior over functions
- Given a training set of observations (x,y) calibrate the GP hyper-parameters
- Use the conditional posterior [f|y] to infer predictions for unobserved x's with quantified uncertainty

covariance function	expression
constant	σ_0^2
linear	$\sum_{d=1}^{D} \sigma_d^2 x_d x'_d$
polynomial	$(\mathbf{x} \cdot \mathbf{x}' + \sigma_0^2)^p$
squared exponential	$\exp(-\frac{r^2}{2\ell^2})$
Matérn	$\frac{1}{2^{\nu-1}\Gamma(\nu)} \left(\frac{\sqrt{2\nu}}{\ell} r\right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu}}{\ell} r\right)$
exponential	$\exp(-\frac{r}{\ell})$
γ-exponential	$\exp\left(-(\frac{r}{\ell})^\gamma\right)$
rational quadratic	$(1 + \frac{r^2}{2\alpha\ell^2})^{-\alpha}$
neural network	$\sin^{-1}\left(\frac{2\tilde{\mathbf{x}}^\top \Sigma \tilde{\mathbf{x}}'}{\sqrt{(1+2\tilde{\mathbf{x}}^\top \Sigma \tilde{\mathbf{x}})(1+2\tilde{\mathbf{x}}'^\top \Sigma \tilde{\mathbf{x}}')}}\right)$





Rasmussen, C. E. Gaussian processes for machine learning (2006)

Training & prediction

Hyper-parameter estimation:

fequentist approach

The vector of hyper-parameters $\boldsymbol{\theta}$ is determined by maximizing the marginal log-likelihood of the observed data (the so called model evidence), i.e.,

$$\log p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\theta}) = -\frac{1}{2}\log|\boldsymbol{K}+\sigma_\epsilon^2\boldsymbol{I}| - \frac{1}{2}\boldsymbol{y}^T(\boldsymbol{K}+\sigma_\epsilon^2\boldsymbol{I})^{-1}\boldsymbol{y} - \frac{N}{2}\log 2\pi \tag{8}$$

Bayesian approach

Assign priors over the hyper parameters and marginalize them out using MCMC.

Prediction:

If we consider a Gaussian likelihood $p(\boldsymbol{y}|\boldsymbol{f}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{f},\sigma_\epsilon^2\boldsymbol{I})$ then the posterior distribution $p(\boldsymbol{f}|\boldsymbol{y},\boldsymbol{X})$ is tractable and can be used to perform predictive inference for a new output f_*, given a new input $\boldsymbol{x}_*$ as

$$p(f_*|\boldsymbol{y},\boldsymbol{X},\boldsymbol{x}_*) = \mathcal{N}(f_*|\mu_*,\sigma_*^2), \tag{5}$$

$$\mu_*(\boldsymbol{x}_*) = \boldsymbol{k}_{*N}(\boldsymbol{K}+\sigma_\epsilon^2\boldsymbol{I})^{-1}\boldsymbol{y}, \tag{6}$$

$$\sigma_*^2(\boldsymbol{x}_*) = \boldsymbol{k}_{**} - \boldsymbol{k}_{*N}(\boldsymbol{K}+\sigma_\epsilon^2\boldsymbol{I})^{-1}\boldsymbol{k}_{N*}, \tag{7}$$

where $\boldsymbol{k}_{*N} = [k(\boldsymbol{x}_*,\boldsymbol{x}_1),\ldots,k(\boldsymbol{x}_*,\boldsymbol{x}_N)]$, $\boldsymbol{k}_{N*} = \boldsymbol{k}_{*N}^T$, and $\boldsymbol{k}_{**} = k(\boldsymbol{x}_*,\boldsymbol{x}_*)$. Predictions are computed using the posterior mean μ_*, while prediction uncertainty is quantified through the posterior variance σ_*^2.

Rasmussen, C. E. Gaussian processes for machine learning (2006)

Active learning of level sets

Goal: Identify the sets $L(t) = \{\boldsymbol{x} : g(\boldsymbol{x}) \leq t\}, \;\; g(\boldsymbol{x}) := \mathcal{R}\left(Y(\boldsymbol{x}, \boldsymbol{\xi})\right)$

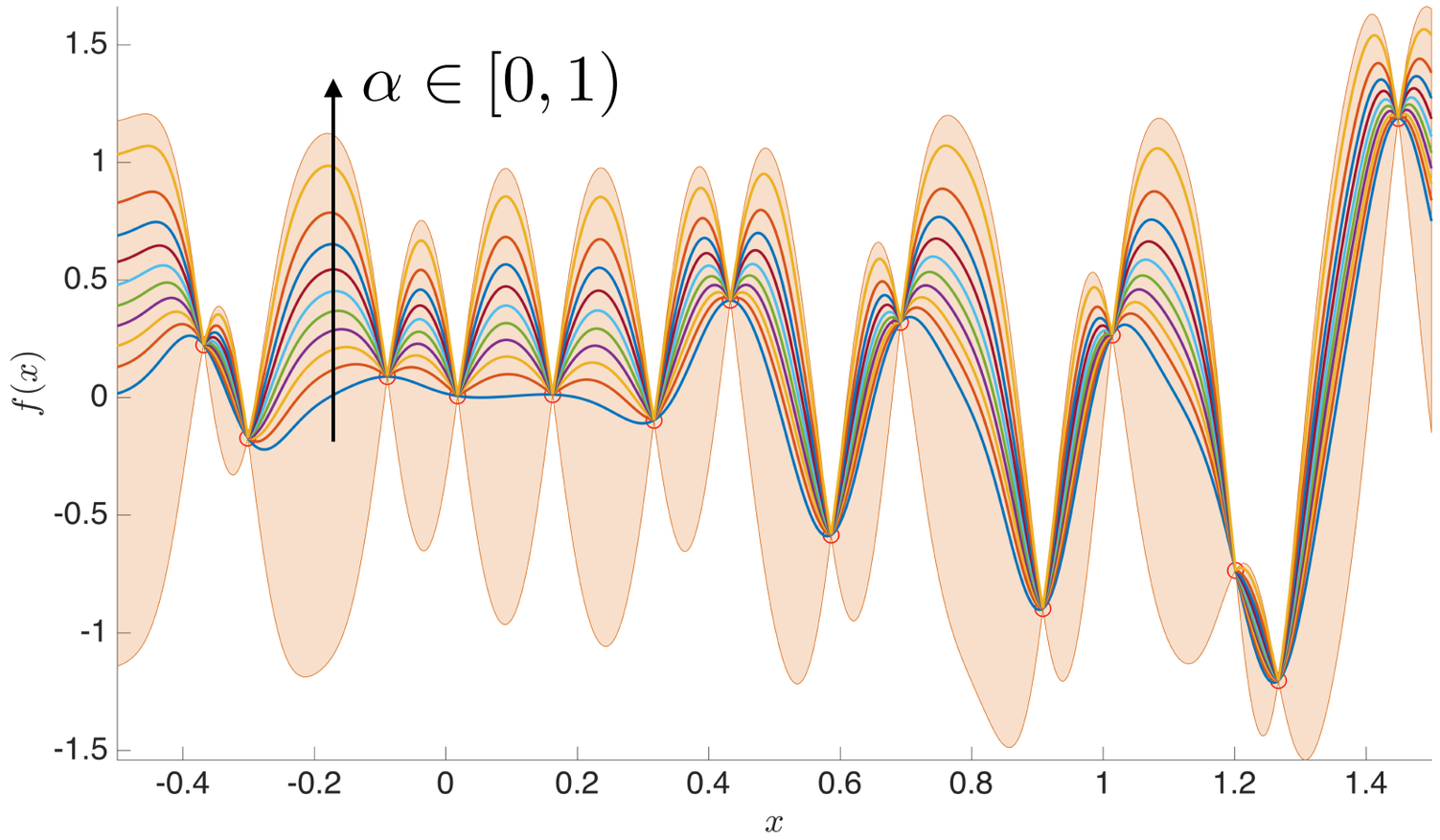
Approximate the true objective with a GP surrogate: $g(\boldsymbol{x}) \approx f(\boldsymbol{x}) \sim \mathcal{GP}(f|0, k(\boldsymbol{x}, \boldsymbol{x}'; \theta))$

f(x) is random so it becomes natural to quantify "surrogate" uncertainty using for e.g. α-superquintile risk measure:

$$\mathcal{R}_\alpha(f(\boldsymbol{x})) = \min_{c \in \mathbb{R}} \left\{ c + \frac{1}{1-\alpha}\mathbb{E}[\max\{0, f(\boldsymbol{x}) - c\}] \right\}$$

average of the worst (1–α)% outcomes of f(x)

For *f(x)* being a GP this can be simplified to: $\mathcal{R}_\alpha(f(\boldsymbol{x})) = \mu(\boldsymbol{x}) + \frac{\phi(\Phi^{-1}(\alpha))}{1-\alpha}\sigma(\boldsymbol{x})$



Active learning of level sets

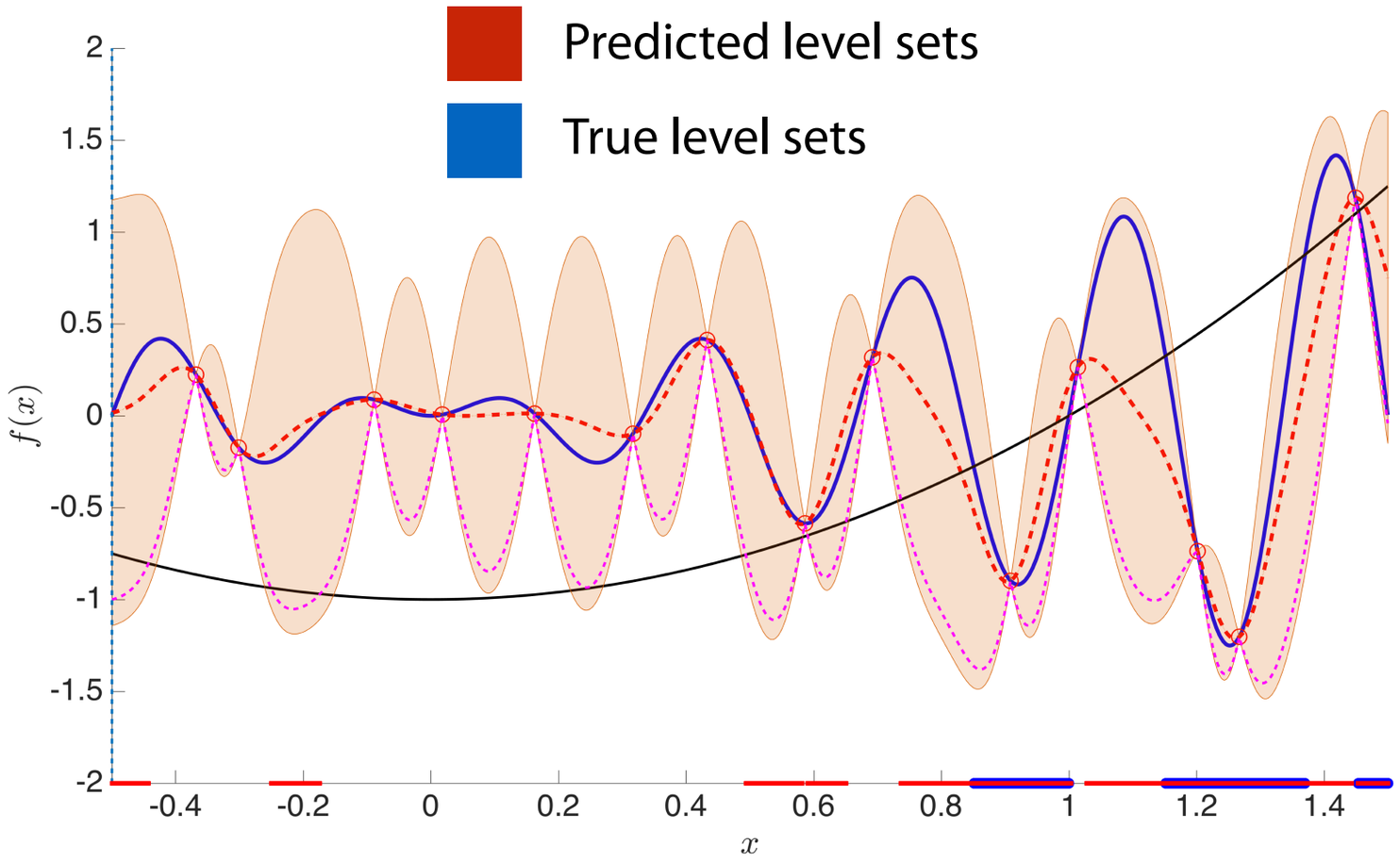
Goal: Identify the sets $L_\alpha(t) = \{\boldsymbol{x} : \mathcal{R}_\alpha(f(\boldsymbol{x})) \leq t\}$

Utilize the posterior to guide a sequential sampling policy by optimizing a chosen expected utility function

$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_\theta \mathbb{E}_{v \mid \mathbf{x},\theta}[U(\mathbf{x}, v, \theta)]$$

e.g. sample at the locations that maximize the posterior variance in L(t)

$$\boldsymbol{x}_{n+1} = \arg \max_{\boldsymbol{x} \in L_\alpha(t)} V(f(\boldsymbol{x}))$$



Terminate iteration when the "volume" of the predicted level sets is below a given threshold:

$$|V_{n+1}(t) - V_n(t)| < \epsilon, \qquad V_n(t) = \int_{L_\alpha(t)} \mathbf{1}_{[-\infty,t]} d\boldsymbol{x}$$

Remarks:

- The choice of risk-averseness level $\alpha \in [0, 1)$ controls the exploration vs exploitation trade-off.
- Upon convergence the predicted levels sets are guaranteed to be a subset of the true level sets.

P. Perdikaris, M. Raissi and J.O. Royset, "Risk-assesment, learning, and optimization using surrogate models", (in preparation), 2017.

Active learning of level sets

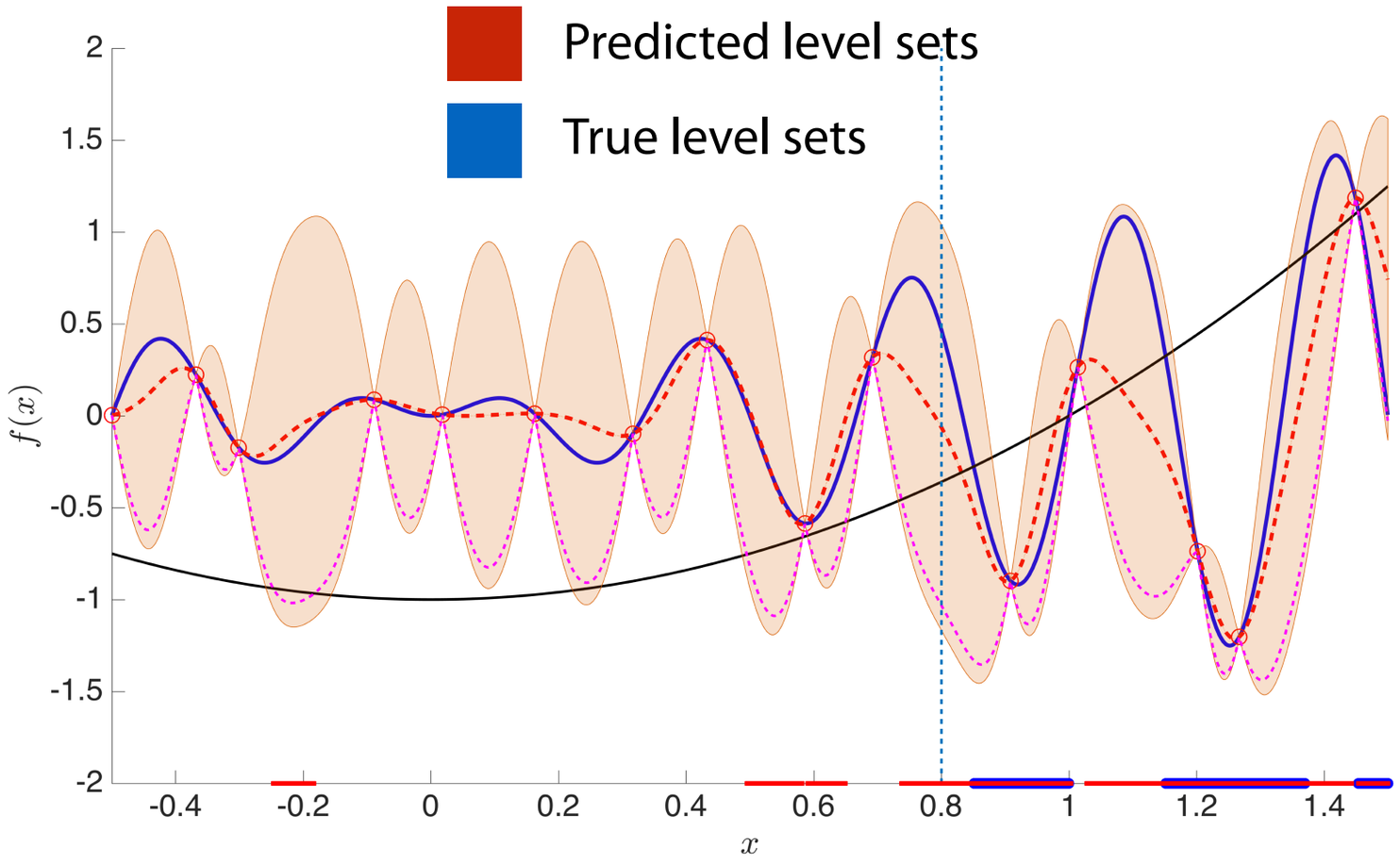
Goal: Identify the sets $L_\alpha(t) = \{\boldsymbol{x} : \mathcal{R}_\alpha(f(\boldsymbol{x})) \leq t\}$

Utilize the posterior to guide a sequential sampling policy by optimizing a chosen expected utility function

$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_\theta \mathbb{E}_{v \mid \mathbf{x}, \theta}[U(\mathbf{x}, v, \theta)]$$

e.g. sample at the locations that maximize the posterior variance in L(t)

$$\boldsymbol{x}_{n+1} = \arg \max_{\boldsymbol{x} \in L_\alpha(t)} V(f(\boldsymbol{x}))$$



Terminate iteration when the "volume" of the predicted level sets is below a given threshold:

$$|V_{n+1}(t) - V_n(t)| < \epsilon, \qquad V_n(t) = \int_{L_\alpha(t)} \mathbf{1}_{[-\infty, t]} d\boldsymbol{x}$$

Remarks:

- The choice of risk-averseness level $\alpha \in [0, 1)$ controls the exploration vs exploitation trade-off.
- Upon convergence the predicted levels sets are guaranteed to be a subset of the true level sets.

P. Perdikaris, M. Raissi and J.O. Royset, "Risk-assesment, learning, and optimization using surrogate models", (in preparation), 2017.

Active learning of level sets

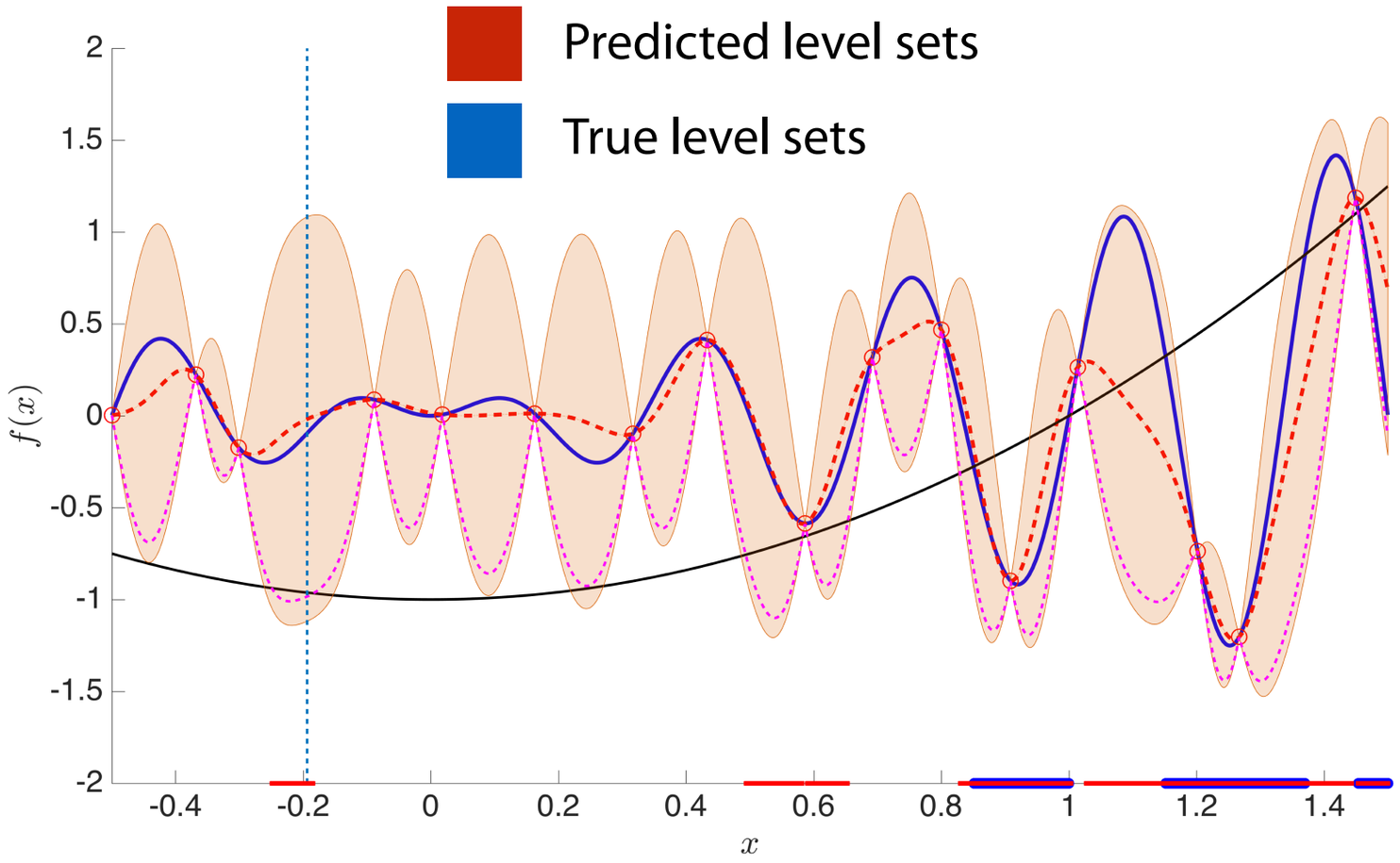
Goal: Identify the sets $L_\alpha(t) = \{\boldsymbol{x} : \mathcal{R}_\alpha(f(\boldsymbol{x})) \leq t\}$

Utilize the posterior to guide a sequential sampling policy by optimizing a chosen expected utility function

$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_\theta \mathbb{E}_{v \,|\, \mathbf{x},\theta}[U(\mathbf{x}, v, \theta)]$$

e.g. sample at the locations that maximize the posterior variance in L(t)

$$\boldsymbol{x}_{n+1} = \arg \max_{\boldsymbol{x} \in L_\alpha(t)} V(f(\boldsymbol{x}))$$



Terminate iteration when the "volume" of the predicted level sets is below a given threshold:

$$|V_{n+1}(t) - V_n(t)| < \epsilon, \qquad V_n(t) = \int_{L_\alpha(t)} \mathbf{1}_{[-\infty,t]} d\boldsymbol{x}$$

Remarks:

- The choice of risk-averseness level $\alpha \in [0, 1)$ controls the exploration vs exploitation trade-off.
- Upon convergence the predicted levels sets are guaranteed to be a subset of the true level sets.

P. Perdikaris, M. Raissi and J.O. Royset, "Risk-assesment, learning, and optimization using surrogate models", (in preparation), 2017.

Active learning of level sets

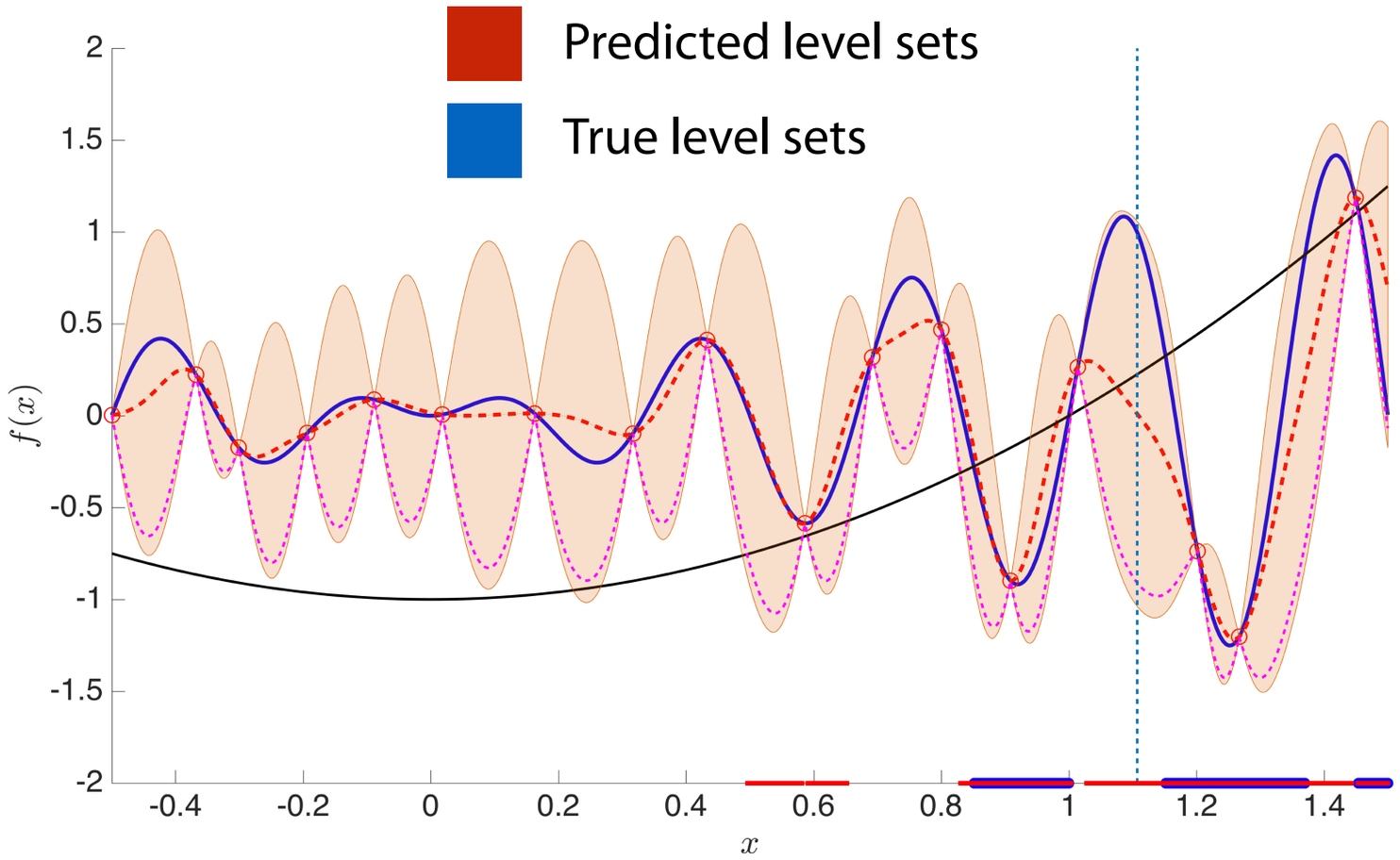
Goal: Identify the sets $L_\alpha(t) = \{\boldsymbol{x} : \mathcal{R}_\alpha(f(\boldsymbol{x})) \leq t\}$

Utilize the posterior to guide a sequential sampling policy by optimizing a chosen expected utility function

$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_\theta \mathbb{E}_{v \mid \mathbf{x},\theta}[U(\mathbf{x}, v, \theta)]$$

e.g. sample at the locations that maximize the posterior variance in L(t)

$$\boldsymbol{x}_{n+1} = \arg \max_{\boldsymbol{x} \in L_\alpha(t)} V(f(\boldsymbol{x}))$$



Terminate iteration when the "volume" of the predicted level sets is below a given threshold:

$$|V_{n+1}(t) - V_n(t)| < \epsilon, \qquad V_n(t) = \int_{L_\alpha(t)} \mathbf{1}_{[-\infty,t]} d\boldsymbol{x}$$

Remarks:

- The choice of risk-averseness level $\alpha \in [0, 1)$ controls the exploration vs exploitation trade-off.
- Upon convergence the predicted levels sets are guaranteed to be a subset of the true level sets.

P. Perdikaris, M. Raissi and J.O. Royset, "Risk-assesment, learning, and optimization using surrogate models", (in preparation), 2017.

Active learning of level sets

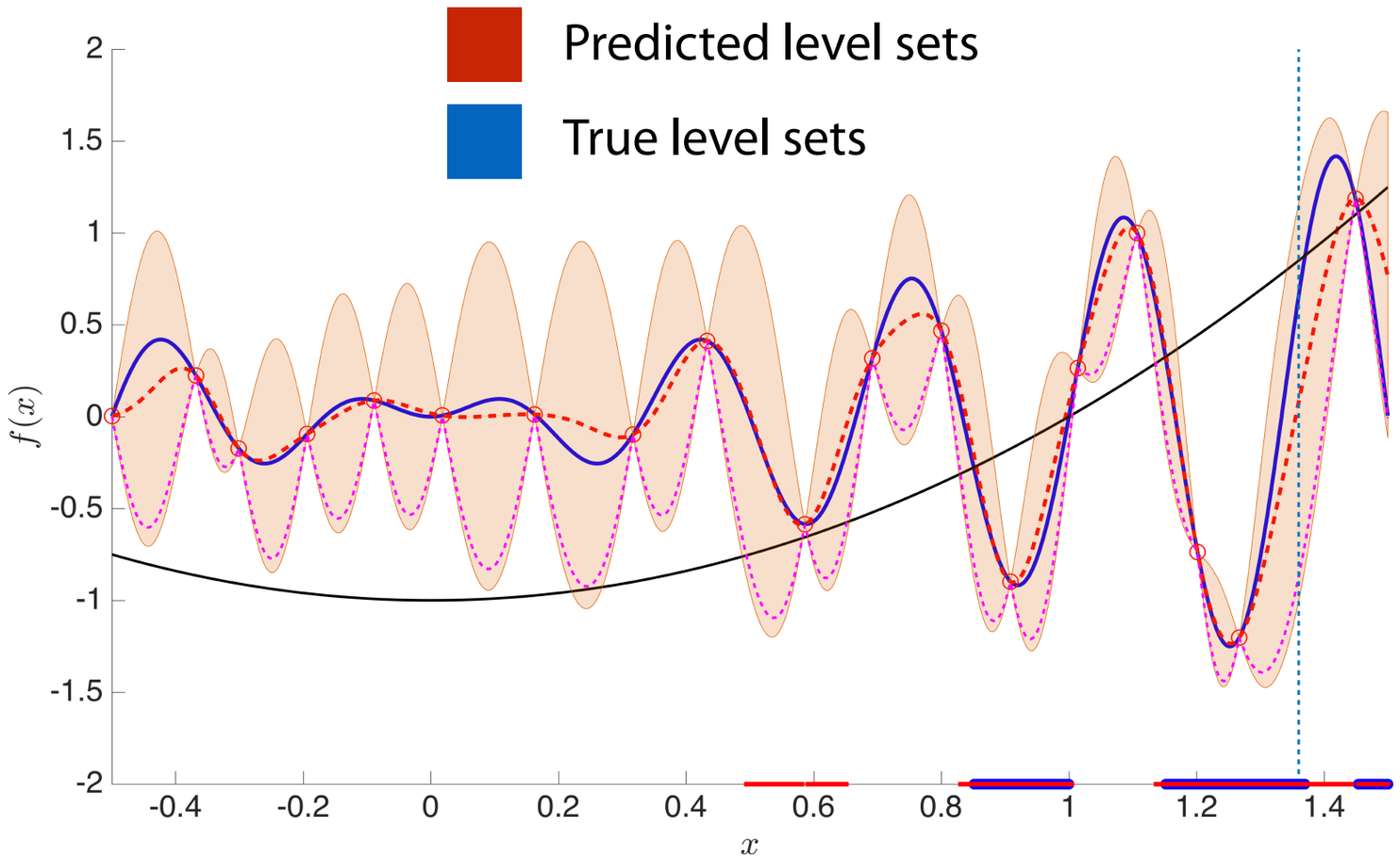
Goal: Identify the sets $L_\alpha(t) = \{\boldsymbol{x} : \mathcal{R}_\alpha(f(\boldsymbol{x})) \leq t\}$

Utilize the posterior to guide a sequential sampling policy by optimizing a chosen expected utility function

$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_\theta \mathbb{E}_{v \mid \mathbf{x},\theta}[U(\mathbf{x}, v, \theta)]$$

e.g. sample at the locations that maximize the posterior variance in L(t)

$$\boldsymbol{x}_{n+1} = \arg \max_{\boldsymbol{x} \in L_\alpha(t)} V(f(\boldsymbol{x}))$$



Terminate iteration when the "volume" of the predicted level sets is below a given threshold:

$$|V_{n+1}(t) - V_n(t)| < \epsilon, \qquad V_n(t) = \int_{L_\alpha(t)} \mathbf{1}_{[-\infty,t]} d\boldsymbol{x}$$

Remarks:

- The choice of risk-averseness level $\alpha \in [0, 1)$ controls the exploration vs exploitation trade-off.
- Upon convergence the predicted levels sets are guaranteed to be a subset of the true level sets.

P. Perdikaris, M. Raissi and J.O. Royset, "Risk-assesment, learning, and optimization using surrogate models", (in preparation), 2017.

Active learning of level sets

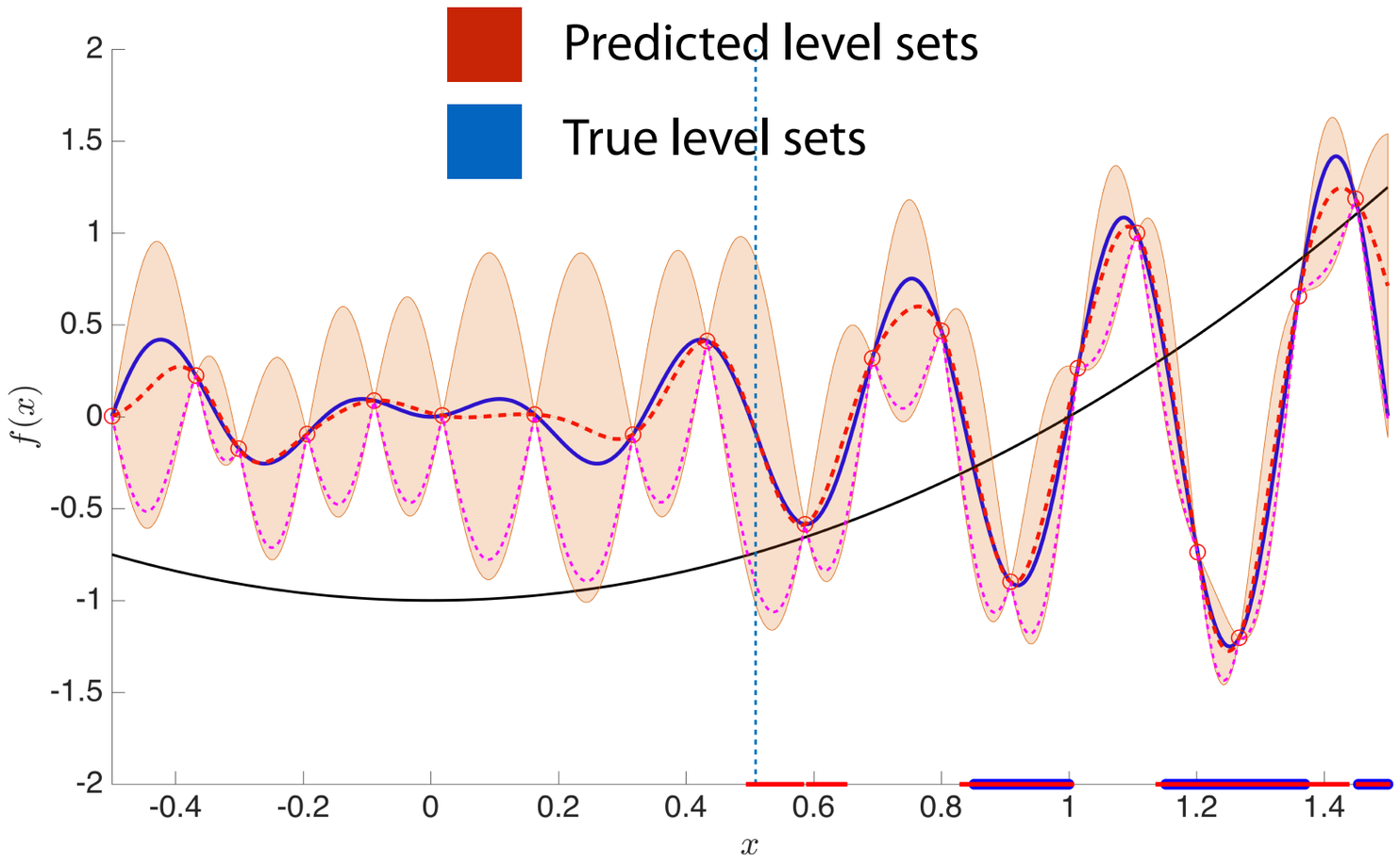
Goal: Identify the sets $L_\alpha(t) = \{\boldsymbol{x} : \mathcal{R}_\alpha(f(\boldsymbol{x})) \leq t\}$

Utilize the posterior to guide a sequential sampling policy by optimizing a chosen expected utility function

$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_\theta \mathbb{E}_{v \mid \mathbf{x},\theta}[U(\mathbf{x}, v, \theta)]$$

e.g. sample at the locations that maximize the posterior variance in L(t)

$$\boldsymbol{x}_{n+1} = \arg \max_{\boldsymbol{x} \in L_\alpha(t)} V(f(\boldsymbol{x}))$$



Terminate iteration when the "volume" of the predicted level sets is below a given threshold:

$$|V_{n+1}(t) - V_n(t)| < \epsilon, \qquad V_n(t) = \int_{L_\alpha(t)} \mathbf{1}_{[-\infty,t]} d\boldsymbol{x}$$

Remarks:

- The choice of risk-averseness level $\alpha \in [0, 1)$ controls the exploration vs exploitation trade-off.
- Upon convergence the predicted levels sets are guaranteed to be a subset of the true level sets.

P. Perdikaris, M. Raissi and J.O. Royset, "Risk-assesment, learning, and optimization using surrogate models", (in preparation), 2017.

Active learning of level sets

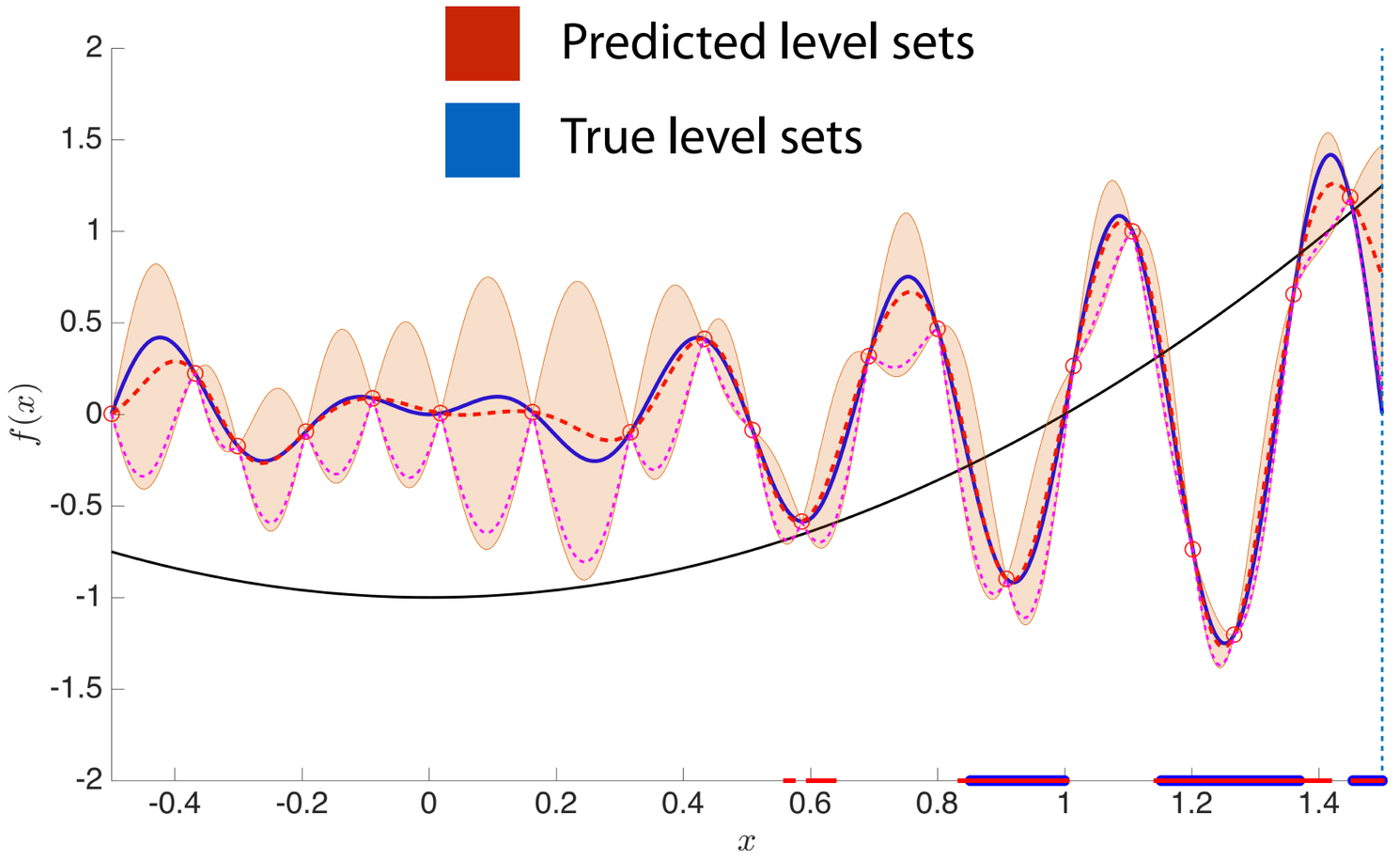
Goal: Identify the sets $L_\alpha(t) = \{\boldsymbol{x} : \mathcal{R}_\alpha(f(\boldsymbol{x})) \leq t\}$

Utilize the posterior to guide a sequential sampling policy by optimizing a chosen expected utility function

$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_\theta \mathbb{E}_{v \mid \mathbf{x},\theta}[U(\mathbf{x}, v, \theta)]$$

e.g. sample at the locations that maximize the posterior variance in L(t)

$$\boldsymbol{x}_{n+1} = \arg \max_{\boldsymbol{x} \in L_\alpha(t)} V(f(\boldsymbol{x}))$$



Terminate iteration when the "volume" of the predicted level sets is below a given threshold:

$$|V_{n+1}(t) - V_n(t)| < \epsilon, \qquad V_n(t) = \int_{L_\alpha(t)} \mathbf{1}_{[-\infty,t]} d\boldsymbol{x}$$

Remarks:

- The choice of risk-averseness level $\alpha \in [0, 1)$ controls the exploration vs exploitation trade-off.
- Upon convergence the predicted levels sets are guaranteed to be a subset of the true level sets.

P. Perdikaris, M. Raissi and J.O. Royset, "Risk-assesment, learning, and optimization using surrogate models", (in preparation), 2017.

Active learning of level sets

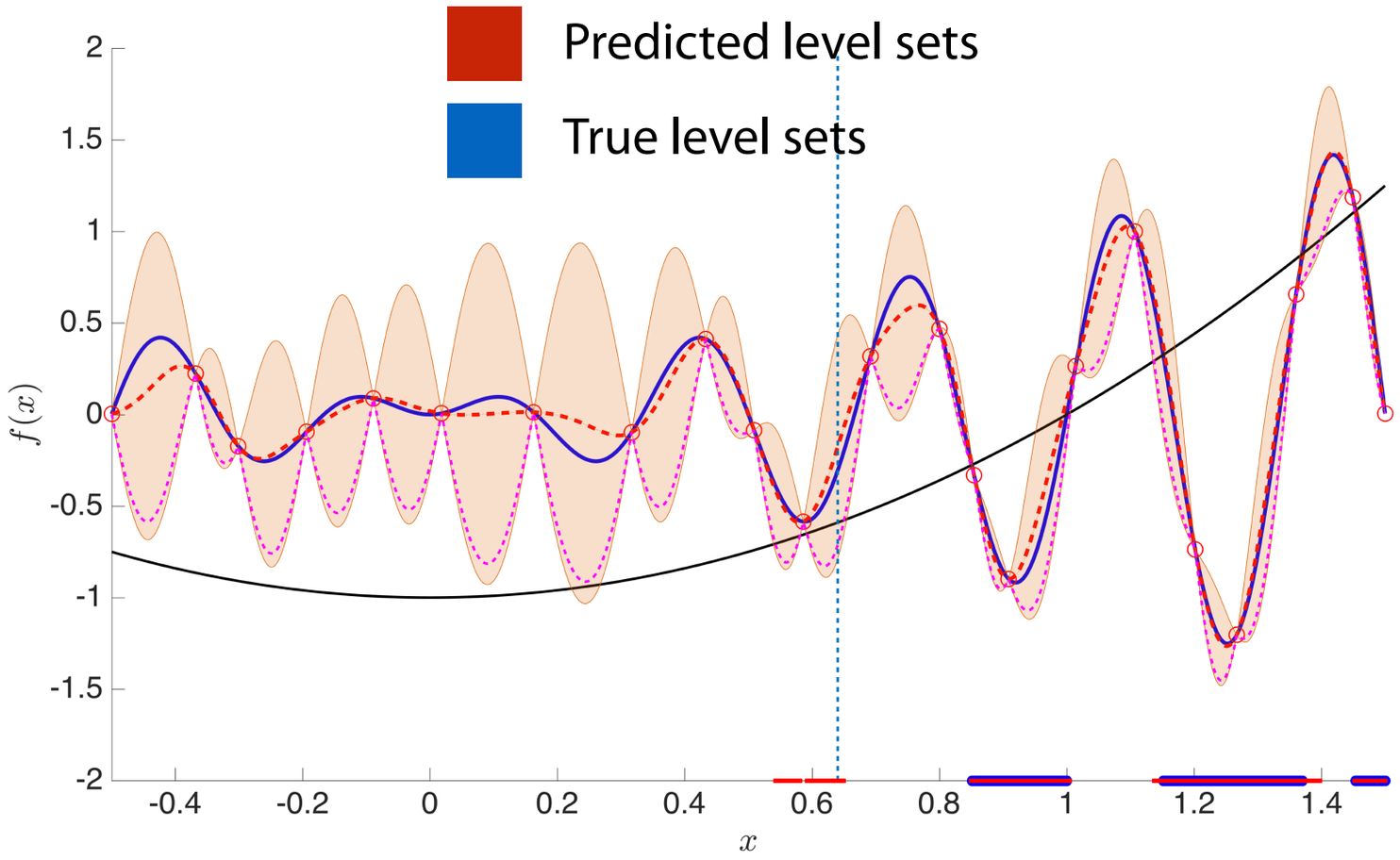
Goal: Identify the sets $L_\alpha(t) = \{\boldsymbol{x} : \mathcal{R}_\alpha(f(\boldsymbol{x})) \leq t\}$

Utilize the posterior to guide a sequential sampling policy by optimizing a chosen expected utility function

$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_\theta \mathbb{E}_{v \mid \mathbf{x}, \theta}[U(\mathbf{x}, v, \theta)]$$

e.g. sample at the locations that maximize the posterior variance in L(t)

$$\boldsymbol{x}_{n+1} = \arg \max_{\boldsymbol{x} \in L_\alpha(t)} V(f(\boldsymbol{x}))$$



Terminate iteration when the "volume" of the predicted level sets is below a given threshold:

$$|V_{n+1}(t) - V_n(t)| < \epsilon, \qquad V_n(t) = \int_{L_\alpha(t)} \mathbf{1}_{[-\infty, t]} d\boldsymbol{x}$$

Remarks:

- The choice of risk-averseness level $\alpha \in [0, 1)$ controls the exploration vs exploitation trade-off.
- Upon convergence the predicted levels sets are guaranteed to be a subset of the true level sets.

P. Perdikaris, M. Raissi and J.O. Royset, "Risk-assesment, learning, and optimization using surrogate models", (in preparation), 2017.

Active learning of level sets

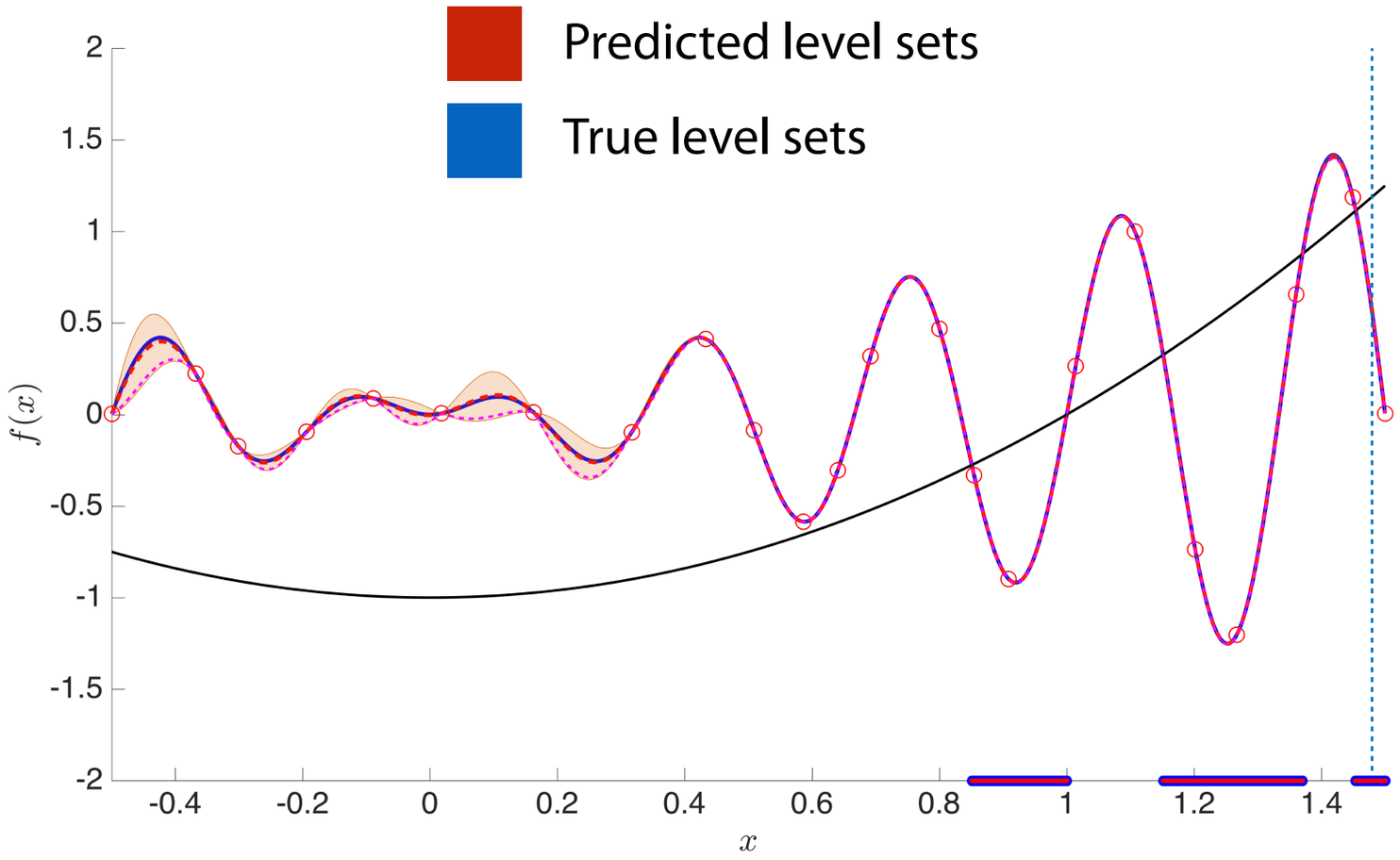
Goal: Identify the sets $L_\alpha(t) = \{\boldsymbol{x} : \mathcal{R}_\alpha(f(\boldsymbol{x})) \leq t\}$

Utilize the posterior to guide a sequential sampling policy by optimizing a chosen expected utility function

$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_\theta \mathbb{E}_{v \mid \mathbf{x}, \theta}[U(\mathbf{x}, v, \theta)]$$

e.g. sample at the locations that maximize the posterior variance in L(t)

$$\boldsymbol{x}_{n+1} = \arg \max_{\boldsymbol{x} \in L_\alpha(t)} V(f(\boldsymbol{x}))$$



Terminate iteration when the "volume" of the predicted level sets is below a given threshold:

$$|V_{n+1}(t) - V_n(t)| < \epsilon, \qquad V_n(t) = \int_{L_\alpha(t)} \mathbf{1}_{[-\infty, t]} d\boldsymbol{x}$$

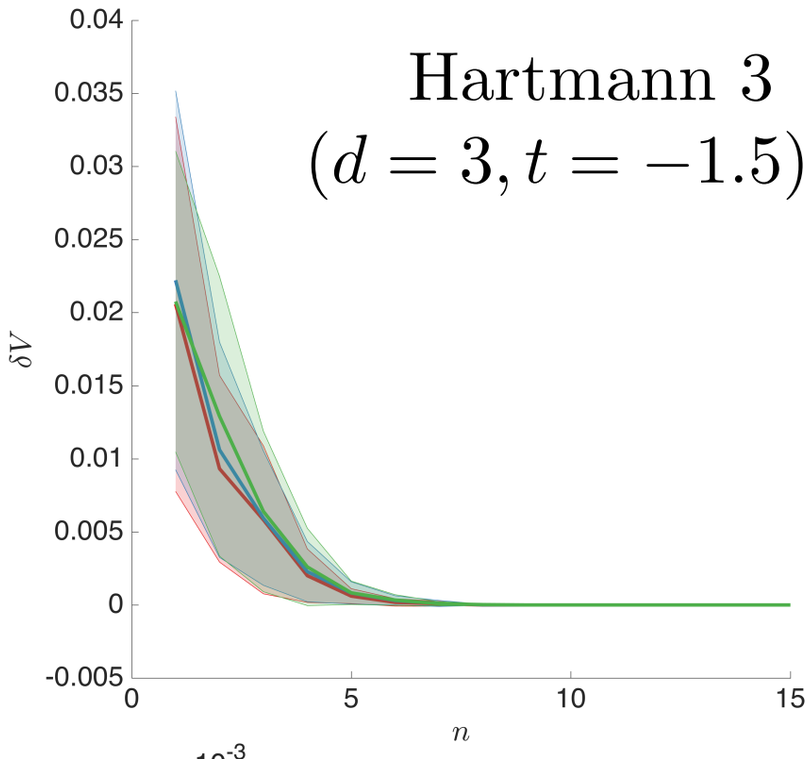
Remarks:

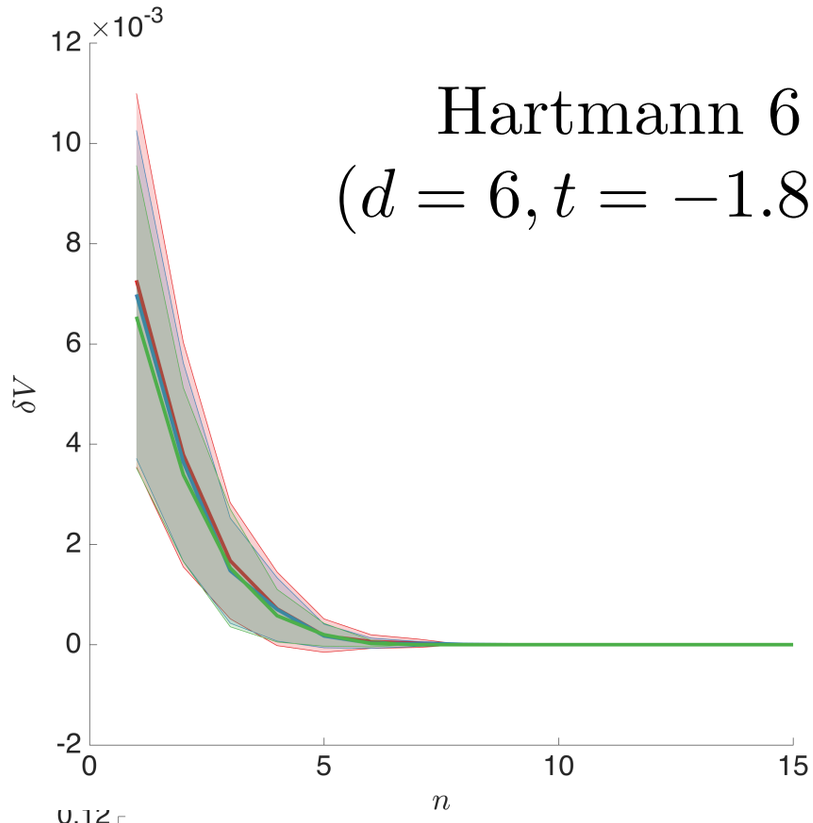
- The choice of risk-averseness level $\alpha \in [0, 1)$ controls the exploration vs exploitation trade-off.
- Upon convergence the predicted levels sets are guaranteed to be a subset of the true level sets.

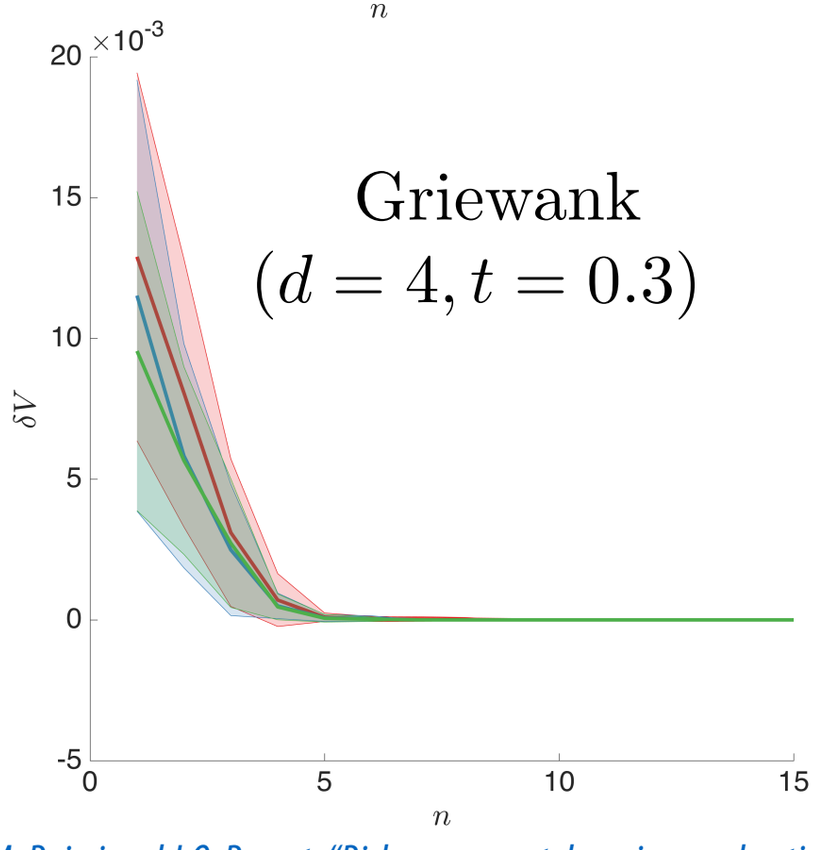
P. Perdikaris, M. Raissi and J.O. Royset, "Risk-assesment, learning, and optimization using surrogate models", (in preparation), 2017.

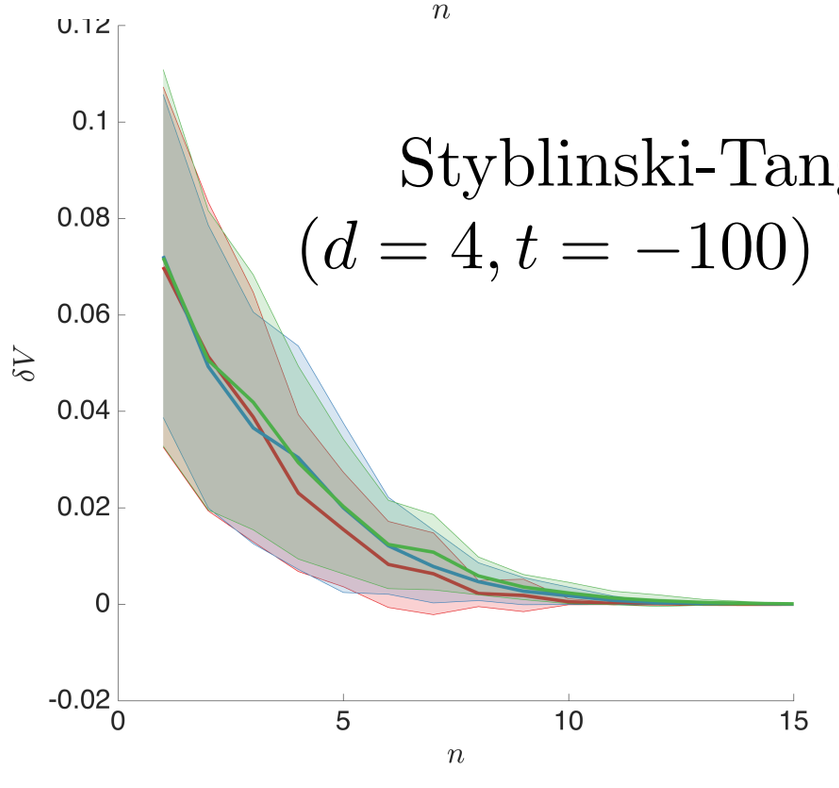
Active learning of level sets

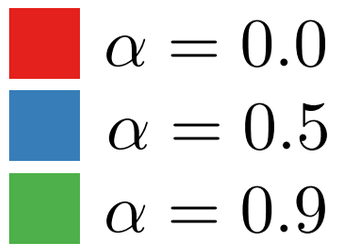
Change in volume after between two consecutive iterations: $\delta V = |V_{n+1} - V_n|$







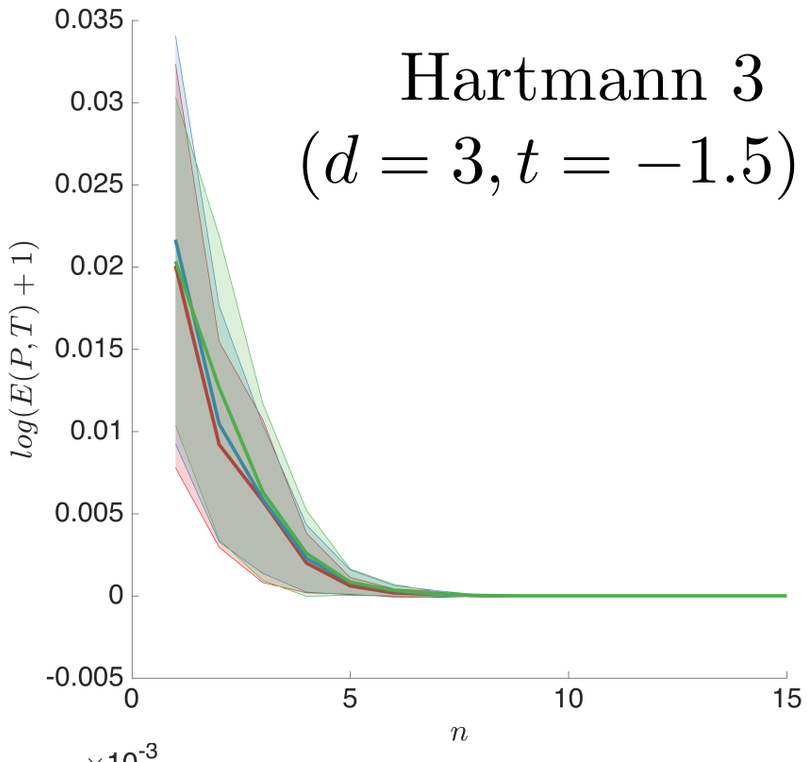


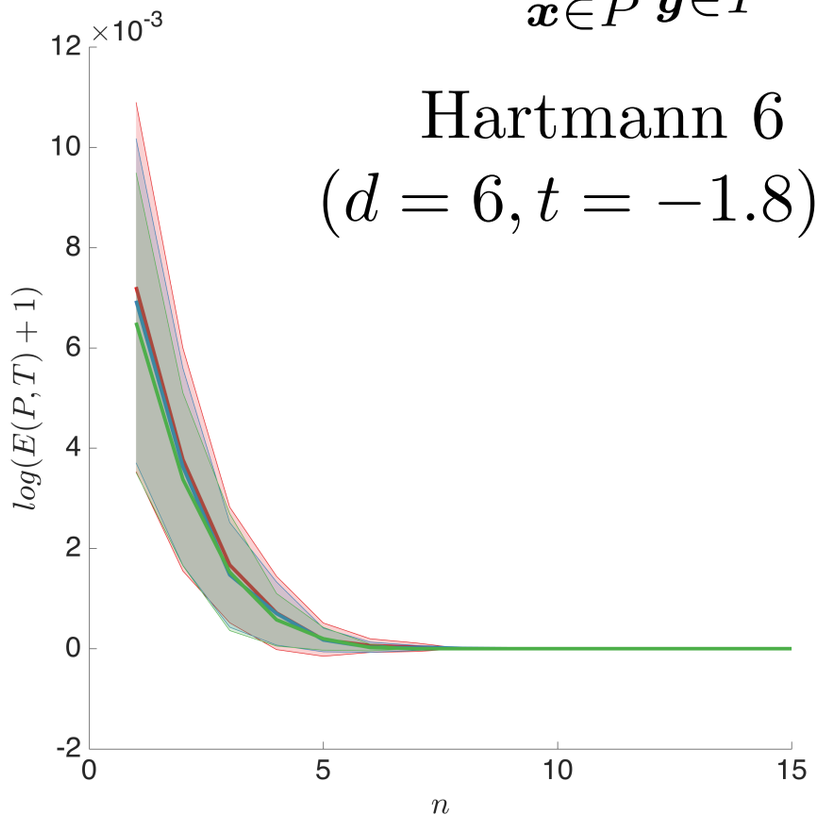


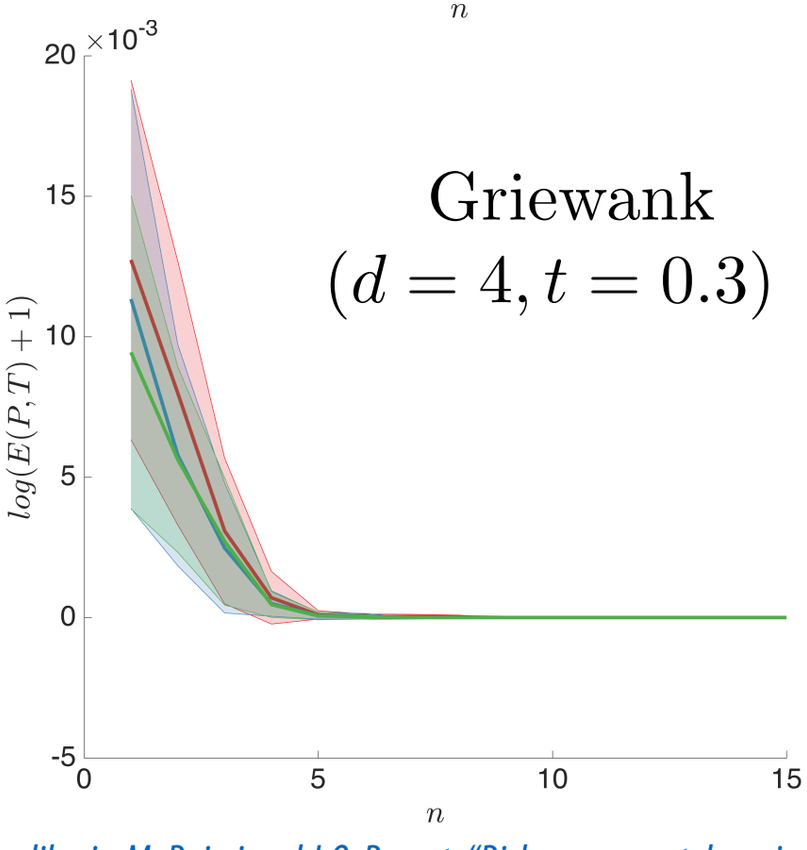
P. Perdikaris, M. Raissi and J.O. Royset, "Risk-assesment, learning, and optimization using surrogate models", (in preparation), 2017.

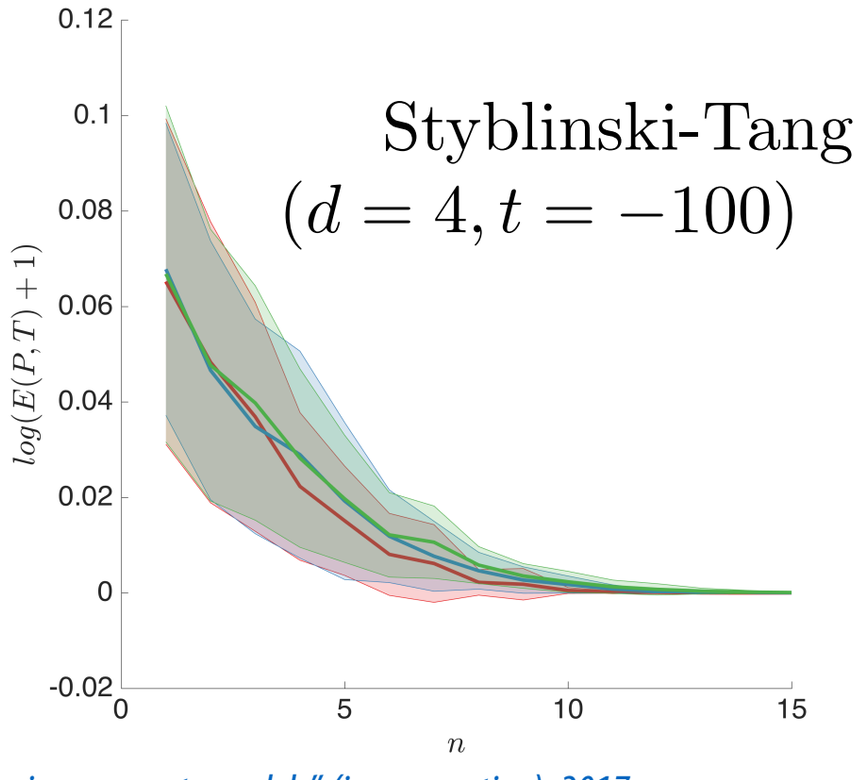
Active learning of level sets

Excess of predicted over the true level sets: $E(P,T) = \sup_{\boldsymbol{x}\in P} \inf_{\boldsymbol{y}\in T} d(x,y)$









$\alpha = 0.0$

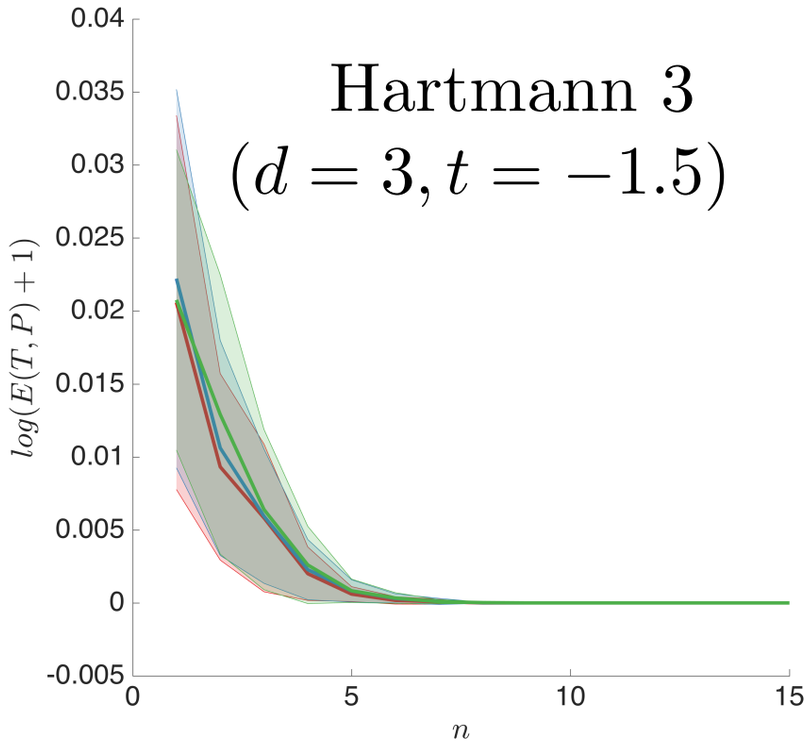
$\alpha = 0.5$

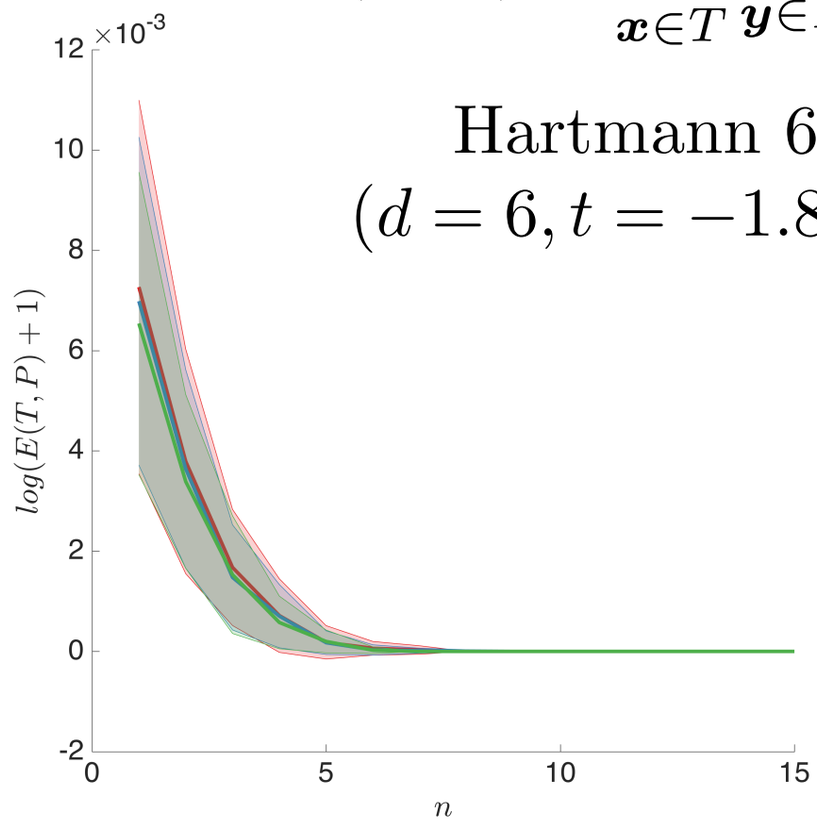
$\alpha = 0.9$

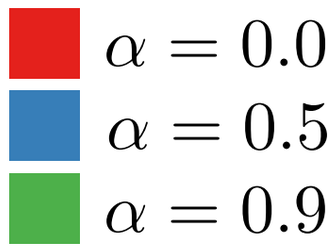
P. Perdikaris, M. Raissi and J.O. Royset, "Risk-assesment, learning, and optimization using surrogate models", (in preparation), 2017.

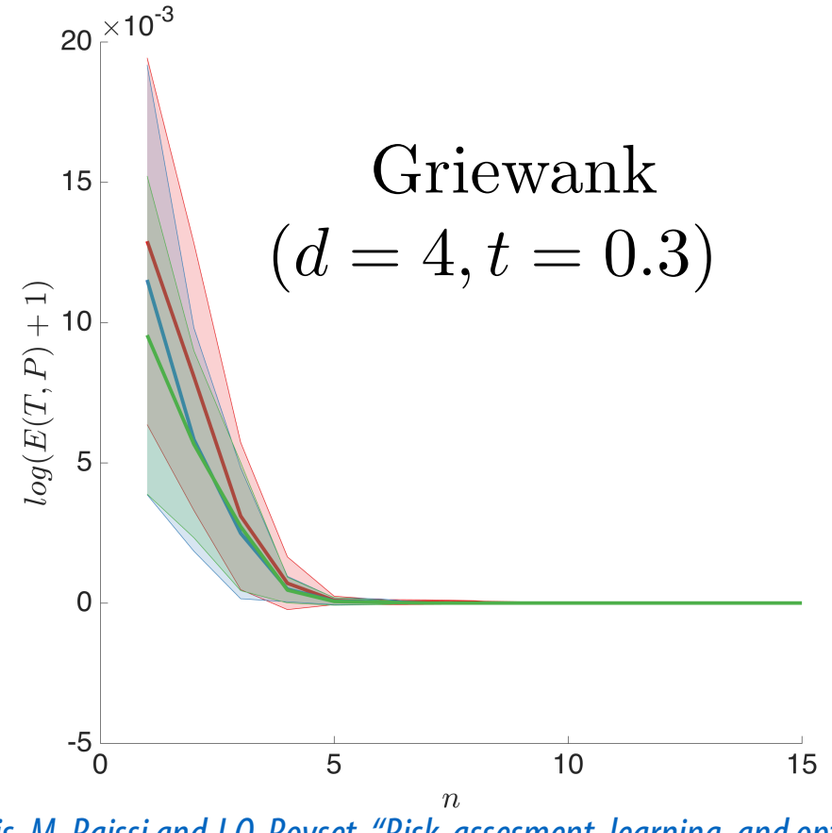
Active learning of level sets

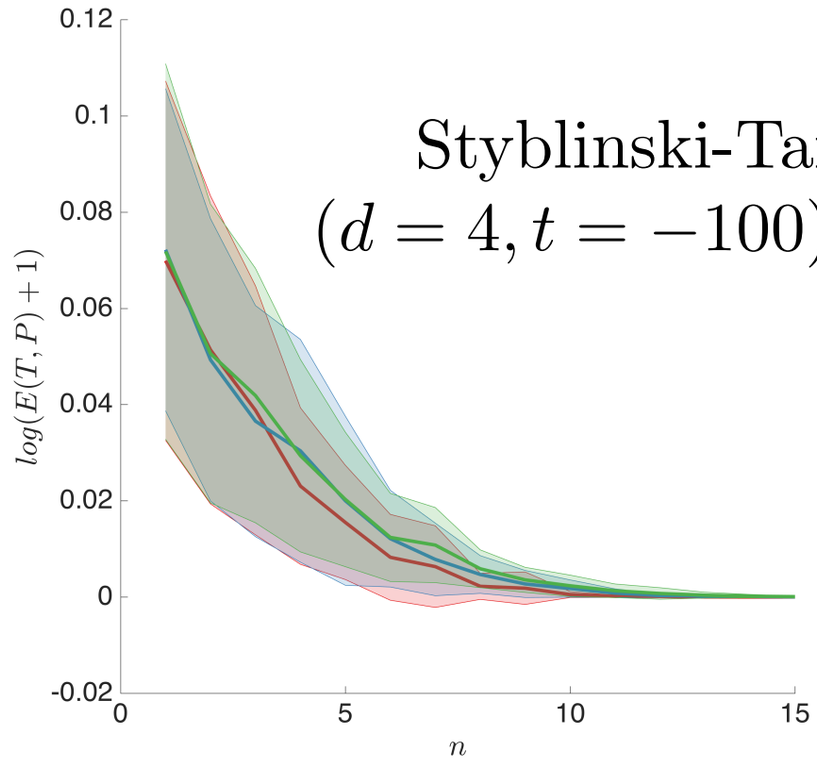
Excess of true over the predicted level sets: $E(T,P) = \sup_{\boldsymbol{x}\in T} \inf_{\boldsymbol{y}\in P} d(x,y)$











P. Perdikaris, M. Raissi and J.O. Royset, "Risk-assesment, learning, and optimization using surrogate models", (in preparation), 2017.

Active learning of level sets

Case \ α	0.0	0.5	0.9
Hartmann $(d = 3)$	6.7	7.3	8.1
Hartmann $(d = 6)$	6.2	6.2	6.5
Griewank $(d = 4)$	6.5	6.5	6.6
Styblinski-Tang $(d = 4)$	9.3	10.6	12.3

Table 1: Average # of iterations to convergence for 100 independent trials.

P. Perdikaris, M. Raissi and J.O. Royset, "Risk-assesment, learning, and optimization using surrogate models", (in preparation), 2017.

Bayesian optimization

Goal: Estimate the global minimum of a function: $\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathbb{R}^d} g(\mathbf{x})$ (potentially intractable)

Setup: g(x) is a black-box and expensive to evaluate objective function, noisy observations, no gradients.

Idea: Approximate g(x) using a GP surrogate: $y = f(\mathbf{x}) + \epsilon, \quad f \sim \mathcal{GP}\left(f|0, k(\mathbf{x}, \mathbf{x}'; \theta)\right)$

Utilize the posterior to guide a sequential or parallel sampling policy by optimizing a chosen expected utility function

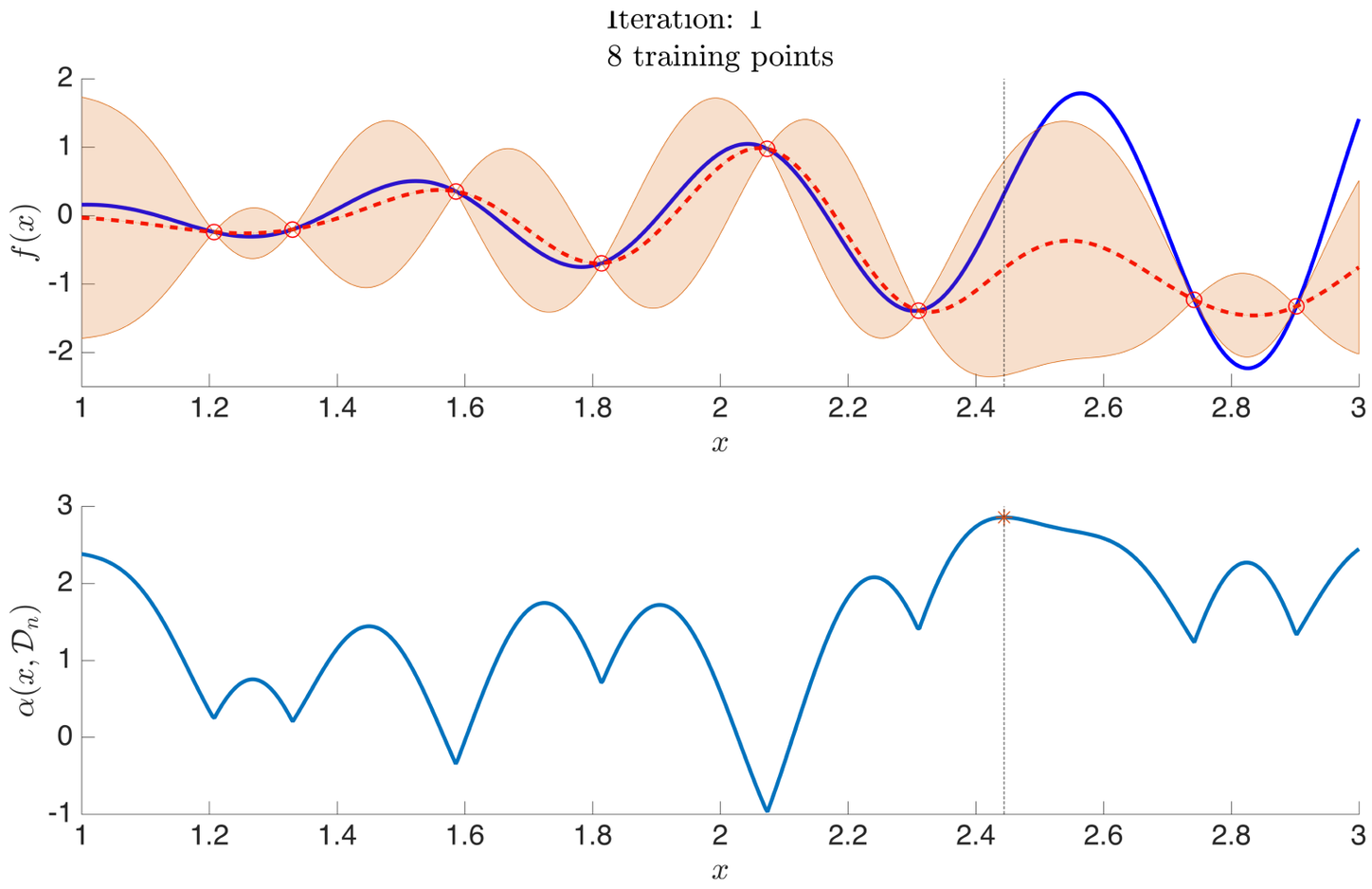
$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_\theta \mathbb{E}_{v\,|\,\mathbf{x},\theta}[U(\mathbf{x}, v, \theta)]$$

The optimization problem is transformed to:

$$\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}} \alpha(\mathbf{x}; \mathcal{D}_n)$$

Remark:

Acquisition functions aim to balance the trade-off between exploration and exploitation.



e.g. sample at the locations that minimize the lower superquintile risk confidence bound

$$\boldsymbol{x}_{n+1} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} \mu(\boldsymbol{x}) - \frac{\phi(\Phi^{-1}(\alpha))}{1-\alpha}\sigma(\boldsymbol{x})$$

P. Perdikaris, M. Raissi and J.O. Royset, "Risk-assesment, learning, and optimization using surrogate models", (in preparation), 2017.

Bayesian optimization

Goal: Estimate the global minimum of a function: $\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathbb{R}^d} g(\mathbf{x})$ (potentially intractable)

Setup: g(x) is a black-box and expensive to evaluate objective function, noisy observations, no gradients.

Idea: Approximate g(x) using a GP surrogate: $y = f(\mathbf{x}) + \epsilon, \quad f \sim \mathcal{GP}\left(f|0, k(\mathbf{x}, \mathbf{x}'; \theta)\right)$

Utilize the posterior to guide a sequential or parallel sampling policy by optimizing a chosen expected utility function

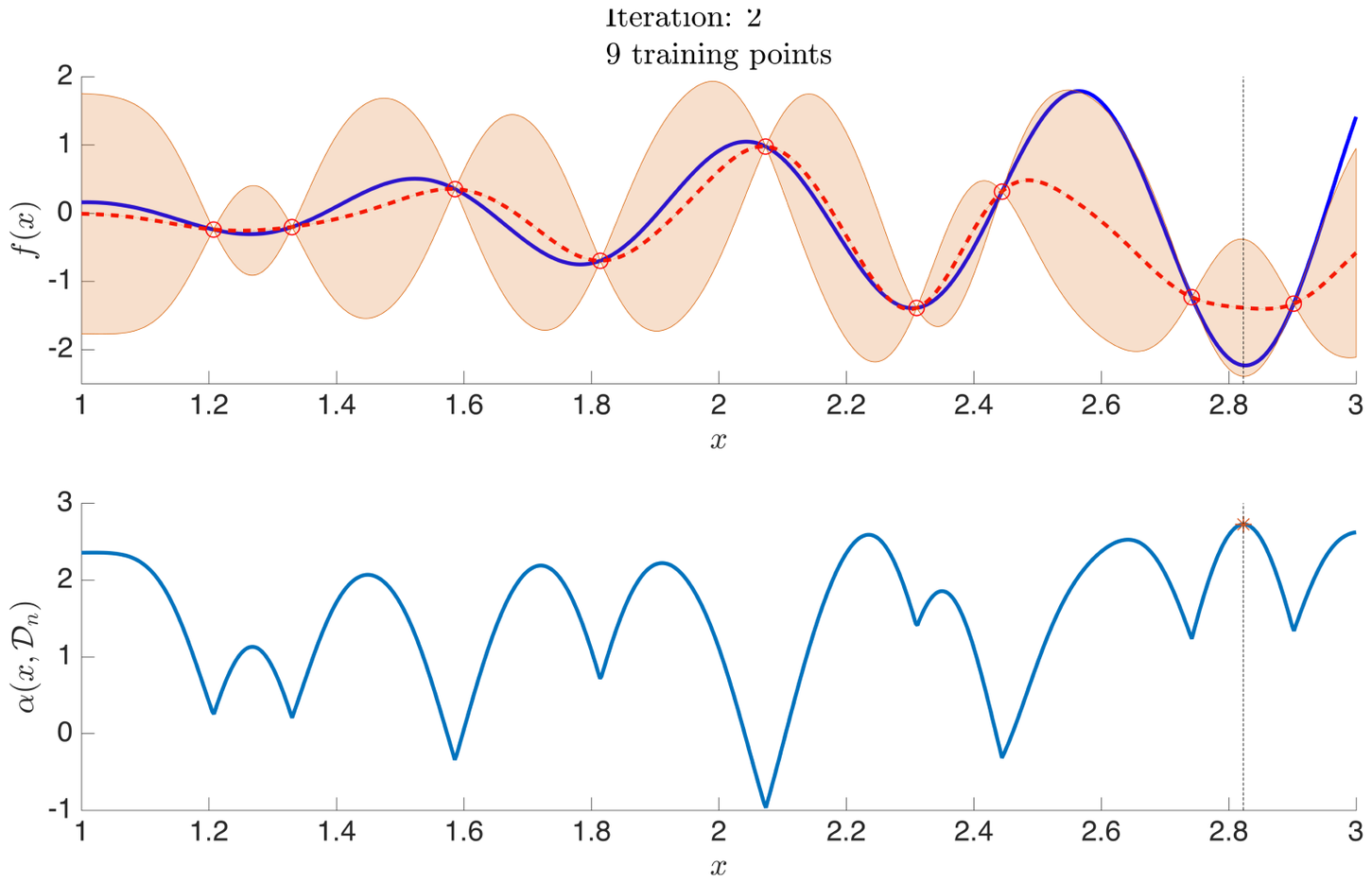
$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_\theta \mathbb{E}_{v\,|\,\mathbf{x},\theta}[U(\mathbf{x}, v, \theta)]$$

The optimization problem is transformed to:

$$\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}} \alpha(\mathbf{x}; \mathcal{D}_n)$$

Remark:

Acquisition functions aim to balance the trade-off between exploration and exploitation.



e.g. sample at the locations that minimize the lower superquintile risk confidence bound

$$\boldsymbol{x}_{n+1} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} \mu(\boldsymbol{x}) - \frac{\phi(\Phi^{-1}(\alpha))}{1-\alpha}\sigma(\boldsymbol{x})$$

P. Perdikaris, M. Raissi and J.O. Royset, "Risk-assesment, learning, and optimization using surrogate models", (in preparation), 2017.

Bayesian optimization

Goal: Estimate the global minimum of a function: $\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathbb{R}^d} g(\mathbf{x})$ (potentially intractable)

Setup: g(x) is a black-box and expensive to evaluate objective function, noisy observations, no gradients.

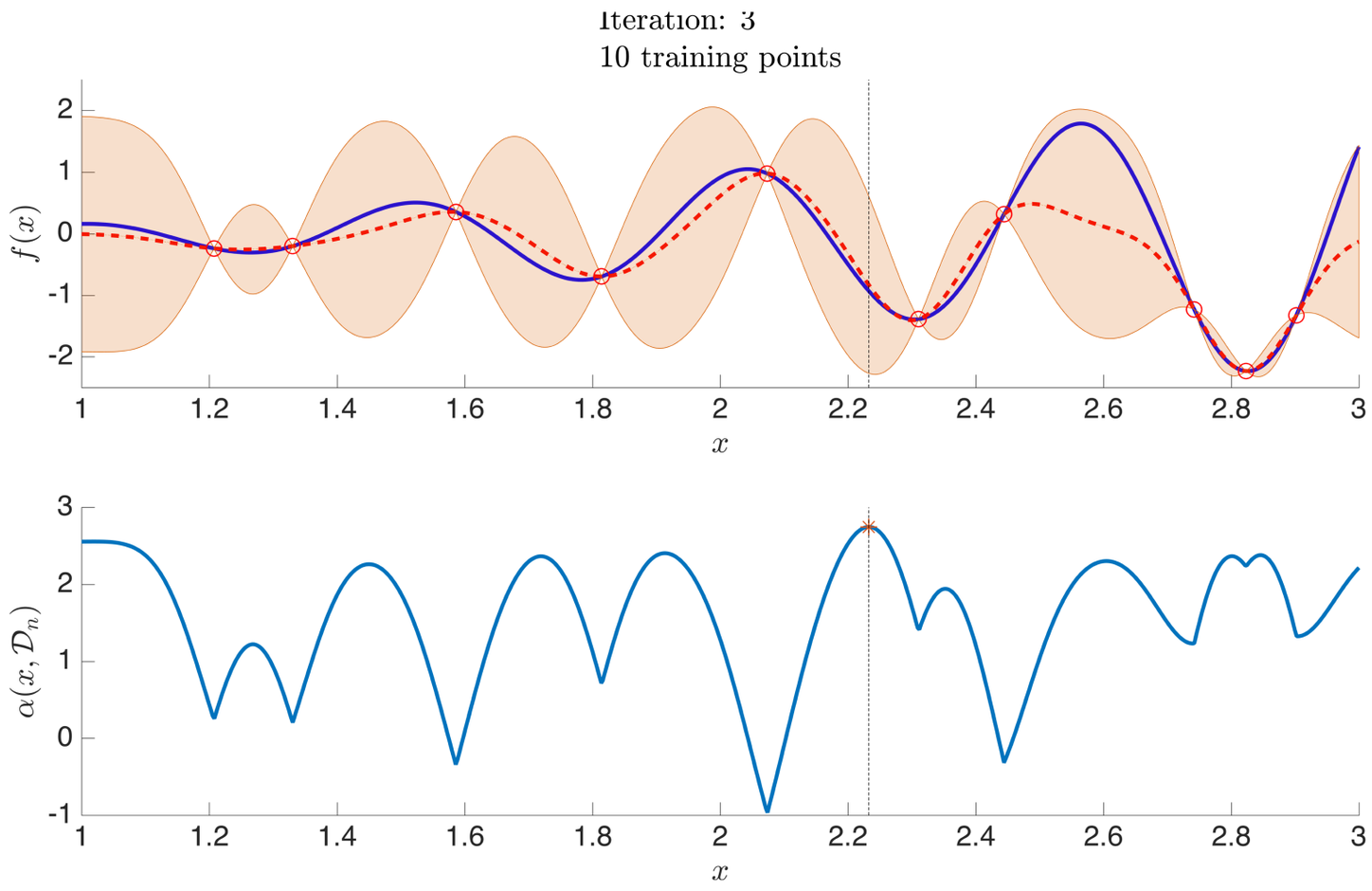
Idea: Approximate g(x) using a GP surrogate: $y = f(\mathbf{x}) + \epsilon, \quad f \sim \mathcal{GP}\left(f|0, k(\mathbf{x}, \mathbf{x}'; \theta)\right)$

Utilize the posterior to guide a sequential or parallel sampling policy by optimizing a chosen expected utility function

$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_\theta \mathbb{E}_{v\,|\,\mathbf{x},\theta}[U(\mathbf{x}, v, \theta)]$$

The optimization problem is transformed to:

$$\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}} \alpha(\mathbf{x}; \mathcal{D}_n)$$



Remark:

Acquisition functions aim to balance the trade-off between exploration and exploitation.

e.g. sample at the locations that minimize the lower superquintile risk confidence bound

$$\boldsymbol{x}_{n+1} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} \mu(\boldsymbol{x}) - \frac{\phi(\Phi^{-1}(\alpha))}{1-\alpha}\sigma(\boldsymbol{x})$$

P. Perdikaris, M. Raissi and J.O. Royset, "Risk-assesment, learning, and optimization using surrogate models", (in preparation), 2017.

Bayesian optimization

Goal: Estimate the global minimum of a function: $\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathbb{R}^d} g(\mathbf{x})$ (potentially intractable)

Setup: g(x) is a black-box and expensive to evaluate objective function, noisy observations, no gradients.

Idea: Approximate g(x) using a GP surrogate: $y = f(\mathbf{x}) + \epsilon, \quad f \sim \mathcal{GP}\left(f|0, k(\mathbf{x}, \mathbf{x}'; \theta)\right)$

Utilize the posterior to guide a sequential or parallel sampling policy by optimizing a chosen expected utility function

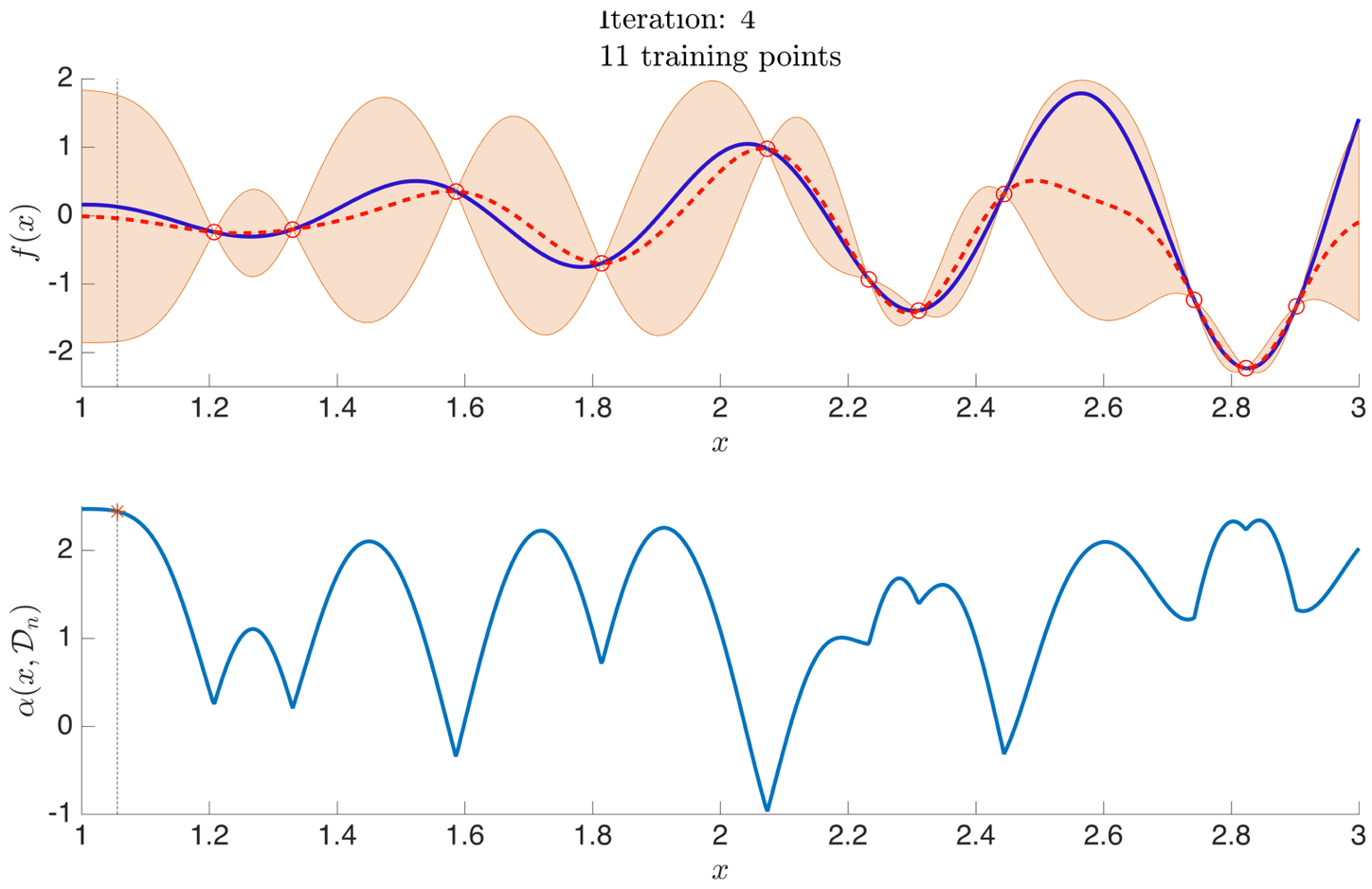
$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_\theta \mathbb{E}_{v\,|\,\mathbf{x},\theta}[U(\mathbf{x}, v, \theta)]$$

The optimization problem is transformed to:

$$\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}} \alpha(\mathbf{x}; \mathcal{D}_n)$$

Remark:

Acquisition functions aim to balance the trade-off between exploration and exploitation.



e.g. sample at the locations that minimize the lower superquintile risk confidence bound

$$\boldsymbol{x}_{n+1} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} \mu(\boldsymbol{x}) - \frac{\phi(\Phi^{-1}(\alpha))}{1-\alpha}\sigma(\boldsymbol{x})$$

P. Perdikaris, M. Raissi and J.O. Royset, "Risk-assesment, learning, and optimization using surrogate models", (in preparation), 2017.

Bayesian optimization

Goal: Estimate the global minimum of a function: $\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathbb{R}^d} g(\mathbf{x})$ (potentially intractable)

Setup: g(x) is a black-box and expensive to evaluate objective function, noisy observations, no gradients.

Idea: Approximate g(x) using a GP surrogate: $y = f(\mathbf{x}) + \epsilon, \quad f \sim \mathcal{GP}\left(f|0, k(\mathbf{x}, \mathbf{x}'; \theta)\right)$

Utilize the posterior to guide a sequential or parallel sampling policy by optimizing a chosen expected utility function

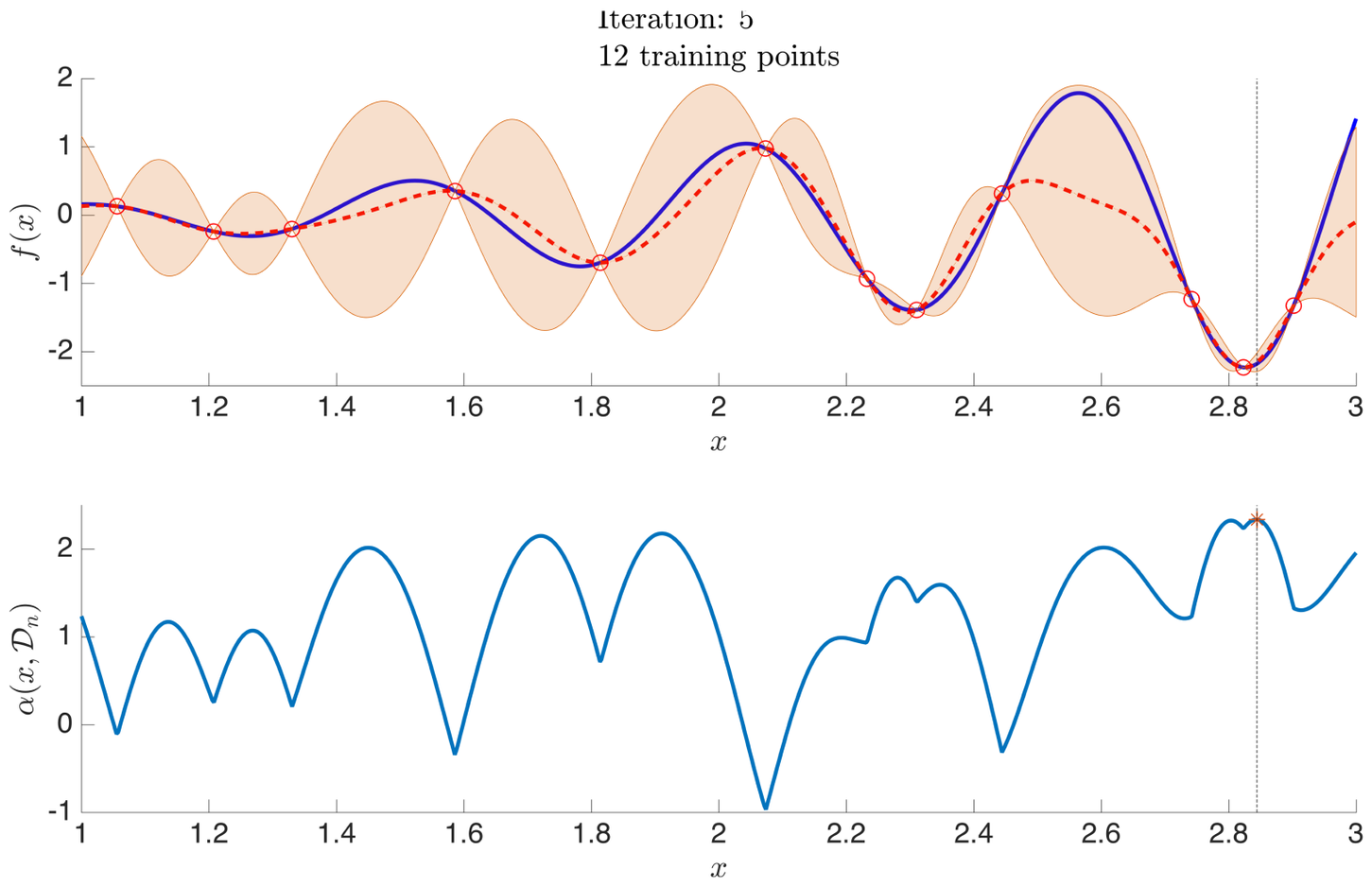
$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_\theta \mathbb{E}_{v\,|\,\mathbf{x},\theta}[U(\mathbf{x}, v, \theta)]$$

The optimization problem is transformed to:

$$\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}} \alpha(\mathbf{x}; \mathcal{D}_n)$$

Remark:

Acquisition functions aim to balance the trade-off between exploration and exploitation.



e.g. sample at the locations that minimize the lower superquintile risk confidence bound

$$\boldsymbol{x}_{n+1} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} \mu(\boldsymbol{x}) - \frac{\phi(\Phi^{-1}(\alpha))}{1-\alpha}\sigma(\boldsymbol{x})$$

P. Perdikaris, M. Raissi and J.O. Royset, "Risk-assesment, learning, and optimization using surrogate models", (in preparation), 2017.

Bayesian optimization

Goal: Estimate the global minimum of a function: $\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathbb{R}^d} g(\mathbf{x})$ (potentially intractable)

Setup: g(x) is a black-box and expensive to evaluate objective function, noisy observations, no gradients.

Idea: Approximate g(x) using a GP surrogate: $y = f(\mathbf{x}) + \epsilon, \quad f \sim \mathcal{GP}\left(f|0, k(\mathbf{x}, \mathbf{x}'; \theta)\right)$

Utilize the posterior to guide a sequential or parallel sampling policy by optimizing a chosen expected utility function

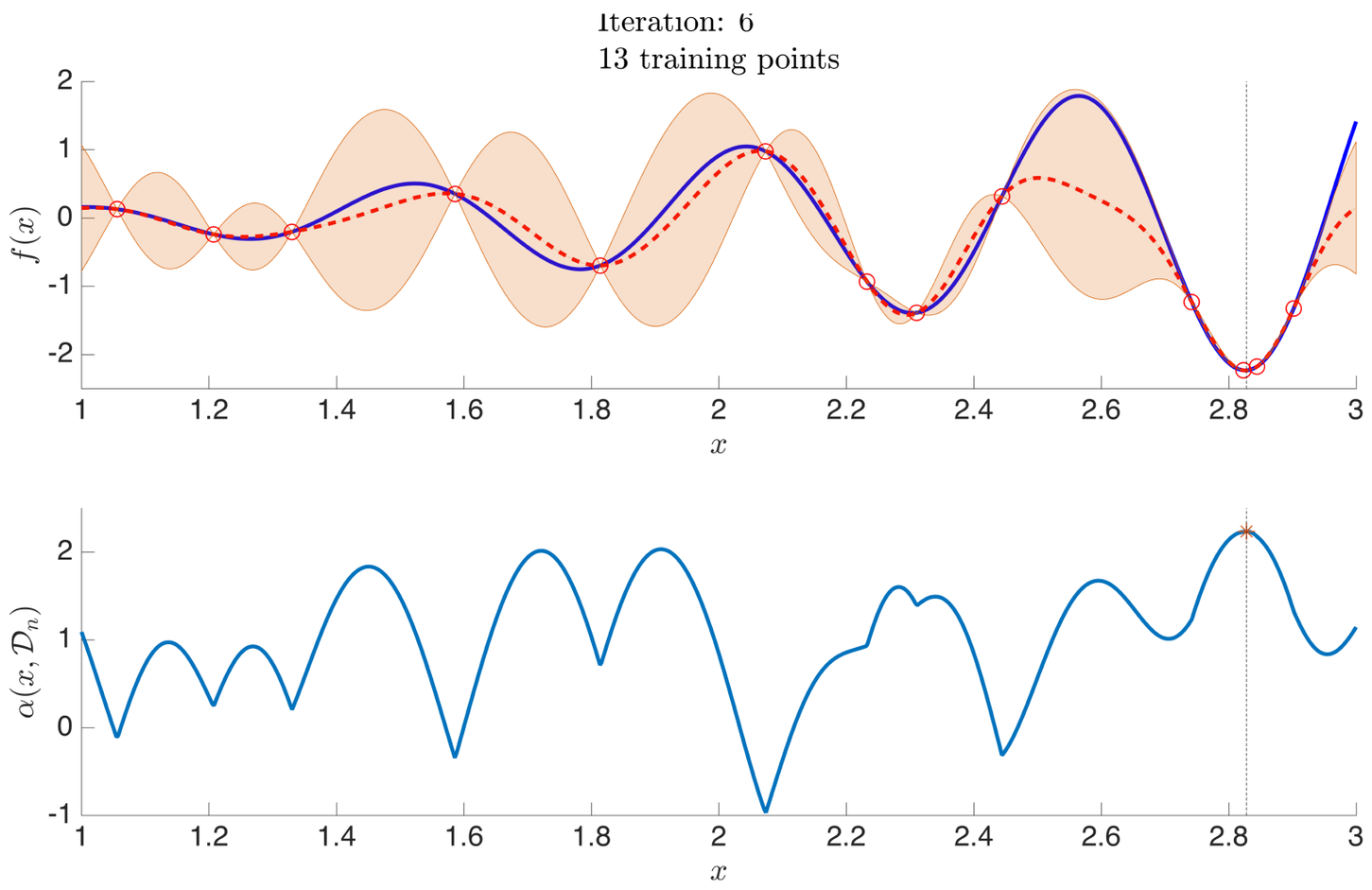
$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_\theta \mathbb{E}_{v\,|\,\mathbf{x},\theta}[U(\mathbf{x}, v, \theta)]$$

The optimization problem is transformed to:

$$\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}} \alpha(\mathbf{x}; \mathcal{D}_n)$$

Remark:

Acquisition functions aim to balance the trade-off between exploration and exploitation.



e.g. sample at the locations that minimize the lower superquintile risk confidence bound

$$\boldsymbol{x}_{n+1} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} \mu(\boldsymbol{x}) - \frac{\phi(\Phi^{-1}(\alpha))}{1-\alpha}\sigma(\boldsymbol{x})$$

P. Perdikaris, M. Raissi and J.O. Royset, "Risk-assesment, learning, and optimization using surrogate models", (in preparation), 2017.

Bayesian optimization

Goal: Estimate the global minimum of a function: $\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathbb{R}^d} g(\mathbf{x})$ (potentially intractable)

Setup: g(x) is a black-box and expensive to evaluate objective function, noisy observations, no gradients.

Idea: Approximate g(x) using a GP surrogate: $y = f(\mathbf{x}) + \epsilon, \quad f \sim \mathcal{GP}\left(f|0, k(\mathbf{x}, \mathbf{x}'; \theta)\right)$

Utilize the posterior to guide a sequential or parallel sampling policy by optimizing a chosen expected utility function

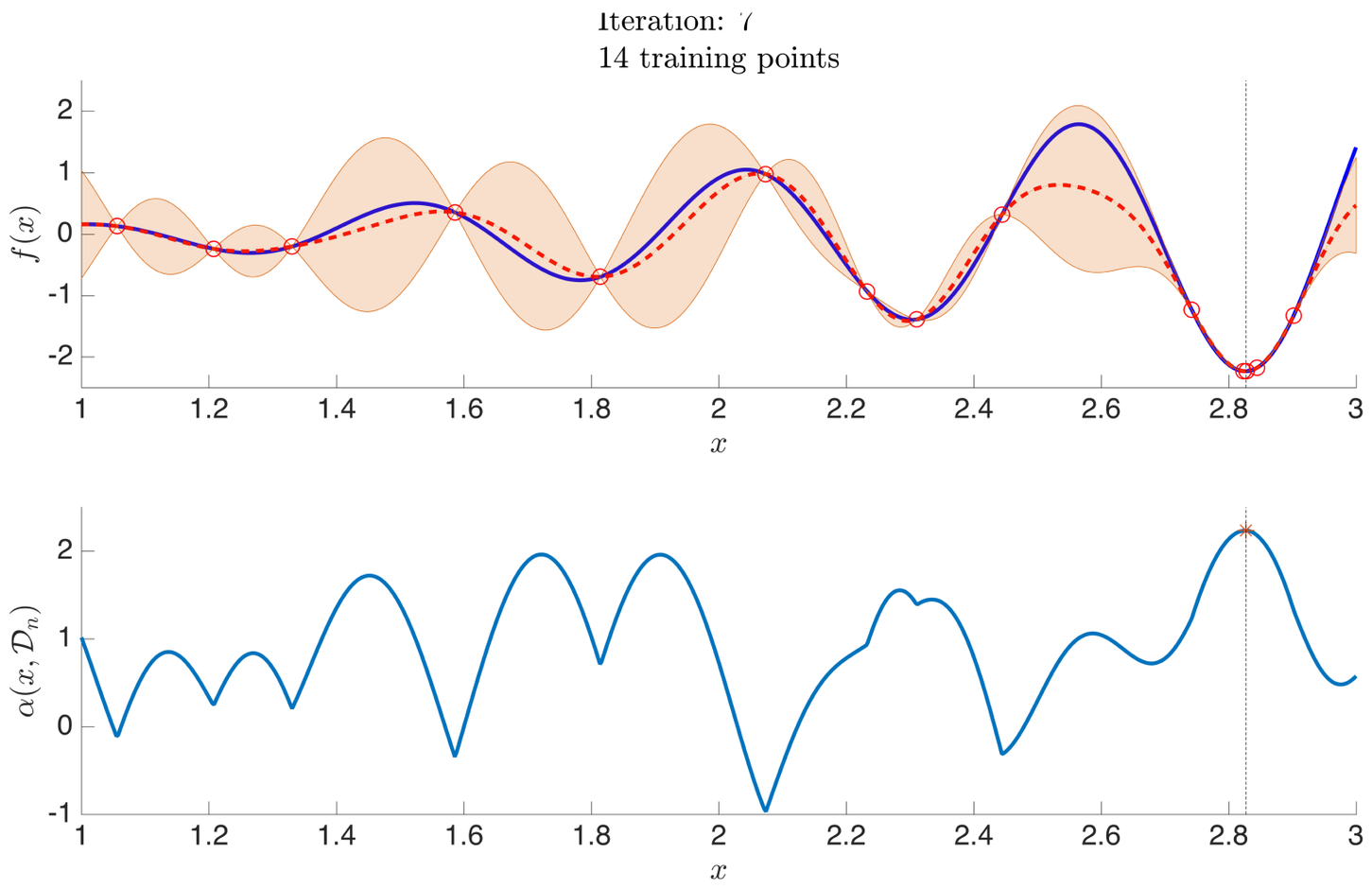
$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_\theta \mathbb{E}_{v\,|\,\mathbf{x},\theta}[U(\mathbf{x}, v, \theta)]$$

The optimization problem is transformed to:

$$\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}} \alpha(\mathbf{x}; \mathcal{D}_n)$$

Remark:

Acquisition functions aim to balance the trade-off between exploration and exploitation.



e.g. sample at the locations that minimize the lower superquintile risk confidence bound

$$\boldsymbol{x}_{n+1} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} \mu(\boldsymbol{x}) - \frac{\phi(\Phi^{-1}(\alpha))}{1-\alpha}\sigma(\boldsymbol{x})$$

P. Perdikaris, M. Raissi and J.O. Royset, "Risk-assesment, learning, and optimization using surrogate models", (in preparation), 2017.

Summary

Web: http://web.mit.edu/parisp/www/ ***Email:*** parisp@mit.edu

Questions?

Funding:

This work received support from DARPA grant HR0011- 14-1-0060

Collaborators:

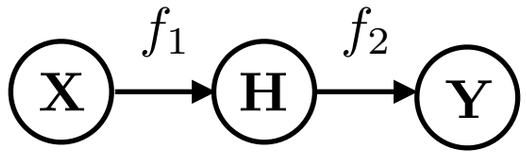
Maziar Raissi (Brown)
Johannes Royset (NPS)

Web: http://web.mit.edu/parisp/www/

Email: parisp@mit.edu

Challenges and limitations

Discontinuities and non-stationarity: GPs struggle with discontinuous data



Use warping functions to transform into a jointly stationary input space

- Log, sigmoid, betaCDF —> "*Warped GPs*" *Snelson, E., C.E. Rasmussen, and Z.Ghahramani. "Warped gaussian processes."*
- Neural networks —> "*Manifold GPs*" *Calandra, R., et al. "Manifold Gaussian processes for regression."*
- Gaussian processes —> "*Deep GPs*" *Damianou, A. C., and N.D. Lawrence. "Deep Gaussian processes."*

Theoretical guarantees: Accuracy, convergence rates, posterior consistency, contraction rates, etc.

Approximation theory in Reproducing Kernel Hilbert Spaces

Stuart, A.M., and A.L. Teckentrup. "Posterior consistency for Gaussian process approximations of Bayesian posterior distributions." arXiv preprint, 2016

Scalability: GPs suffer from a cubic scaling with the data

Low-rank approximations to the covariance

Snelson, E., and Z. Ghahramani. "Sparse Gaussian processes using pseudo-inputs."

Frequency-domain learning algorithms

Perdikaris P., D. Venturi, G.E. Karniadakis "Multi-fidelity information fusion algorithms for high dimensional systems and massive data-sets", SIAM J. Sci. Comput., 2016

Stochastic variational inference

Hensman, J., N. Fusi, and N.D. Lawrence. "Gaussian processes for big data."

High-dimensions: Tensor product kernels suffer from the curse of dimensionality, i.e. the require an exponentially increasing amount of training data

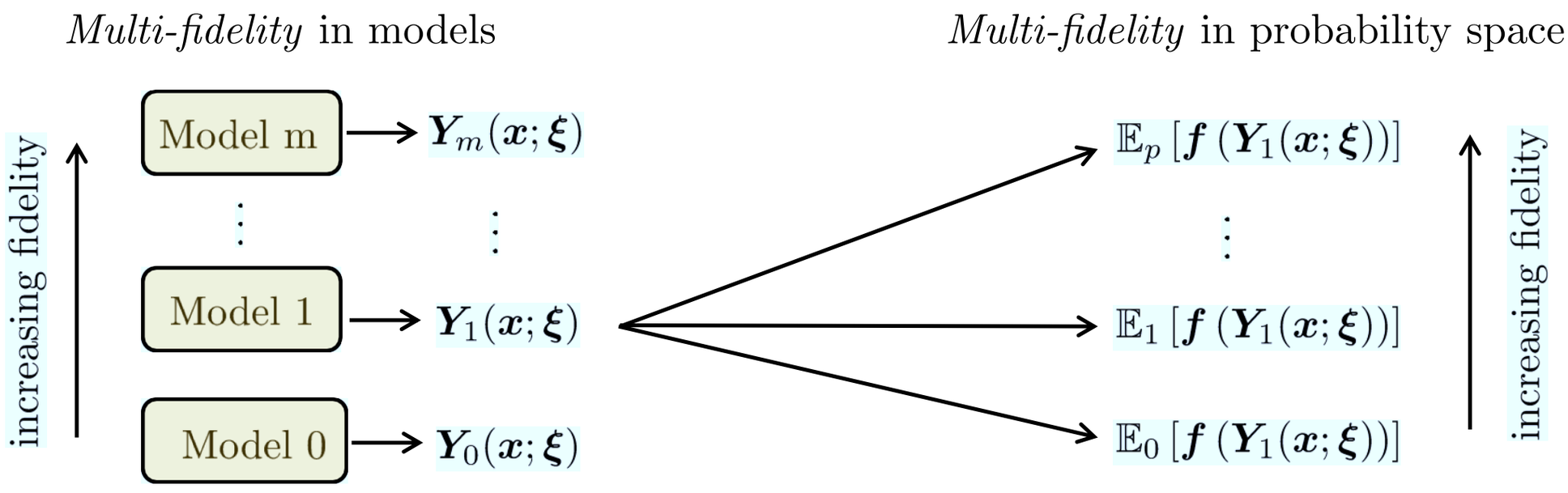
Data-driven additive kernels

Perdikaris P., D. Venturi, G.E. Karniadakis "Multi-fidelity information fusion algorithms for high dimensional systems and massive data-sets", SIAM J. Sci. Comput., 2016

Unsupervised dimensionality-reduction (*GPLVM*, *variational auto-encoders*)

Lawrence, N.D. "Gaussian process latent variable models for visualisation of high dimensional data."

Multi-fidelity in physical models and in probability space



PROCEEDINGS OF THE ROYAL SOCIETY A

rspa.royalsocietypublishing.org

Research

Article submitted to journal

Multi-fidelity modeling via recursive co-kriging and Gaussian Markov random fields

P. Perdikaris[1], D. Venturi[1], J.O. Royset[2], and G.E. Karniadakis[1]

[1]Division of Applied Mathematics, Brown University, Providence, RI 02912, USA

[2]Operations Research Department, Naval Postgraduate School, Monterey, CA 93943, USA

$$\mathbb{E}_{k+1}[f(\mathbf{Y}_l(\mathbf{x};\xi))] = \rho_{k+1}\mathbb{E}_k[f(\mathbf{Y}_l(\mathbf{x};\xi))] + \delta_{k+1}(\mathbf{x}), \quad k \leq p, \quad l \leq m$$

$$\begin{pmatrix} \mathbb{E}_1\left[\boldsymbol{f}\left(\boldsymbol{Y}_1\right)\right] & \mathbb{E}_1\left[\boldsymbol{f}\left(\boldsymbol{Y}_2\right)\right] & \cdots & \mathbb{E}_1\left[\boldsymbol{f}\left(\boldsymbol{Y}_m\right)\right] \\ \mathbb{E}_2\left[\boldsymbol{f}\left(\boldsymbol{Y}_1\right)\right] & \mathbb{E}_2\left[\boldsymbol{f}\left(\boldsymbol{Y}_2\right)\right] & \cdots & \mathbb{E}_2\left[\boldsymbol{f}\left(\boldsymbol{Y}_m\right)\right] \\ \vdots & & \ddots & \vdots \\ \mathbb{E}_p\left[\boldsymbol{f}\left(\boldsymbol{Y}_1\right)\right] & \mathbb{E}_p\left[\boldsymbol{f}\left(\boldsymbol{Y}_2\right)\right] & \cdots & \mathbb{E}_p\left[\boldsymbol{f}\left(\boldsymbol{Y}_m\right)\right] \end{pmatrix}$$

Fidelity in probability space

Fidelity in physical models

Importance of the prior

The choice of the covariance kernel has a big impact on the model as it is tightly related to:

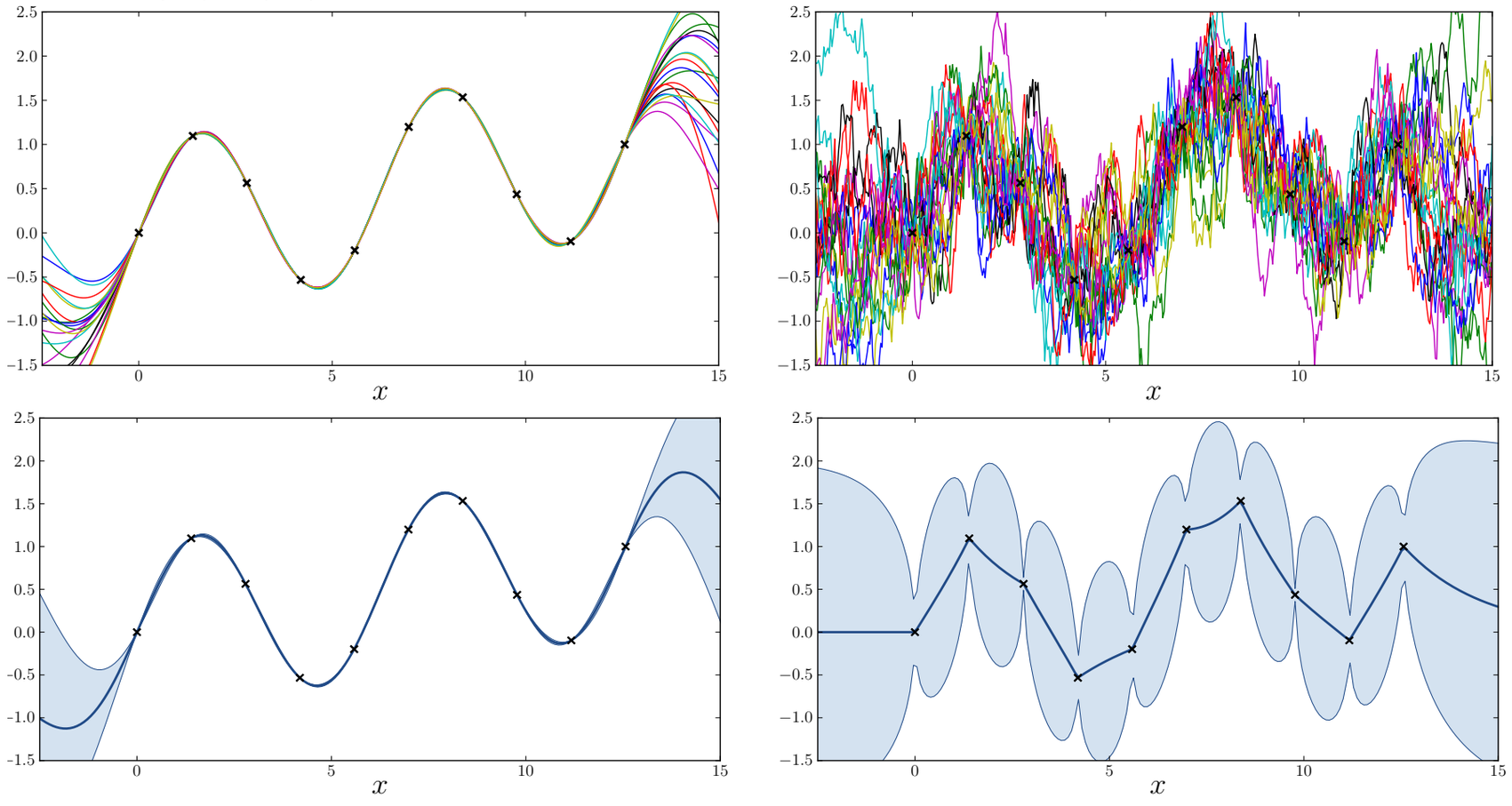
- The smoothness of the sample paths, hence the regularity of the predictor.
- The accuracy and uncertainty of the predictor.
- The conditioning of the correlation matrix, hence the efficiency of the learning algorithms.

Samples from a GP posterior

Posterior mean and variance

$K(\mathbf{x}, \mathbf{x}'; \theta) = \sigma^2 e^{-\theta|\mathbf{x}-\mathbf{x}'|^2}$

$K(\mathbf{x}, \mathbf{x}'; \theta) = \sigma^2 e^{-\theta|\mathbf{x}-\mathbf{x}'|}$



http://www.automaticstatistician.com/index/

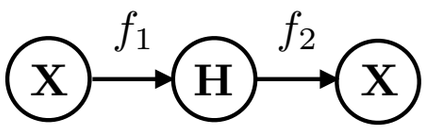
Duvenaud, D. K., Lloyd, J. R., Grosse, R. B., Tenenbaum, J. B., & Ghahramani, Z.. "Structure Discovery in Nonparametric Regression through Compositional Kernel Search", 2013

Model inversion in high-dimensions

Goal: Developed scalable algorithms for solving high- dimensional inverse problems

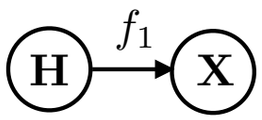
Optimization in physical space $\min_{\mathbf{x}\in\mathbb{R}^d} ||g(\mathbf{x}) - y^\star||$ ⟶ non-linear dim reduction $q << d$ ⟶ $\min_{\mathbf{h}\in\mathbf{H}^q} ||g(\mathbf{h}) - y^\star||$ *Optimization in latent space*

Deep auto-encoder (supervised)

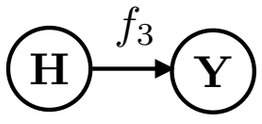


(a)

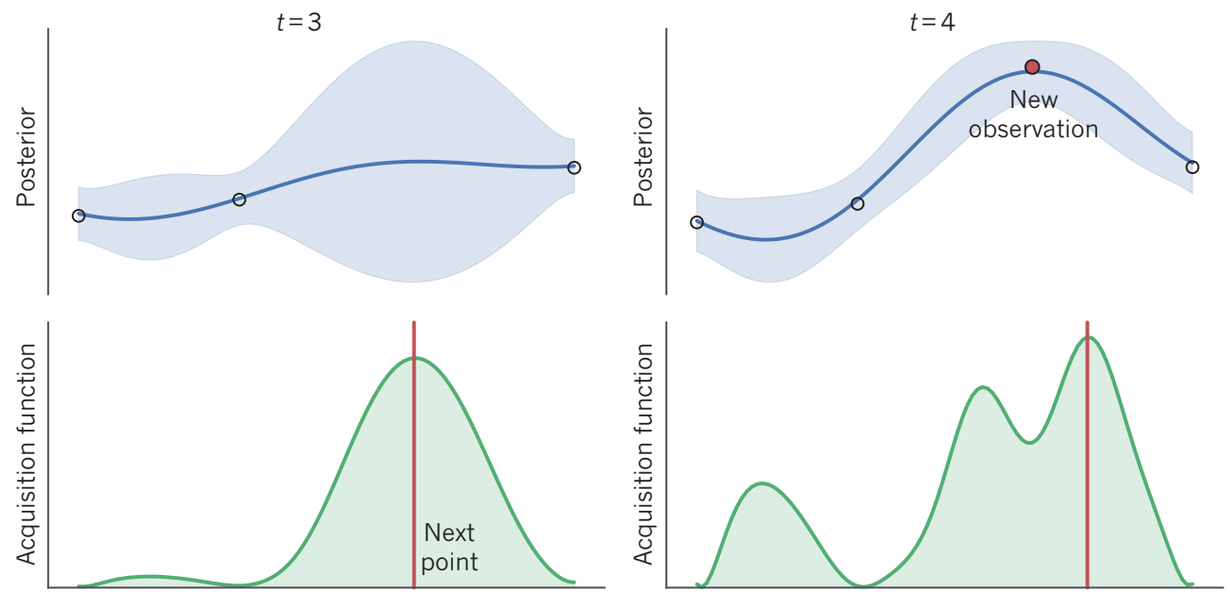
GPLVM (unsupervised)



(b)



$f_i \sim \mathcal{GP}$



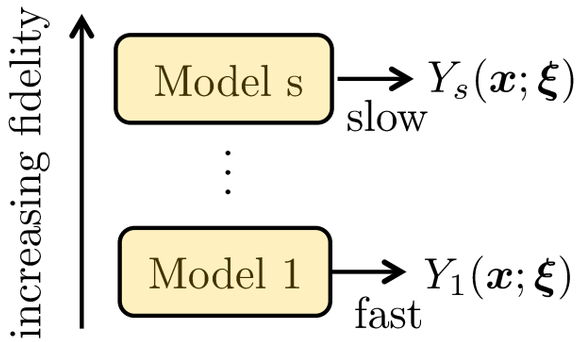
Bayesian optimization in latent space

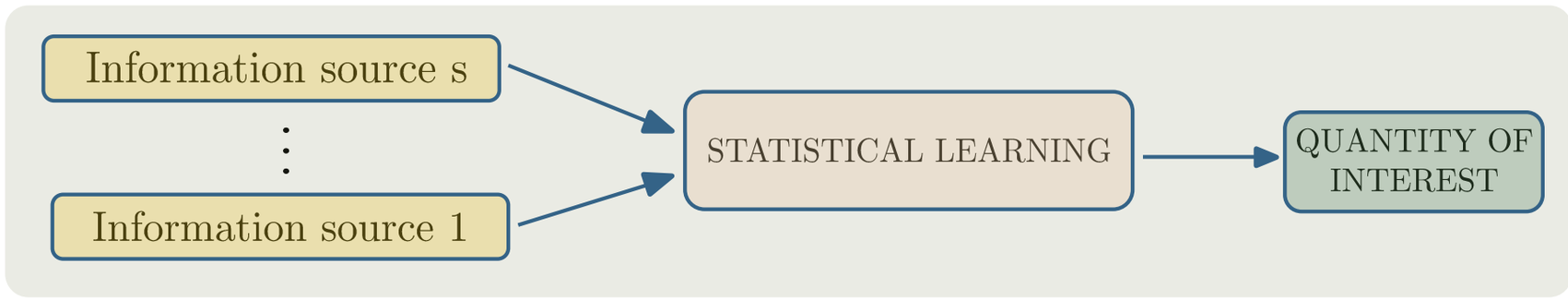
Technical approach:

- Non-linear dimensionality reduction using supervised deep auto-encoders and/or unsupervised GPLVMs
- Bayesian optimization in the low-dimensional latent space

Lawrence, N. D. "Gaussian process latent variable models for visualisation of high dimensional data." Advances in neural information processing systems 16.3 (2004): 329-336.
Shahriari, Bobak, et al. "Taking the human out of the loop: A review of bayesian optimization." Proceedings of the IEEE 104.1 (2016): 148-175.

Multi-fidelity modeling





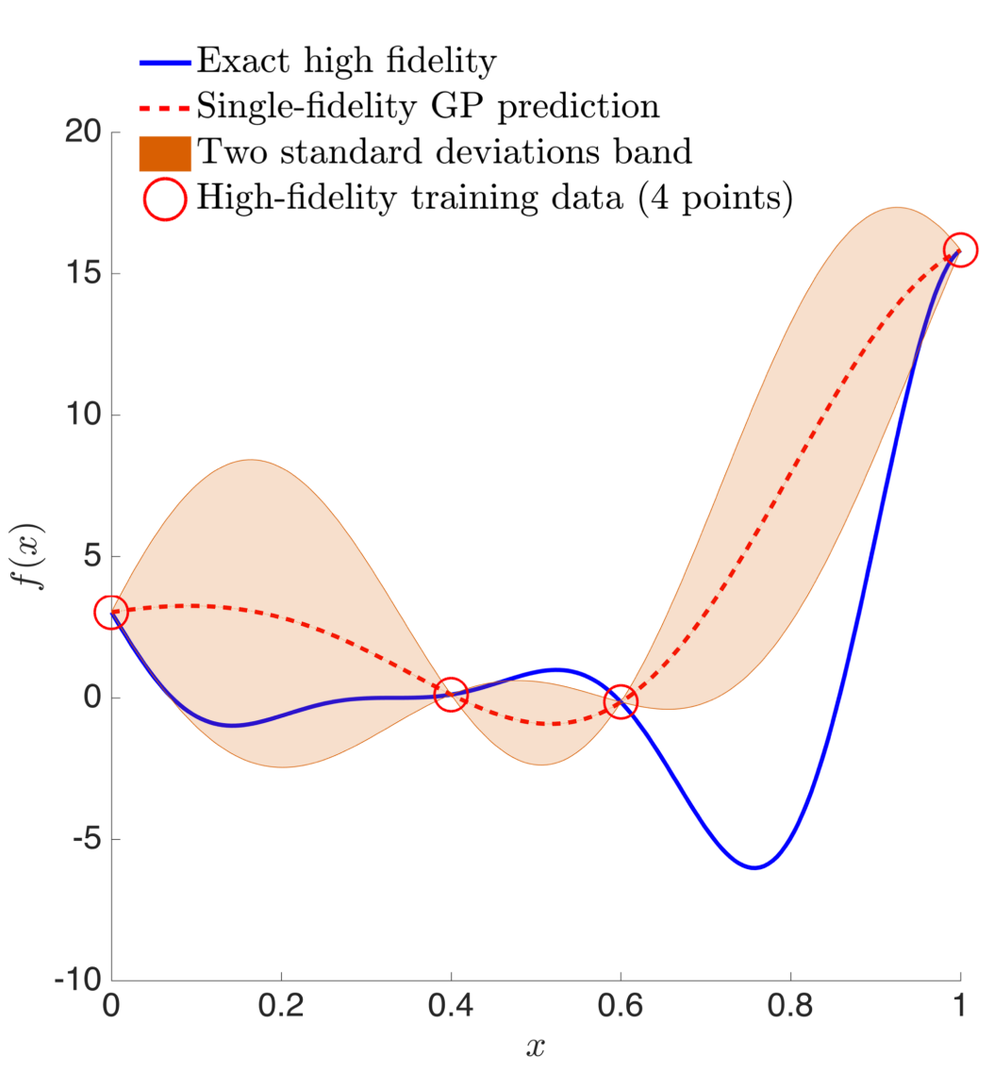
Number of runs is limited by time and computational resources

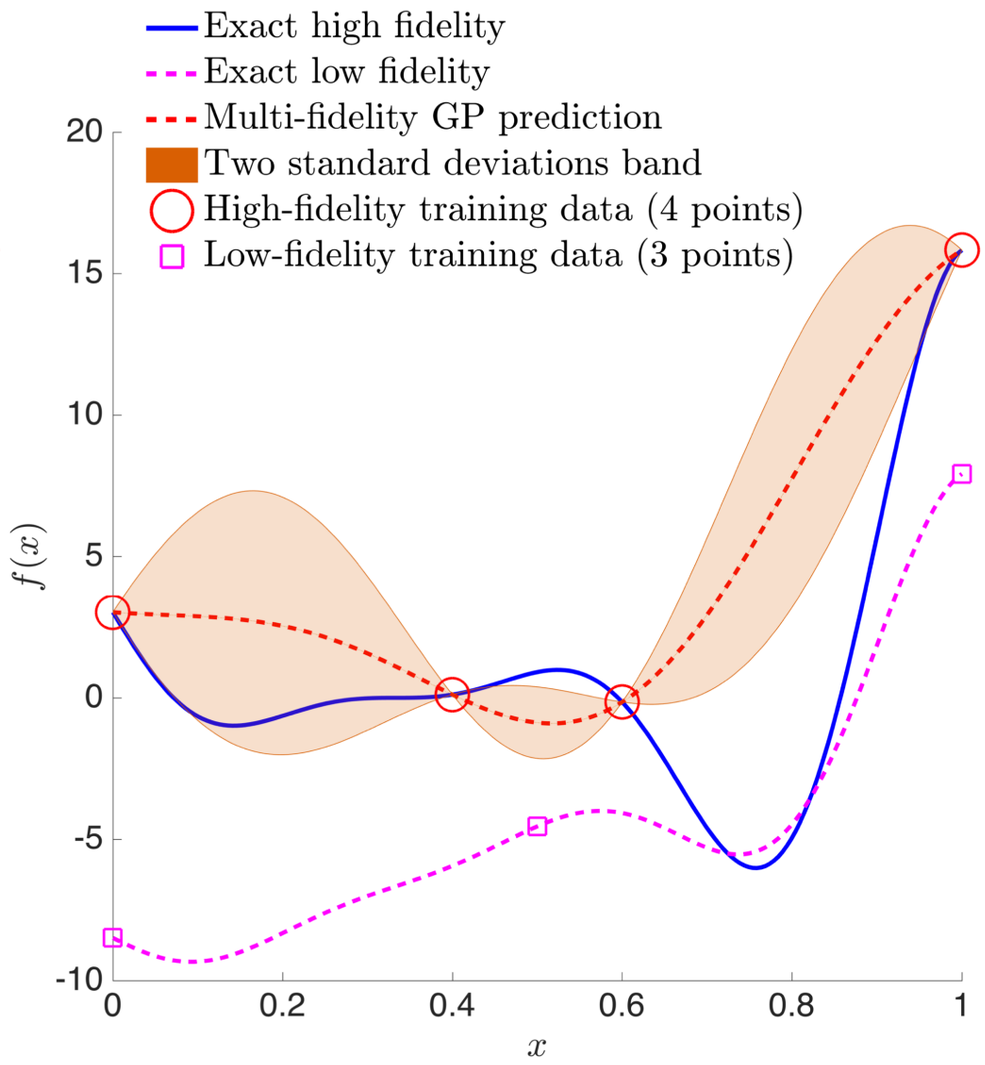
⇓

We cannot compute at all $(\boldsymbol{x};\boldsymbol{\xi})$

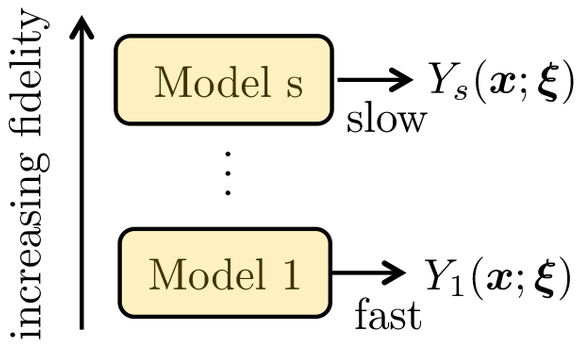
⇓

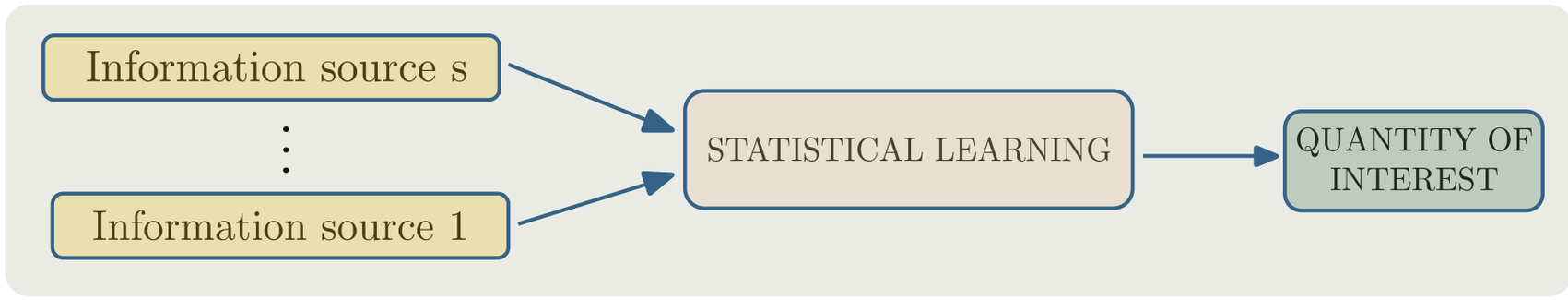
Prediction of $Z_i(\boldsymbol{x}) = \mathbb{E}[f(Y_i(\boldsymbol{x};\boldsymbol{\xi}))]$ is a problem of statistical inference





Multi-fidelity modeling





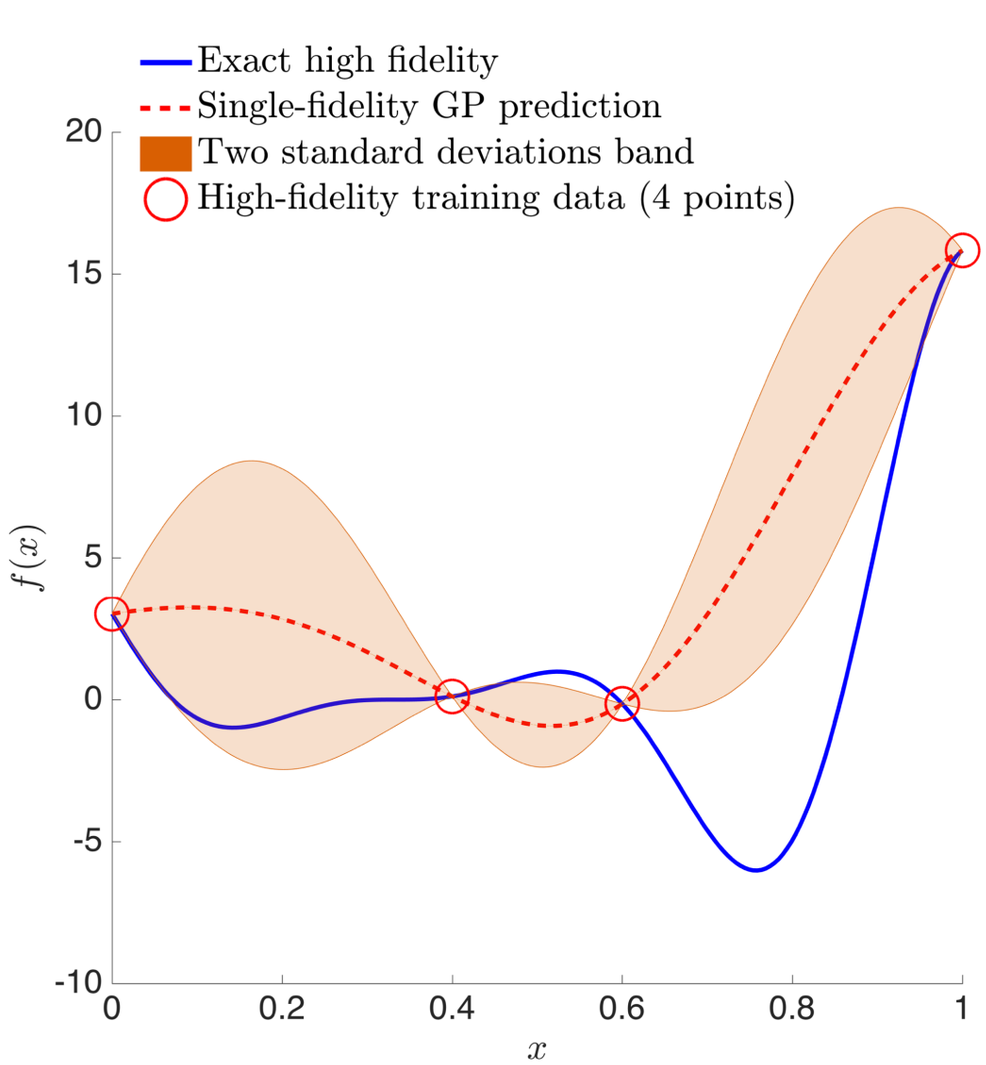
Number of runs is limited by time and computational resources

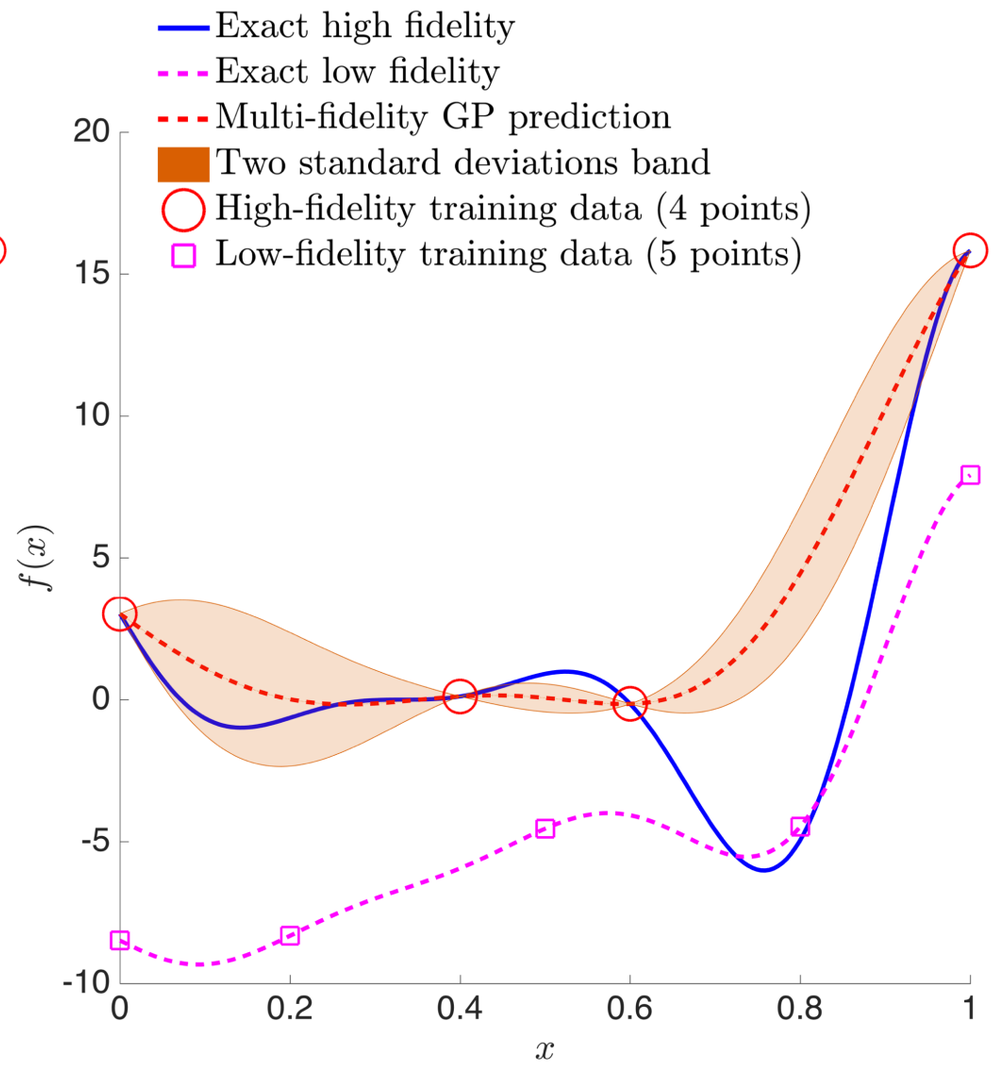
⇓

We cannot compute at all $(\boldsymbol{x};\boldsymbol{\xi})$

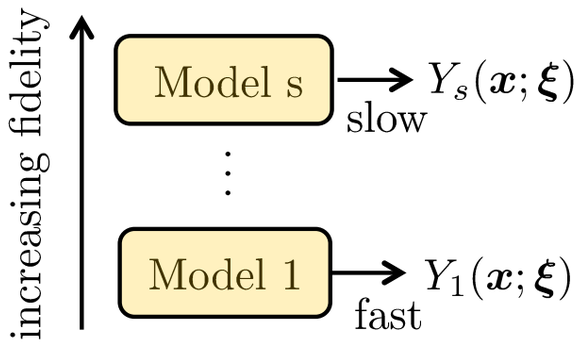
⇓

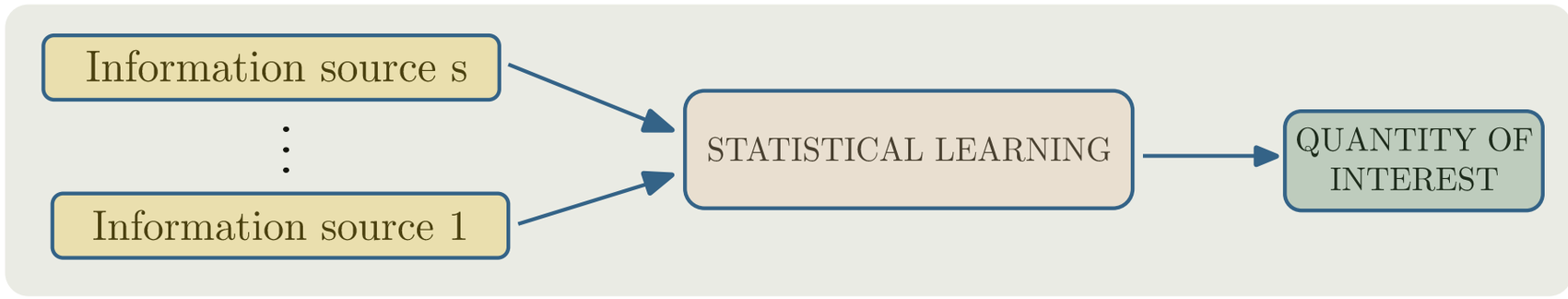
Prediction of $Z_i(\boldsymbol{x}) = \mathbb{E}[f(Y_i(\boldsymbol{x};\boldsymbol{\xi}))]$ is a problem of statistical inference





Multi-fidelity modeling





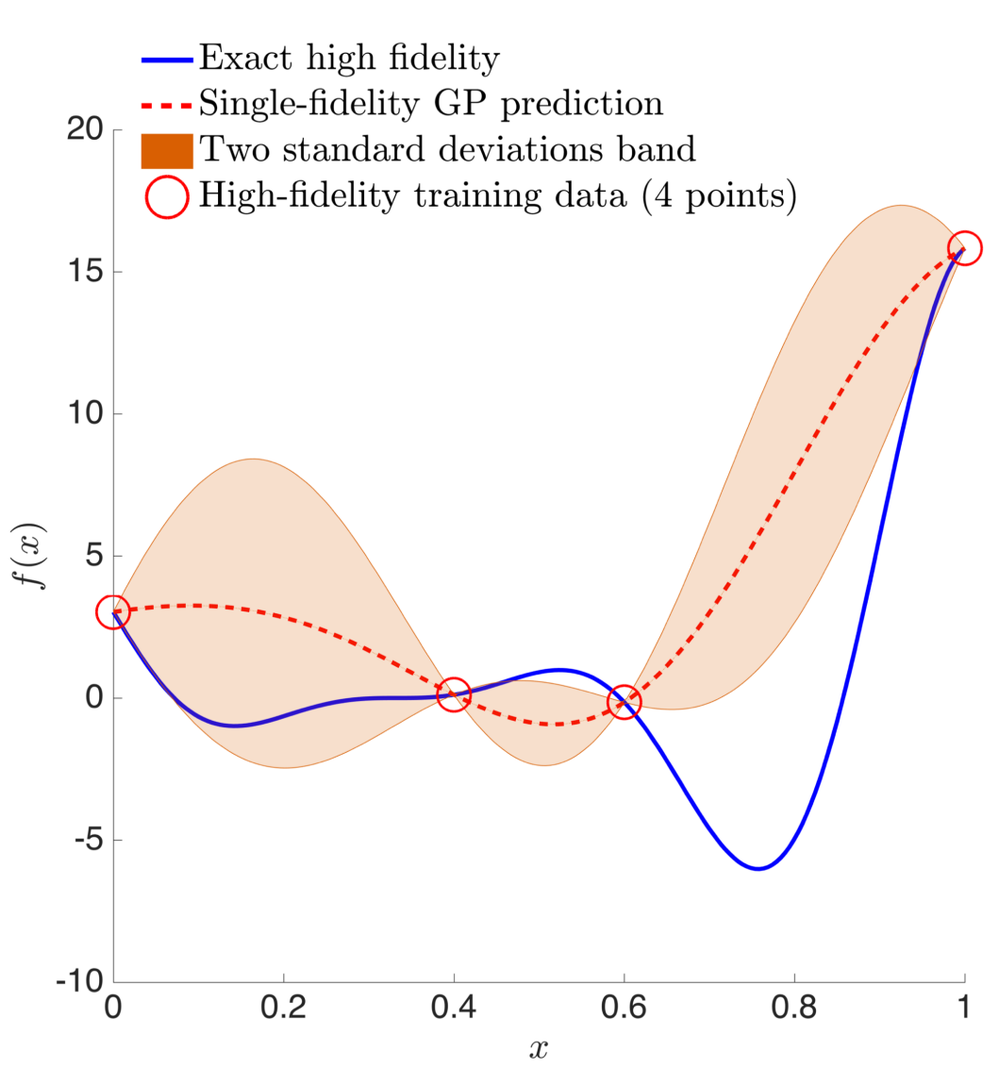
Number of runs is limited by time and computational resources

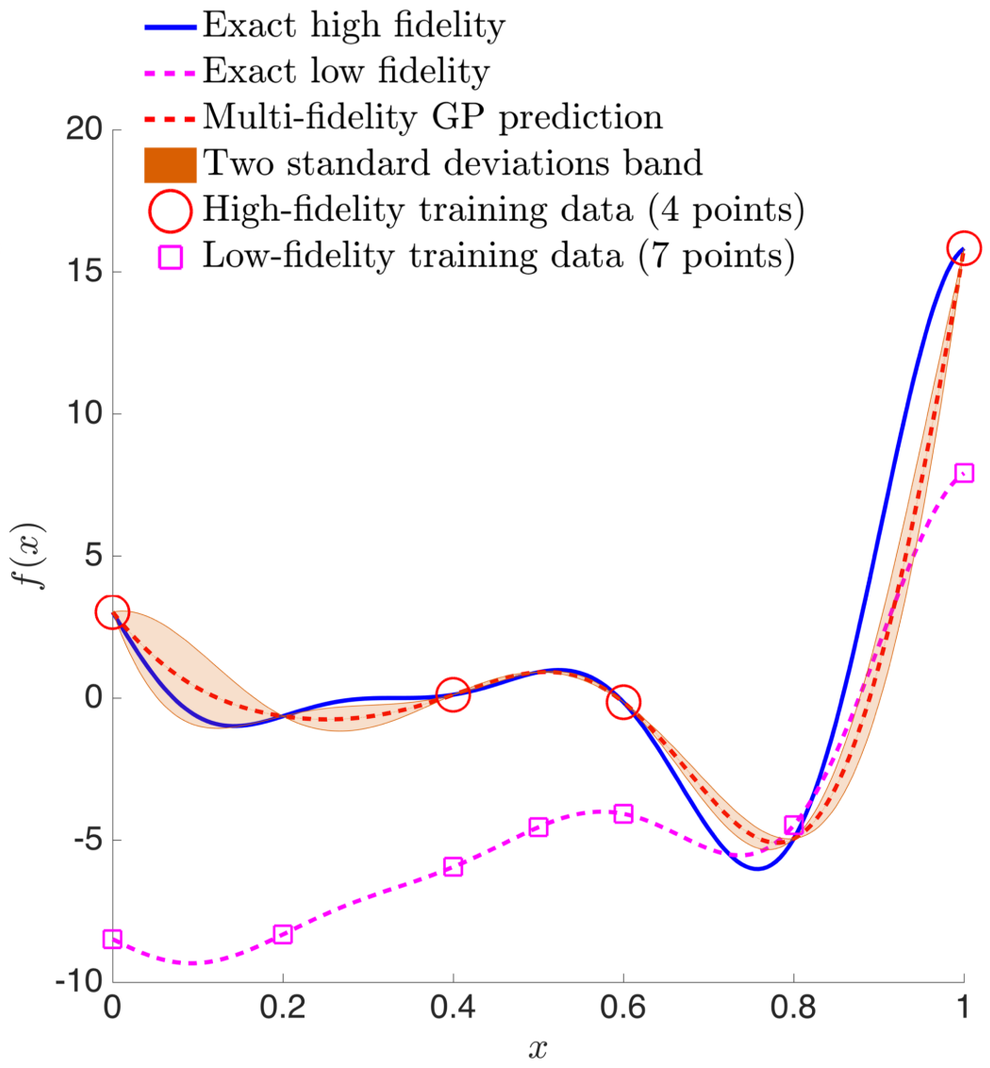
⇓

We cannot compute at all $(\boldsymbol{x};\boldsymbol{\xi})$

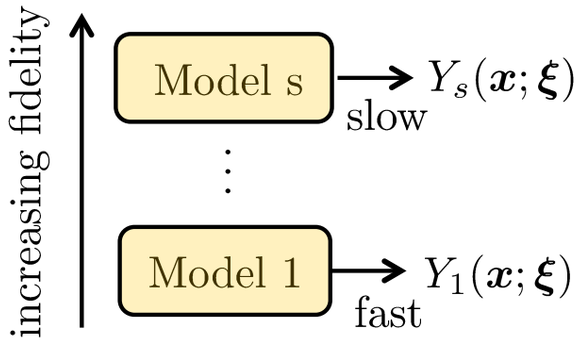
⇓

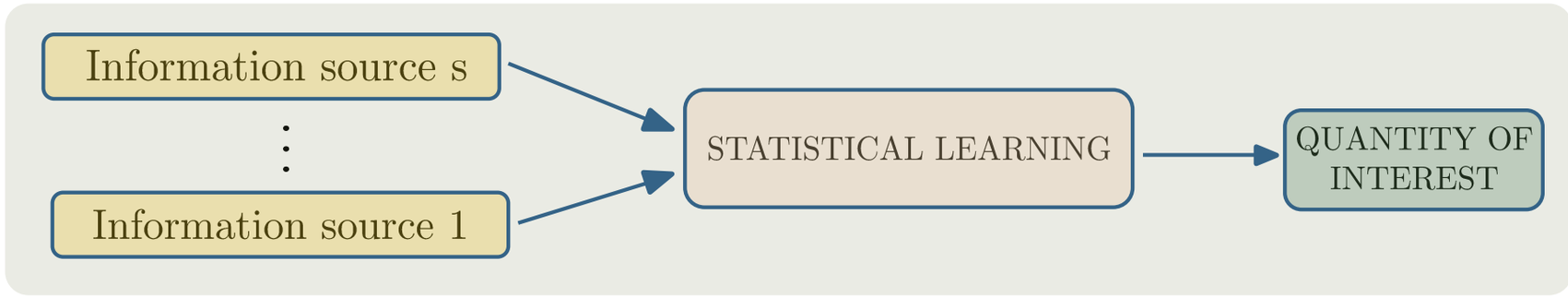
Prediction of $Z_i(\boldsymbol{x}) = \mathbb{E}[f(Y_i(\boldsymbol{x};\boldsymbol{\xi}))]$ is a problem of statistical inference





Multi-fidelity modeling





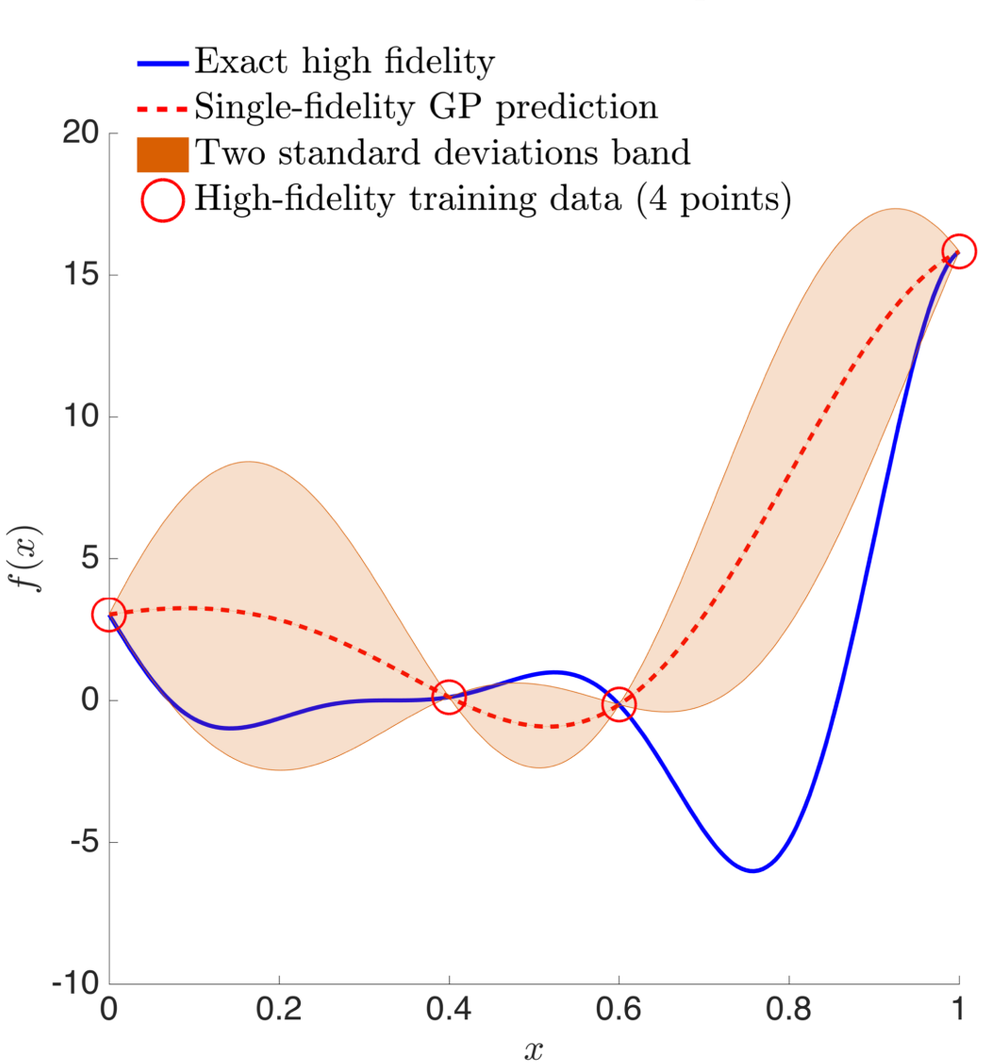
Number of runs is limited by time and computational resources

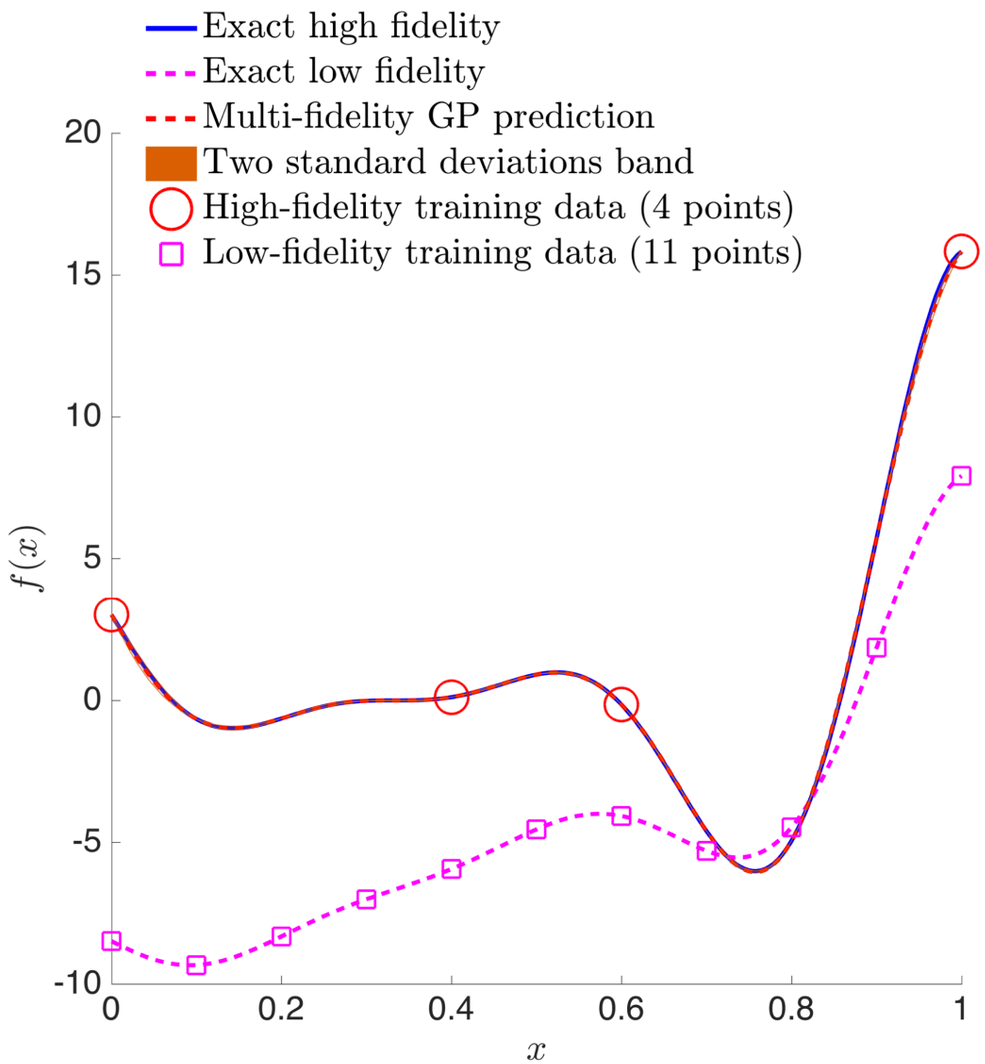
⇓

We cannot compute at all $(\boldsymbol{x};\boldsymbol{\xi})$

⇓

Prediction of $Z_i(\boldsymbol{x}) = \mathbb{E}[f(Y_i(\boldsymbol{x};\boldsymbol{\xi}))]$ is a problem of statistical inference



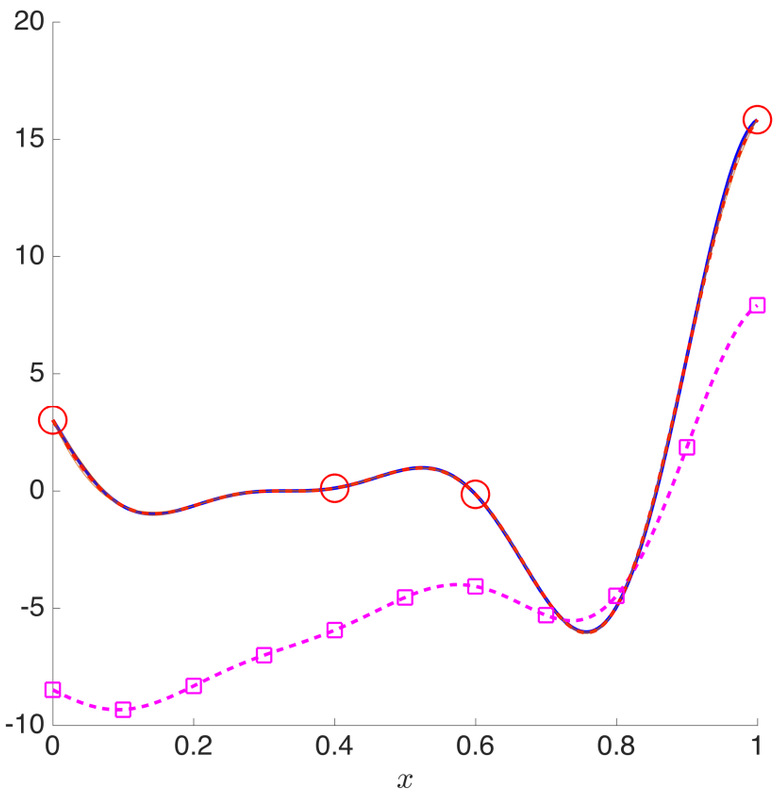


Multi-fidelity modeling with GPs

Auto-regressive model: AR1

$$f_t(\mathbf{x}) = \rho_{t-1}(\mathbf{x}) f_{t-1}(\mathbf{x}) + \delta_t(\mathbf{x})$$

$$t = 1, \ldots, s$$



Predicting the Output from a Complex Computer Code When Fast Approximations Are Available

M. C. Kennedy; A. O'Hagan

Biometrika, Vol. 87, No. 1. (Mar., 2000), pp. 1-13.

Predictive posterior

$$p(f_*|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{x}_*) = \mathcal{N}(f_*|\mu_*, \sigma_*^2),$$

$$\mu_*(\boldsymbol{x}_*) = \boldsymbol{k}_{*N}(\boldsymbol{K} + \sigma_\epsilon^2 \boldsymbol{I})^{-1}\boldsymbol{y},$$

$$\sigma_*^2(\boldsymbol{x}_*) = \boldsymbol{k}_{**} - \boldsymbol{k}_{*N}(\boldsymbol{K} + \sigma_\epsilon^2 \boldsymbol{I})^{-1}\boldsymbol{k}_{N*},$$

$$\begin{matrix} N_1 \\ N_2 \end{matrix} \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}$$

Block covariance matrix

Key idea: Replace f_{t-1} with the GP posterior of the previous level $\tilde{f}_{t-1}$

$$f_t(\mathbf{x}) = \rho_{t-1}(\mathbf{x}) f_{t-1}(\mathbf{x}) + \delta_t(\mathbf{x}) \qquad \tilde{f}_{t-1} \sim f_{t-1}|\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_{t-1}$$

Theorem (LeGratiet, 2014):

The predictive posterior of the recursive scheme has exactly the same distribution with the the fully coupled model given a nested experimental design. *(under the assumption of nested training sets!)*

M.C Kennedy, and A. O'Hagan. *Predicting the output from a complex computer code when fast approximations are available,* 2000.

L. Le Gratiet, and J. Garnier, *"Recursive co-kriging model for design of computer experiments with multiple levels of fidelity." International Journal for Uncertainty Quantification,* 2014