ENM 540: Data-driven modeling and probabilistic scientific computing

Scaling up Gaussian processes
Active learning & Bayesian optimization

Paris Perdikaris
March 13, 2018

Penn
University of Pennsylvania

Sparse Gaussian processes

Sparse Gaussian Processes using Pseudo-inputs

Edward Snelson **Zoubin Ghahramani**

Inference in a GP has the following demands:

$$\begin{aligned} \text{Complexity:} \quad & \mathcal{O}(n^3) \\ \text{Storage:} \quad & \mathcal{O}(n^2) \end{aligned}$$

Inference in a *sparse* GP has the following demands:

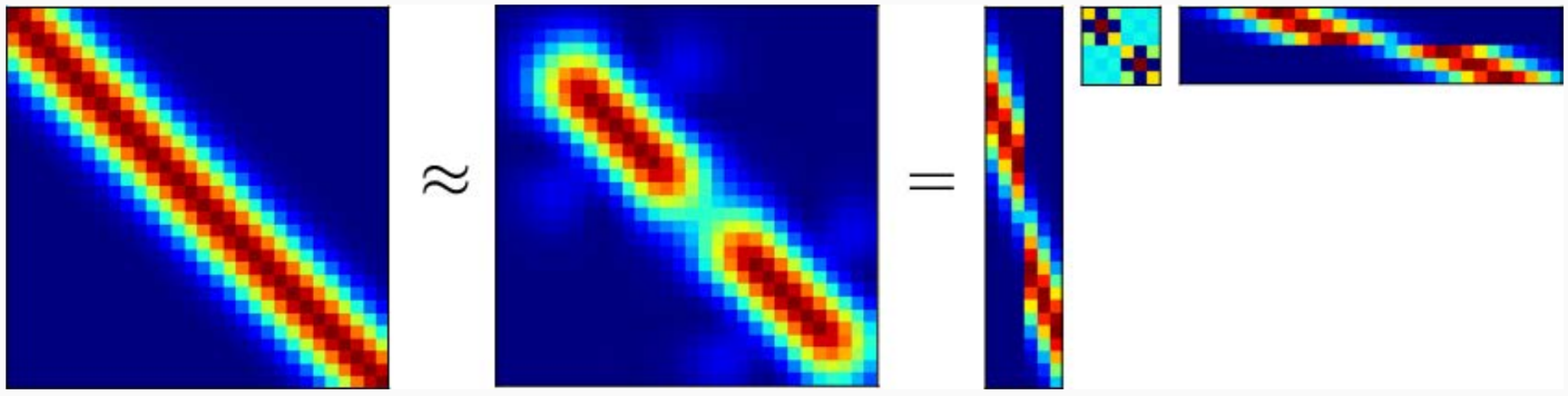
$$\begin{aligned} \text{Complexity:} \quad & \mathcal{O}(nm^2) \\ \text{Storage:} \quad & \mathcal{O}(nm) \end{aligned}$$

where we get to pick m!

http://gpss.cc/gpss15/talks/talk_james.pdf

Snelson, E., & Ghahramani, Z. (2006). Sparse Gaussian processes using pseudo-inputs. In Advances in neural information processing systems (pp. 1257-1264).

Sparse Gaussian processes



$$\mathbf{K}_{nn} \approx \mathbf{Q}_{nn} = \mathbf{K}_{nm}\mathbf{K}_{mm}^{-1}\mathbf{K}_{mn}$$

Instead of inverting $\mathbf{K}_{nn}$, we make a low rank (or Nyström) approximation, and invert $\mathbf{K}_{mm}$ instead.

http://gpss.cc/gpss15/talks/talk_james.pdf

Snelson, E., & Ghahramani, Z. (2006). Sparse Gaussian processes using pseudo-inputs. In Advances in neural information processing systems (pp. 1257-1264).

Sparse Gaussian processes

CHRONOLOGY

Subset of data

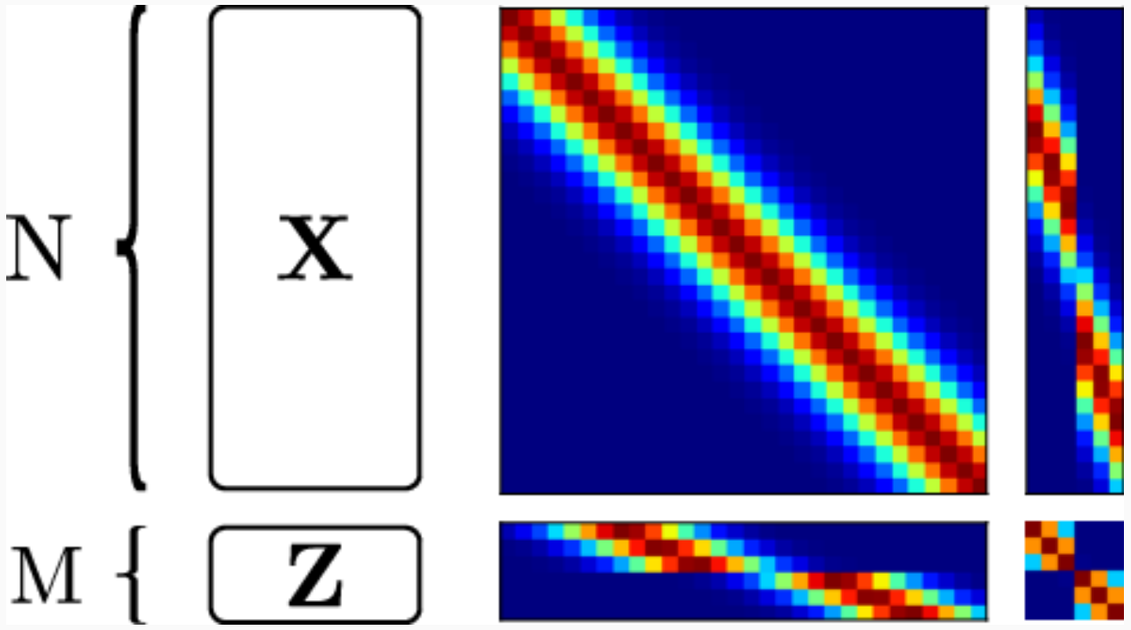
- Silverman 1985 (subset of regressors)
- Smola and Bartlett 2001 (greedy selection)

Pseudo-input approximaitons

- Snelson and Ghahramani (2005), Snelson (2007)

Variational approximations

- Titsias (2009) – derived a variational bound
- Matthews et al. (2015) – showed this minimised KL between processes



http://gpss.cc/gpss15/talks/talk_james.pdf

Snelson, E., & Ghahramani, Z. (2006). Sparse Gaussian processes using pseudo-inputs. In Advances in neural information processing systems (pp. 1257-1264).

Sparse Gaussian processes

Take and extra M points on the function, $\mathbf{u} = \mathsf{f}(\mathsf{Z})$.

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

$$p(\mathbf{y} \mid \mathbf{f}) = \mathcal{N}\left(\mathbf{y} | \mathbf{f}, \sigma^2 \mathbf{I}\right)$$
$$p(\mathbf{f} \mid \mathbf{u}) = \mathcal{N}\left(\mathbf{f} | \mathsf{K}_{nm}\mathsf{K}_{mm}\mathbf{u}, \widetilde{\mathsf{K}}\right)$$
$$p(\mathbf{u}) = \mathcal{N}\left(\mathbf{u} | \mathbf{0}, \mathsf{K}_{mm}\right)$$

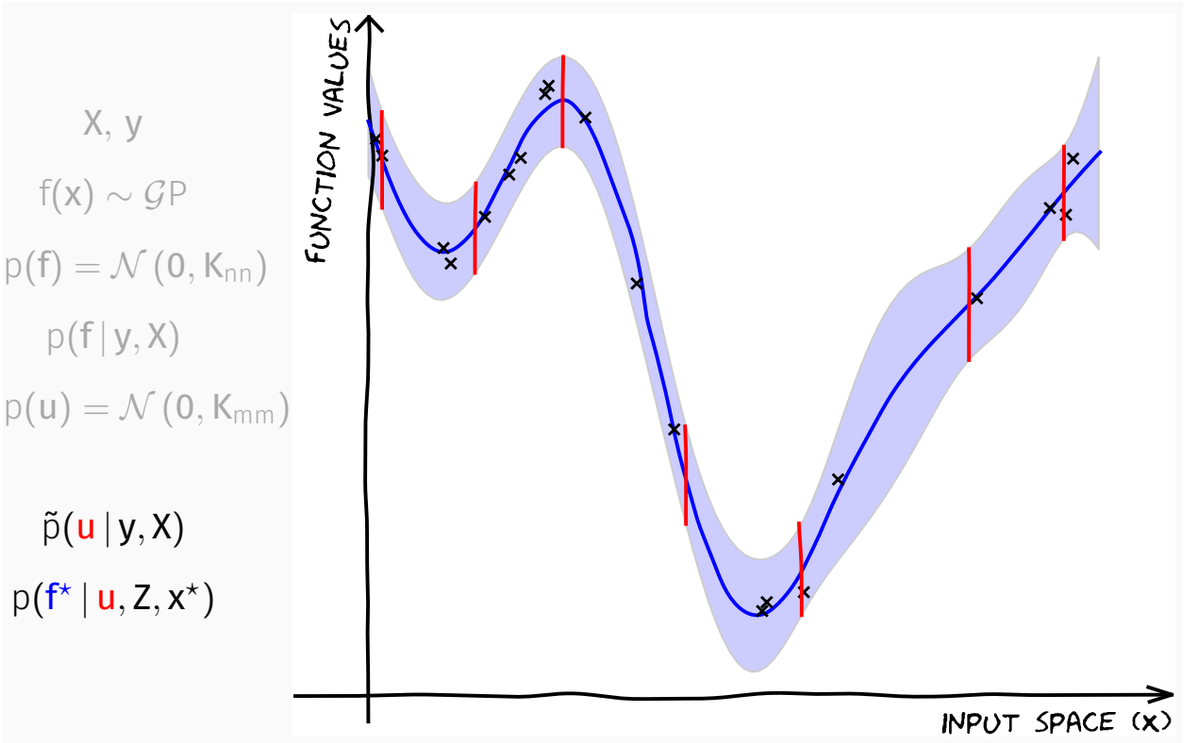
Instead of doing

$$p(\mathbf{f} \mid \mathbf{y}, \mathsf{X}) = \frac{p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{f} \mid \mathsf{X})}{\int p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{f} \mid \mathsf{X}) d\mathbf{f}}$$

We'll do

$$p(\mathbf{u} \mid \mathbf{y}, \mathsf{Z}) = \frac{p(\mathbf{y} \mid \mathbf{u}) p(\mathbf{u} \mid \mathsf{Z})}{\int p(\mathbf{y} \mid \mathbf{u}) p(\mathbf{u} \mid \mathsf{Z}) d\mathbf{u}}$$

http://gpss.cc/gpss15/talks/talk_james.pdf



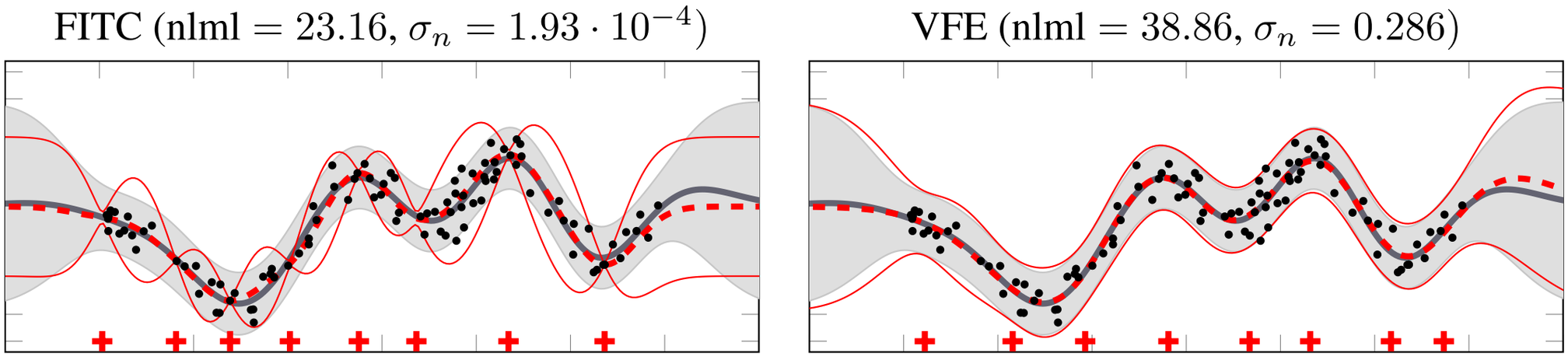
Sparse Gaussian processes

$$Q_{\mathbf{ff}} = K_{\mathbf{fu}} K_{\mathbf{uu}}^{-1} K_{\mathbf{uf}}$$

$$\mathcal{F} = \frac{N}{2}\log(2\pi) + \underbrace{\frac{1}{2}\log|Q_{\mathbf{ff}} + G|}_{\text{complexity penalty}} + \underbrace{\frac{1}{2}\mathbf{y}^{\mathsf{T}}(Q_{\mathbf{ff}} + G)^{-1}\mathbf{y}}_{\text{data fit}} + \underbrace{\frac{1}{2\sigma_n^2}\operatorname{tr}(T)}_{\text{trace term}},$$

$$\begin{aligned} G_{\mathrm{FITC}} &= \operatorname{diag}[K_{\mathbf{ff}} - Q_{\mathbf{ff}}] + \sigma_n^2 I \\ T_{\mathrm{FITC}} &= 0 \end{aligned} \qquad \begin{aligned} G_{\mathrm{VFE}} &= \sigma_n^2 I \\ T_{\mathrm{VFE}} &= K_{\mathbf{ff}} - Q_{\mathbf{ff}}. \end{aligned}$$

Fully Independent Training Conditional (FITC) vs *Variational Free Energy (VFE)*



Snelson, E., & Ghahramani, Z. (2006). Sparse Gaussian processes using pseudo-inputs. In Advances in neural information processing systems (pp. 1257-1264).

Titsias, M. (2009, April). Variational learning of inducing variables in sparse Gaussian processes. In Artificial Intelligence and Statistics (pp. 567-574).

Gaussian processes for big data

Gaussian Processes for Big Data

James Hensman*
Dept. Computer Science
The University of Sheffield
Sheffield, UK

Nicolò Fusi*
Dept. Computer Science
The University of Sheffield
Sheffield, UK

Neil D. Lawrence*
Dept. Computer Science
The University of Sheffield
Sheffield, UK

$$\log p(\mathbf{y} \mid \mathbf{X}) \geq \left\langle \mathcal{L}_1 + \log p(\mathbf{u}) - \log q(\mathbf{u}) \right\rangle_{q(\mathbf{u})} \triangleq \mathcal{L}_3$$

$$\begin{aligned} \mathcal{L}_3 = \sum_{i=1}^{n} \Bigg\{ & \log \mathcal{N}\left(y_i | \mathbf{k}_i^\top \mathbf{K}_{mm}^{-1} \mathbf{m}, \beta^{-1}\right) \\ & - \frac{1}{2}\beta \widetilde{k}_{i,i} - \frac{1}{2}\mathrm{tr}\left(\mathbf{S}\boldsymbol{\Lambda}_i\right) \Bigg\} \\ & - \mathrm{KL}\left(q(\mathbf{u}) \,\|\, p(\mathbf{u})\right) \end{aligned}$$

Hensman, J., Fusi, N., & Lawrence, N. D. (2013). Gaussian processes for big data. arXiv preprint arXiv:1309.6835.

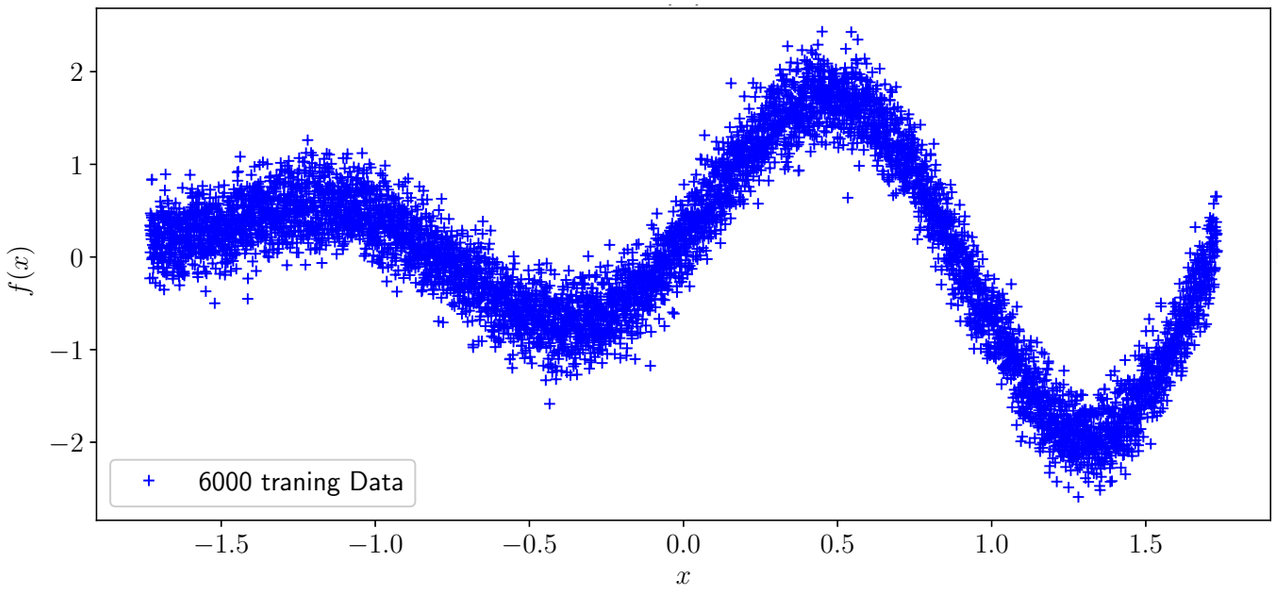
Gaussian processes for big data

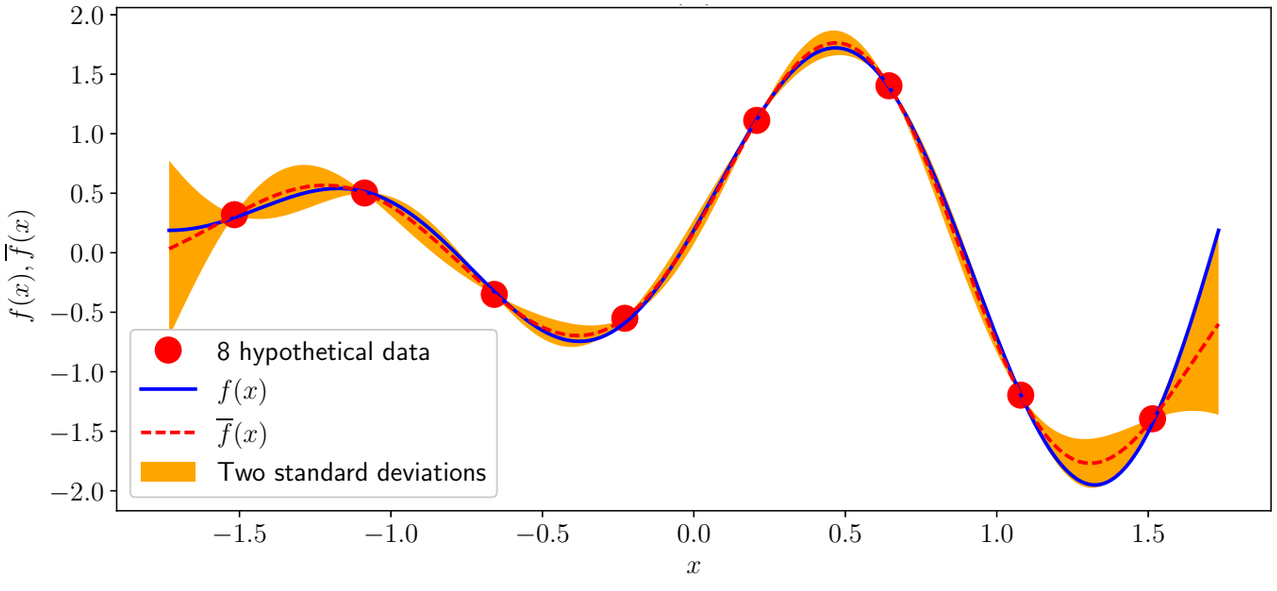
Parametric Gaussian Process Regression for Big Data

Maziar Raissi*
Division of Applied Mathematics
Brown University
Providence, RI 02912
maziar_raissi@brown.edu

$$\boldsymbol{u} \sim \mathcal{N}(\boldsymbol{m}, \boldsymbol{S})$$

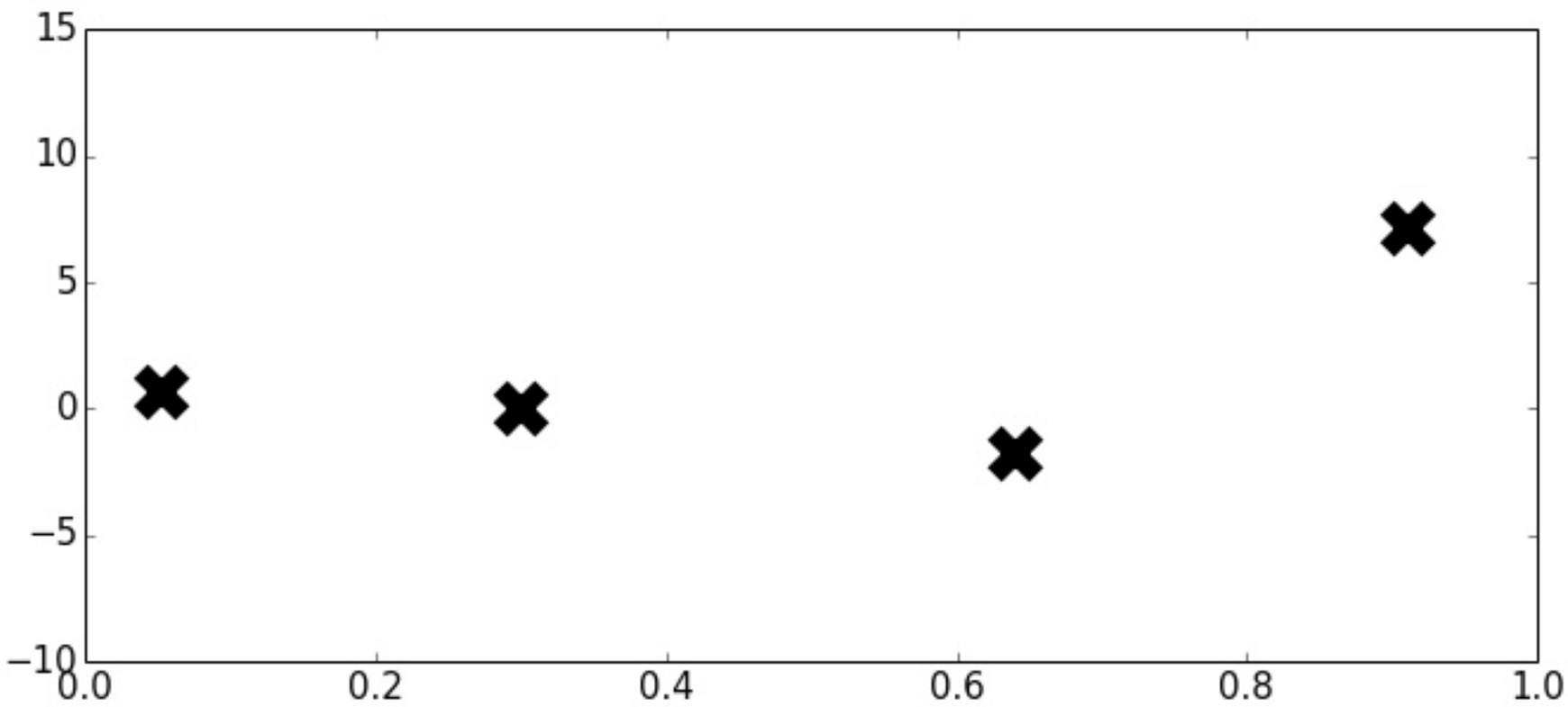
$$\begin{aligned} \boldsymbol{m} &\leftarrow \mu(\boldsymbol{Z};\boldsymbol{\theta},\boldsymbol{m}) + \Sigma(\boldsymbol{Z},\widetilde{\boldsymbol{X}};\boldsymbol{\theta},\boldsymbol{S})\left(\Sigma(\widetilde{\boldsymbol{X}},\widetilde{\boldsymbol{X}};\boldsymbol{\theta},\boldsymbol{S}) + \sigma_\epsilon^2\boldsymbol{I}\right)^{-1}\left[\widetilde{\boldsymbol{y}} - \mu(\widetilde{\boldsymbol{X}};\boldsymbol{\theta},\boldsymbol{m})\right], \\ \boldsymbol{S} &\leftarrow \Sigma(\boldsymbol{Z},\boldsymbol{Z};\boldsymbol{\theta},\boldsymbol{S}) - \Sigma(\boldsymbol{Z},\widetilde{\boldsymbol{X}};\boldsymbol{\theta},\boldsymbol{S})\left(\Sigma(\widetilde{\boldsymbol{X}},\widetilde{\boldsymbol{X}};\boldsymbol{\theta},\boldsymbol{S}) + \sigma_\epsilon^2\boldsymbol{I}\right)^{-1}\Sigma(\widetilde{\boldsymbol{X}},\boldsymbol{Z};\boldsymbol{\theta},\boldsymbol{S}). \end{aligned}$$





Raissi, M. (2017). Parametric Gaussian Process Regression for Big Data. arXiv preprint arXiv:1704.03144.

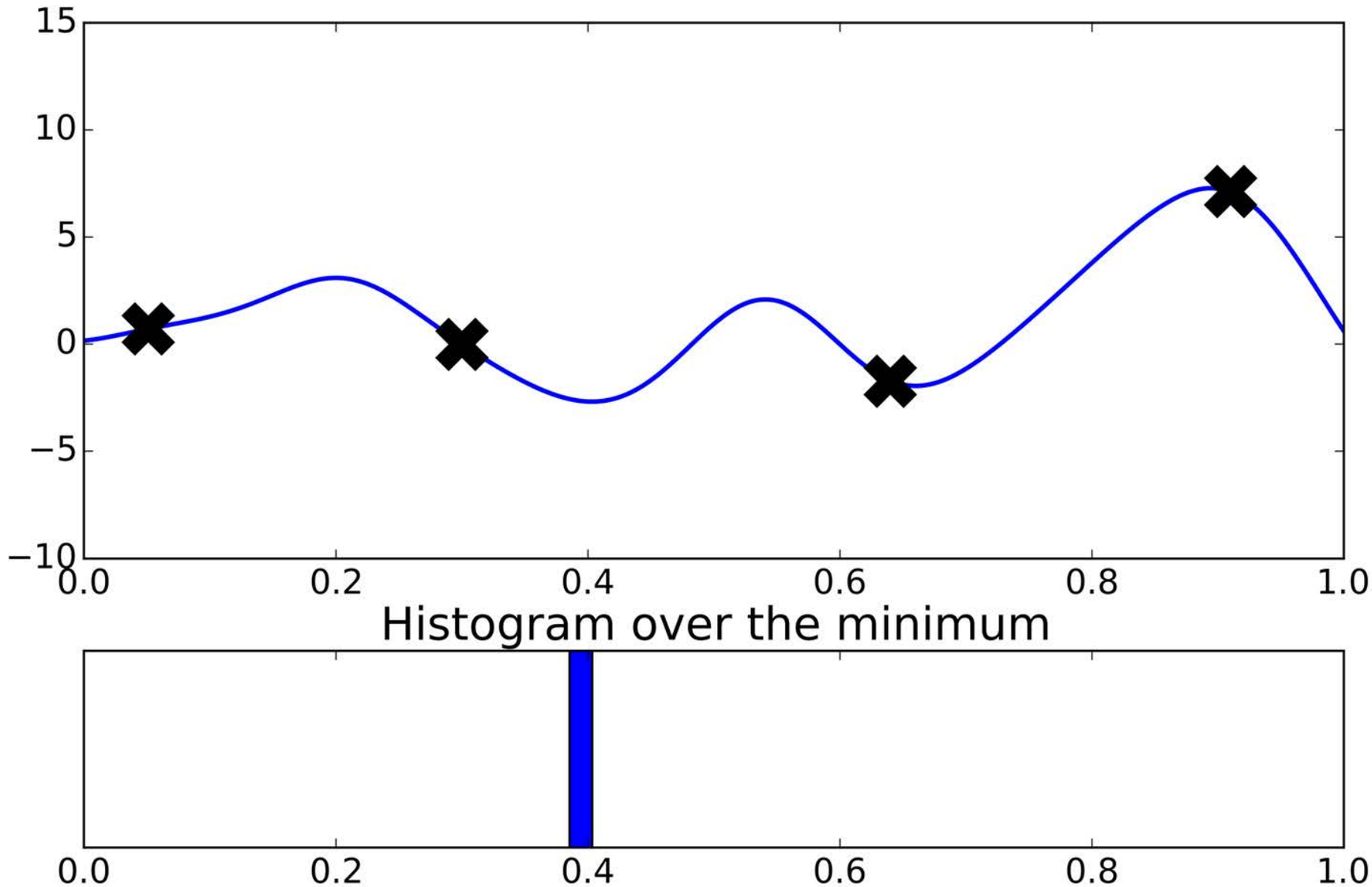
Bayesian optimization



Where is the minimum of f?
Where should the take the next evaluation?

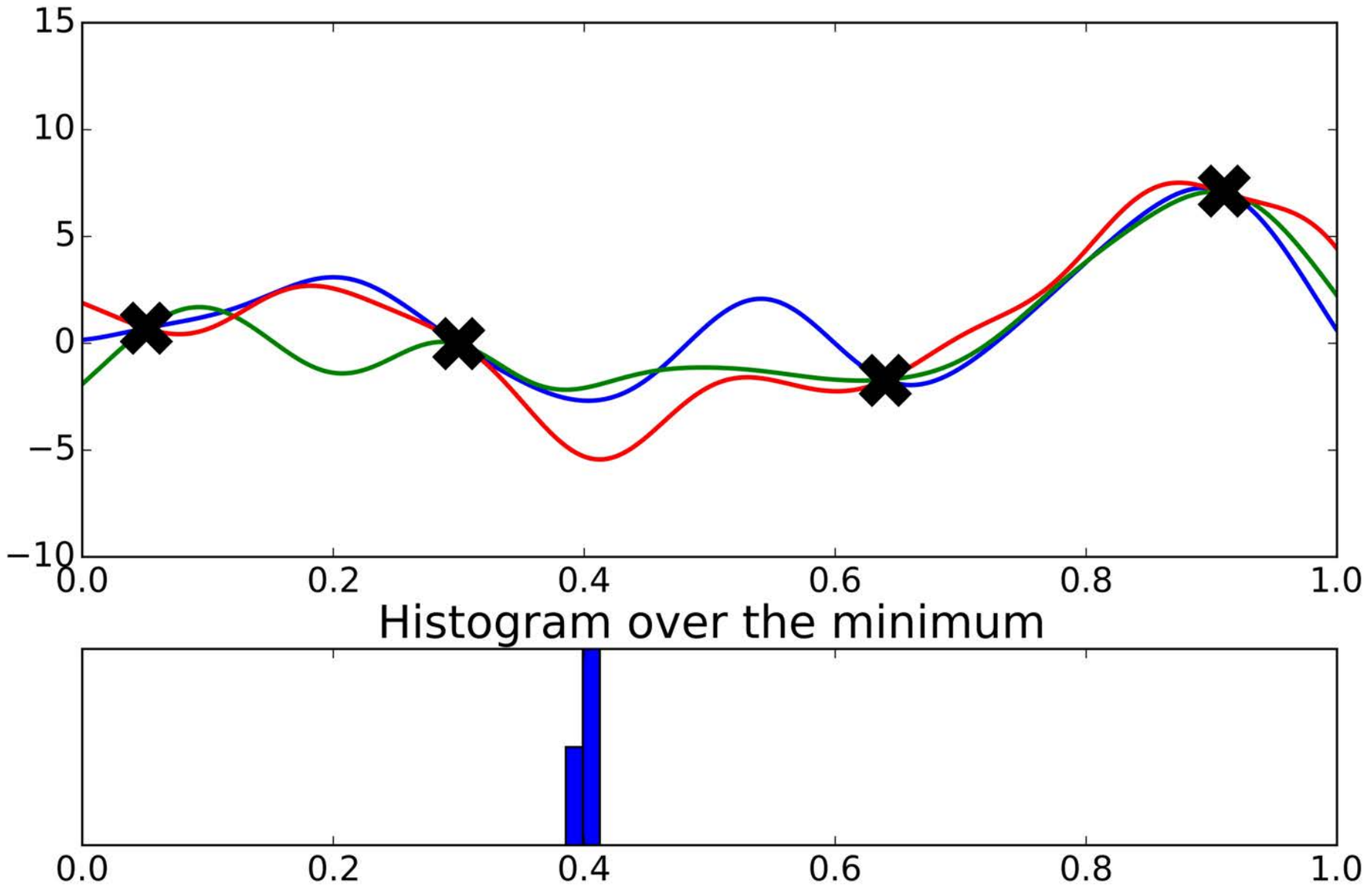
http://gpss.cc/gpmc17/slides/LancasterMasterclass_1.pdf

Bayesian optimization



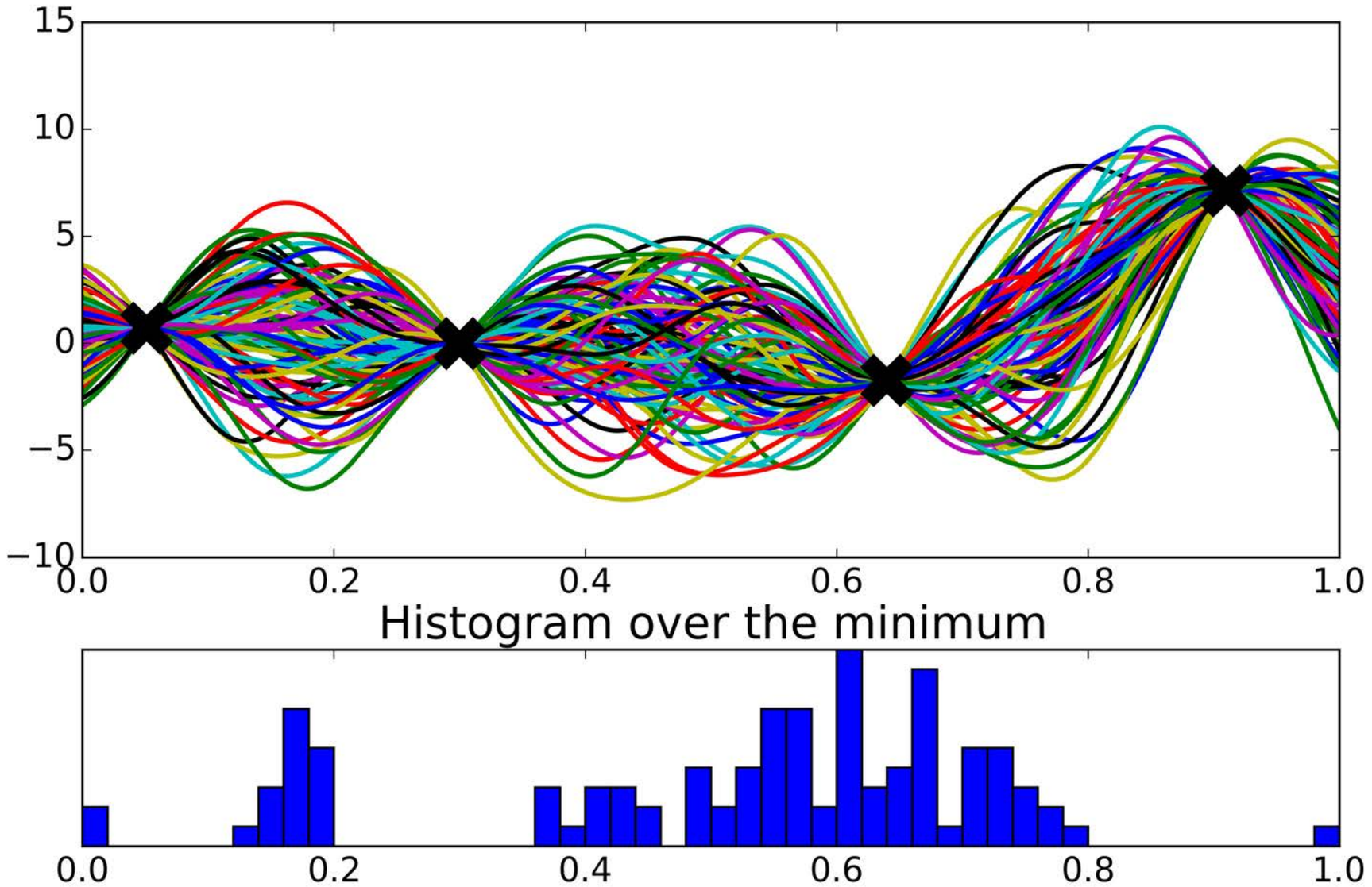
http://gpss.cc/gpmc17/slides/LancasterMasterclass_1.pdf

Bayesian optimization



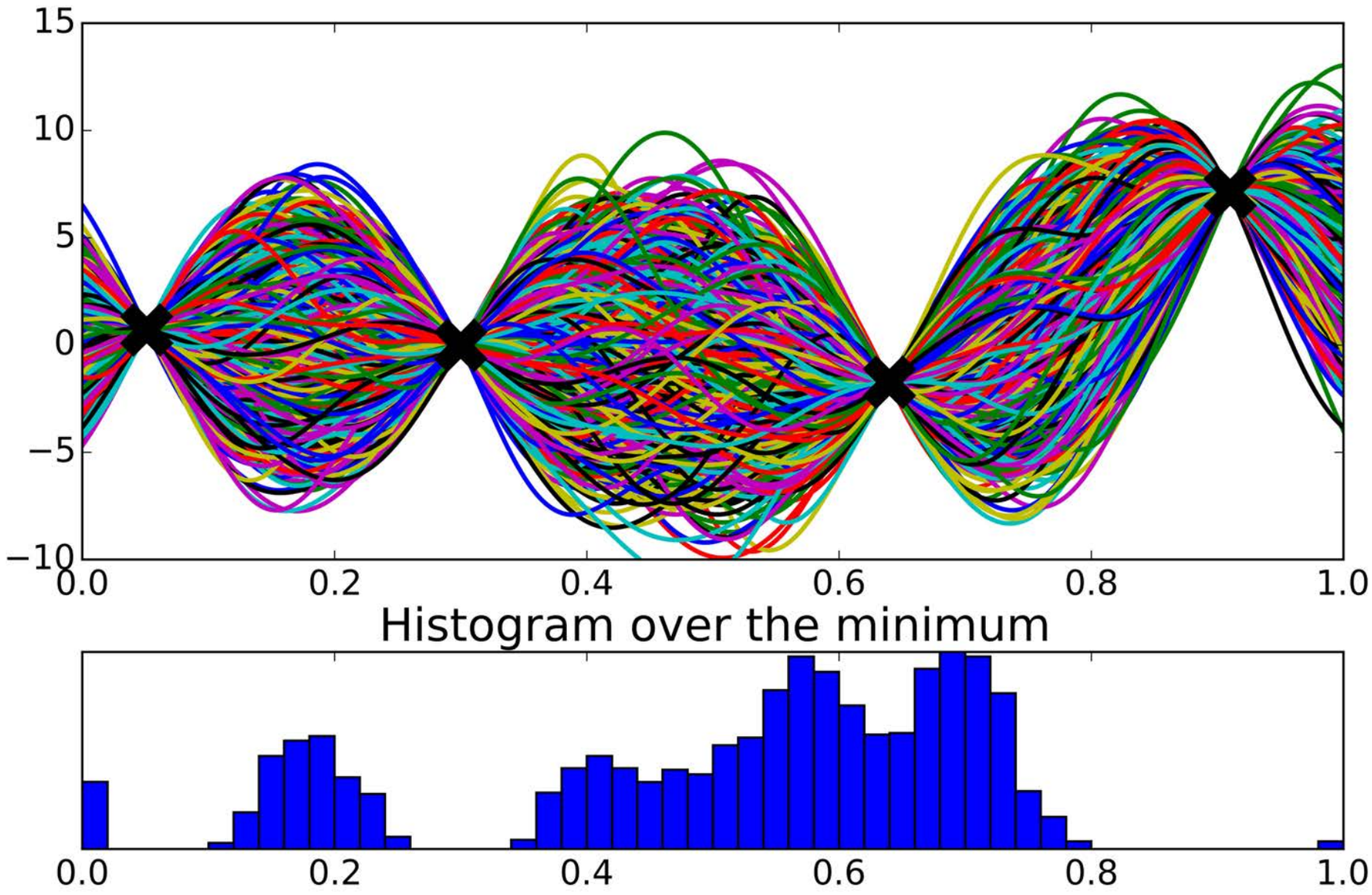
http://gpss.cc/gpmc17/slides/LancasterMasterclass_1.pdf

Bayesian optimization



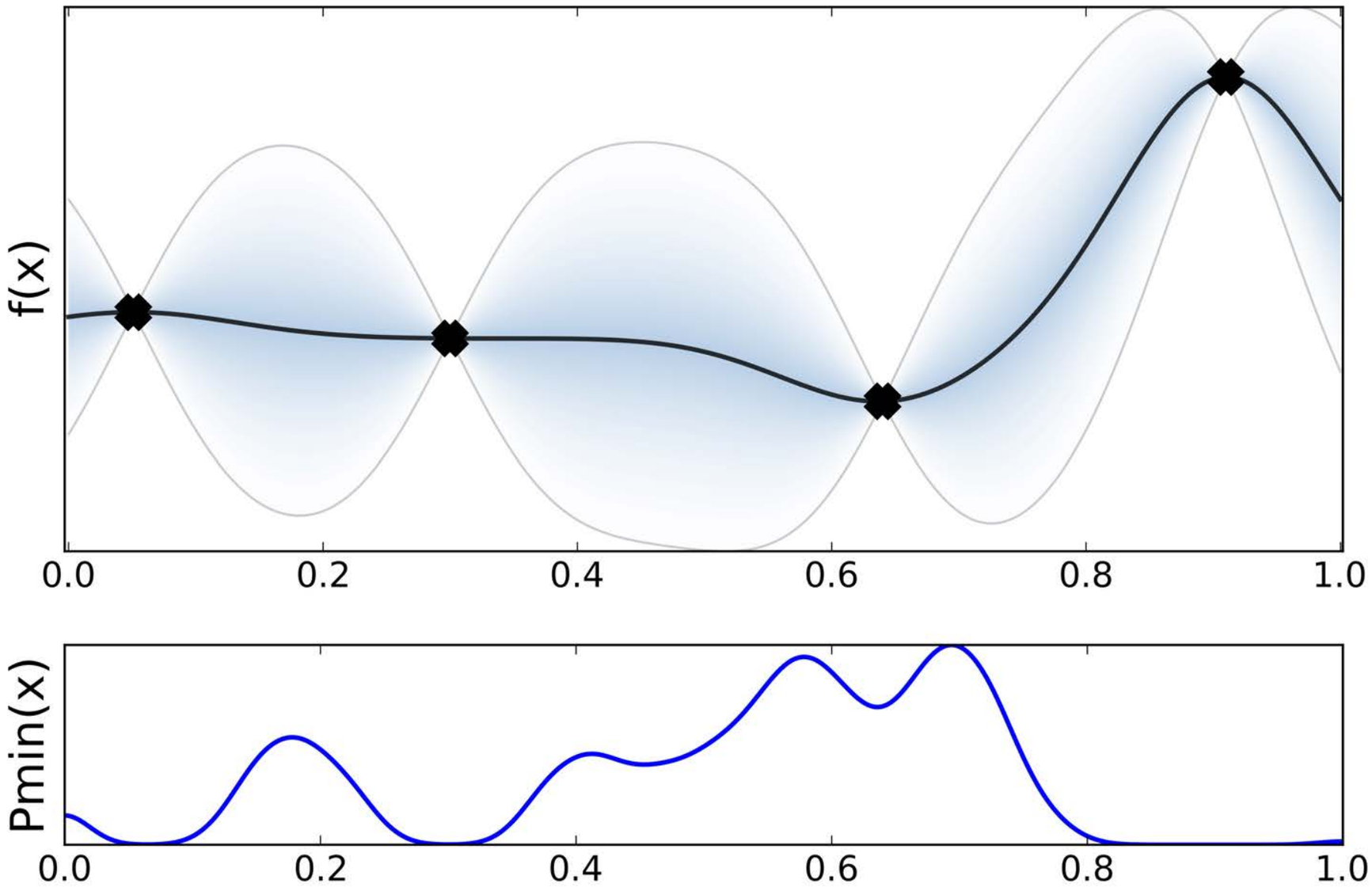
http://gpss.cc/gpmc17/slides/LancasterMasterclass_1.pdf

Bayesian optimization



http://gpss.cc/gpmc17/slides/LancasterMasterclass_1.pdf

Bayesian optimization



$x^* = \arg\min f(x)$ ..but $p(x^*|\mathcal{D})$ is not tractable.

http://gpss.cc/gpmc17/slides/LancasterMasterclass_1.pdf

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

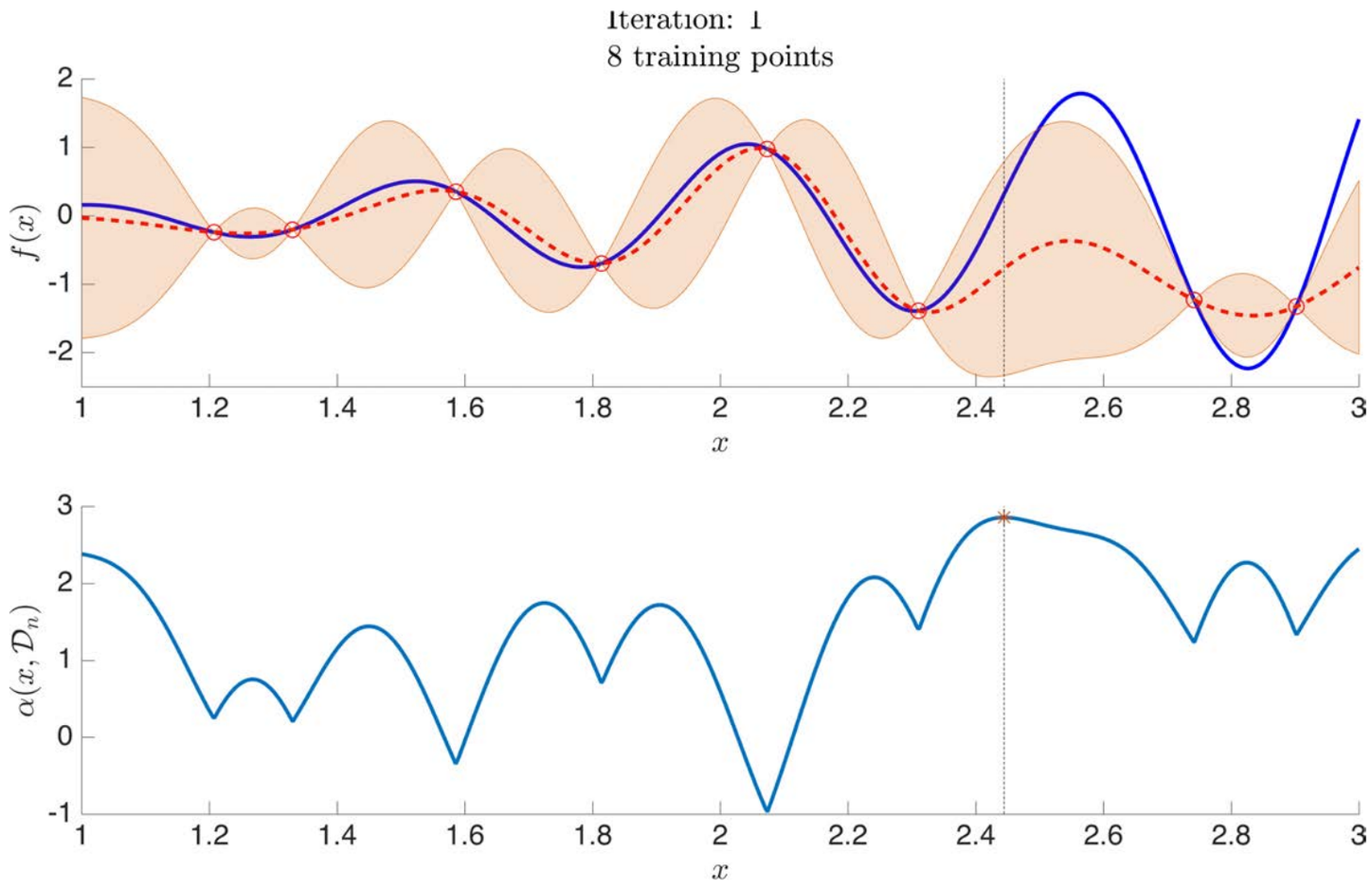
$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_\theta \mathbb{E}_{v\,|\,\mathbf{x},\theta}[U(\mathbf{x}, v, \theta)]$$

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 super-quintile risk confidence bound

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

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

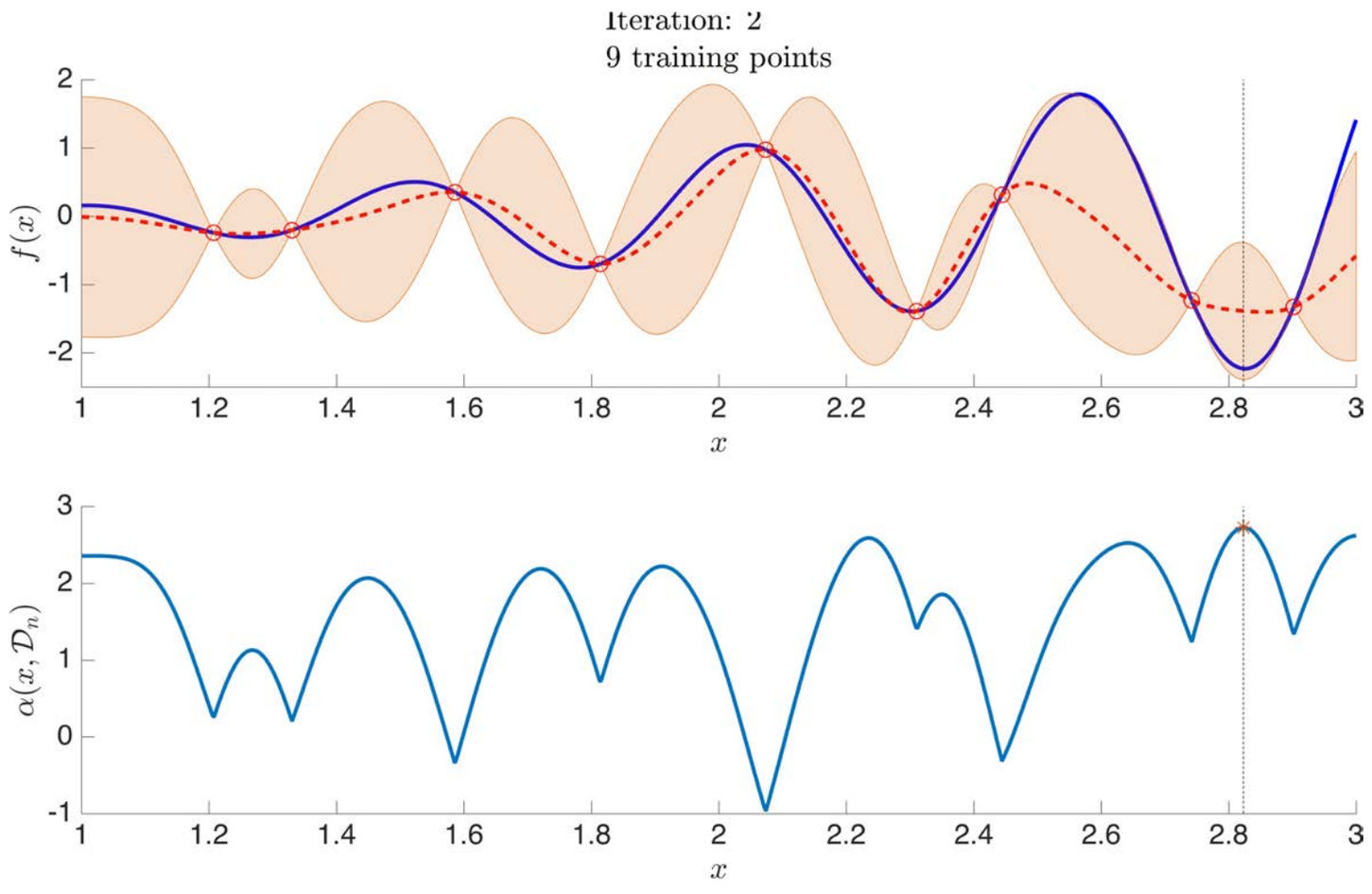
$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_\theta \mathbb{E}_{v\,|\,\mathbf{x},\theta}[U(\mathbf{x}, v, \theta)]$$

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 super-quintile risk confidence bound

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

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

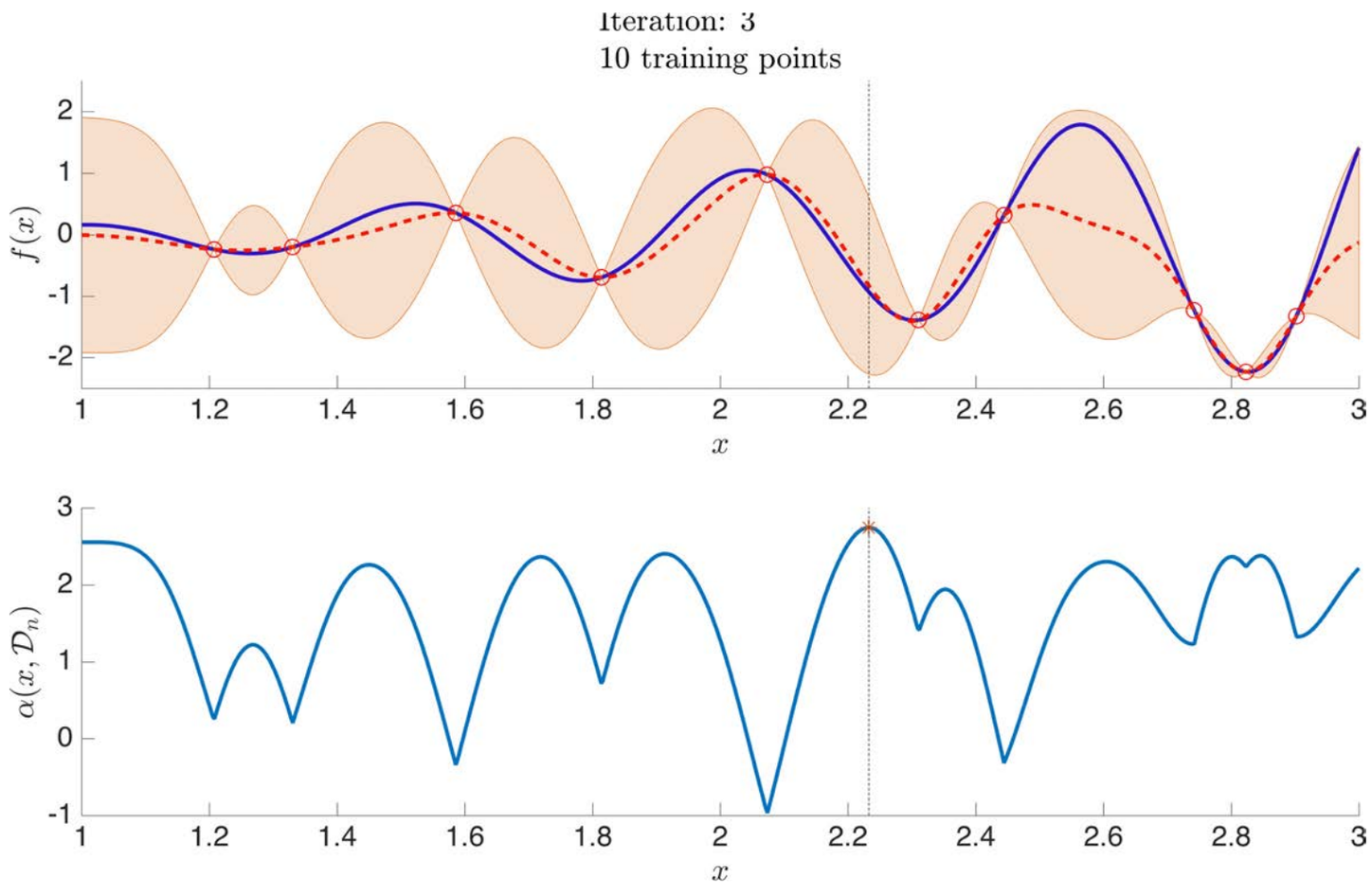
$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_\theta \mathbb{E}_{v\,|\,\mathbf{x},\theta}[U(\mathbf{x}, v, \theta)]$$

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 super-quintile risk confidence bound

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

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

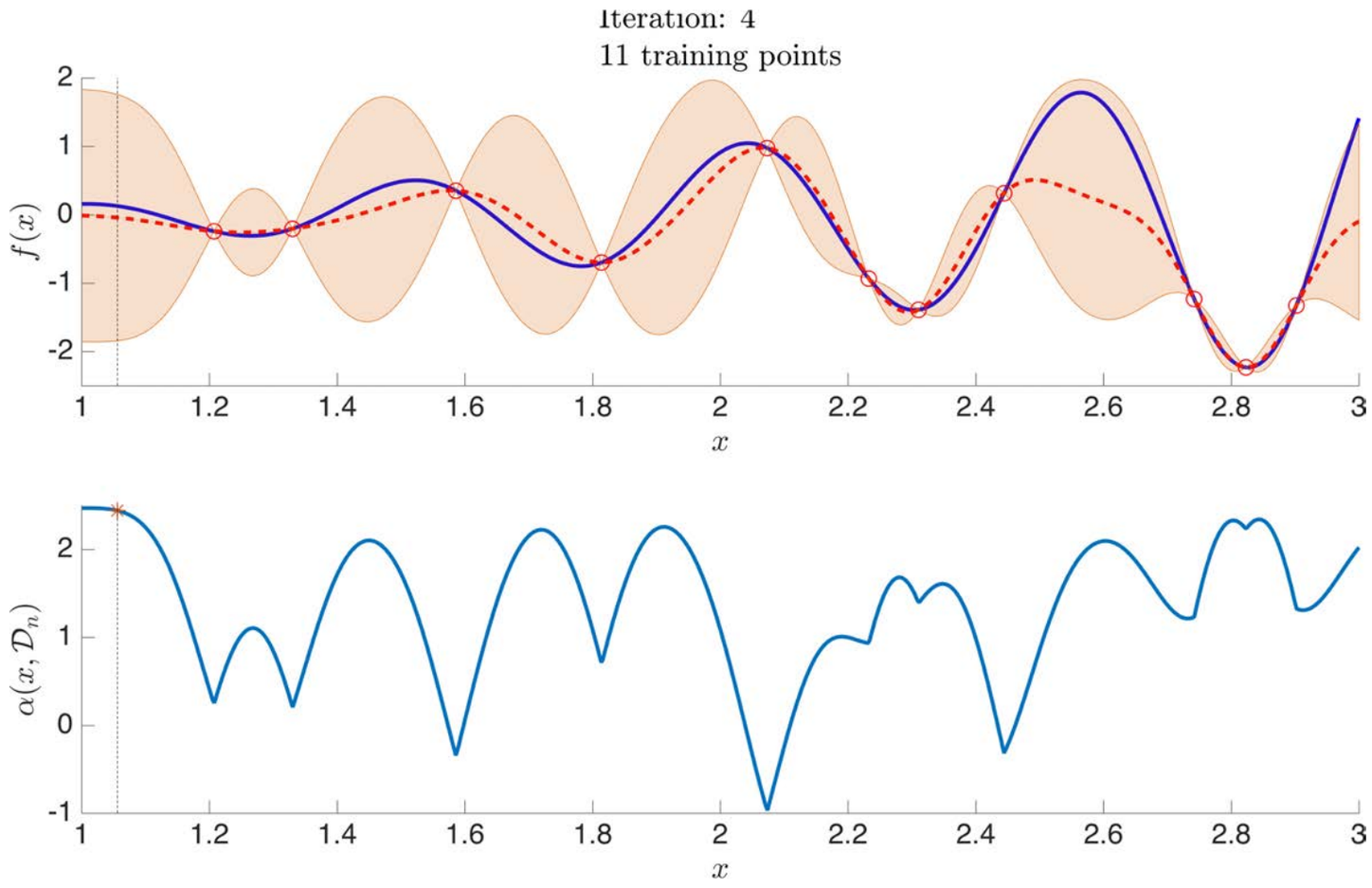
$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_\theta \mathbb{E}_{v\,|\,\mathbf{x},\theta}[U(\mathbf{x}, v, \theta)]$$

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 super-quintile risk confidence bound

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

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

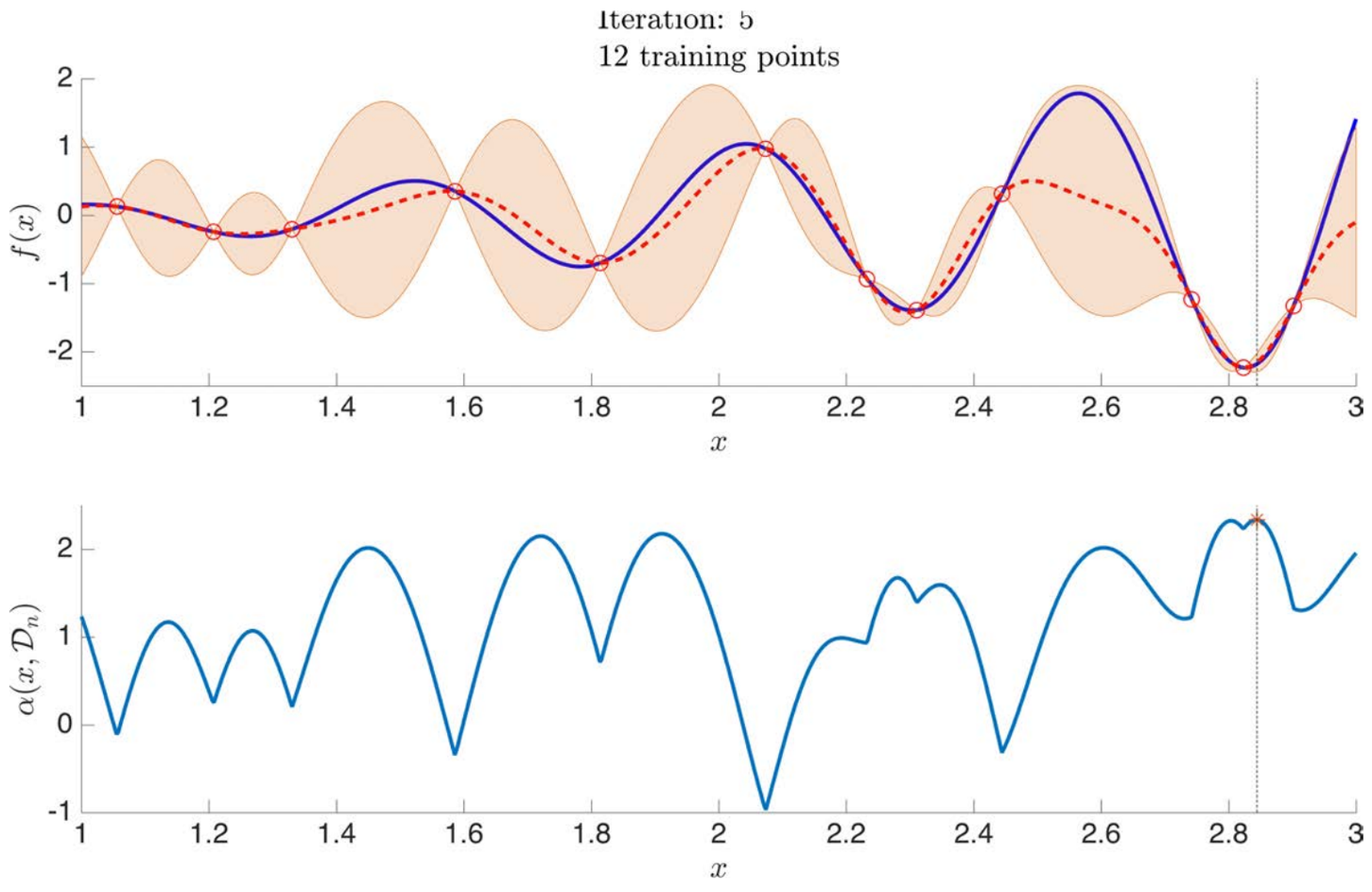
$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_\theta \mathbb{E}_{v\,|\,\mathbf{x},\theta}[U(\mathbf{x}, v, \theta)]$$

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 super-quintile risk confidence bound

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

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

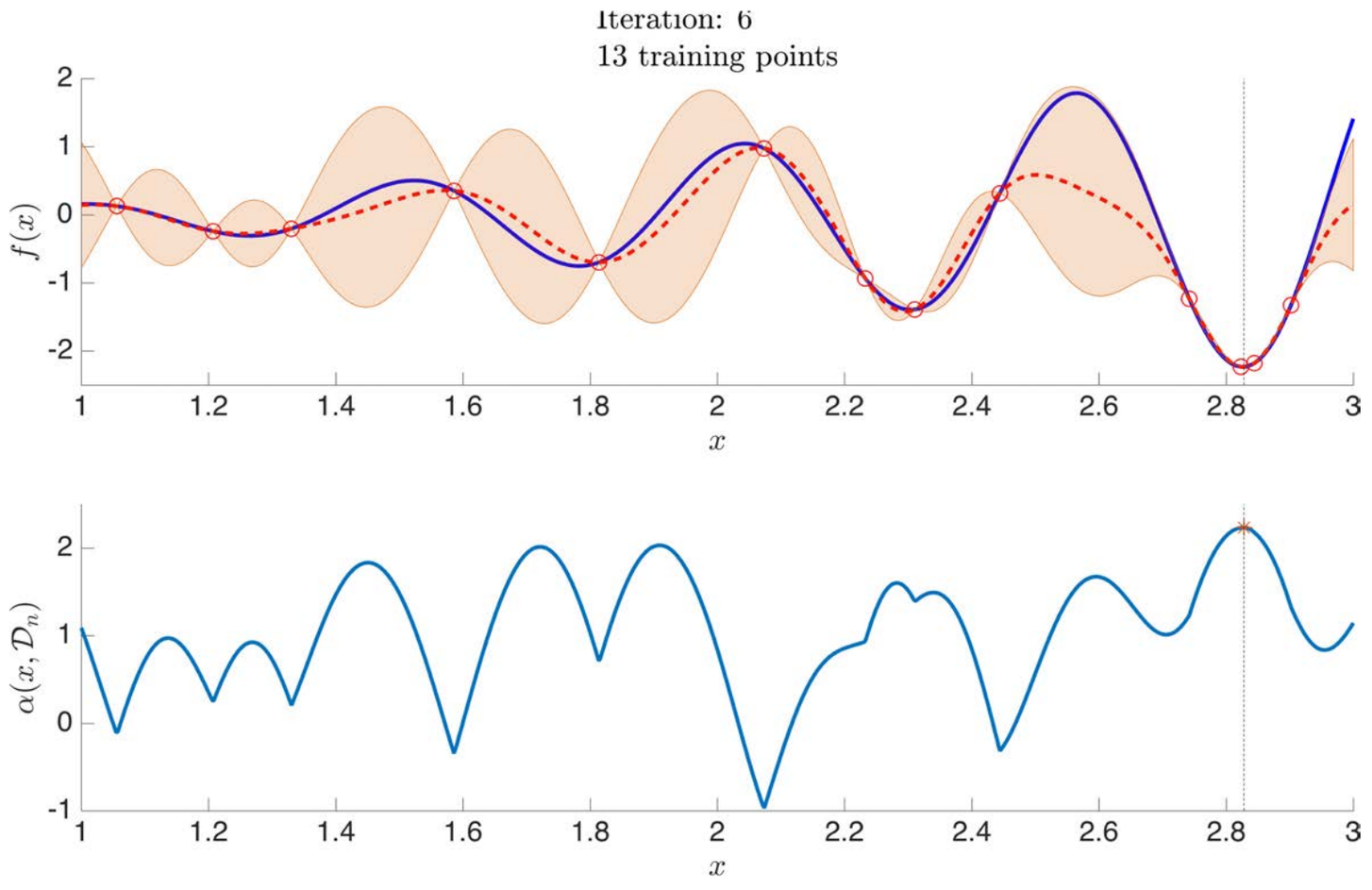
$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_\theta \mathbb{E}_{v\,|\,\mathbf{x},\theta}[U(\mathbf{x}, v, \theta)]$$

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 super-quintile risk confidence bound

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

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

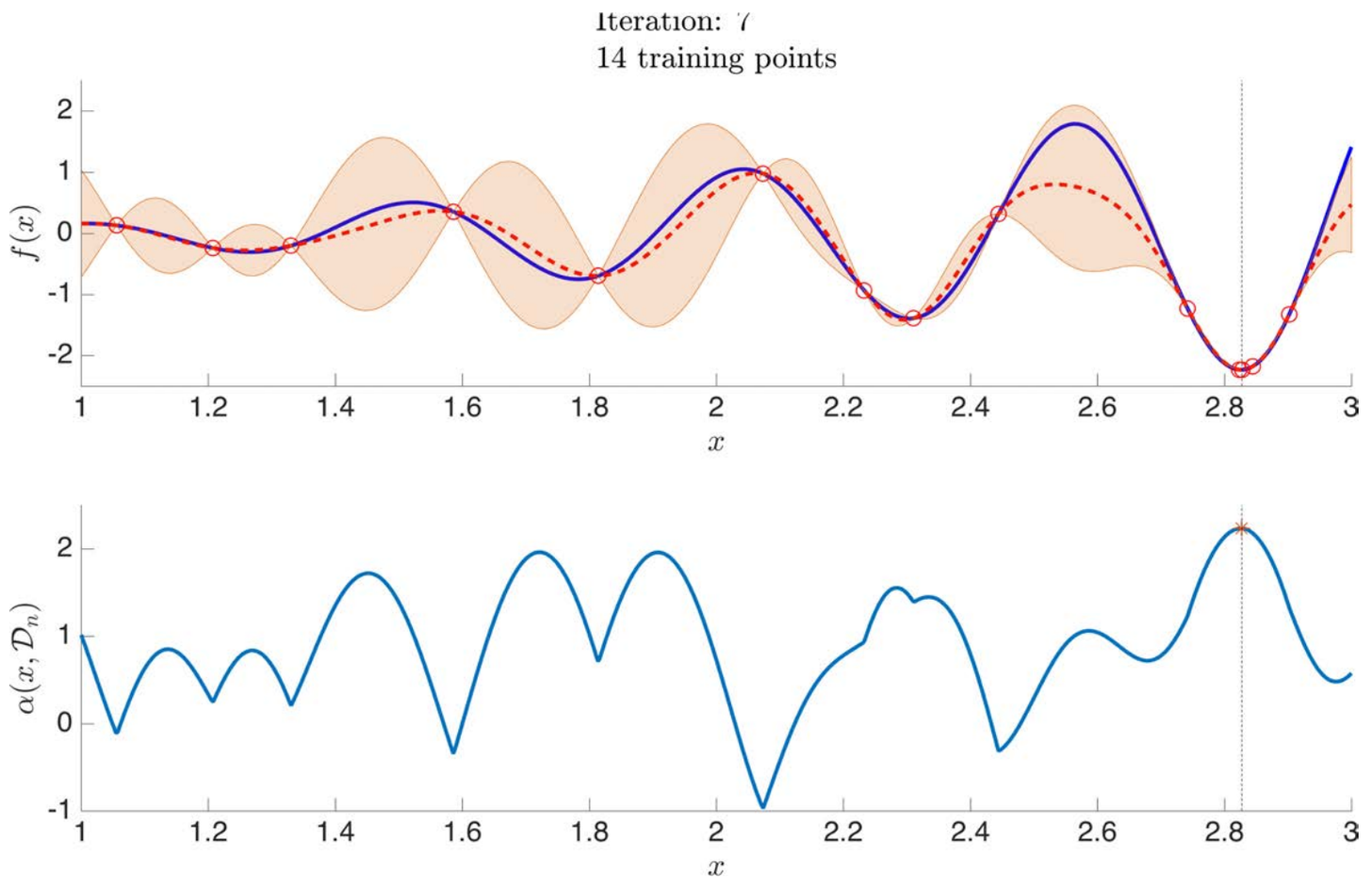
$$\alpha(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_\theta \mathbb{E}_{v\,|\,\mathbf{x},\theta}[U(\mathbf{x}, v, \theta)]$$

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 super-quintile risk confidence bound

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

Learning level sets

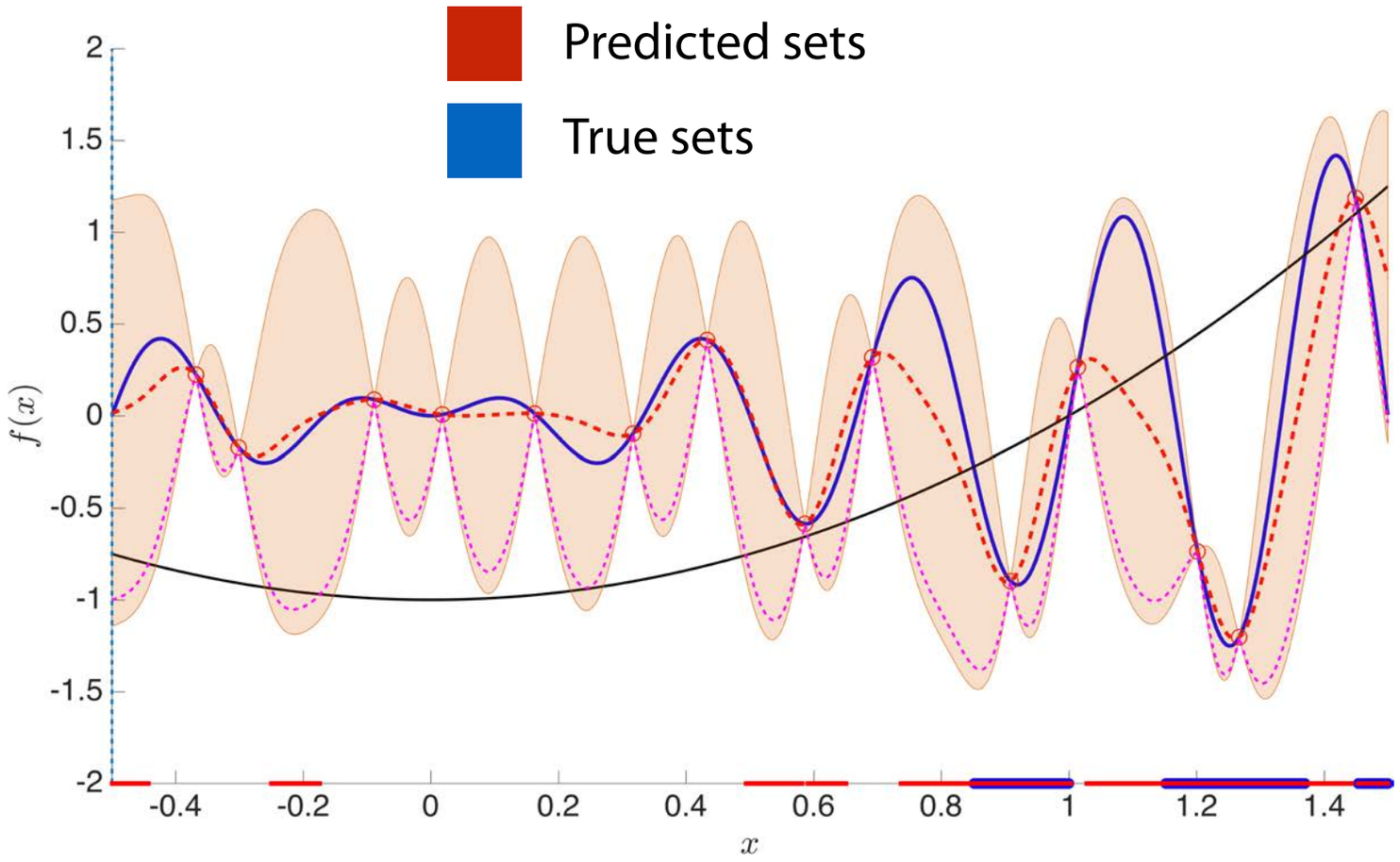
Goal: Identify the sets $L_\alpha(t) = \{\boldsymbol{x} : \mathcal{R}_\alpha(f(\boldsymbol{x})) \leq t(\boldsymbol{x})\}$

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 sets are guaranteed to be a subset of the true sets.

Learning level sets

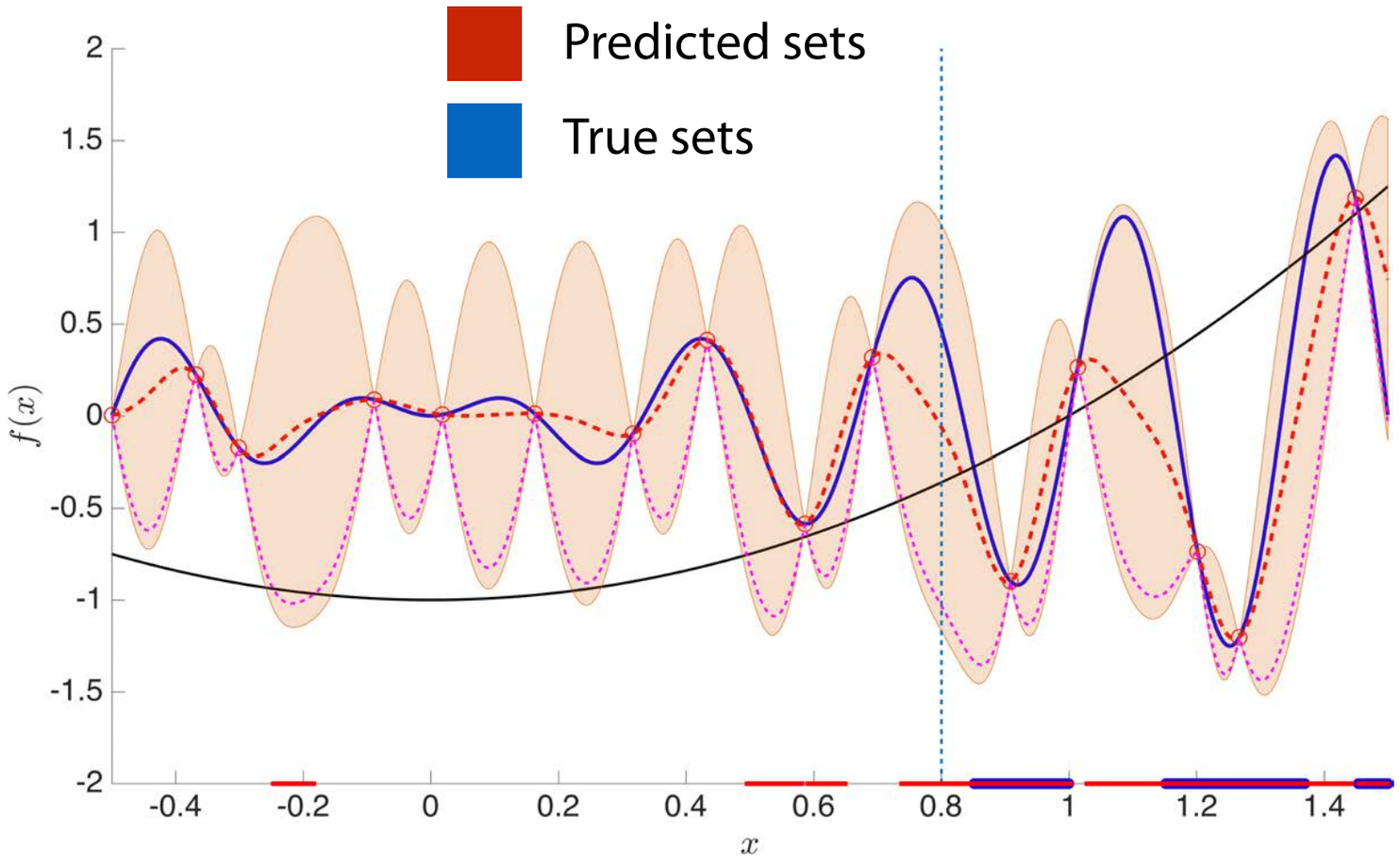
Goal: Identify the sets $L_\alpha(t) = \{\boldsymbol{x} : \mathcal{R}_\alpha(f(\boldsymbol{x})) \leq t(\boldsymbol{x})\}$

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 sets are guaranteed to be a subset of the true sets.

Learning level sets

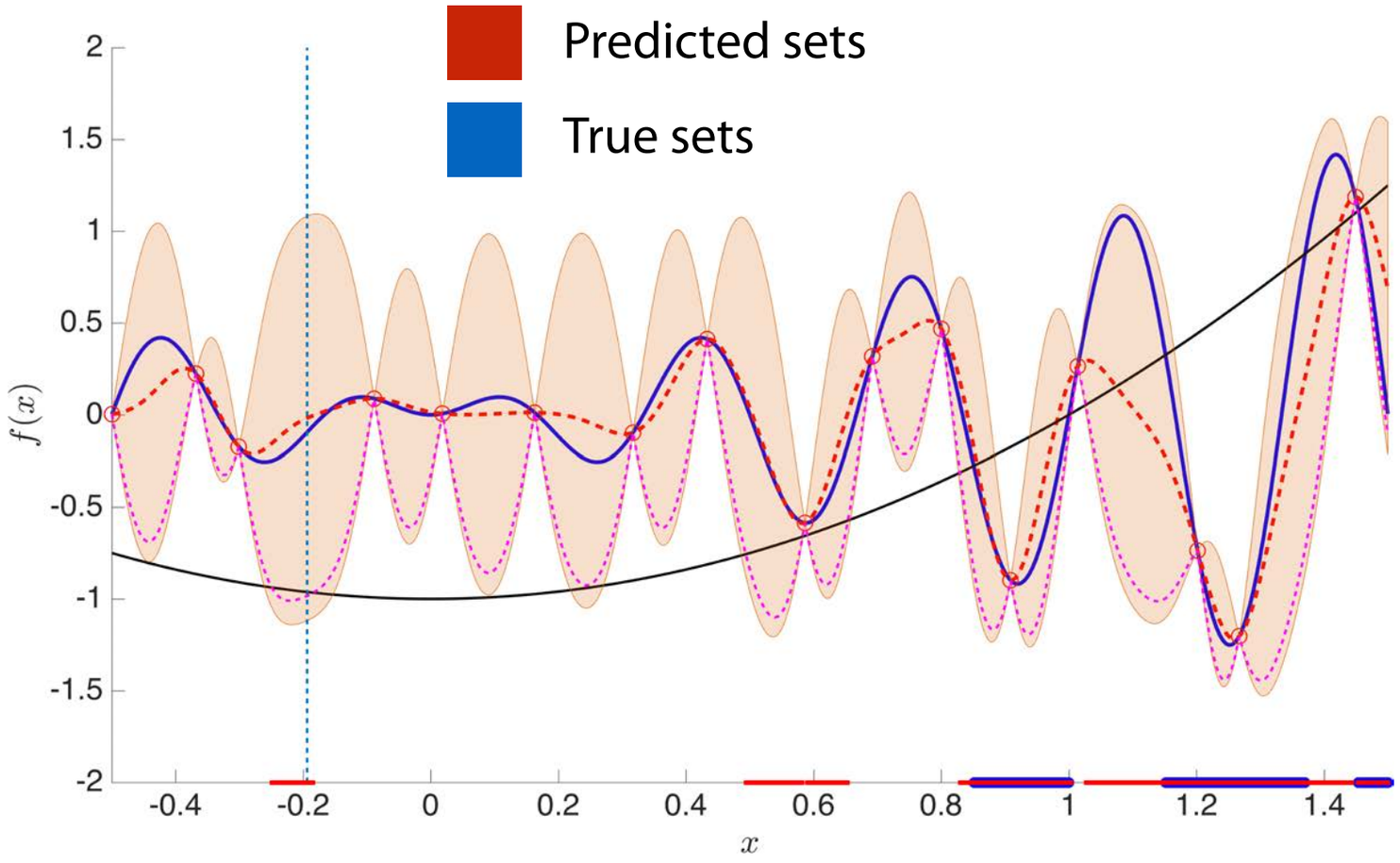
Goal: Identify the sets $L_\alpha(t) = \{\boldsymbol{x} : \mathcal{R}_\alpha(f(\boldsymbol{x})) \leq t(\boldsymbol{x})\}$

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 sets are guaranteed to be a subset of the true sets.

Learning level sets

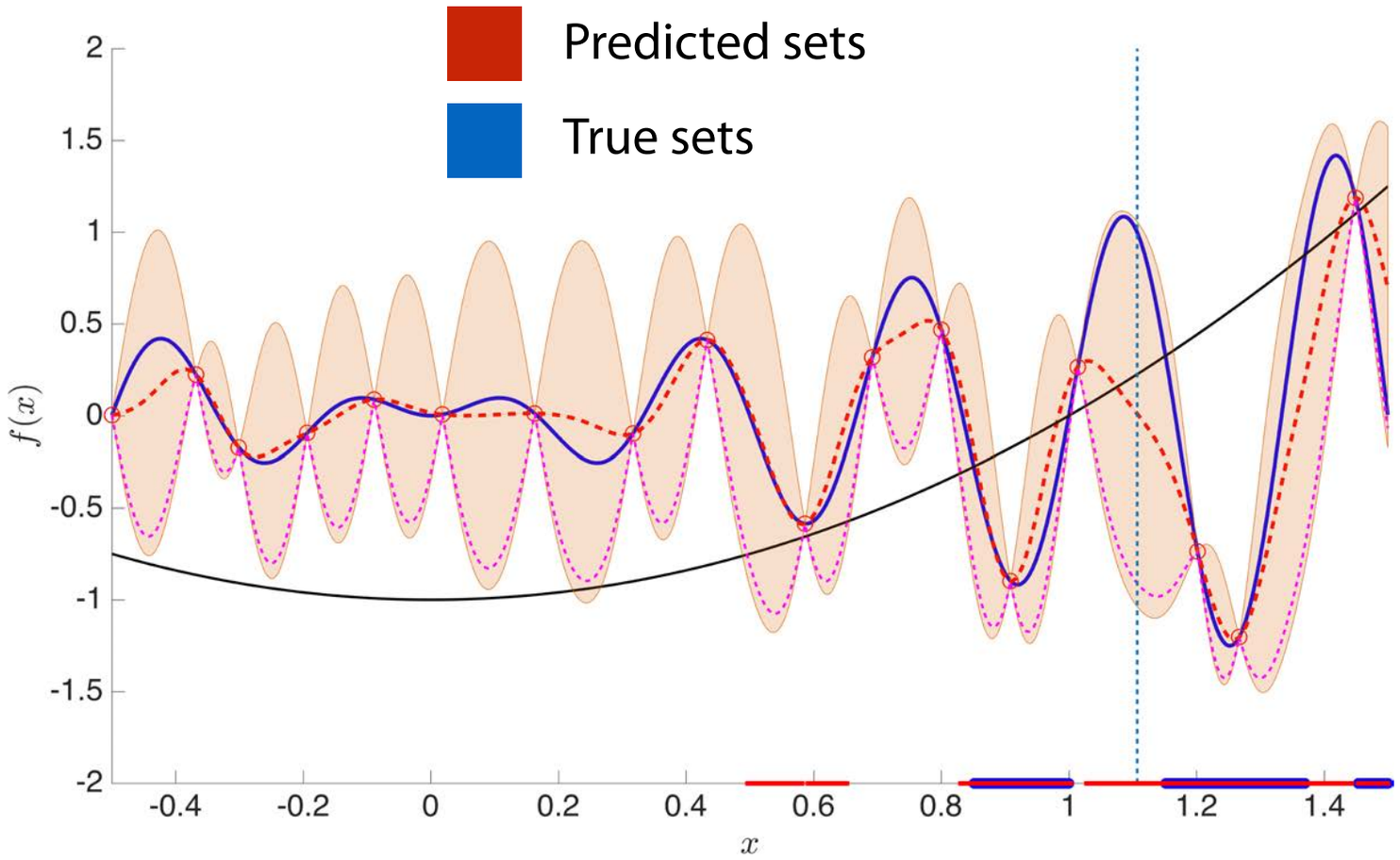
Goal: Identify the sets $L_\alpha(t) = \{\boldsymbol{x} : \mathcal{R}_\alpha(f(\boldsymbol{x})) \leq t(\boldsymbol{x})\}$

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 sets are guaranteed to be a subset of the true sets.

Learning level sets

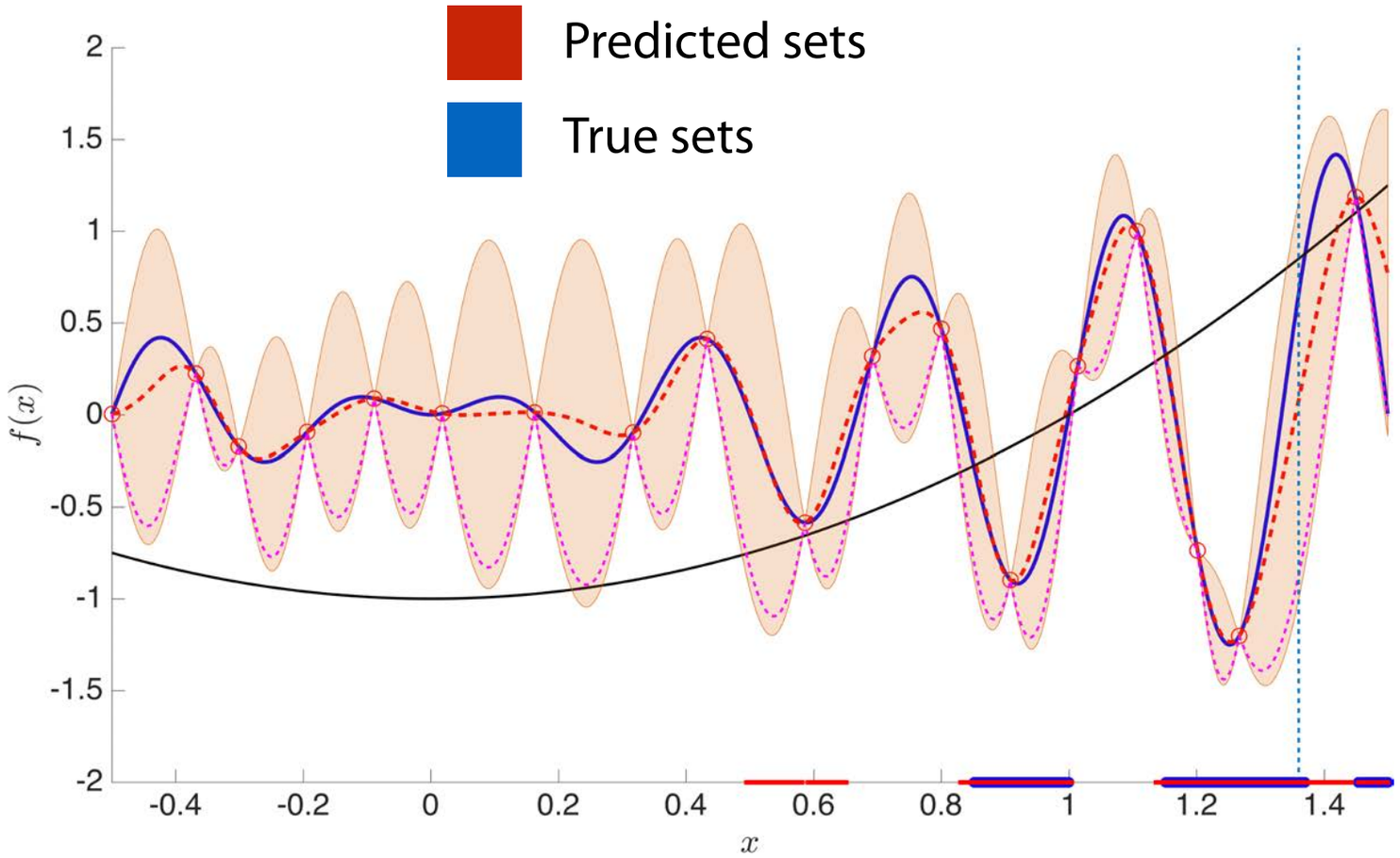
Goal: Identify the sets $L_\alpha(t) = \{\boldsymbol{x} : \mathcal{R}_\alpha(f(\boldsymbol{x})) \leq t(\boldsymbol{x})\}$

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 sets are guaranteed to be a subset of the true sets.

Learning level sets

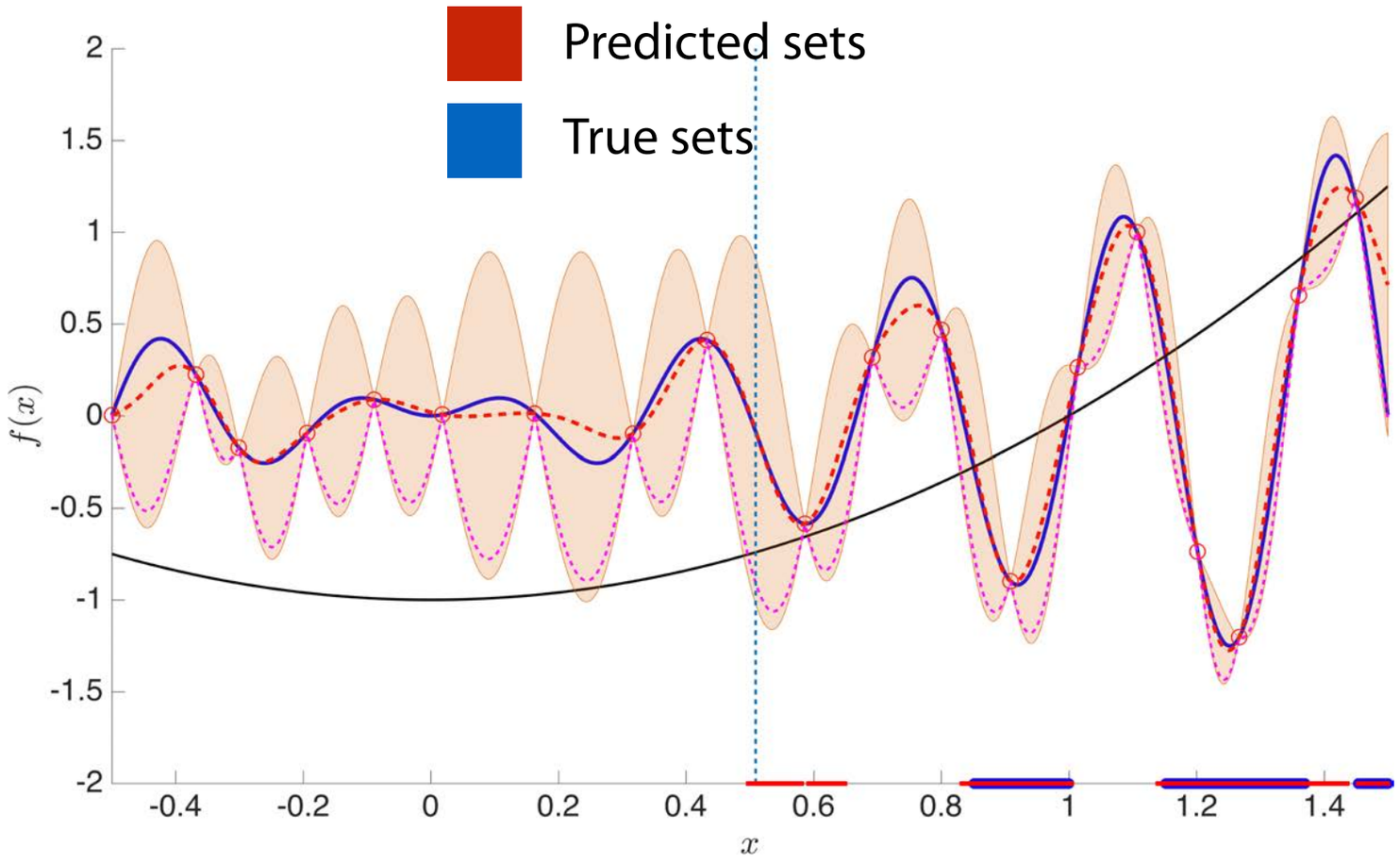
Goal: Identify the sets $L_\alpha(t) = \{\boldsymbol{x} : \mathcal{R}_\alpha(f(\boldsymbol{x})) \leq t(\boldsymbol{x})\}$

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 sets are guaranteed to be a subset of the true sets.

Learning level sets

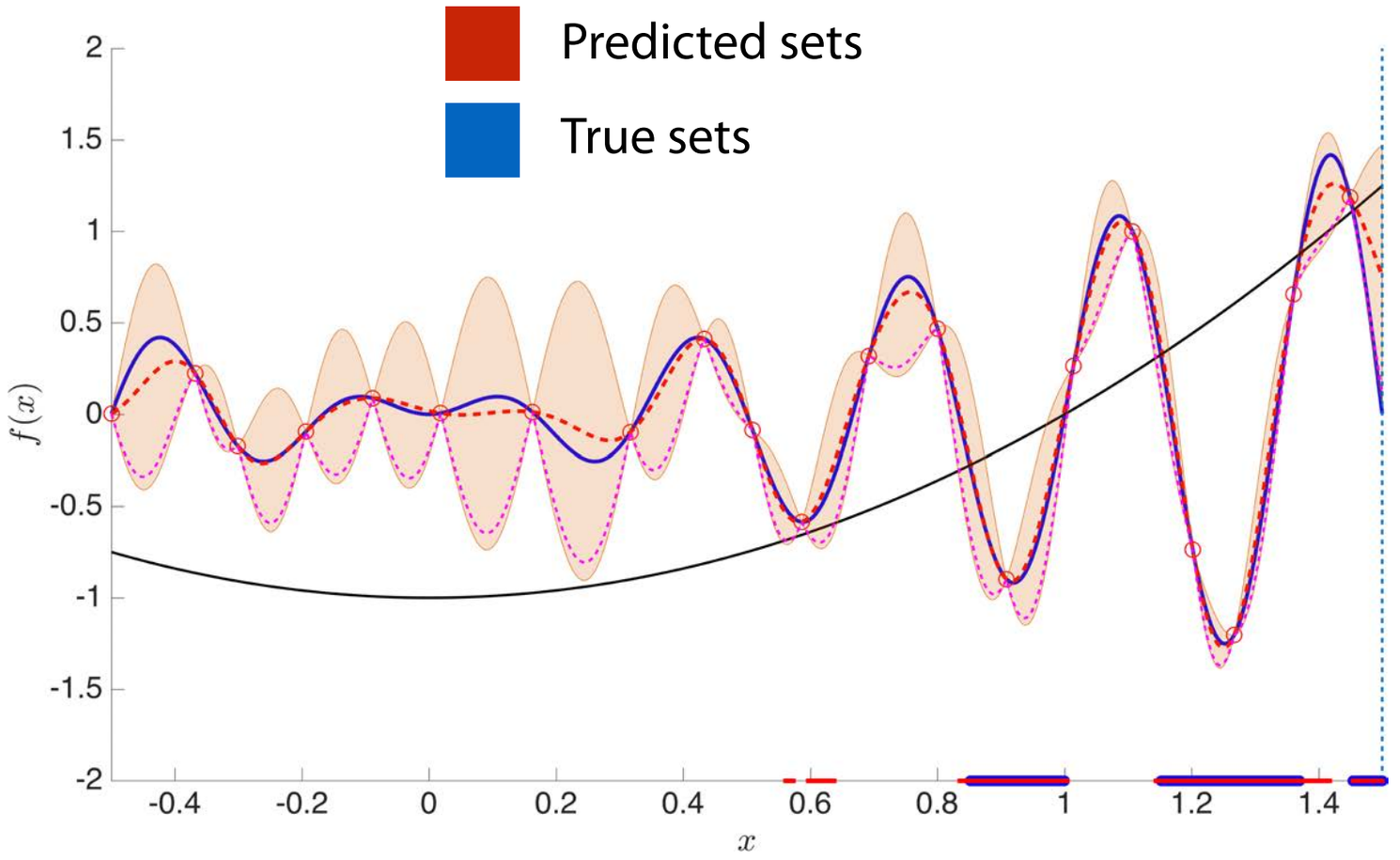
Goal: Identify the sets $L_\alpha(t) = \{\boldsymbol{x} : \mathcal{R}_\alpha(f(\boldsymbol{x})) \leq t(\boldsymbol{x})\}$

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 sets are guaranteed to be a subset of the true sets.

Learning level sets

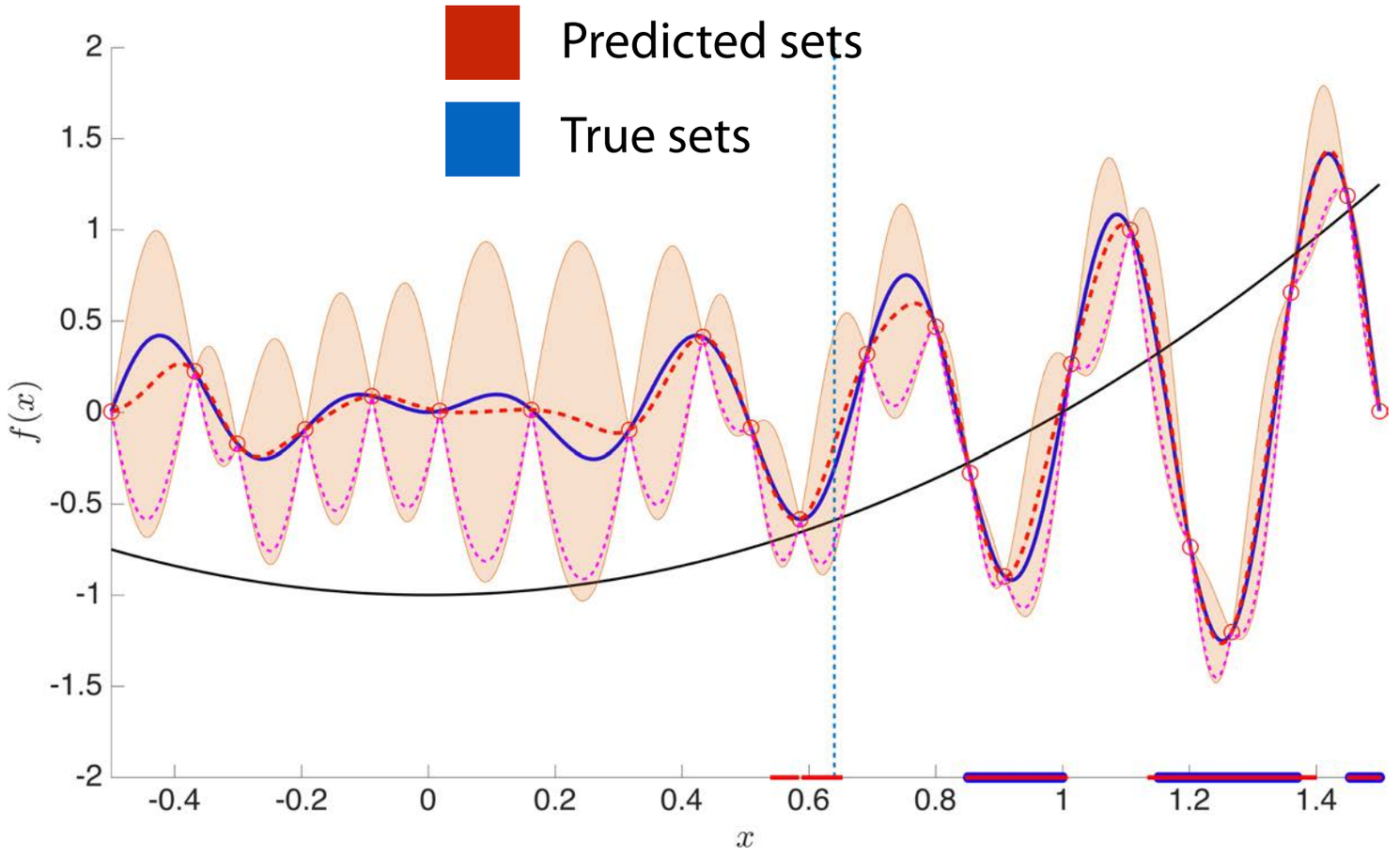
Goal: Identify the sets $L_\alpha(t) = \{\boldsymbol{x} : \mathcal{R}_\alpha(f(\boldsymbol{x})) \leq t(\boldsymbol{x})\}$

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 sets are guaranteed to be a subset of the true sets.

Learning level sets

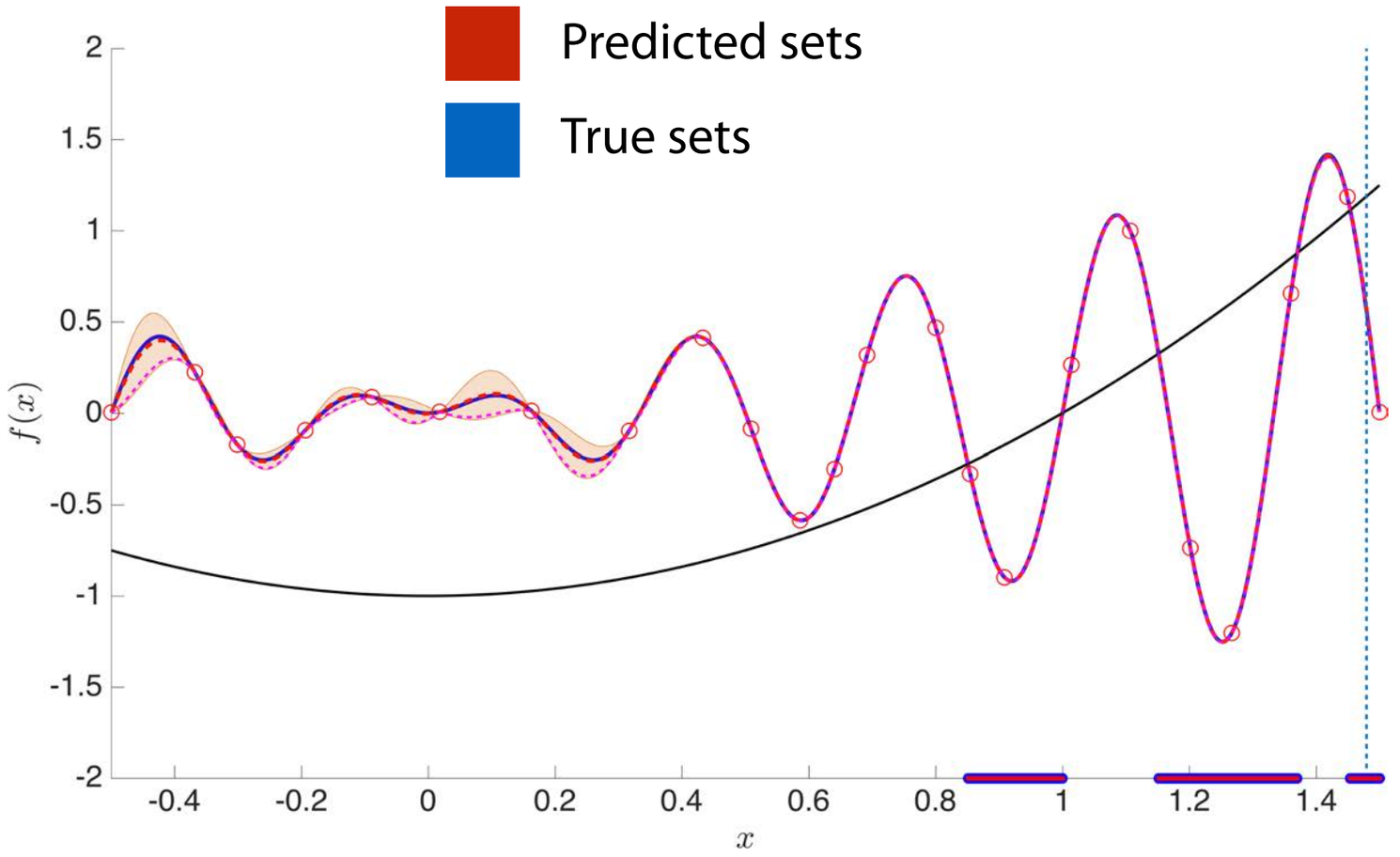
Goal: Identify the sets $L_\alpha(t) = \{\boldsymbol{x} : \mathcal{R}_\alpha(f(\boldsymbol{x})) \leq t(\boldsymbol{x})\}$

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 sets are guaranteed to be a subset of the true sets.