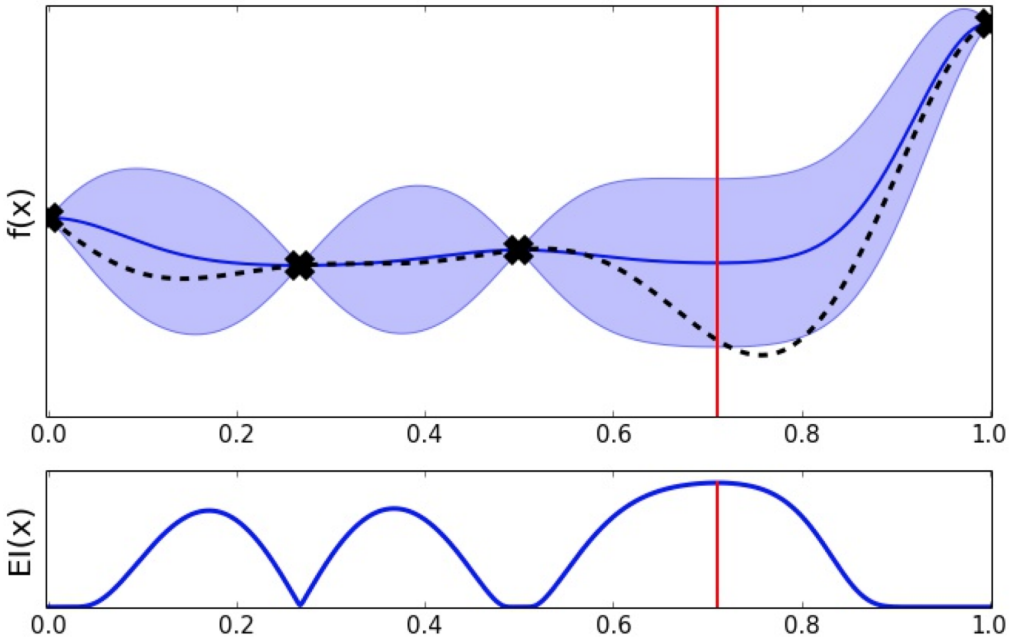
Bayesian optimization: theory and practice

A. Zaytsev

Sber,
Moscow

4 June, 2021



Selection of the next point to evaluate the target function with Bayesian optimization

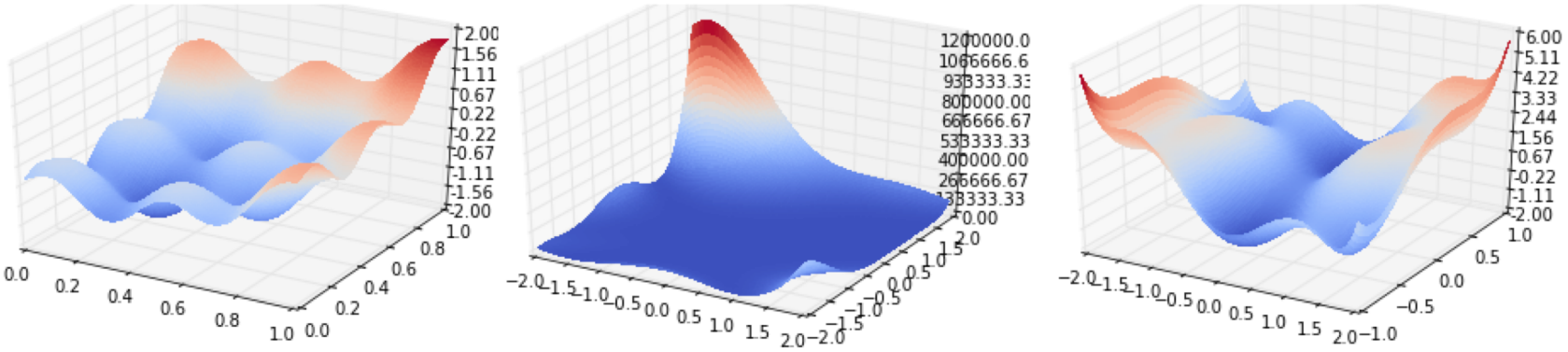
Bayesian optimization

Slides in this section are mostly based on slides by J. Gonzalez and E. Burnaev

Global optimization

Consider a "well behaved" (e.g. smooth) function $f : \mathcal{X} \to \mathbb{R}$, with $\mathcal{X} \subseteq \mathbb{R}^d$ being a compact set

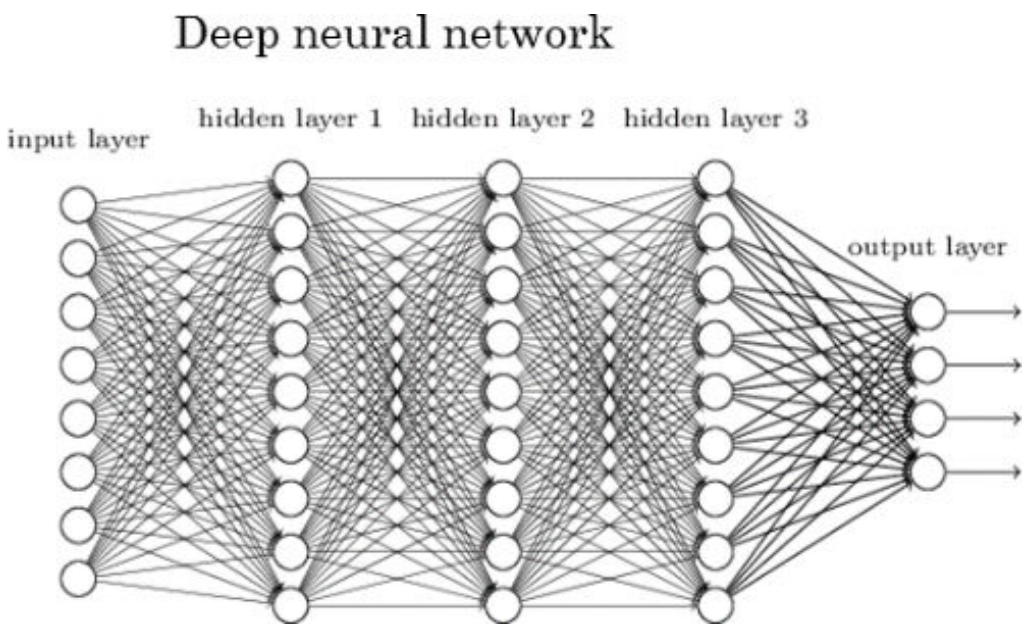
$$\mathbf{x}_{\min} = \arg\min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$$



- f is explicitly unknown and multimodal
- Only to the noised evaluations of f available
- Each evaluation of f is expensive

http://gpss.cc/gpss18/slides/Gonzalez2018.pdf

Expensive functions are common in ML



- Inputs $\mathbf{x}$:
  - Number of layers/units per layer
  - Types of each layer
  - Regularization coefficients
  - Learning rates, etc.
- Output $f(\mathbf{x})$: error rate $(1 - \text{accuracy})$

http://gpss.cc/gpss18/slides/Gonzalez2018.pdf

Common options for global optimization

Option 1: Use previous knowledge

Option 2: Grid search? f is L-Lipchitz continuous and we are in a noise-free domain. To propose $\mathbf{x}_n$ such that

$$f(\mathbf{x}_{\min}) - f(\mathbf{x}_n) \leq \varepsilon$$

we need to evaluate on a d-dimensional unit hypercube: $n = (L/\varepsilon)^d$ evaluations!

$$(10/0.01)^5 = 10^{15},$$

but function evaluations are expensive!

Option 3: Sample uniformly [Bergsta, Bengio, 2012, JMLR]

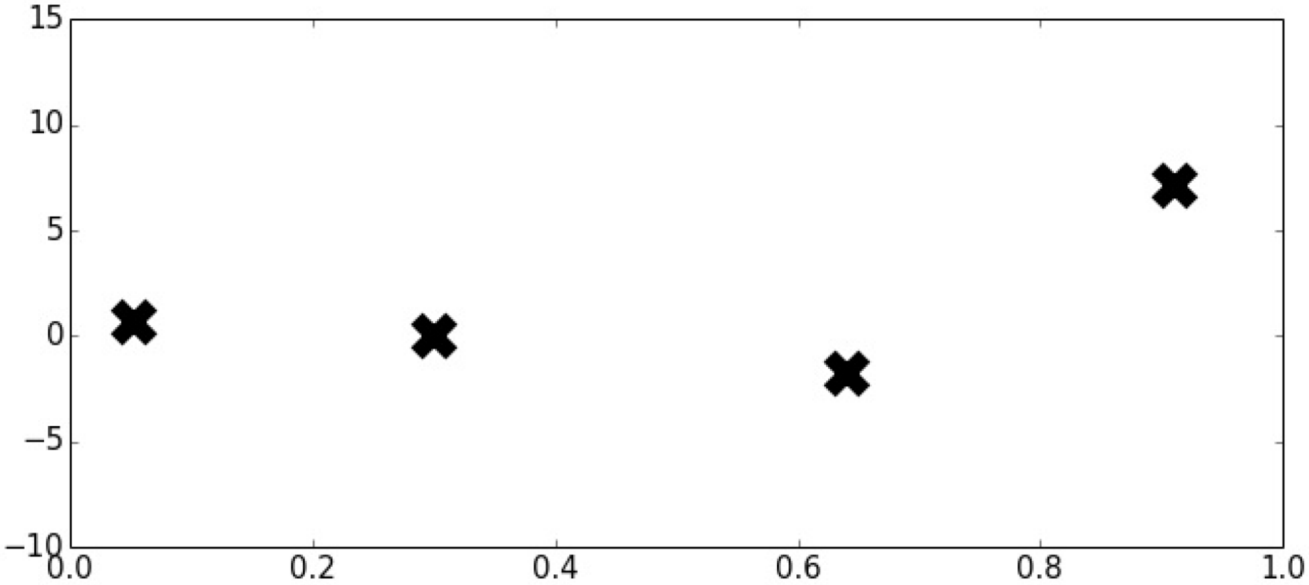
http://gpss.cc/gpss18/slides/Gonzalez2018.pdf

Problem

Select the next point for evaluation to get better understanding of

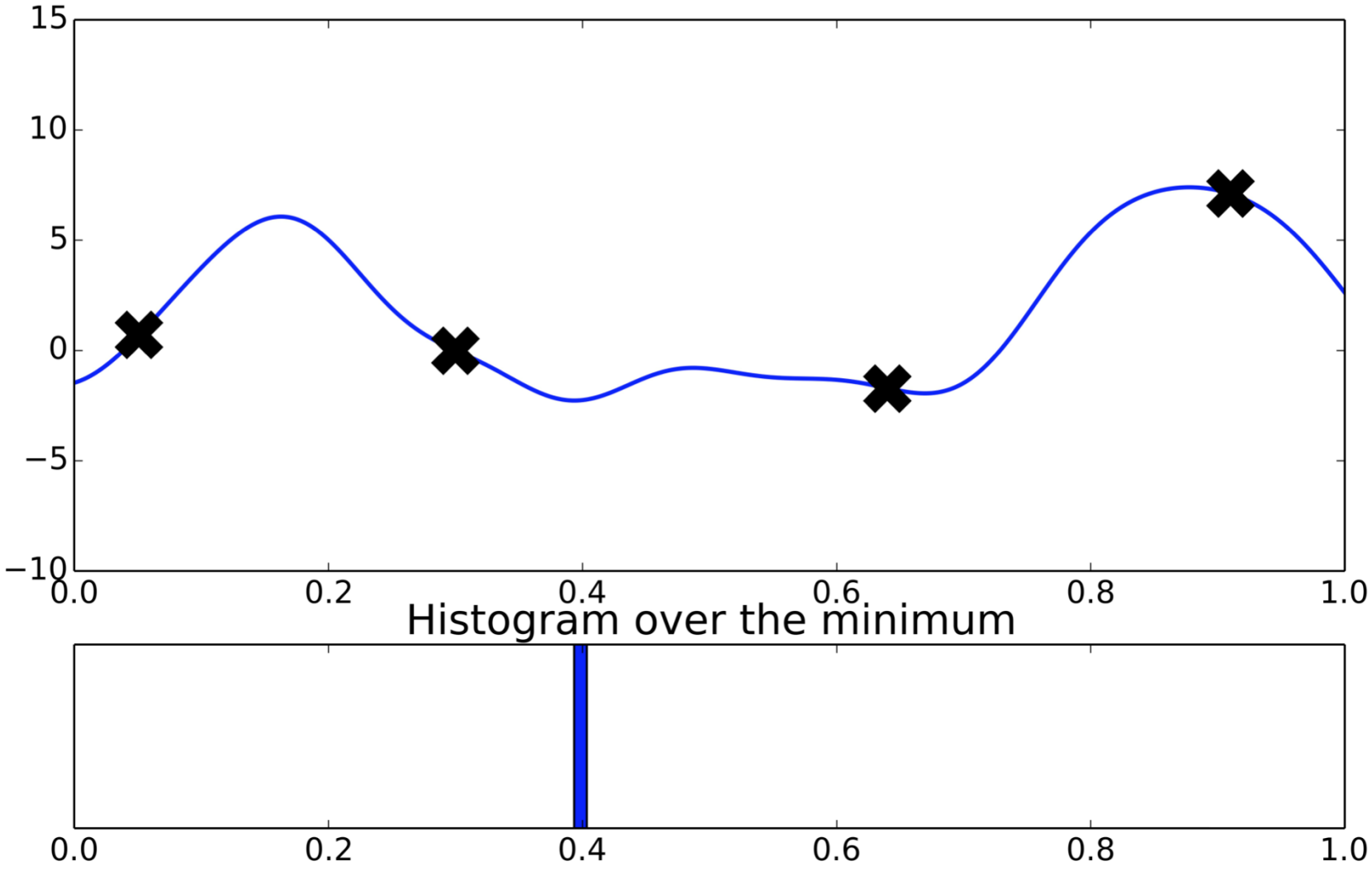
- where is the minimum value of a fuction y_{best};
- what is this value s.t. $f(\mathbf{x}_{\min}) = y_{best}$.

Example one dimensional problem

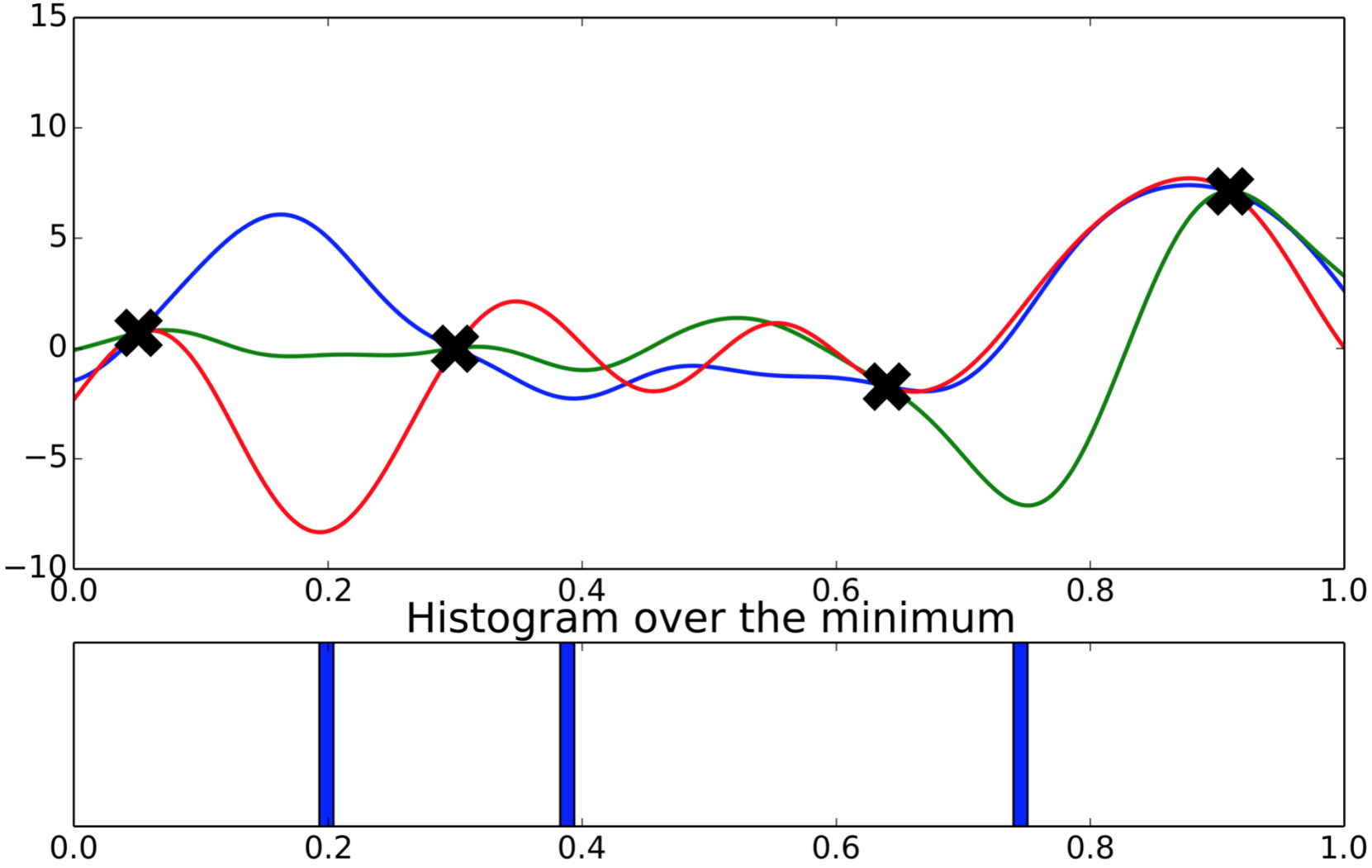


Where is the minimum of f? Where should the take the next evaluation?

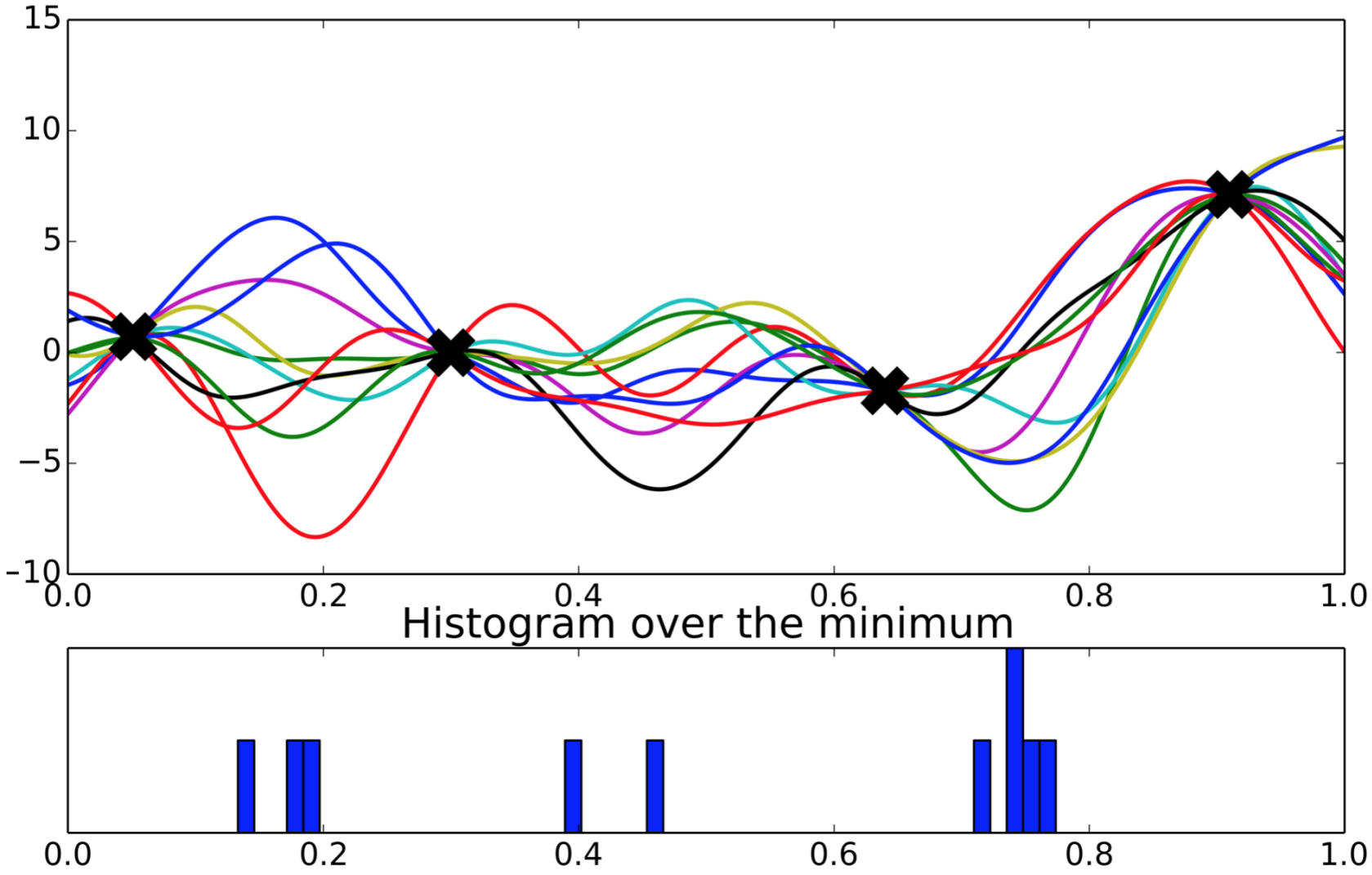
Intuition: one curve from our prior over functions



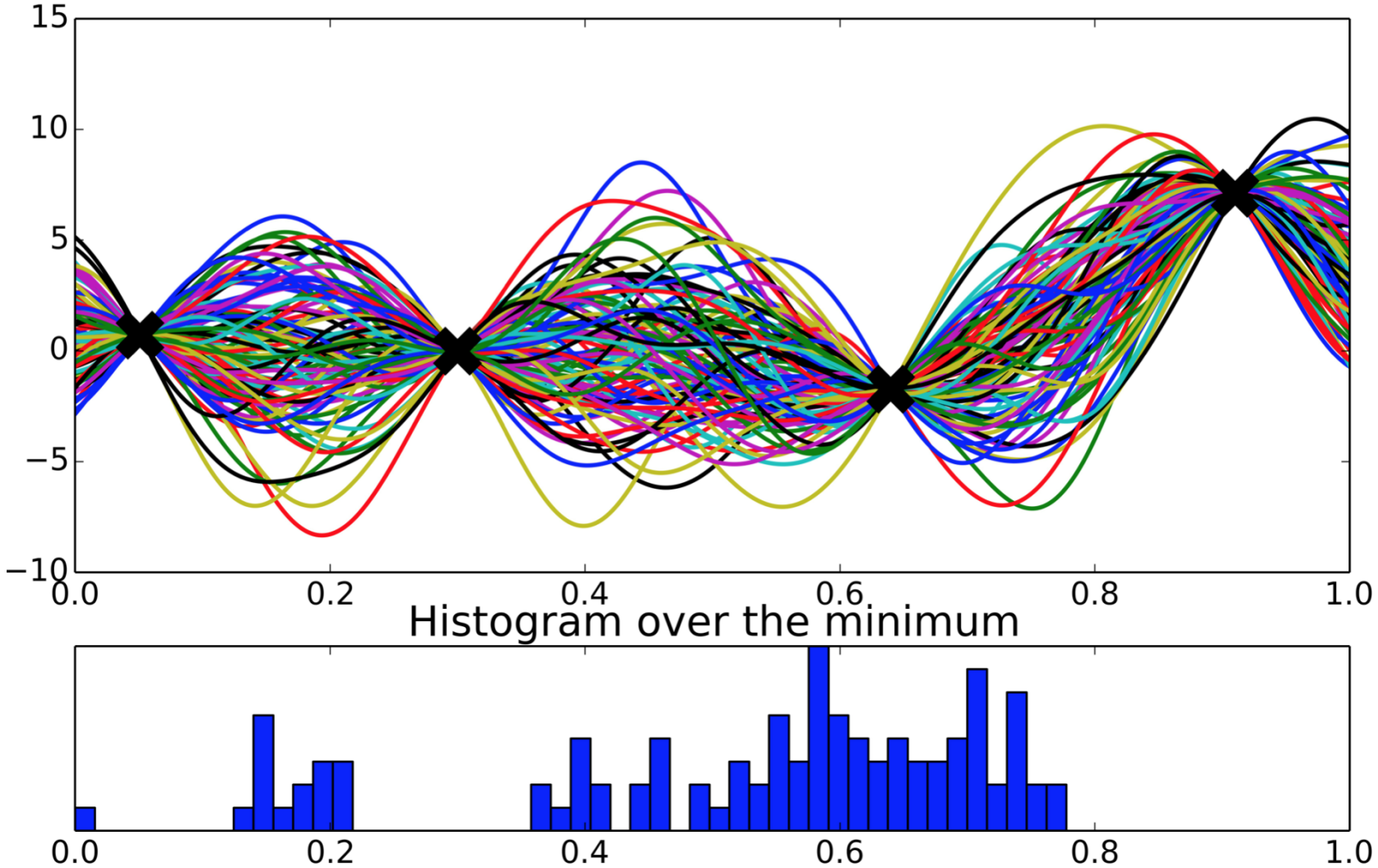
Intuition: three curves



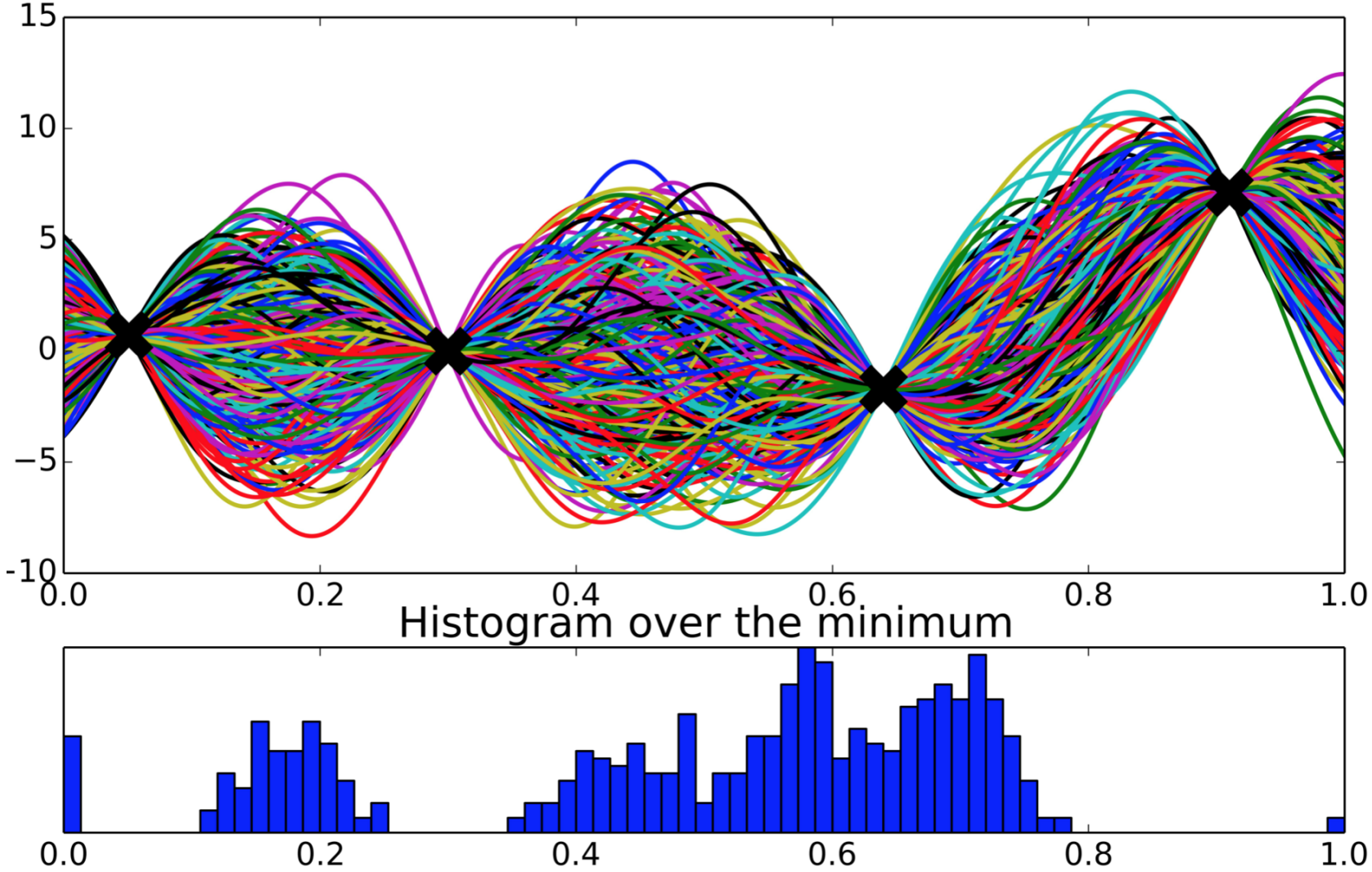
Intuition: ten curves



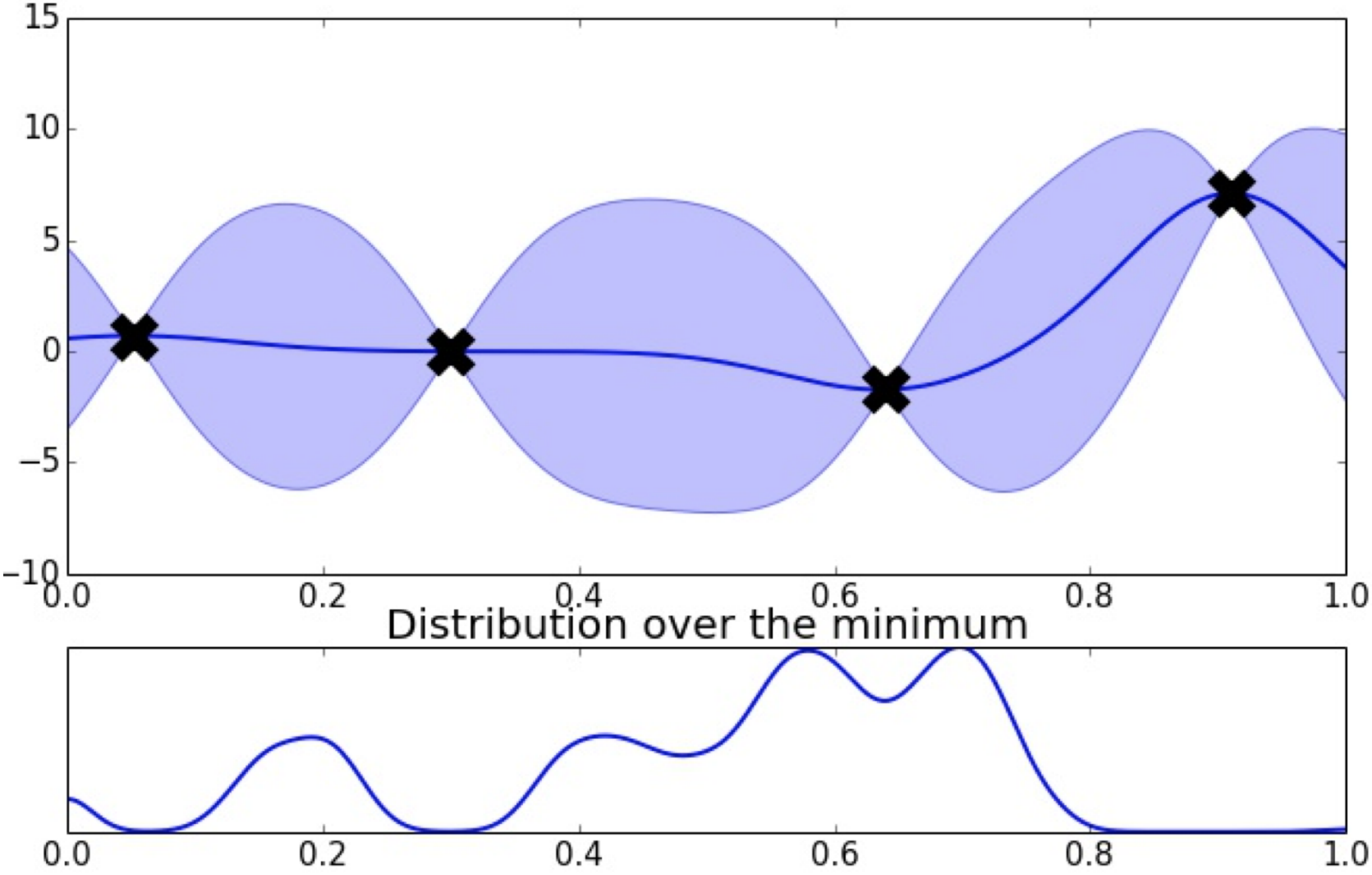
Intuition: hundred curves



Intuition: many curves



Intuition: infinite number of curves



General surrogate modeling

- Use a surrogate model $\widehat{f}$ of f to carry out the optimization
- Select a new point for evaluation of f according to some criterion.
  - We believe that $\widehat{f} \approx f$,

$$\mathbf{x}_{n+1} = \arg\min \widehat{f}(\mathbf{x}).$$

  - We want to explore $\mathcal{X}$: active learning,

$$\mathbf{x}_{n+1} = \arg\max \sigma_{\widehat{f}}(\mathbf{x}).$$

Acquisition functions

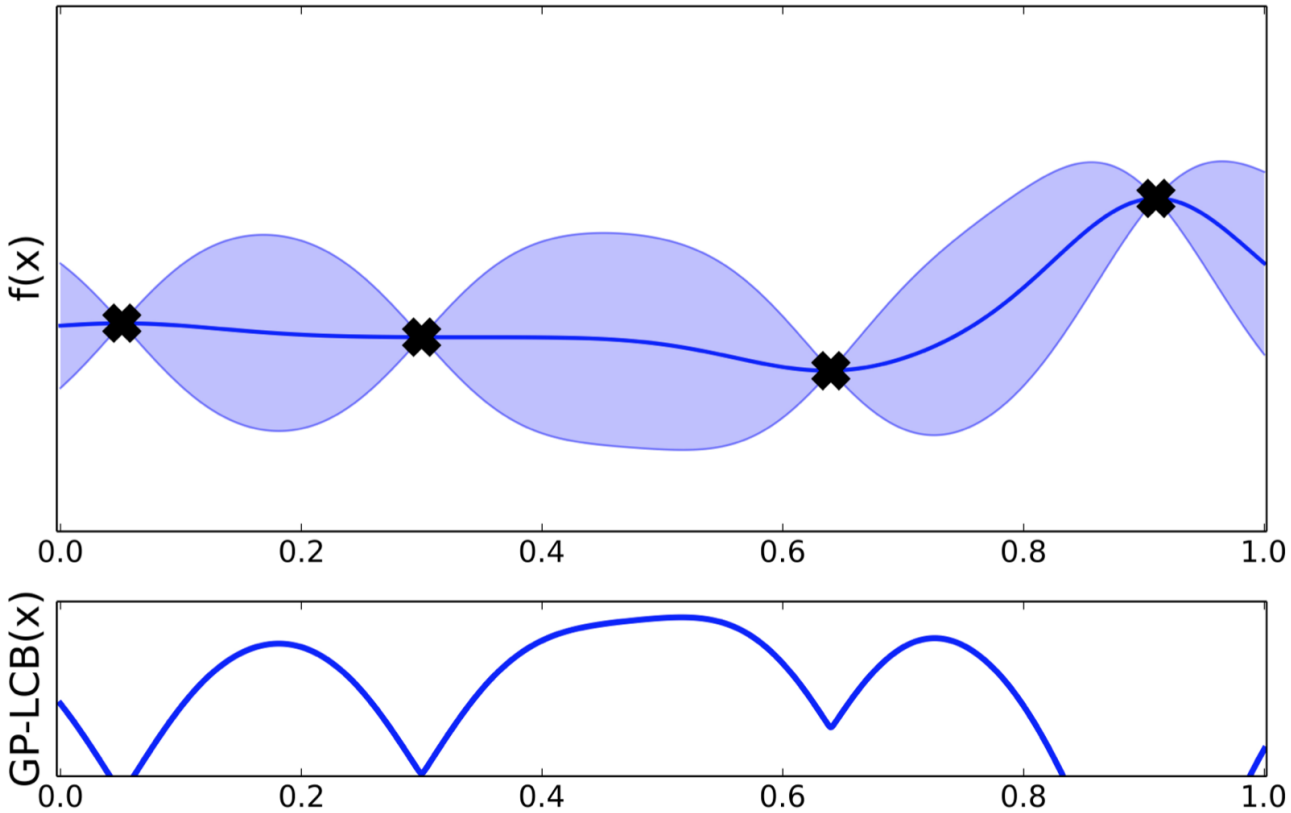
- Use GP $\mathcal{GP}(\cdot|\mu(\mathbf{x}), K(\mathbf{x}, \mathbf{x}'))$ as a prior for $f(\cdot)$
- GP has marginal closed-form for the posterior mean $\mu_*(\mathbf{x})$ and variance $\sigma_*^2(\mathbf{x})$ $\Rightarrow$ efficient calculation of acquisition function
  - **Exploration**: Evaluate in places where the variance is large
  - **Exploitation**: Evaluate in places where the mean is low

Acquisition functions balance these two factors to determine where to evaluate next

GP Upper (lower) Confidence Bound

Direct balance between exploration and exploitation (ζ is a user-defined parameter):

$$\alpha_{LCB}(\mathbf{x}) = -\mu_*(\mathbf{x}) + \zeta \cdot \sigma_*(\mathbf{x})$$



GP Upper Confidence Bound

- In noiseless cases, it is a lower bound of the function to minimize.
- This allows to computer a bound on how close we are to the minimum.
- Exact result exists (I hope we'll have time for it later):

Theorem 1 *Let* $\delta \in (0,1)$ *and* $\beta_t = 2\log(|D|t^2\pi^2/6\delta)$. *Running* GP-UCB *with* β_t *for a sample* f *of a GP with mean function zero and covariance function* $k(\boldsymbol{x}, \boldsymbol{x}')$, *we obtain a regret bound of* $\mathcal{O}^*(\sqrt{T\gamma_T \log|D|})$ *with high probability. Precisely,*

$$\Pr\left\{R_T \leq \sqrt{C_1 T \beta_T \gamma_T} \quad \forall T \geq 1\right\} \geq 1 - \delta.$$

where $C_1 = 8/\log(1+\sigma^{-2})$.

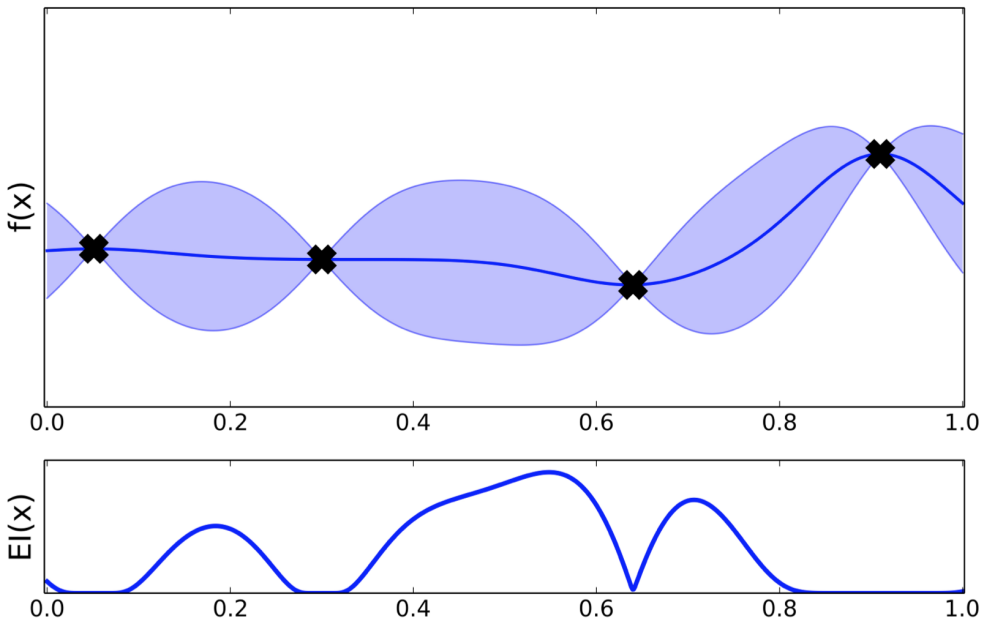
Srinivas, Niranjan, et al. "Gaussian process optimization in the bandit setting: no regret and experimental design." ICML. 2010.

Expected Improvement

Let us denote by $\Delta(\mathbf{x}) = y_{\text{best}} - \mu_*(\mathbf{x})$, then

$$\alpha_{EI}(\mathbf{x}) = \mathbb{E}\Delta(\mathbf{x})_+ \int_y \max(0, y_{\text{best}} - y_*) p(y_*|\mathbf{x}) dy_* =$$

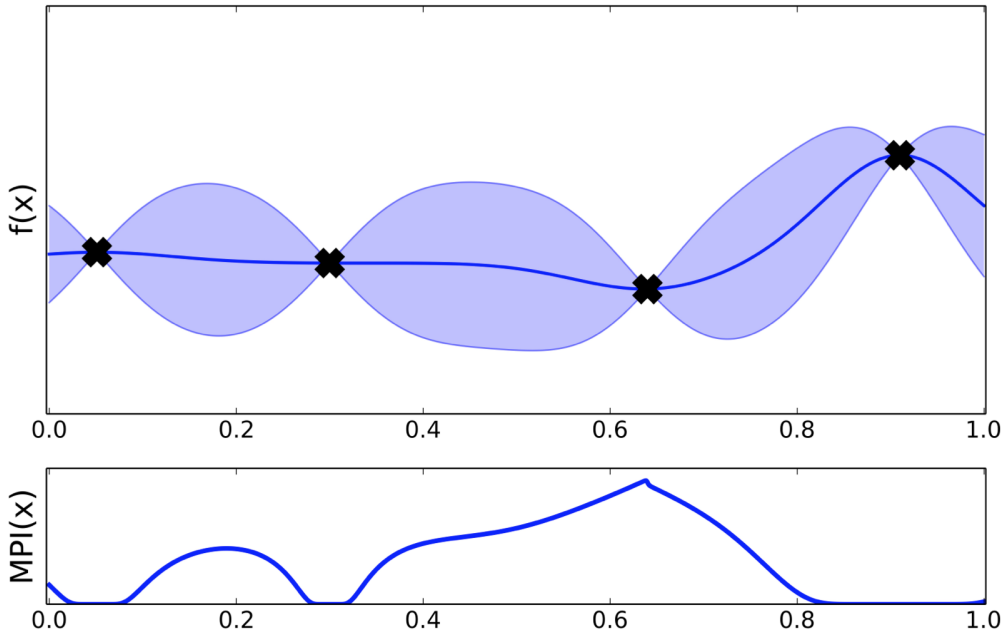
$$= \max(0, \Delta(\mathbf{x})) - \sigma_*(\mathbf{x})\varphi\left(\frac{\Delta(\mathbf{x})}{\sigma_*(\mathbf{x})}\right) + |\Delta(\mathbf{x})|\Phi\left(-\frac{|\Delta(\mathbf{x})|}{\sigma_*(\mathbf{x})}\right)$$



Maximum Probability of Improvement

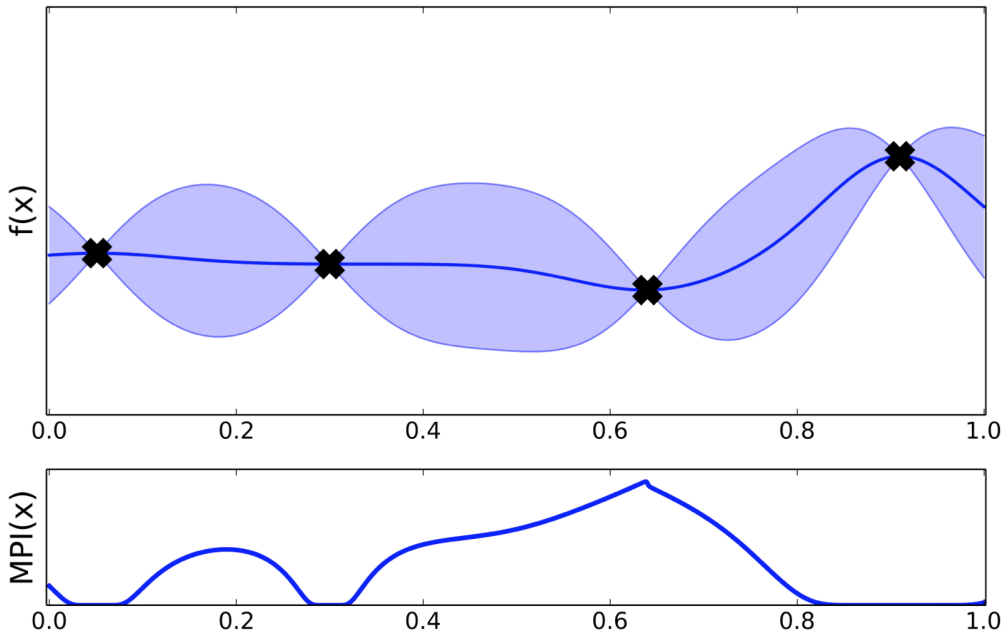
$$\gamma(\mathbf{x}) = \frac{\mu(\mathbf{x}) - y_{\text{best}}}{\sigma(\mathbf{x})}$$

$$\alpha_{MPI}(\mathbf{x}) = \mathbb{P}(f(\mathbf{x}) < y_{\text{best}}) = \Phi(\gamma(\mathbf{x}))$$



Information-theoretic approaches: Entropy search

$$\alpha_{ES}(\mathbf{x}) = \mathcal{H}[p(\mathbf{x}_{\min}|\mathbf{y})] - E_{p(y|\mathbf{y},\mathbf{x})}[\mathcal{H}[p(\mathbf{x}_{\min}|\mathbf{y} \cup \{\mathbf{x}, y\})]]$$



Bayesian Optimization

- BO is an strategy to transform the problem

$$\mathbf{x}_{\min} = \arg\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$$

unsolvable!

into a series of problems

$$\mathbf{x}_{t+1} = \arg\max_{\mathbf{x}\in\mathcal{X}} \alpha(\mathbf{x}|D_t),$$

solvable!

where

- $\alpha(\mathbf{x})$ is not so expensive to evaluate
- Gradients of $\alpha(\mathbf{x})$ are typically available
- Still need to find $\mathbf{x}_{t+1}$: DIRECT, gradient methods, SA

Formalization

- We made prior assumption about f
- Information about the minimum is now encoded in a new function (the probability distribution $p_{\min}$ over the minimum in this case)
- We can use $p_{\min}$ (or a functional of it) to decide where to sample next
- Other functions to encode relevant information about the minimum are possible, e. g. the "marginal expected gain" at each location.

Bayesian Optimization

Methodology to perform global optimization of multimodal black-box functions

1. Choose some prior measure over the space of possible objectives f
2. Combine prior and the likelihood to get a posterior over the objective given some observations
3. Use the posterior to decide where to take the next evaluation according to some acquisition function
4. Augment the data set

Iterate between 2 and 4 until the evaluation budget is over

Mockus, Jonas, Vytautas Tiesis, and Antanas Zilinskas. "The application of Bayesian methods for seeking the extremum." Towards global optimization 2.117-129 (1978): 2.

Practical issues for Bayesian optimization

Some slides in this section are based on slides by J. Gonzalez

What kind of surrogate do we need?

We need a surrogate $\widehat{f}(\mathbf{x})$:

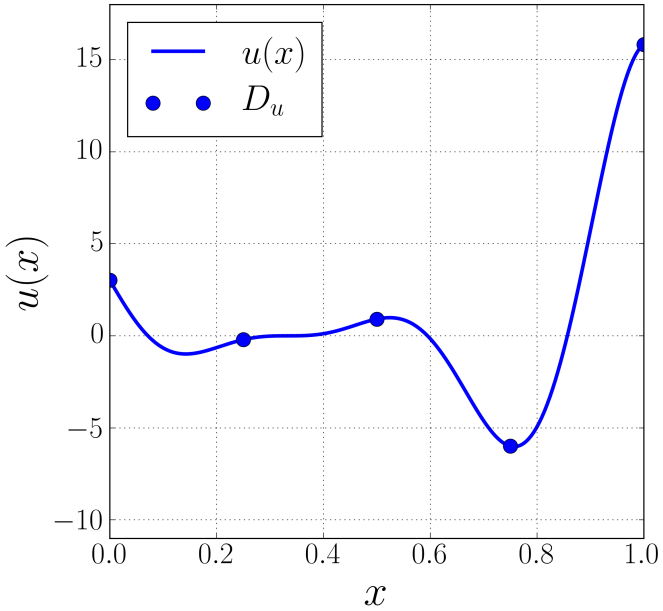
- Smooth (if no noise)
- Has probability density $p_{\mathbf{x}}(y)$ from posterior or at least uncertainty estimate
- Works with noise
- Nonlinear

...and we don't care too much about computational complexity, as f is expensive, so we have only a small sample of points.

Possible candidates: how they should look like

- We have a sample $D = (X, \mathbf{y}) = \{\mathbf{x}_i, y(\mathbf{x}_i)\}_{i=1}^n$, $\mathbf{x}_i \in [0,1]^d$, $y(\mathbf{x}) \in \mathbb{R}$.
- Using data D we construct a surrogate model

$$\widetilde{f}(\mathbf{x}) \approx f(\mathbf{x}).$$



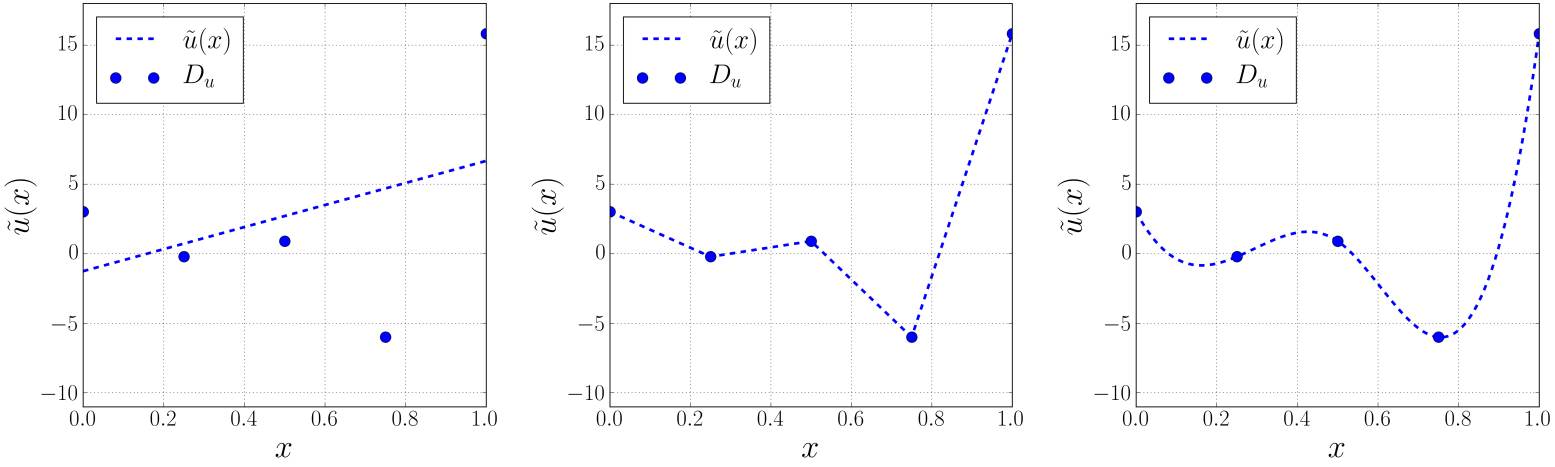


Figure: Exampples of $\widetilde{f}(\mathbf{x})$

Gaussian process regression is a good candidate

- No universal approach
- Kernel Gaussian process regression:
  - Uncertainty estimate $\sigma_{\widehat{f}}(\mathbf{x})$ and posterior distribution.
  - Interpolation or taking into account noise.
  - Smooth and nonlinear.
  - Computation complexity is $O(n^3)$:(

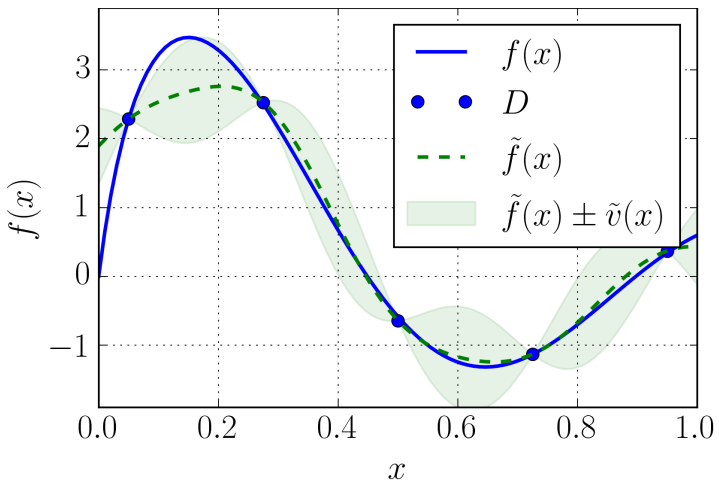
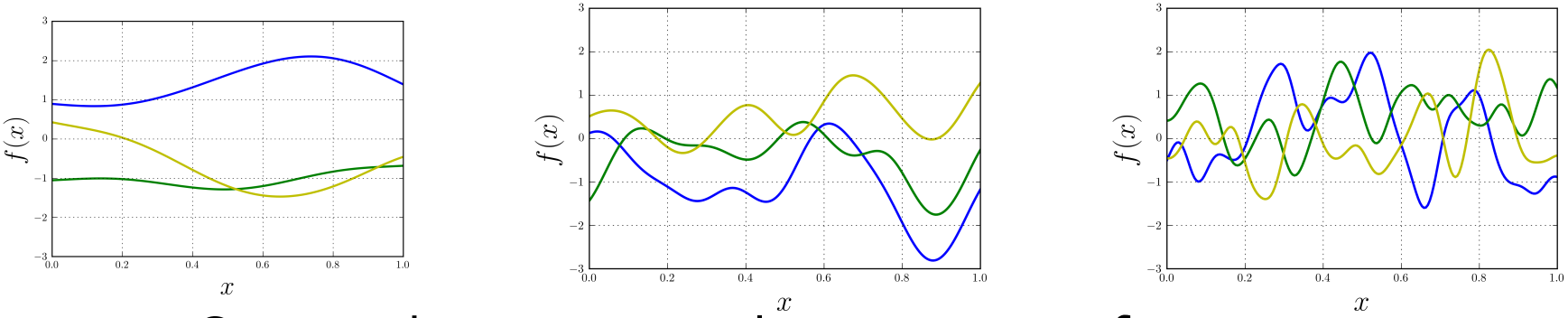
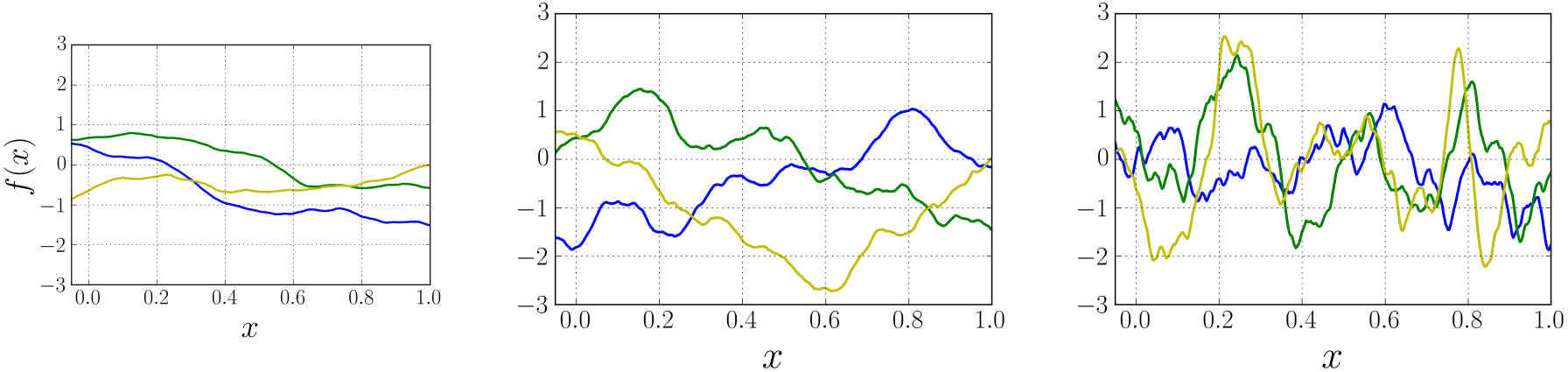


Figure: Gaussian process regression for $\mathbf{x} \in \mathbb{R}$

The prediction depends on the covariance function



Squared exponential covariance function



Matern $\frac{3}{2}$ covariance function

Bayesian optimization in practice

- Squared exponential covariance function
- Upper confidence bound or Expected improvement acquisition functions [see AlphaGo e.g. preprint]
- Batch execution
- Control that the model is adequate (e.g. regularization for parameters)

Chen, Yutian, et al. "Bayesian optimization in AlphaGo." arXiv preprint arXiv:1812.06855 (2018).
Shahriari, Bobak, et al. "Taking the human out of the loop: A review of Bayesian optimization." Proceedings of the IEEE 104.1 (2015): 148-175.
Snoek, Jasper, Hugo Larochelle, and Ryan P. Adams. "Practical Bayesian optimization of machine learning algorithms." Advances in neural information processing systems. 2012.

Bayesian optimization in practice: hyperparameters

- Optuna or Hyperopt python libraries
- For Optuna the model is Tree-structured Parzen Estimator Approach (TPE) [2]
- Can handle continuous and disrete function inputs
- Typically better than random search
- Can work in relatively high dimension

1. Akiba, Takuya, et al. "Optuna: A next-generation hyperparameter optimization framework." Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019.
2. Bergstra, James, et al. "Algorithms for hyper-parameter optimization." 25th annual conference on neural information processing systems (NIPS 2011). Vol. 24. Neural Information Processing Systems Foundation, 2011.

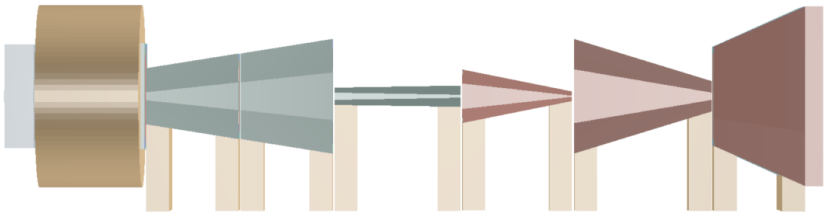
Bayesian optimization in practice: airfoils

- GP-based optimization
- Dimension reduction for inputs
- Multifidelity optimization: many precisions are available

1. Forrester, Alexander, Andras Sobester, and Andy Keane. Engineering design via surrogate modelling: a practical guide. John Wiley & Sons, 2008.
2. Koziel, Slawomir, and Leifur Leifsson. Surrogate-based modeling and optimization. New York: Springer, 2013.

Bayesian optimization in practice: particle detectors

- SHiP shield optimization: minimize passing of known particles
- 8 magnets, 7 parameters for each magnet: length, width, etc.
- Obtained configuration is 25% lighter with the same rejection capability



1. Baranov, A., et al. "Optimising the active muon shield for the SHiP experiment at CERN." Journal of Physics: Conference Series. Vol. 934. No. 1. IOP Publishing, 2017.

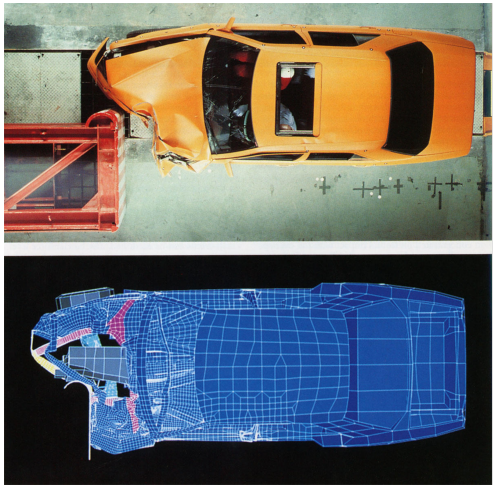
Multi fidelity Bayesian optimization

Multifidelity model

- A high fidelity sample
  $D_h = (X_h, \mathbf{y}_h) = \left\{\mathbf{x}_i^h, y_h(\mathbf{x}_i^u)\right\}_{i=1}^{n_h}$
- A low fidelity sample
  $D_l = (X_l, \mathbf{y}_l) = \left\{\mathbf{x}_i^l, y_l(\mathbf{x}_i^l)\right\}_{i=1}^{n_l}$,
  $\mathbf{x}_i^h, \mathbf{x}_i^l \in [0,1]^d$, $y_h(\mathbf{x}), y_l(\mathbf{x}) \in \mathbb{R}$.

Multifidelity model

- A high fidelity sample
  $D_h = (X_h, \mathbf{y}_h) = \left\{\mathbf{x}_i^h, y_h(\mathbf{x}_i^u)\right\}_{i=1}^{n_h}$
- A low fidelity sample
  $D_l = (X_l, \mathbf{y}_l) = \left\{\mathbf{x}_i^l, y_l(\mathbf{x}_i^l)\right\}_{i=1}^{n_l}$,
  $\mathbf{x}_i^h, \mathbf{x}_i^l \in [0,1]^d$, $y_h(\mathbf{x}), y_l(\mathbf{x}) \in \mathbb{R}$.
  - CFD code vs exact crash test
  - Smaller number of epochs for low fidelity
  - different datasets

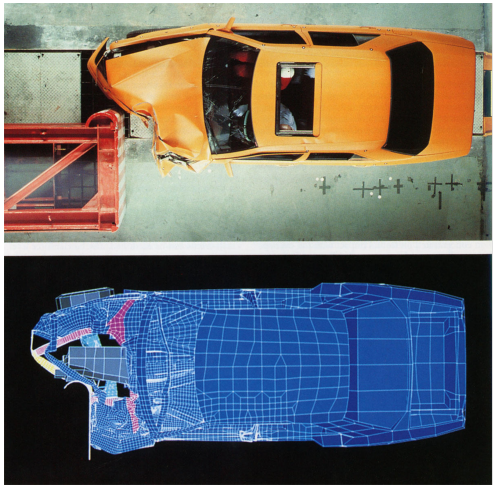


$u(\mathbf{x})$

$f(\mathbf{x})$

Multifidelity model

- A high fidelity sample
  $D_h = (X_h, \mathbf{y}_h) = \left\{\mathbf{x}_i^h, y_h(\mathbf{x}_i^u)\right\}_{i=1}^{n_h}$
- A low fidelity sample
  $D_l = (X_l, \mathbf{y}_l) = \left\{\mathbf{x}_i^l, y_l(\mathbf{x}_i^l)\right\}_{i=1}^{n_l}$,
  $\mathbf{x}_i^h, \mathbf{x}_i^l \in [0,1]^d$, $y_h(\mathbf{x}), y_l(\mathbf{x}) \in \mathbb{R}$.
  - CFD code vs exact crash test
  - Smaller number of epochs for low fidelity
  - different datasets
- Using data D_h and D_l, we construct model
  $$\widetilde{f}_h(\mathbf{x}) \approx f(\mathbf{x}).$$



$u(\mathbf{x})$

$f(\mathbf{x})$

Learning curve prediction

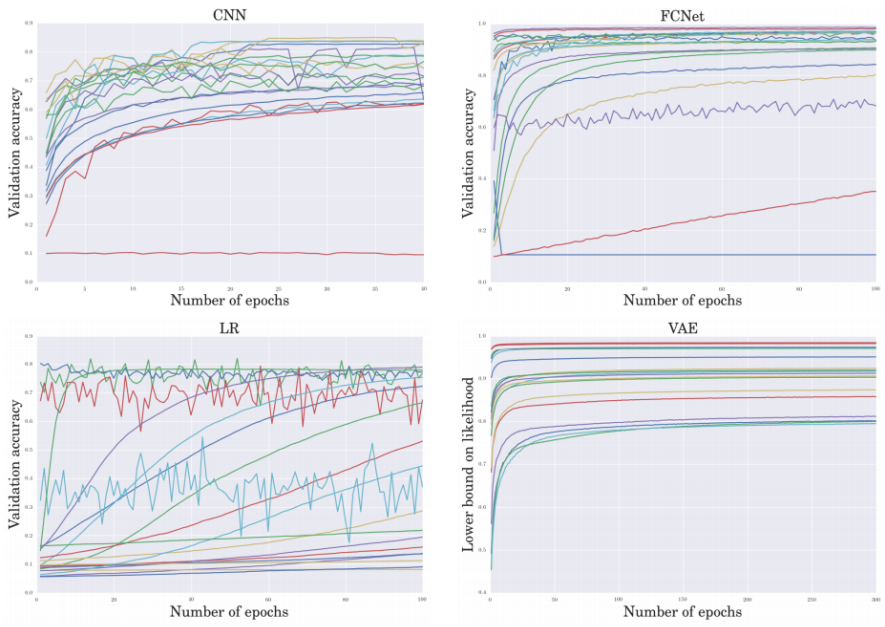


Figure 1: Example learning curves of random hyperparameter configurations of 4 different iterative machine learning methods: convolutional neural network (CNN), fully connected neural network (FCNet), logistic regression (LR), and variational auto-encoder (VAE). Although different configurations lead to different learning curves, they usually share some characteristics for a certain algorithm and dataset (but vary across these).

Klein, Aaron, et al. "Learning curve prediction with Bayesian neural networks." (2016).

Learning curve prediction quality

- If no uncertainties are required, LastSeenValue is hard to beat. This is because many configurations approach their final performance quite quickly.
- The GP mean predictions are very competitive, but the average log-likelihood indicates overconfidence, especially for short learning curves. We assume that the prior assumption of an exponential function is not flexible enough in practice and that after observing some data points the Gaussian process becomes too certain of its predictions.
- The random forest's mean predictions almost match the quality of the GP. The uncertainty estimates are better, but still too aggressive when only a small fraction of the learning curve was observed. Random forests do not extrapolate and similar to LastSeenValue also achieve a very good mean prediction if the learning curve has almost converged.
- Local models for single curves clearly outperform global models for almost complete learning curves. Global models, like our

Hyperband improvement

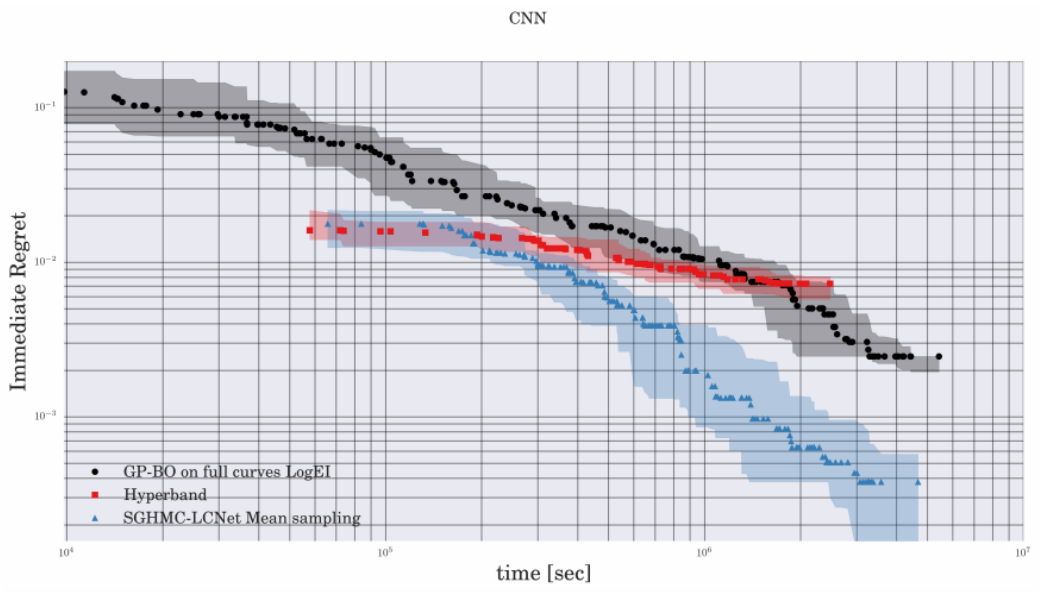


Figure 8: Comparison of Hyperband, Hyperband with our model, and standard Bayesian optimization on the CNN benchmark. Hyperband finds a good configuration faster than standard Bayesian optimization, but it only approaches the global optimum quickly when extended with our model.

Klein, Aaron, et al. "Learning curve prediction with Bayesian neural networks." (2016).

General approach for early stopping

We need the following models:

- A surrogate model $f(\mathbf{x}, \tau)$ that predicts the final performance on the base of hyperparameters $\mathbf{x}$ and time step τ.
- Our uncertainty estimate $\sigma(\mathbf{x}, \tau)$

Algorithm:

1. Select n configurations $\mathbf{x}_i$, $i = 1, \ldots, n$ and threshold Δ
2. Run full training for them to get initial sample
3. Construct models $f(\mathbf{x}, \tau)$, $\sigma(\mathbf{x}, \tau)$
4. Train new models with early stopping criterion: $P(f(\mathbf{x}, \tau) < y_{best}) < \Delta$.

Baker, Bowen, et al. "Accelerating neural architecture search using performance prediction." arXiv preprint arXiv:1705.10823 (2017).

General approach for early stopping

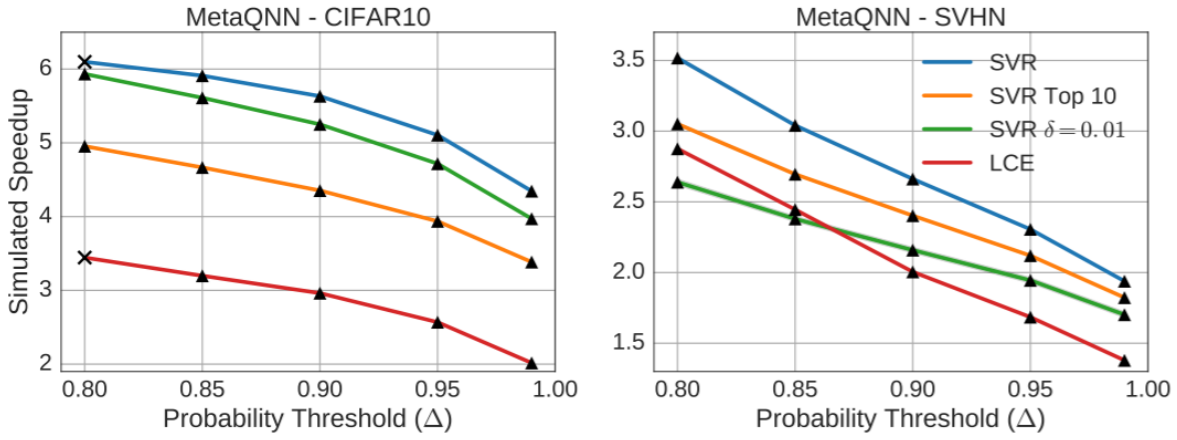


Figure 4: **Simulated Speedup in MetaQNN Search Space:** We compare the three variants of the early stopping algorithm presented in Section 4. Each ν-SVR SRM is trained using the first 100 learning curves, and each algorithm is tested on 10 independent orderings of the model configurations. Triangles indicate an algorithm that successfully recovered the optimal model for more than half of the 10 orderings, and X's indicate those that did not.

Baker, Bowen, et al. "Accelerating neural architecture search using performance prediction." arXiv preprint arXiv:1705.10823 (2017).

General approach for early stopping

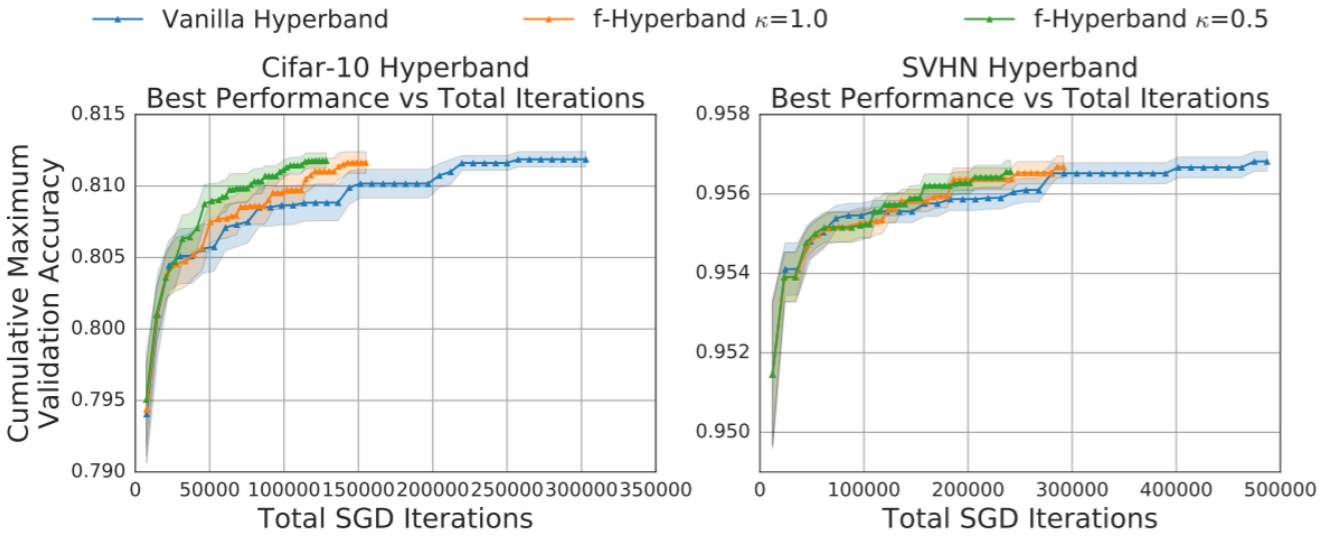


Figure 6: **Simulated Max Accuracy vs SGD Iterations for Hyperband:** We show the trajectories of the maximum performance so far versus total computational resources used for 40 consecutive Hyperband runs with $\eta = 3.0$ and $\Delta = 0.95$. f-Hyperband remains above the Hyperband curve at all iterations, and less aggressive settings for κ converge to the same or better final accuracy. Each triangle marks the completion of full Hyperband algorithm.

Baker, Bowen, et al. "Accelerating neural architecture search using performance prediction." arXiv preprint arXiv:1705.10823 (2017).

Upper bounds on quality of Bayesian optimization

Problem statement

- An unknown reward function

$$f : D \to \mathbb{R}.$$

- In each round t choose a point $\mathbf{x}_t \in D$
- Then we see the function value, perturbed by noise:

$$y_t = f(\mathbf{x}_t) + \varepsilon_t.$$

- Our goal is to maximize the sum of rewards $\sum_{t=1}^{T} f(\mathbf{x}_t)$,
- Or perform essentially as well as $\mathbf{x}^*$:

$$\mathbf{x}^* = \operatorname*{argmax}_{\mathbf{x} \in D} f(\mathbf{x}).$$

Srinivas, Niranjan, et al. "Gaussian process optimization in the bandit setting: No regret and experimental design." arXiv preprint arXiv:0912.3995 (2009)

The estimation of quality of optimization is a regret

Suppose

- The unknown function is f
- The maximum point $\mathbf{x}^*$:

$$\mathbf{x}^* = \operatorname*{argmax}_{\mathbf{x} \in D} f(\mathbf{x}).$$

- The *instantaneous regret* r_t at round t:

$$r_t = f(\mathbf{x}^*) - f(\mathbf{x}_t)$$

- The *cumulative regret* R_T after T rounds

$$R_T = \sum_{t=1}^{T} r_t.$$

- We want to work in a *no-regret* setting:

$$\lim_{T \to \infty} R_T / T = 0.$$

GP UCB algorithm

- **Input:** Input space D; GP Prior $\mu_0 = 0,\ \sigma_0,\ k$
- **For:** $t = 1, 2, \ldots$
- Choose $\mathbf{x}_t = \operatorname{argmax}_{\mathbf{x} \in D} \mu_{t-1}(\mathbf{x}) + \sqrt{\beta_t}\sigma_{t-1}(\mathbf{x})$
- Sample $y_t = f(\mathbf{x}_t) + \varepsilon_t$
- Perform Bayesian update to obtain μ_t and σ_t

(a) Example of temperature data collected by a network of 46 sensors at Intel Research Berkeley. (b,c) Two iterations of the GP UCB algorithm. It samples points that are either uncertain (b) or have high posterior mean (c).

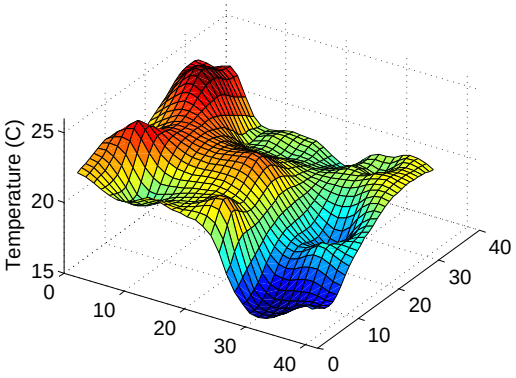


Figure: (a) Temperature data

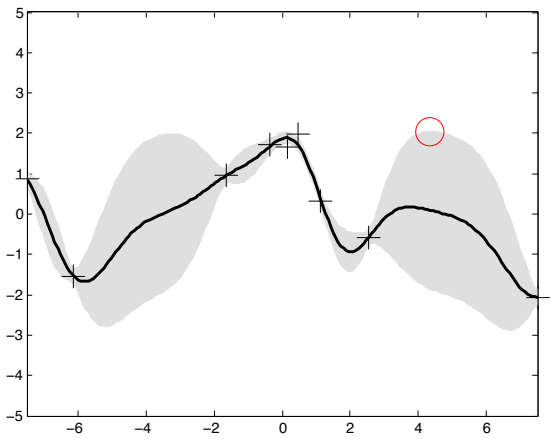


Figure: (b) Iteration t

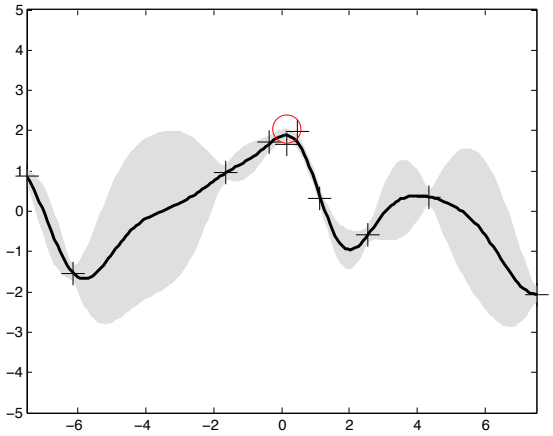


Figure: (c) Iteration $t+1$

Information Gain

1. The informativeness of a set of sampling points $A \subset D$ about f is measured by the *information gain*, which is the mutual information between f and observations $\mathbf{y}_A = \mathbf{f}_A + \epsilon_A$ at these points:

$$\mathrm{I}(\mathbf{y}_A; f) = \mathrm{H}(\mathbf{y}_A) - \mathrm{H}(\mathbf{y}_A | f), \tag{1}$$

2. For a Gaussian, $\mathrm{H}(N(\boldsymbol{\mu}, \boldsymbol{\Sigma})) = \frac{1}{2} \log |2\pi e \boldsymbol{\Sigma}|$, so that in our setting $\mathrm{I}(\mathbf{f}_A; f) = \mathrm{I}(\mathbf{y}_A; \mathbf{f}_A) = \frac{1}{2} \log |I + \sigma^{-2} \mathrm{K}A|$, where $\mathrm{K}A = [k(\mathbf{x}, \mathbf{x}')]_{\mathbf{x}, \mathbf{x}' \in A}$.
3. *The maximum information gain* γ_T after T rounds is:

$$\gamma_T := \max_{A \subset D : |A| = T} \mathrm{I}(\mathbf{y}_A; \mathbf{f}_A), \tag{2}$$

Upper bound for finite design space

For finite D, we obtain the following bound.

Theorem

Let $\delta \in (0,1)$ and $\beta_t = 2\log(|D|t^2\pi^2/6\delta)$.

Running GP-UCB with β_t for a sample f of a GP with mean function zero and covariance function $k(\mathbf{x},\mathbf{x}')$, we obtain a regret bound of $\mathcal{O}^*(\sqrt{T\gamma_T \log|D|})$ with high probability.

Precisely, we have

$$\Pr\left\{R_T \leq \sqrt{C_1 T \beta_T \gamma_T} \quad \forall T \geq 1\right\} \geq 1-\delta.$$

with $C_1 = 8/\log(1+\sigma^{-2})$.

Upper bound for continuous design space

Theorem

Let $D \subset \mathbb{R}^d$ be compact and convex, $d \in \mathbb{N}$. Suppose the kernel $k(\mathbf{x}, \mathbf{x}')$ satisfies the high probability bound on the derivatives of GP sample paths f: for some constants $a, b > 0$

$$\Pr\{\sup_{\mathbf{x} \in D} |\partial f / \partial x_j| > L\} \leq a e^{-(L/b)^2}, j = 1, \ldots, d$$

Pick $\delta \in (0, 1)$, and define

$$\beta_t = 2 \log(t^2 2\pi^2/(3\delta)) + 2d \log\left(t^2 dbr \sqrt{\log(4da/\delta)}\right).$$

Running the GP-UCB with β_t for a sample f of a GP with mean function zero and covariance function $k(\mathbf{x}, \mathbf{x}')$, we obtain a regret bound of $\mathcal{O}^*(\sqrt{dT\gamma_T})$ with high probability. More precisely,

$$\Pr\left\{R_T \leq \sqrt{C_1 T \beta_T \gamma_T} \quad \forall T \geq 1\right\} \geq 1 - \delta,$$

where $C_1 = 8/\log(1 + \sigma^{-2})$ is independent of T.

The assumption on sample paths often hold

1. The assumption on sample pathes holds for four-times differentiable kernels, such as Squared Exponential and Matern with $\nu > 2$.
2. The assumption doesn't hold for the Ornstein-Uhlenbeck kernel (Matern with $\nu = 1/2$; a stationary variant of the Wiener process).
3. For the latter covariance function, sample paths f are nondifferentiable almost everywhere with probability one and come with independent increments.

Particular covariance functions

- The *Finite Dimensional Linear* kernel:

$$k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^{\mathrm{T}}\mathbf{x}'$$

  GPs with this kernel correspond to random linear functions $f(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x}$, $\mathbf{w} \sim N(\mathbf{0}, \boldsymbol{I})$.

- The *Squared Exponential* kernel:

$$k(\mathbf{x}, \mathbf{x}') = \exp(-(2l^2)^{-1}\|\mathbf{x} - \mathbf{x}'\|^2),$$

  l is a lengthscale parameter.

- The *Matérn* kernel:

$$k(\mathbf{x}, \mathbf{x}') = (2^{1-\nu}/\Gamma(\nu))r^{\nu}B_{\nu}(r), r = (\sqrt{2\nu}/l)\|\mathbf{x} - \mathbf{x}'\|,$$

  where ν controls the smoothness of sample paths, B_{ν} is a modified Bessel function.

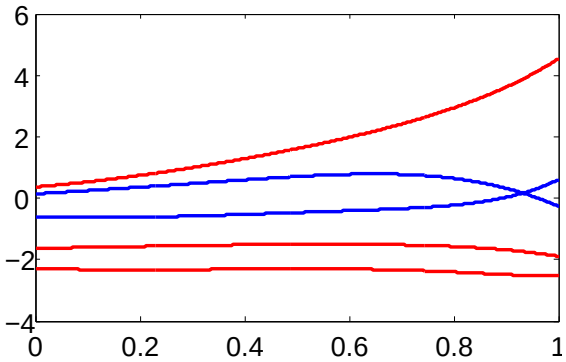


Figure: *Bayesian linear regression*

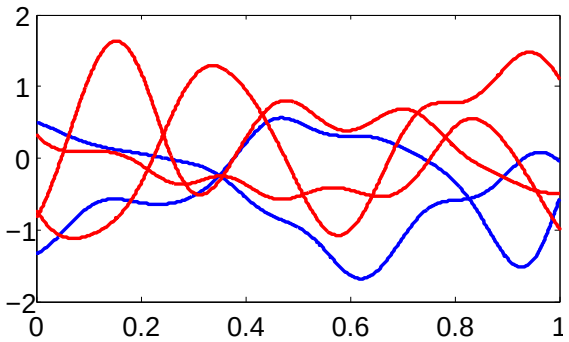


Figure: *Squared Exponential*

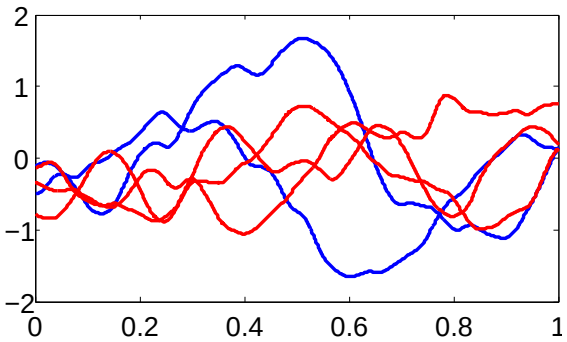


Figure: *Matérn*

Regret bounds for particular covariance functions

Kernel	Linear	RBF	Matérn
Regret R_T	$d\sqrt{T}$	$\sqrt{T(\log T)^{d+1}}$	$T^{\frac{\nu+d(d+1)}{2\nu+d(d+1)}}$

Figure: Upper regret bounds (up to polylog factors) for linear, radial basis, and Matérn kernels — d is the dimension, T is the time horizon, and ν is a Matérn parameter.

There exists some more recent lower bounds on the same regret

1. Bull, A. D. Convergence rates of efficient global optimization algorithms. JMLR, 2011.
2. Scarlett, J., Bogunovic, I., and Cevher, V. Lower bounds on regret for noisy Gaussian process bandit optimization. COLT, 2017.
3. Scarlett, J. Tight Regret Bounds for Bayesian Optimization in One Dimension. ICML, 2018.

To prove the results in [ICML, 2018] Scarlett uses Fano's inequality

Fano's inequality often used to prove various lower bounds

Theorem (Fano's inequality)

Let $z \to y \to \widehat{z}$ be a Markov chain, and let $p_e = \Pr\{\widehat{z} \neq z\}$ be an error for selection from one of U_z possible values of z. Let $H(p_e)$ denote the binary entropy function computed at p_e. Then,

$$H(p_e) + p_e \log(|U_z| - 1) \geq H(z|\widehat{z}) \geq H(z|y)$$

Sketch of the proof

Theorem: Upper bound for finite design space

For finite D, we obtain the following bound.

Theorem

Let $\delta \in (0,1)$ and $\beta_t = 2\log(|D|t^2\pi^2/6\delta)$.

Running GP-UCB with β_t for a sample f of a GP with mean function zero and covariance function $k(\mathbf{x},\mathbf{x}')$, we obtain a regret bound of $\mathcal{O}^*(\sqrt{T\gamma_T \log|D|})$ with high probability.

Precisely, we have

$$\Pr\left\{R_T \leq \sqrt{C_1 T \beta_T \gamma_T} \quad \forall T \geq 1\right\} \geq 1-\delta.$$

with $C_1 = 8/\log(1+\sigma^{-2})$.

The idea of the proof

The strategy is to show that $|f(\mathbf{x}) - \mu_{t-1}(\mathbf{x})| \leq \beta_t^{1/2} \sigma_{t-1}(\mathbf{x})$ for all $t \in \mathbb{N}$ and all $\mathbf{x} \in D$, or in the infinite case, all $\mathbf{x}$ in a discretization of D which becomes dense as t gets large.

Lemma

If $r \sim N(0,1)$ *then*

$$\Pr\{r > c\} \leq (1/2)e^{-c^2/2}.$$

Proof.

If $r \sim N(0,1)$, then

$$\Pr\{r > c\} = e^{-c^2/2}(2\pi)^{-1/2} \int e^{-(r-c)^2/2 - c(r-c)}\, dr$$
$$\leq e^{-c^2/2} \Pr\{r > 0\} = (1/2)e^{-c^2/2}$$

for $c > 0$, since $e^{-c(r-c)} \leq 1$ for $r \geq c$. □

Lemma

Pick $\delta \in (0,1)$ *and set* $\beta_t = 2\log(|D|\pi_t/\delta)$, *where* $\sum_{t\geq 1} \pi_t^{-1} = 1$, $\pi_t > 0$. *Then,*

$$|f(\mathbf{x}) - \mu_{t-1}(\mathbf{x})| \leq \beta_t^{1/2}\sigma_{t-1}(\mathbf{x}) \quad \forall \mathbf{x} \in D\, \forall t \geq 1$$

holds with probability $\geq 1 - \delta$.

Proof.

Fix $t \geq 1$ and $\mathbf{x} \in D$. Conditioned on $\mathbf{y}_{t-1} = (y_1, \ldots, y_{t-1})$, $\{\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}\}$ are deterministic, and $f(\mathbf{x}) \sim N(\mu_{t-1}(\mathbf{x}), \sigma_{t-1}^2(\mathbf{x}))$. Applying lemma 5 we get
$\Pr\{|f(\mathbf{x}) - \mu_{t-1}(\mathbf{x})| > \beta_t^{1/2}\sigma_{t-1}(\mathbf{x})\} \leq e^{-\beta_t/2}$, using
$r = (f(\mathbf{x}) - \mu_{t-1}(\mathbf{x}))/\sigma_{t-1}(\mathbf{x})$ and $c = \beta_t^{1/2}$. Applying the union bound,

$$|f(\mathbf{x}) - \mu_{t-1}(\mathbf{x})| \leq \beta_t^{1/2}\sigma_{t-1}(\mathbf{x}) \quad \forall \mathbf{x} \in D$$

holds with probability $\geq 1 - |D|e^{-\beta_t/2}$. Choosing $|D|e^{-\beta_t/2} = \delta/\pi_t$ and using the union bound for $t \in \mathbb{N}$, the statement holds. For

Lemma

Fix $t \geq 1$. If $|f(\mathbf{x}) - \mu_{t-1}(\mathbf{x})| \leq \beta_t^{1/2}\sigma_{t-1}(\mathbf{x})$ for all $\mathbf{x} \in D$, then the regret r_t is bounded by $2\beta_t^{1/2}\sigma_{t-1}(\mathbf{x}_t)$.

Proof.

By definition of $\mathbf{x}_t$:
$\mu_{t-1}(\mathbf{x}_t) + \beta_t^{1/2}\sigma_{t-1}(\mathbf{x}_t) \geq \mu_{t-1}(\mathbf{x}^*) + \beta_t^{1/2}\sigma_{t-1}(\mathbf{x}^*) \geq f(\mathbf{x}^*)$.
Therefore,

$$\begin{aligned} r_t = f(\mathbf{x}^*) - f(\mathbf{x}_t) &\leq \beta_t^{1/2}\sigma_{t-1}(\mathbf{x}_t) + \mu_{t-1}(\mathbf{x}_t) - f(\mathbf{x}_t) \\ &\leq 2\beta_t^{1/2}\sigma_{t-1}(\mathbf{x}t). \end{aligned}$$

□

Lemma

The information gain for the points selected can be expressed in terms of the predictive variances. If $\mathbf{f}_T = (f(\mathbf{x}_t)) \in I\!R^T$*:*

$$\mathrm{I}(\mathbf{y}_T; \mathbf{f}_T) = \frac{1}{2} \sum\nolimits_{t=1}^{T} \log\left(1 + \sigma^{-2}\sigma_{t-1}^2(\mathbf{x}_t)\right).$$

Proof.

Recall that $\mathrm{I}(\mathbf{y}_T; \mathbf{f}_T) = \mathrm{H}(\mathbf{y}_T) - (1/2)\log|2\pi e\sigma^2 I|$. Now, $\mathrm{H}(\mathbf{y}_T) = \mathrm{H}(\mathbf{y}_{T-1}) + \mathrm{H}(y_T|\mathbf{y}_{T-1}) = \mathrm{H}(\mathbf{y}_{T-1}) + \log(2\pi e(\sigma^2 + \sigma_{t-1}^2(\mathbf{x}_T)))/2$. Here, we use that $\mathbf{x}_1, \ldots, \mathbf{x}_T$ are deterministic conditioned on $\mathbf{y}_{T-1}$, and that the conditional variance $\sigma_{T-1}^2(\mathbf{x}_T)$ does not depend on $\mathbf{y}_{T-1}$. The result follows by induction. □

Lemma

Pick $\delta \in (0,1)$ *and let* $\beta_t = 2\log(|D|\pi_t/\delta)$. *Then, the following holds with probability* $\geq 1-\delta$*:*

$$\sum_{t=1}^{T} r_t^2 \leq \beta_T C_1 \mathrm{I}(\mathbf{y}_T; \mathbf{f}_T) \leq C_1 \beta_T \gamma_T \quad \forall T \geq 1,$$

where $C_1 := 8/\log(1+\sigma^{-2}) \geq 8\sigma^2$.

Proof.

By Lemma 2 and 3, we have that $\{r_t^2 \leq 4\beta_t \sigma_{t-1}^2(\mathbf{x}_t)\ \forall t \geq 1\}$ with probability $\geq 1-\delta$. Now, β_t is nondecreasing, so that

$$\begin{aligned} 4\beta_t \sigma_{t-1}^2(\mathbf{x}_t) &\leq 4\beta_T \sigma^2(\sigma^{-2}\sigma_{t-1}^2(\mathbf{x}_t)) \\ &\leq 4\beta_T \sigma^2 C_2 \log(1+\sigma^{-2}\sigma_{t-1}^2(\mathbf{x}_t)) \end{aligned}$$

with $C_2 = \sigma^{-2}/\log(1+\sigma^{-2}) \geq 1$, since $s^2 \leq C_2 \log(1+s^2)$ for $s \in [0, \sigma^{-2}]$, and $\sigma^{-2}\sigma_{t-1}^2(\mathbf{x}_t) \leq \sigma^{-2} k(\mathbf{x}_t, \mathbf{x}_t) \leq \sigma^{-2}$. Noting that $C_1 = 8\sigma^2 C_2$, the result follows by plugging in the representation of

Proofs of Theorem 1 and Theorem 2

- Theorem 1 is a simple consequence of Lemma 5, since $R_T^2 \leq T \sum_{t=1}^{T} r_t^2$ by the Cauchy-Schwarz inequality.
- For continuous case we follow the same procedure, but use the assumptions to be sure that everything is all right if we have a dense enough grid of points.

Concluding remarks

Advantages and disadvantages of Bayesian optimization

+ Natural approach
+ Empirical evidence suggests that this is faster, than global optimization
– But the benefits are often not that big
+ Flexible approach, we can include many optimization scenarios and assumptions about the target function
– There are some hyperparameters to select
+ Theoretical guarantees
– Assumptions are quite restrictive (and often violated in practice)

Research directions, 2020

- Hyperparameters optimization and model selection
- Variable fidelity & early stopping
- Combine with Neural Architecture search
- Use neural networks as a surrogate model (and develop good uncertainty estimates)