



UNIVERSITY OF
CAMBRIDGE

Machine Learning and the Physical World

Lecture 6 : Sequential Decision Making - Bayesian
Optimisation

Carl Henrik Ek - che29@cam.ac.uk

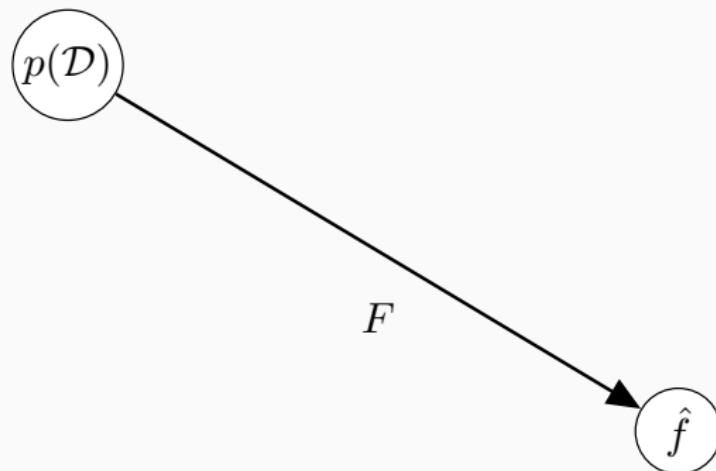
23rd of October, 2020

<http://carlhenrik.com>

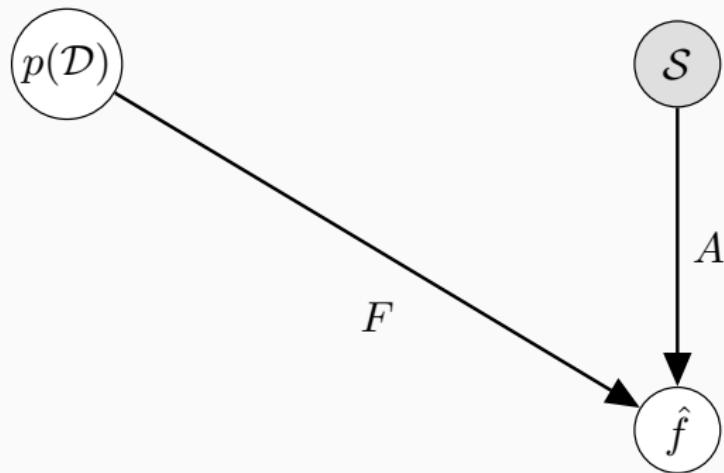
Learning Theory

$$\hat{f} = \operatorname{argmin}_{f \in \mathcal{F}} \int \ell(\mathcal{A}_{\mathcal{F}}(\mathcal{S}), x, y) p(x, y) dx dy$$

- \mathcal{F} space of functions
- \mathcal{A} learning algorithm
- $\mathcal{S} \sim P(\mathcal{X} \times \mathcal{Y})$
- $\ell(\mathcal{A}_{\mathcal{F}}(\mathcal{S}), x, y)$ loss function

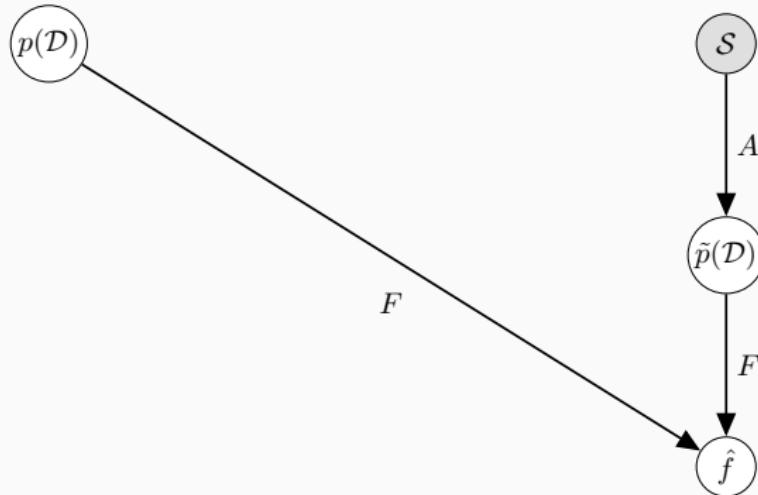


$$F : p(\mathcal{D}) \rightarrow p(\mathcal{Y}|\mathcal{X})$$



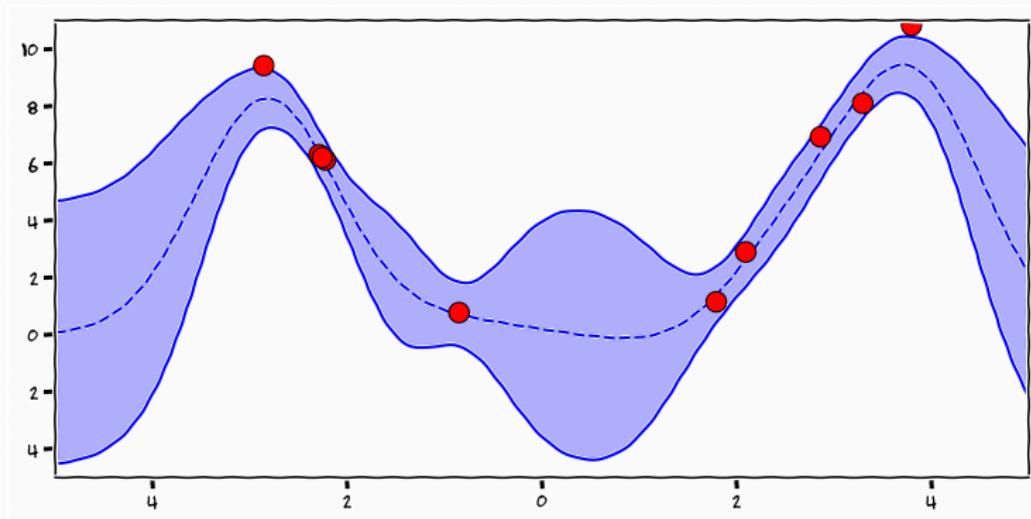
$$A \circ S \approx F \circ p(\mathcal{D})$$

Formalisation

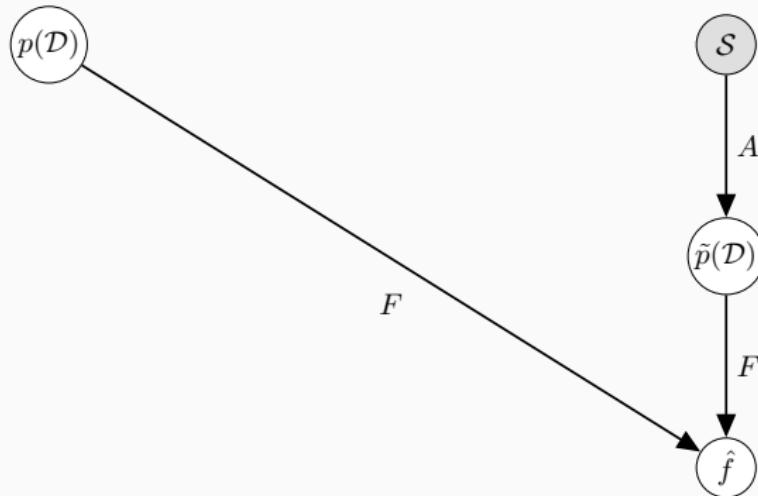


$$A \circ \mathcal{S} = \tilde{p}(\mathcal{D}) \approx p(\mathcal{D})$$

Regression

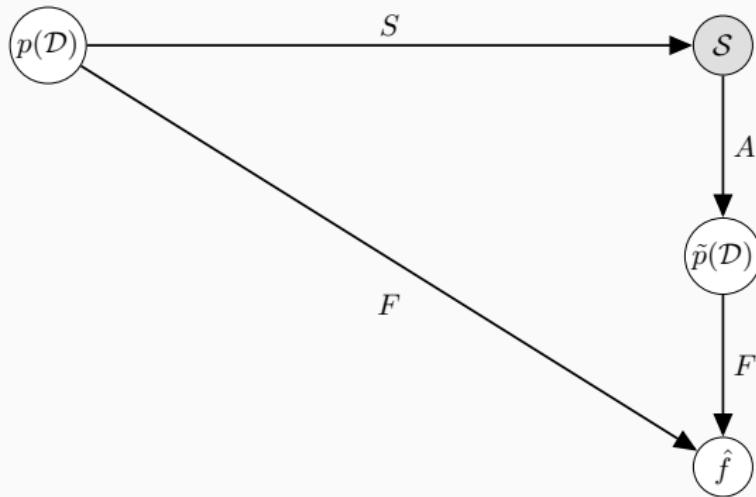


Formalisation



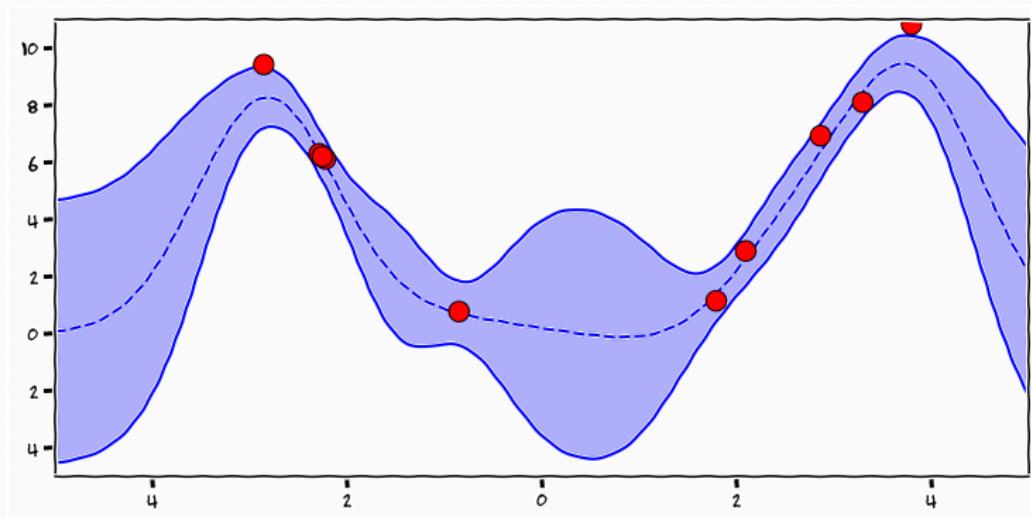
$$A \circ \mathcal{S} = \tilde{p}(\mathcal{D}) \approx p(\mathcal{D})$$

Formalisation

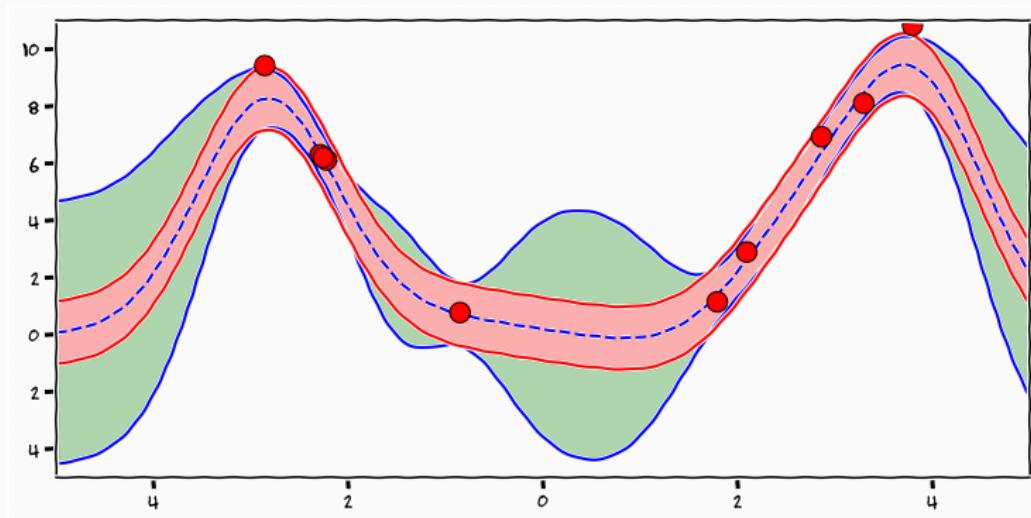


$$A \circ S \circ p(\mathcal{D}) \approx p(\mathcal{D})$$

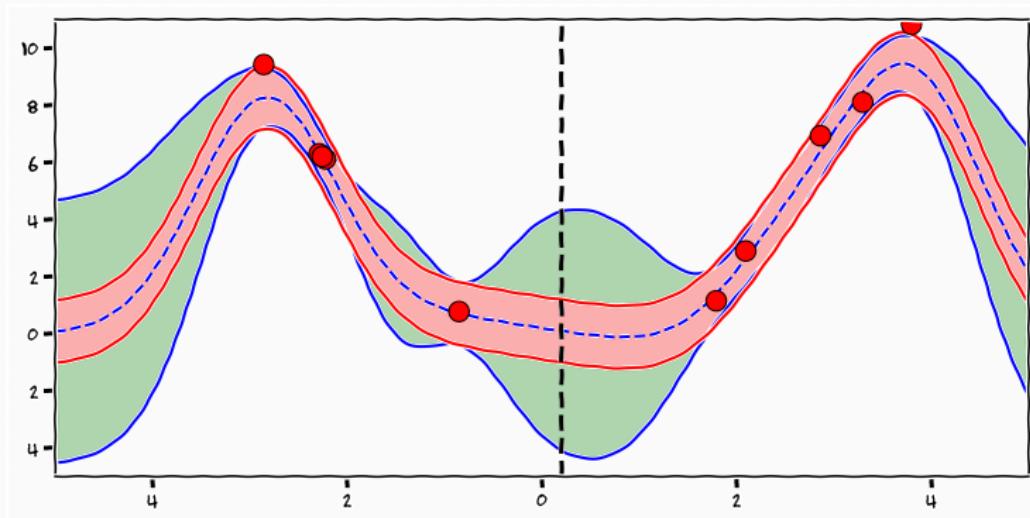
Regression



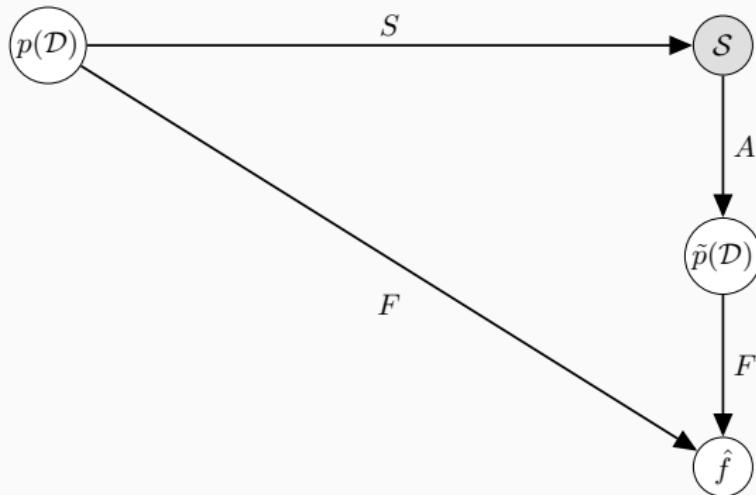
Uncertainty Quantification/Factorisation



Not all uncertainty is equal



Formalisation



$$A \circ S \circ p(\mathcal{D}) \approx p(\mathcal{D})$$

Examples

Quadrature Rules where should you query the integrand?

Examples

Quadrature Rules where should you query the integrand?

Differential Equations which step-size should you do?

Examples

Quadrature Rules where should you query the integrand?

Differential Equations which step-size should you do?

Reinforcement Learning which roll-outs should you do?

Black-Box Optimisation how can we find the minima of an explicitly unknown function?

Black-Box Optimisation how can we find the minima of an explicitly unknown function?

Surrogate Models how can we build a model as a surrogate for the unknown function?

Black-Box Optimisation how can we find the minima of an explicitly unknown function?

Surrogate Models how can we build a model as a surrogate for the unknown function?

Sequential decision making how can we come up with a strategy for sequentially exploring the function?

Bayesian Optimisation

Black-box Optimisation

$$x^{(*)} = \operatorname{argmin}_{x \in \mathcal{X}} f(x)$$

- \mathcal{X} is a bounded domain

Black-box Optimisation

$$x^{(*)} = \operatorname{argmin}_{x \in \mathcal{X}} f(x)$$

- \mathcal{X} is a bounded domain
- f is explicitly unknown

Black-box Optimisation

$$x^{(*)} = \operatorname{argmin}_{x \in \mathcal{X}} f(x)$$

- \mathcal{X} is a bounded domain
- f is explicitly unknown
- Evaluations of f may be noisy

Black-box Optimisation

$$x^{(*)} = \operatorname{argmin}_{x \in \mathcal{X}} f(x)$$

- \mathcal{X} is a bounded domain
- f is explicitly unknown
- Evaluations of f may be noisy
- Evaluations of f is expensive

- Lipschitz Continuity

$$\|f(x_1) - f(x_2)\| \leq C\|x_1 - x_2\|$$

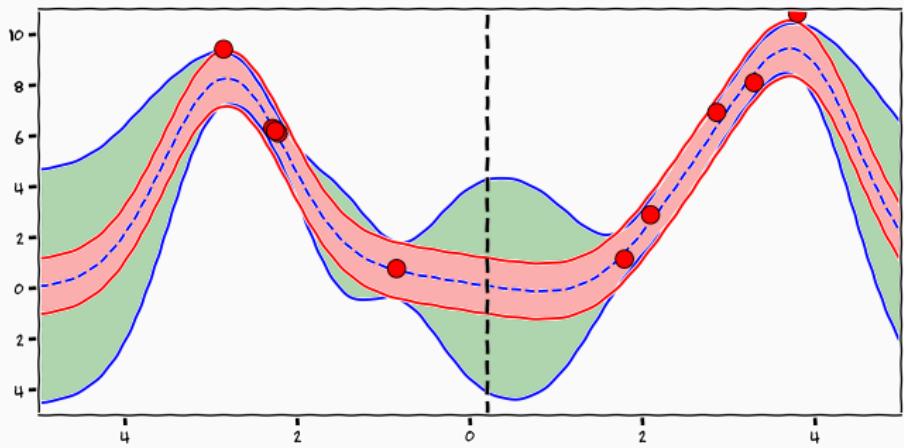
- Random Search

$$f(x^{(-)}) \leq f(x^{(*)}) - \epsilon$$

- Requires $\left(\frac{C}{2\epsilon}\right)^d$ evaluations on a d -dimensional hypercube
 - Surrogate model $p(f)$
 - *one-step*¹ or *average-case* optimisation

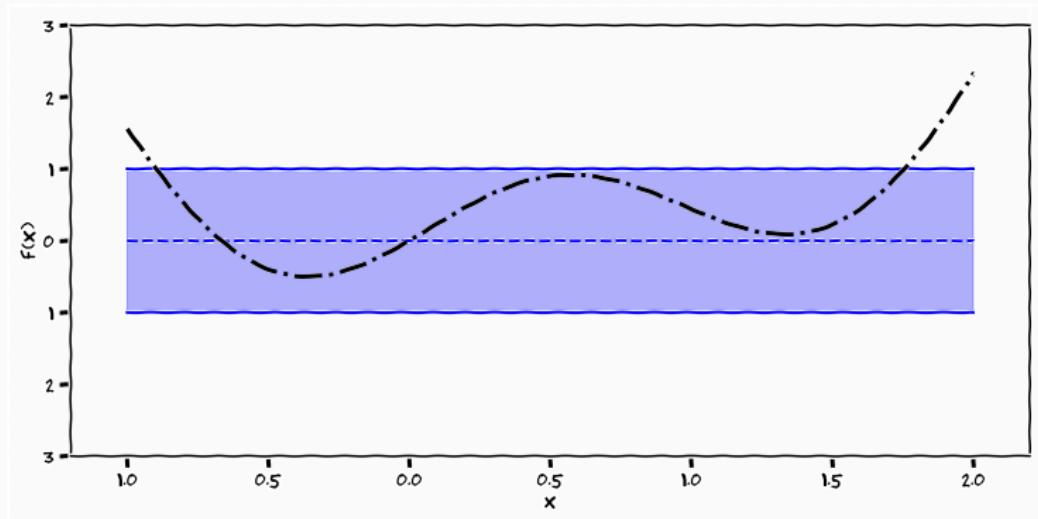
¹Močkus, 1975

Gaussian Process Surrogate

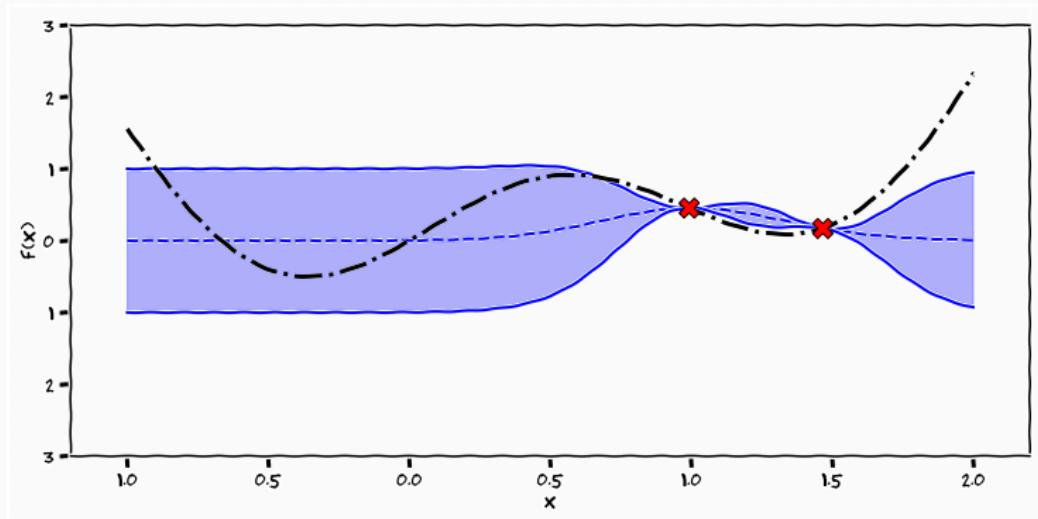


- allows for principled priors and **narrow** priors
- provides belief over the whole domain

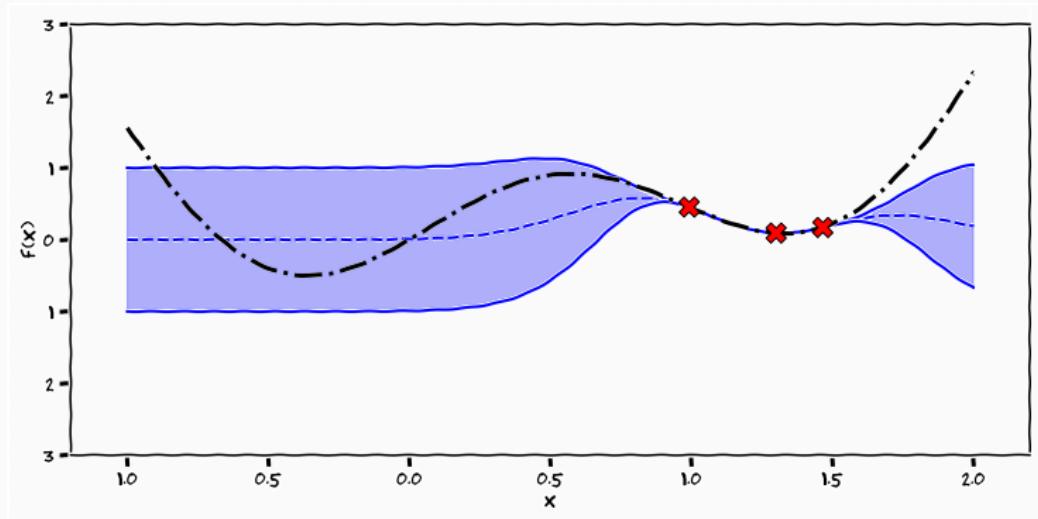
Posterior Search



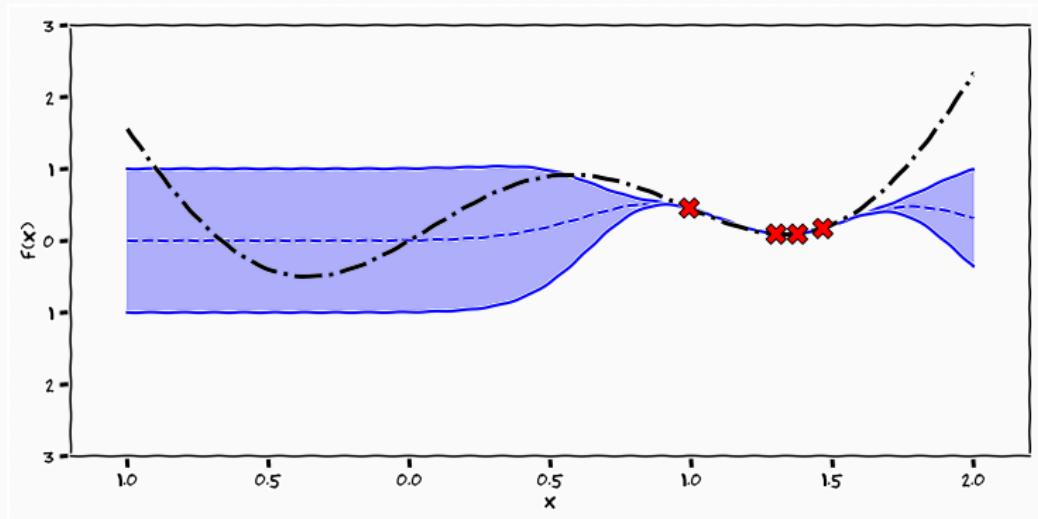
Posterior Search



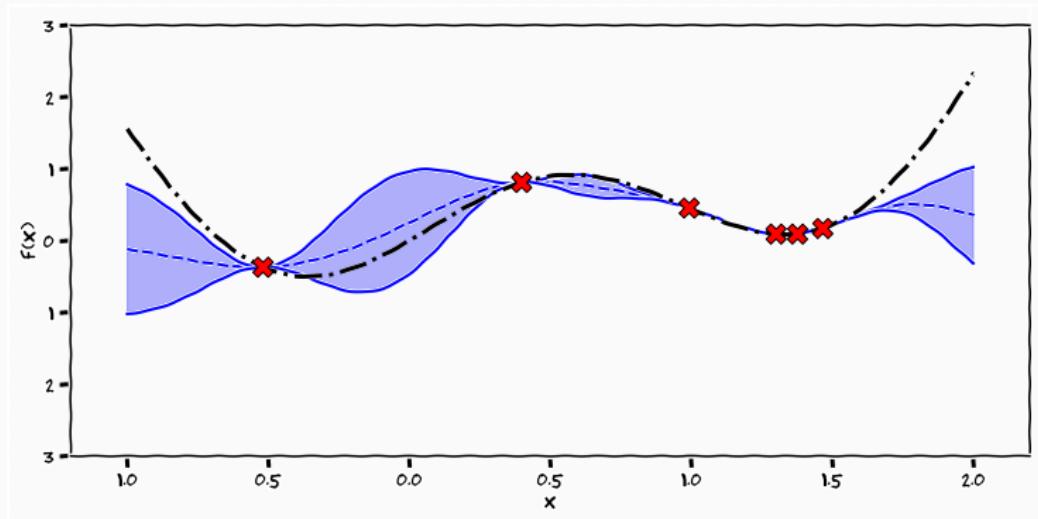
Posterior Search



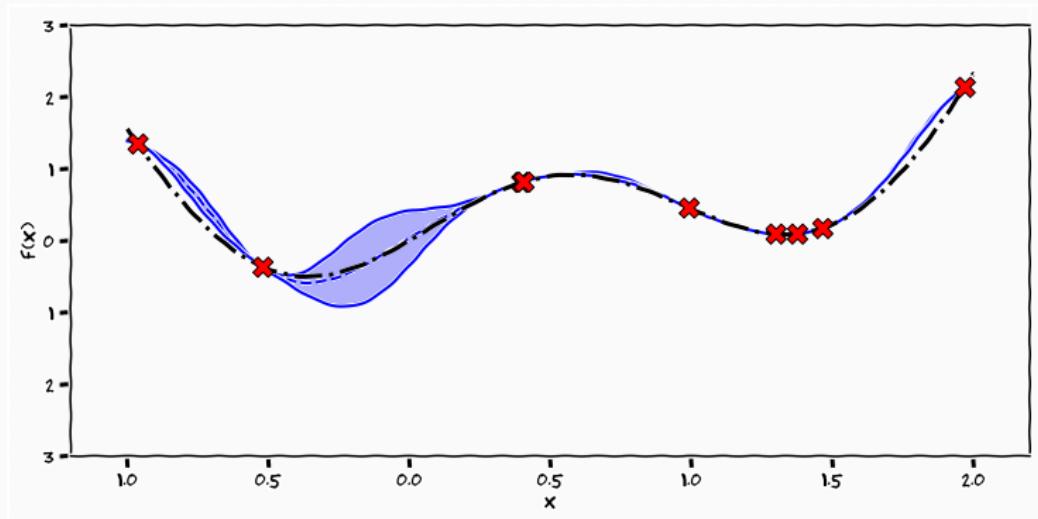
Posterior Search



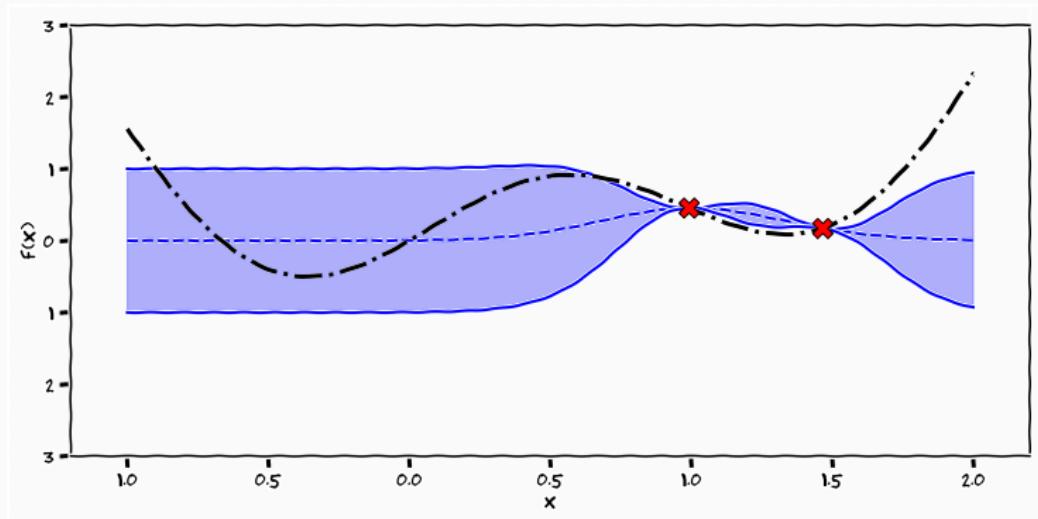
Posterior Search



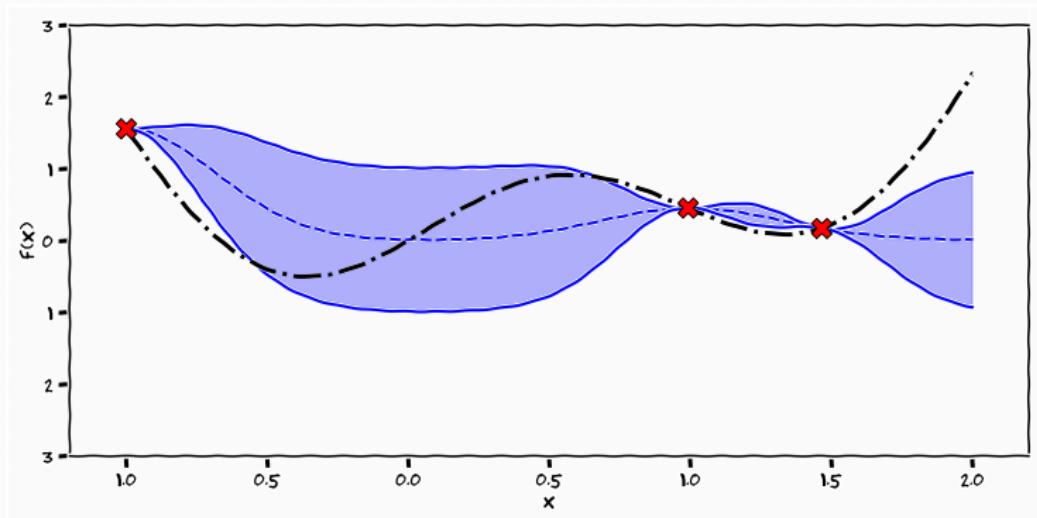
Posterior Search



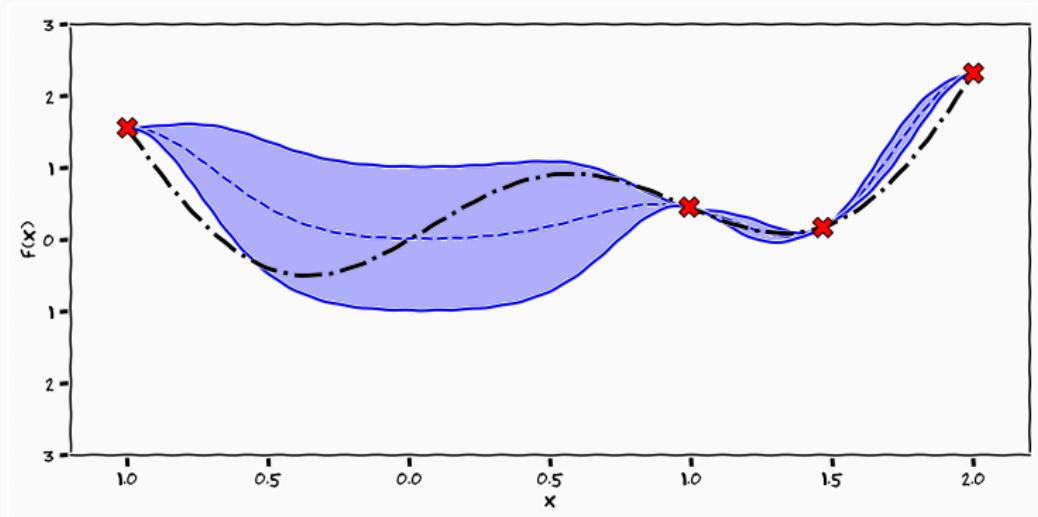
Posterior Search



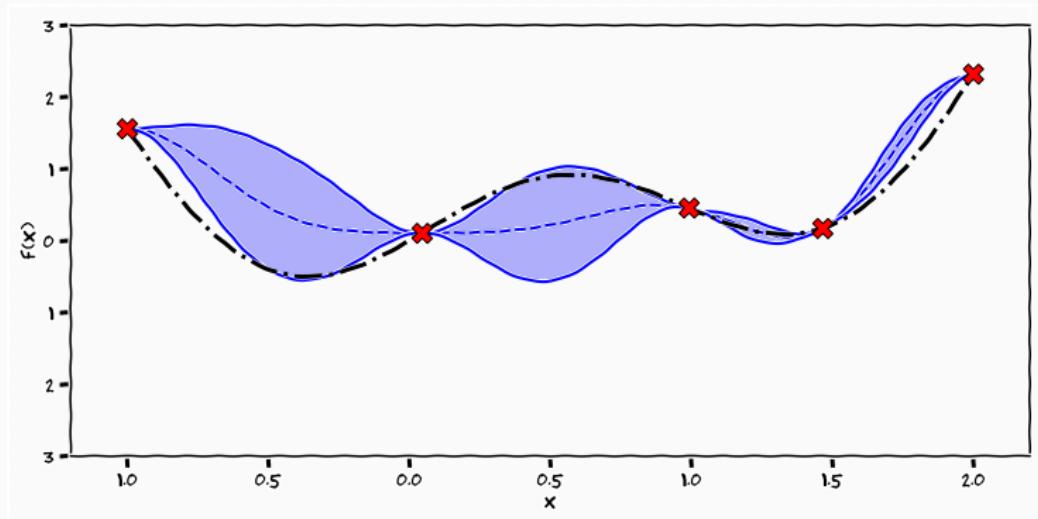
Posterior Search



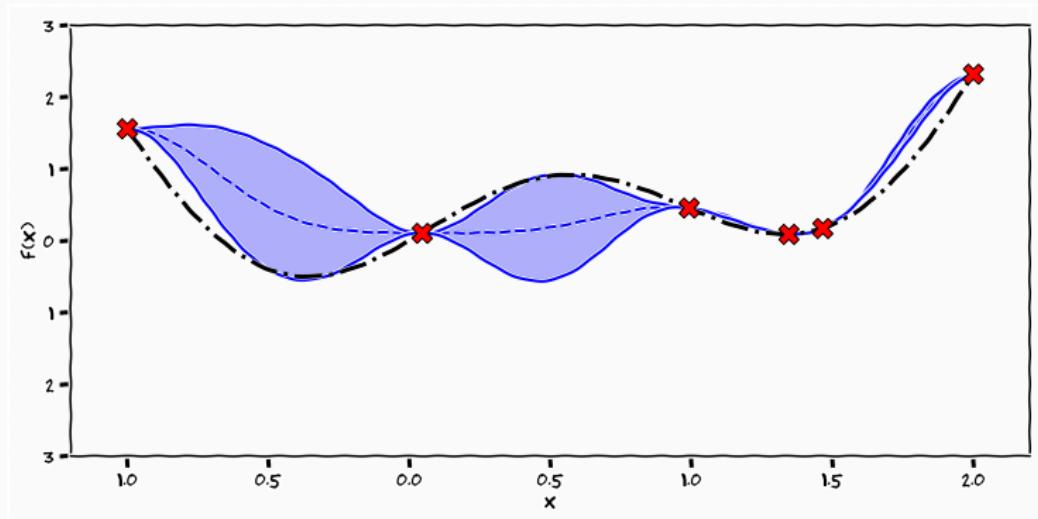
Posterior Search



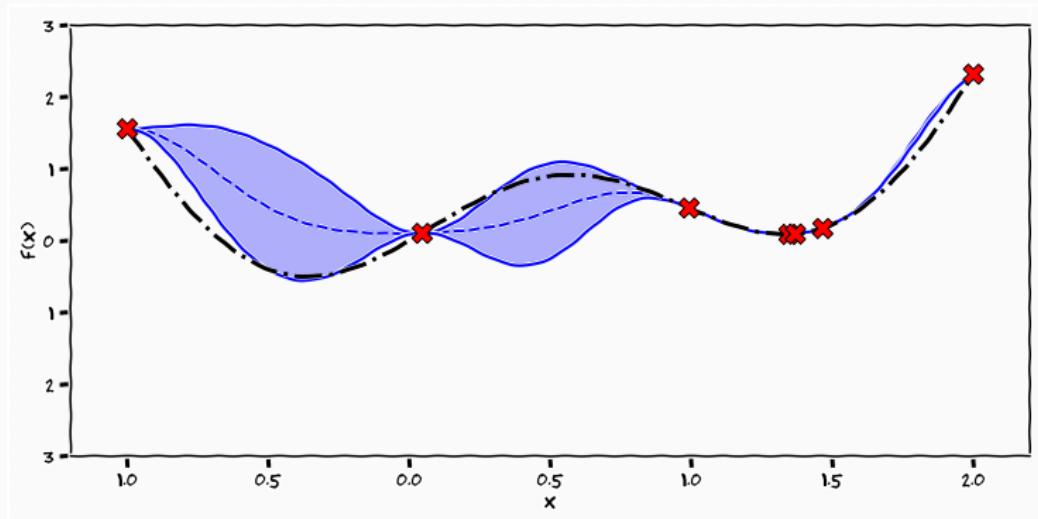
Posterior Search



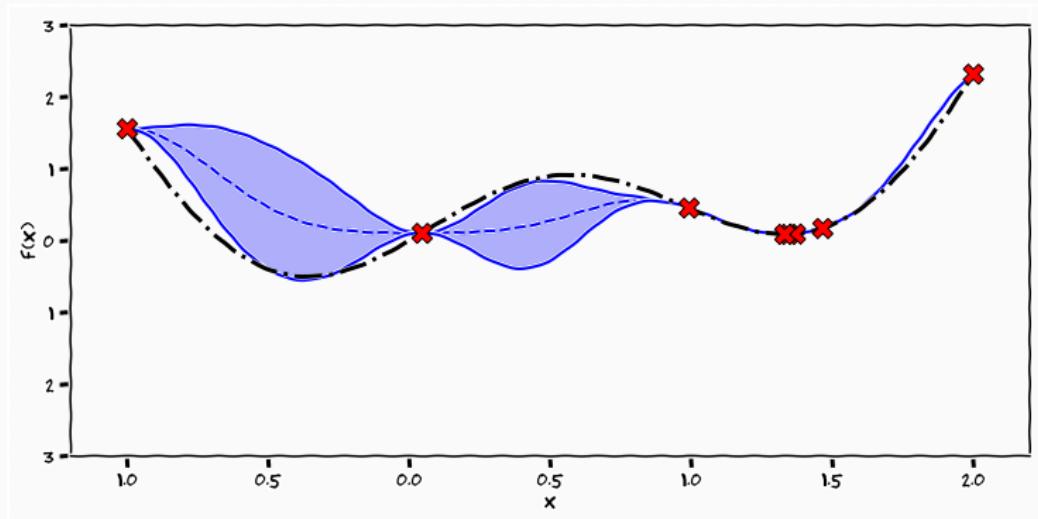
Posterior Search



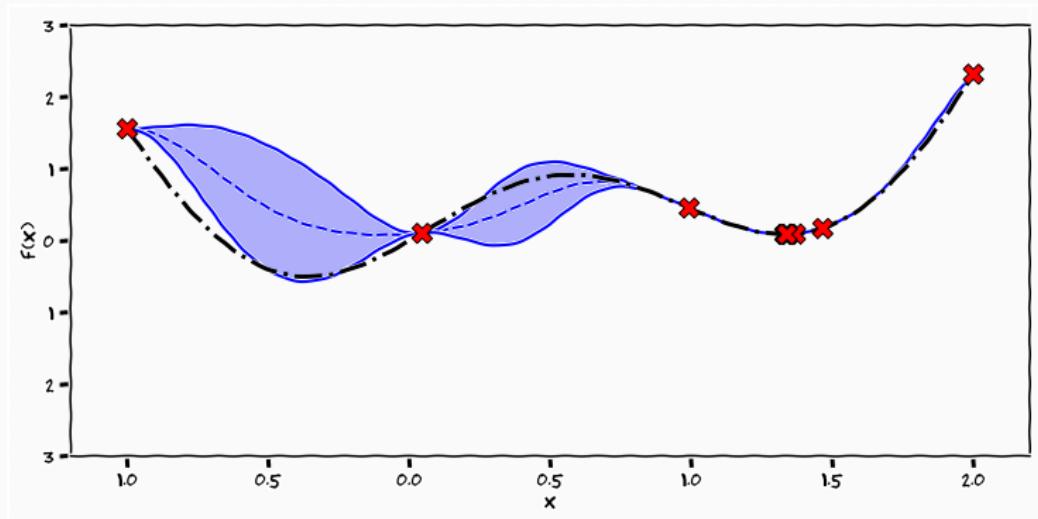
Posterior Search



Posterior Search



Posterior Search



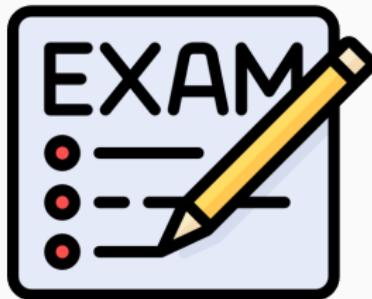








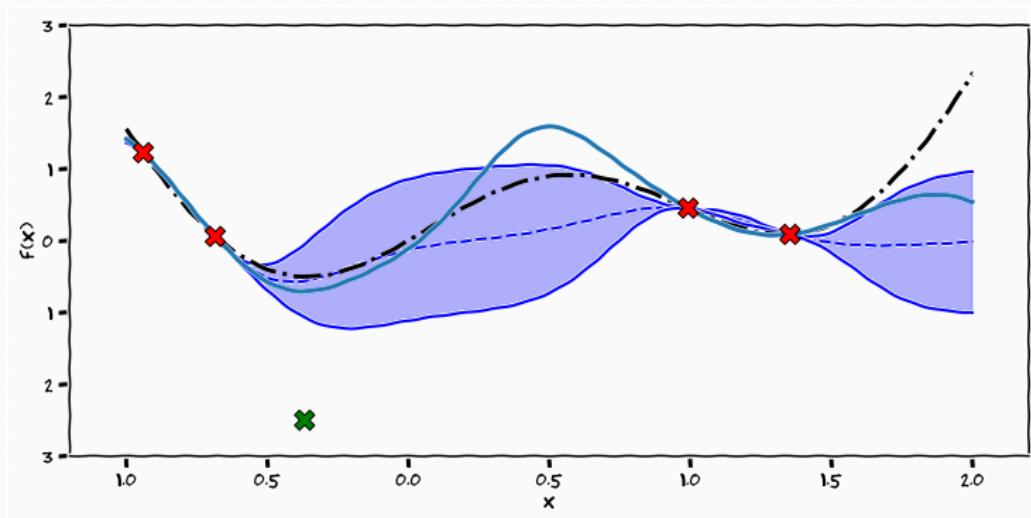
Exploration and Exploitation



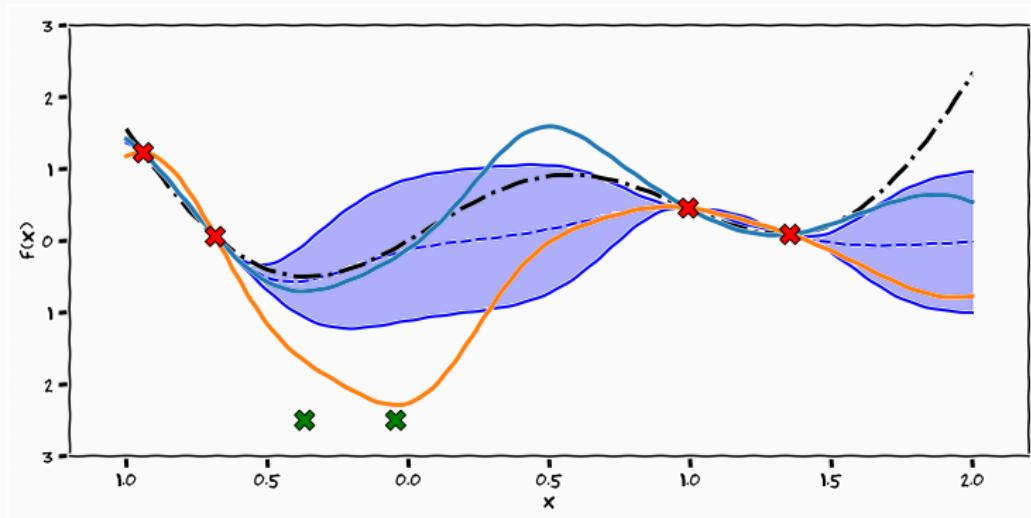
Exploitation use the knowledge that we currently have

Exploration try to gain new knowledge by trying new things

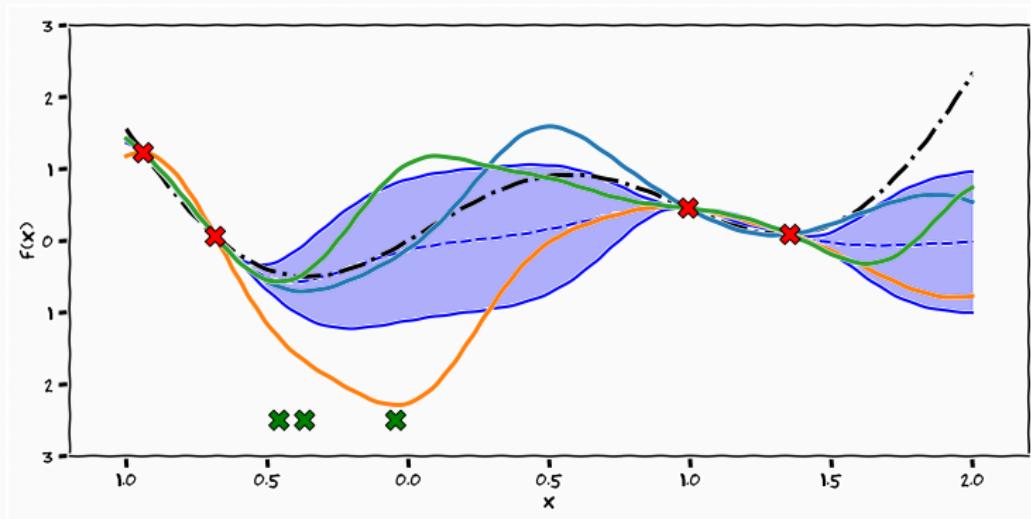
Surrogate Uncertainty



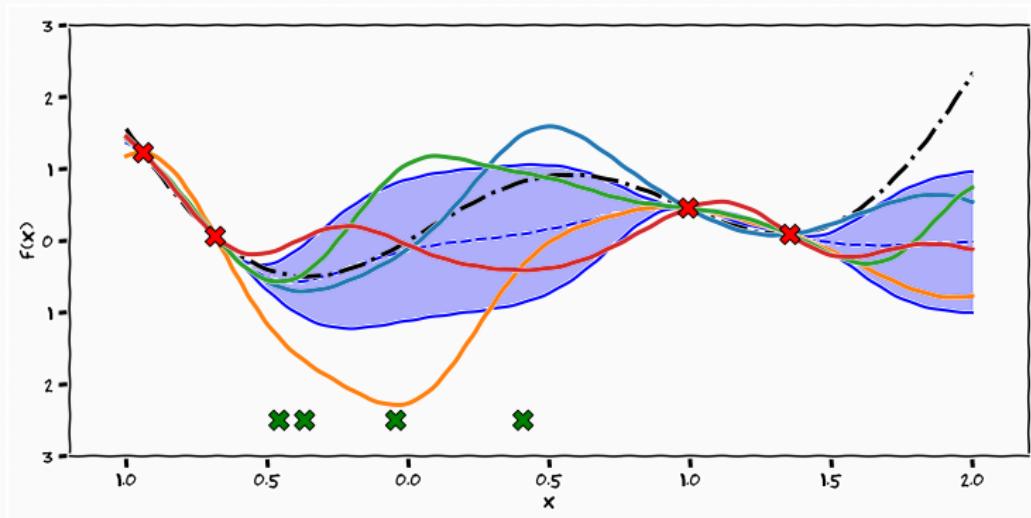
Surrogate Uncertainty



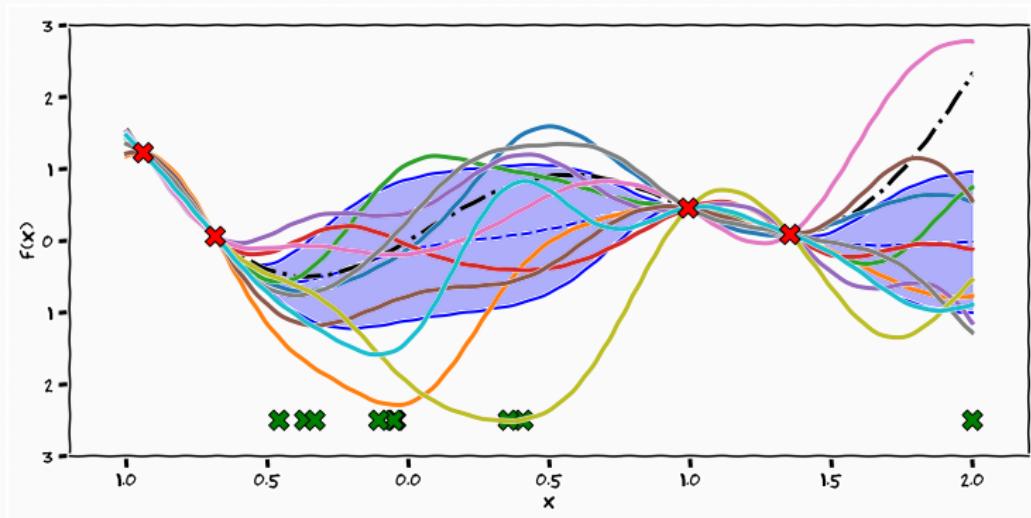
Surrogate Uncertainty



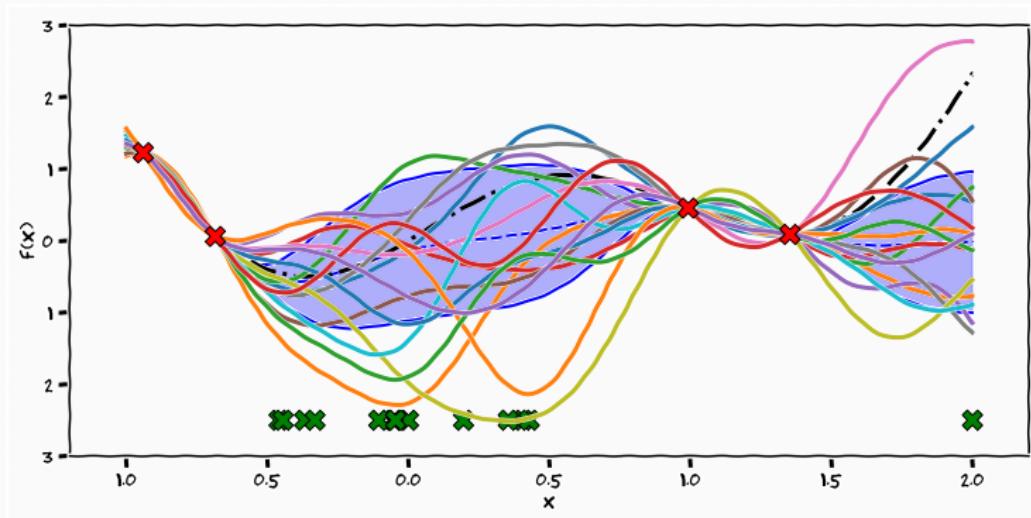
Surrogate Uncertainty



Surrogate Uncertainty



Surrogate Uncertainty

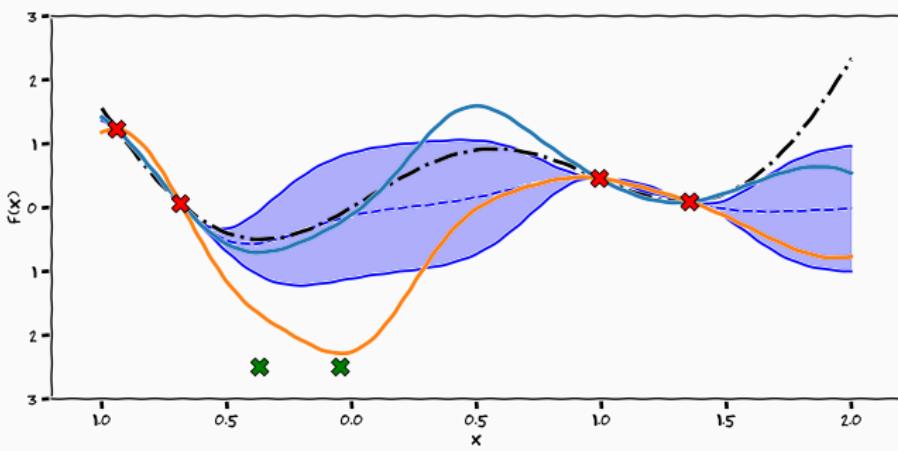


Acquisition Function

$$x_{n+1} = \operatorname{argmax}_{x \in \mathcal{X}} \alpha(x; \{x_i, y_i\}_{i=1}^n, \mathcal{M}_n)$$

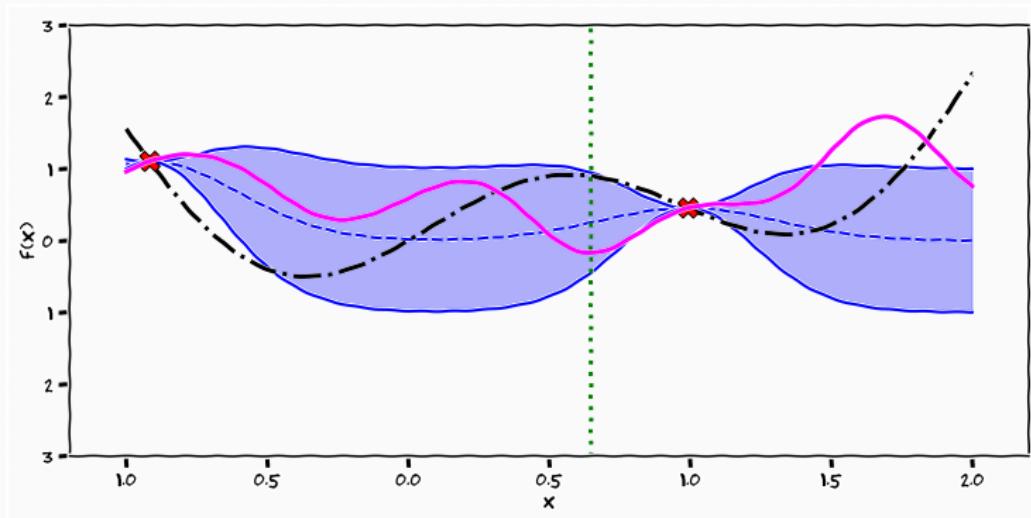
- Formulate a sequential decision problem
- This will work well if $\alpha(x)$
 - is cheap to compute
 - balances *exploration* and *exploitation*

Thompson Sampling Thompson, 1933

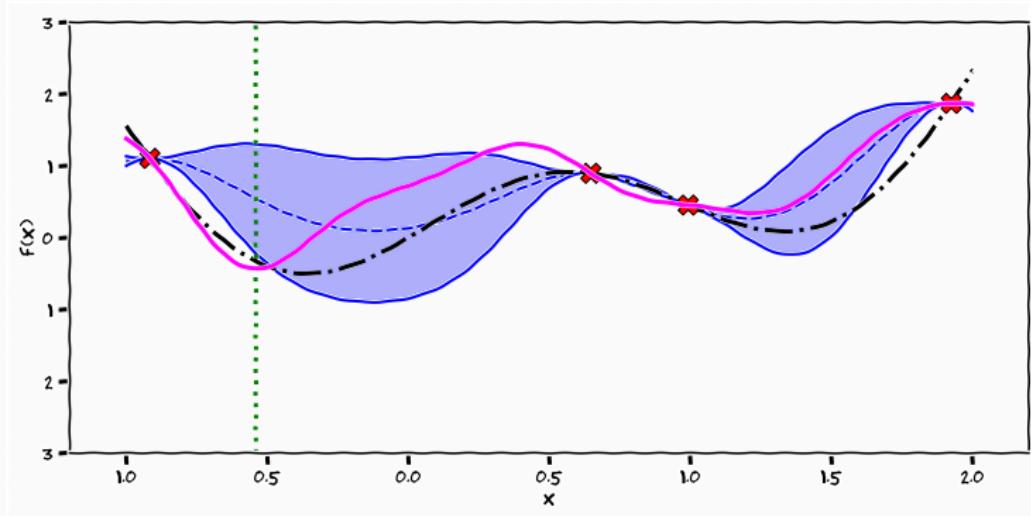


$$-\alpha(x; \{x_i, y_i\}_{i=1}^n, \mathcal{M}_n) \sim p(f \mid \{x_i, y_i\}_{i=1}^n)$$

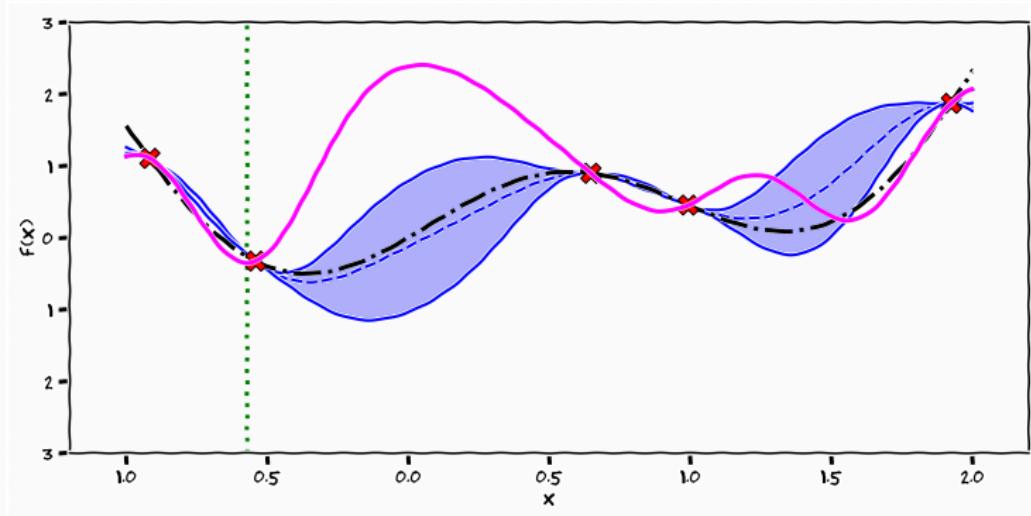
Thompson Sampling



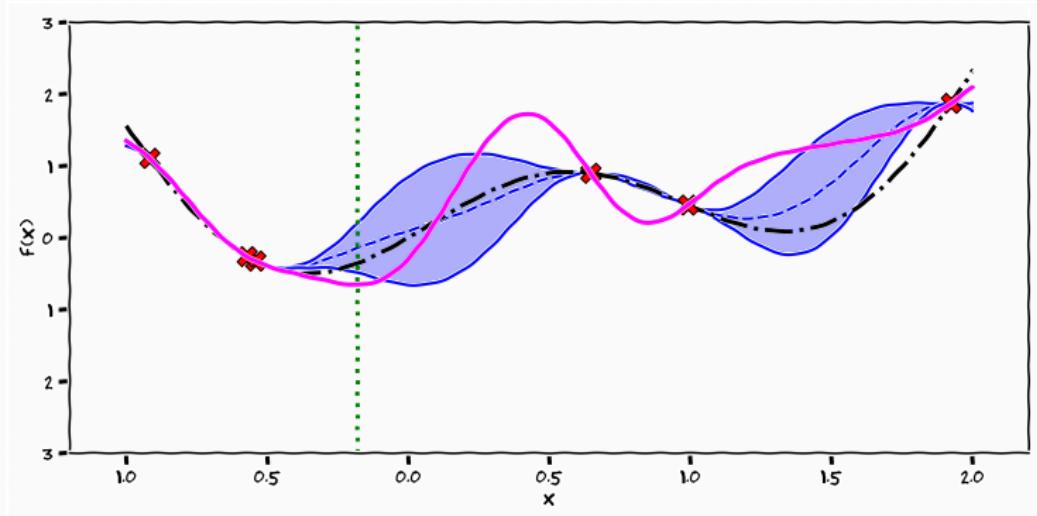
Thompson Sampling



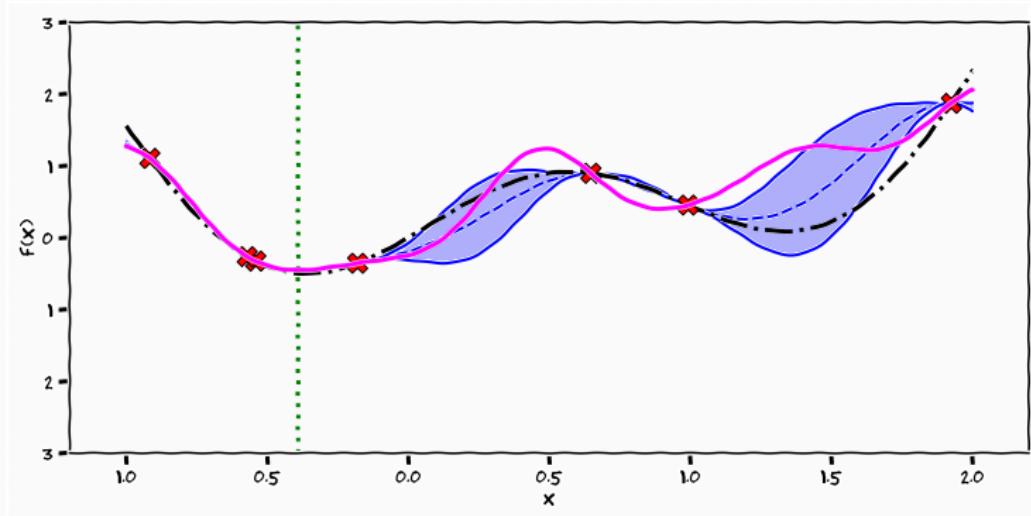
Thompson Sampling



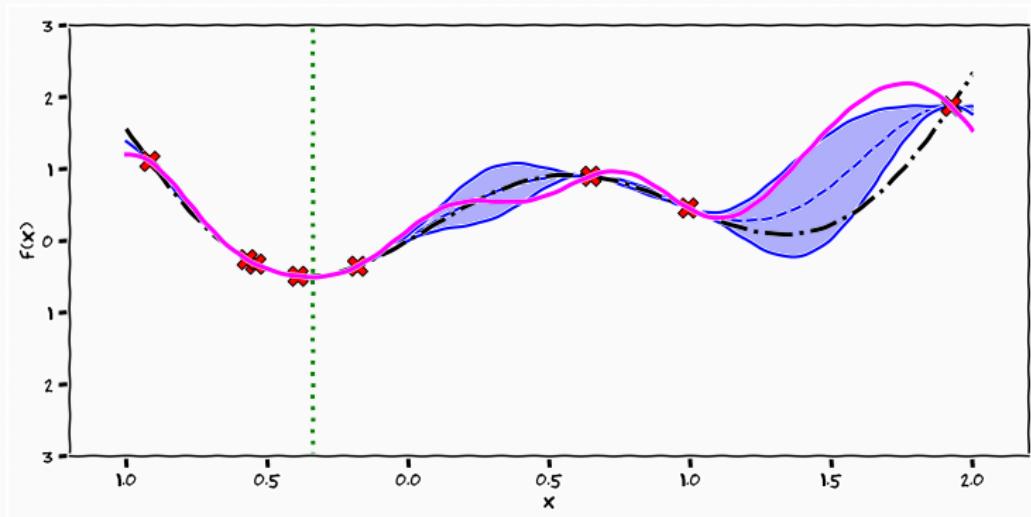
Thompson Sampling



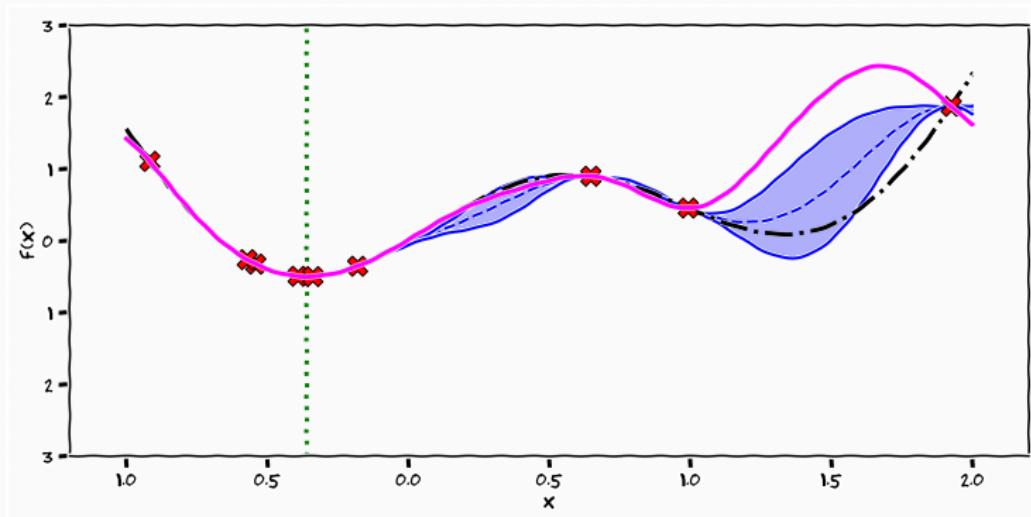
Thompson Sampling

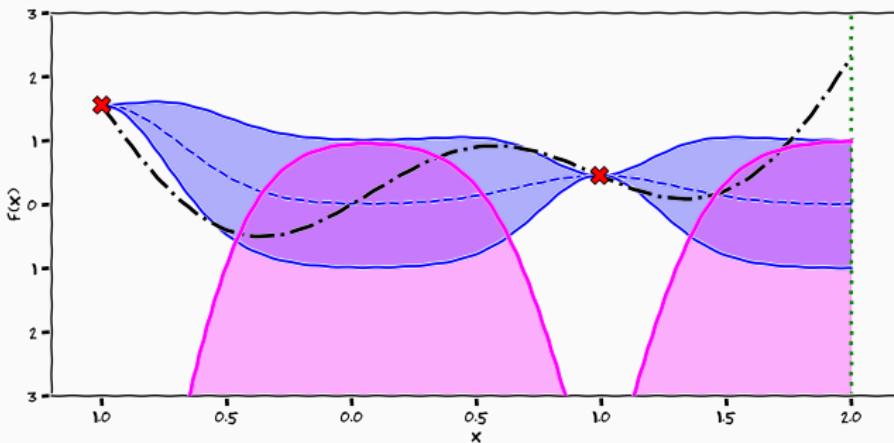


Thompson Sampling



Thompson Sampling

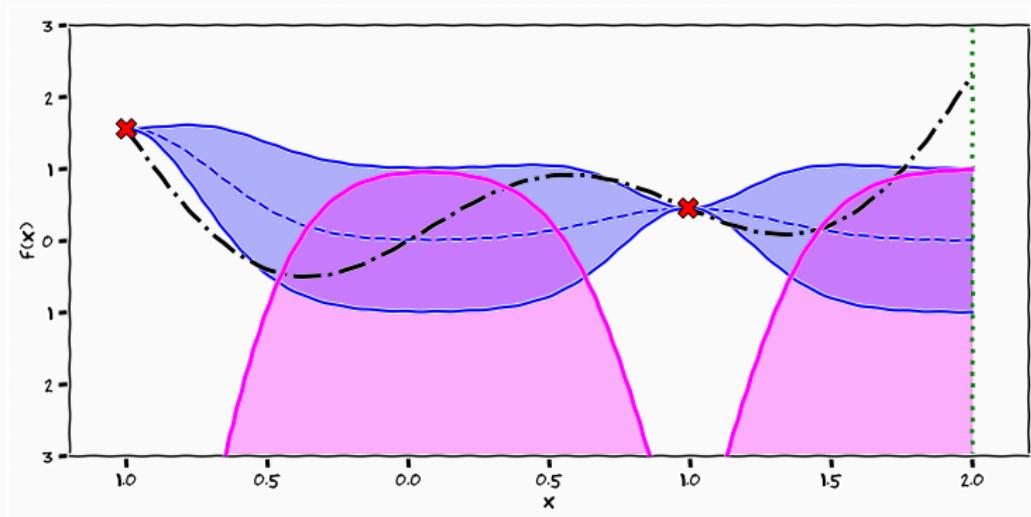




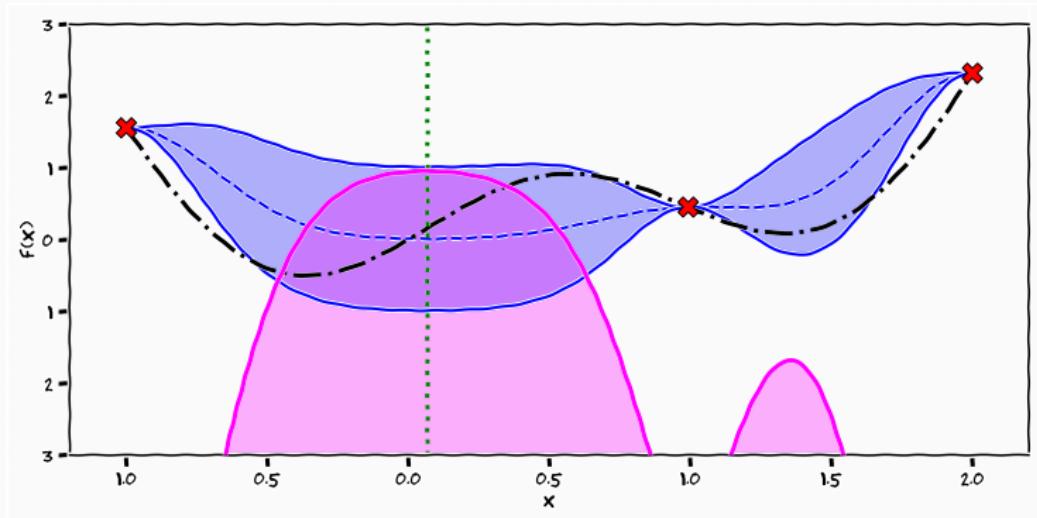
- Acquisition Function

$$\alpha(x; \{x_i, y_i\}_{i=1}^n, \mathcal{M}_n) = \mu(x; \{x_i, y_i\}_{i=1}^n) + \beta \sigma(x; \{x_i, y_i\}_{i=1}^n)$$

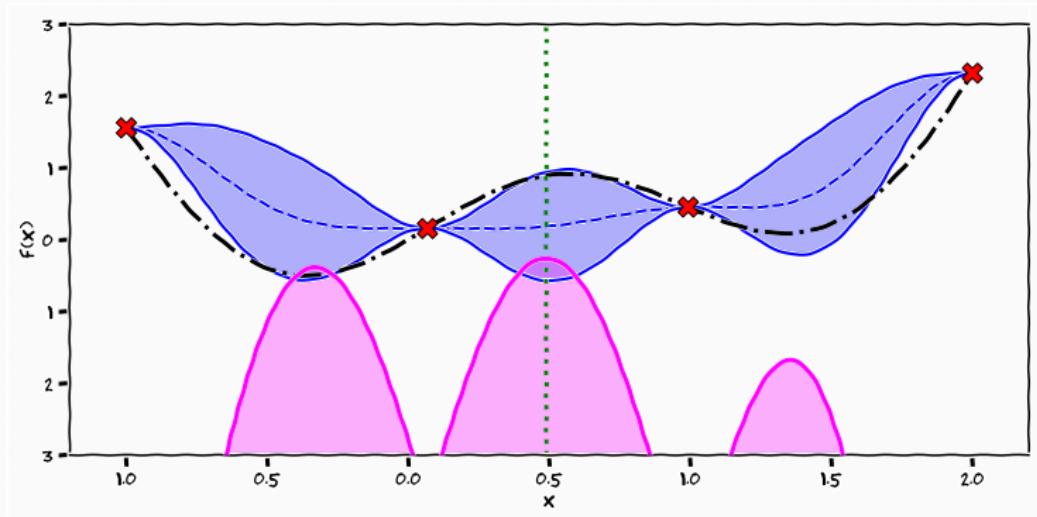
Upper Confidence Bound



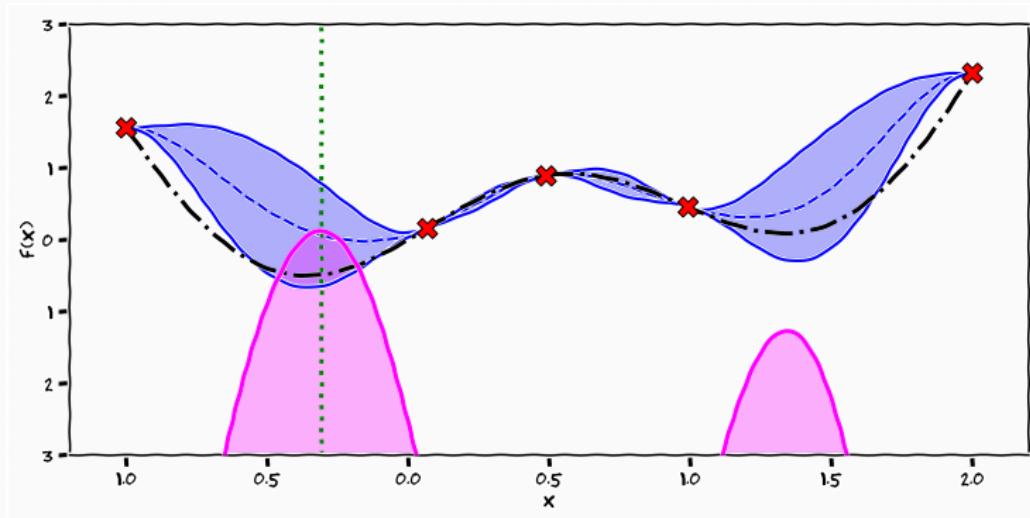
Upper Confidence Bound



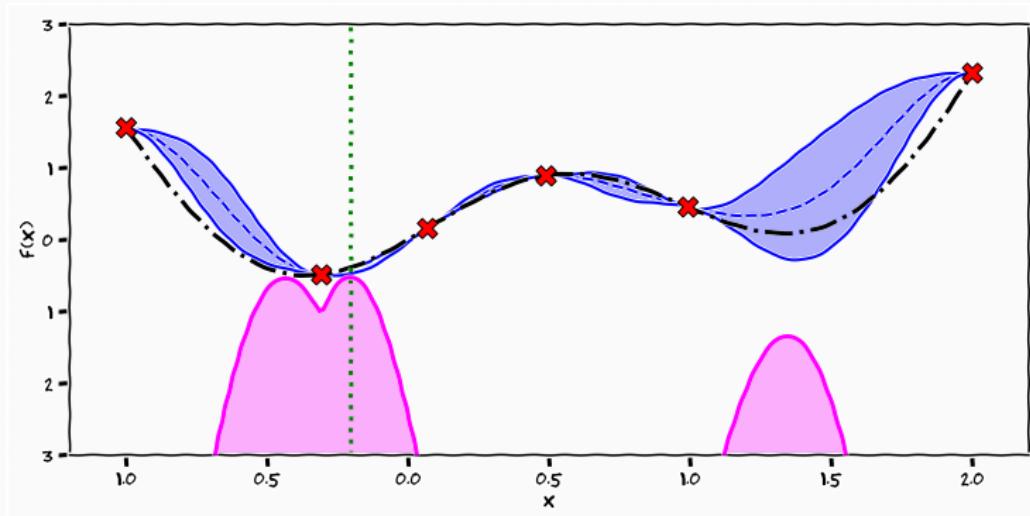
Upper Confidence Bound



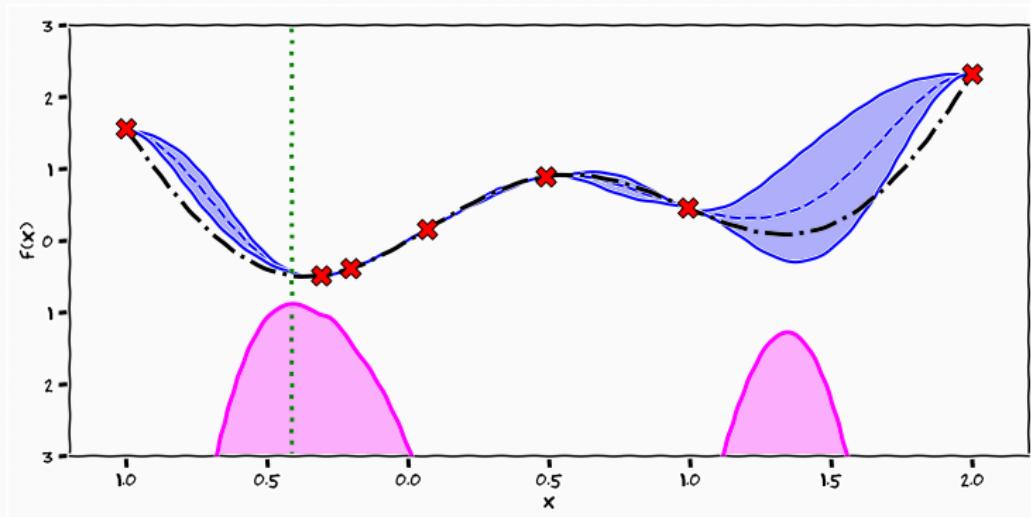
Upper Confidence Bound



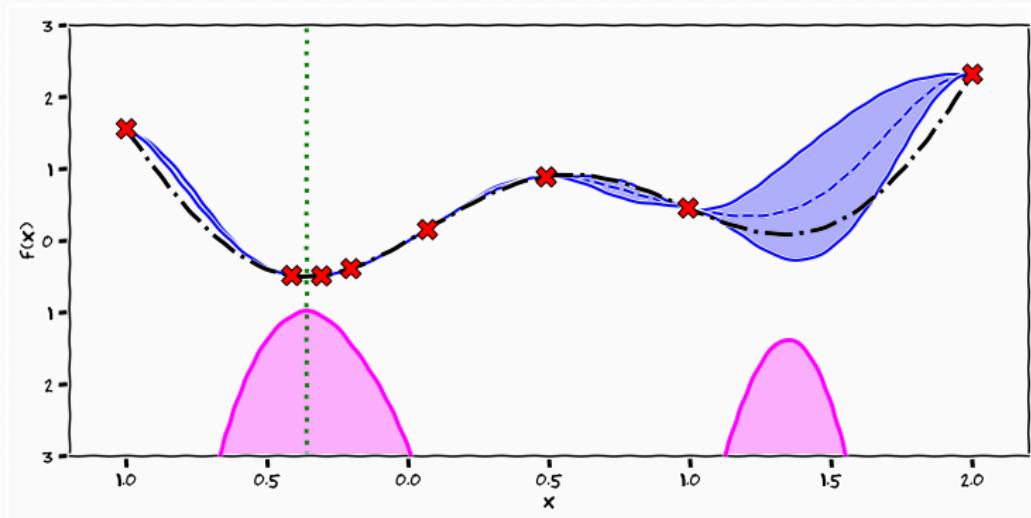
Upper Confidence Bound



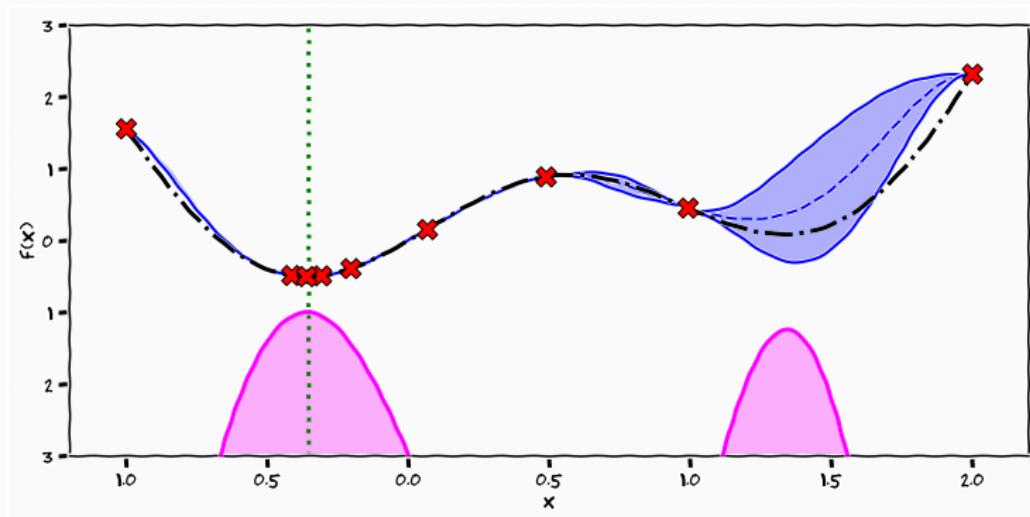
Upper Confidence Bound



Upper Confidence Bound



Upper Confidence Bound



Utility

- We can come up with lots of heuristics of how to define acquisition functions

Utility

- We can come up with lots of heuristics of how to define acquisition functions
- Define a function that defines the **utility** of observing each location

$$u(x, f(x^{(*)}), \mathcal{M}_n)$$

Utility

- We can come up with lots of heuristics of how to define acquisition functions
- Define a function that defines the **utility** of observing each location

$$u(x, f(x^{(*)}), \mathcal{M}_n)$$

- Define the acquisition function as the expected utility

$$\begin{aligned}\alpha(x; \{x_i, y_i\}_{i=1}^n, \mathcal{M}_n) &= \mathbb{E}_{p(f)}[u(x)] \\ &= \int u(x, f(x^{(*)}), \mathcal{M}_n) p(f \mid \{x_i, y_i\}_{i=1}^n) df\end{aligned}$$

- Utility Function

$$u(x) = \begin{cases} 0 & f(x) > f(x^{(*)}) \\ 1 & f(x) \leq f(x^{(*)}) \end{cases}$$

Probability of Improvement Kushner, 1963

- Utility Function

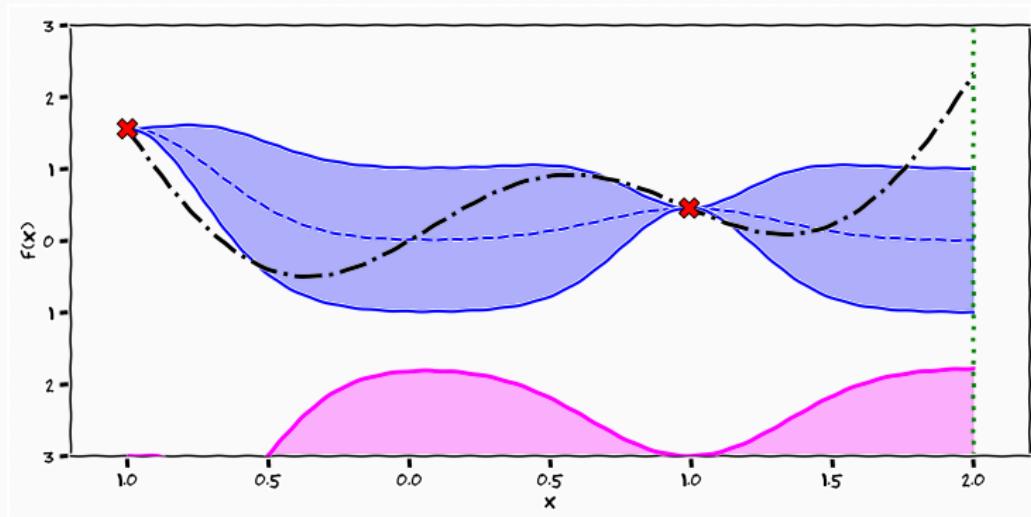
$$u(x) = \begin{cases} 0 & f(x) > f(x^{(*)}) \\ 1 & f(x) \leq f(x^{(*)}) \end{cases}$$

- Acquisition Function

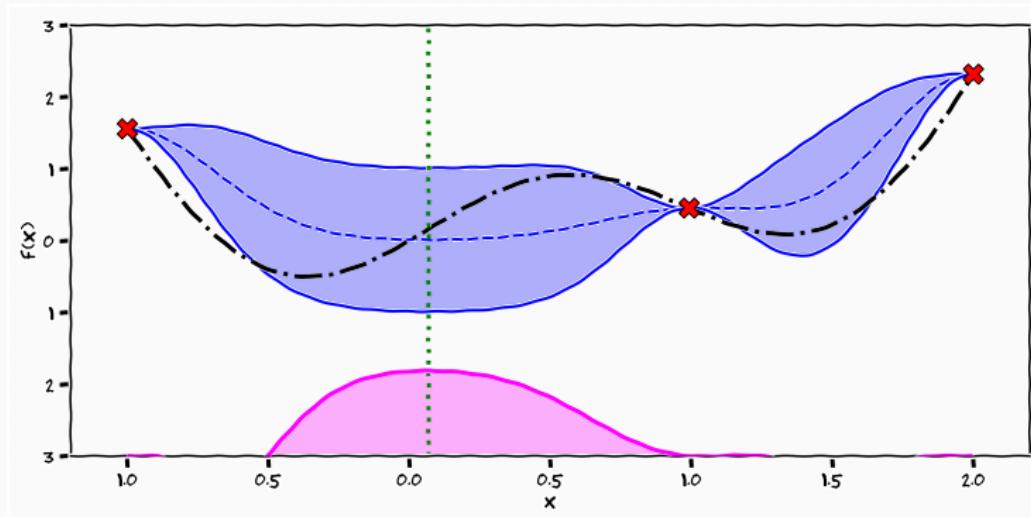
$$\alpha(x; \{x_i, y_i\}_{i=1}^n, f(x^{(*)}), \mathcal{M}_n) = \mathbb{E}[u(x)] = p(f(x) \geq f(x^{(*)}))$$

$$\begin{aligned} &= \int_{-\infty}^{f(x^{(*)})} \mathcal{N}(f \mid \mu(x), K(x, x)) df \\ &= \Phi\left(f(x^{(*)}) \mid \mu(x), K(x, x)\right) \end{aligned}$$

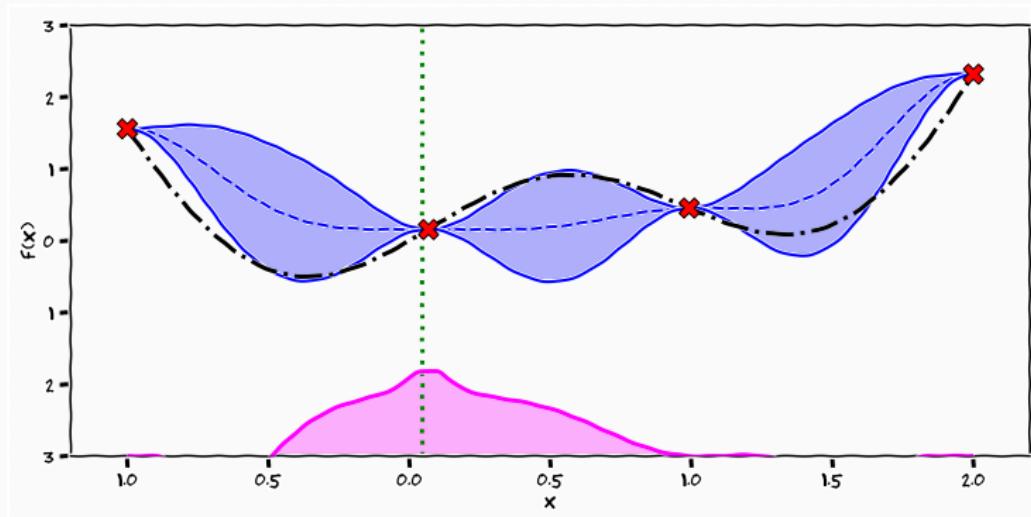
Probability of Improvement



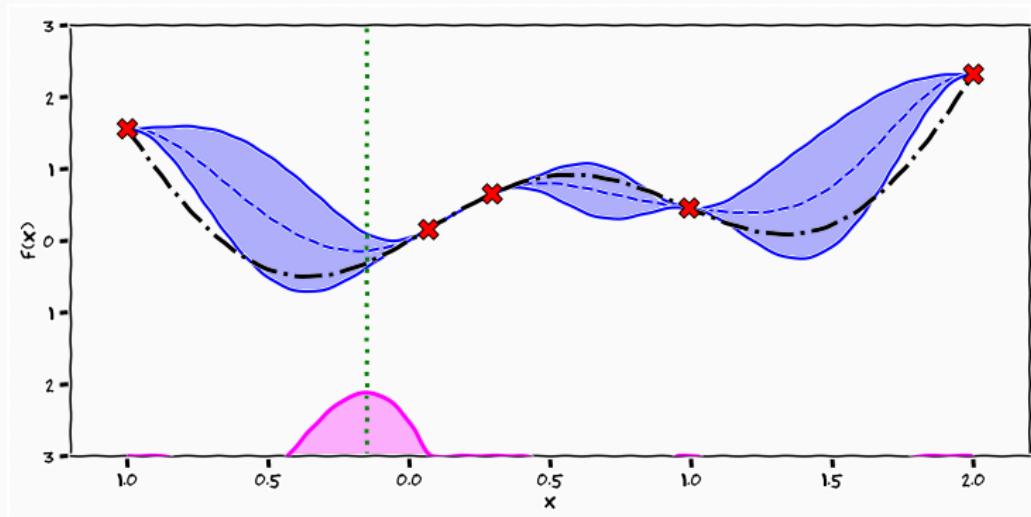
Probability of Improvement



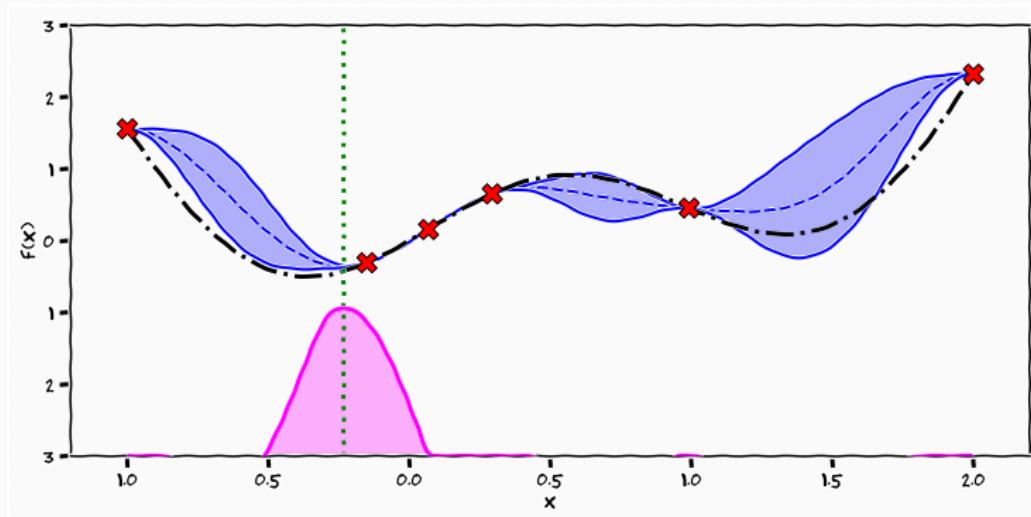
Probability of Improvement



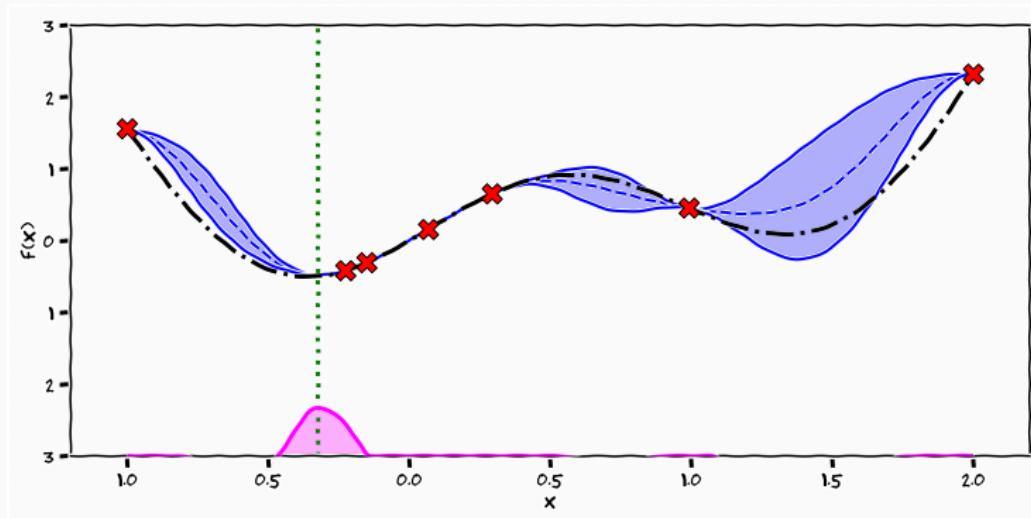
Probability of Improvement



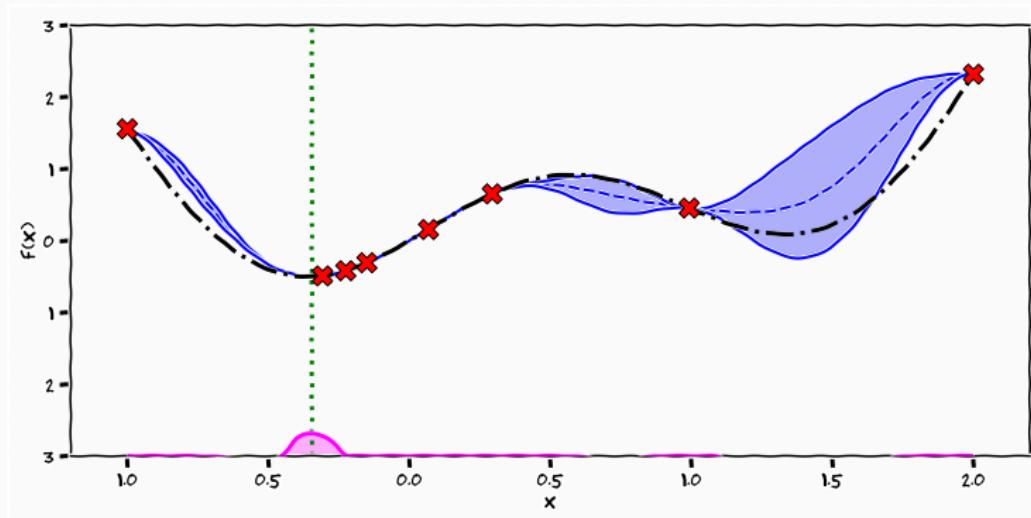
Probability of Improvement



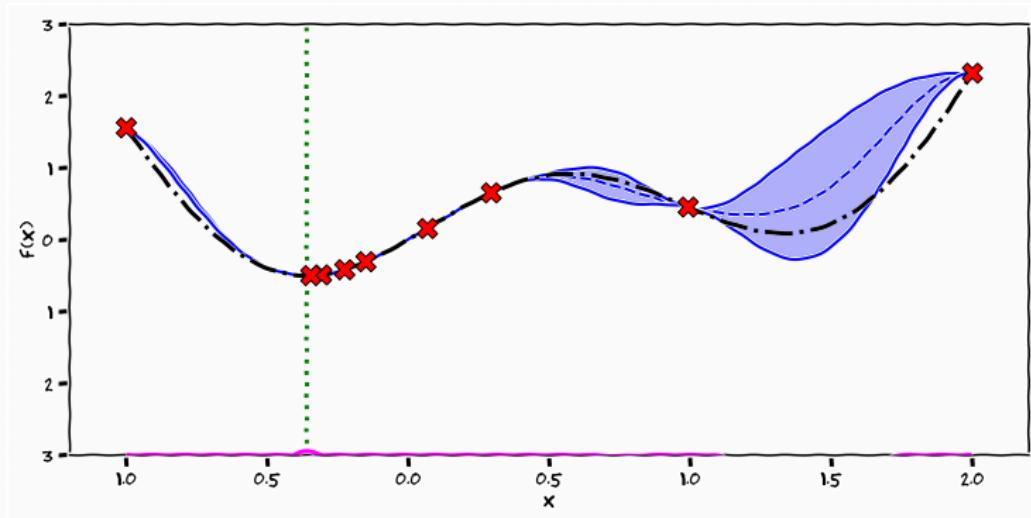
Probability of Improvement



Probability of Improvement



Probability of Improvement



- Utility Function

$$u(x) = \max(0, f(x^{(*)}) - f(x))$$

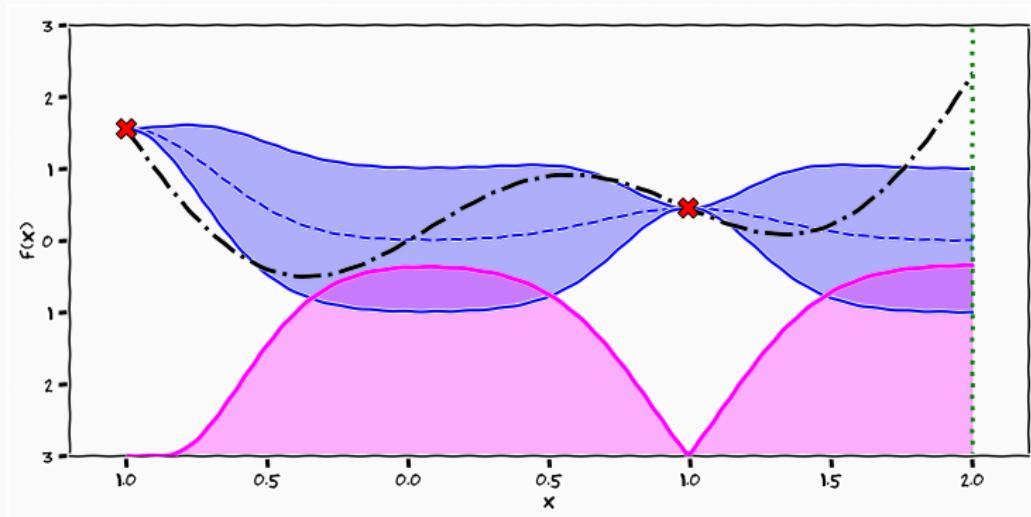
- Utility Function

$$u(x) = \max(0, f(x^{(*)}) - f(x))$$

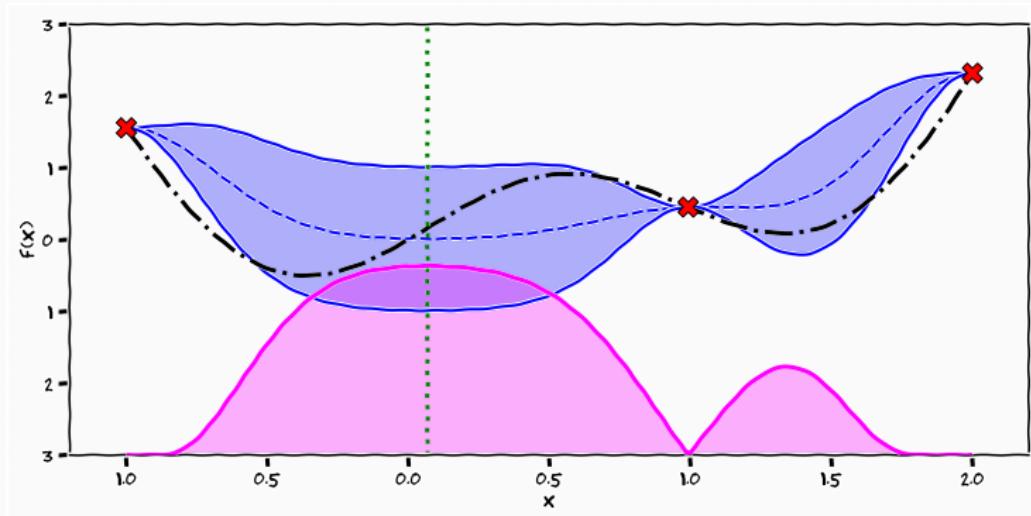
- Acquisition Function

$$\begin{aligned}\alpha(x; \{x_i, y_i\}_{i=1}^n, f(x^{(*)}), \mathcal{M}_n) &= \mathbb{E}[u(x)] \\ &= \int_{-\infty}^{f(x^{(*)})} (f(x^{(*)}) - f) \mathcal{N}(f \mid \mu(x), K(x, x)) df \\ &= (f(x^{(*)}) - \mu(x)) \Phi \left(f(x^{(*)}) \mid \mu(x), K(x, x) \right) \\ &\quad + K(x, x) \mathcal{N} \left(f(x^{(*)}) \mid \mu(x), K(x, x) \right)\end{aligned}$$

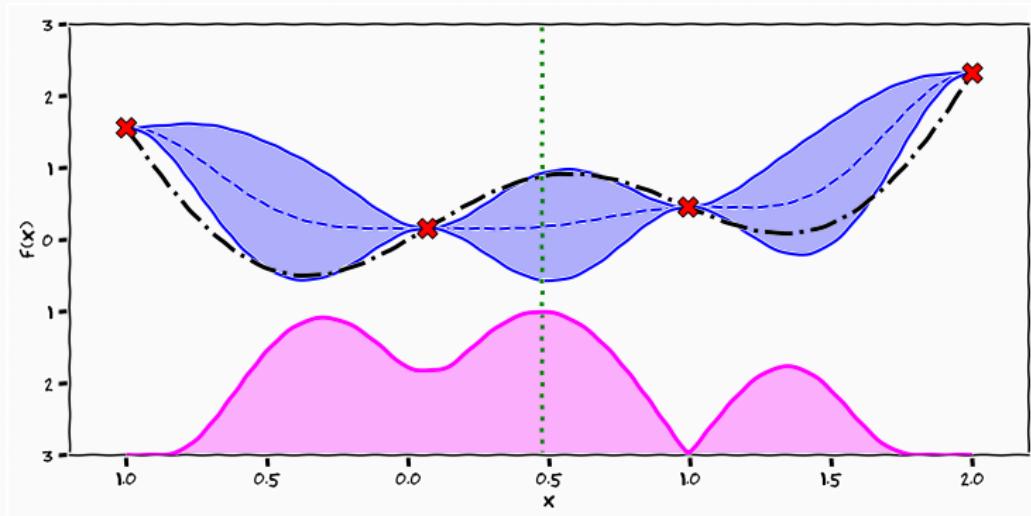
Expected Improvement



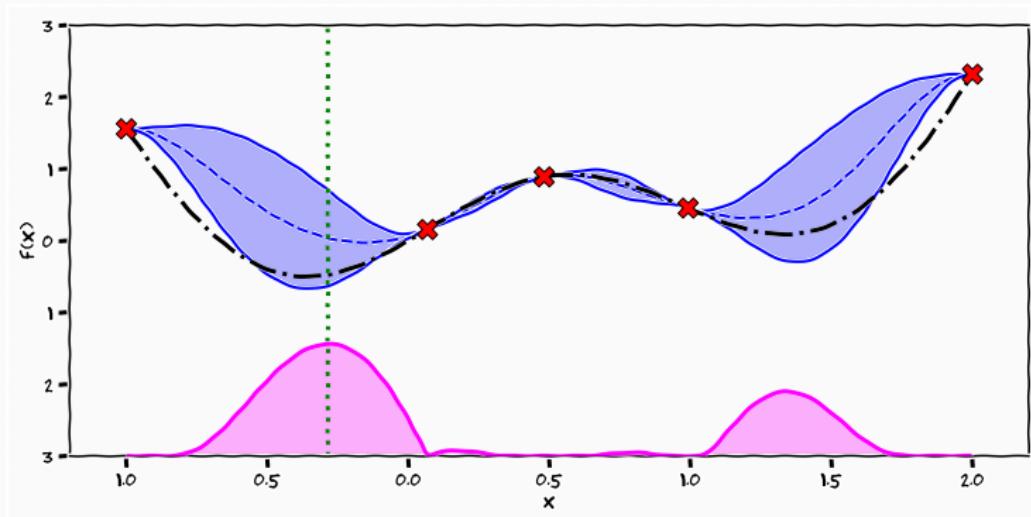
Expected Improvement



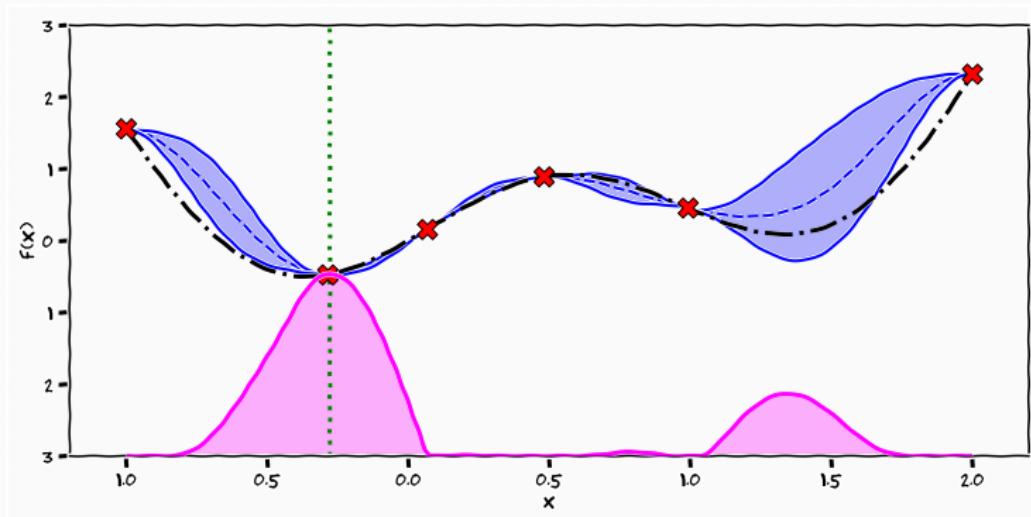
Expected Improvement



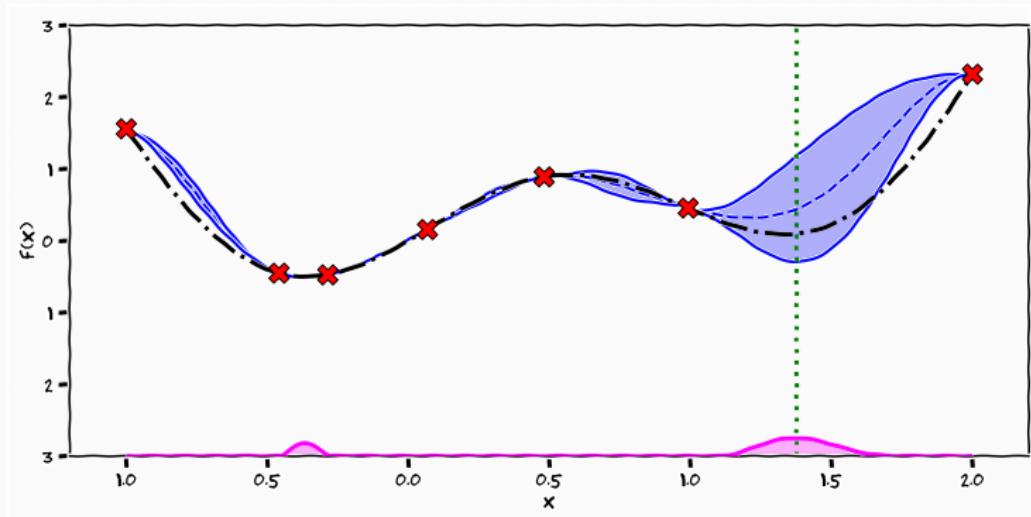
Expected Improvement



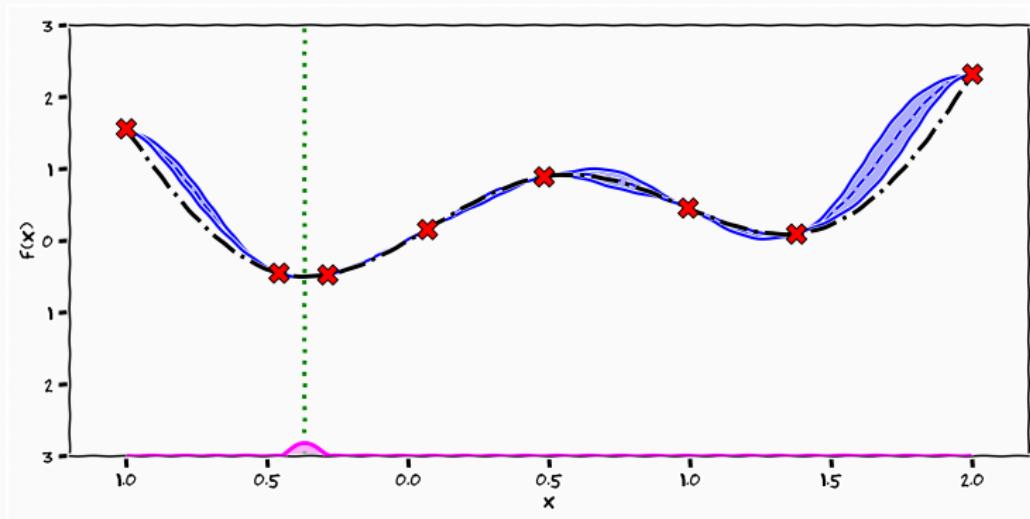
Expected Improvement



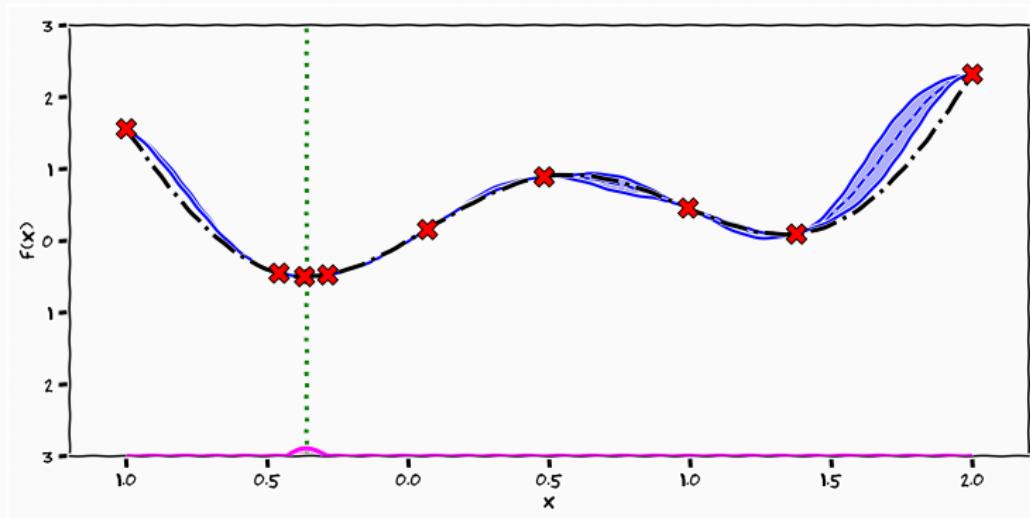
Expected Improvement



Expected Improvement



Expected Improvement



Task 1 encode your knowledge about **the function** in the GP prior

²till they open the door to the exam.

Task 1 encode your knowledge about **the function** in the GP prior

Task 2 randomly sample some data

²till they open the door to the exam.

Task 1 encode your knowledge about **the function** in the GP prior

Task 2 randomly sample some data

Task 3 specify your acquisition function

²till they open the door to the exam.

Task 1 encode your knowledge about **the function** in the GP prior

Task 2 randomly sample some data

Task 3 specify your acquisition function

Task 4 evaluate and maximise the acquisition function

²till they open the door to the exam.

Task 1 encode your knowledge about **the function** in the GP prior

Task 2 randomly sample some data

Task 3 specify your acquisition function

Task 4 evaluate and maximise the acquisition function

Task 5 add new data to model and **re-estimate** hyperparameters

²till they open the door to the exam.

Task 1 encode your knowledge about **the function** in the GP prior

Task 2 randomly sample some data

Task 3 specify your acquisition function

Task 4 evaluate and maximise the acquisition function

Task 5 add new data to model and **re-estimate** hyperparameters

Loop 4-5 till budget is gone²

²till they open the door to the exam.

Learning Hyper-parameters

$$\{\hat{\beta}, \hat{\ell}, \hat{\sigma}\} = \operatorname{argmax}_{\beta, \ell, \sigma} \log \int p(y | f) p(f) df$$

- Update the **hyper-parameters** of the GP
- We can do this by gradient based optimisation
 - *or with a Bayes Opt loop!*

³....and here I was thinking that you guys had some principles in life

Learning Hyper-parameters

$$\{\hat{\beta}, \hat{\ell}, \hat{\sigma}\} = \operatorname{argmax}_{\beta, \ell, \sigma} \log \int p(y | f) p(f) df$$

- Update the **hyper-parameters** of the GP
- We can do this by gradient based optimisation
 - or with a *Bayes Opt loop!*
- Are we doing **MLE**³?

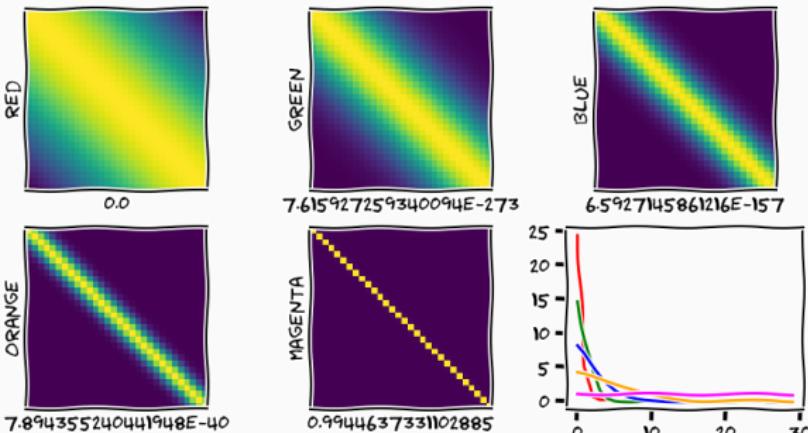
³....and here I was thinking that you guys had some principles in life

Marginal Likelihood

$$\begin{aligned}\{\hat{\beta}, \hat{\ell}, \hat{\sigma}\} &= \operatorname{argmax}_{\beta, \ell, \sigma} \log \int p(y | f) p(f) df \\&= \operatorname{argmin}_{\beta, \ell, \sigma} -\log p(y) \\&= \operatorname{argmin}_{\beta, \ell, \sigma} \frac{1}{2} \text{trace}(\mathbf{YK}^{-1}\mathbf{Y}^T) + \frac{1}{2} \log|K| + \frac{N}{2} \log(2\pi)\end{aligned}$$

- *Data – fit* how well does the observations fit the model
- *"Complexity"* how "smooth" is the functions

Determinant



$$\log|K| = \sum_{i=1}^N \lambda_i$$

How to implement



Summary

Summary

- GPs are quite useful surrogates!

Summary

- GPs are quite useful surrogates!
- Degrees of beliefs are **really** useful

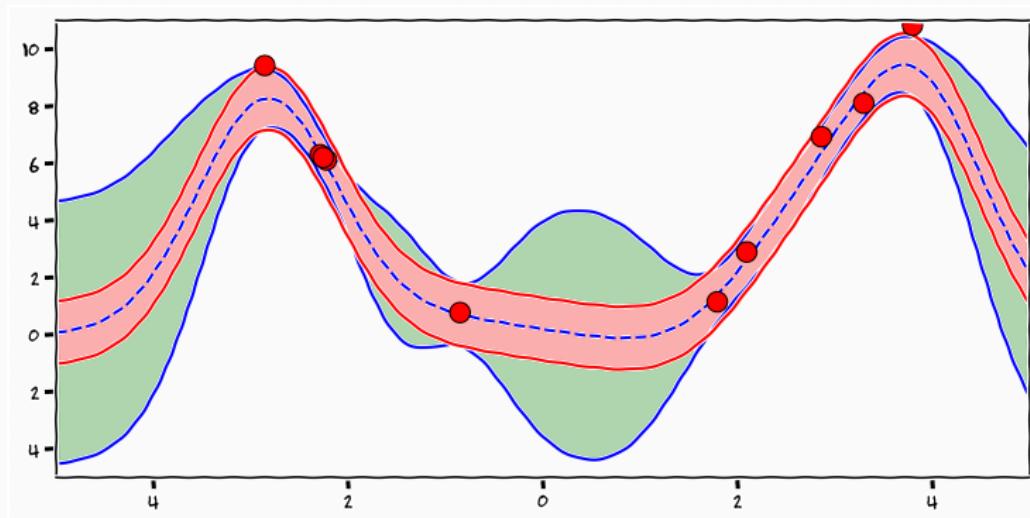
Summary

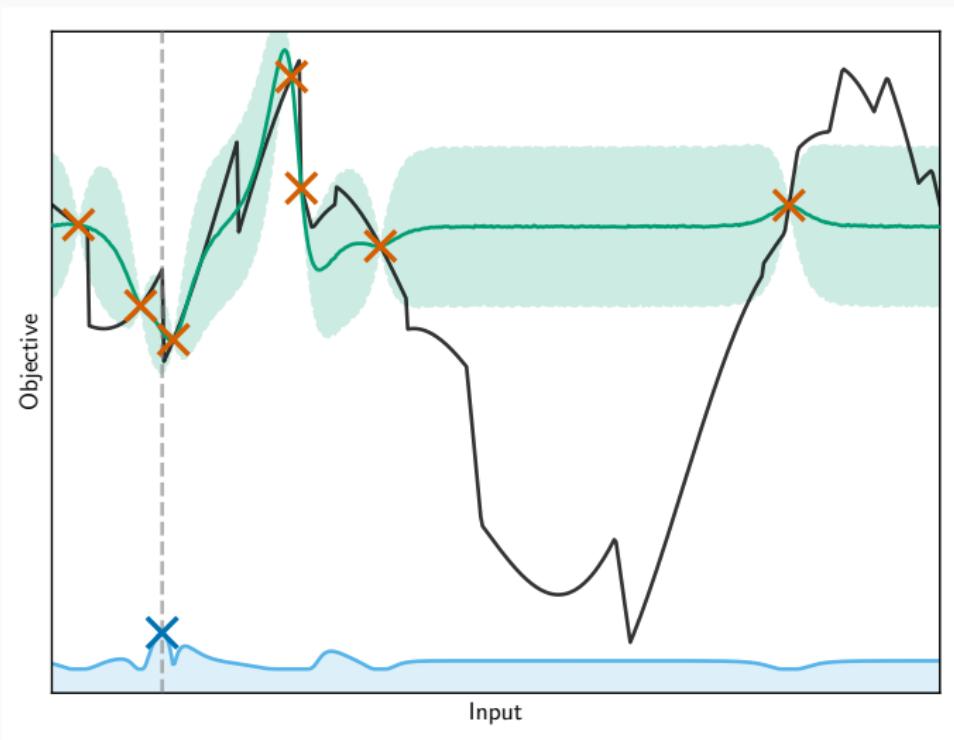
- GPs are quite useful surrogates!
- Degrees of beliefs are **really** useful
- The uncertainty allows us to design rich strategies for how to acquire data

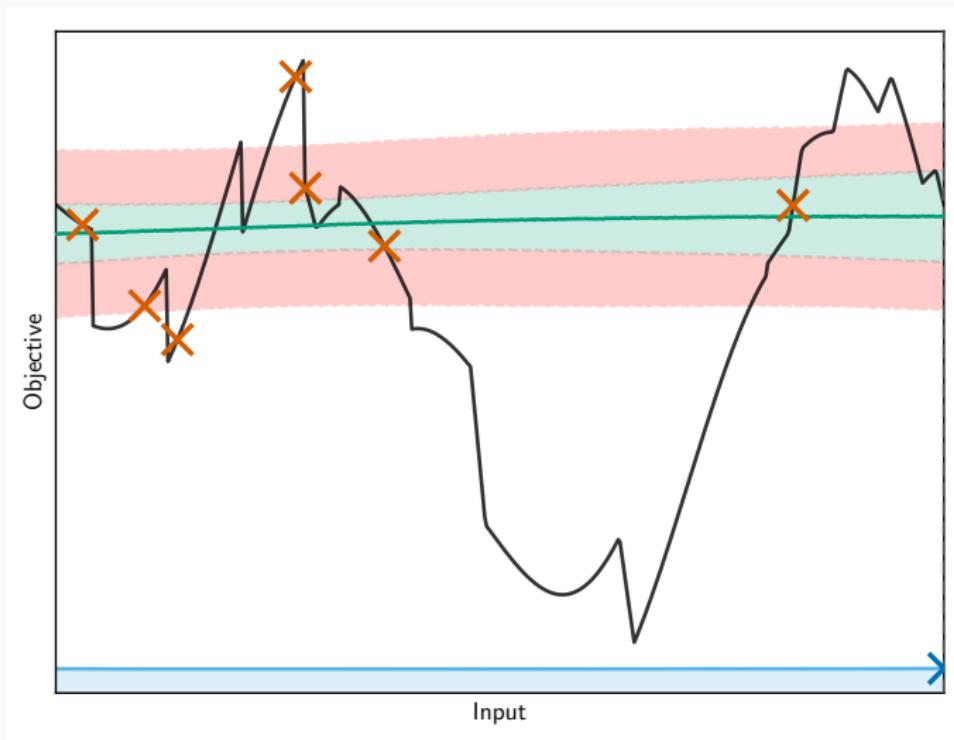
Summary

- GPs are quite useful surrogates!
- Degrees of beliefs are **really** useful
- The uncertainty allows us to design rich strategies for how to acquire data
- The factorisation of uncertainty allows us to describe search strategies in simple acquisition functions

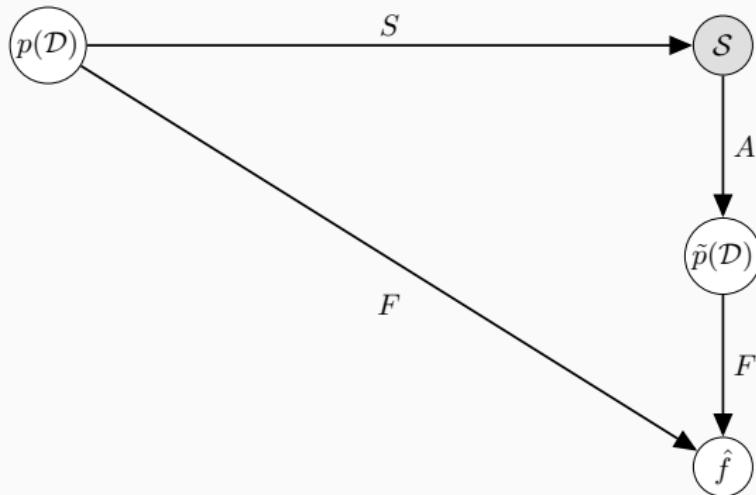
Uncertainty Quantification/Factorisation





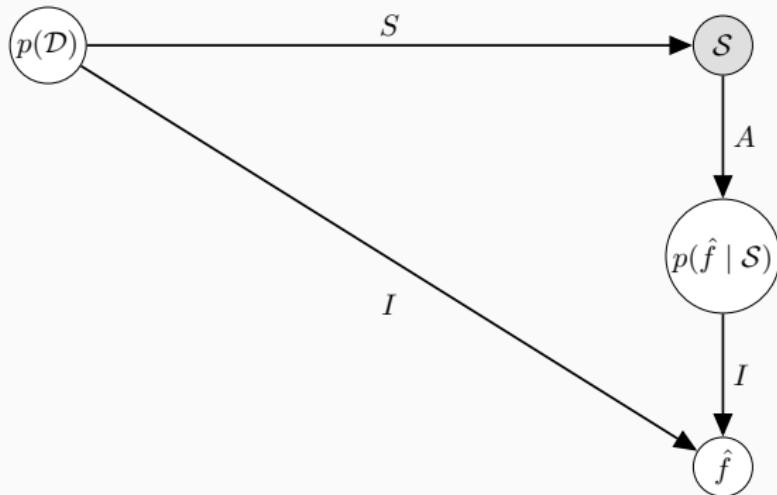


Formalisation

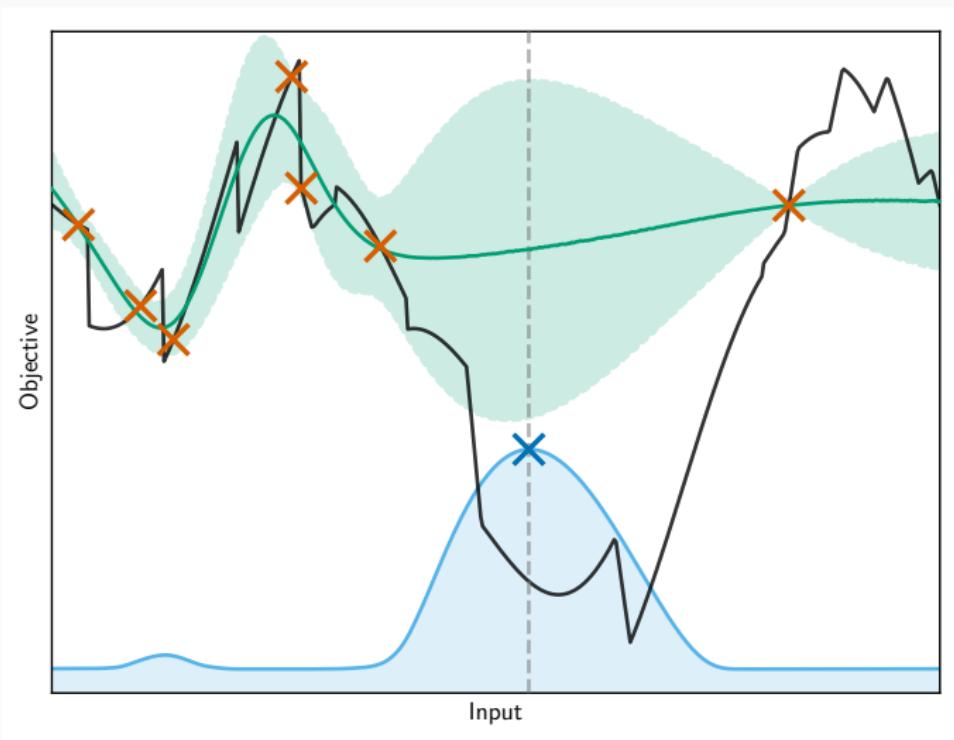


$$A \circ S \circ p(\mathcal{D}) \approx p(\mathcal{D})$$

Formalisation



$$A \circ S \circ p(\mathcal{D}) = p(\hat{f} \mid \mathcal{S})$$



- There are tons of numerical algorithms every problem under the sun

- There are tons of numerical algorithms every problem under the sun
- They work really well

- There are tons of numerical algorithms every problem under the sun
- They work really well
- They give different results on the same problem

- There are tons of numerical algorithms every problem under the sun
- They work really well
- They give different results on the same problem
- The No-Free Lunch

- There are tons of numerical algorithms every problem under the sun
- They work really well
- They give different results on the same problem
- The No-Free Lunch
- *what is the prior they implement?*

eof

References

-  Bodin, Erik et al. (2020). "Modulating Surrogates for Bayesian Optimization.". In: *Proceedings of the 37th International Conference on Machine Learning, ICML 2019, 12-18 July 2020, Virtual.*
-  Brochu, Eric, Vlad M. Cora, and Nando de Freitas (2010). "A Tutorial on Bayesian Optimization of Expensive Cost Functions, With Application To Active User Modeling and Hierarchical Reinforcement Learning". In: *CoRR*.
-  Cockayne, Jon, Chris Oates, Tim Sullivan, and Mark Girolami (2017). "Bayesian Probabilistic Numerical Methods". In: *CoRR*.

-  Cox, Dennis and Susan John (Mar. 1997). "SDO: A Statistical Method for Global Optimization". In: *Multidisciplinary Design Optimization: State of the Art*. Ed. by M. N. Alexandrov and M. Y. Hussaini, pp. 315–329.
-  Hennig, Philipp, Michael A Osborne, and Mark Girolami (July 2015). "Probabilistic numerics and uncertainty in computations". In: *Proc. R. Soc. A* 471.2179, p. 20150142.
-  Kushner, Harold J. (1963). "A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise". Undetermined. In: *Joint Automatic Control Conference* 1, pp. 69 –79.
-  Močkus, J. (1975). "On bayesian methods for seeking the extremum". In: *Optimization Techniques IFIP Technical Conference Novosibirsk, July 1–7, 1974*. Ed. by G. I. Marchuk. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 400–404.

- Mockus, J., Vytautas Tesis, and Antanas Zilinskas (Sept. 1978).
“The application of Bayesian methods for seeking the extremum”.
In: vol. 2, pp. 117–129.
- Thompson, William R. (1933). “On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples”. In: *Biometrika* 25.3/4, pp. 285–294.