CS-E4895 Gaussian Processes
Lecture 12: Sequential Decision Making

Aidan Scannell

Aalto University

Tuesday 4.4.2023

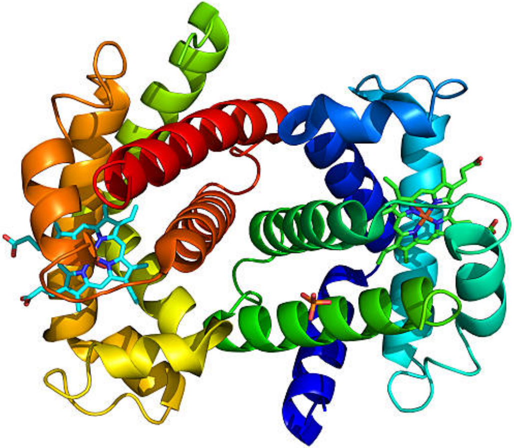
Agenda for today

1. Black-box optimisation
2. Motivation for Bayesian Optimization
3. Gaussian process surrogate
4. Decision making under uncertainty
5. Model-based reinforcement learning

Examples: Robotics and control

Examples: Neural network hyperparameter optimisation

Examples: Protein engineering



Black-box Optimisation

Goal We want to maximize (or minimize) a function $f(\cdot)$ over bounded set $\mathcal{X}$:

$$x^* = \arg \max_{x \in \mathcal{X} \subseteq \mathbb{R}^D} f(x) \tag{1}$$

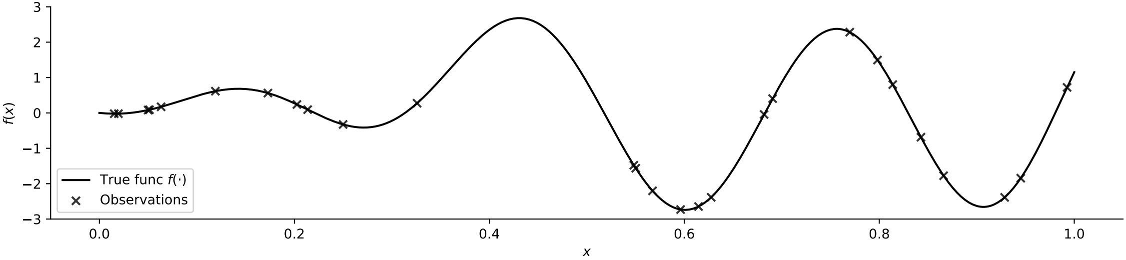
- $\mathcal{X}$ is a bounded domain
- $f(\cdot)$ is explicitly unknown
- Samples of $f(\cdot)$ may be noisy
- $f(\cdot)$ is expensive to evaluate

Random Search (No Exploitation)

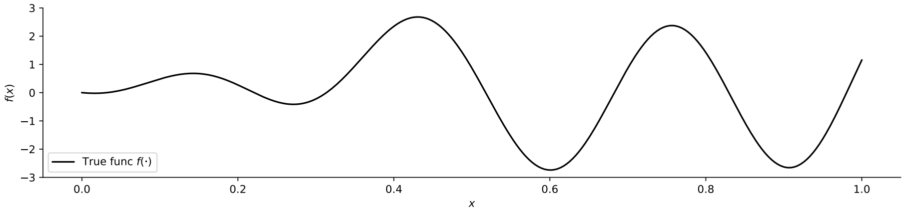
- Random search

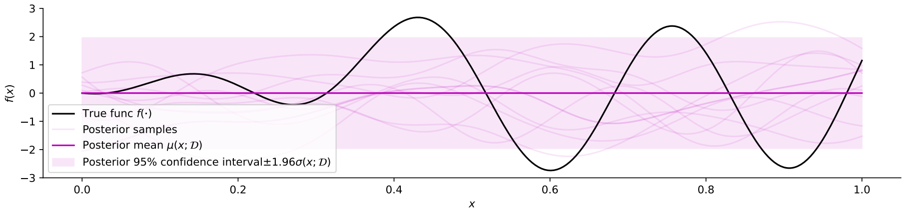
$$f(x^+) \geq f(x^*) - \epsilon \tag{2}$$

- Lipschitz continuos: $\|f(x_1) - f(x_2)\| \leq C\|x_1 - x_2\|$
- Requires $(\frac{C}{2\epsilon})^d$ samples on d-dimensional unit hypercube



Gaussian Process Surrogate





- Probability measure on f, e.g. place a GP prior over f
  - Principled prior to encode our belief
  - Update prior to posterior using available data

Acquisition function

Formulate a sequential decision-making problem:

$$x_{n+1} = \arg\max_{x \in \mathcal{X}} \alpha(x; \mathcal{D}), \qquad \mathcal{D} = \{x_i, y_i\}_{i=0}^{n} \tag{3}$$

- Acquisition function $\alpha : \mathcal{X} \rightarrow \mathbb{R}$ assigns score to each potential observation location
- We want to make sequence of N samples, $x_1, \ldots, x_N$, which minimises regret

$$r = N f(x^*) - \sum_{n=1}^{N} f(x_n) \tag{4}$$

- Replace hard optimisation (expensive+no gradients) with another:
  - α should be cheap to evaluate
  - α needs to balance exploration/exploitation
    - Minimise number of objective function evaluations
    - Whilst maximising information gain about **global** optimum, i.e performs well when objective has multiple local maxima

Bayesian Optimisation

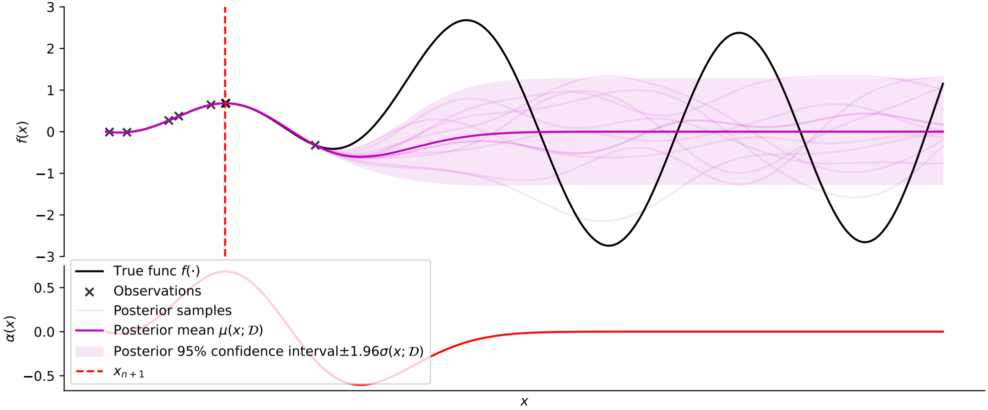
- **Input:** Initial dataset $\mathcal{D}$
- **Repeat:**
  - GP $\leftarrow$ FIT($\mathcal{D}$)
  - $x \leftarrow$ POLICY(GP)
  - $y \leftarrow$ OBSERVE(x)
  - $\mathcal{D}' \leftarrow \mathcal{D} \cup (x, y)$
- **Until** Termination condition is met.
- What is our policy?
- Predictive posterior at n^{th} sample

$$p(f(x) \mid x, \mathcal{D}) = \mathcal{N}(f(x) \mid \mu(x; \mathcal{D}), \sigma^2(x; \mathcal{D})) \tag{5}$$

with data $\mathcal{D} = \{x_i, y_i\}_{i=0}^{n}$

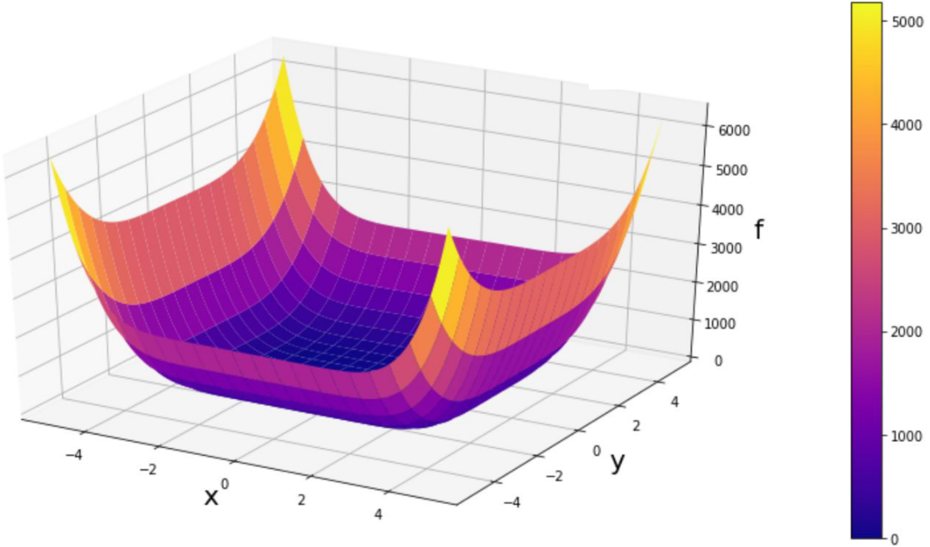
Acquisition Function: Posterior Mean (No Exploration)

$$\alpha_{\mu}(x \mid \mathcal{D}) = \mu(x; \mathcal{D}) \tag{6}$$



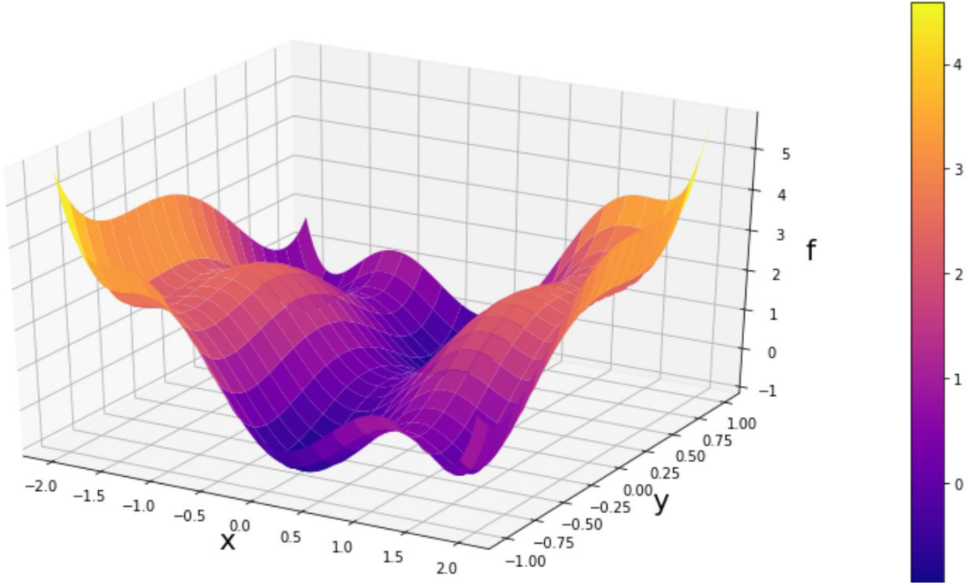
Need to explore sometimes

- Consider the 6 Hump Camel function



Need to explore sometimes

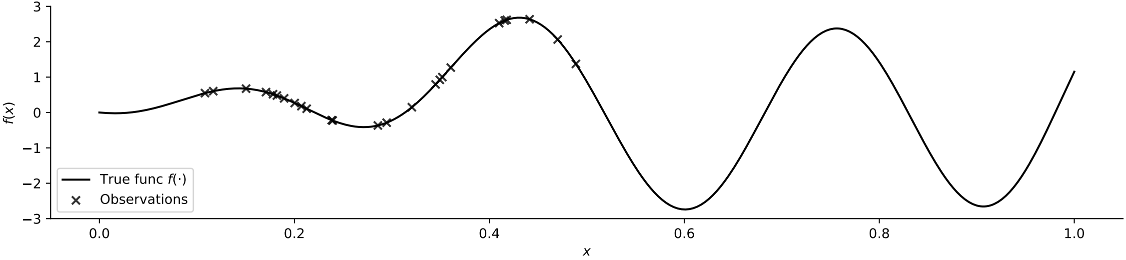
- We **cannot** use a local optimizer!

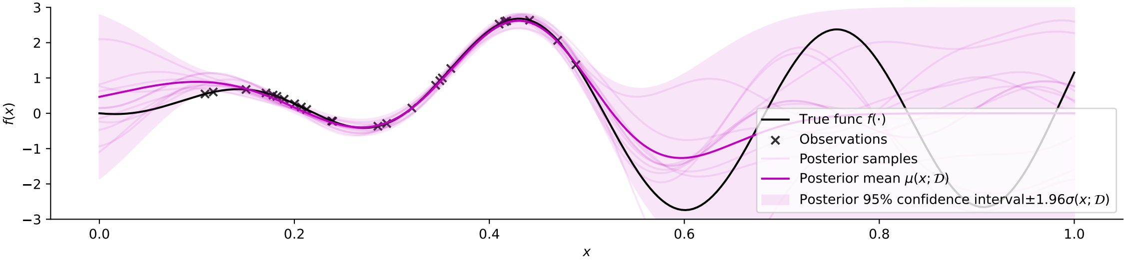


Exploration vs Exploitation

- **Exploitation** - use the knowledge we have
  - i.e. pick x where we expect the objective function to be high
- **Exploration** - attempt to gain new knowledge
  - i.e. pick x where the objective function is uncertain

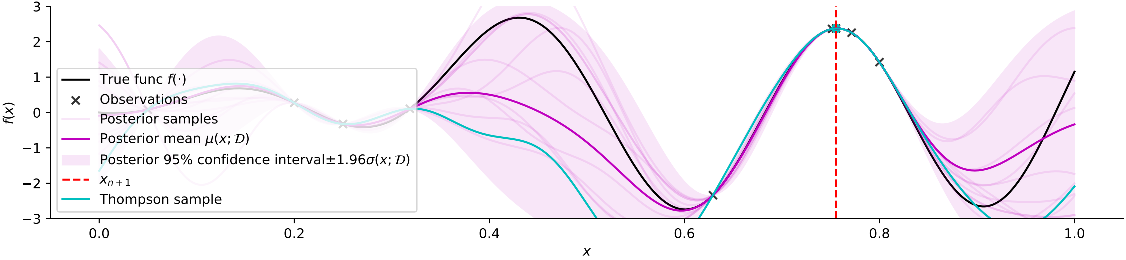
Sources of Uncertainty





Thompson sampling

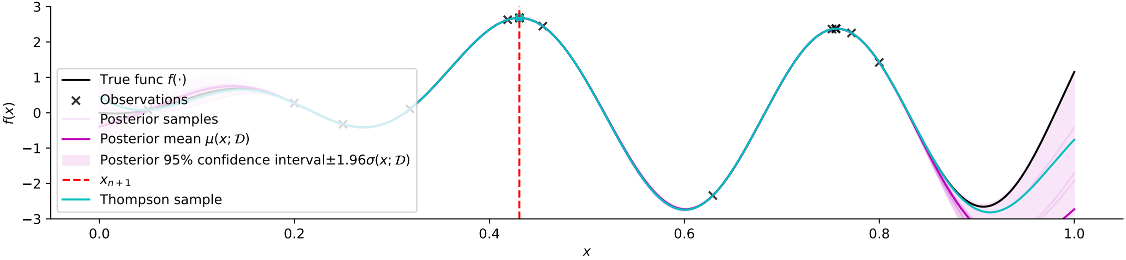
$$x_{n+1} = \arg\max_{x \in \mathcal{X}} \alpha_{\mathsf{TS}}(x; \mathcal{D}), \qquad \alpha_{\mathsf{TS}}(x; \mathcal{D}) \sim p(f(x) \mid x, \mathcal{D}) \tag{7}$$



- Sampling functions is not trivial
- Easy solution for 'small' domains
- Not so easy in multiple dimensional and bigger domains

Thompson sampling

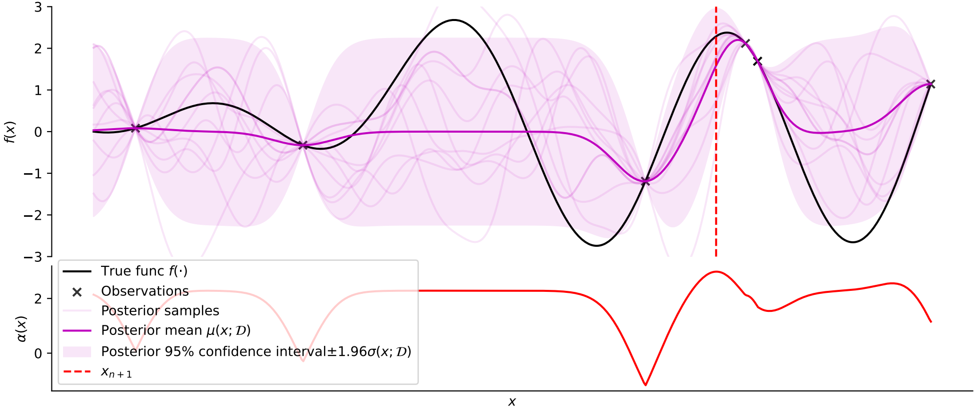
$$x_{n+1} = \arg\max_{x \in \mathcal{X}} \alpha_{\mathsf{TS}}(x; \mathcal{D}), \qquad \alpha_{\mathsf{TS}}(x; \mathcal{D}) \sim p(f(x) \mid x, \mathcal{D}) \tag{7}$$



- Sampling functions is not trivial
- Easy solution for 'small' domains
- Not so easy in multiple dimensional and bigger domains

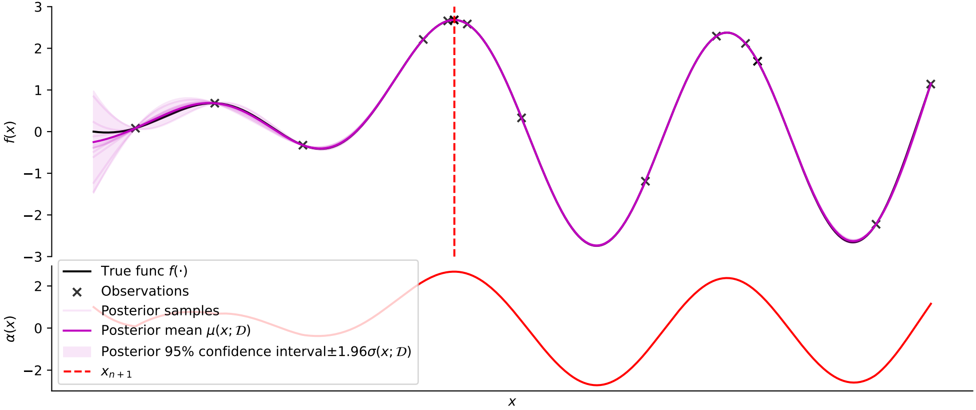
Upper Confidence Bound

$$x_{n+1} = \arg\max_{x \in \mathcal{X}} \alpha_{\text{UCB}}(x; \mathcal{D}), \qquad \alpha_{\text{UCB}}(x \mid \mathcal{D}) = \mu(x; \mathcal{D}) + \beta_n \sigma(x; \mathcal{D}) \tag{8}$$



Upper Confidence Bound

$$x_{n+1} = \arg\max_{x \in \mathcal{X}} \alpha_{\mathsf{UCB}}(x; \mathcal{D}), \qquad \alpha_{\mathsf{UCB}}(x \mid \mathcal{D}) = \mu(x; \mathcal{D}) + \beta_n \sigma(x; \mathcal{D}) \tag{8}$$



Utility

- Lots of heuristics for defining acquisition functions
- Specify **utility function** $u(x, f(x^+))$ that defines utility of observing each location
- Data at n^{th} iteration $\mathcal{D} = \{x_i, y_i\}_{i=0}^n$
- Define **acquisition function** as expected marginal utility after observing new x

$$\alpha(x; \mathcal{D}) = \mathbb{E}_{p(f(x) | x, \mathcal{D})}[u(x)] \tag{9}$$

under GP posterior

$$p(f(x) \mid x, \mathcal{D}) = \mathcal{N}(f(x) \mid \mu(x; \mathcal{D}), \sigma^2(x; \mathcal{D})) \tag{10}$$

- Different utility functions leads to different acquisition functions

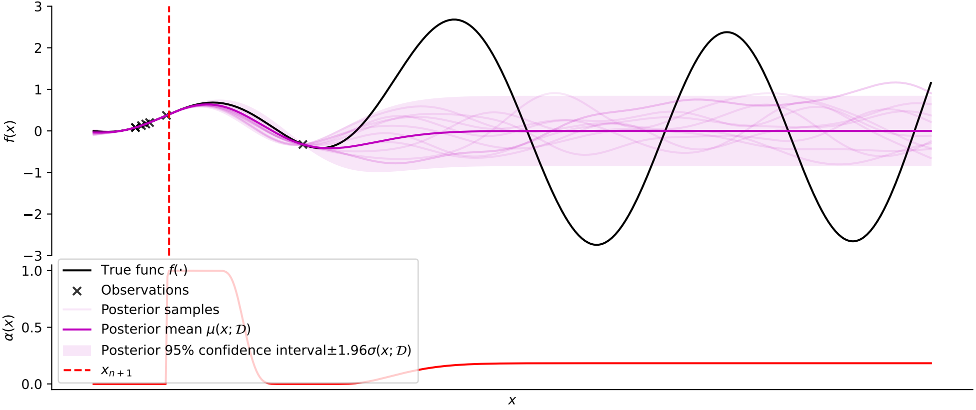
Probability of Improvement

$$u(x) = \begin{cases} 0 & f(x) \leq f\left(x^{+}\right) \\ 1 & f(x) > f\left(x^{+}\right) \end{cases} \quad \text{where } f(x^{+}) = \max_{x_i \in x_{0:n}} f(x_i) \tag{11}$$

$$\alpha_{\mathsf{PI}}(x \mid \mathcal{D}) = \mathbb{E}[u(x)] = \Pr\left(f(x) \geq f(x^{+})\right) = \Phi\left(\frac{\mu(x; \mathcal{D}) - f(x^{+})}{\sigma(x; \mathcal{D})}\right)$$

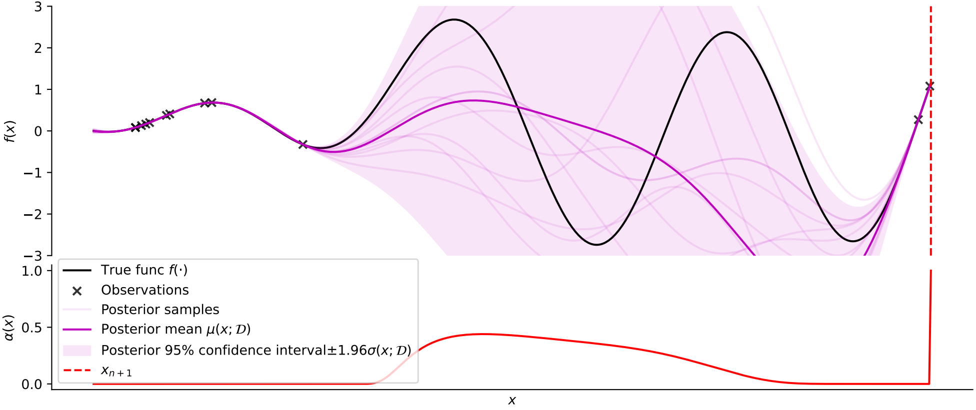
Probability of Improvement

$$\alpha_{\mathsf{PI}}(x \mid \mathcal{D}) = \mathbb{E}[u(x)] = \Pr\left(f(x) \geq f(x^{+})\right) = \Phi\left(\frac{\mu(x;\mathcal{D}) - f(x^{+})}{\sigma(x;\mathcal{D})}\right)$$



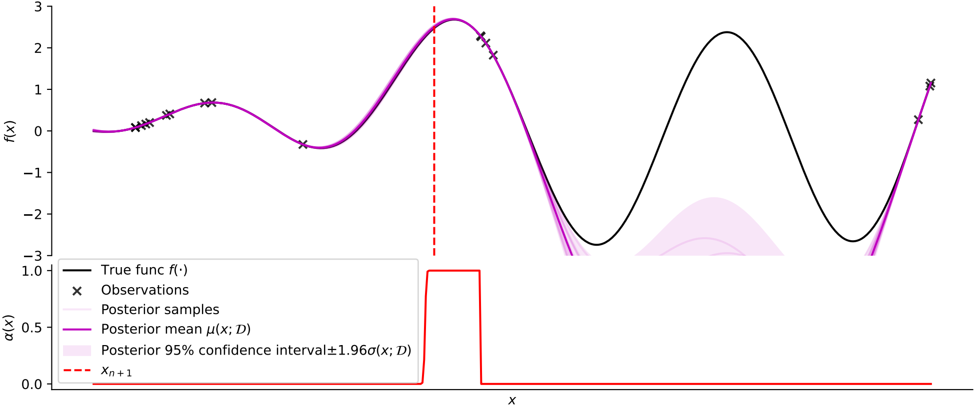
Probability of Improvement

$$\alpha_{\mathsf{PI}}(x \mid \mathcal{D}) = \mathbb{E}[u(x)] = \Pr\left(f(x) \geq f(x^{+})\right) = \Phi\left(\frac{\mu(x;\mathcal{D}) - f(x^{+})}{\sigma(x;\mathcal{D})}\right)$$



Probability of Improvement

$$\alpha_{\mathsf{PI}}(x \mid \mathcal{D}) = \mathbb{E}[u(x)] = \Pr\left(f(x) \geq f(x^{+})\right) = \Phi\left(\frac{\mu(x;\mathcal{D}) - f(x^{+})}{\sigma(x;\mathcal{D})}\right)$$



Expected Improvement

- Define the utility function as: $u(\mathcal{D}) = \max(0, f(x) - f(x^{+}))$

$$
\begin{aligned}
\alpha_{EI}(x; \mathcal{D}) = \mathbb{E}[u(x)] &= \int \max(0, f(x) - f(x^{+})) p(f(x) \mid x, \mathcal{D}) df(x) \\
&= \int \max(0, f(x) - f(x^{+})) \mathcal{N}(f(x); \mu(x; \mathcal{D}), \sigma^2(x; \mathcal{D})) df(x) \\
&= \int_{f(x^{+})}^{\infty} f(x) \mathcal{N}(f(x); \mu(x; \mathcal{D}), \sigma^2(x; \mathcal{D})) df(x) \\
&\quad - f(x^{+}) \int_{f(x^{+})}^{\infty} \mathcal{N}(f(x); \mu(x; \mathcal{D}), \sigma^2(x; \mathcal{D})) df(x)
\end{aligned}
$$

- First term is truncated expected value and second term compliment of CDF multiplied by a constant.

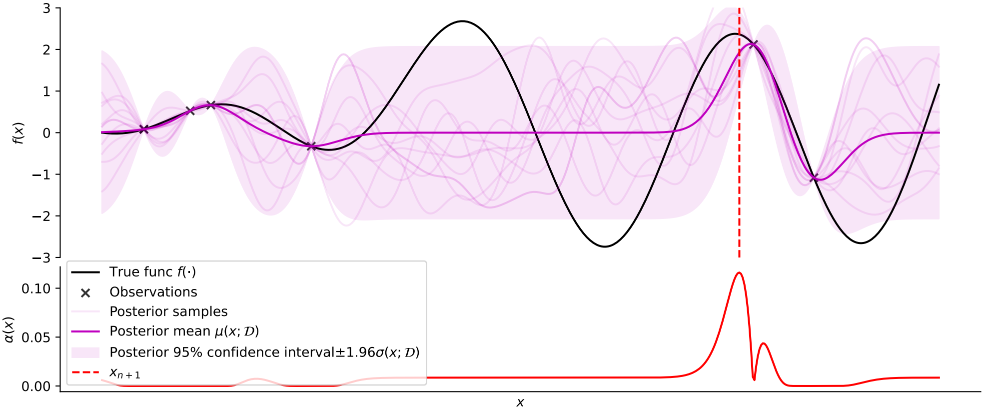
Expected Improvement

- Left with something closed form and easy to compute.

$$\alpha_{EI}(x;\mathcal{D}) = (\mu(x;\mathcal{D}) - f(x^+))\Phi(\frac{\mu(x;\mathcal{D}) - f(x^+)}{\sigma(x;\mathcal{D})}) + \sigma(x;\mathcal{D})\phi(\frac{\mu(x;\mathcal{D}) - f(x^+)}{\sigma(x;\mathcal{D})})$$

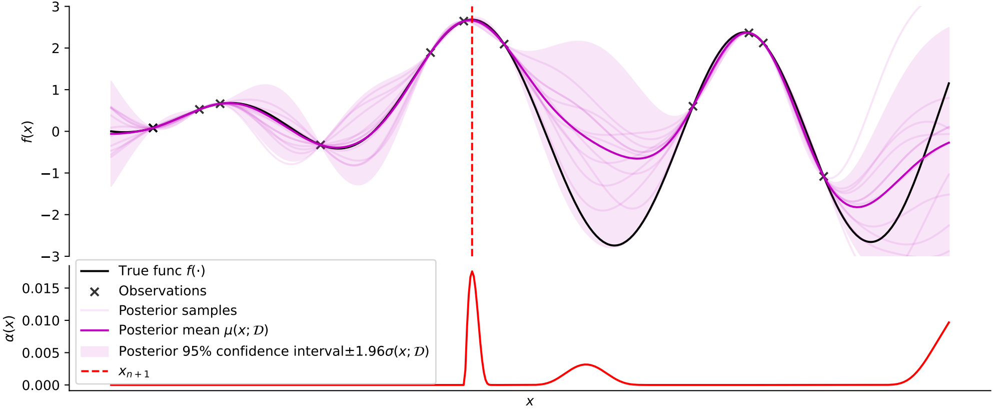
- Φ and ϕ are CDF and PDF of a standard normal distribution.
- How does Expected Improvement change with μ and σ?

$$\alpha_{EI}(x;\mathcal{D}) = (\mu(x;\mathcal{D}) - f(x^+))\Phi(\frac{\mu(x;\mathcal{D}) - f(x^+)}{\sigma(x;\mathcal{D})}) + \sigma(x;\mathcal{D})\phi(\frac{\mu(x;\mathcal{D}) - f(x^+)}{\sigma(x;\mathcal{D})})$$



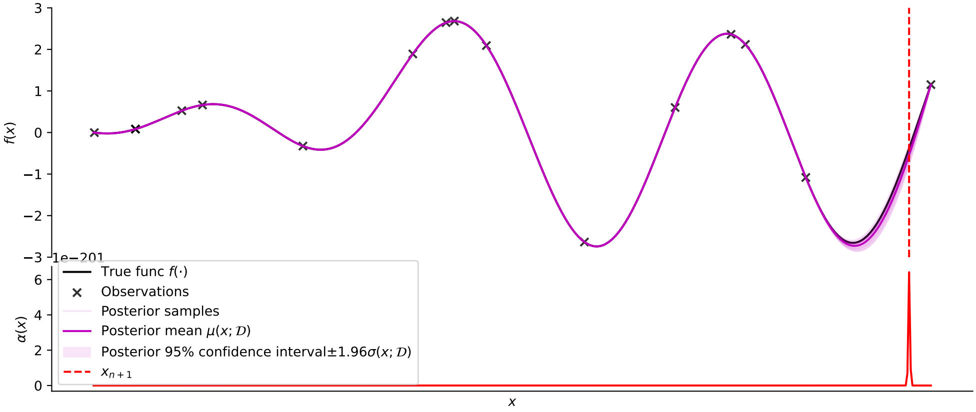
Expected Improvement

$$\alpha_{EI}(x;\mathcal{D}) = (\mu(x;\mathcal{D}) - f(x^+))\Phi(\frac{\mu(x;\mathcal{D}) - f(x^+)}{\sigma(x;\mathcal{D})}) + \sigma(x;\mathcal{D})\phi(\frac{\mu(x;\mathcal{D}) - f(x^+)}{\sigma(x;\mathcal{D})})$$



Expected Improvement

$$\alpha_{EI}(x;\mathcal{D}) = (\mu(x;\mathcal{D}) - f(x^+))\Phi(\frac{\mu(x;\mathcal{D}) - f(x^+)}{\sigma(x;\mathcal{D})}) + \sigma(x;\mathcal{D})\phi(\frac{\mu(x;\mathcal{D}) - f(x^+)}{\sigma(x;\mathcal{D})})$$



Bayesian Optimisation Summary

- EI and PI have simple closed form expressions
- Thompson sampling is more complicated to evaluate
- We covered basic set up but there are many extensions (loop remains the same)
  - e.g. Entropy search, knowledge gradient

Model-based Reinforcement Learning

- **Dynamics**

$$s_{t+1} = f_{\text{env}}(s_t, a_t) + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2 \mathbf{I}) \tag{12}$$

  with states $s \in \mathcal{S}$, actions $a \in \mathcal{A}$ and transition noise ϵ

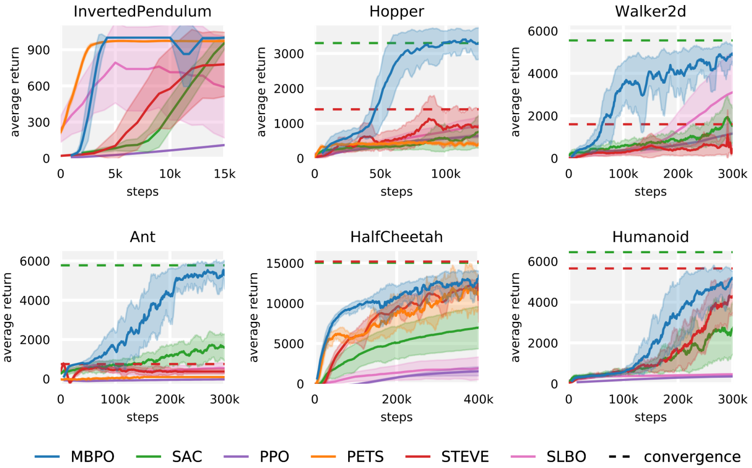
- **Goal**: find policy $\pi \in \Pi$ that maximises expected sum of discounted rewards:

$$\arg\max_{\pi \in \Pi} \mathbb{E}_{\substack{a_t \sim \pi(\cdot | s_t) \\ s_{t+1} \sim p(\cdot | s_t, a_t)}} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right] \tag{13}$$

  with reward function $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, discount factor $\gamma \in [0, 1]$

- Expectation is over transition noise $\epsilon_{0:\infty}$
- We considered myopic Bayesian optimisation, but RL considers:
  - infinite horizon
  - dynamics constraints

Why Model-based Reinforcement Learning



- Model-based RL is more sample efficient
- Used to lack asymptomatic performance but not anymore

Issues in Model-based Reinforcement Learning

- Model bias
  - Overfitting in supervised learning
    - Model performs well on training data but poorly on test data
    - i.e. model overfits to training data
  - Overfitting in model-based RL - known as "model bias"
    - Policy learning exploits model inaccuracies due to lack of training data
    - i.e. policy overfits to inaccurate dynamics model
- Compound error
  - Errors compound when making multi-step predictions
- Objective mismatch
  - Model training is a simple optimization problem disconnected from reward

Gaussian Process Dynamic Model

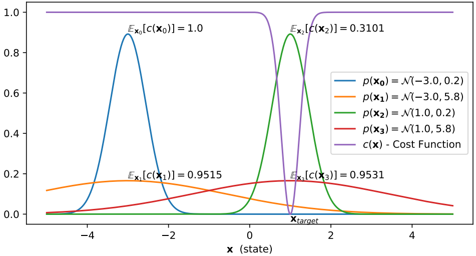
$$p(s_{t+1} \mid s_t, a_t) = \int \underbrace{p(s_{t+1} \mid f(s_t, a_t), \sigma)}_{\text{Gaussian likelihood}} \underbrace{p(f(s_t, a_t) \mid s_t, a_t)}_{\text{GP prior}} \mathrm{d}f(s_t, a_t) \tag{14}$$

- Learn a single-step dynamic model, using GP regression
- How to use *epistemic* uncertainty?

$$p(f(s_t, a_t) \mid s_t, a_t) = \mathcal{N}\left(f(s_t, a_t) \mid \mu_f(s_t, a_t), \Sigma_f^2(s_t, a_t)\right) \tag{15}$$

Greedy Exploitation

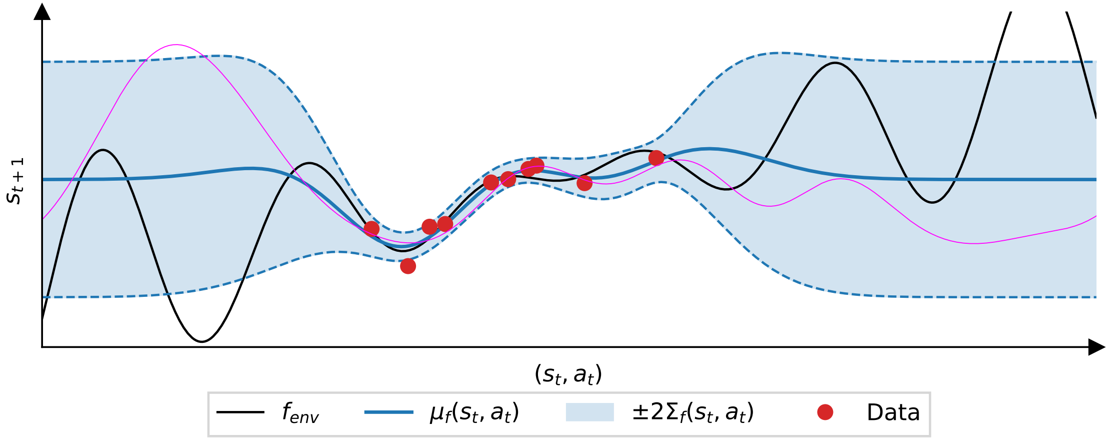
$$\pi_{\text{greedy}} = \arg\max_{\pi \in \Pi} \mathbb{E}_{f \sim p(f \mid \mathcal{D})} \left[J(f, \pi) \right] \qquad J(f, \pi) = \mathbb{E}_{\epsilon_{0:\infty}} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right] \tag{16}$$



- Expectation over posterior combats model bias
- No exploration guarantees

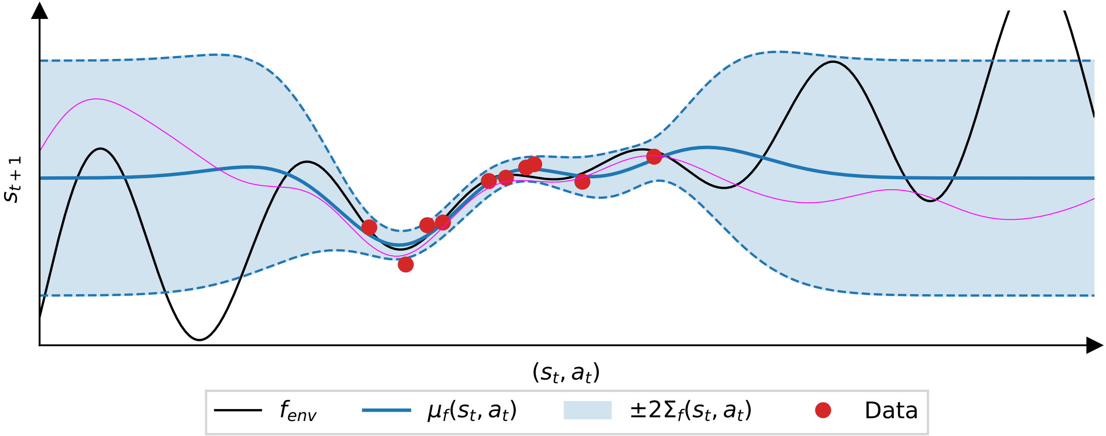
Posterior (Thompson) Sampling

$$\pi_{\mathsf{PS}} = \arg\max_{\pi \in \Pi} \left[J(\hat{f}, \pi) \right] \quad \hat{f} \sim p(f \mid \mathcal{D}) \qquad J(f, \pi) = \mathbb{E}_{\epsilon_{0:\infty}} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right] \tag{17}$$



Posterior (Thompson) Sampling

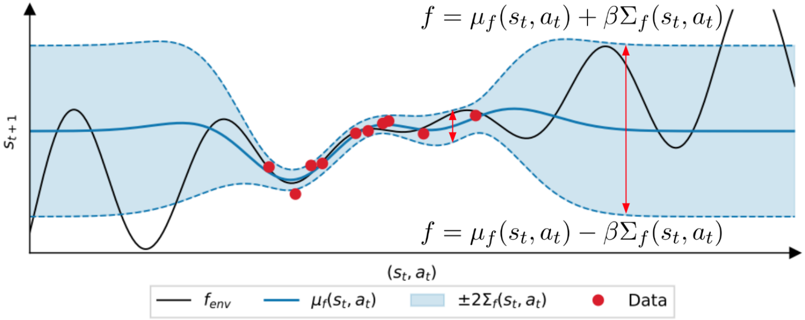
$$\pi_{\mathsf{PS}} = \arg\max_{\pi \in \Pi} \left[J(\hat{f}, \pi) \right] \quad \hat{f} \sim p(f \mid \mathcal{D}) \qquad J(f, \pi) = \mathbb{E}_{\epsilon_{0:\infty}} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right] \tag{17}$$



Upper Confidence Bound

$$\pi_{\text{UCB}} = \arg\max_{\pi \in \Pi} \max_{\hat{f} \in \mathcal{M}} \left[J(\hat{f}, \pi) \right] \quad \mathcal{M} = \{ f \,||\, f(s,a) - \mu_f(s,a)| \leq \beta \Sigma_f(s,a) \} \tag{18}$$

- Optimism in the face of uncertainty
- Inner maximisation hard to compute
- Recent practical implementation for deep model-based RL



Main Takeaways

- Uncertainty quantification is useful for sequential decision making
- There are lots of ways to use uncertainty

Things to check out

- Check out Trieste's Bayesian optimisation notebooks
- PILCO: Probabilistic Inference for Learning cOntrol
  www.youtube.com/watch?v=XiigTGKZfkst=1sab$_channel = PilcoLearner$
- Efficient Model-Based Reinforcement Learning through Optimistic Policy Search and Planning, Curi, Sebastian and Berkenkamp, Felix and Krause, Andreas, Advances in Neural Information Processing Systems 33 (NeurIPS 2020)

What next?

- This was the last lecture
- Last sef of exercises to be published this week
- Course feedback (Webropol) opening soon
  (you should receive a personal link via email)