Sequential Data Acquisition

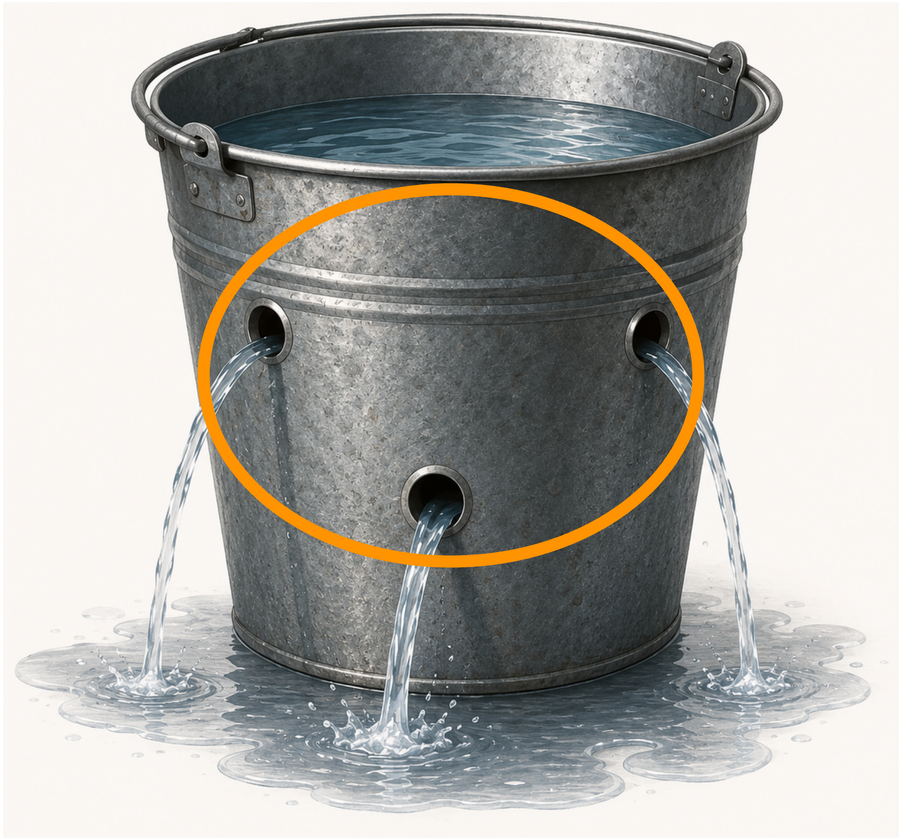
An Introduction

People and Planet Reading Group
20 May 2026

Rui-Yang Zhang

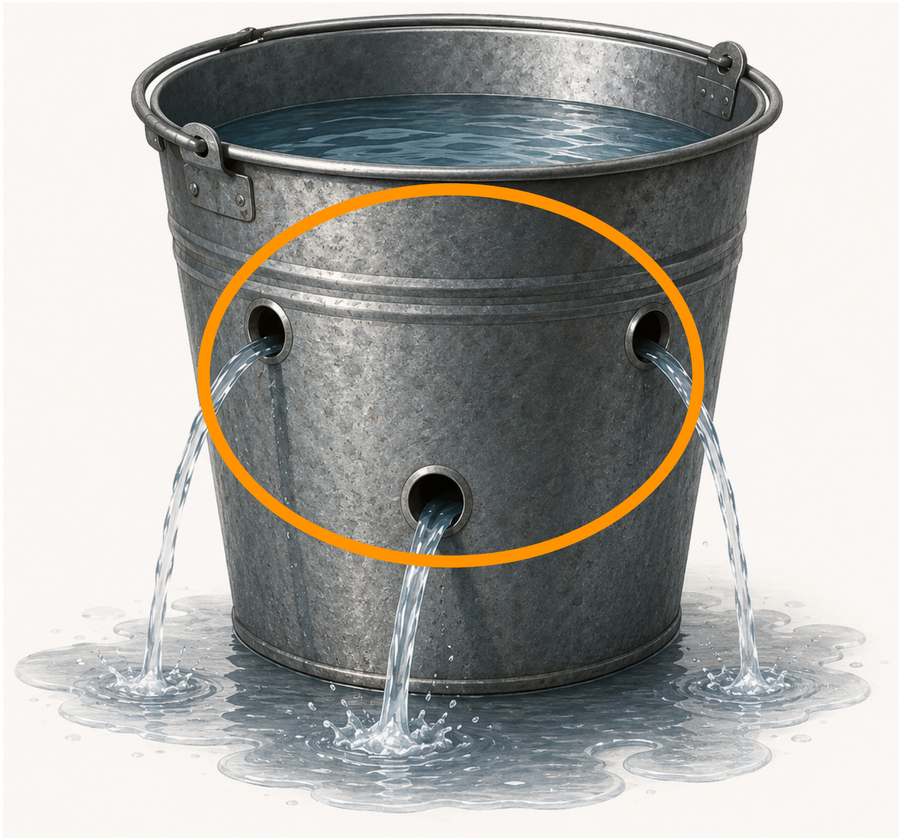
The Bucket of Statistical Inference

The Bucket of Statistical Inference



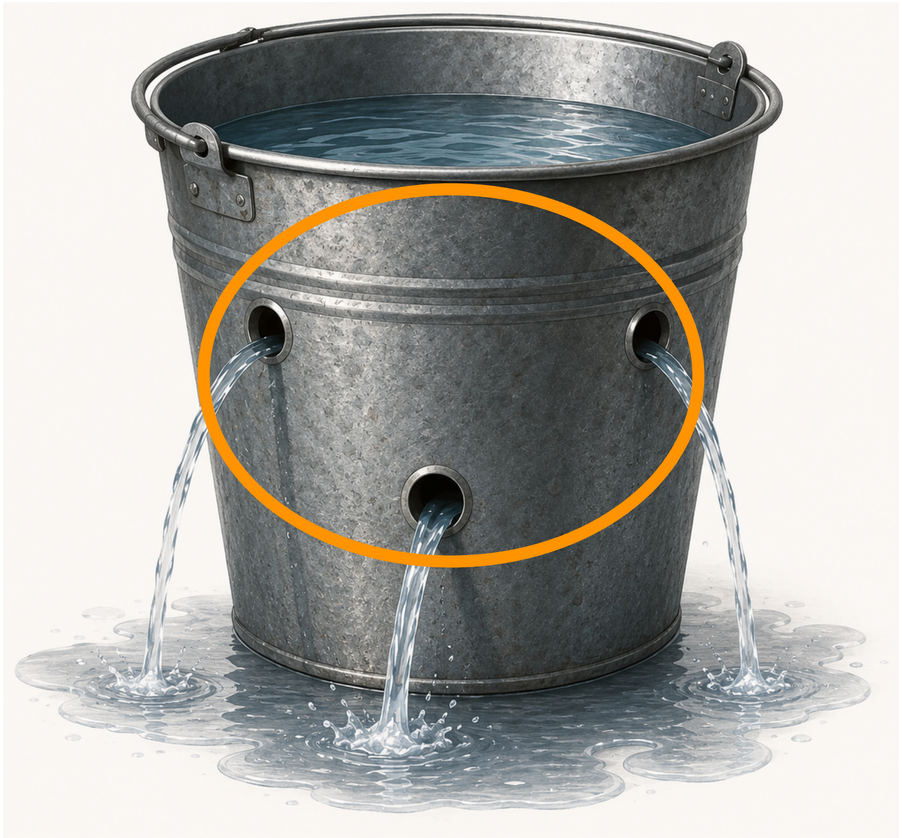
The Bucket of Statistical Inference

Statistical Models



The Bucket of Statistical Inference

Statistical Models

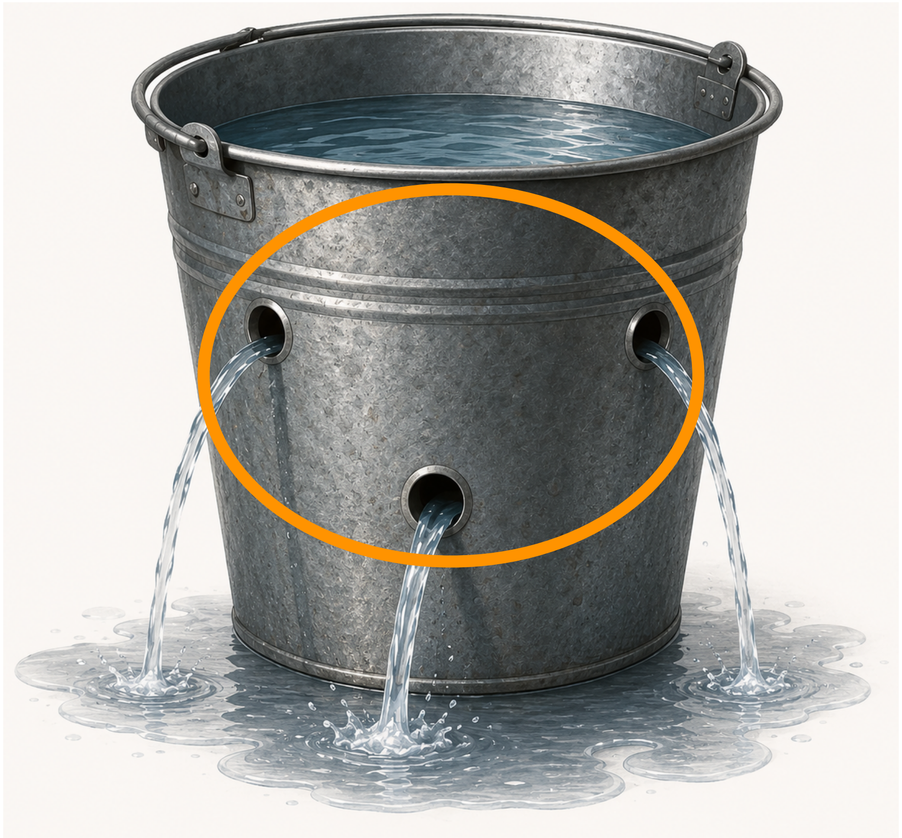


Inference Objectives

The Bucket of Statistical Inference

Statistical Models

Prior Assumptions

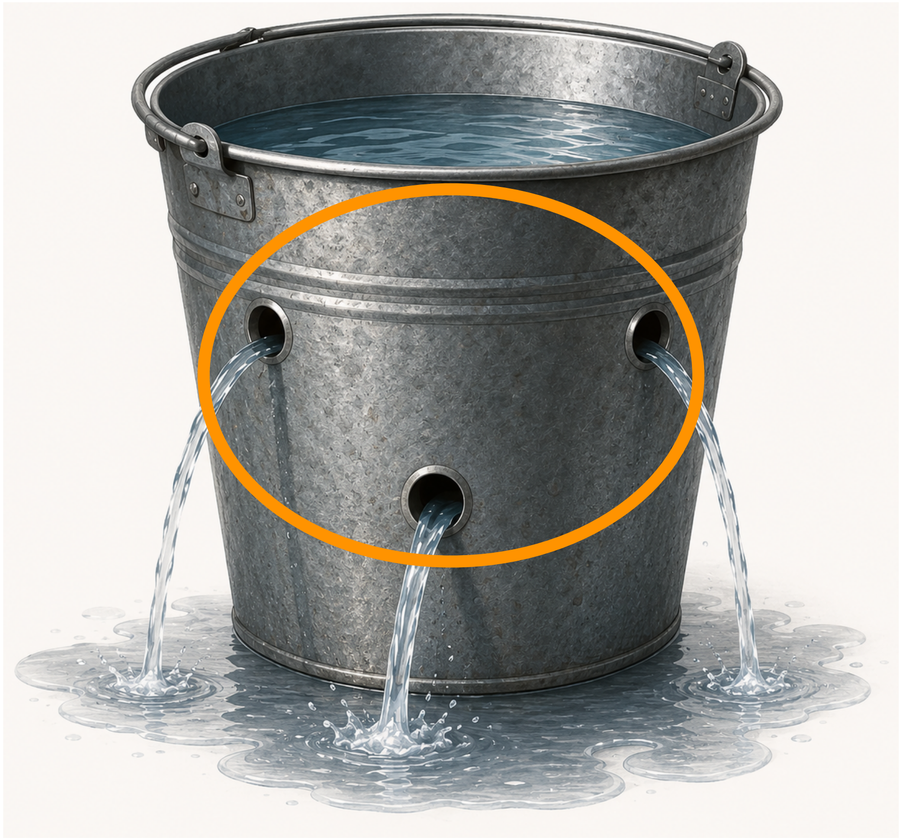


Inference Objectives

The Bucket of Statistical Inference

Statistical Models

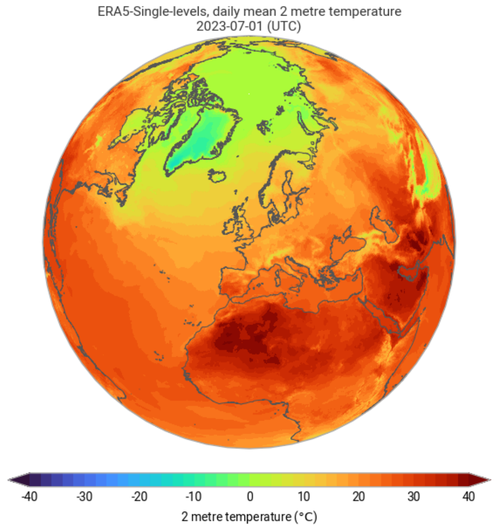
Prior Assumptions



Inference Objectives

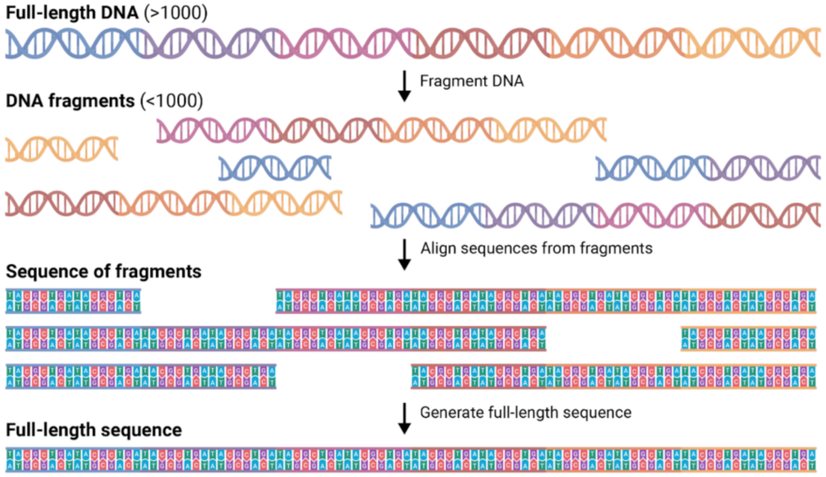
Data

Weather



ERA5
0.25° Pressure and Surface Level
2000TB

Genome



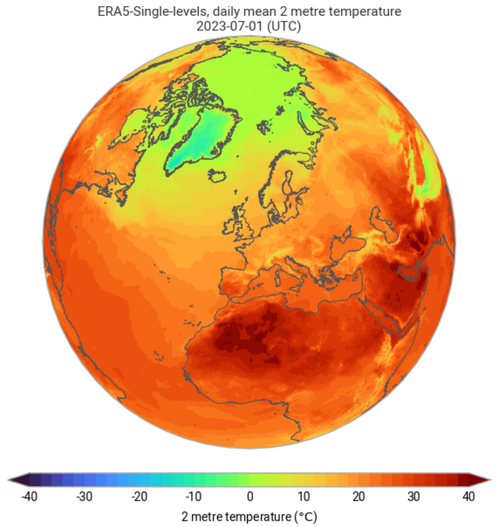
Human Genome Project
DNA Sequences
200GB

Language

LLM Training
The Internet
50TB

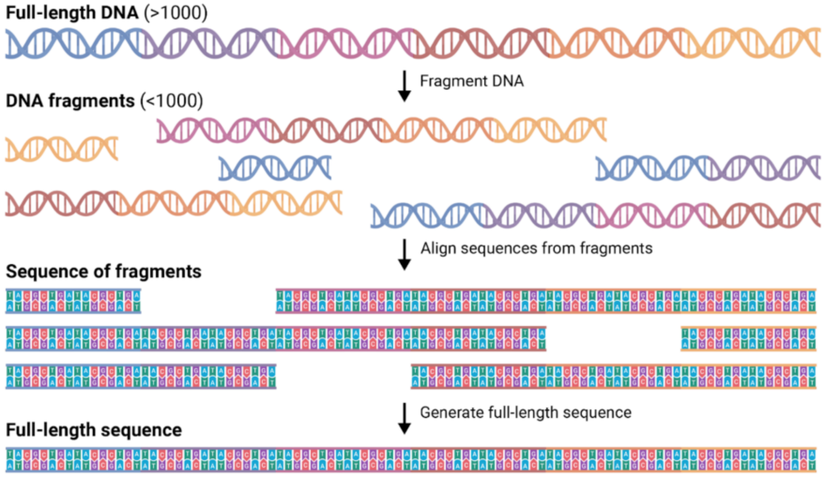
So Much Data!

Weather



ERA5
0.25° Pressure and Surface Level
2000TB

Genome



Human Genome Project
DNA Sequences
200GB

Language

LLM Training
The Internet
50TB

Pediatric Oncology

Tumour Growth

Ocean Engineering

Seabed Structure

Ecology

Species Extinction

So Little Data!

Pediatric Oncology

Tumour Growth

Ocean Engineering

Seabed Structure

Ecology

Species Extinction

Motivation

Motivation

- Quantity of data matters, but quality often matters more!

Motivation

- Quantity of data matters, but quality often matters more!

- Sometimes, data are expensive (time, money, ethics …) to obtain.

Motivation

- Quantity of data matters, but quality often matters more!
- Sometimes, data are expensive (time, money, ethics …) to obtain.
- How do we extract more information using limited budget?

Outline

Outline

- Sequential Data Acquisition Framework

Outline

- Sequential Data Acquisition Framework

- An Introduction to Gaussian Processes

Outline

- Sequential Data Acquisition Framework
- An Introduction to Gaussian Processes
- Common Acquisition Functions

Outline

- Sequential Data Acquisition Framework
- An Introduction to Gaussian Processes
- Common Acquisition Functions
- Selected Topics

Sequential Data Acquisition Framework

Sequential Data Acquisition

Sequential Data Acquisition

- Decision space X (all potential observation locations)

Sequential Data Acquisition

- Decision space X (all potential observation locations)
- Outcome space Y (observed values, e.g. temperature, salinity)

Sequential Data Acquisition

- Decision space X (all potential observation locations)
- Outcome space Y (observed values, e.g. temperature, salinity)
- Utility function $U(y)$ (the information content of an observation)

Sequential Data Acquisition

- Decision space X (all potential observation locations)
- Outcome space Y (observed values, e.g. temperature, salinity)
- Utility function $U(y)$ (the information content of an observation)
- Predictive model $p(y \mid x)$ (observation value forecasts)

Sequential Data Acquisition

- Decision space X (all potential observation locations)
- Outcome space Y (observed values, e.g. temperature, salinity)
- Utility function $U(y)$ (the information content of an observation)
- Predictive model $p(y \mid x)$ (observation value forecasts)
- Existing data $\mathscr{D}_n = \{(x_i, y_i)\}_{i=1}^{n}$ (observed location-value pairs so far)

Sequential Data Acquisition

- Decision space X (all potential observation locations)
- Outcome space Y (observed values, e.g. temperature, salinity)
- Utility function $U(y)$ (the information content of an observation)
- Predictive model $p(y \mid x)$ (observation value forecasts)
- Existing data $\mathscr{D}_n = \{(x_i, y_i)\}_{i=1}^{n}$ (observed location-value pairs so far)

$$x^*_{n+1} := \underset{x \in X}{\operatorname{argmax}} \; \mathbb{E}_{y \sim p(\cdot \mid x, \mathscr{D}_n)}[U(y)]$$

Sequential Data Acquisition

- Decision space X (all potential observation locations)
- Outcome space Y (observed values, e.g. temperature, salinity)
- Utility function $U(y)$ (the information content of an observation)
- Predictive model $p(y \mid x)$ (observation value forecasts)
- Existing data $\mathscr{D}_n = \{(x_i, y_i)\}_{i=1}^{n}$ (observed location-value pairs so far)

$$x^*_{n+1} := \underset{x \in X}{\operatorname{argmax}} \; \mathbb{E}_{y \sim p(\cdot \mid x, \mathscr{D}_n)}[U(y)]$$

acquisition function

Sequential Data Acquisition

Sequential Data Acquisition

- Start with an initial dataset $\mathscr{D}_0$.

Sequential Data Acquisition

- Start with an initial dataset $\mathscr{D}_0$.
- For decision numbers $n = 1, \ldots, N$:

Sequential Data Acquisition

- Start with an initial dataset $\mathscr{D}_0$.
- For decision numbers $n = 1, \ldots, N$:
  - Update predictive model $p(\cdot \mid D_{n-1})$.

Sequential Data Acquisition

- Start with an initial dataset $\mathscr{D}_0$.
- For decision numbers $n = 1, \ldots, N$:
  - Update predictive model $p(\,\cdot\,|\,D_{n-1})$.
  - Compute $x_n^* := \underset{x \in X}{\operatorname{argmax}}\ \mathbb{E}_{y \sim p(\cdot|x, \mathscr{D}_{n-1})}[U(y)]$.

Sequential Data Acquisition

- Start with an initial dataset $\mathscr{D}_0$.
- For decision numbers $n = 1, \ldots, N$:
  - Update predictive model $p(\cdot \mid D_{n-1})$.
  - Compute $x_n^* := \underset{x \in X}{\operatorname{argmax}} \, \mathbb{E}_{y \sim p(\cdot \mid x, \mathscr{D}_{n-1})}[U(y)]$.
  - Make decision and observe $\{(x_n^*, y_n)\}$.

Sequential Data Acquisition

- Start with an initial dataset $\mathscr{D}_0$.
- For decision numbers $n = 1, \ldots, N$:
  - Update predictive model $p(\cdot \mid D_{n-1})$.
  - Compute $x_n^* := \underset{x \in X}{\operatorname{argmax}} \, \mathbb{E}_{y \sim p(\cdot \mid x, \mathscr{D}_{n-1})}[U(y)]$.
  - Make decision and observe $\{(x_n^*, y_n)\}$.
  - Append observation $\mathscr{D}_n = \mathscr{D}_{n-1} \cup \{(x_n^*, y_n)\}$.

Sequential Data Acquisition

- Start with an initial dataset $\mathscr{D}_0$.
- For decision numbers $n = 1, \ldots, N$:
  - Update predictive model $p(\cdot \mid D_{n-1})$.
  - Compute $x_n^* := \underset{x \in X}{\operatorname{argmax}} \, \mathbb{E}_{y \sim p(\cdot \mid x, \mathscr{D}_{n-1})}[U(y)]$.
  - Make decision and observe $\{(x_n^*, y_n)\}$.
  - Append observation $\mathscr{D}_n = \mathscr{D}_{n-1} \cup \{(x_n^*, y_n)\}$.

What if my decision yields a sequence of outcomes?

Sequential Data Acquisition

- Start with an initial dataset $\mathscr{D}_0$.
- For decision numbers $n = 1, \ldots, N$:
  - Update predictive model $p(\cdot \mid D_{n-1})$.
  - Compute $x_n^* := \underset{x \in X}{\operatorname{argmax}} \, \mathbb{E}_{y \sim p(\cdot \mid x, \mathscr{D}_{n-1})}[U(y)]$.
  - Make decision and observe $\{(x_n^*, y_n)\}$.
  - Append observation $\mathscr{D}_n = \mathscr{D}_{n-1} \cup \{(x_n^*, y_n)\}$.

What if my decision yields a sequence of outcomes?

What if I want to make multiple decisions at once?

Sequential Data Acquisition

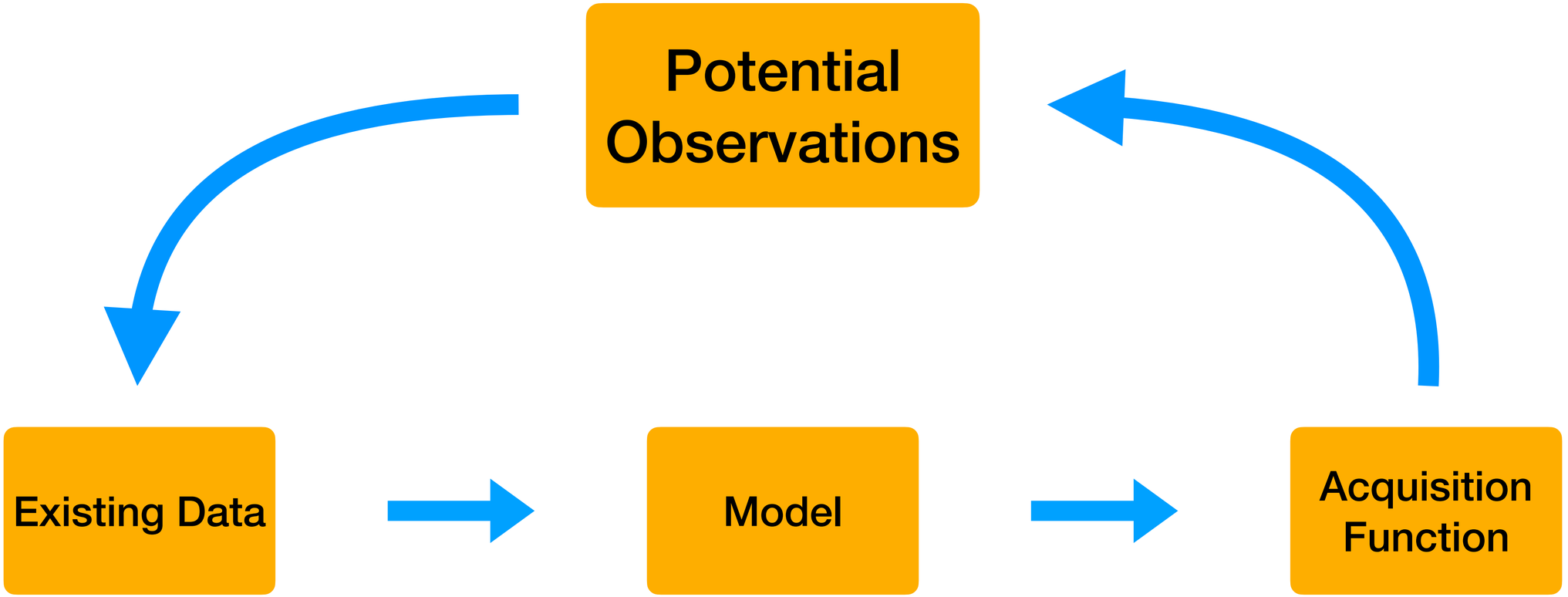
- Start with an initial dataset $\mathscr{D}_0$.
- For decision numbers $n = 1, \ldots, N$:
  - Update predictive model $p(\cdot \mid D_{n-1})$.
  - Compute $x_n^* := \underset{x \in X}{\operatorname{argmax}} \, \mathbb{E}_{y \sim p(\cdot \mid x, \mathscr{D}_{n-1})}[U(y)]$.
  - Make decision and observe $\{(x_n^*, y_n)\}$.
  - Append observation $\mathscr{D}_n = \mathscr{D}_{n-1} \cup \{(x_n^*, y_n)\}$.

What if my decision yields a sequence of outcomes?

What if I want to make multiple decisions at once?

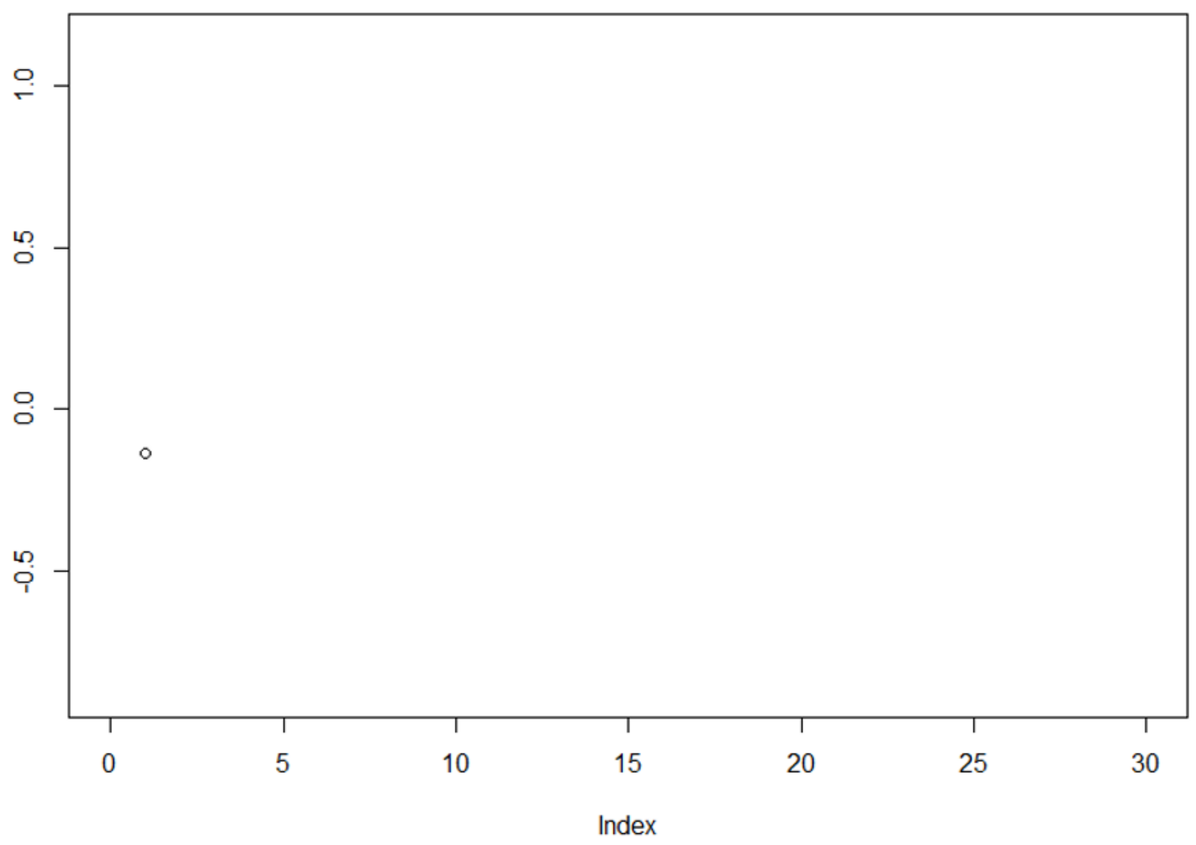
Shouldn't we be (even) more strategic and consider future decisions?

Sequential Data Acquisition



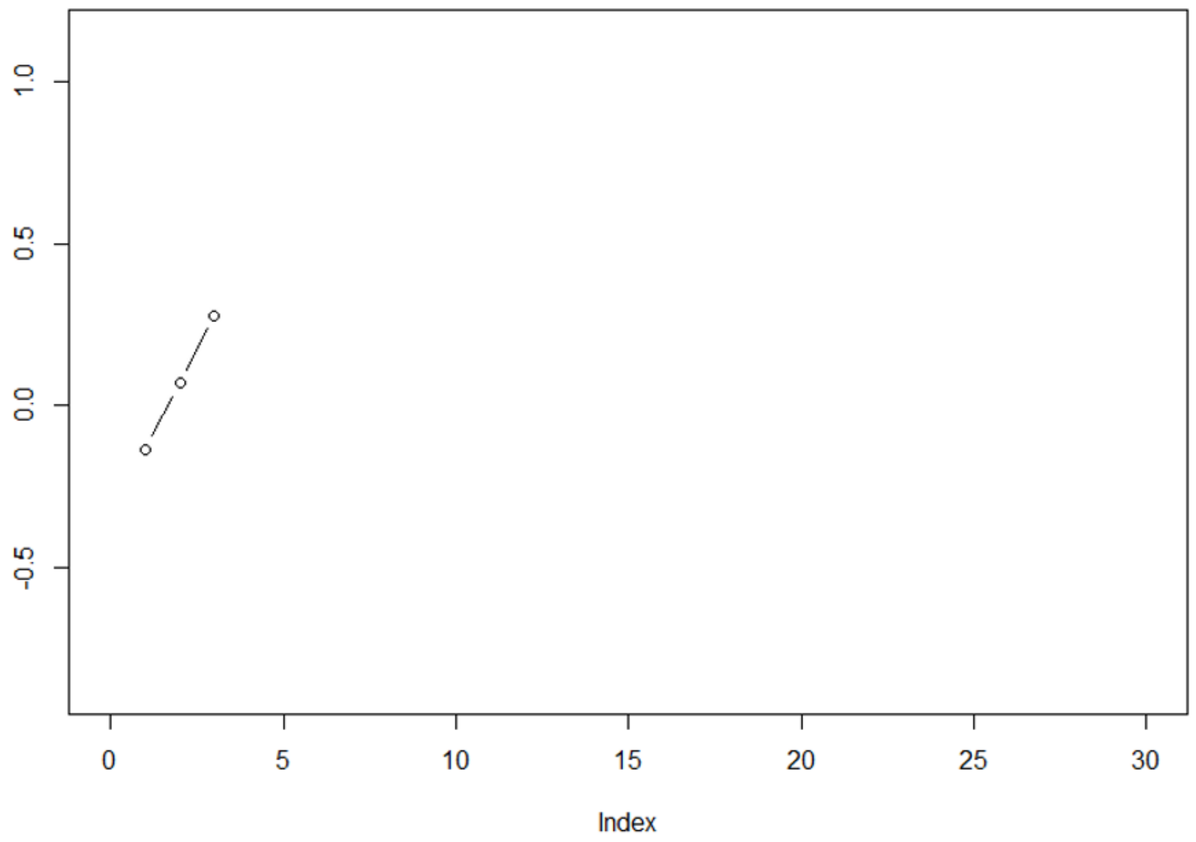
An Introduction to Gaussian Processes

An Introduction to Gaussian Processes



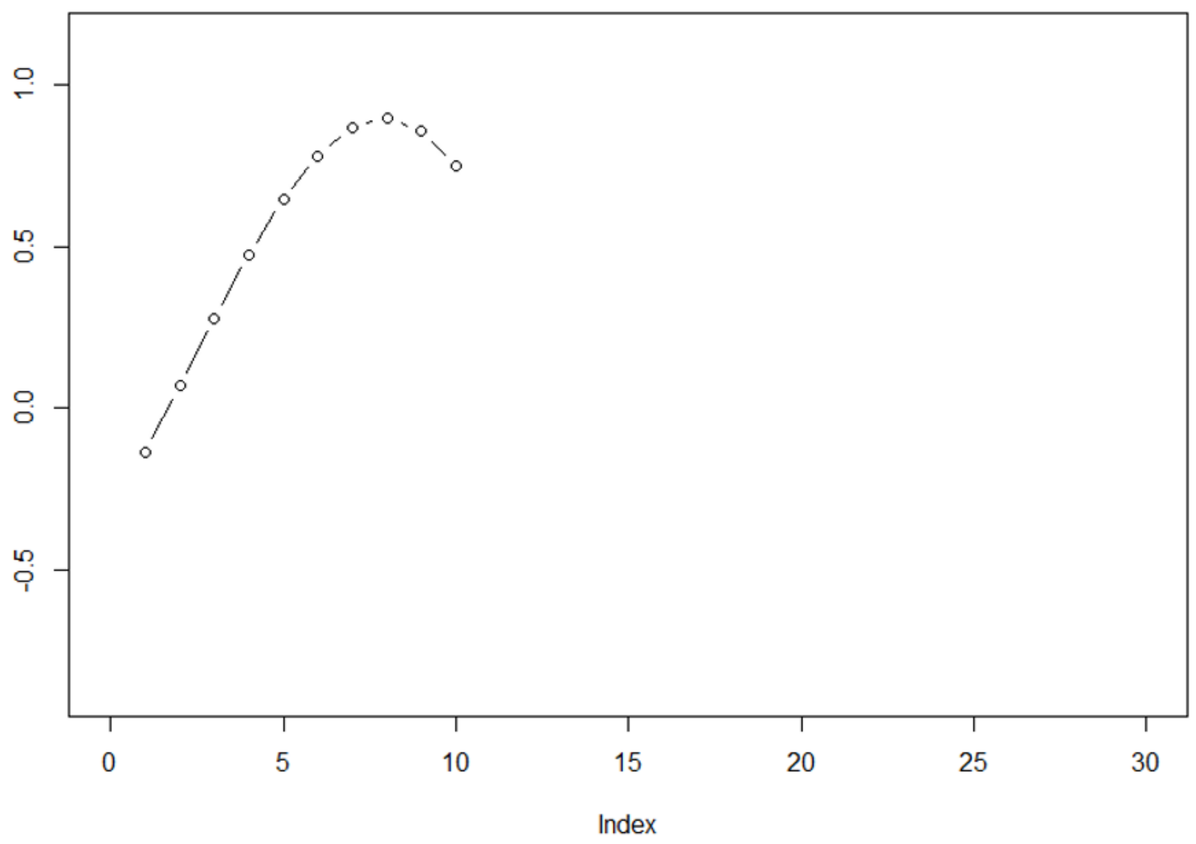
- Multivariate Gaussian
- Centred, Correlated
- Horizontally Displayed

An Introduction to Gaussian Processes



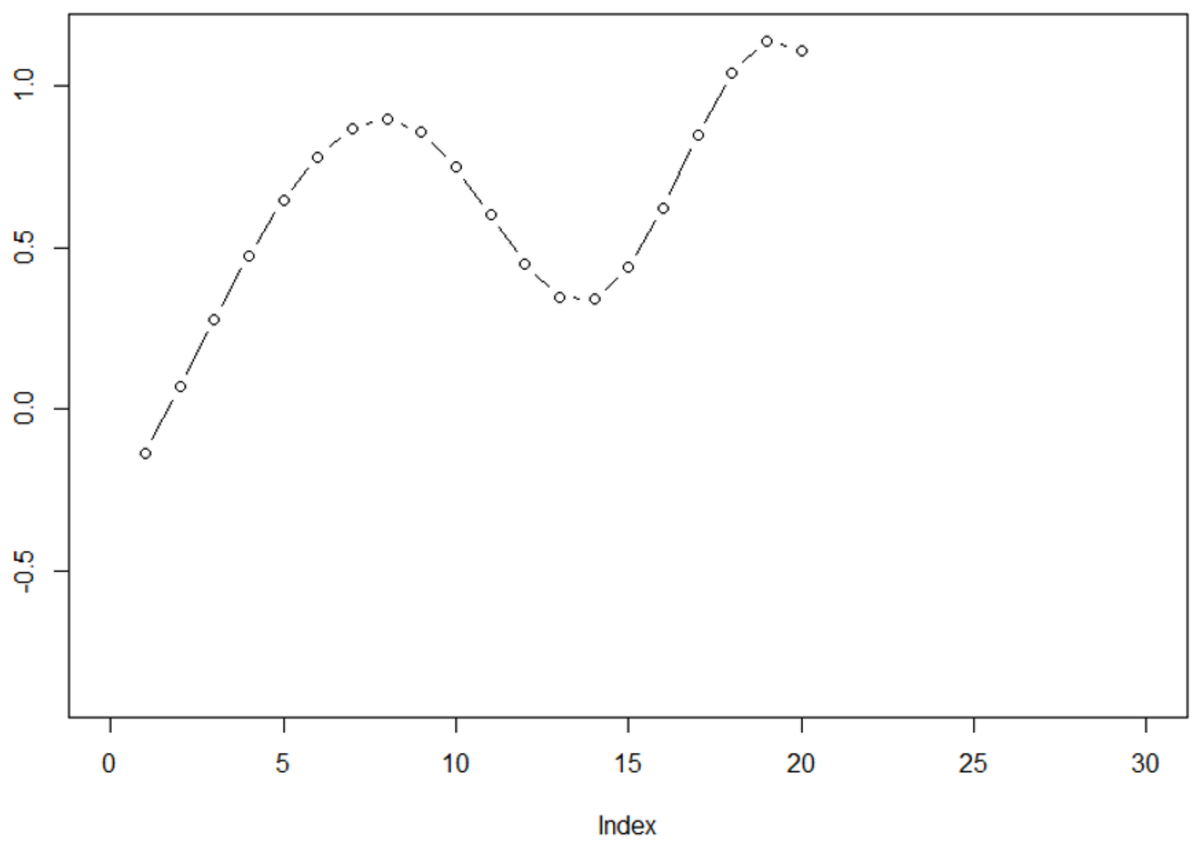
- Multivariate Gaussian
- Centred, Correlated
- Horizontally Displayed

An Introduction to Gaussian Processes



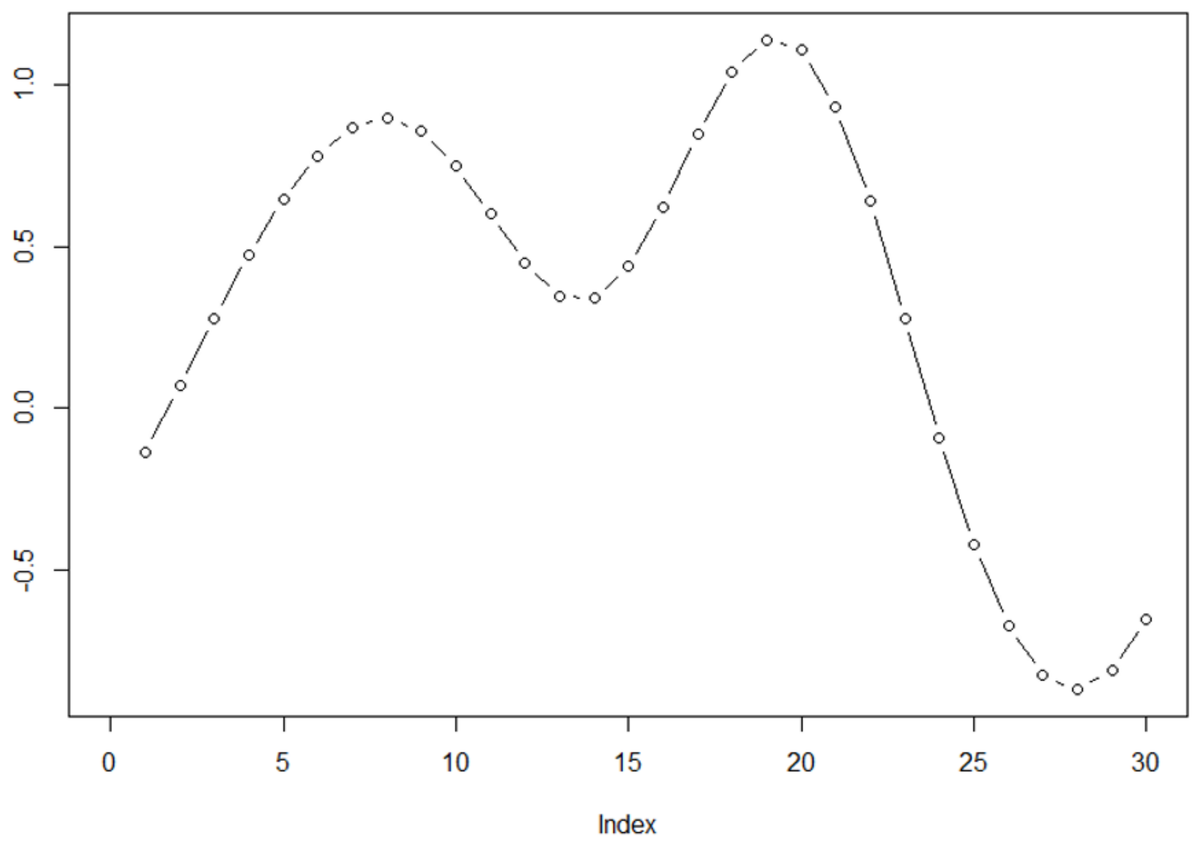
- Multivariate Gaussian
- Centred, Correlated
- Horizontally Displayed

An Introduction to Gaussian Processes



- Multivariate Gaussian
- Centred, Correlated
- Horizontally Displayed

An Introduction to Gaussian Processes



- Multivariate Gaussian
- Centred, Correlated
- Horizontally Displayed

An Introduction to Gaussian Processes

Kernel Choices

Formally, the **Gaussian process** (GP) $y = \{y(x)\}_{x \in \mathcal{X}}$ is a stochastic process indexed by space $x \in \mathcal{X}$ with a mean function $\mu(\cdot)$ and a covariance function $k(\cdot, \cdot)$ such that $\mathbb{E}[y(x)] = \mu(x)$ and $\text{Cov}(y(x), y(x')) = k(x, x')$. We will denote such a process as

$$y(\cdot) \sim GP(\mu(\cdot), k(\cdot, \cdot)).$$

An Introduction to Gaussian Processes

Kernel Choices

Formally, the **Gaussian process** (GP) $y = \{y(x)\}_{x \in \mathcal{X}}$ is a stochastic process indexed by space $x \in \mathcal{X}$ with a mean function $\mu(\cdot)$ and a covariance function $k(\cdot, \cdot)$ such that $\mathbb{E}[y(x)] = \mu(x)$ and $\text{Cov}(y(x), y(x')) = k(x, x')$. We will denote such a process as

set to zero, or considered separately for covariates

$$y(\cdot) \sim GP(\mu(\cdot), k(\cdot, \cdot)).$$

An Introduction to Gaussian Processes

Kernel Choices

Formally, the **Gaussian process** (GP) $y = \{y(x)\}_{x \in \mathcal{X}}$ is a stochastic process indexed by space $x \in \mathcal{X}$ with a mean function $\mu(\cdot)$ and a covariance function $k(\cdot, \cdot)$ such that $\mathbb{E}[y(x)] = \mu(x)$ and $\mathrm{Cov}(y(x), y(x')) = k(x, x')$. We will denote such a process as

set to zero, or considered separately for covariates

key design choice!

$$y(\cdot) \sim GP(\mu(\cdot), k(\cdot, \cdot)).$$

An Introduction to Gaussian Processes

Kernel Choices

Formally, the **Gaussian process** (GP) $y = \{y(x)\}_{x \in \mathcal{X}}$ is a stochastic process indexed by space $x \in \mathcal{X}$ with a mean function $\mu(\cdot)$ and a covariance function $k(\cdot, \cdot)$ such that $\mathbb{E}[y(x)] = \mu(x)$ and $\mathrm{Cov}(y(x), y(x')) = k(x, x')$. We will denote such a process as

set to zero, or considered separately for covariates

key design choice!

$$y(\cdot) \sim GP(\mu(\cdot), k(\cdot, \cdot)).$$

- Any kernel k would work …

An Introduction to Gaussian Processes

Kernel Choices

Formally, the **Gaussian process** (GP) $y = \{y(x)\}_{x \in \mathcal{X}}$ is a stochastic process indexed by space $x \in \mathcal{X}$ with a mean function $\mu(\cdot)$ and a covariance function $k(\cdot, \cdot)$ such that $\mathbb{E}[y(x)] = \mu(x)$ and $\mathrm{Cov}(y(x), y(x')) = k(x, x')$. We will denote such a process as

set to zero, or considered separately for covariates

key design choice!

$$y(\cdot) \sim GP(\mu(\cdot), k(\cdot, \cdot)).$$

- Any kernel k would work …
- as long as the induced GP is always marginally multivariate Gaussian (i.e. with positive semi-definite covariance matrix).

An Introduction to Gaussian Processes

Kernel Choices

Formally, the **Gaussian process** (GP) $y = \{y(x)\}_{x \in \mathcal{X}}$ is a stochastic process indexed by space $x \in \mathcal{X}$ with a mean function $\mu(\cdot)$ and a covariance function $k(\cdot, \cdot)$ such that $\mathbb{E}[y(x)] = \mu(x)$ and $\mathrm{Cov}(y(x), y(x')) = k(x, x')$. We will denote such a process as

set to zero, or considered separately for covariates

key design choice!

$$y(\cdot) \sim GP(\mu(\cdot), k(\cdot, \cdot)).$$

- Any kernel k would work …
- as long as the induced GP is always marginally multivariate Gaussian (i.e. with positive semi-definite covariance matrix).
- Thus k must be a positive semi-definite function.

An Introduction to Gaussian Processes

Kernel Choices

Formally, the **Gaussian process** (GP) $y = \{y(x)\}_{x \in \mathcal{X}}$ is a stochastic process indexed by space $x \in \mathcal{X}$ with a mean function $\mu(\cdot)$ and a covariance function $k(\cdot, \cdot)$ such that $\mathbb{E}[y(x)] = \mu(x)$ and $\mathrm{Cov}(y(x), y(x')) = k(x, x')$. We will denote such a process as

set to zero, or considered separately for covariates

key design choice!

$$y(\cdot) \sim GP(\mu(\cdot), k(\cdot, \cdot)).$$

- Any kernel k would work …
- as long as the induced GP is always marginally multivariate Gaussian (i.e. with positive semi-definite covariance matrix).
- Thus k must be a positive semi-definite function.
- One easy way to check is via Bochner's theorem.

An Introduction to Gaussian Processes

Kernel Choices

Formally, the **Gaussian process** (GP) $y = \{y(x)\}_{x \in \mathcal{X}}$ is a stochastic process indexed by space $x \in \mathcal{X}$ with a mean function $\mu(\cdot)$ and a covariance function $k(\cdot, \cdot)$ such that $\mathbb{E}[y(x)] = \mu(x)$ and $\mathrm{Cov}(y(x), y(x')) = k(x, x')$. We will denote such a process as

set to zero, or considered separately for covariates

key design choice!

$$y(\cdot) \sim GP(\mu(\cdot), k(\cdot, \cdot)).$$

- Any kernel k would work …
- as long as the induced GP is always marginally multivariate Gaussian (i.e. with positive semi-definite covariance matrix).
- Thus k must be a positive semi-definite function.
- One easy way to check is via Bochner's theorem.
- Bochner: PSD of (Stationary) Kernels = Non-Negativity of Spectral Density.

An Introduction to Gaussian Processes

Kernel Choices

Formally, the **Gaussian process** (GP) $y = \{y(x)\}_{x \in \mathcal{X}}$ is a stochastic process indexed by space $x \in \mathcal{X}$ with a mean function $\mu(\cdot)$ and a covariance function $k(\cdot, \cdot)$ such that $\mathbb{E}[y(x)] = \mu(x)$ and $\text{Cov}(y(x), y(x')) = k(x, x')$. We will denote such a process as

$$y(\cdot) \sim GP(\mu(\cdot), k(\cdot, \cdot)).$$

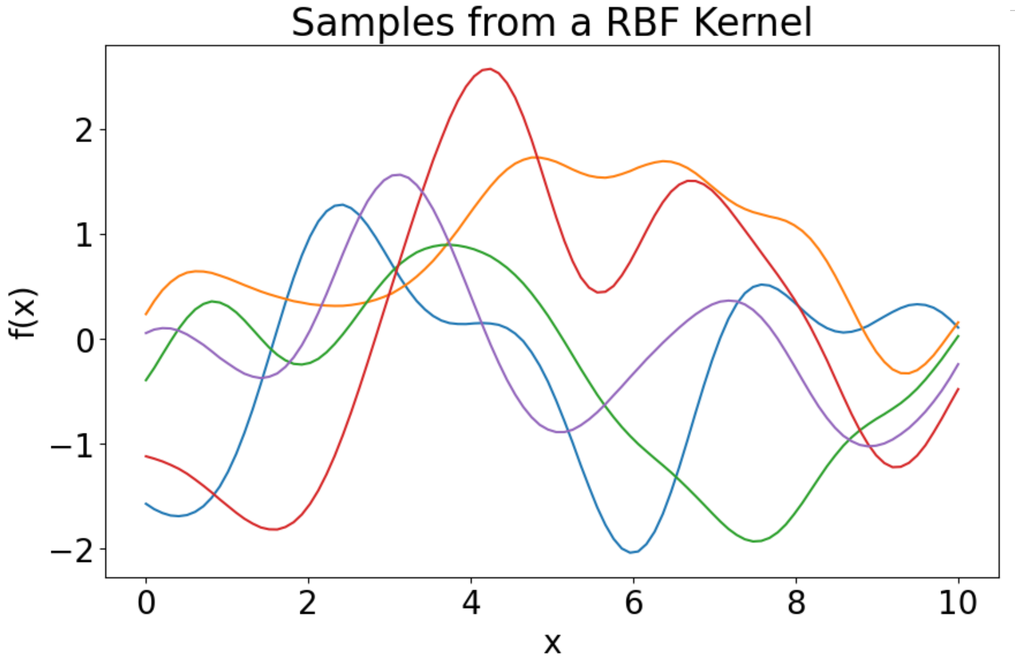
An Introduction to Gaussian Processes

Kernel Choices

Formally, the **Gaussian process** (GP) $y = \{y(x)\}_{x \in \mathcal{X}}$ is a stochastic process indexed by space $x \in \mathcal{X}$ with a mean function $\mu(\cdot)$ and a covariance function $k(\cdot, \cdot)$ such that $\mathbb{E}[y(x)] = \mu(x)$ and $\mathrm{Cov}(y(x), y(x')) = k(x, x')$. We will denote such a process as

$$y(\cdot) \sim GP(\mu(\cdot), k(\cdot, \cdot)).$$

$$k_{RBF}(x, x') = \exp\left[-\frac{\|x - x'\|^2}{2l^2}\right]$$



An Introduction to Gaussian Processes

Kernel Choices

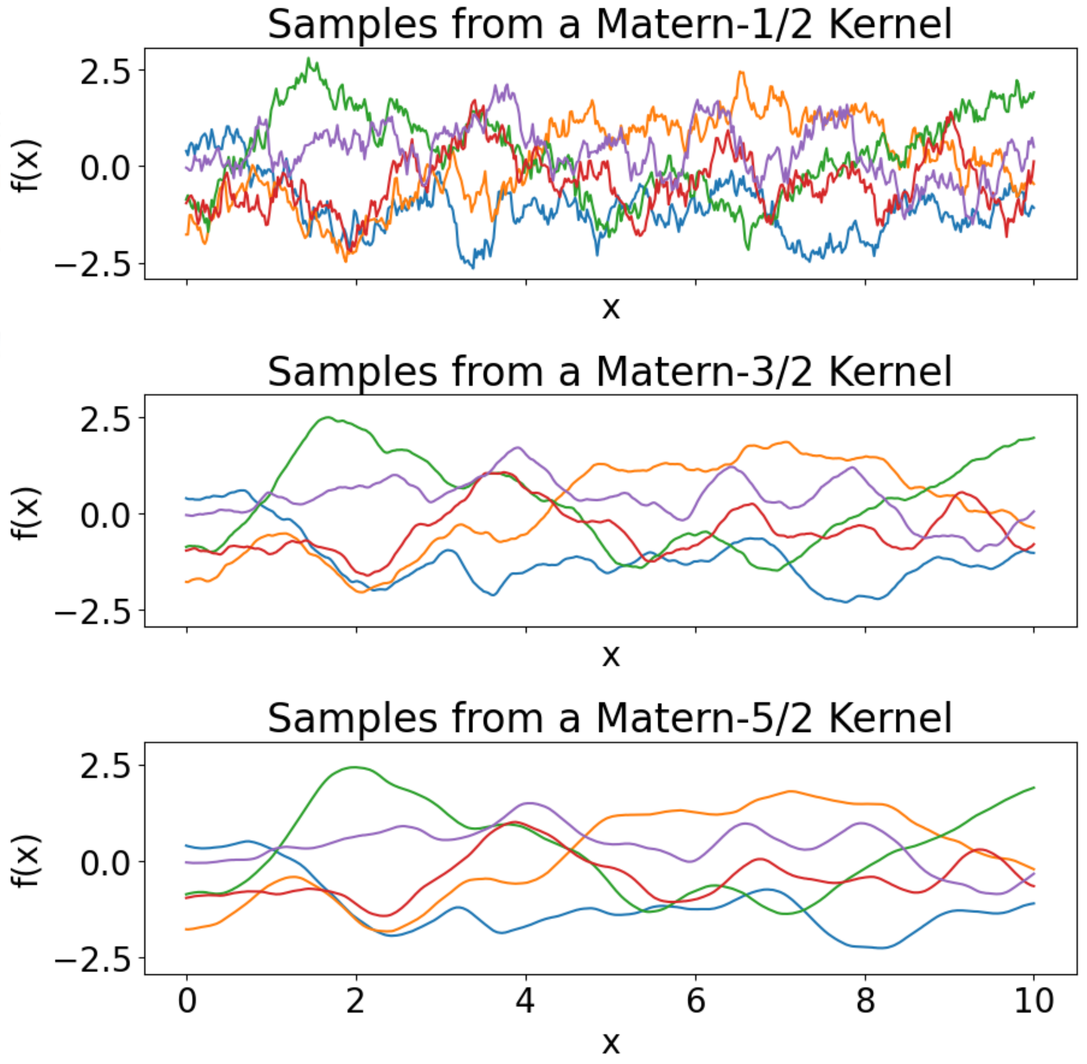
Formally, the **Gaussian process** (GP) $y = \{y(x)\}_{x\in$
$x \in \mathcal{X}$ with a mean function $\mu(\cdot)$ and a covariance fu
$\mathrm{Cov}(y(x), y(x')) = k(x, x')$. We will denote such a prc

$$y(\cdot) \sim GP(\mu(\cdot), k$$

$$k_{Mat}^{\nu=1/2}(x, x') = \exp\left[-\frac{\|x - x'\|^2}{l}\right]$$

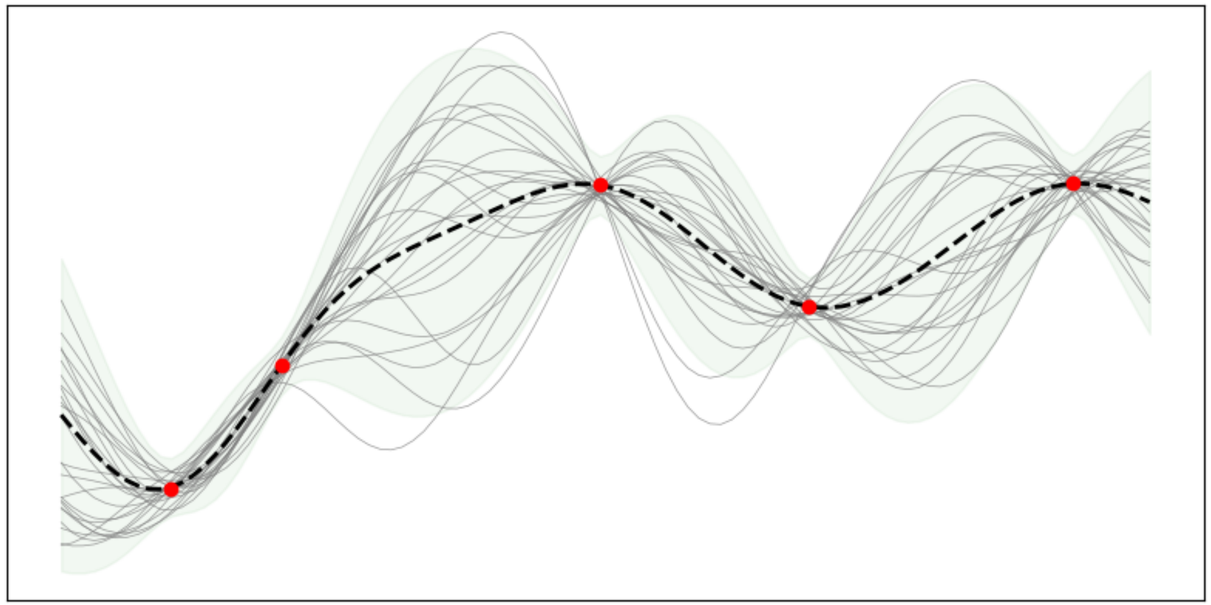
$$k_{Mat}^{\nu=3/2}(x, x') = \left[1 + \frac{\sqrt{3}\|x - x'\|}{l}\right] \exp\left[-\frac{\sqrt{3}\|x - x'\|^2}{l}\right]$$

$$k_{Mat}^{\nu=5/2}(x, x') = \left[1 + \frac{\sqrt{5}\|x - x'\|}{l} + \frac{5\|x - x'\|^2}{3l^2}\right] \exp\left[-\frac{\sqrt{5}\|x - x'\|^2}{l}\right]$$



An Introduction to Gaussian Processes

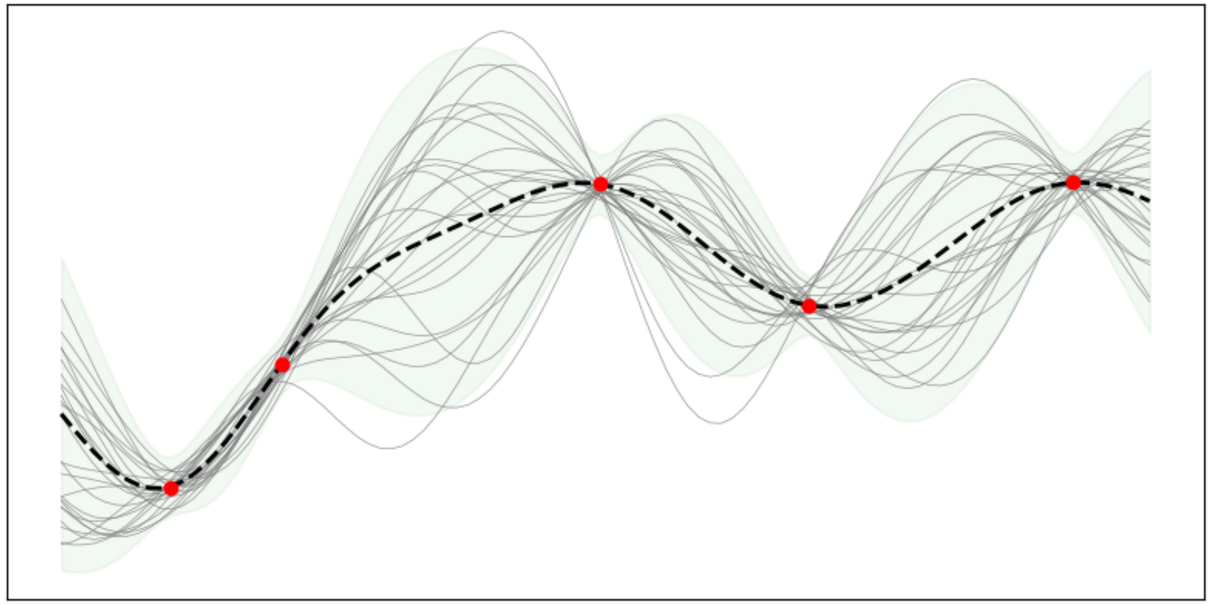
Regression



An Introduction to Gaussian Processes

Regression

Conditioning on n observations $\mathscr{D} = \{X, y\}$ with $X \in \mathbb{R}^{n \times n}$, $y \in \mathbb{R}^{y}$, predicting on m test points $x_* \in \mathbb{R}^{m}$.

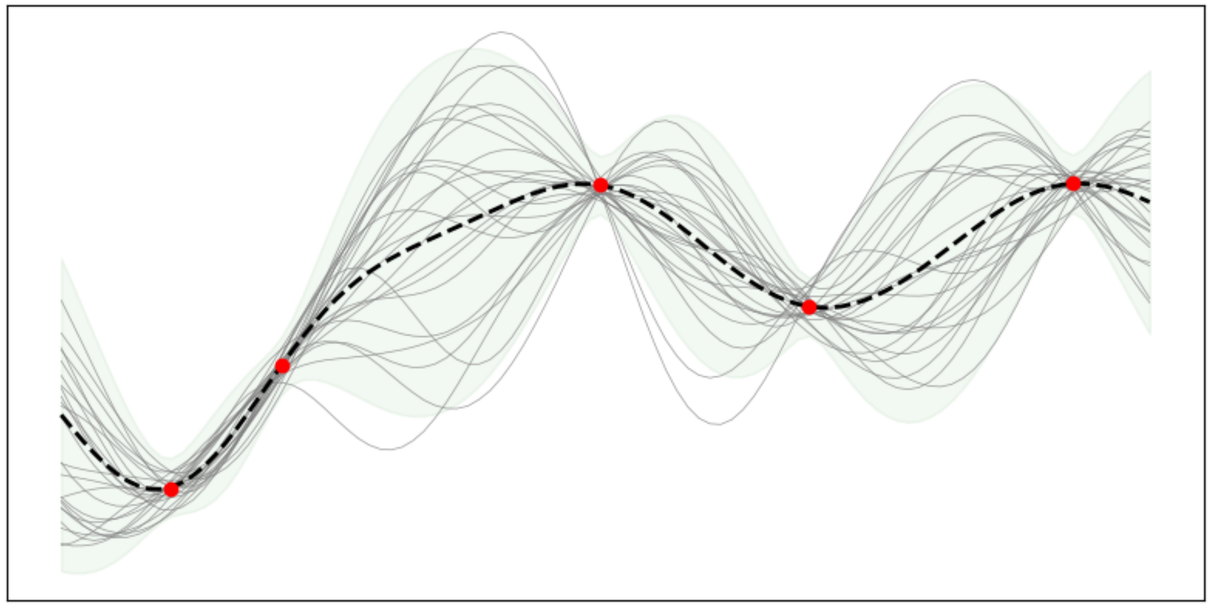


An Introduction to Gaussian Processes

Regression

Conditioning on n observations $\mathcal{D} = \{X, y\}$ with $X \in \mathbb{R}^{n \times n}$, $y \in \mathbb{R}^{y}$, predicting on m test points $x_* \in \mathbb{R}^{m}$.

Denote Gram matrices $K = k(X, X) \in \mathbb{R}^{n \times n}$, $K_* = k(X, x_*) \in \mathbb{R}^{n \times m}$, $K_{**} = k(x_*, x_*) \in \mathbb{R}^{m \times m}$.

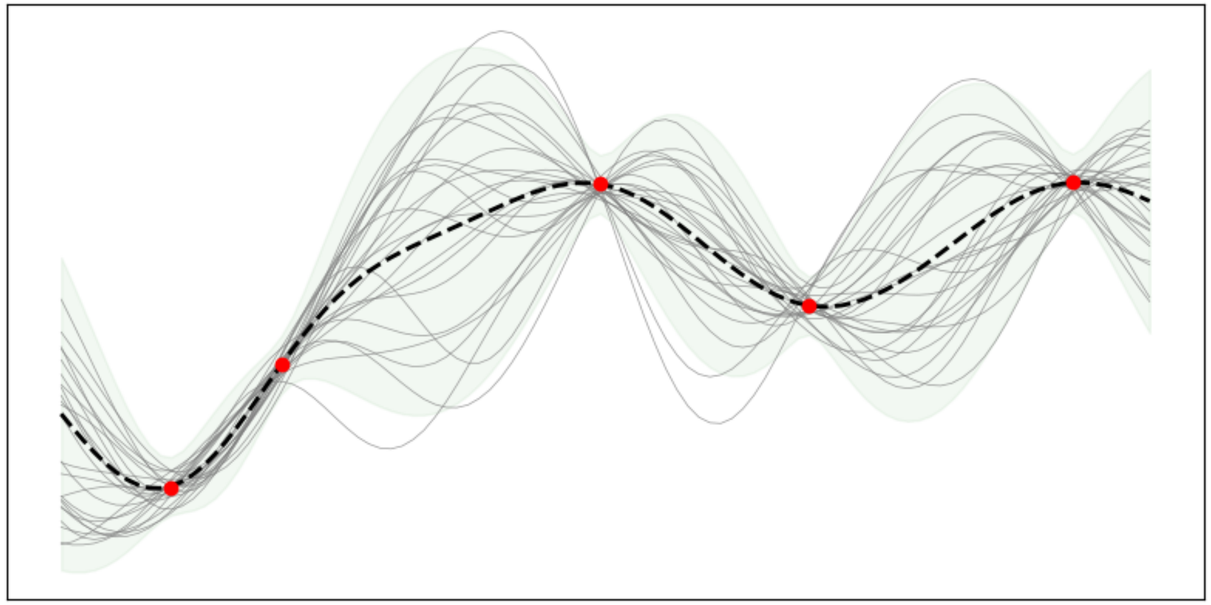


An Introduction to Gaussian Processes

Regression

Conditioning on n observations $\mathcal{D} = \{X, y\}$ with $X \in \mathbb{R}^{n \times n}$, $y \in \mathbb{R}^{y}$, predicting on m test points $x_* \in \mathbb{R}^{m}$.

Denote Gram matrices $K = k(X, X) \in \mathbb{R}^{n \times n}$, $K_* = k(X, x_*) \in \mathbb{R}^{n \times m}$, $K_{**} = k(x_*, x_*) \in \mathbb{R}^{m \times m}$.



$$y_* \,|\, x_*, \mathcal{D}, f \sim N(\mu_{y_*|\mathcal{D}}, K_{y_*|\mathcal{D}})$$

$$\mu_{y_*|\mathcal{D}} = \mu(X_*) + K_*^T (K + \sigma^2 I)^{-1} (y - \mu(X))$$

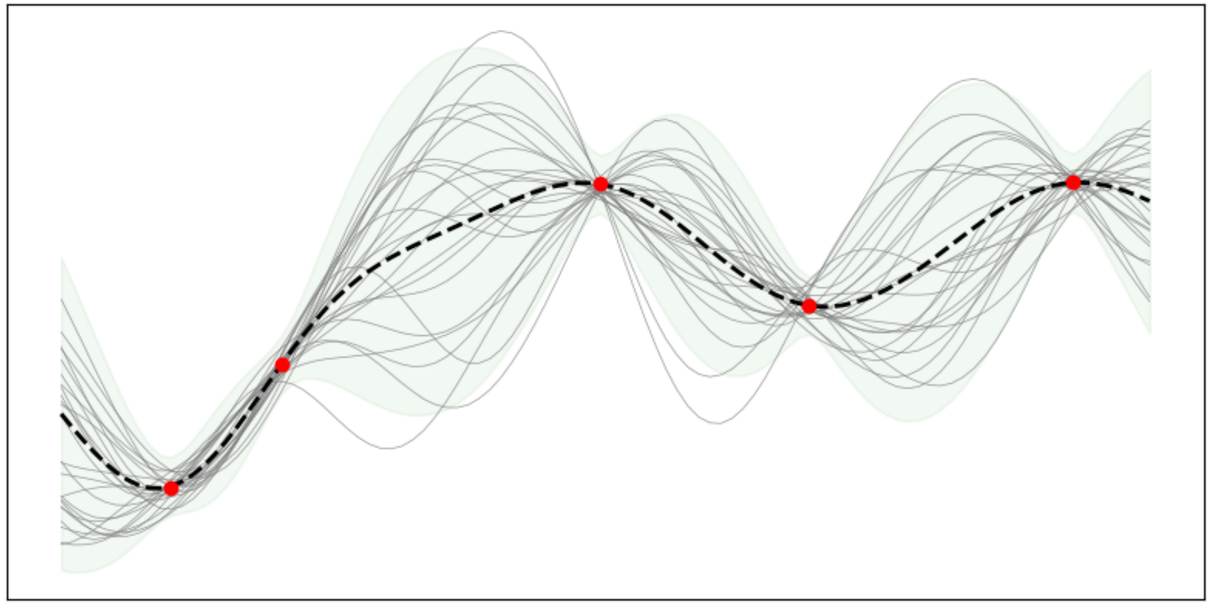
$$K_{y_*|\mathcal{D}} = K_{**} - K_*^T (K + \sigma^2 I)^{-1} K_*$$

An Introduction to Gaussian Processes

Regression

Conditioning on n observations $\mathcal{D} = \{X, y\}$ with $X \in \mathbb{R}^{n \times n}$, $y \in \mathbb{R}^{y}$, predicting on m test points $x_* \in \mathbb{R}^{m}$.

Denote Gram matrices $K = k(X, X) \in \mathbb{R}^{n \times n}$, $K_* = k(X, x_*) \in \mathbb{R}^{n \times m}$, $K_{**} = k(x_*, x_*) \in \mathbb{R}^{m \times m}$.



$$y_* \,|\, x_*, \mathcal{D}, f \sim N(\mu_{y_*|\mathcal{D}}, K_{y_*|\mathcal{D}})$$

$$\mu_{y_*|\mathcal{D}} = \mu(X_*) + K_*^T (K + \sigma^2 I)^{-1} (y - \mu(X))$$

$$K_{y_*|\mathcal{D}} = K_{**} - K_*^T (K + \sigma^2 I)^{-1} K_*$$

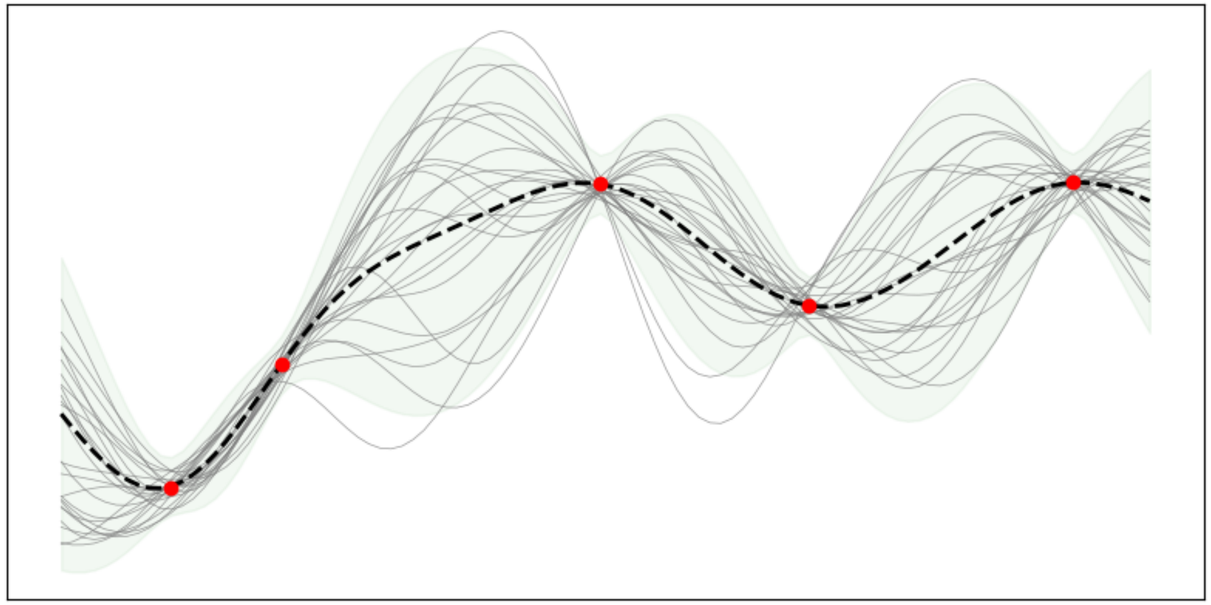
Computational Cost: $O(n^3)$!

An Introduction to Gaussian Processes

Regression

Conditioning on n observations $\mathcal{D} = \{X, y\}$ with $X \in \mathbb{R}^{n \times n}$, $y \in \mathbb{R}^{y}$, predicting on m test points $x_* \in \mathbb{R}^{m}$.

Denote Gram matrices $K = k(X, X) \in \mathbb{R}^{n \times n}$, $K_* = k(X, x_*) \in \mathbb{R}^{n \times m}$, $K_{**} = k(x_*, x_*) \in \mathbb{R}^{m \times m}$.



$$y_* \,|\, x_*, \mathcal{D}, f \sim N(\mu_{y_*|\mathcal{D}}, K_{y_*|\mathcal{D}})$$

$$\mu_{y_*|\mathcal{D}} = \mu(X_*) + K_*^T (K + \sigma^2 I)^{-1} (y - \mu(X))$$

$$K_{y_*|\mathcal{D}} = K_{**} - K_*^T (K + \sigma^2 I)^{-1} K_*$$

Computational Cost: $O(n^3)$!

Consequence of a *static* perspective …

An Introduction to Gaussian Processes

Scalability (Non-Exhaustive List)

Method	Ideas	REF

An Introduction to Gaussian Processes

Scalability (Non-Exhaustive List)

Method	Ideas	REF
Inducing Points	Select a small collection of observations (not necessarily a subset of data) to represent the full dataset.	Felix et al. (2020)

An Introduction to Gaussian Processes

Scalability (Non-Exhaustive List)

Method	Ideas	REF
Inducing Points	Select a small collection of observations (not necessarily a subset of data) to represent the full dataset.	Felix et al. (2020)
Vecchia Approximation	Rewrite the full likelihood into a product of conditionals then drop dependencies from “far away” (according to an ordering) observations.	Katzfuss & Guinness (2021)

An Introduction to Gaussian Processes

Scalability (Non-Exhaustive List)

Method	Ideas	REF
Inducing Points	Select a small collection of observations (not necessarily a subset of data) to represent the full dataset.	Felix et al. (2020)
Vecchia Approximation	Rewrite the full likelihood into a product of conditionals then drop dependencies from “far away” (according to an ordering) observations.	Katzfuss & Guinness (2021)
Random Fourier Feature	Approximate Gram matrices with low-rank approximations with randomly selected Fourier bases.	Rahimi & Recht (2007)

An Introduction to Gaussian Processes

Scalability (Non-Exhaustive List)

Method	Ideas	REF
Inducing Points	Select a small collection of observations (not necessarily a subset of data) to represent the full dataset.	Felix et al. (2020)
Vecchia Approximation	Rewrite the full likelihood into a product of conditionals then drop dependencies from “far away” (according to an ordering) observations.	Katzfuss & Guinness (2021)
Random Fourier Feature	Approximate Gram matrices with low-rank approximations with randomly selected Fourier bases.	Rahimi & Recht (2007)
SPDE-GP (Sarkka et al’s Version)	Reformulate the GP as the stationary solution to an S(P)DE and thus convert the static regression to dynamic sequential inference.	Särkkä, Solin & Hartikainen (2013)

An Introduction to Gaussian Processes

Scalability (Non-Exhaustive List)

Method	Ideas	REF
Inducing Points	Select a small collection of observations (not necessarily a subset of data) to represent the full dataset.	Felix et al. (2020)
Vecchia Approximation	Rewrite the full likelihood into a product of conditionals then drop dependencies from “far away” (according to an ordering) observations.	Katzfuss & Guinness (2021)
Random Fourier Feature	Approximate Gram matrices with low-rank approximations with randomly selected Fourier bases.	Rahimi & Recht (2007)
SPDE-GP (Sarkka et al’s Version)	Reformulate the GP as the stationary solution to an S(P)DE and thus convert the static regression to dynamic sequential inference.	Särkkä, Solin & Hartikainen (2013)
SPDE-GP (Lindgren et al’s Version)	Reformulate the GP as the solution to an SPDE which gets reformulated into a Gaussian Markov Random Field via a FEM discretisation.	Lindgren, Rue & Lindström (2011)

An Introduction to Gaussian Processes

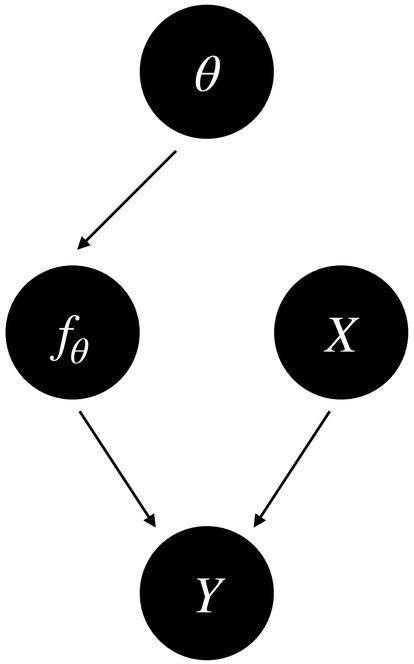
(Hyper)Parameters

$$k_{RBF}(x, x') = \exp\left[-\frac{\|x - x'\|^2}{2l^2}\right]$$

An Introduction to Gaussian Processes

(Hyper)Parameters

$$k_{RBF}(x, x') = \exp\left[-\frac{\|x - x'\|^2}{2l^2}\right]$$

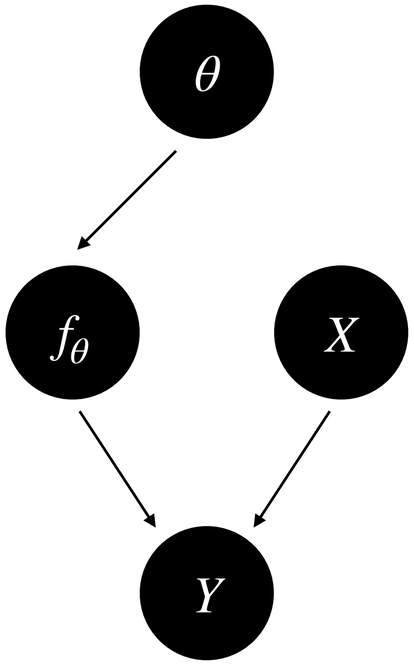


An Introduction to Gaussian Processes

(Hyper)Parameters

$$k_{RBF}(x, x') = \exp\left[-\frac{\|x - x'\|^2}{2l^2}\right]$$

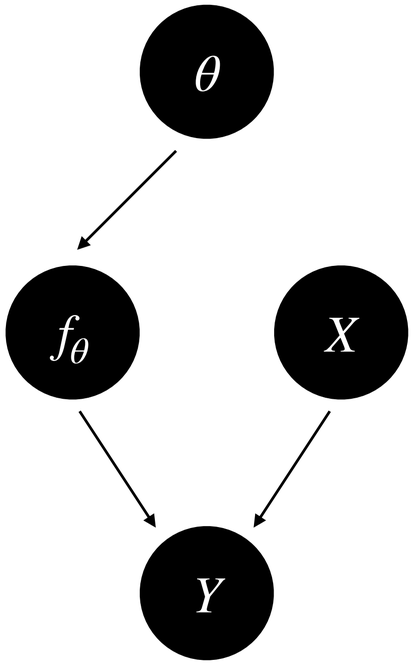
- Option 1: Pre-specify a constant θ.



An Introduction to Gaussian Processes

(Hyper)Parameters

$$k_{RBF}(x, x') = \exp\left[-\frac{\|x - x'\|^2}{2l^2}\right]$$

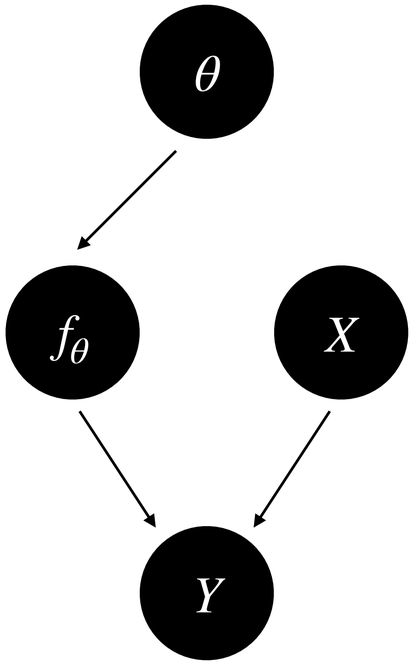


- Option 1: Pre-specify a constant θ.
- Option 2: Learn a θ from data. [Empirical Bayes]

An Introduction to Gaussian Processes

(Hyper)Parameters

$$k_{RBF}(x, x') = \exp\left[-\frac{\|x - x'\|^2}{2l^2}\right]$$

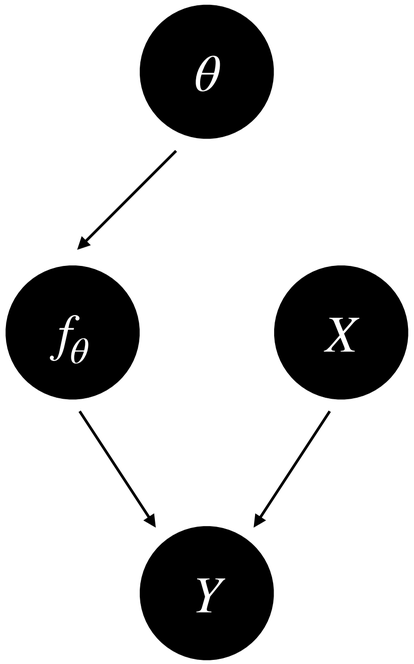


- Option 1: Pre-specify a constant θ.
- Option 2: Learn a θ from data. [Empirical Bayes]
    - Maximum likelihood

An Introduction to Gaussian Processes

(Hyper)Parameters

$$k_{RBF}(x, x') = \exp\left[-\frac{\|x - x'\|^2}{2l^2}\right]$$

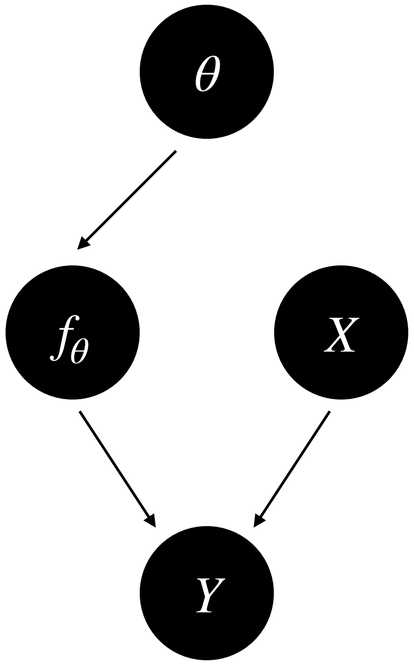


- Option 1: Pre-specify a constant θ.
- Option 2: Learn a θ from data. [Empirical Bayes]
  - Maximum likelihood
  - Select the max-likelihood candidate from a list.

An Introduction to Gaussian Processes

(Hyper)Parameters

$$k_{RBF}(x, x') = \exp\left[-\frac{\|x - x'\|^2}{2l^2}\right]$$

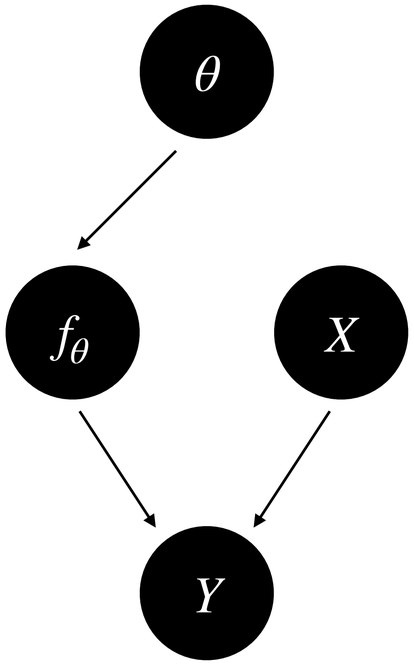


- Option 1: Pre-specify a constant θ.
- Option 2: Learn a θ from data. [Empirical Bayes]
    - Maximum likelihood
    - Select the max-likelihood candidate from a list.
- Option 3: Set a distributional prior on θ.

An Introduction to Gaussian Processes

(Hyper)Parameters

$$k_{RBF}(x, x') = \exp\left[-\frac{\|x - x'\|^2}{2l^2}\right]$$

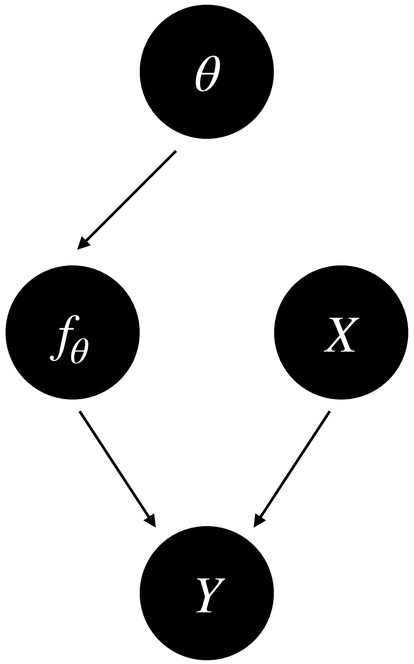


- Option 1: Pre-specify a constant θ.
- Option 2: Learn a θ from data. [Empirical Bayes]
    - Maximum likelihood
    - Select the max-likelihood candidate from a list.
- Option 3: Set a distributional prior on θ.
    - Loses conjugacy in regression.

An Introduction to Gaussian Processes

(Hyper)Parameters

$$k_{RBF}(x, x') = \exp\left[-\frac{\|x - x'\|^2}{2l^2}\right]$$



- Option 1: Pre-specify a constant θ.
- Option 2: Learn a θ from data. [Empirical Bayes]
    - Maximum likelihood
    - Select the max-likelihood candidate from a list.
- Option 3: Set a distributional prior on θ.
    - Loses conjugacy in regression.
    - MCMC for exact inference, VB for approximate.

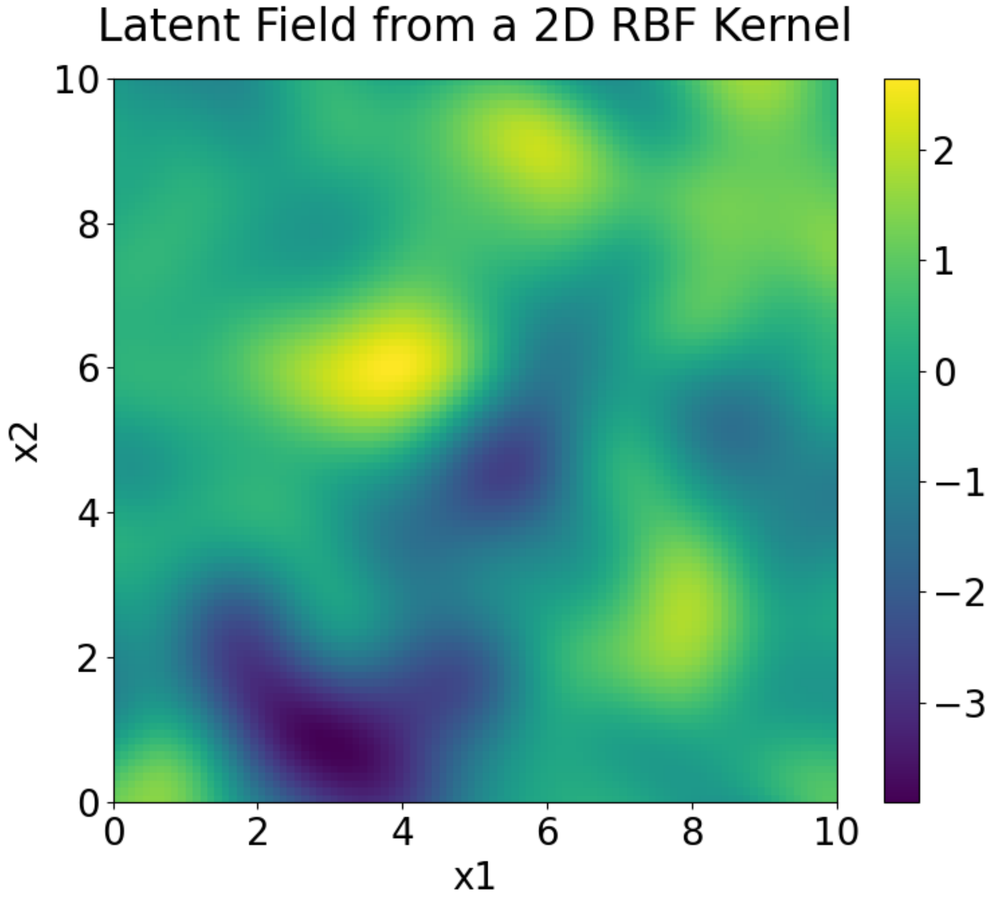
An Introduction to Gaussian Processes

The Recurring Example

$$k_{RBF}(x, x') = \exp\left[-\frac{\|x - x'\|^2}{2l^2}\right]$$

$$x \in \mathbb{R}^2, \qquad l = 1$$

Goal: Learn the latent field using as few observations as possible.



An Introduction to Gaussian Processes

The Recurring Example

An Introduction to Gaussian Processes

The Recurring Example

Initial (Space-Filling) Observations $\mathscr{D}_0$
via Low-Discrepancy Sequences

An Introduction to Gaussian Processes

The Recurring Example

Initial (Space-Filling) Observations $\mathscr{D}_0$
via Low-Discrepancy Sequences

- Latin Hypercube

An Introduction to Gaussian Processes

The Recurring Example

Initial (Space-Filling) Observations $\mathscr{D}_0$
via Low-Discrepancy Sequences

- Latin Hypercube
- Sobol Sequence

An Introduction to Gaussian Processes

The Recurring Example

Initial (Space-Filling) Observations $\mathscr{D}_0$
via Low-Discrepancy Sequences

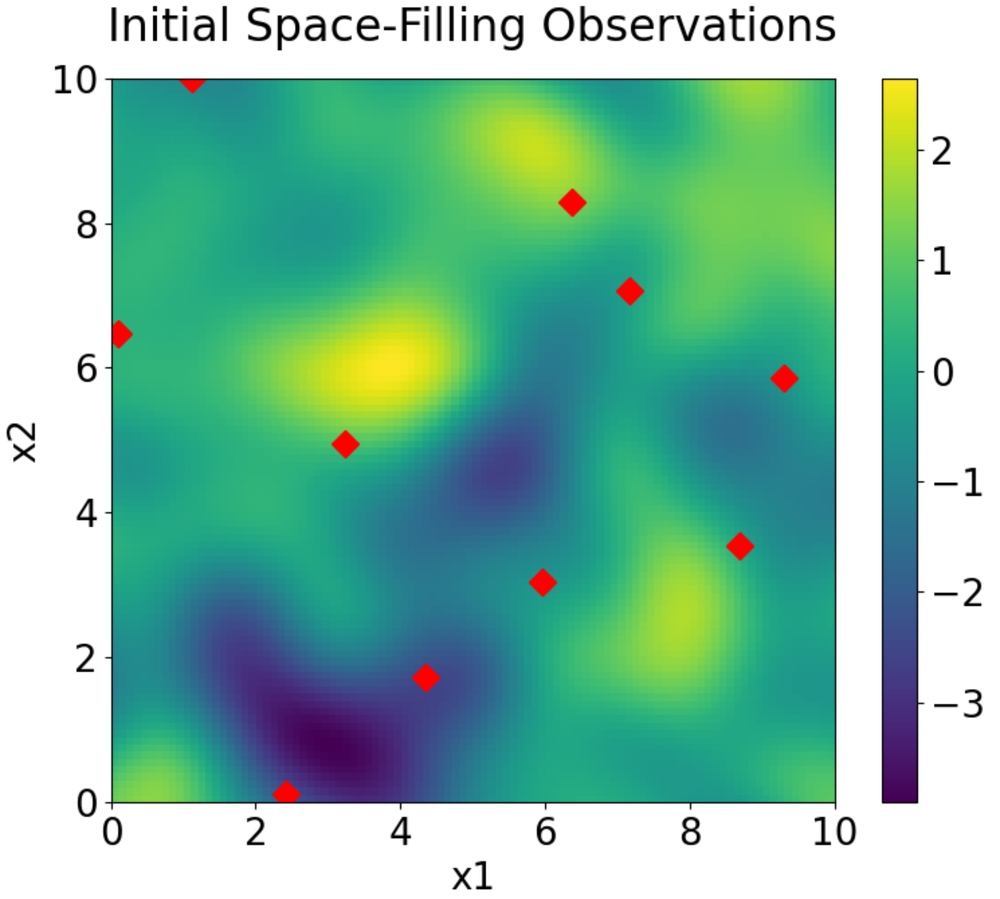
- Latin Hypercube
- Sobol Sequence
- Halton Sequence

An Introduction to Gaussian Processes

The Recurring Example

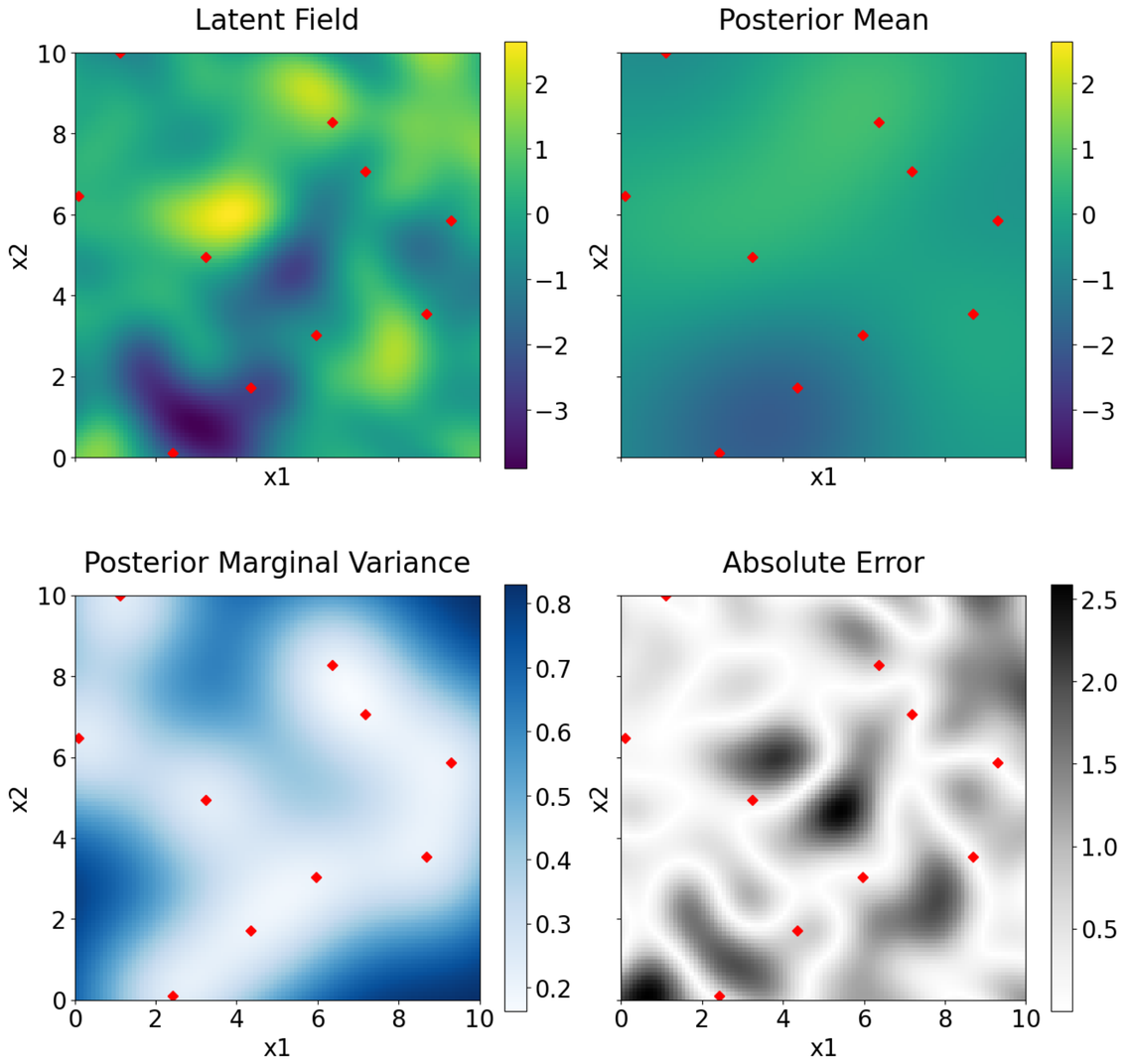
Initial (Space-Filling) Observations $\mathscr{D}_0$
via Low-Discrepancy Sequences

- Latin Hypercube
- Sobol Sequence
- Halton Sequence



An Introduction to Gaussian Processes

The Recurring Example



Hyperparameters (kernel lengthscale, variance, and observation noise) learned via MLE.

Common Acquisition Functions

Common Acquisition Functions

$$x^*_{n+1} := \underset{x \in X}{\operatorname{argmax}}\ \mathbb{E}_{y \sim p(\cdot|x,\mathscr{D}_n)}[U(y)] =: \underset{x \in X}{\operatorname{argmax}}\ \operatorname{acq}(x)$$

Common Acquisition Functions

$$x^*_{n+1} := \underset{x \in X}{\operatorname{argmax}}\ \mathbb{E}_{y \sim p(\cdot|x,\mathcal{D}_n)}[U(y)] =: \underset{x \in X}{\operatorname{argmax}}\ \operatorname{acq}(x)$$

Acquisition function measures the ‘value’ of an observation location.

Common Acquisition Functions

$$x^*_{n+1} := \underset{x \in X}{\operatorname{argmax}}\ \mathbb{E}_{y \sim p(\cdot|x,\mathcal{D}_n)}[U(y)] =: \underset{x \in X}{\operatorname{argmax}}\ \operatorname{acq}(x)$$

Acquisition function measures the ‘value’ of an observation location.

What is the goal of our data acquisitions?

$$x^*_{n+1} = \underset{x \in X}{\operatorname{argmax}}\ \operatorname{acq}(x)$$

Common Acquisition Functions

Maximum Variance

$$x^*_{n+1} = \underset{x \in X}{\text{argmax}}\ \text{acq}(x)$$

Common Acquisition Functions

Maximum Variance

MaxVar: observe at locations which we are the most uncertain about (highest variance).

$$x^*_{n+1} = \underset{x \in X}{\operatorname{argmax}}\ \operatorname{acq}(x)$$

Common Acquisition Functions

Maximum Variance

MaxVar: observe at locations which we are the most uncertain about (highest variance).

$$\operatorname{acq}^{\operatorname{MaxVar}}(x) := \operatorname{Var}_{y \sim p(\cdot|x,\mathscr{D})}[y]$$

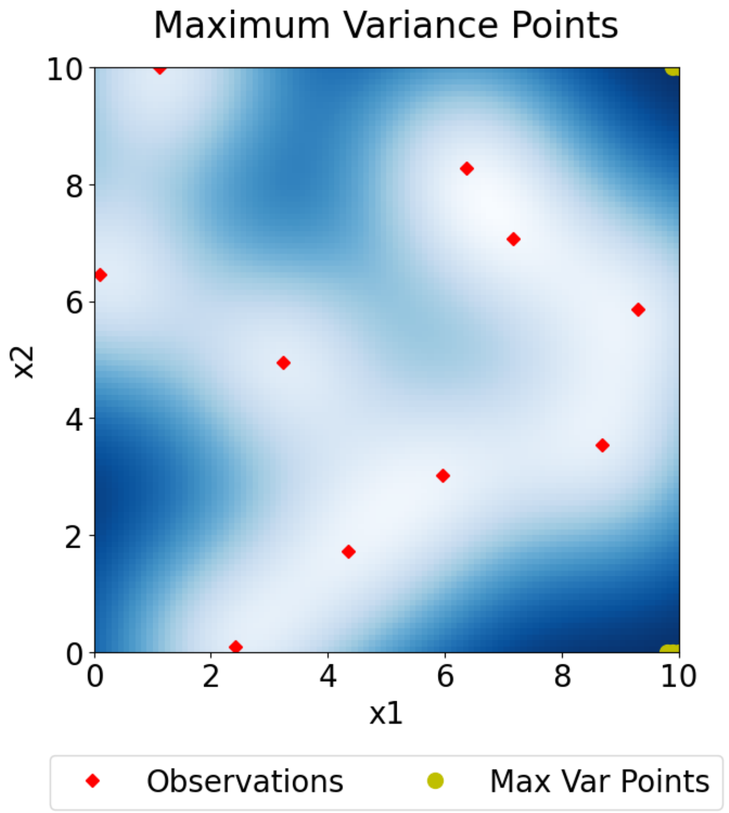
$$x^*_{n+1} = \underset{x \in X}{\operatorname{argmax}}\ \operatorname{acq}(x)$$

Common Acquisition Functions

Maximum Variance

MaxVar: observe at locations which we are the most uncertain about (highest variance).

$$\operatorname{acq}^{\operatorname{MaxVar}}(x) := \operatorname{Var}_{y \sim p(\cdot|x,\mathcal{D})}[y]$$



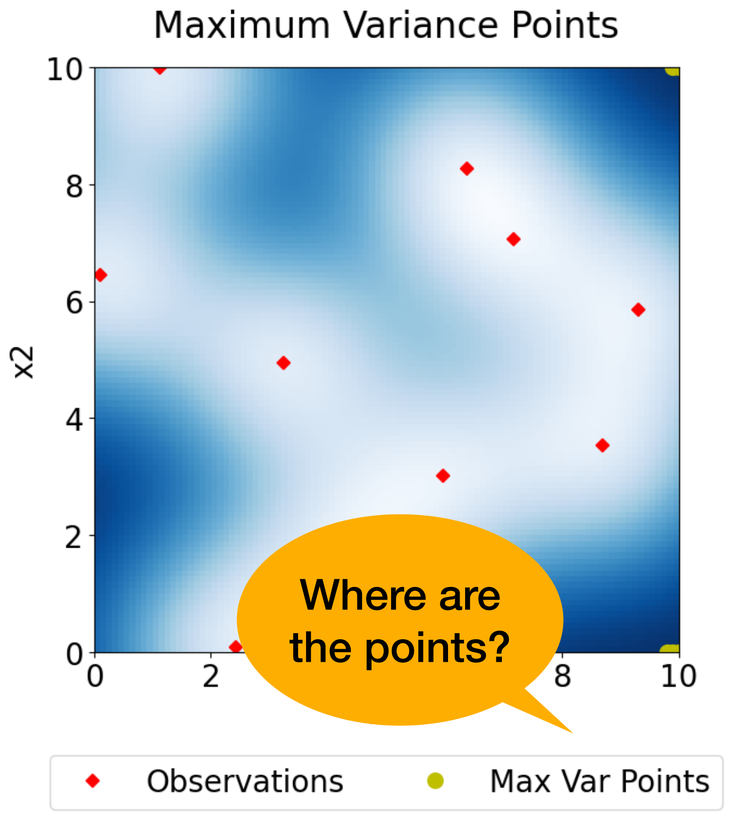
$$x^*_{n+1} = \underset{x \in X}{\operatorname{argmax}}\ \operatorname{acq}(x)$$

Common Acquisition Functions

Maximum Variance

MaxVar: observe at locations which we are the most uncertain about (highest variance).

$$\operatorname{acq}^{\operatorname{MaxVar}}(x) := \operatorname{Var}_{y \sim p(\cdot|x,\mathcal{D})}[y]$$



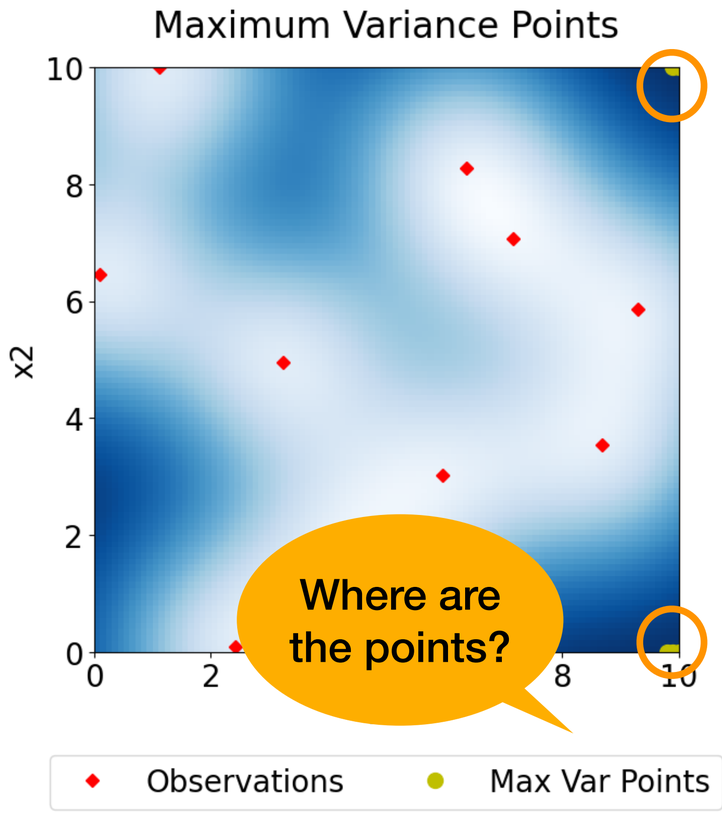
$$x^*_{n+1} = \underset{x \in X}{\operatorname{argmax}}\ \operatorname{acq}(x)$$

Common Acquisition Functions

Maximum Variance

MaxVar: observe at locations which we are the most uncertain about (highest variance).

$$\operatorname{acq}^{\operatorname{MaxVar}}(x) := \operatorname{Var}_{y \sim p(\cdot|x,\mathcal{D})}[y]$$



$$x^*_{n+1} = \underset{x \in X}{\operatorname{argmax}}\ \operatorname{acq}(x)$$

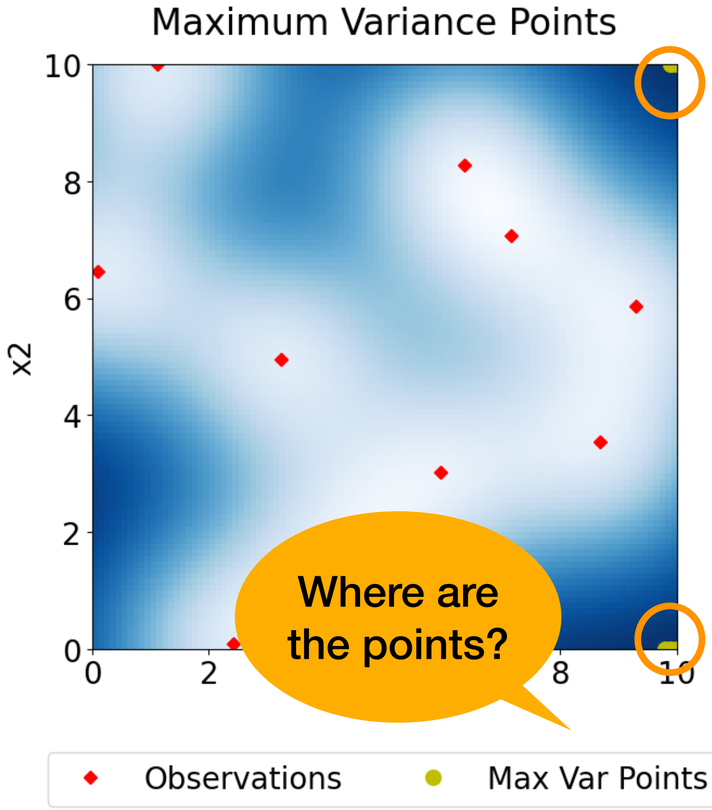
Common Acquisition Functions

Maximum Variance

MaxVar: observe at locations which we are the most uncertain about (highest variance).

$$\operatorname{acq}^{\operatorname{MaxVar}}(x) := \operatorname{Var}_{y \sim p(\cdot|x,\mathcal{D})}[y]$$

$$k_{RBF}(x, x') = \exp\left[-\frac{\|x - x'\|^2}{2l^2}\right]$$



$$x^*_{n+1} = \underset{x \in X}{\text{argmax}}\ \text{acq}(x)$$

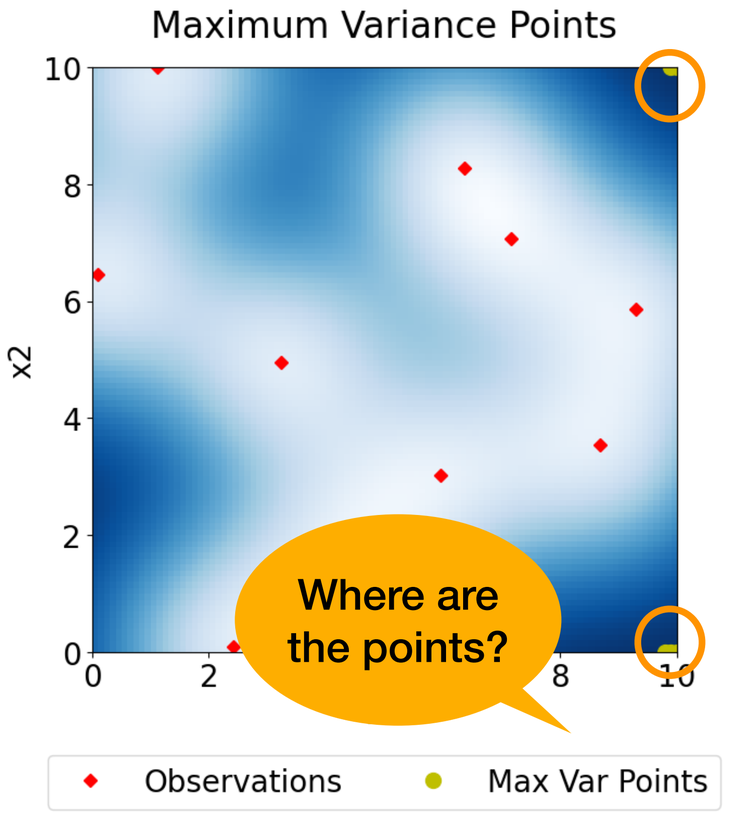
Common Acquisition Functions

Maximum Variance

MaxVar: observe at locations which we are the most uncertain about (highest variance).

$$\text{acq}^{\text{MaxVar}}(x) := \text{Var}_{y \sim p(\cdot|x,\mathcal{D})}[y]$$

$$k_{RBF}(x, x') = \exp\left[-\frac{\|x - x'\|^2}{2l^2}\right]$$



$$x^*_{n+1} = \underset{x \in X}{\text{argmax}}\ \text{acq}(x)$$

Common Acquisition Functions

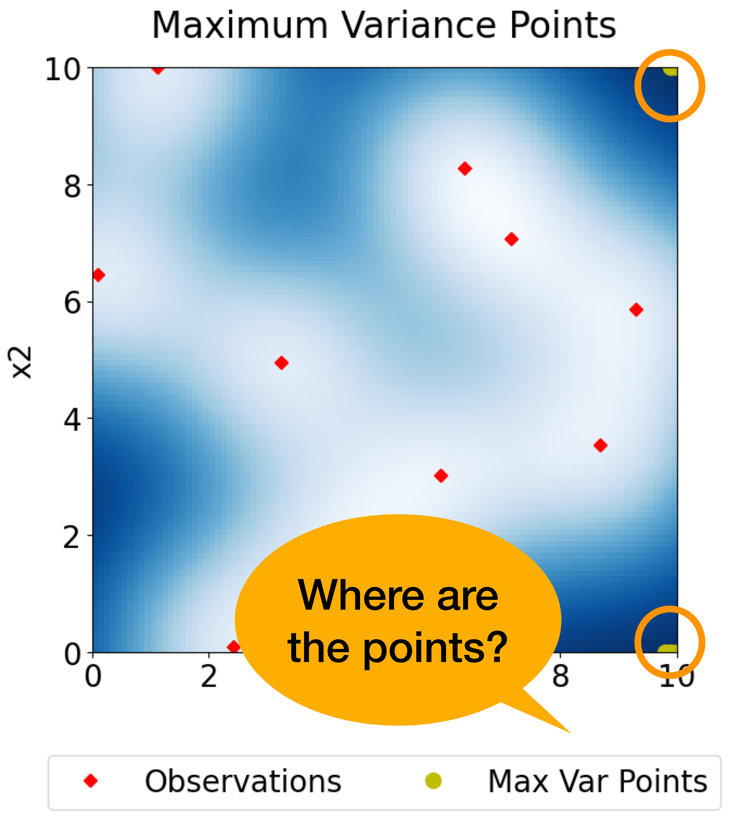
Maximum Variance

MaxVar: observe at locations which we are the most uncertain about (highest variance).

$$\text{acq}^{\text{MaxVar}}(x) := \text{Var}_{y \sim p(\cdot|x,\mathcal{D})}[y]$$

- Common kernels are distance-driven.

$$k_{RBF}(x, x') = \exp\left[-\frac{\|x - x'\|^2}{2l^2}\right]$$



$$x^*_{n+1} = \underset{x \in X}{\text{argmax}}\ \text{acq}(x)$$

Common Acquisition Functions

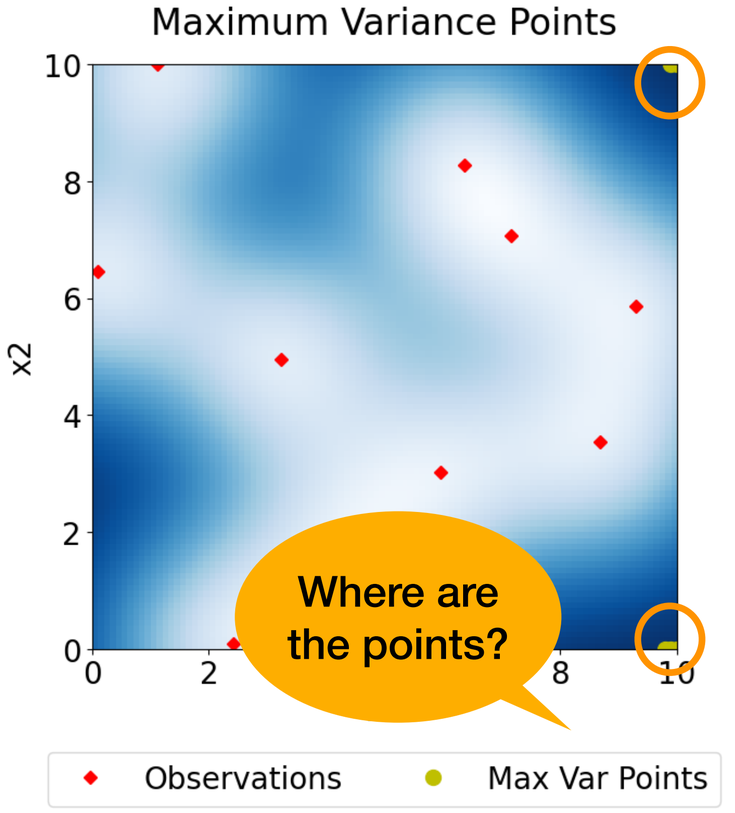
Maximum Variance

MaxVar: observe at locations which we are the most uncertain about (highest variance).

$$\text{acq}^{\text{MaxVar}}(x) := \text{Var}_{y \sim p(\cdot|x,\mathcal{D})}[y]$$

- Common kernels are distance-driven.
- So predictive variance increases as points get further away from observations.

$$k_{RBF}(x, x') = \exp\left[-\frac{\|x - x'\|^2}{2l^2}\right]$$



$$x^*_{n+1} = \underset{x \in X}{\operatorname{argmax}}\ \operatorname{acq}(x)$$

Common Acquisition Functions

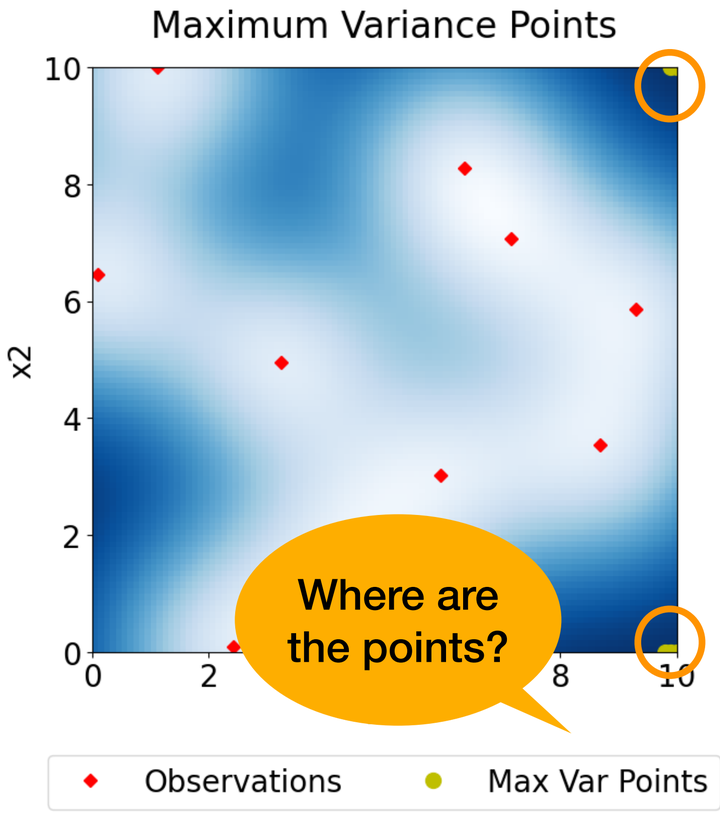
Maximum Variance

MaxVar: observe at locations which we are the most uncertain about (highest variance).

$$\operatorname{acq}^{\operatorname{MaxVar}}(x) := \operatorname{Var}_{y \sim p(\cdot|x,\mathcal{D})}[y]$$

- Common kernels are distance-driven.
- So predictive variance increases as points get further away from observations.
- Max-Var seeks boundary points as they tend to be the furthest from existing observations — undesirable!!

$$k_{RBF}(x, x') = \exp\left[-\frac{\|x - x'\|^2}{2l^2}\right]$$



$$x^*_{n+1} = \underset{x \in X}{\operatorname{argmax}}\ \operatorname{acq}(x)$$

Common Acquisition Functions

Maximum Integrated Variance

$$x^*_{n+1} = \underset{x \in X}{\operatorname{argmax}}\ \operatorname{acq}(x)$$

Common Acquisition Functions

Maximum Integrated Variance

MaxIntVar: observe at locations which reduces the total predictive variance the most.

$$x^*_{n+1} = \underset{x \in X}{\operatorname{argmax}}\ \operatorname{acq}(x)$$

Common Acquisition Functions

Maximum Integrated Variance

MaxIntVar: observe at locations which reduces the total predictive variance the most.

$$\operatorname{acq}^{\text{MaxIntVar}}(x) := \sum_{x'} \operatorname{Var}_{y' \sim p(\cdot|x', \mathcal{D})}[y'] - \sum_{x'} \operatorname{Var}_{y' \sim p(\cdot|x', \mathcal{D} \cup \{x\})}[y']$$

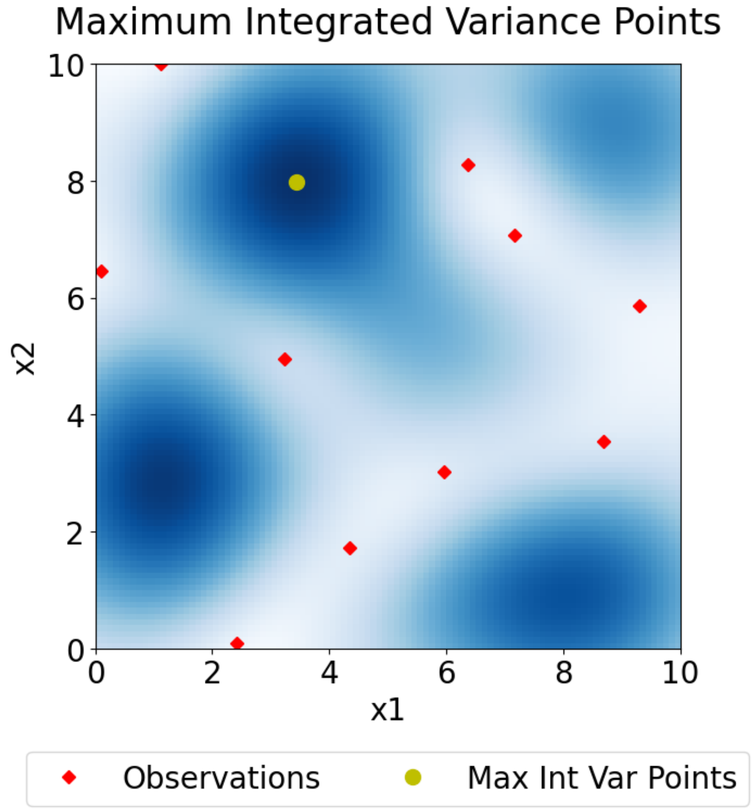
$$x^*_{n+1} = \underset{x \in X}{\operatorname{argmax}}\ \operatorname{acq}(x)$$

Common Acquisition Functions

Maximum Integrated Variance

MaxIntVar: observe at locations which reduces the total predictive variance the most.

$$\operatorname{acq}^{\text{MaxIntVar}}(x) := \sum_{x'} \operatorname{Var}_{y' \sim p(\cdot|x', \mathcal{D})}[y'] - \sum_{x'} \operatorname{Var}_{y' \sim p(\cdot|x', \mathcal{D} \cup \{x\})}[y']$$



$$x^*_{n+1} = \underset{x \in X}{\operatorname{argmax}}\ \operatorname{acq}(x)$$

Common Acquisition Functions

Expected Information Gain

$$x^*_{n+1} = \underset{x \in X}{\text{argmax}}\ \text{acq}(x)$$

Common Acquisition Functions

Expected Information Gain

EIG: observe at locations which gain the most information content (i.e. reduces the most *entropy*).

$$x^*_{n+1} = \underset{x \in X}{\operatorname{argmax}}\ \operatorname{acq}(x)$$

Common Acquisition Functions

Expected Information Gain

EIG: observe at locations which gain the most information content (i.e. reduces the most *entropy*).

$$H(A) := \mathbb{E}_A[-\log p(A)] = -\sum_a p(a)\log p(a)$$

$$x^*_{n+1} = \underset{x \in X}{\operatorname{argmax}}\ \operatorname{acq}(x)$$

Common Acquisition Functions

Expected Information Gain

EIG: observe at locations which gain the most information content (i.e. reduces the most *entropy*).

$$H(A) := \mathbb{E}_A[-\log p(A)] = -\sum_a p(a)\log p(a)$$

Examples

$$x^*_{n+1} = \underset{x \in X}{\operatorname{argmax}}\ \operatorname{acq}(x)$$

Common Acquisition Functions

Expected Information Gain

EIG: observe at locations which gain the most information content (i.e. reduces the most *entropy*).

$$H(A) := \mathbb{E}_A[-\log p(A)] = -\sum_a p(a)\log p(a)$$

Examples

- The entropy of a fair coin flip is $H(\text{coin flip}) = -0.5 \times \log(1/2) * 2 = \log 2.$

$$x^*_{n+1} = \underset{x \in X}{\operatorname{argmax}}\ \operatorname{acq}(x)$$

Common Acquisition Functions

Expected Information Gain

EIG: observe at locations which gain the most information content (i.e. reduces the most *entropy*).

$$H(A) := \mathbb{E}_A[-\log p(A)] = -\sum_a p(a)\log p(a)$$

Examples

- The entropy of a fair coin flip is $H(\text{coin flip}) = -0.5 \times \log(1/2) * 2 = \log 2.$
- The entropy of a multivariate Gaussian $G \sim N_d(\mu, \Sigma)$ is $H(G) = \frac{d}{2}[1 + \log(2\pi)] + \log\det(\Sigma).$

$$x^*_{n+1} = \underset{x \in X}{\operatorname{argmax}}\ \operatorname{acq}(x)$$

Common Acquisition Functions

Expected Information Gain

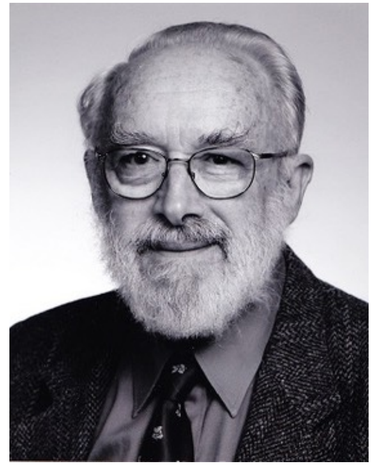
EIG: observe at locations which gain the most information content (i.e. reduces the most *entropy*).

December, 1956

On a Measure of the Information Provided by an Experiment

D. V. Lindley

Ann. Math. Statist. 27(4): 986-1005 (December, 1956). DOI: 10.1214/aoms/1177728069



OG Bayesian

- The entropy of a fair coin flip is $H(\text{coin flip}) = -0.5 \times \log(1/2) * 2 = \log 2.$
- The entropy of a multivariate Gaussian $G \sim N_d(\mu, \Sigma)$ is $H(G) = \frac{d}{2}[1 + \log(2\pi)] + \log\det(\Sigma).$

$$x^*_{n+1} = \underset{x \in X}{\operatorname{argmax}}\ \operatorname{acq}(x)$$

Common Acquisition Functions

Expected Information Gain

$$x^*_{n+1} = \underset{x \in X}{\operatorname{argmax}}\ \operatorname{acq}(x)$$

Common Acquisition Functions

Expected Information Gain

EIG: observe at locations which gain the most information content (i.e. reduces the most entropy).

$$x^*_{n+1} = \underset{x \in X}{\operatorname{argmax}}\ \operatorname{acq}(x)$$

Common Acquisition Functions

Expected Information Gain

EIG: observe at locations which gain the most information content (i.e. reduces the most entropy).

$$\operatorname{acq}^{\text{MaxIntVar}}(x) := \mathbb{E}_{y \sim p(\cdot|x,\mathscr{D})}\left[H[p(\,\cdot\,|\,\mathscr{D})] - H[p(\,\cdot\,|\,\mathscr{D} \cup \{(x,y)\})]\right]$$

$$x^*_{n+1} = \underset{x \in X}{\operatorname{argmax}}\ \operatorname{acq}(x)$$

Common Acquisition Functions

Expected Information Gain

EIG: observe at locations which gain the most information content (i.e. reduces the most entropy).

$$\operatorname{acq}^{\text{MaxIntVar}}(x) := \mathbb{E}_{y \sim p(\cdot|x,\mathscr{D})}\left[H[p(\,\cdot\,|\,\mathscr{D})] - H[p(\,\cdot\,|\,\mathscr{D} \cup \{(x,y)\})]\right]$$

When we are interested in the predictive distribution of a GP on *finite* test points, the (expected) entropy admits a closed form as the predictive distribution is a MVN.

$$x^*_{n+1} = \underset{x \in X}{\operatorname{argmax}}\ \operatorname{acq}(x)$$

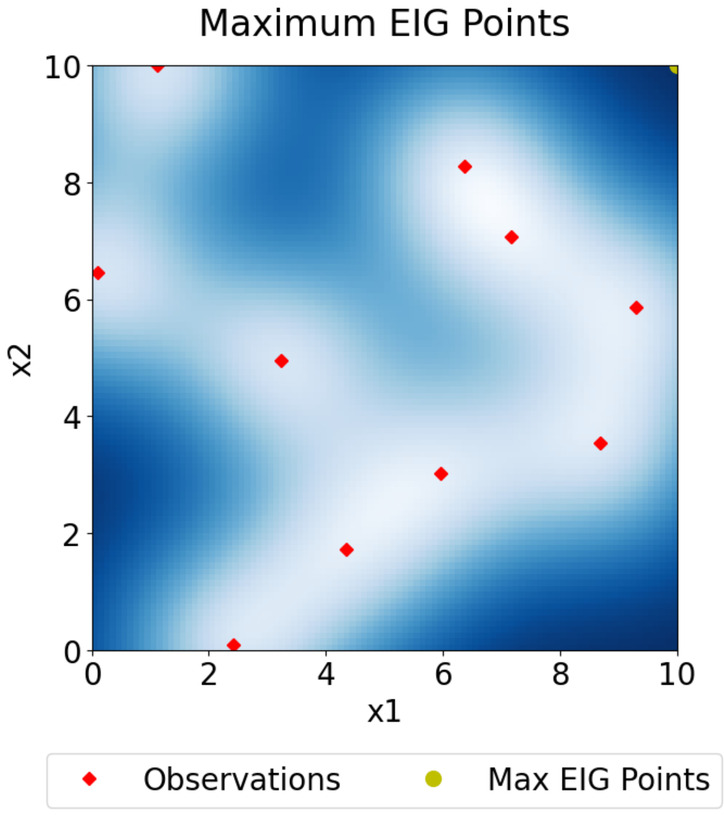
Common Acquisition Functions

Expected Information Gain

EIG: observe at locations which gain the most information content (i.e. reduces the most entropy).

$$\operatorname{acq}^{\text{MaxIntVar}}(x) := \mathbb{E}_{y \sim p(\cdot|x,\mathscr{D})}\left[H[p(\,\cdot\,|\mathscr{D})] - H[p(\,\cdot\,|\mathscr{D} \cup \{(x,y)\})]\right]$$

When we are interested in the predictive distribution of a GP on *finite* test points, the (expected) entropy admits a closed form as the predictive distribution is a MVN.



$$x^*_{n+1} = \underset{x \in X}{\operatorname{argmax}}\ \operatorname{acq}(x)$$

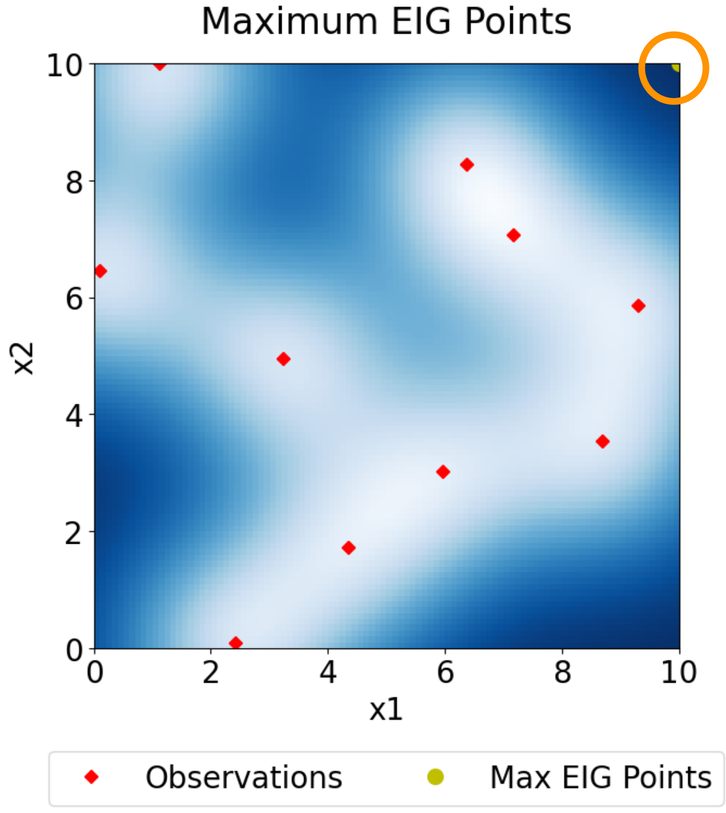
Common Acquisition Functions

Expected Information Gain

EIG: observe at locations which gain the most information content (i.e. reduces the most entropy).

$$\operatorname{acq}^{\text{MaxIntVar}}(x) := \mathbb{E}_{y \sim p(\cdot|x,\mathscr{D})}\left[H[p(\,\cdot\,|\,\mathscr{D})] - H[p(\,\cdot\,|\,\mathscr{D} \cup \{(x,y)\})]\right]$$

When we are interested in the predictive distribution of a GP on *finite* test points, the (expected) entropy admits a closed form as the predictive distribution is a MVN.



$$x^*_{n+1} = \underset{x \in X}{\operatorname{argmax}}\ \operatorname{acq}(x)$$

Common Acquisition Functions

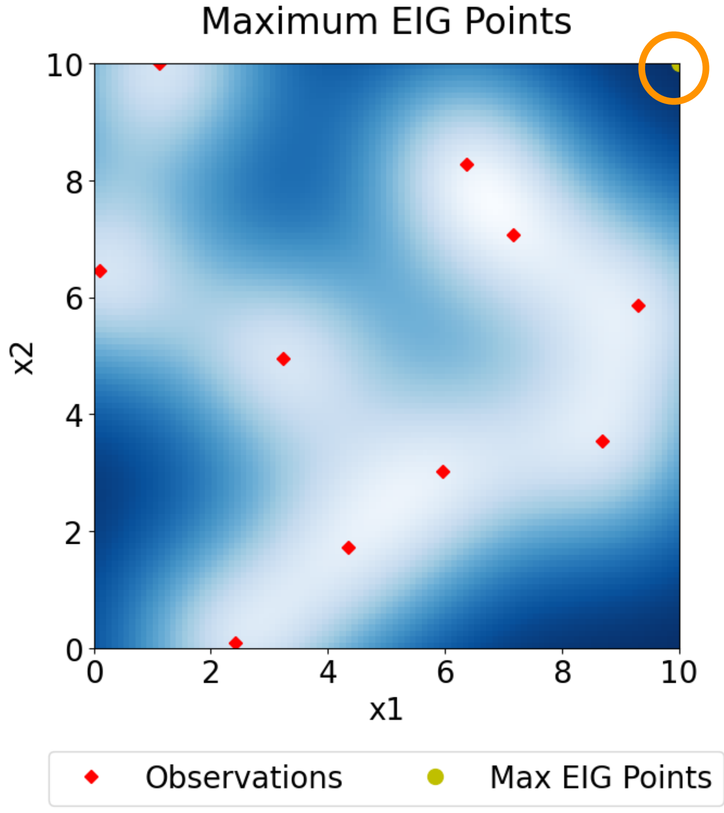
Expected Information Gain

EIG: observe at locations which gain the most information content (i.e. reduces the most entropy).

$$\operatorname{acq}^{\text{MaxIntVar}}(x) := \mathbb{E}_{y \sim p(\cdot|x,\mathscr{D})}\left[H[p(\,\cdot\,|\mathscr{D})] - H[p(\,\cdot\,|\mathscr{D} \cup \{(x,y)\})]\right]$$

When we are interested in the predictive distribution of a GP on *finite* test points, the (expected) entropy admits a closed form as the predictive distribution is a MVN.

*It can be shown that if one cares only about **one** more point's information gain, it is equivalent to do MaxVar.*



$$x^*_{n+1} = \underset{x \in X}{\operatorname{argmax}}\ \operatorname{acq}(x)$$

Common Acquisition Functions

$$x^*_{n+1} = \underset{x \in X}{\text{argmax}}\ \text{acq}(x)$$

Common Acquisition Functions

Strategies

1. Variance.
2. Integrated Variance Change.
3. Information Gain.

$$x^*_{n+1} = \underset{x \in X}{\operatorname{argmax}}\ \operatorname{acq}(x)$$

Common Acquisition Functions

Strategies
1. Variance.
2. Integrated Variance Change.
3. Information Gain.

Objectives
1. Predictive distribution.
2. Model Parameters.
3. Quantities of Interests.

$$x^*_{n+1} = \underset{x \in X}{\operatorname{argmax}}\ \operatorname{acq}(x)$$

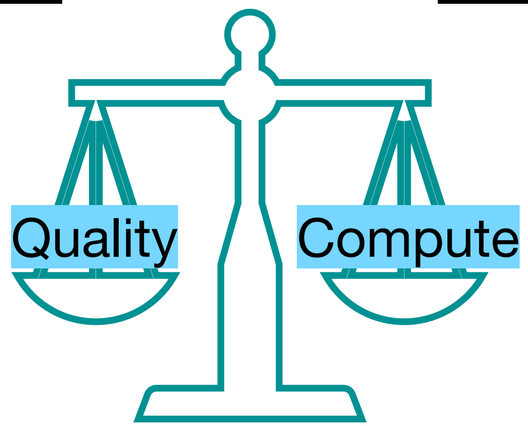
Common Acquisition Functions

Strategies		Objectives
1. Variance.	×	1. Predictive distribution.
2. Integrated Variance Change.		2. Model Parameters.
3. Information Gain.		3. Quantities of Interests.

$$x^*_{n+1} = \underset{x \in X}{\operatorname{argmax}}\ \operatorname{acq}(x)$$

Common Acquisition Functions

Strategies		Objectives
1. Variance.	×	1. Predictive distribution.
2. Integrated Variance Change.		2. Model Parameters.
3. Information Gain.		3. Quantities of Interests.



Selected Topics

Sequential Decision-Making

- Start with an initial dataset $\mathscr{D}_0$.
- For decision numbers $n = 1, \ldots, N$:
  - Update predictive model $p(\cdot \mid D_{n-1})$.
  - Compute $x_n^* := \underset{x \in X}{\operatorname{argmax}} \, \mathbb{E}_{y \sim p(\cdot \mid x, \mathscr{D}_{n-1})}[U(y)]$.
  - Make decision and observe $\{(x_n^*, y_n)\}$.
  - Append observation $\mathscr{D}_n = \mathscr{D}_{n-1} \cup \{(x_n^*, y_n)\}$.

What if my decision yields a sequence of outcomes?

What if I want to make multiple decisions at once?

Shouldn't we be (even) more strategic and consider future decisions?

Selected Topics

What if my decision yields a sequence of outcomes?

What if I want to make multiple decisions at once?

Shouldn’t
we be (even) more strategic and consider future decisions?

Selected Topics

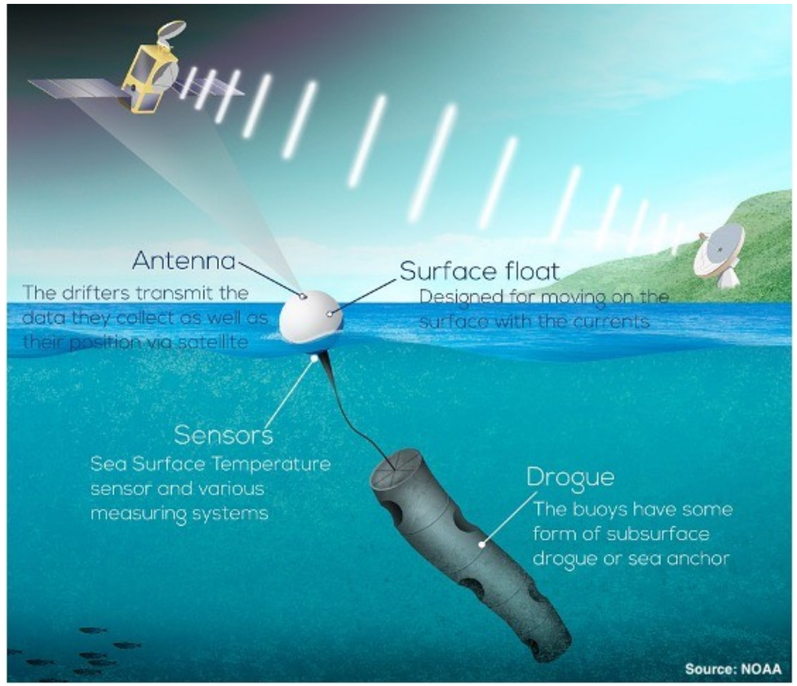
Trajectory Look-Aheads

What if my decision yields a sequence of outcomes?

Selected Topics

Trajectory Look-Aheads

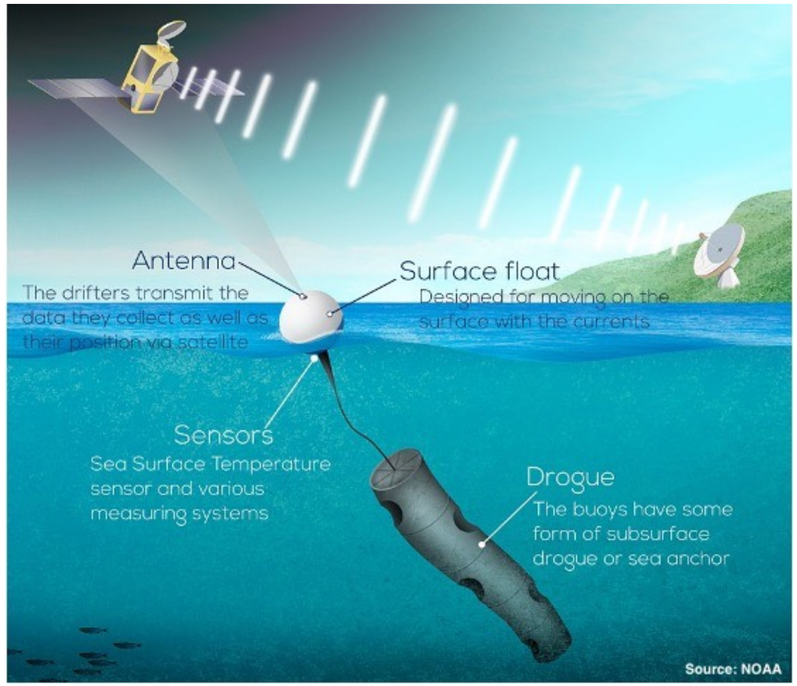
What if my decision yields a sequence of outcomes?

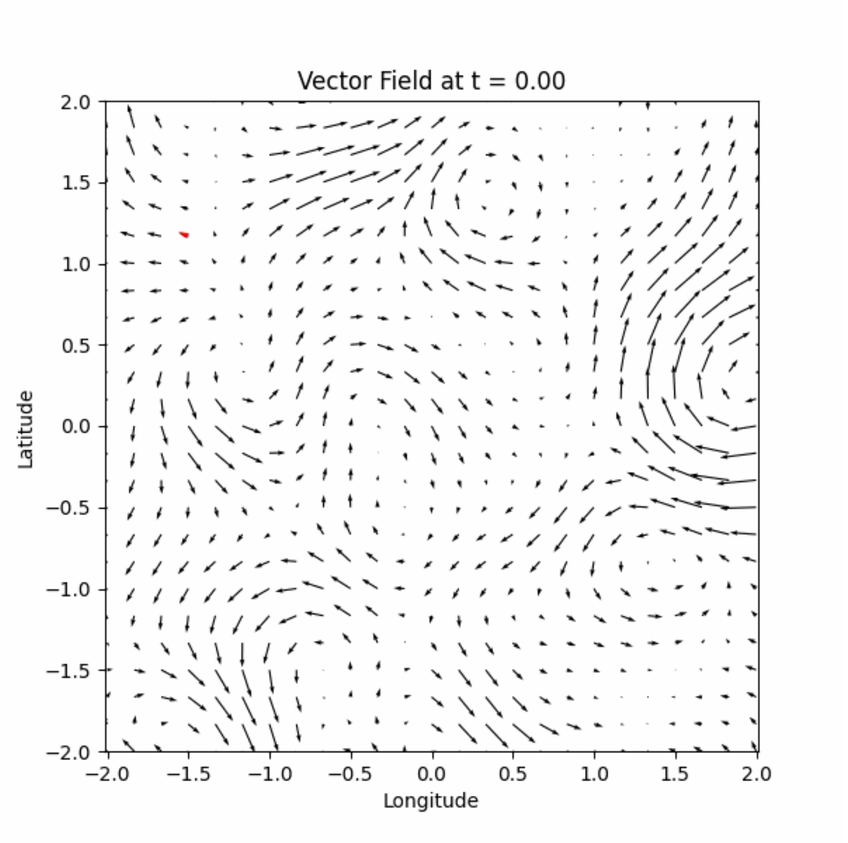


Selected Topics

Trajectory Look-Aheads

What if my decision yields a sequence of outcomes?





Selected Topics

Trajectory Look-Aheads

What if my decision yields a sequence of outcomes?

Selected Topics

Trajectory Look-Aheads

What if my decision yields a sequence of outcomes?

$$x^*_{n+1} := \underset{x \in X}{\text{argmax}} \; \mathbb{E}_{y \sim p(\cdot|x,\mathscr{D}_n)}[U(y)]$$

Selected Topics

Trajectory Look-Aheads

What if my decision yields a sequence of outcomes?

$$x^*_{n+1} := \underset{x \in X}{\operatorname{argmax}} \; \mathbb{E}_{y \sim p(\cdot|x,\mathscr{D}_n)}[U(y)]$$

Unable to capture future observations made by the same sensor.

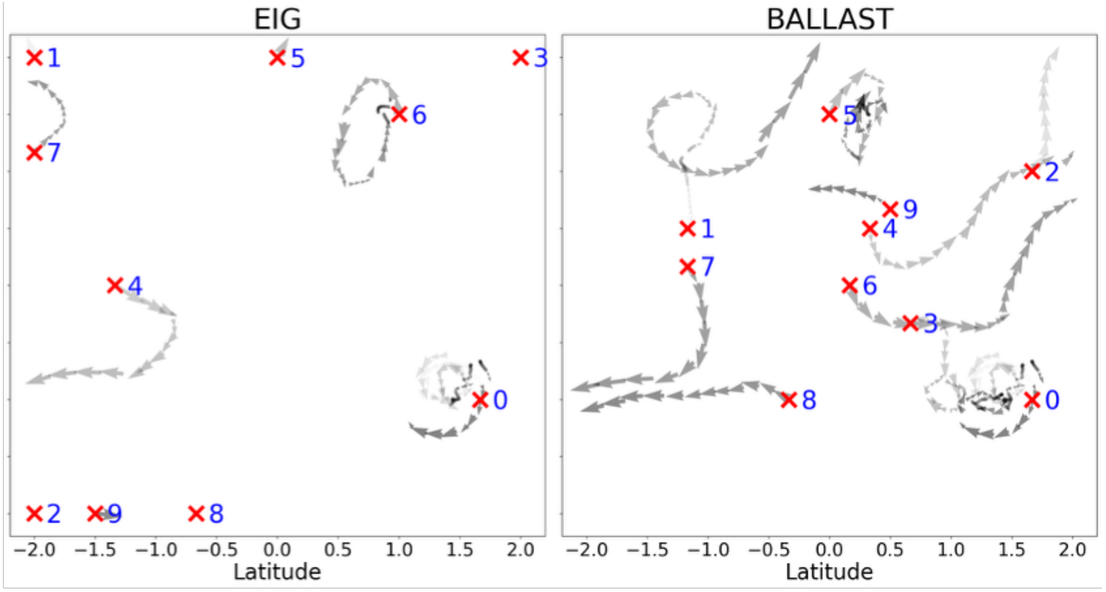
Selected Topics

Trajectory Look-Aheads

What if my decision yields a sequence of outcomes?

$$x^*_{n+1} := \underset{x \in X}{\operatorname{argmax}} \; \mathbb{E}_{y \sim p(\cdot|x,\mathscr{D}_n)}[U(y)]$$

Unable to capture future observations made by the same sensor.



Selected Topics

Trajectory Look-Aheads

What if my decision yields a sequence of outcomes?

$$x^*_{n+1} := \underset{x \in X}{\text{argmax}} \; \mathbb{E}_{y \sim p(\cdot|x,\mathscr{D}_n)}[U(y)]$$

Selected Topics

Trajectory Look-Aheads

What if my decision yields a sequence of outcomes?

$$x^*_{n+1} := \underset{x \in X}{\operatorname{argmax}} \; \mathbb{E}_{y \sim p(\cdot|x,\mathscr{D}_n)}[U(y)]$$

Project possible future trajectories and aggregate the utilities.

Selected Topics

Trajectory Look-Aheads

What if my decision yields a sequence of outcomes?

$$x^*_{n+1} := \underset{x \in X}{\operatorname{argmax}} \; \mathbb{E}_{y \sim p(\cdot|x,\mathscr{D}_n)}[U(y)]$$

Project possible future trajectories and aggregate the utilities.

$$x^*_{n+1} := \underset{x \in X}{\operatorname{argmax}} \; \mathbb{E}_{F,y} \left[U(P(x, F, T) \right]$$

Selected Topics

Batch Designs

What if I want to make multiple decisions at once?

Selected Topics

Batch Designs

What if I want to make multiple decisions at once?

$$x^*_{n+1} := \underset{x \in X}{\text{argmax}} \; \mathbb{E}_{y \sim p(\cdot|x,\mathscr{D}_n)}[U(y)]$$

Selected Topics

Batch Designs

What if I want to make multiple decisions at once?

$$x^*_{n+1} := \underset{x \in X}{\operatorname{argmax}} \ \mathbb{E}_{y \sim p(\cdot|x,\mathscr{D}_n)}[U(y)]$$

Instead of $x \in X$, we consider $\mathbf{x} \in X^m$ for an m-batch acquisition.

Selected Topics

Batch Designs

What if I want to make multiple decisions at once?

$$x^*_{n+1} := \underset{x \in X}{\operatorname{argmax}} \ \mathbb{E}_{y \sim p(\cdot|x, \mathscr{D}_n)}[U(y)]$$

Instead of $x \in X$, we consider $\mathbf{x} \in X^m$ for an m-batch acquisition.

$$\mathbf{x}^*_{n+1} := \underset{\mathbf{x} \in X^m}{\operatorname{argmax}} \ \mathbb{E}_{\mathbf{y} \sim p(\cdot|\mathbf{x}, \mathscr{D}_n)}[U(\mathbf{y})]$$

Selected Topics

Batch Designs

What if I want to make multiple decisions at once?

$$\mathbf{x}^*_{n+1} := \underset{\mathbf{x} \in X^m}{\operatorname{argmax}} \; \mathbb{E}_{\mathbf{y} \sim p(\cdot|\mathbf{x}, \mathscr{D}_n)}[U(\mathbf{y})]$$

Selected Topics

Batch Designs

What if I want to make multiple decisions at once?

$$\mathbf{x}^*_{n+1} := \underset{\mathbf{x}\in X^m}{\operatorname{argmax}}\ \mathbb{E}_{\mathbf{y}\sim p(\cdot|\mathbf{x},\mathscr{D}_n)}[U(\mathbf{y})]$$

The Greedy

- Select the best $\hat{x}_1$ from $\{\mathrm{acq}(x)\}_x$.
- Append $\hat{x}_1$ to $\mathscr{D}_n$ and get $\mathscr{D}_n^{+1}$.
- Select the best $\hat{x}_2$ with $\mathscr{D}_n^{+1}$.
- Repeat m times.

Selected Topics

Batch Designs

What if I want to make multiple decisions at once?

$$\mathbf{x}^*_{n+1} := \underset{\mathbf{x}\in X^m}{\operatorname{argmax}}\ \mathbb{E}_{\mathbf{y}\sim p(\cdot|\mathbf{x},\mathscr{D}_n)}[U(\mathbf{y})]$$

The Greedy

- Select the best $\hat{x}_1$ from $\{\mathrm{acq}(x)\}_x$.
- Append $\hat{x}_1$ to $\mathscr{D}_n$ and get $\mathscr{D}_n^{+1}$.
- Select the best $\hat{x}_2$ with $\mathscr{D}_n^{+1}$.
- Repeat m times.

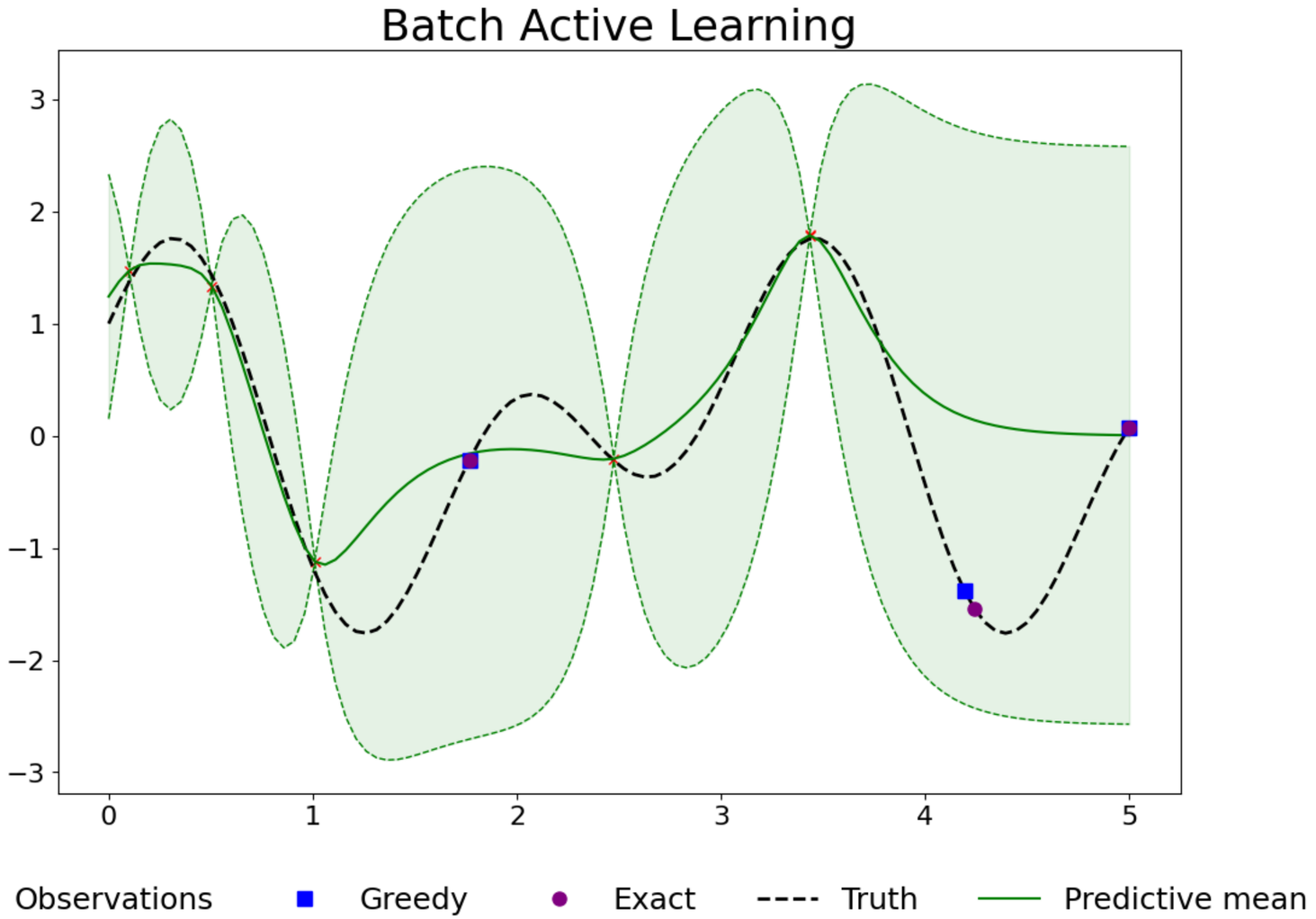
The Exact

- Solve the (at least) m -dimensional optimization problem.

Selected Topics

Batch Designs

What if I want to make multiple decisions at once?



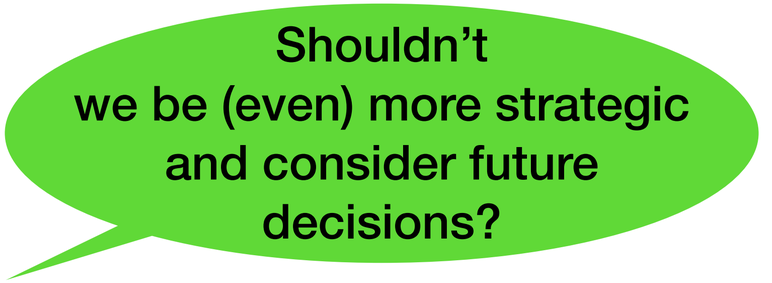
Selected Topics

Non-Myopic Designs

Shouldn’t
we be (even) more strategic
and consider future
decisions?

Selected Topics

Non-Myopic Designs



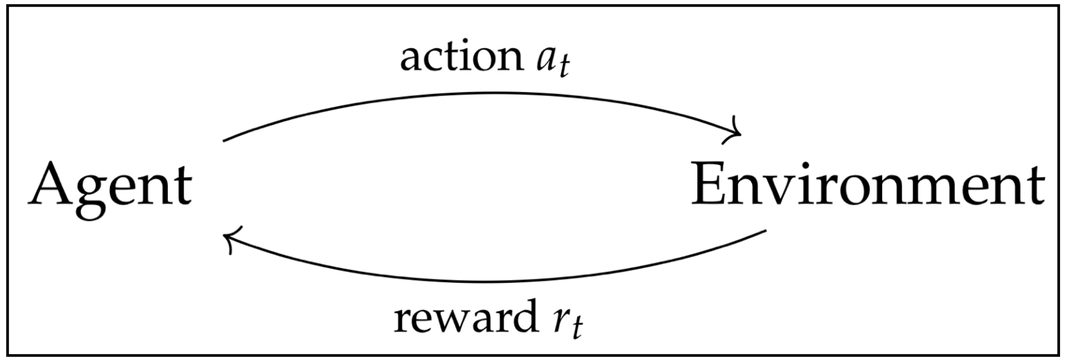
Sequential Designs are actually (greedy) Markov Decision Processes in disguise!

Selected Topics

Non-Myopic Designs

Shouldn't
we be (even) more strategic
and consider future
decisions?

Sequential Designs are actually (greedy) Markov Decision Processes in disguise!

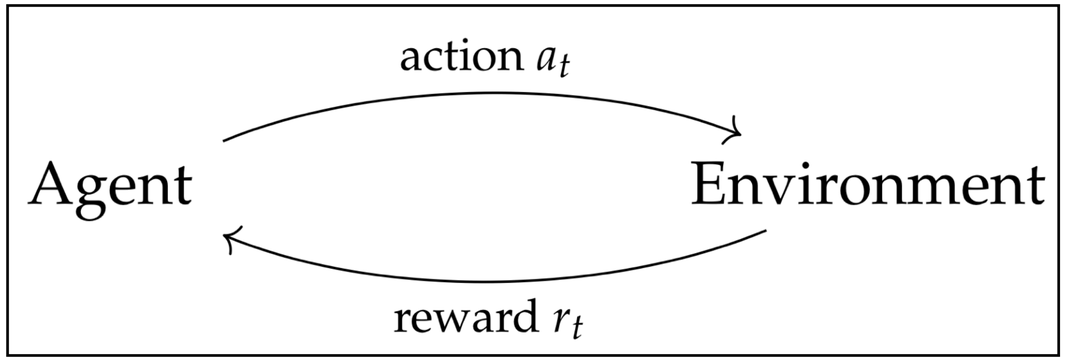


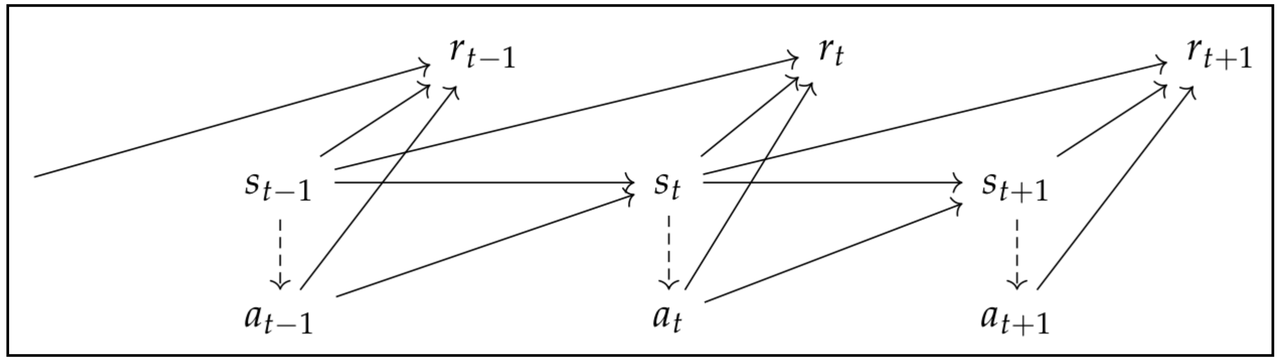
Selected Topics

Non-Myopic Designs

Shouldn't
we be (even) more strategic
and consider future
decisions?

Sequential Designs are actually (greedy) Markov Decision Processes in disguise!

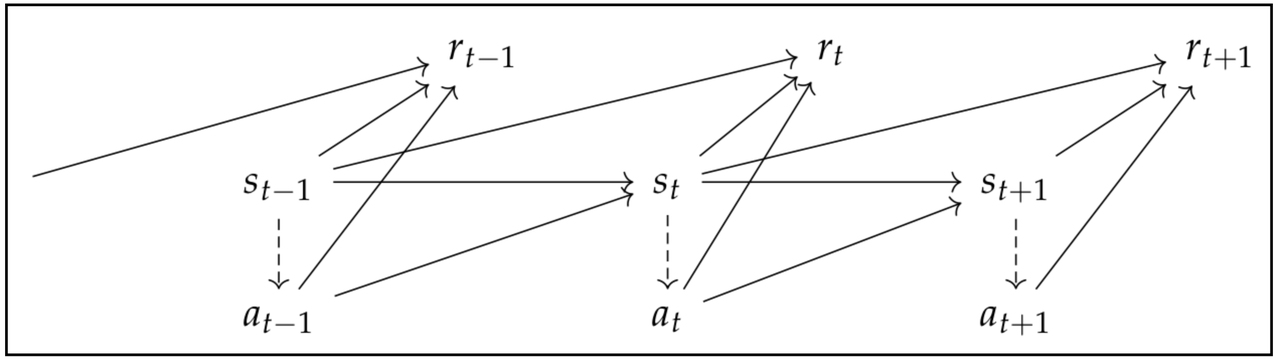




Selected Topics

Non-Myopic Designs

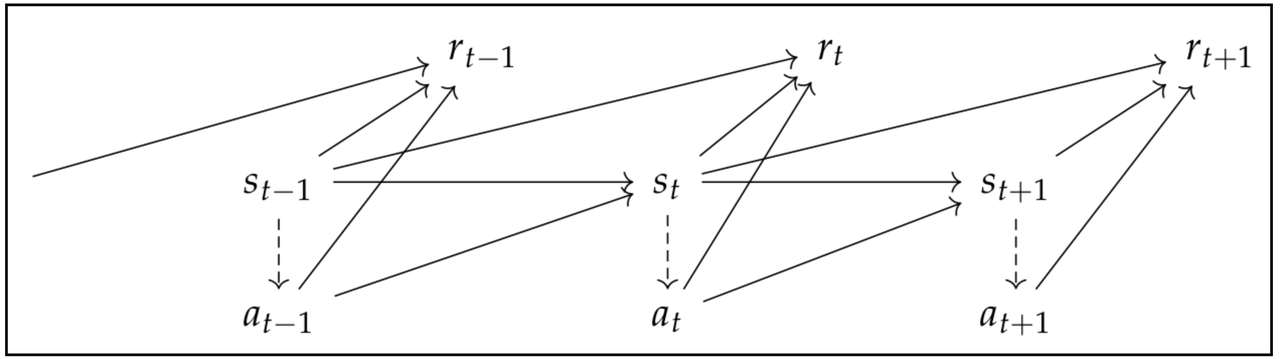
Shouldn't
we be (even) more strategic
and consider future
decisions?



Selected Topics

Non-Myopic Designs

Shouldn't
we be (even) more strategic
and consider future
decisions?

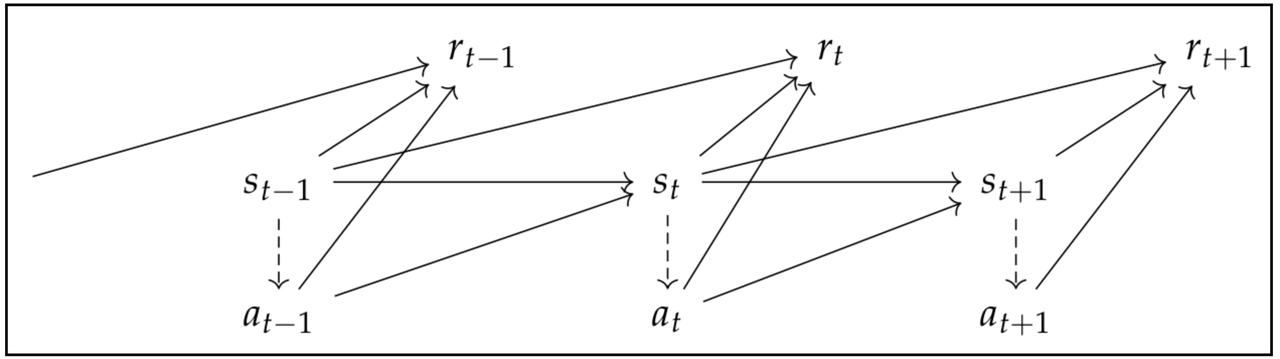


$$\pi_*(s_0) := \underset{\pi}{\operatorname{argmax}}\ \mathbb{E}_{s,a\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t r_t(s_t, a_t)\right]$$

Selected Topics

Non-Myopic Designs

Shouldn't
we be (even) more strategic
and consider future
decisions?



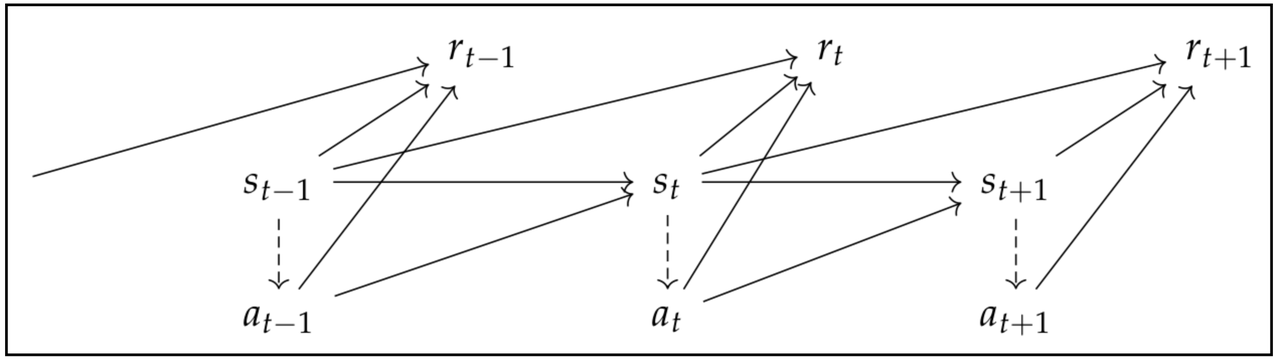
$$\pi_*(s_0) := \underset{\pi}{\operatorname{argmax}}\ \mathbb{E}_{s,a\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t r_t(s_t, a_t)\right]$$

full decision horizon!

Selected Topics

Non-Myopic Designs

Shouldn't
we be (even) more strategic
and consider future
decisions?



$$\pi_*(s_0) := \underset{\pi}{\operatorname{argmax}}\ \mathbb{E}_{s,a\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t r_t(s_t, a_t)\right]$$

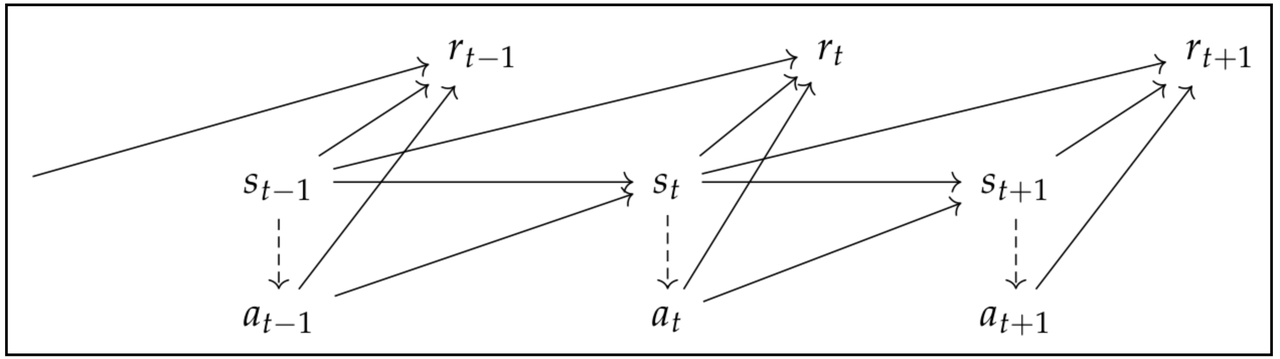
full decision horizon!

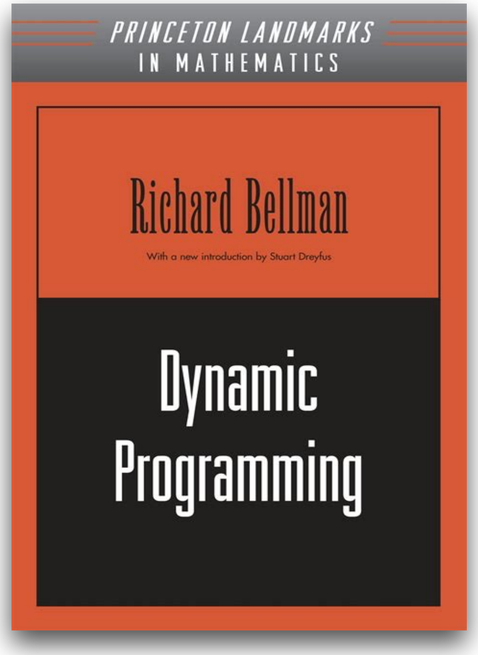
Well-studied objective in Dynamic Programming / Reinforcement Learning.

Selected Topics

Non-Myopic Designs

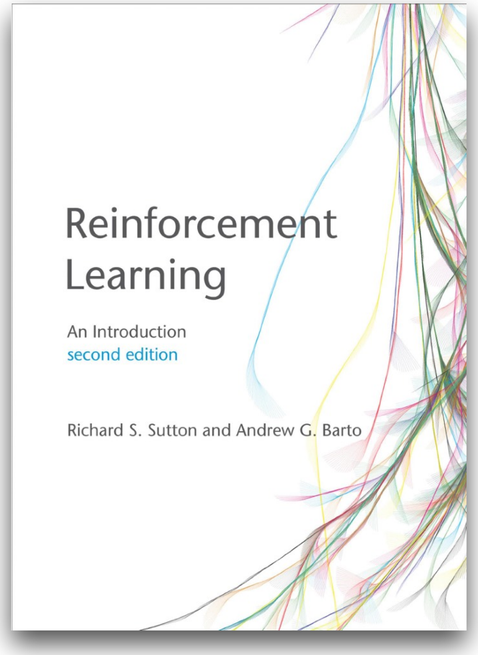
Shouldn't
we be (even) more strategic
and consider future
decisions?





$$\pi_*(s_0) := \underset{\pi}{\text{argmax}}\ \mathbb{E}_{s,a\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t r_t(s_t, a_t)\right]$$

full decision horizon!



Well-studied objective in Dynamic Programming / Reinforcement Learning.

Selected Topics

Non-Myopic Designs

Shouldn’t
we be (even) more strategic
and consider future
decisions?

Selected Topics

Non-Myopic Designs

Shouldn't
we be (even) more strategic
and consider future
decisions?

BINOCULARS for efficient, nonmyopic sequential experimental design

Shali Jiang, Henry Chai, Javier Gonzalez, Roman Garnett *Proceedings of the 37th International Conference on Machine Learning*, PMLR 119:4794-4803, 2020.

Abstract

Finite-horizon sequential experimental design (SED) arises naturally in many contexts, including hyperparameter tuning in machine learning among more traditional settings. Computing the optimal policy for such problems requires solving Bellman equations, which are generally intractable. Most existing work resorts to severely myopic approximations by limiting the decision horizon to only a single time-step, which can underweight exploration in favor of exploitation. We present BINOCULARS: Batch-Informed NOnmyopic Choices, Using Long-horizons for Adaptive, Rapid SED, a general framework for deriving efficient, nonmyopic approximations to the optimal experimental policy. Our key idea is simple and surprisingly effective: we first compute a one-step optimal batch of experiments, then select a single point from this batch to evaluate. We realize BINOCULARS for Bayesian optimization and Bayesian quadrature – two notable example problems with radically different objectives – and demonstrate that BINOCULARS significantly outperforms significantly outperforms myopic alternatives in real-world scenarios.

GLASSES: Relieving The Myopia Of Bayesian Optimisation

Javier Gonzalez, Michael Osborne, Neil Lawrence *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, PMLR 51:790-799, 2016.

Abstract

We present GLASSES: Global optimisation with Look-Ahead through Stochastic Simulation and Expected-loss Search. The majority of global optimisation approaches in use are myopic, in only considering the impact of the next function value; the non-myopic approaches that do exist are able to consider only a handful of future evaluations. Our novel algorithm, GLASSES, permits the consideration of dozens of evaluations into the future. This is done by approximating the ideal look-ahead loss function, which is expensive to evaluate, by a cheaper alternative in which the future steps of the algorithm are simulated beforehand. An Expectation Propagation algorithm is used to compute the expected value of the loss. We show that the far-horizon planning thus enabled leads to substantive performance gains in empirical tests.

Selected Topics

Non-Myopic Designs

Shouldn't
we be (even) more strategic
and consider future
decisions?

BINOCULARS for efficient, nonmyopic sequential experimental design

Shali Jiang, Henry Chai, Javier Gonzalez, Roman Garnett *Proceedings of the 37th International Conference on Machine Learning*, PMLR 119:4794-4803, 2020.

Abstract

Finite-horizon sequential experimental design (SED) arises naturally in many contexts, including hyperparameter tuning in machine learning among more traditional settings. Computing the optimal policy for such problems requires solving Bellman equations, which are generally intractable. Most existing work resorts to severely myopic approximations by limiting the decision horizon to only a single time-step, which can underweight exploration in favor of exploitation. We present BINOCULARS: Batch-Informed NOnmyopic Choices, Using Long-horizons for Adaptive, Rapid SED, a general framework for deriving efficient, nonmyopic approximations to the optimal experimental policy. Our key idea is simple and surprisingly effective: we first compute a one-step optimal batch of experiments, then select a single point from this batch to evaluate. We realize BINOCULARS for Bayesian optimization and Bayesian quadrature – two notable example problems with radically different objectives – and demonstrate that BINOCULARS significantly outperforms significantly outperforms myopic alternatives in real-world scenarios.

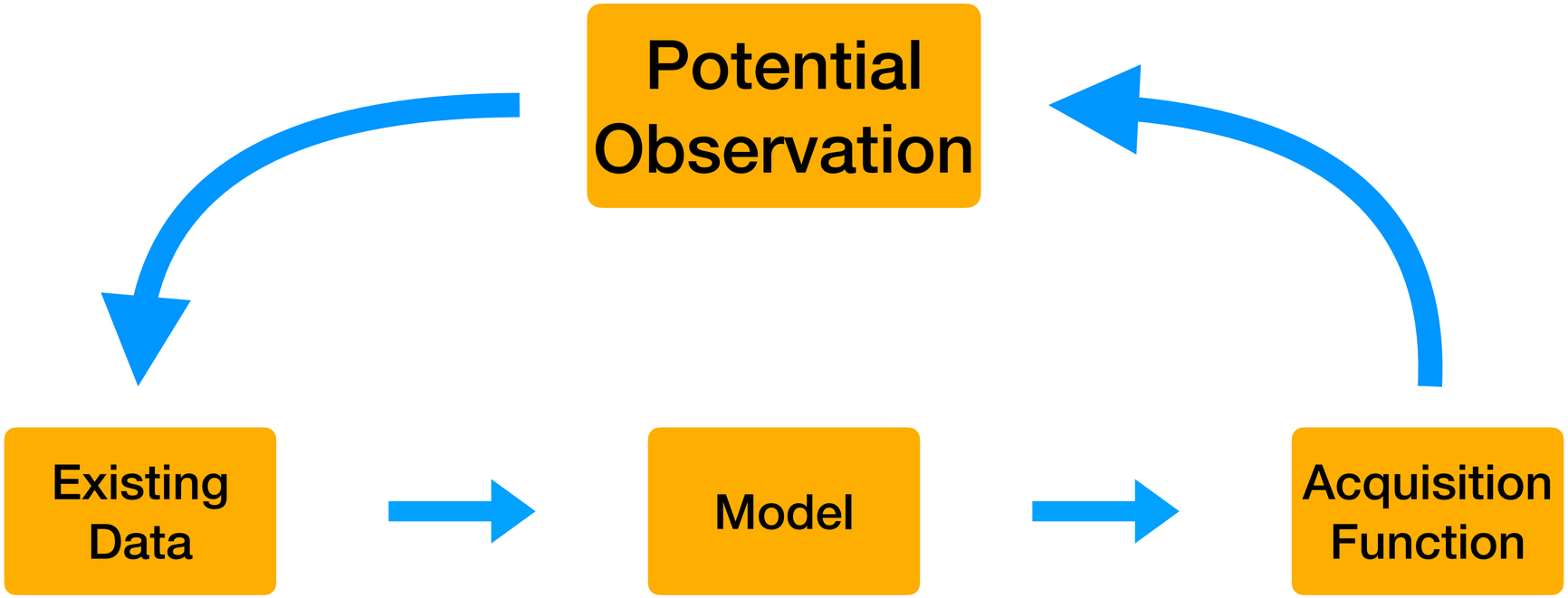
GLASSES: Relieving The Myopia Of Bayesian Optimisation

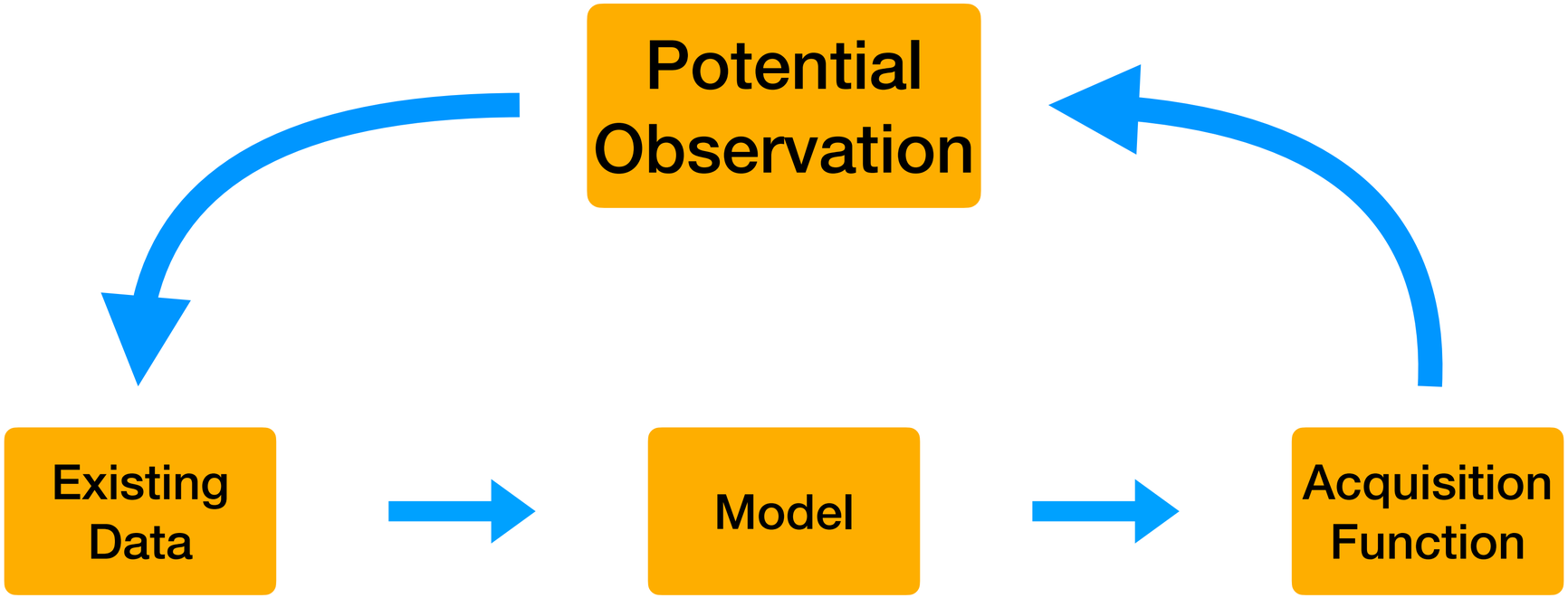
Javier Gonzalez, Michael Osborne, Neil Lawrence *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, PMLR 51:790-799, 2016.

Abstract

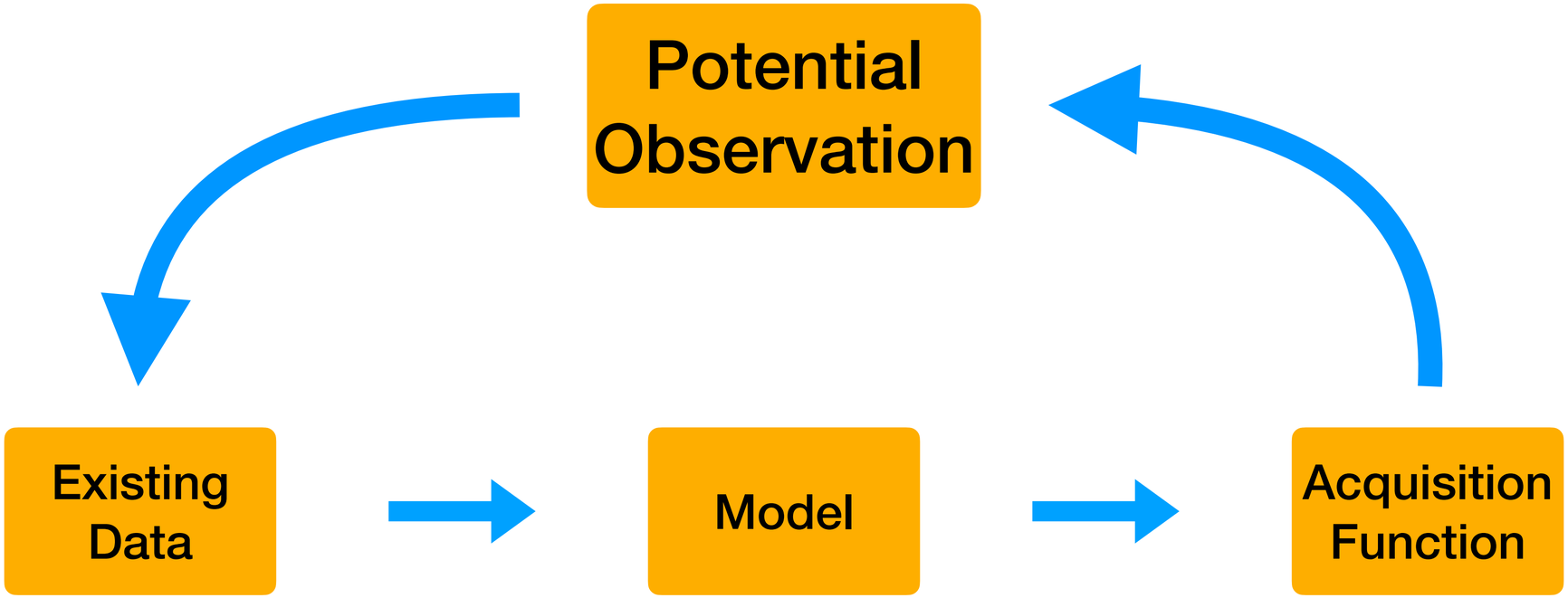
We present GLASSES: Global optimisation with Look-Ahead through Stochastic Simulation and Expected-loss Search. The majority of global optimisation approaches in use are myopic, in only considering the impact of the next function value; the non-myopic approaches that do exist are able to consider only a handful of future evaluations. Our novel algorithm, GLASSES, permits the consideration of dozens of evaluations into the future. This is done by approximating the ideal look-ahead loss function, which is expensive to evaluate, by a cheaper alternative in which the future steps of the algorithm are simulated beforehand. An Expectation Propagation algorithm is used to compute the expected value of the loss. We show that the far-horizon planning thus enabled leads to substantive performance gains in empirical tests.

Construct simulated 'roll-outs' for the next couple of decisions to reduce greediness.


Potential Observation
Existing Data
Model
Acquisition Function

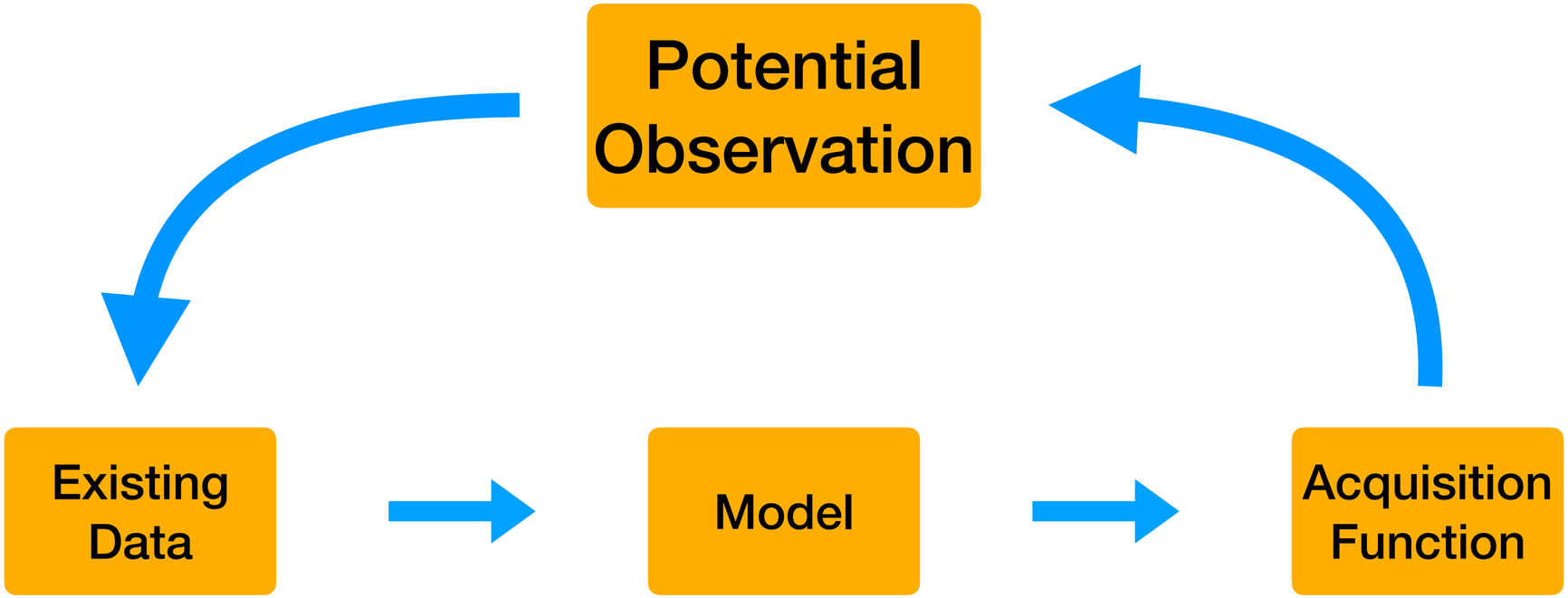


Open Questions



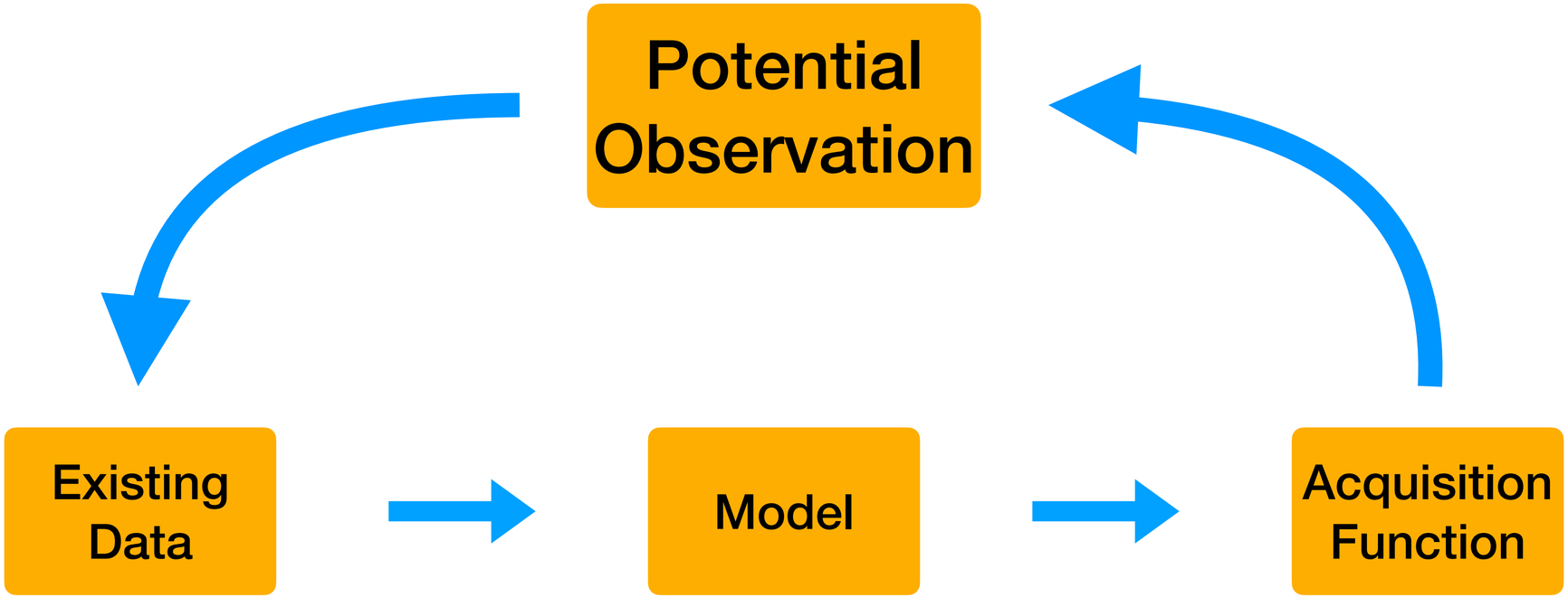
Open Questions

- What if our model is misspecified?



Open Questions

- What if our model is misspecified?
- How to make the acquisition optimisation fast?



Open Questions

- What if our model is misspecified?
- How to make the acquisition optimisation fast?
- How would adaptively collected data impact inference?

References

References

Gaussian Process

General

- Williams, C. K., & Rasmussen, C. E. (2006). Gaussian processes for machine learning. Cambridge, MA: MIT press.

Scalability

- Leibfried, F., Dutordoir, V., John, S. T., & Durrande, N. (2020). A tutorial on sparse Gaussian processes and variational inference. arXiv preprint arXiv:2012.13962.
- Katzfuss, M., & Guinness, J. (2021). A general framework for Vecchia approximations of Gaussian processes. Statistical Science, 36(1), 124-141.
- Rahimi, A., & Recht, B. (2007). Random features for large-scale kernel machines. Advances in neural information processing systems, 20.
- Särkkä, S., Solin, A., & Hartikainen, J. (2013). Spatiotemporal learning via infinite-dimensional Bayesian filtering and smoothing: A look at Gaussian process regression through Kalman filtering. IEEE Signal Processing Magazine, 30(4), 51-61.
- Lindgren, F., Rue, H., & Lindström, J. (2011). An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. Journal of the Royal Statistical Society Series B: Statistical Methodology, 73(4), 423-498.

References

Experimental Design

General

- Gramacy, R. B. (2020). Surrogates: Gaussian process modeling, design, and optimization for the applied sciences. Chapman and Hall/CRC.
- Krause, A., & Hübotter, J. (2025). Probabilistic artificial intelligence. arXiv preprint arXiv:2502.05244.
- Rainforth, Tom, Adam Foster, Desi R Ivanova, and Freddie Bickford Smith. 2024. “Modern Bayesian Experimental Design.” Statistical Science 39 (1): 100–114.
- Lindley, D. V. (1956). On a measure of the information provided by an experiment. The Annals of Mathematical Statistics, 27(4), 986-1005.

Selected Topics

- Zhang, R. Y., Moss, H. B., Astfalck, L., Cripps, E., & Leslie, D. S. (2026). BALLAST: Bayesian Active Learning with Look-ahead Amendment for Sea-drifter Trajectories under Spatio-Temporal Vector Fields. ICML.
- Qing, J., Knudde, N., Couckuyt, I., Dhaene, T., & Shintani, K. (2020, December). Batch Bayesian active learning for feasible region identification by local penalization. In 2020 Winter Simulation Conference (WSC) (pp. 2779-2790). IEEE.
- González, J., Osborne, M., & Lawrence, N. (2016, May). GLASSES: Relieving the myopia of Bayesian optimisation. In Artificial Intelligence and Statistics (pp. 790-799). PMLR.
- Jiang, S., Chai, H., Gonzalez, J., & Garnett, R. (2020, November). BINOCULARS for efficient, nonmyopic sequential experimental design. In International Conference on Machine Learning (pp. 4794-4803). PMLR.