Machine Learning and Financial Applications

Lecture 8
Improving trading strategies with Bayesian optimization

Liu Peng
liupeng@smu.edu.sg

Video tutorial

- Simple and exponential smoothing averages https://youtu.be/nRNa0eElb50
- Moving averages and trend following strategy https://youtu.be/qD4sgAHB0hY
- Implementing the trend following strategy https://youtu.be/haB-LkS8EsA
- Bayesian optimization https://youtu.be/luQLG3ZEtYc
- Gaussian process and acquisition function https://youtu.be/laziyKslmQY
- Improving pairs trading strategy using Bayesian optimization https://youtu.be/621QTVIkUfo

Introducing trend trading

- Trend trading is a type of trading strategy that captures gains by analyzing an asset's momentum in a particular direction.
- The trend refers to an asset price that is moving in one overall direction, such as up or down. The momentum refers to the capacity for the asset’s price trend to sustain itself going forward.
- The trend-following strategy is designed to take advantage of forward-looking uptrends with new highs or anticipated downtrends with new lows.

Understanding technical indicators

- Technical indicators are mathematical calculations based on historical price (high, low, open, close, etc.) or volume, and can be used to determine entry and exit points for trades.
- Technical indicators are highly security-dependent: what can be a good technical indicator for a particular security might not hold the case for the other.
- Technical indicators help confirm if the market is following a trend or in a range-bound situation, oscillating within a price range.

Working with moving averages

- Moving average, also called rolling average, is the mean or average of the specified data field (e.g. daily closing price) for a given set of consecutive periods.
- As new data becomes available, the mean of the data is computed by dropping the oldest value and adding the latest one. It is rolling along with the data, hence the name “Moving Average”.
- When working with time series data such as daily stock price, the averaging effect can also be considered as smoothening the time series, reducing short-term fluctuations and temporary variations in the data.

Simple moving average (SMA)

- The simple moving average SMA_t at time t is defined as follows:

$$\mathrm{SMA}_t = \frac{S_{t-(M-1)} + \cdots + S_{t-1} + S_t}{M}$$

- In other words, to calculate SMA_t, we would take M historical price points, including the current period, and then take the average of these M price points.

- The simple moving average (SMA) is the unweighted mean of the previous M price points. Here, M is a user-defined input. It depends on the amount of smoothing desired, since increasing the value of M leads to a smoother curve, while a smaller M reduces the smoothness.

Simple moving average (SMA)

Calculating the simple moving average

$$\mathrm{SMA}_t = \frac{S_{t-(M-1)} + \cdots + S_{t-1} + S_t}{M}$$

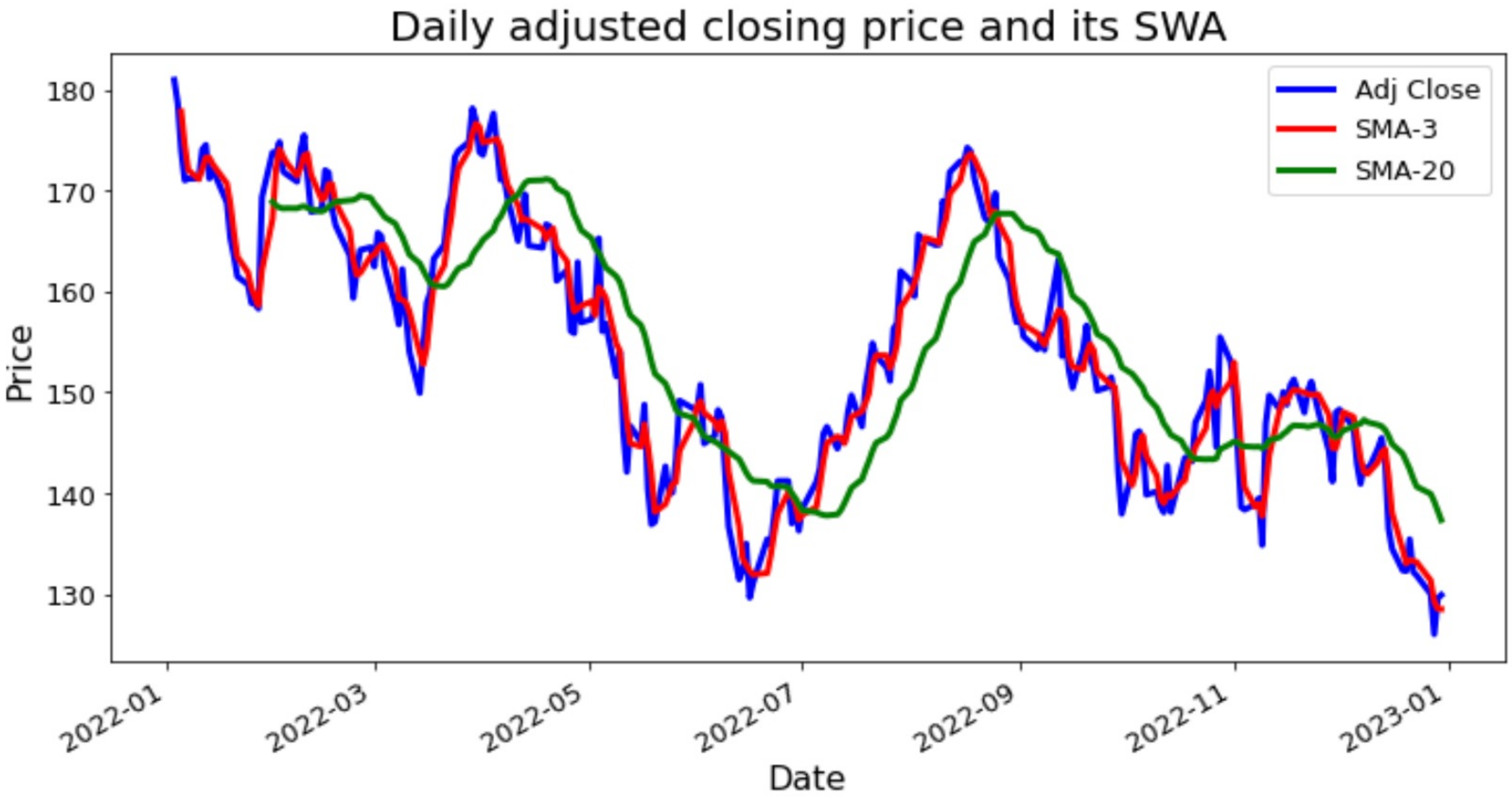
Created using .rolling(window_size).mean()

Date	Adj Close	SMA-3
2022-01-03 00:00:00-05:00	180.959747	NaN
2022-01-04 00:00:00-05:00	178.663086	NaN
2022-01-05 00:00:00-05:00	173.910645	177.844493
2022-01-06 00:00:00-05:00	171.007523	174.527084
2022-01-07 00:00:00-05:00	171.176529	172.031565

Empty due to incomplete values in the rolling window

$$177.844493=\frac{180.959747+178.663086+173.910645}{3}$$

Simple moving average (SMA)

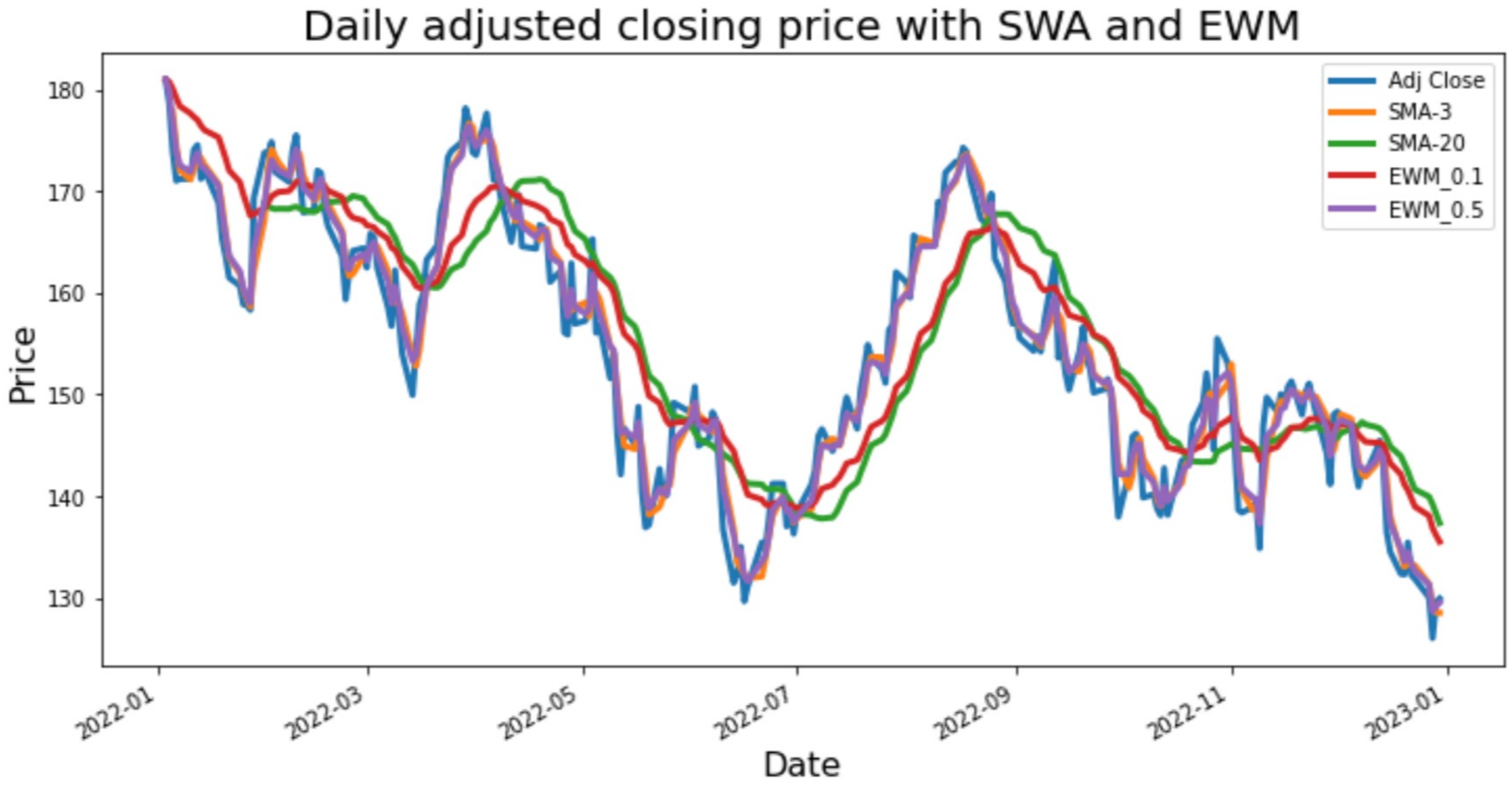


Exponential moving averages (EMA)

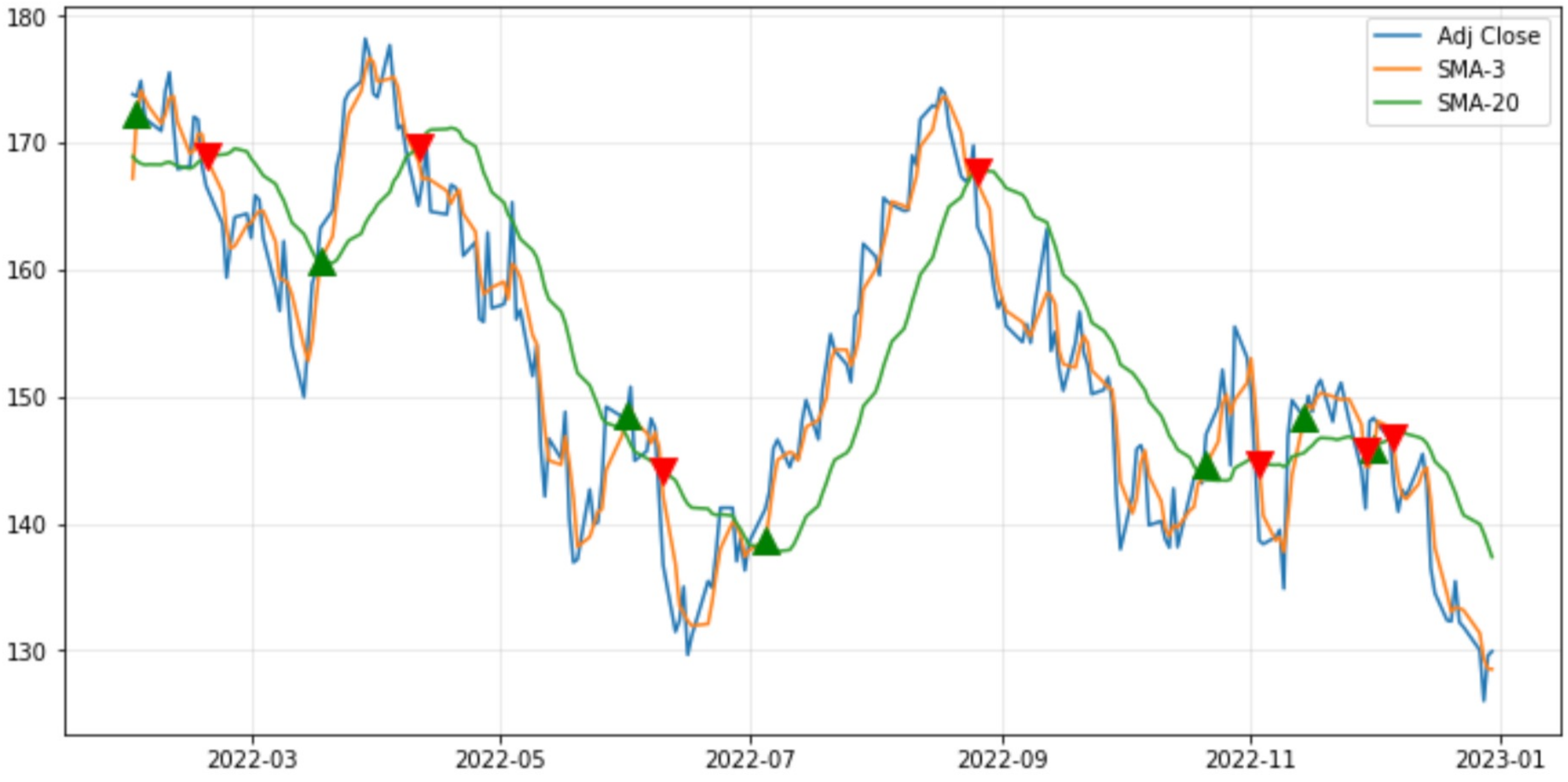
- The exponential moving average (EMA) is a widely used method to reduce the noise in the data and identify long-term trends.
- EMA assumes that recent data is more relevant than old data. Such an assumption has its merit, since EMA can react faster to changes and is thus more sensitive to recent movements as compared to the simple moving average.
- This also means that there is no window size to be specified by the function, since all historical data points are in use.

$$\mathrm{EWMA}_t = \begin{cases} S_0, & t = 0 \\ \alpha S_t + (1 - \alpha) EWMA_{t-1}, & t > 0 \end{cases}$$

Exponential moving averages (EMA)



Implementing the trend-following strategy



Implementing the trend-following strategy

Figure 5.12 Comparing the cumulative return of buy-and-hold and trend-following strategies for one share of Apple's stock.

Lab session: implementing trend following strategy

Exploration vs. Exploitation

Optimizing trading strategies

- Since different testing periods likely exhibit different characteristics in terms of the asset price curve, a robust approach is to backtest a specific set of parameters over different test periods that cover most representative scenarios.

- However, manually fine-tuning a trading strategy by setting different parameter values is an extremely time-consuming process. On the one hand, the number of possible parameter values to test out may simply be too large.

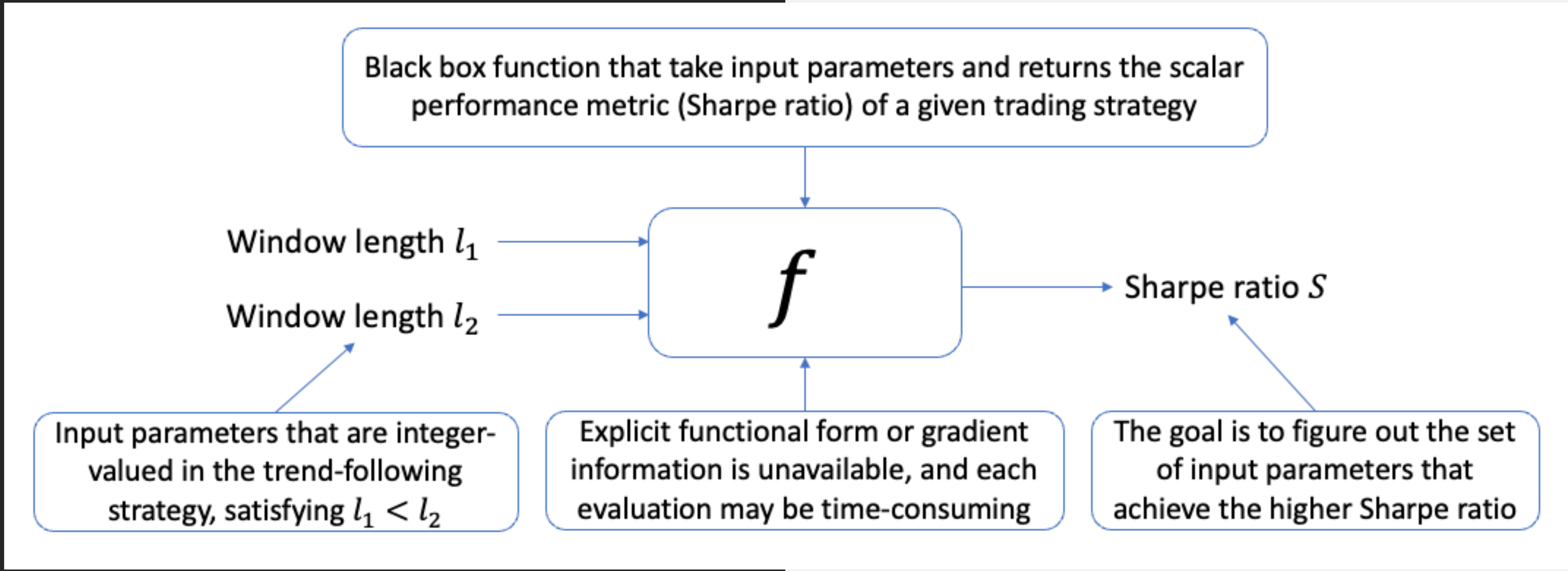
- On the other hand, backtesting each specific set of parameters is not instantaneous. Instead, each round of execution may take very long, thus further exacerbating the challenge in the global search for the optimal strategy.

Parametric trading strategies

- The parameters serve as the input variables to a specific trading strategy. A typical trading strategy has one or more parameters, each assuming a particular value within the pre-specified range.

- Example: the trend following strategy covered earlier. This trading strategy relies on two moving averages to generate a trading signal: a short-term moving average and a long-term moving average.

- This results in an objective function, where the output is the Sharpe ratio S over a specific backtesting period, the input parameters are window lengths l_1 and l_2, and we can represent the objective function as $S = f(l_1, l_2)$. Here, f represents a black-box function, which means we do not have its explicit mathematical form or its derivative information.



The optimization problem

- The selected trading strategy manifests as an unknown function, and our goal is to search for the optimal set of window lengths that deliver the highest performance metric, the Sharpe ratio in this case.

Global optimization

- Optimization aims to locate the optimal set of parameters of interest across the whole search domain by carefully allocating limited resources.
- The optimization procedure tries to locate the global maximum f^* or its corresponding location x^* in a principled and systematic manner. Mathematically, we wish to locate f^* where

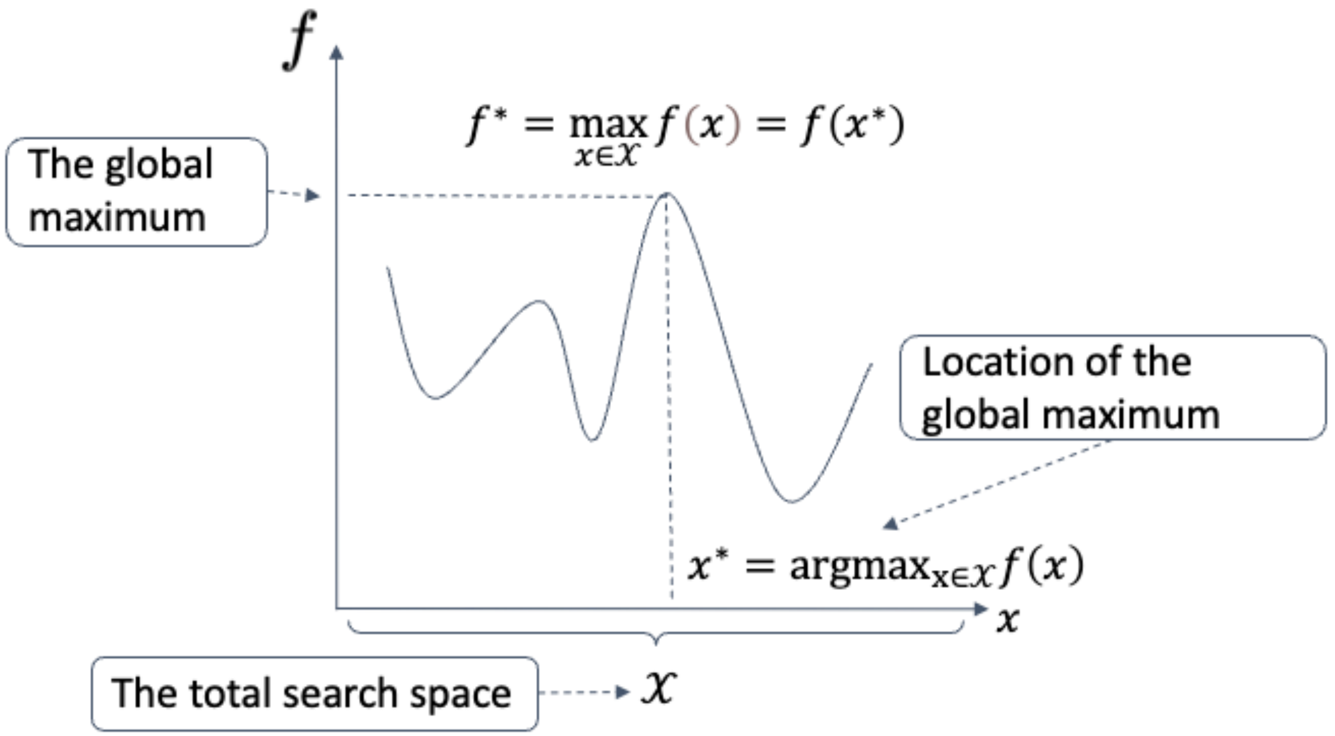
$$f^* = \max_{x \in \mathcal{X}} f(x) = f(x^*)$$

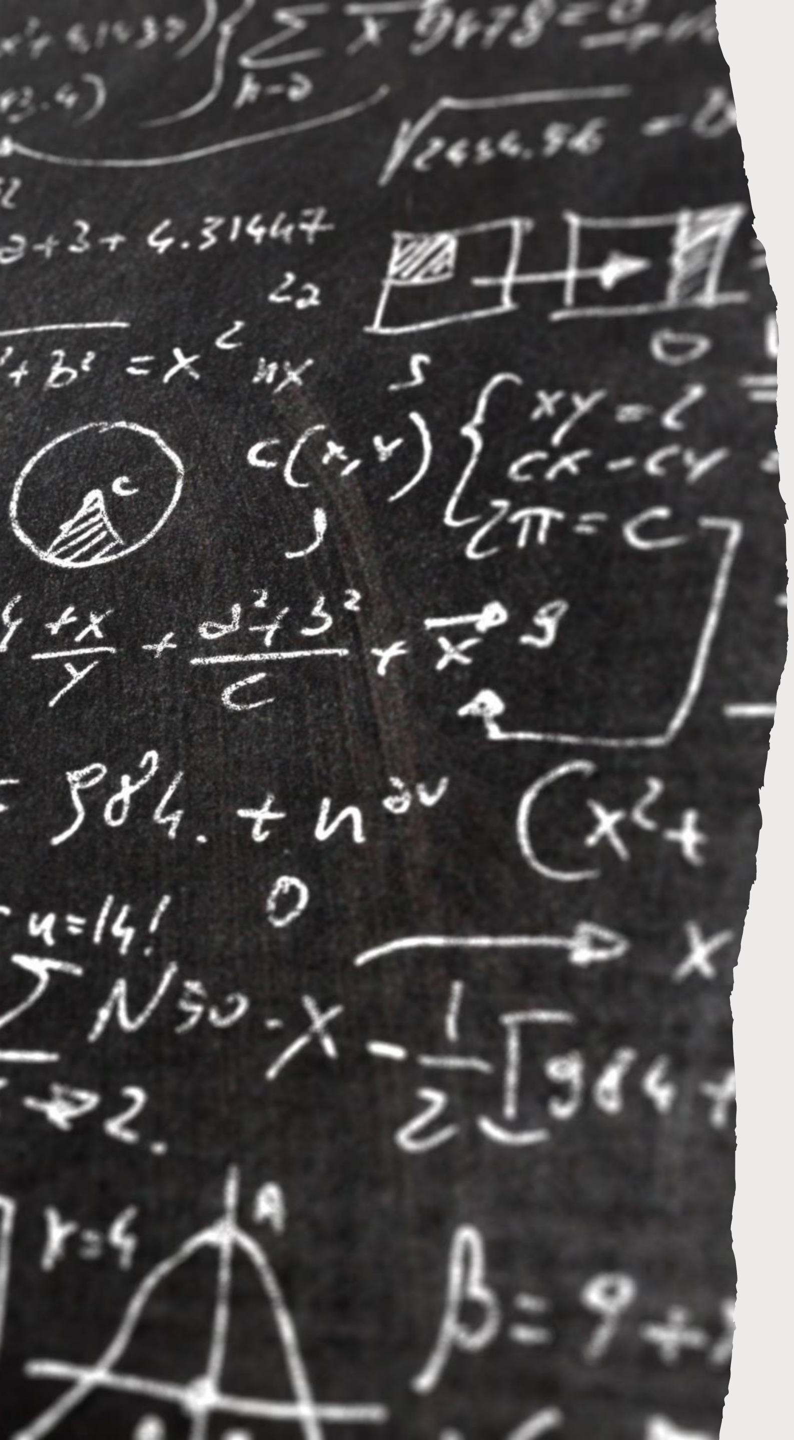
- Or equivalently, we are interested in its location x^* where

$$x^* = \mathrm{argmax}_{x \in \mathcal{X}} f(x)$$

Global optimization

- An example objective function with the global maximum f^* and its location x^*. The goal of global optimization is to systematically reason about a series of sampling decisions so as to locate the global maximum as fast as possible.



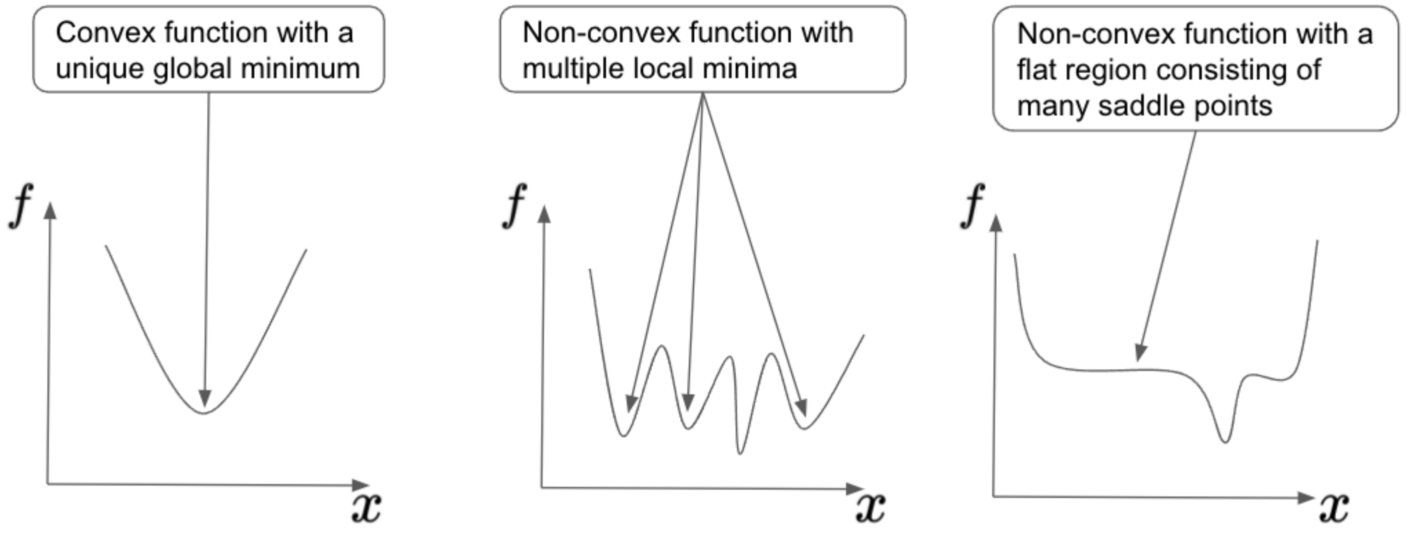


Objective function

- The objective function governs how the quantity of interest is generated.
- For the specific type of objective functions that govern the performance of trading strategies, we summarize the following common attributes:
  - We do not have access to the explicit expression of the objective function, making it a “black box” function.
  - The returned value by probing at a specific input parameter value is highly sensitive to the choice of backtesting period.
  - Each functional evaluation is costly, thus ruling out the option for an exhaustive probing exercise.
  - We do not have access to its gradient.

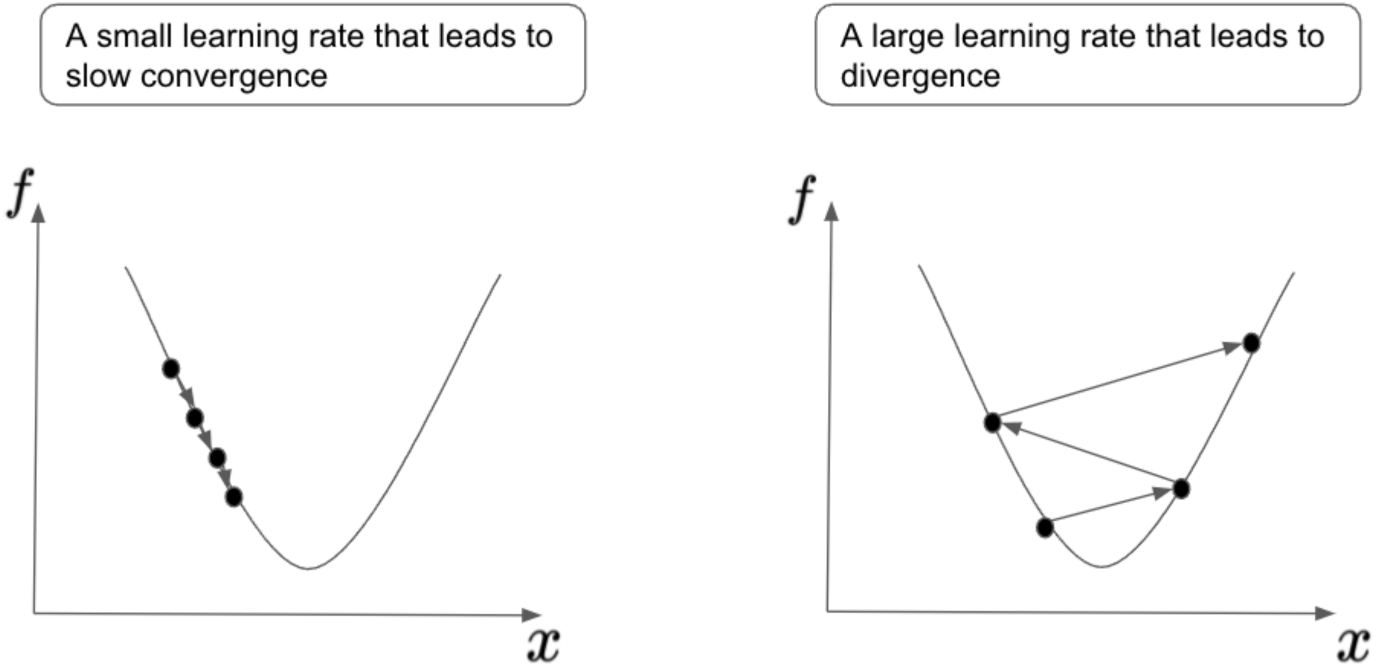
Objective function

Three possible functional forms. On the left is a convex function whose optimization is easy. In the middle is a nonconvex function with multiple local minina, and on the right is also a nonconvex function with a wide flat region full of saddle points. Optimization for the latter two cases takes a lot more work than for the first case.



Hyperparameter tuning

- Slow convergence due to a small learning rate on the left and divergence due to a large learning rate on the right.



Bayesian optimization

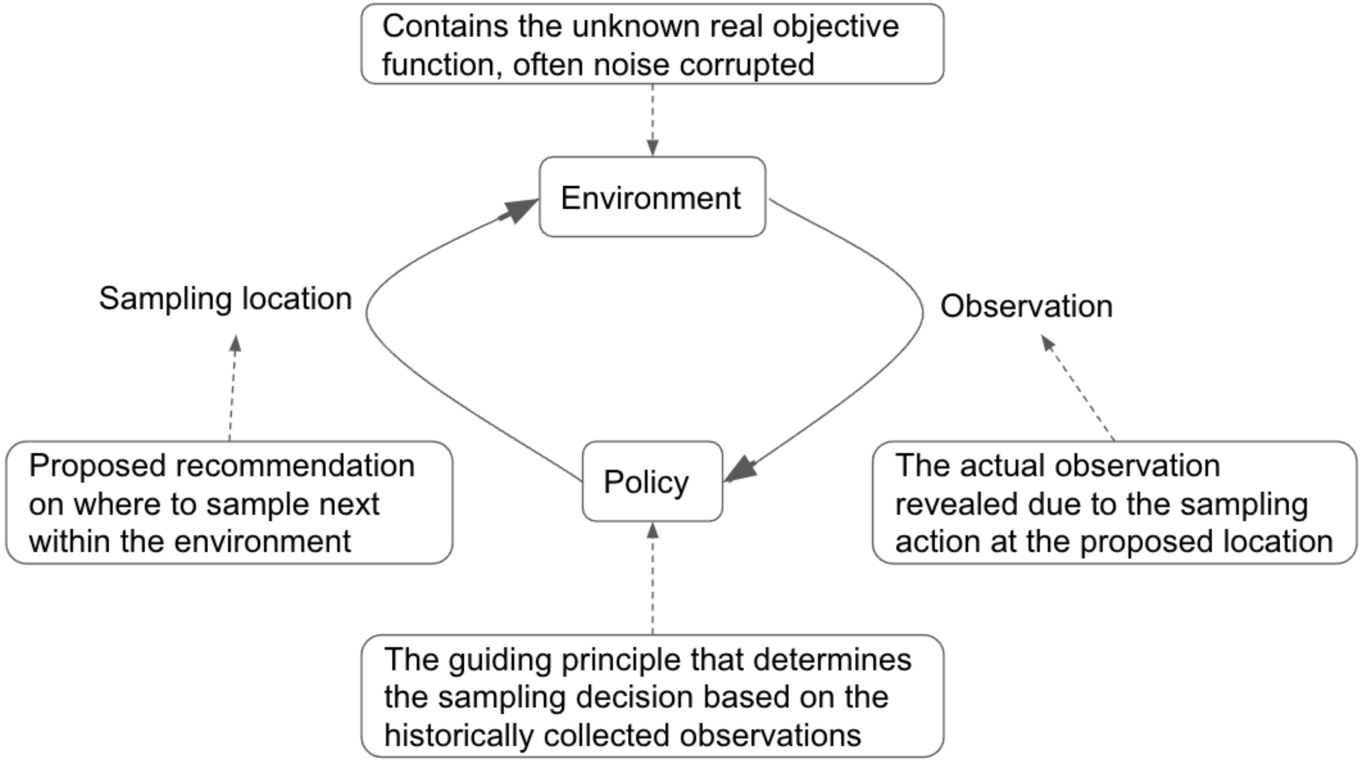
- Bayesian optimization is an area that studies optimization problems using the Bayesian approach.

- Optimization aims at locating the optimal objective value (i.e., a global maximum or minimum) of all possible values or the corresponding location of the optimum over the search domain, also called the environment.

- The search process starts at a specific initial location and follows a particular policy to iteratively guide the following sampling locations, collect new observations, and refresh the guiding search policy.

Bayesian optimization

- The overall bayesian optimization process. The policy digests the historical observations and proposes a new sampling location. The environment governs how the (possibly noise-corrupted) observation at the newly proposed location is revealed to the policy. Our goal is to learn an efficient and effective policy that could navigate toward the global optimum as quickly as possible.

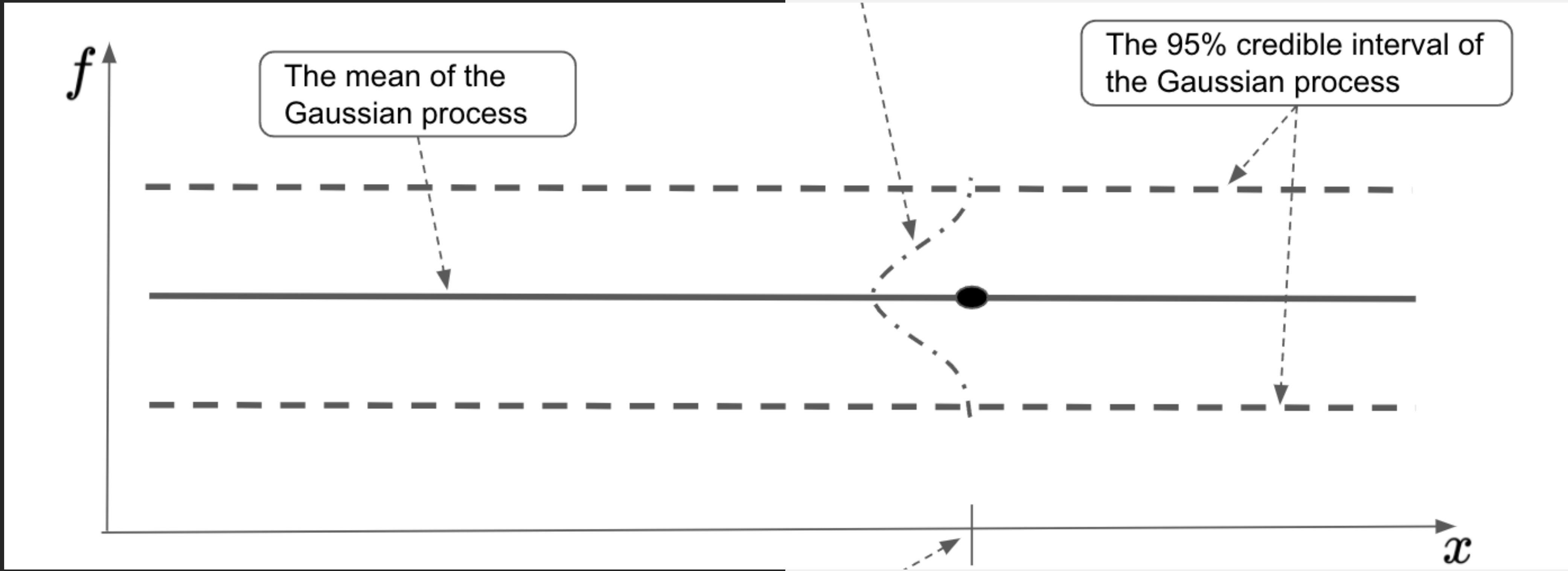


Gaussian process

- As a widely used stochastic process (able to model an unknown black-box function and the corresponding uncertainties of modeling), the Gaussian process takes the finite-dimensional probability distributions one step further into a continuous search domain that contains an infinite number of variables

- Any finite set of points in the domain jointly forms a multivariate Gaussian distribution.

- It is a flexible framework to model a broad family of functions and quantify their uncertainties, thus being a powerful surrogate model used to approximate the true underlying function.

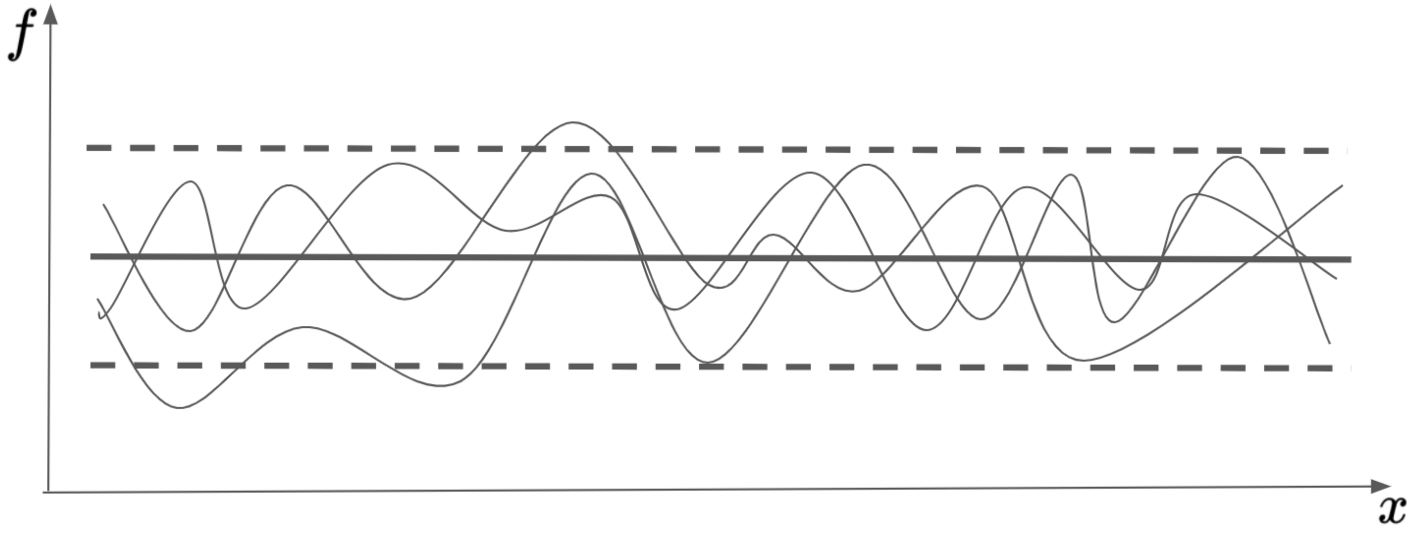


Gaussian process

A sample prior belief of the Gaussian process represented by the mean and 95% credible interval for each location in the domain. Every objective value is modeled by a random variable that follows a normal prior predictive distribution. Collecting the distributions of all random variables and updating these distributions as more observations are collected could help us quantify the potential shape of the true underlying function and its probability.

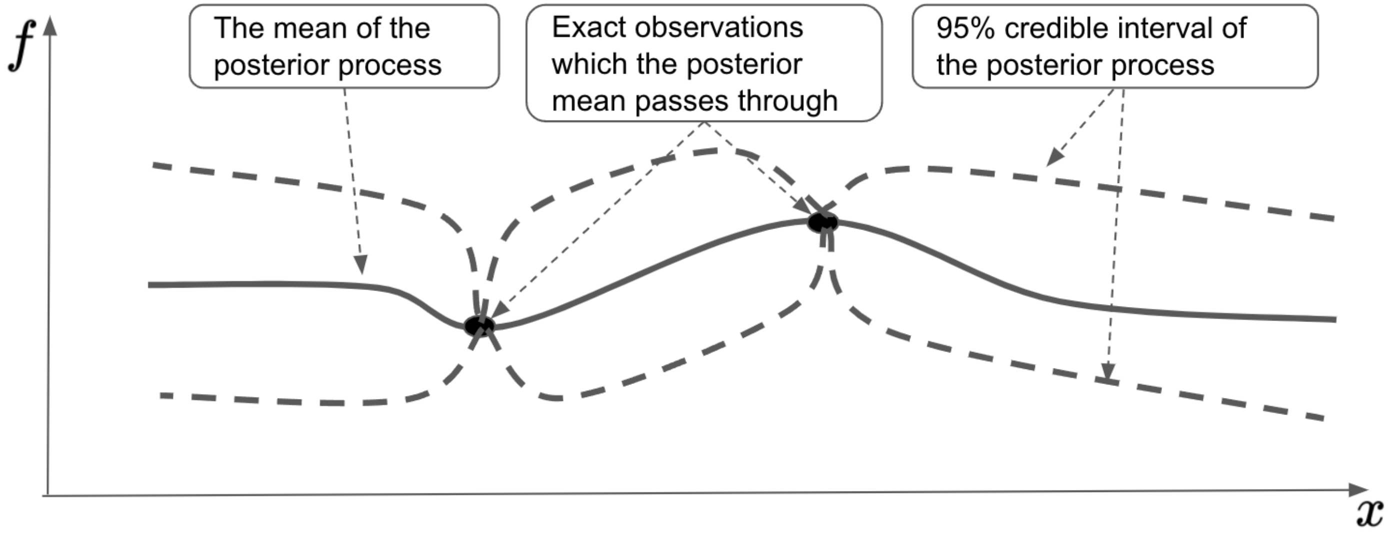
Gaussian process

- Three example functions sampled from the prior process, where the majority of the functions fall within the 95% credible interval.



Gaussian process (cont'd)

- Updated posterior process after incorporating two exact observations in the Gaussian process. The posterior mean interpolates through the observations, and the associated variance reduces as we move nearer the observations.



Gaussian process

- Mathematically, for a new sampling location $\mathbf{x}_* \in \mathcal{X}$, the corresponding functional evaluation f_* following the Gaussian process would assume a conditional normal distribution:

$$p(f_*; \mathbf{x}_*, D_n) = N(f_* | \mu_*, \sigma_*^2)$$

- $D_n = \{(\mathbf{x}_i, \mathbf{f}_i)\}_{i=1}^{n}$ contains the historical observed in pairs of sampling locations and scalar observations. The closed form of the posterior mean and variance functions can be derived by invoking the multivariate Gaussian theorem, giving:

$$\mu_* = \mathbf{k}(\mathbf{x}_{1:n}, \mathbf{x}_*)\mathbf{K}(\mathbf{x}_{1:n}, \mathbf{x}_{1:n})^{-1}\mathbf{f}_{1:n}$$
$$\sigma_*^2 = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}(\mathbf{x}_{1:n}, \mathbf{x}_*)\mathbf{K}(\mathbf{x}_{1:n}, \mathbf{x}_{1:n})^{-1}\mathbf{k}(\mathbf{x}_{1:n}, \mathbf{x}_*)$$

Acquisition function

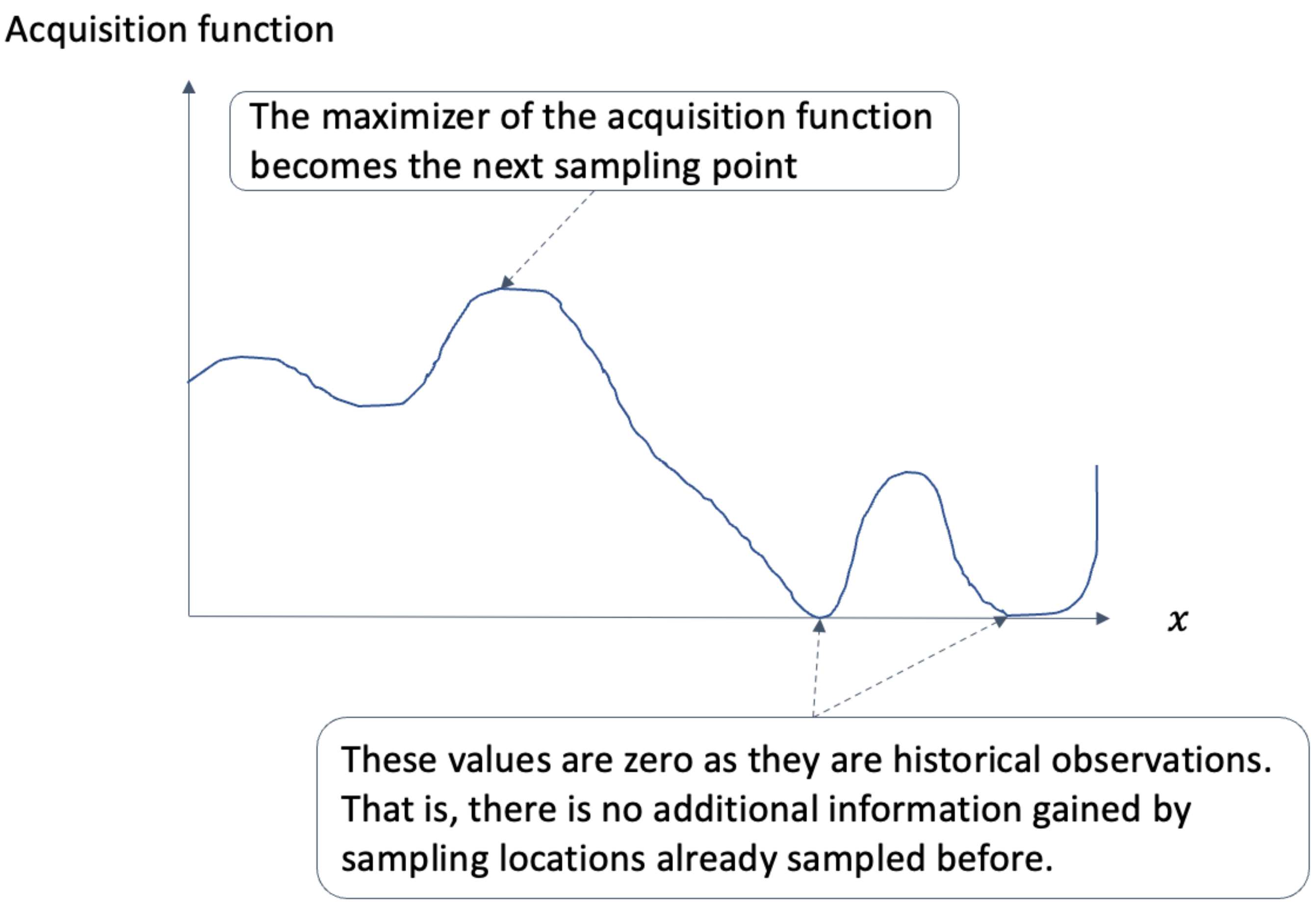
- We need to build a policy (by maximizing the acquisition function) that absorbs the most updated information on the objective function and recommends the following most-promising sampling location in the face of uncertainties across the domain.

- An acquisition function is a manually designed mechanism that evaluates the relative potential of each candidate location in the form of a scalar score, and the location with the maximum score will be used as the next sampling choice.

- The acquisition function is also cheap to evaluate as a side computation since we need to evaluate it at every candidate location and then locate the maximum utility score, posing another (inner) optimization problem.

Acquisition function

Illustrating a sample acquisition function curve. The location that corresponds to the highest value of the acquisition function is the next location (parameter value of a trading strategy) to sample. Since there is no value-added if we were to sample those locations already sampled earlier, the acquisition function thus reports zero at these locations.



Expected improvement

- The expected improvement chooses the historical maximum of the observed value as the benchmark for comparison upon selecting an additional sampling location.
- It also implicitly assumes that only one more additional sampling is left before the optimization process terminates.
- The expected marginal gain in utility (i.e., the acquisition function) becomes the expected improvement in the maximal observation, calculated as the expected difference between the observed maximum and the new observation after the additional sampling at an arbitrary sampling location.

Expected improvement

- $u(\mathcal{D}_{n+1}) - u(\mathcal{D}_n) = \max\{f_{n+1}, f_n^*\} - f_n^* = \max\{f_{n+1} - f_n^*, 0\}$

- $\alpha_{\mathrm{EI}}(x_{n+1}; \mathcal{D}_n) = \mathbb{E}[u(\mathcal{D}_{n+1}) - u(\mathcal{D}_n) | x_{n+1}, \mathcal{D}_n]$
$$= \int \max\{f_{n+1} - f_n^*, 0\} p(f_{n+1} | x_{n+1}, \mathcal{D}_n) df_{n+1}$$

- $\alpha_{\mathrm{EI}}(x_{n+1}; \mathcal{D}_n) = (\mu_{n+1} - f_n^*)\Phi\left(\frac{\mu_{n+1} - f_n^*}{\sigma_{n+1}}\right) + \sigma_{n+1}\phi(\frac{\mu_{n+1} - f_n^*}{\sigma_{n+1}})$
- where f_n^* is the best-observed value so far, ϕ and Φ denote the probability and cumulative density function of a standard normal distribution at the tentative point x_{n+1}, respectively. μ_{n+1} and σ_{n+1} denote the posterior mean and standard deviation at x_{n+1}.

- The closed-form EI consists of two components: exploitation (the first term) and exploration (the second term). Exploitation means continuing sampling the neighborhood of the observed region with a high posterior mean, and exploration encourages sampling an unvisited area where the posterior uncertainty is high. The expected improvement acquisition function thus implicitly balances off these two opposing forces.

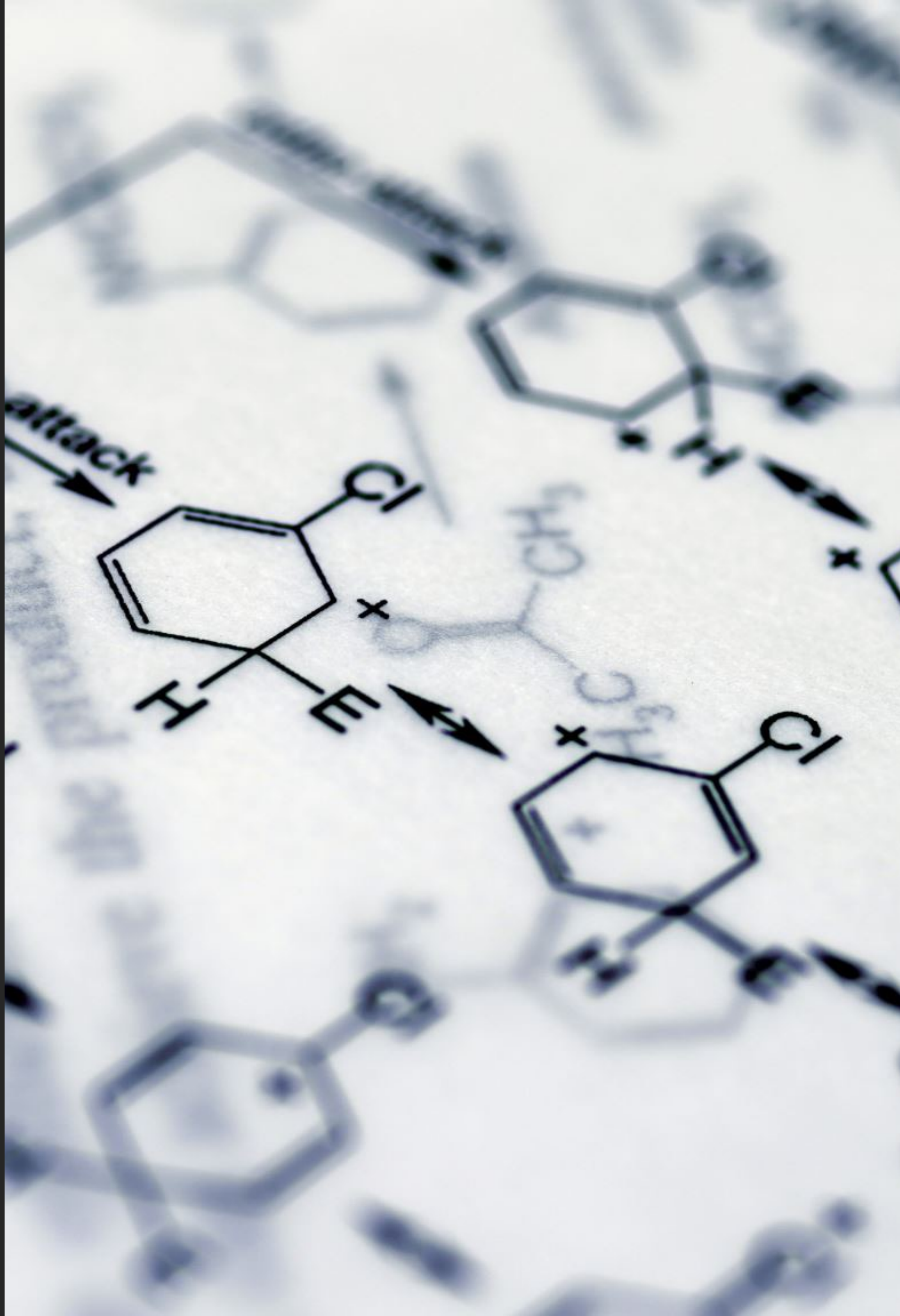
Upper confidence bound

- $\alpha_{\text{UCB}}(x_{n+1}; \mathcal{D}_n) = \mu_{n+1} + \beta_{n+1}\sigma_{n+1}$

- where β_{n+1} is a user-defined stage-wise hyper-parameter that controls the trade-off between the posterior mean and standard deviation. A low value of β_{n+1} encourages exploitation, and a high value of β_{n+1} leans more towards exploration.

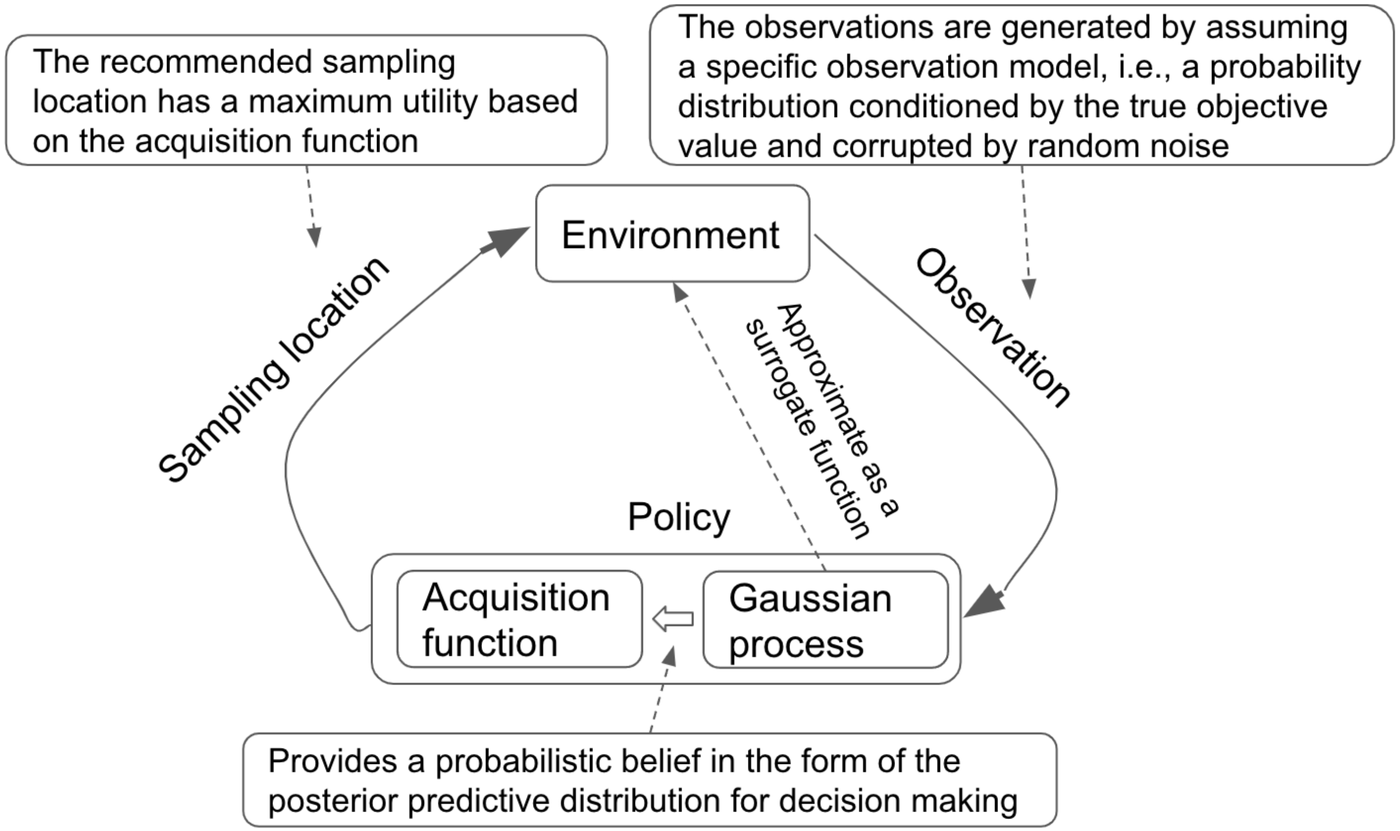
The full BO loop

- Bayesian optimization is an iterative process between the (uncontrolled) environment and the (controlled) policy.
- The policy involves two components supporting the sequential decision-making: a Gaussian process as the surrogate model to approximate the true underlying function (i.e., the environment), and an acquisition function to recommend the best sampling location.
- The environment receives the probing request at a specific location and responds by revealing a new observation that follows a particular observation model.
- The Gaussian process surrogate model then uses the new observation to obtain a posterior process in support of follow-up decision-making by the pre-set acquisition function.
- This process continues until the stopping criterion, such as exhausting a given budget, is met.



The full BO loop

- The full Bayesian optimization loop featuring an iterative interaction between the unknown (black-box) environment and the decision-making policy that consists of a Gaussian process for probabilistic evaluation and acquisition function for utility assessment of candidate locations in the environment.



Lab session: optimizing trading strategies using BO