BO from scratch

Global optimization: finding an input that results in the minimum or maximum cost of a given objective function.

The form of the objective function is complex and intractable to analyze and is often non-convex, nonlinear, high dimension, noisy, and computationally expensive to evaluate.

BO works by building a probabilistic model of the objective function, called the surrogate function,

that is then searched efficiently with an acquisition function

before candidate samples are chosen for evaluation on the real objective function.

Samples are comprised of one or more variables. One sample is often defined as a vector of variables with a predefined range in an n-dimensional space.

This space must be sampled and explored in order to find the specific combination of variable values that result in the best cost.

The objective function is often easy to specify but can be computationally challenging to calculate or result in a noisy calculation of cost over time. The **form of the objective function is unknown** and is often highly nonlinear, and highly multi-dimensional defined by the number of input variables.

Although little is known about the objective function, (it is known whether the minimum or the maximum cost from the function is sought), and as such, it is often referred to as a **black box function** and the search process as black box optimization.

A directed approach to global optimization that uses probability is called Bayesian Optimization. It uses [Bayes Theorem](https://machinelearningmastery.com/bayes-theorem-for-machine-learning/) to direct the search in order to find the minimum or maximum of an objective function.

* Posterior = Likelihood \* Prior

This provides a framework that can be used to quantify the beliefs about an unknown objective function

given samples from the domain and

their evaluation via the objective function.

1. We can devise specific samples (*x1, x2, …, xn*) and evaluate them using the objective function *f(xi)*
2. Samples and their outcome are collected sequentially
3. Define our data *D*, e.g. *D = {xi, f(xi), … xn, f(xn)}* and is used to define the prior.
4. The likelihood function is defined as the probability of observing the data given the function *P(D | f)*. This likelihood function will change as more observations are collected.

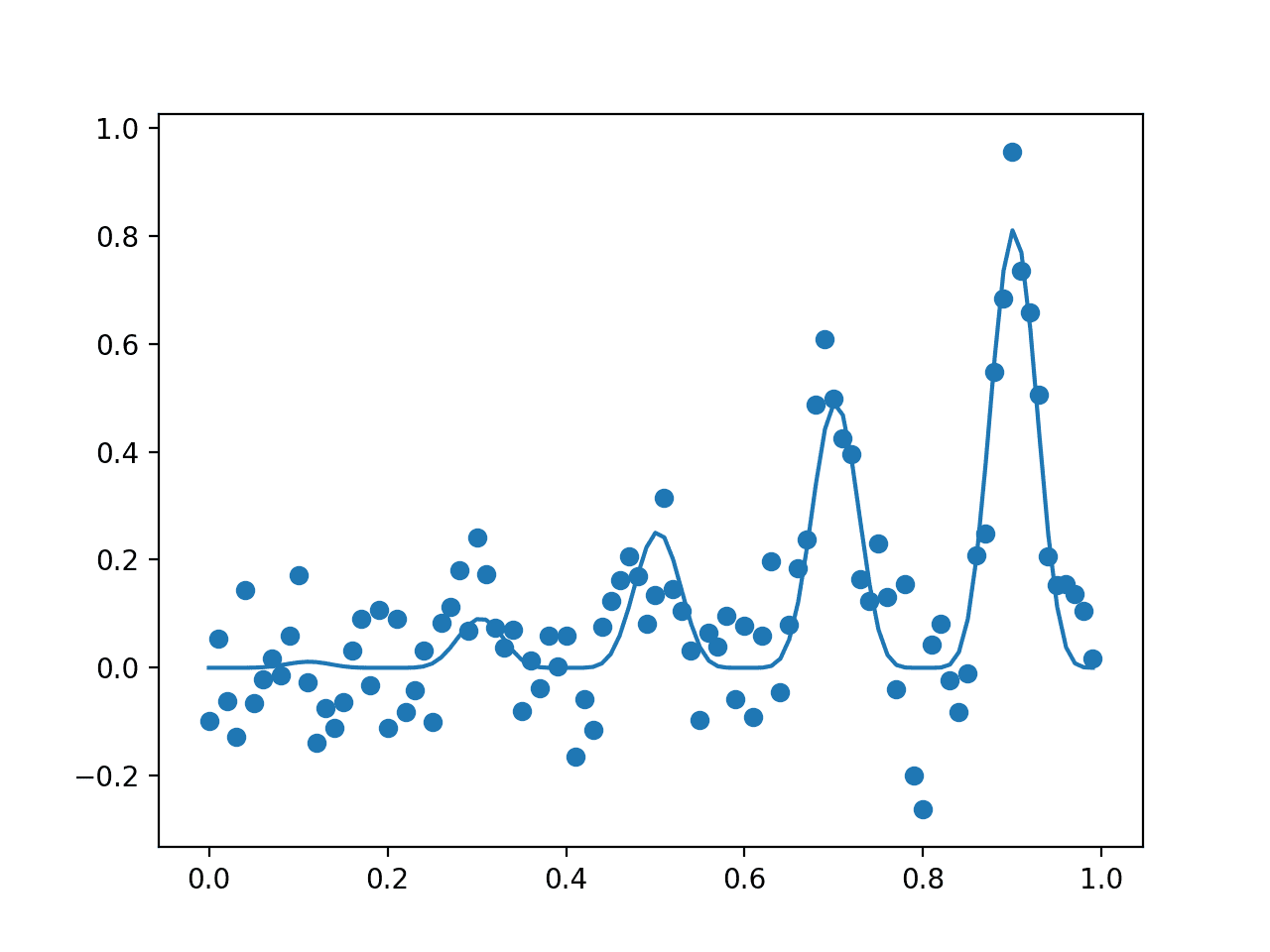
**P(f|D) = P(D|f) \* P(f)**

The posterior represents everything we know about the objective function.

It is an approximation of the objective function and can be used to estimate the cost of different candidate samples that we may want to evaluate.

posterior probability is a surrogate objective function.

**Test problem**

****

**Surrogate Function**: The surrogate function is a technique used to best approximate the mapping of input examples to an output score. The surrogate function is a technique used to best approximate the mapping of input examples to an output score.

Using a GP regression model is often preferred.

We can fit a GP regression model using the [GaussianProcessRegressor](https://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.GaussianProcessRegressor.html) scikit-learn implementation from a sample of inputs (*X*) and noisy evaluations from the objective function (*y*).

The surrogate function gives us an estimate of the objective function, used to direct future sampling. Sampling involves careful use of the posterior in a function known as the “*acquisition*” function.

**Acquisition Function**: Technique by which the posterior is used to select the next sample from the search space.