Azeotropy : Optimizer

Omkar Shirpure (22B0910)

March 9, 2024

Contents

1 Introduction

This problem statement is all about minimizing a function which is unknown based upon the values of the parameters which we need to optimize to get the minimum value. In our case the two variables were risk and cost for which the total cost of T which is not known.

There are multiple approaches but since in our case the constraints are the requests are costly and we need to minimize them as much as possible and find the minimum value for T.

Several approaches can be considered for this optimization task. One option is to utilize polynomial regression, which is a common technique for fitting a function to data points. However, if the number of data points is limited, polynomial regression may not provide accurate results.

We could also choose another method where we uniformly distribute the points all over the range to calculate the values and take the minimum of those values but since those values may not be accurate and are highly inaccurate as the function may not always be lying on those points.

For this question I chose to use Bayesian estimation using Gaussian process regression method. What this method basically does is It calculates probability of the values at some points given the value of some fixed known data points. We start by fitting a model through the gather points and we somehow calculate the next best estimate for the point So as to get to the minimum.

2 Usage

The final solution is given in the "sols" folder named as final-solution.py. The required packages are mentioned in it.

Run the Python file to get the plot of the estimated curve and also the minimum of that function and for what values of C and R.

There are also multiple test Python notebooks that I made in order to test our methods of Naive queries, Gaussian Regression, etc. Those are also given in the same folder.

All this can be found at the GitHub link: `https://github.com/sudo-boo/Azeotropy-optimizer`. There are multiple images in.

3 Methodology

3.1 Modeling the Objective Function

We model the objective function using Gaussian process regression, which provides a distribution over possible functions that could describe the data. This allows us to quantify uncertainty and make informed decisions about where to sample next.

3.2 Acquisition Function

An acquisition function is a heuristic that guides the selection of the next point to evaluate. In this code, the expected improvement is used as the acquisition function. It quantifies the potential improvement over the current best observation and weighs it by the uncertainty in the prediction.

The goal is to find the point with the highest expected improvement, indicating that it has the potential to lead to a better solution.

3.3 Iterative Optimization

The optimization process is iterative. We start with some initial data points and train the Gaussian process model. Then, we iteratively select new points to evaluate using the acquisition function, evaluate the objective function at these points, and update the model with the new data. This process continues until a stopping criterion is met (in our case the number of data points hits max), such as reaching a maximum number of epochs or convergence to a satisfactory solution.

3.4 Visualization

Finally, we evaluate the optimized solution by predicting the minimum value and corresponding variables using the trained GP model.

We recursively find new best values and plot the fitted model. We choosing the values where confidence interval is high and also where the function is minimum. There we follow a trade-off between those two values.

In the given figures, the curve is given and the 95% confidence interval is also represented by light-blue layer.

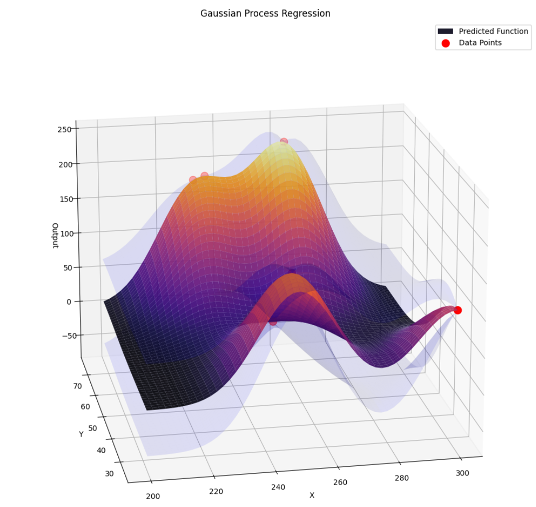

Figure 1: Curve fit after 6 data points

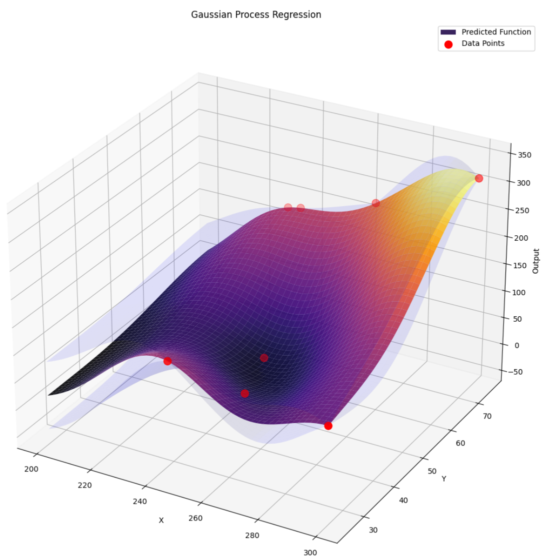

Figure 2: Curve fit after 8 data points

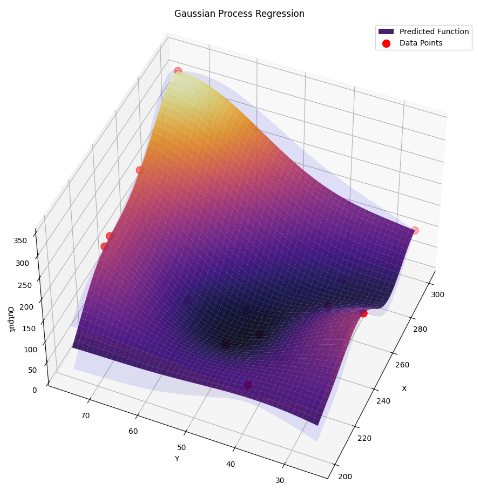

Figure 3: Curve fit after 11 data points

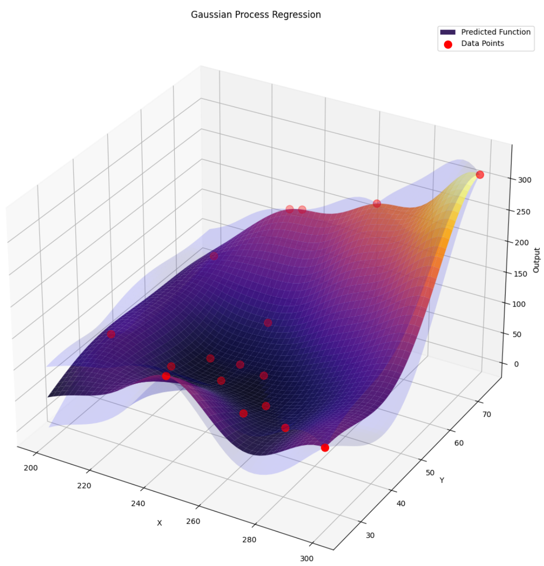

Figure 4: Curve fit after 16 data points

3.5 Result Calculation

The minimum value and corresponding input variables represent the optimized solution found by the Bayesian optimization algorithm.

This here in our case is calculated by forming multiple random Gaussian curves formed by our trained model within the 95% confidence interval, then taking its mean and choosing the values where the value of those mean of the functions is minimum.

In summary, the code implements Bayesian optimization using Gaussian process regression to find the optimal solution for a given objective function within a defined search space. It iteratively refines its estimate of the optimal solution based on the observed data points and the probabilistic model of the objective function.

4 Results

The values for which I got minimum were:

$$C = 253.76884422110552$$

$$R = 46.35678391959799$$

And the minimum cost found was:

$$T = 32.433675076245216$$

These values can also be seen when you run the final-solutions.py.