Project Report

Alexis Park, Jonathan Chen, Kevin Xie
CSE515T: Bayesian Methods in Machine Learning

December 3, 2019

Data visualization

The Branin function is defined as follows:

$$f(\boldsymbol{x}) = a(x_2 - bx_1^2 + cx_1 - r)^2 + s(1-t)\cos(x_1) + s \tag{0.1}$$

With domain X=[-5, 10] x [0, 15] with 1000 values per dimension, figure 1 shows a heatmap of the value of the Branin function.

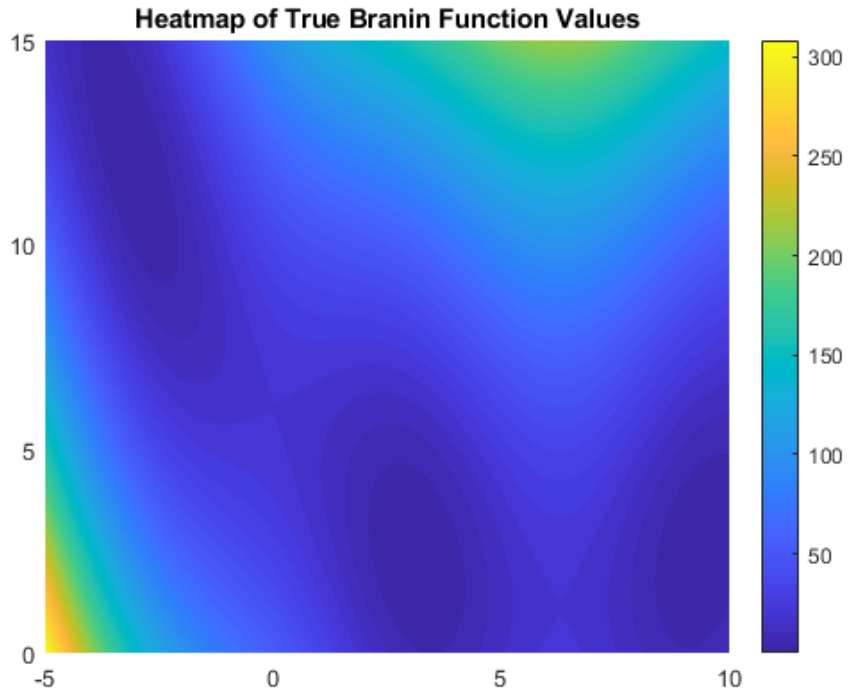

Figure 1: Heatmap of True Branin Function Values

From the plot above, one can see that the function's values fluctuate in a somewhat sinusoidal manner, with a large minimal "trench" spanning diagonally across the center of the domain with steadly increasing regions along the trench's sides. Therefore the function does not appear to be stationary.

By analyzing the equation, we note that the sinusoidal fluctuations can be attributed to both the added cosine term, and the 6th order polynomial term. We thus attempted to pass the data through a transformation that combined an inverse cosine function (arccos) with the cumulative density function of the normal probability distribution (normal cdf). Specfically, we scaled the data from $[0, 1]$ using the normal cdf and passed the modified data through arccos to obtain a set of transformed data. The resulting plot of the transformed data is shown below.

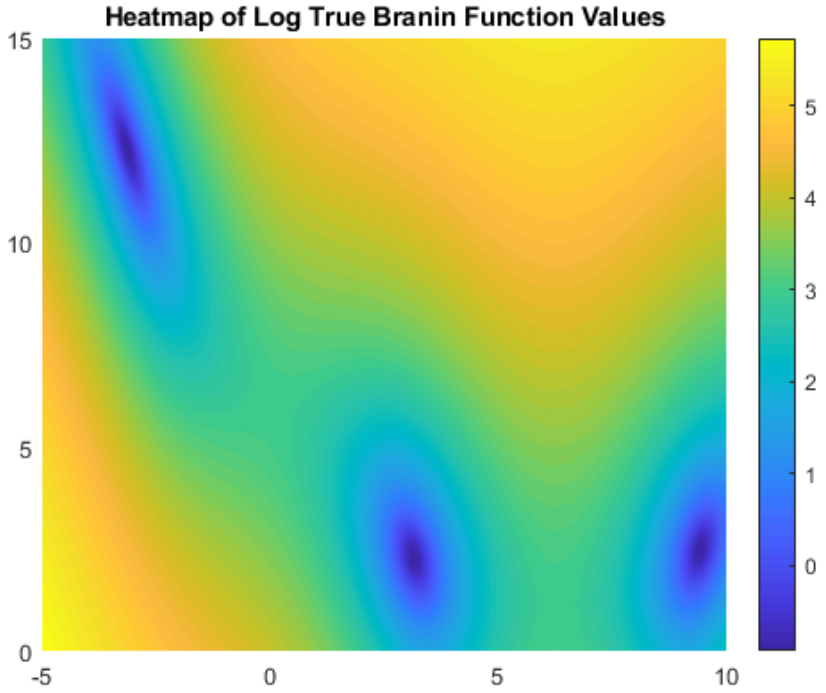

Figure 2:

Figure 2 shows more stationary [INSERT MORE INTERPRETATION]
Now, we plotted kernel density estimates of LDA and SVM benchmark data.

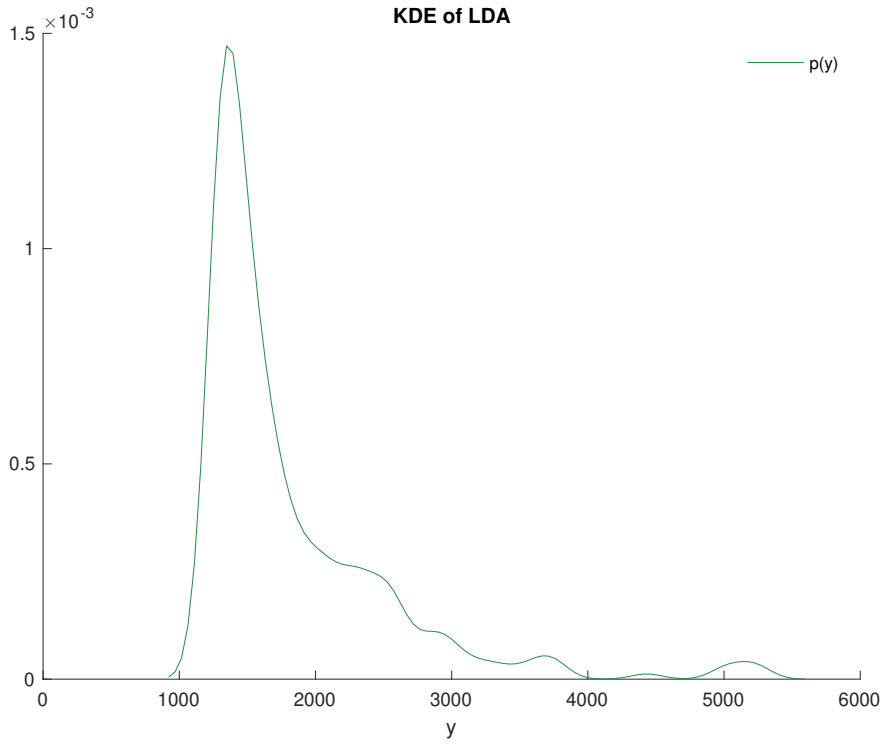

Figure 3:

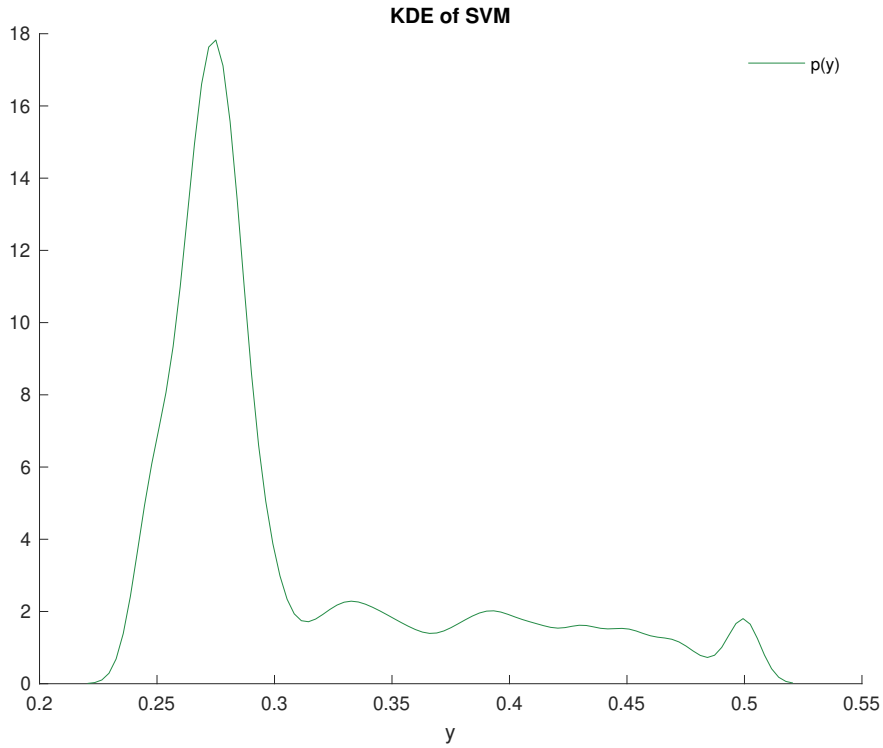

Figure 4:

We plot the kernel density estimates for the SVM and LDA benchmarks below. It can be seen

that the two estimates have relatively similar relative behavior but on significantly different scales.

Similar to Branin function, we also took log of the LDA and SVM values to make the performance better. Plotted data is shown in figure 5 and 6.

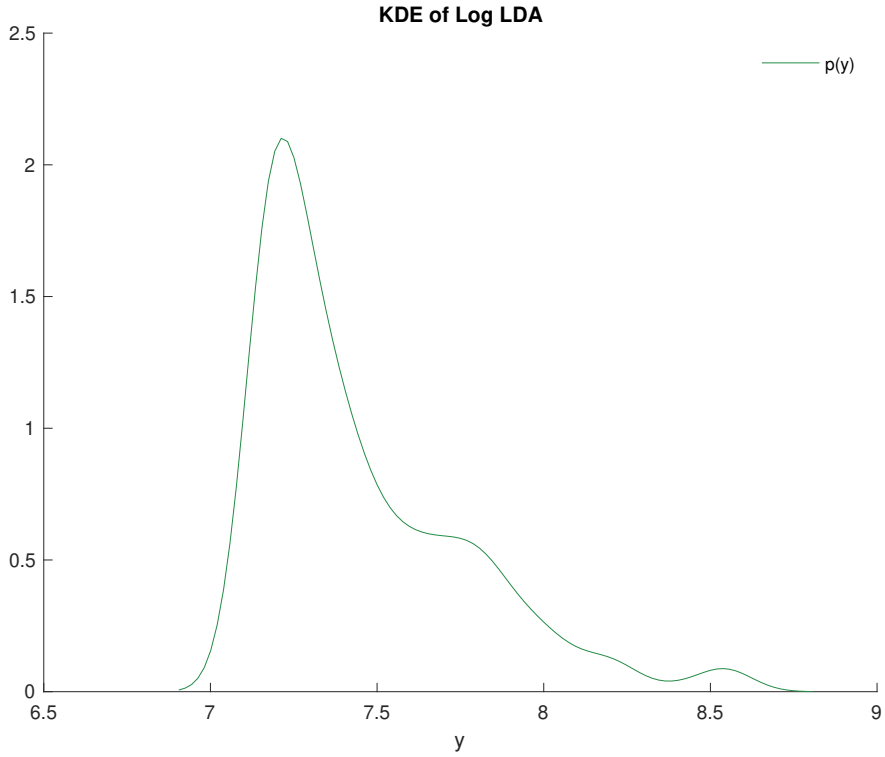

Figure 5:

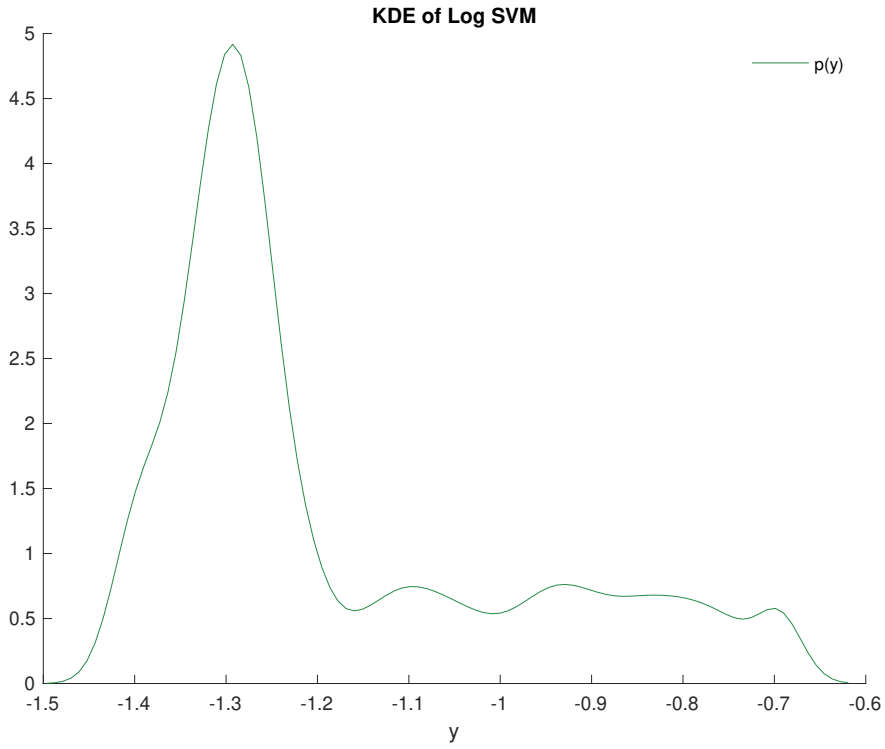

Figure 6:

Model fitting

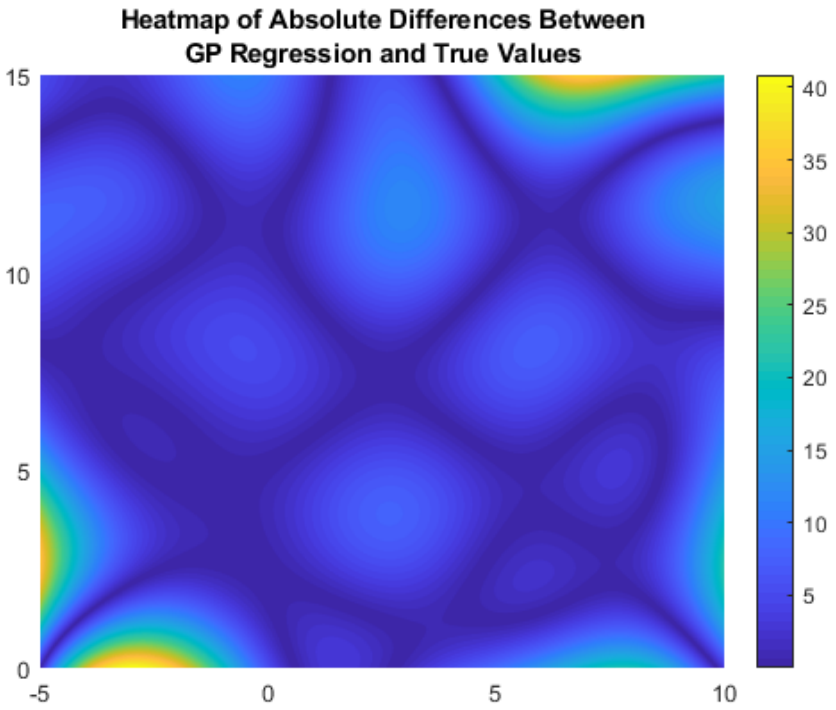

Figure 7:

Q: Compare the predicted values with the true values. Do you see systematic errors?

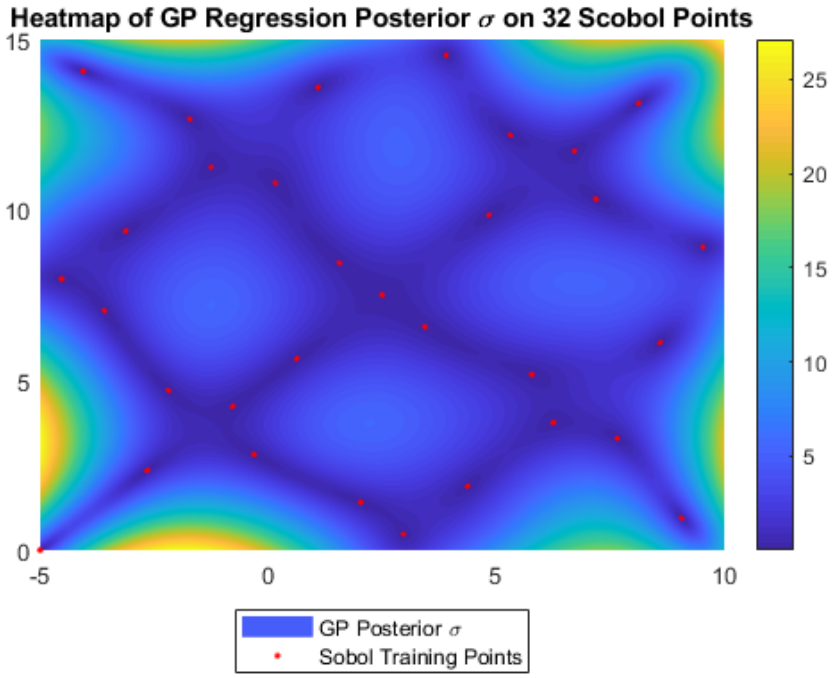

Figure 8:

Q: Do the values make sense? Does the scale make sense? Does the standard deviation drop to near zero at your data points?

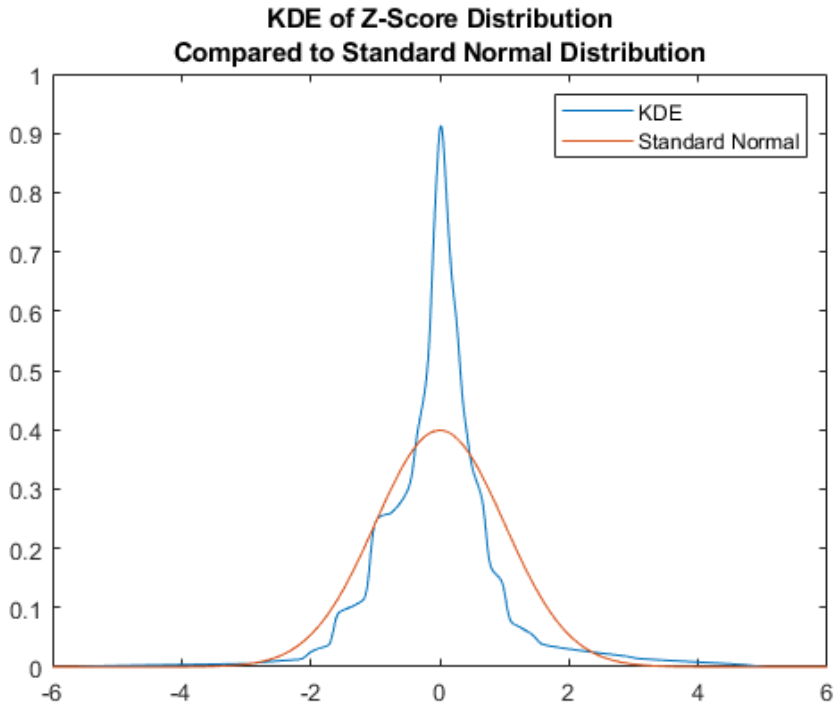

Figure 9:

Based on figure 9, the KDE of Z-score distribution follows approximately standard normal distrubtion with more concentrated peak in the middle.

Now, we repeated model fitting using a log transformation to the output of the Branin function.

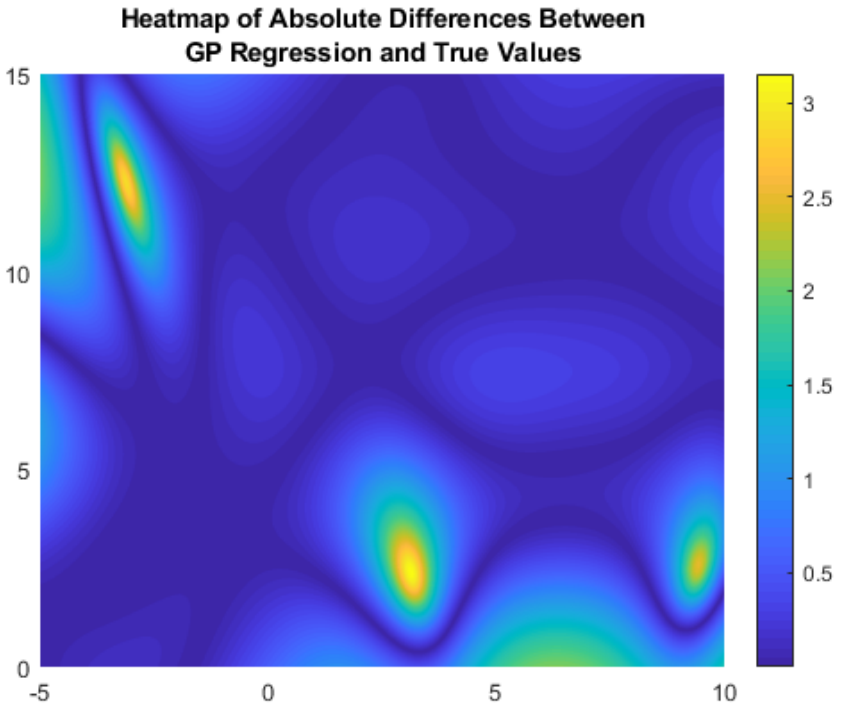

Figure 10:

Q: Compare the predicted values with the true values. Do you see systematic errors?

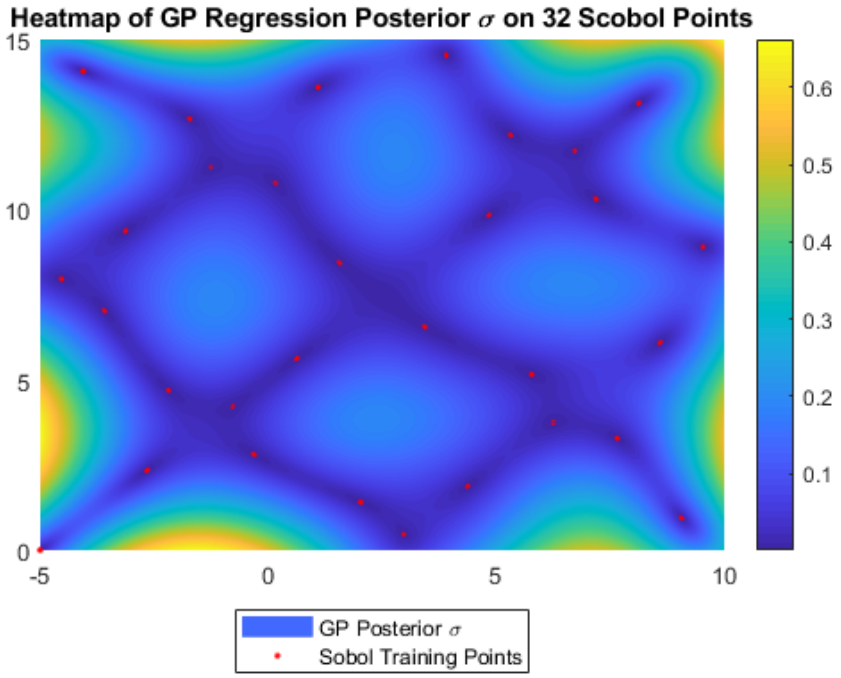

Figure 11:

Q: Do the values make sense? Does the scale make sense? Does the standard deviation drop to near zero at your data points?

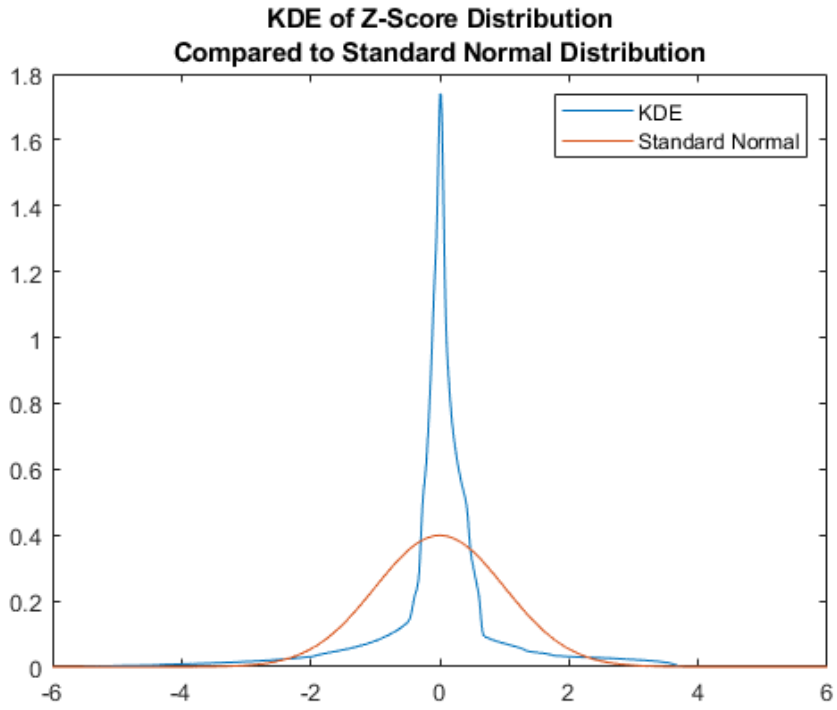

Figure 12:

LOGS: Q: Does the marginal likelihood improve? Does the model appear better calibrated?

Bayesian optimization

The best-fitting models were selected from the previous experiments. Specifically, the Branin Model used a log-transformed dataset, a Constant Mean function, and a Squared Exponential covariance function. Similarly, the LDA Model used a log-transformed dataset, a Constant Mean Function, and a Rational Quadratic covariance Function. Finally, the SVM Model used a normal dataset with a Constant Mean Function and a product of the Rational Quadratic and Squared Exponential functions as its covariance function.

We then used the Expected Improvement (EI) Acquisition Function, defined from the course notes as:

$$a_{ei}(\boldsymbol{x}) = (f' - \mu(\boldsymbol{x}))\Phi(f'; \mu(\boldsymbol{x}), K(\boldsymbol{x},\boldsymbol{x})) + K(\boldsymbol{x},\boldsymbol{x})\mathcal{N}(f'; \mu(\boldsymbol{x}), K(\boldsymbol{x},\boldsymbol{x})) \tag{0.2}$$

Where $\Phi(\boldsymbol{x})$ is the Cumulative Probablity Density of the Normal Distribution, and f' is the minimum value of the current observations.

Using the previously selected 32 points, a GP model was fit using the aforementioned model settings and the following heatmaps of the posterior mean and standard deviation of the Branin function were created:

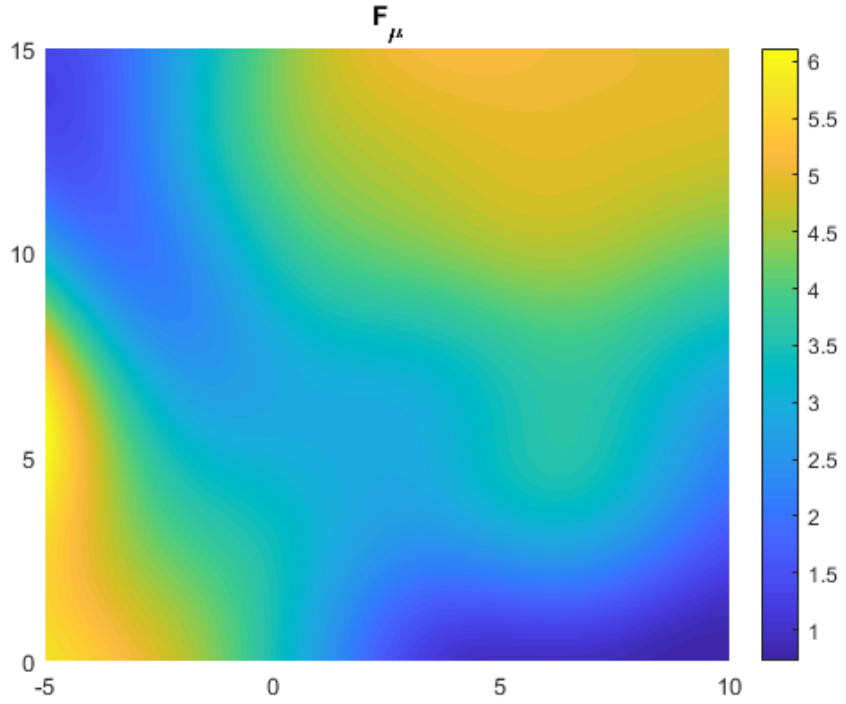

Figure 13: Predictive Posterior Mean of the log Branin Function, calculated using a previously optimized GP model trained on 32 Sobol Sequence points. Warmer colors indicate higher values. Note the predicted minimum areas in dark blue at the top left and bottom right corners of the plot.

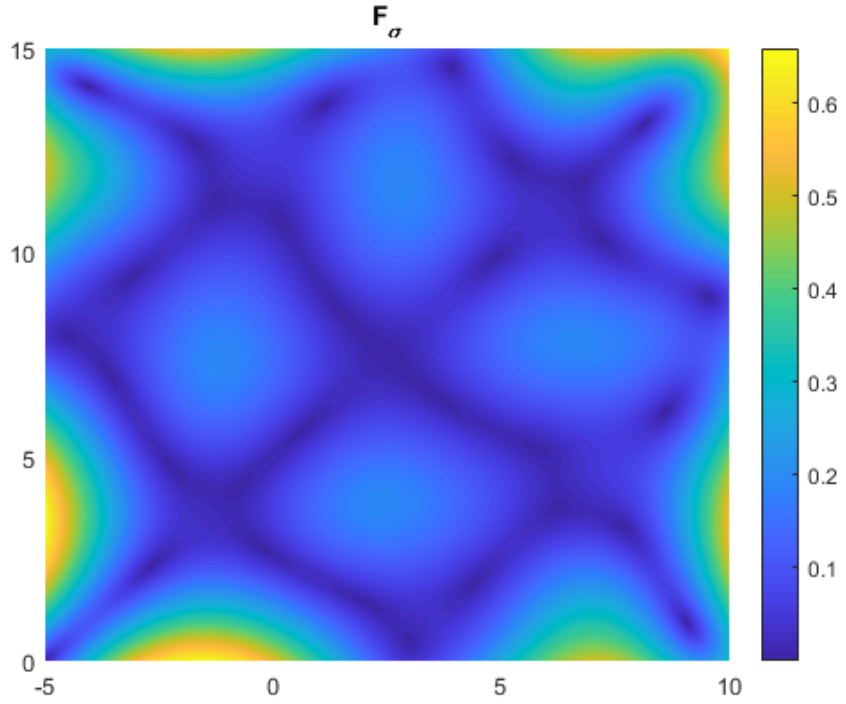

Figure 14: Predictive Posterior Standard Deviation of the log Branin Function, calculated usinga previously optimized GP model trained on 32 Sobol Sequence points. Warmer colors indicate higher deviations and imply greater uncertainty in the prediction.

The EI value was then calculated using the posteriors and identified the point [7.658, 0] as the optimal point to test next. A heatmap of the EI distribution is shown below:

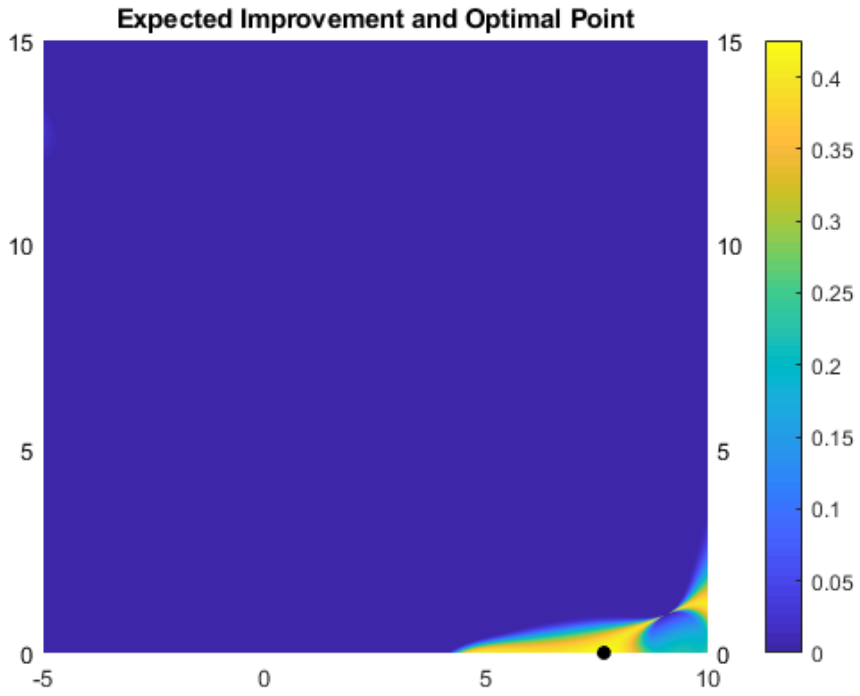

Figure 15: Expected Improvement acquisition values of the optimized GP model trained on 32 Sobol Sequence points. Warmer colors indicate higher expected improvement. The maximum expected improvement is marked as a black point and denotes the optimal location for the next observation.

Analyzing 13, 14, and 15 above, we reason that the proposed optimal testing point is ideal. From the posterior mean, it can be seen that the point [7.658, 0] is within a region of predicted minimal values. Futhermore, from the posterior standard deviation, it can be seen that the point is within a region of higher uncertainty and therefore reduced predictive confidence. Thus, it is plausible that the EI acquisition function would seek to test this area and point in order to identify a possiblly global minimum and improve confidence in an area of uncertainty.

The following bayesian active learning experiment was then applied independently to the Branin, LDA and SVM functions:

1. 5 initial observations were randomly selected, constituting the initial dataset $\mathcal{D}$
2. For the Branin Function only, a dense grid of 250,000 points was generated within the domain of the function
3. A GP model using the respective aforementioned optimized settings was fit to $\mathcal{D}$
4. A new point x was found using the EI acquisition function and the GP predictive posterior
5. The function value $f(x)$ was calculated and the point $(x, f(x))$ was added to $\mathcal{D}$
6. Steps 3-5 were repeated 30 times, resulting in a final dataset $\mathcal{D}$ of 35 points

The performance of each of the above experiments were evaluated using the "gap" measure, defined for minimization as:

$$\text{gap} = \frac{f(\text{best found}) - f(\text{best initial})}{f(\text{maximum}) - f(\text{best initial})} \tag{0.3}$$

The gaps for the Branin, LDA, and SVM models were calculated to be 1.0000, 0.7932, and 0.9604, respectively. This implies that EI successfully found a global minimum of the Branin function, but missed the global minimum of the LDA and SVM functions.

The above bayesian active learning experiment was then modified as such:

1. A seed for the random number generator (RNG) was chosen
2. 5 initial observations were randomly selected, constituting the initial dataset $\mathcal{D}$
3. For the Branin Function only, a dense grid of 250,000 points was generated within the domain of the function for use in calculating the GP predictive posterior
4. A GP model using the respective aforementioned optimized settings was fit to $\mathcal{D}$
5. A new point x was found using the EI acquisition function and the GP predictive posterior
6. The function value *f(x)* was calculated and the point $(x, f(x))$ was added to $\mathcal{D}$
7. Steps 4-6 were repeated 30 times, resulting in a final dataset $\mathcal{D}$ of 35 points
8. A new GP model using the respective aforementioned optimized setting was fit to the original initial dataset $\mathcal{D}$, now called $\mathcal{D}$'
9. A new point x was found using the Random acquisition function, which randomly selects a new point
10. The function value *f(x)* was calculated and the point $(x, f(x))$ was added to $\mathcal{D}$'
11. Steps 8-10 were repeated 150 times, resulting in a final dataset $\mathcal{D}$' of 155 points

repeated 20 times with 20 different random number generator seeds to create different random initializations. However, in addition to the EI acquisition function, a random search acquisition function that randomly selected the next point to observe was implemented as a baseline...

Bonus

For the bonus section, we implemented two more acquisition functions: Lower Confidence Bound (LCB) and Max Variance. We then compared their performances with EI when used on their own and with two heuristics.

First, we implemented a wrapper for aquisition functions that selects a point at random with probability $p = 0.1$; we dubbed this heuristic "random restarts" (RR). We also tried each aquisition function while minimizing the model's hyperparameters after each iteration; we called this "online optimization" (OO).