Gaussian Processes Visual Tool

Eduardo Adame Salles
eduardo.salles@fgv.br

Rio de Janeiro, Brazil
July 6, 2023

Introduction

GP Visual Tool is called to be a system for the interactive modeling, fitting and interpreting of Gaussian processes.

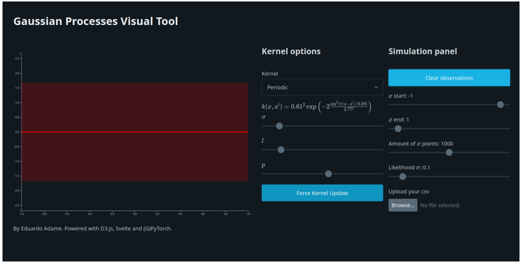

Figure: Overall architecture of the system.

It allows the user to rigorously specify a model by choosing different sets of hyperparameters.

What is a Gaussian process?

Typically, Gaussian processes (GPs) can be seen as a generalization of the Bayesian Regression. Suppose we want to model a $f : \mathbb{R} \to \mathbb{R}$ as a GP. We would define it as following:

$$
\begin{aligned}
\boldsymbol{y} \mid (\boldsymbol{f}, \boldsymbol{x}) &\sim \mathcal{N}(\boldsymbol{f}, \sigma^2 I), \\
\boldsymbol{f} \mid \boldsymbol{x} &\sim \mathcal{GP}(m, k) \equiv \mathcal{N}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x})),
\end{aligned}
$$

for $\boldsymbol{x} = x_1, \cdots, x_N$, $\boldsymbol{y} = y_1, \cdots, y_N$ and $\boldsymbol{f} = f(x_1), \cdots, f(x_N)$. We often refer to m as the mean function (usually $m(x_i) = 0$) and k as the kernel function. Our goal is to find the *posterior* distribution of f.

Means and Kernels

In order to define our *priors* distributions, we need to specify the mean and kernel functions. The mean function is usually set to zero, but the kernel function is a bit more complicated.

There are some properties of kernels:

- Symmetry: $k(x, x') = k(x', x)$;
- Positive definiteness: $\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j k(x_i, x_j) \geq 0$ for any $n \in \mathbb{N}$, $c_1, \cdots, c_n \in \mathbb{R}$ and $x_1, \cdots, x_n \in \mathbb{R}$;
- Stationarity: $k(x, x') = k(x - x')$;
- Isotropy: $k(x, x') = k(\|x - x'\|)$.

Finding the *posterior* distribution

Fitting a GP to a dataset is equivalent to finding the *posterior* distribution of f given $\boldsymbol{y}$ and $\boldsymbol{x}$. This is done by using Bayes' theorem:

$$
\begin{aligned}
p(\boldsymbol{f} \mid \boldsymbol{y}, \boldsymbol{x}) &= \frac{p(\boldsymbol{y} \mid \boldsymbol{f}, \boldsymbol{x}) p(\boldsymbol{f} \mid \boldsymbol{x})}{p(\boldsymbol{y} \mid \boldsymbol{x})} \\
&= \frac{p(\boldsymbol{y} \mid \boldsymbol{f}, \boldsymbol{x}) p(\boldsymbol{f} \mid \boldsymbol{x})}{\int p(\boldsymbol{y} \mid \boldsymbol{f}, \boldsymbol{x}) p(\boldsymbol{f} \mid \boldsymbol{x}) d\boldsymbol{f}} \\
&= \frac{p(\boldsymbol{y} \mid \boldsymbol{f}, \boldsymbol{x}) p(\boldsymbol{f} \mid \boldsymbol{x})}{\int p(\boldsymbol{y} \mid \boldsymbol{f}, \boldsymbol{x}) p(\boldsymbol{f} \mid \boldsymbol{x}) d\boldsymbol{f}} \\
&\propto p(\boldsymbol{y} \mid \boldsymbol{f}, \boldsymbol{x}) p(\boldsymbol{f} \mid \boldsymbol{x}) \\
&\propto \mathcal{N}(\boldsymbol{y} \mid \boldsymbol{f}, \sigma^2 I) \mathcal{N}(\boldsymbol{f} \mid \boldsymbol{x}, k(\boldsymbol{x}, \boldsymbol{x})).
\end{aligned}
$$

Some approaches to get optimal hyperparams

As much of the work in the field of Bayesian statistics, we need to specify priors for our hyperparameters. This is usually done by using Gamma and Inverse Gamma distributions. But somewhere we have to choose the hyperparameters.
Maximizing the evidence is a common approach to get the optimal hyperparameters.
This is done by maximizing the log-marginal likelihood:

$$
\begin{aligned}
\log p(\boldsymbol{y} \mid \boldsymbol{x}) &= \log \int p(\boldsymbol{y} \mid \boldsymbol{f}, \boldsymbol{x}) p(\boldsymbol{f} \mid \boldsymbol{x}) d\boldsymbol{f} \\
&= \log \mathcal{N}(\boldsymbol{y} \mid \boldsymbol{x}, k(\boldsymbol{x}, \boldsymbol{x}) + \sigma^2 I).
\end{aligned}
$$

Applications of GPs

As any Machine Learning model, GPs can be used in a wide range of applications. Some of them are:

- Emulating expensive functions;
- Regression problems;
- Classification problems;
- Time series prediction;
- Anomaly detection;

And some of its advantages are:

- Uncertainty estimation;
- Interpretability;***
- Flexibility.

Life could be a dream.

As one would say, there is no free lunch. GPs are not an exception. Some of its limitations are:

- Computational complexity;
- Memory requirements;
- Non-convex optimization;

Developing a visual tool

FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

In order to understand the behavior of GPs, and each decision we make, a visual tool could be very useful. That's when Gaussian Processes Visual Tool comes in.
We want to, but not limited to:

- Visualize the behavior of GPs;
- Add new observations interactively;
- Change the kernel function;
- Choose hyperparameters in real-time;
- Use our own custom datasets;

Technologies

In order to develop this tool, we will use:

- Python and Javascript as programming languages;
- PyTorch and GPyTorch to handle the GPs;
- Flask to serve our application;
- D3.js to deal with the visualizations;
- Svelte to give us a reactive frontend;

Let's start sampling

Our main interactive visualization is a line plot of the current GP. We can sample from it by clicking on any point of the plot. This will add a new point to the plot, and a new sample to the posterior distribution. Clicking in an existing point will remove it from the plot, and the sample from the posterior.

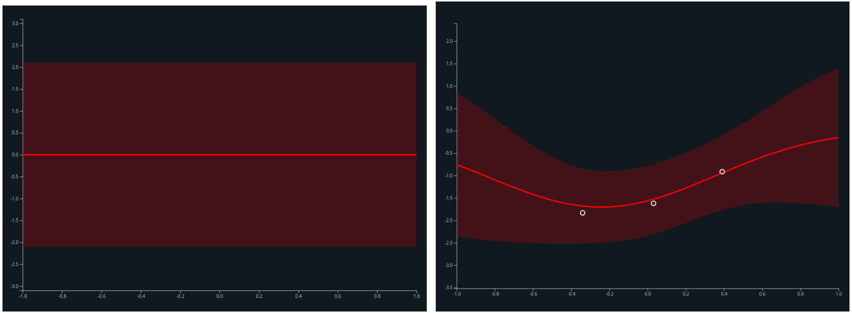

Figure: Sampling from the GP.

Hovering, you can see the marginal distribution of the point.

Beyond choosing a kernel

As we said before, we can change the kernel function. We can also change the hyperparameters of the kernel. This will change the behavior of the GP.

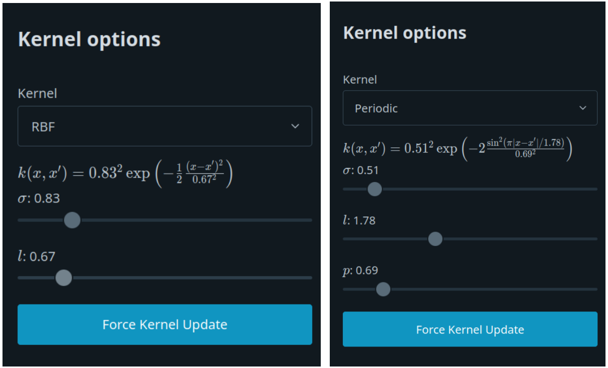

Figure: Changing kernels.

Customization is allowed

You can also upload your own dataset. This will allow you to play with the tool, and see how the GP behaves with your own data. You can set the noise level, and axis limits.

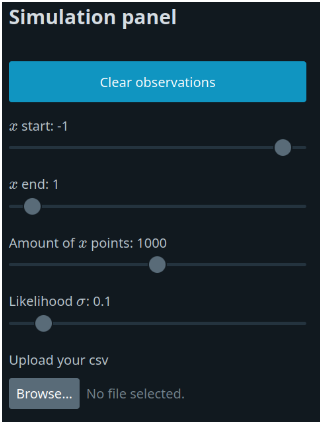

Figure: Simulation options

Future work

There are some things we would like to add to the tool:

- Add more visualizations, like a heatmap of the covariance matrix;
- Add more kernels;
- Include export options;
- Make it run on GPU by default;
- Deploy it to a server;
- Add math explanations to the tool;
- Operations between kernels;

Thanks!