Robot Physicist

Manoj Kumar, Phil Yeres, Michele Ceru

Abstract—The abstract goes here.

1 Introduction

Particle physics experiments generally require expensive shared equipment, or time consuming simulations. Therefore, optimizing the experiment simulation procedure to minimize the time required to achieve results is a subject of importance to physicists. The aim of our project is to use data science techniques to reduce resources used in the experimental process. The experimental model has three components:

1) Experimental configurations such as the energy required to run the particular experiment (phi)
2) Quantities that we would like to infer from the experiment (theta), for instance, we may want to estimate the Weinberg angle (a fundamental parameter of the Standard Model of particle physics).
3) Data generated from the experiment (X)

The aim of our project is to estimate the underlying data generating distribution from simulations which we?ll use to predict the settings of the experiment that allow us to obtain theta with high confidence. If we can predict these settings, we can reduce the number of experiments we?d need to run.

1.1 Toy data

As a starting point we would estimate theta and phi on toy problems where we know the ground truth of the data generating distribution. For example, X can follow a gaussian distribution with mean theta and variance phi. For this toy data we will take the following steps:

1) For the toy-data generating distribution we would generate a number of samples for a fixed value of theta and phi
2) We would then estimate $P(X|\theta,\phi)$. For a simple distribution, we could use a histogram to approximate this using evenly-spaced grids.
3) The third step would be to infer $P(\theta|X,\phi)$ from $P(X|\theta,\phi)$ using Bayes theorem:

$$P(\theta|X,\phi) = P(X|\theta,\phi)P(\theta|\phi)/P(X|\phi).$$

We would also fix the prior P(theta — phi) to some simple distribution and not infer this from data.
4) Run loops from 1-3 for different values of phi such that the information gain in P(theta — X, phi) is maximized. As a possible extension of our project, we may perform bayesian optimization to choose the next value phi intelligently given previous evaluations.

1.2 Physics data

After we ha ve created the toy data pipeline, we?ll extend it for the actual simulated experiments provided by our project advisor (https://github.com/lukasheinrich/higgs-mc-studies). For the actual simulated experiments, the ground truth is not known, so we plan to make use of a liklihood-free inference toolbox (https://github.com/diana-hep/carl) to estimate $P(X|\theta,\phi)$. When we move from using the toy data to actual experimental data, and the computational load will increase significantly. Therefore, as discussed with our project advisor, we will look for opportunities to parallelize computations, and we will make use of AWS and docker to deploy the experiment simulator to the cloud.

2 Bayesian motivation

We indicate with x the data that is the output of the experiment. This output depends on θ (that represent a constant of nature) and Φ (that represent the setting to the experiment). We call $P(x|\theta,\Phi)$ the probability distribution of the data given θ and Φ.
The information gain for this distribution is defined as:

$$EIG(\Phi) = \int P(x|\Phi)\Big[H[P(\theta)] - H[P(\theta|x,\Phi)]\Big]dx \quad (1)$$

where we indicated with H the entropy, that for a discrete distribution is defined as:

$$H[P] = -\sum_{k\geq 1} p_k \log p_k \quad (2)$$

To compute that we need to sample from $P(x|\Phi)$ and know $P(\theta|x,\Phi)$.

- We don't have $P(x|\Phi)$ because the data of the experiment is conditioned on θ as well, consequently we can only sample from the distribution $P(x|\theta,\Phi)$. But we can calculate it using the following:

$$P(x|\Phi) = \int P(x,\theta|\Phi)d\theta = \int P(x|\theta,\Phi)P(\theta|\Phi)d\theta \quad (3)$$

Where the last equality is the expected value of $P(x|\theta,\Phi)$ under the distribution $P(\theta|\Phi)$. Discretising the integral:

$$P(x|\Phi) = \sum_{i=1}^{n} P(x|\theta_i,\Phi)P(\theta_i|\Phi) = \frac{1}{n}\sum_{i=1}^{n} P(x|\theta_i,\Phi) \quad (4)$$

Where in the last equality we are assuming to have an uniform distribution $P(\theta_i|\Phi) = 1/n$. We use the black box to generate the distributions $P(x|\theta_i, \Phi)$ for n values of θ:

$$\begin{aligned} &\theta_1 \rightarrow P(x|\theta_1, \Phi) \\ &\theta_2 \rightarrow P(x|\theta_2, \Phi) \\ &\dots \\ &\theta_n \rightarrow P(x|\theta_n, \Phi) \end{aligned} \tag{5}$$

each of these distribution is obtained from a histogram of the sampled data. Using these we can compute $P(x|\Phi)$.

Since $P(x|\Phi)$ is now discretized because of the histogram, we sample from a multinomial with the parameters, x_i's given by the bin centers and the probability computed from the normalized histogram. This is done from lines 92 to 107 in the code.

- To calculate the posterior $P(\theta|x_j, \Phi)$ where x_j is sampled using the method described above, we can use Bayes theorem:

$$P(\theta|x_j, \Phi) = \frac{P(x_j|\theta, \Phi)P(\theta|\Phi)}{P(x_j|\Phi)} \tag{6}$$

- After doing this, we average out $-H[P(\theta|x_j, \Phi)]$ across all x_j's to calculate the Expected Information Gain.

3 Toy model implementation

3.1 Program structure

Same description of the toy model code with reference to the flowchart in Fig. 1.

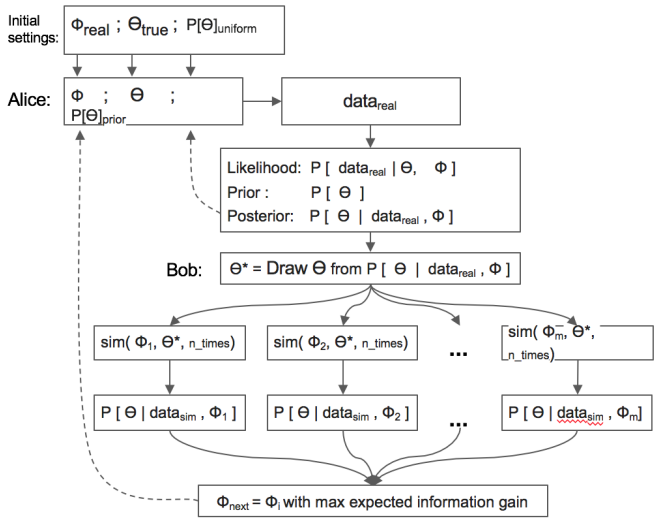

Fig. 1. Flowchart of the program

3.2 Results

Same description of the results

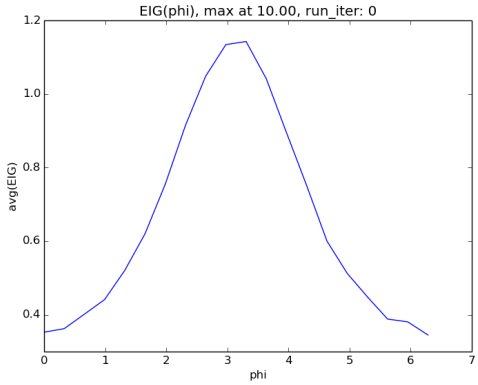

Fig. 2. Expected Information Gain

4 Application to Physics Data

5 Conclusion

The conclusion goes here.

Appendix A
Title

Appendix one text goes here.

Appendix B

Appendix two text goes here.

Acknowledgments

The authors would like to thank...

References

[1] References go here