# Robot Physicist

Manoj Kumar, Phil Yeres, Michele Ceru

**Abstract**—We integrate the necessary components and demonstrate a proof-of-concept for a "robot physicist" that is able to plan the next experiment needed to most efficiently measure a fundamental constant of nature. We then apply the algorithm to simulated data.

## 1 INTRODUCTION

PARTICLE physics experiments generally seek to estimate fundamental constants of nature. These constants can be measured with varying levels of uncertainty, depending on the choice of experimental parameters. Our project is designed to demonstrate a proof-of-concept for automating the identification of the experimental parameters which minimize the uncertainty associated with measurement of the quantity of interest in the experiment. This proof of concept can be extended to real physics simulations.

This "robot physicist" system can be understood using the simple analogy of two experimenters, Alice and Bob. Alice runs experiments which generate data, and she uses the data to update her prior on the distribution of the natural constant of interest. Bob uses the posterior from Alice's experiment to run a set of simulated experiments with various possible experimental settings, and he identifies the experimental settings that maximize the expected information gain. Bob then hands the best experimental settings back to Alice, and Alice repeats the process.

### 1.1 Notation

The experimental model has three main components:

1) $\phi$ : Experimental configurations such as the energy required to run the particular experiment.
2) $\theta$: Quantities that we would like to infer from the experiment
3) $X$: Data generated from the experiment

## 2 PROOF-OF-CONCEPT MODEL

### 2.1 Black-box

For our proof-of-concept we use a black-box simulator which generates data, $X$, such that $X$ follows a univariate Gaussian distribution with mean $\theta$ and variance $2 + \cos(\phi)$. This proof-of-concept would hold for any black box, but this configuration was chosen for convenience, as $\phi$ is unconstrained, and variance is always non-negative. This black-box serves three purposes.

1) purpose 1
2) purpose 2
3) purpose 3

### 2.2 Algorithm

#### 2.2.1 Generate a posterior distribution on theta

This step contains two notable processes:

**Algorithm 1:** Main algorithm

**Result**: estimate $\theta_{true}$ and $\phi_{real}$
initialize $\theta_{true}$, $\phi_{real}$ and $P(\theta)_{prior}$;
**while** *iter* $<$ *num iters* **do**
  $P(\theta_{pos})$ = `gen_theta_posterior(`$\theta_{true}$`,` $\phi_{real}$`,` $P(\theta)_{prior}$`);`
  $P(\theta_{prior}) = P(\theta_{pos})$;
  $\phi_{real}$ = `optimize_phi(`$\theta_{prior}$`,` $H(\theta)$`);`
**end**
function `gen_theta_posterior` ($\theta$, $\phi$, $P(\theta)_{prior}$);
$X$ = `black_box(`$\theta$`,` $\phi$`);`
$P(X|\theta,\phi)$ = `likelihood(X,` $\theta$`,` $\phi$`);`
**return** `normalize(`$P(\theta)$ $\times$ $P(X|\theta,\phi)$`);`

function `optimize_phi`($P(\theta)$, $H(\theta)$;
**return** $\arg\max_\phi$ =
$\int P(X|\Phi)(H[P(\theta)] - H[P(\theta|X,\Phi)])\ dX$

1) Computing the likelihood of $X$ given $\theta$ and $\phi$:
We generate an empirical pdf using a normalized histogram with samples generated from the black-box using $\theta$ and $\phi$.
2) Finding an appropriate step size for $\theta$
Given an lower- and upper-bound for theta (which should be available for most physical constants of interest), and, for convenience, using a univariate Gaussian distribution, $\sigma_{theta_{posterior}}$, which scales as $\frac{\sigma_{theta_{prior}}}{\sqrt{n}}$, with $n$ = the number of samples used to compute the likelihood.

After calculating these components, the computation of the posterior is trivial.

#### 2.2.2 Optimize phi

We use the following simplifications to optimize over $\phi$ in our toy model.

1) We use a greedy procedure in which we compute the information gain for a set of plausible $\phi$ 's (experimental settings) and we return the $\phi$ that maximizes the expected information gain.
2) For a fixed value of phi, we approximate $\int P(X|\Phi)(H[P(\theta)] - H[P(\theta|X,\Phi)])\ dX$ with $H[P(\theta)] - \frac{\sum_{i=1}^{N} H[P(\theta|X_i,\Phi)])}{N}$ where $N$ is the number of times the experiment is carried out with the same experimental settings.

3) To estimate the integrated expected information gain, we need to integrate $H[P(\theta|X,\Phi)]$ over $\theta$. We in-turn replace this integration with a point-estimate using the MAP estimate of P($\theta$)

## 3 Improvements and Extensions

### 3.1 Improvements to the toy model implementation

#### 3.1.1 Likelihood-free inference

Our toy model demonstrates the case where $X$ is a univariate gaussian. For multi-dimensional $X$ , we can replace the usage of histograms with likelihood-free inference techniques to compute $P(\theta|X,\phi)$

#### 3.1.2 Bayesian optimization

Currently we optimize over $\phi$ by choosing a range of values with a fixed step size. The optimization over $\phi$ is a technique that very naturally fits into a bayesian optimization framework since the function that computes the expected information gain is a very expensive function that includes many simulator calls.

#### 3.1.3 Bobs choice of $\theta$

Computing the expected information gain using the MAP estimate of $\theta$ could be replaced by computing the mean information gain by sampling from the posterior.

### 3.2 Extension to a real physics simulator

#### 3.2.1 Integrating a physics experiment simulator

We can connect our toy data pipeline to a real physics experiment simulation developed by Lukas Heinrich (https://github.com/lukasheinrich/higgs-mc-studies). The experiment takes has $\phi$ = (sqrtshalf, polbeam1, polbeam2), and it generates experimental data (events) as json.

#### 3.2.2 Extension for Likelihood-free inference

For the actual simulated experiments, the ground truth is not known, and this proof of concept can be extended to make use of a likelihood-free inference toolbox (https://github.com/diana-hep/carl) to estimate $P(X|\theta,\phi)$ .

#### 3.2.3 Scaling the proof-of-concept to large experiments

1) Caching:
2) Parallelism:

## 4 Bayesian motivation

We indicate with $x$ the data that is the output of the experiment. This output depends on $\theta$ (that represent a constant of nature) and $\Phi$ (that represent the setting to the experiment). We call $P(x|\theta,\Phi)$ the probability distribution of the data given $\theta$ and $\Phi$. Using Bayesian theory we can compute the posterior of $\theta$ given $x$ and $\Phi$ as follow:

$$P(\theta|x,\Phi) = \frac{P(x|\theta,\Phi)P(\theta|\Phi)}{P(x|\Phi)} \tag{1}$$

When $\Phi$ varies this distribution changes and so does the uncerainty on the mesure of $\theta$. The goal is to find the value of $\phi$ that generate a distribution with the smallest possible uncerainty. To do that we define the information gain as a function of $\phi$ and than we find the value of $\phi$ that maximizes it. The information gain is definde as:

$$EIG(\Phi) = \int P(x|\Phi)\Big[H[P(\theta)] - H[P(\theta|x,\Phi)]\Big]dx \tag{2}$$

where we indicated with $H$ the entropy. We approximate it discretizing the distribution on $theta$, as:

$$H[P] \simeq -\sum_{k\geq 1} p_k \log p_k \tag{3}$$

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 $\theta$ as well, consequently we can only sample from the distribution $P(x|\theta,\Phi)$. But we can calculate it using the following:

$$P(x|\Phi) = \int P(x,\theta|\Phi)d\theta = \int P(x|\theta,\Phi)P(\theta|\Phi)d\theta \tag{4}$$

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

$$P(x|\Phi) \simeq \sum_{i=1}^{n} P(x|\theta_i,\Phi)P(\theta_i|\Phi) = \frac{1}{n}\sum_{i=1}^{n} P(x|\theta_i,\Phi) \tag{5}$$

  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 $\theta$:

$$\begin{aligned} &\theta_1 \rightarrow P(x|\theta_1,\Phi)\\ &\theta_2 \rightarrow P(x|\theta_2,\Phi)\\ &\dots\\ &\theta_n \rightarrow P(x|\theta_n,\Phi) \end{aligned} \tag{6}$$

  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:

$$P(\theta|x_j,\Phi) = \frac{P(x_j|\theta,\Phi)P(\theta|\Phi)}{P(x_j|\Phi)} \tag{7}$$

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

## 5 Results

### 5.1 Toy model

We ran the toy model with $\theta_{true} = 1.0$ and we scanned $\theta$ in the range from $-3$ to 3 with 200 steps. Then we loop over the possible values of $\phi$ in the range $\phi = 2 + w$ with $w$ in

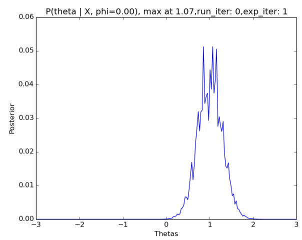


Fig. 1. $\phi = 0$

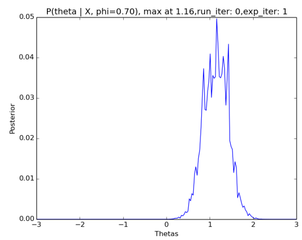


Fig. 2. $\phi = 0.70$

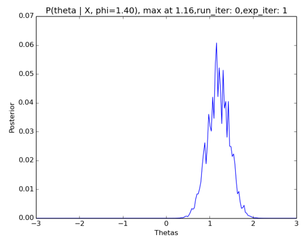


Fig. 3. $\phi = 1.40$

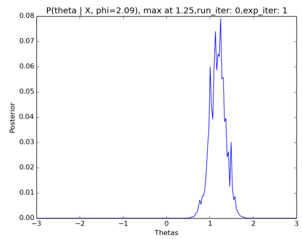


Fig. 4. $\phi = 2.09$

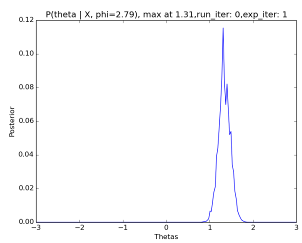


Fig. 5. $\phi = 2.79$

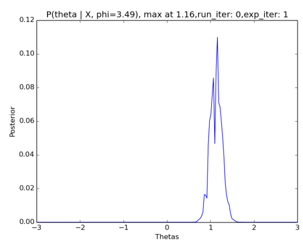


Fig. 6. $\phi = 3.49$

$(0 - 2\pi)$. In the set of Figures from 1 to 6 we report the plots of the posterior for different value of $w$. The information gain is plotted in Figure 7. We notice that the posterior gets better (with less variance and with mean value closer to $\theta_{true}$) when $\phi$ approaches the value that correspond to the max information gain.

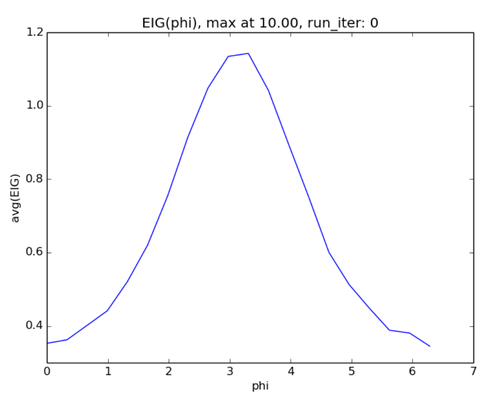


Fig. 7. Expected Information Gain

## 6 Conclusion

We have designed a proof-of-concept for a robot physicist system which is capable of searching for the experimental

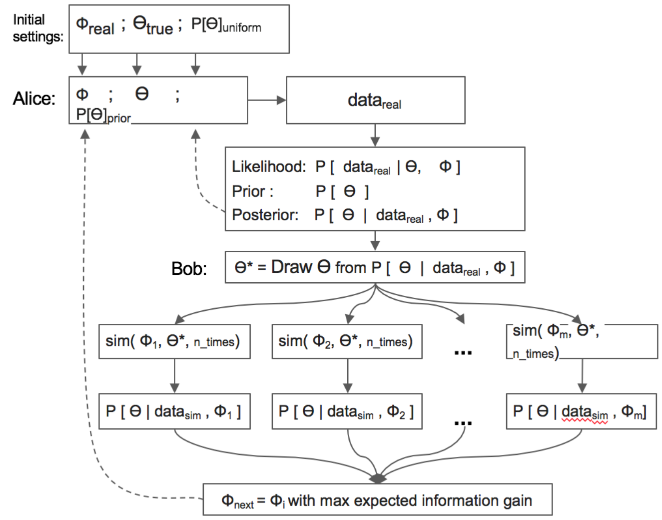


Fig. 8. Flowchart of the program

settings which most effectively measure a quantity of interest. We demonstrated a working example of the system using toy data, we outlined next steps to improve the toy model, and proposed next steps for scaling the model to real physics simulations.

## Appendix

Appendix two text goes here. 9.

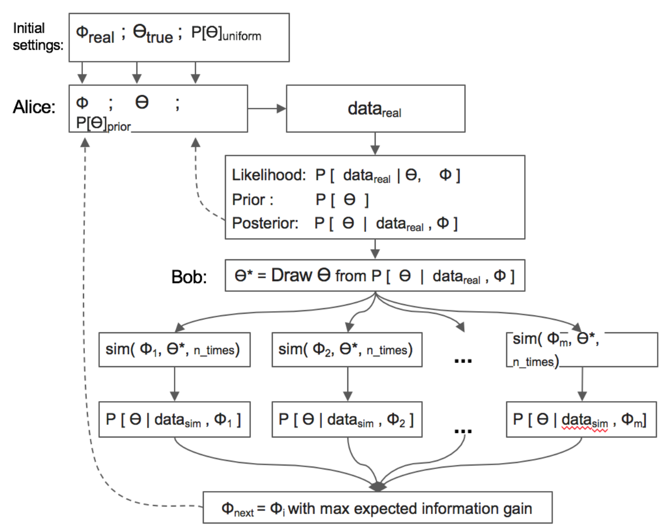


Fig. 9. Flowchart of the program

## Acknowledgments

The authors would like to thank Professor Kyle Cranmer for his guidance on this project; Lukas Heinrich for making a physics simulator available to us; and Professor Claudio Silva and our classmates for providing helpful feedback during our project update presentations.

## REFERENCES

[1] A. Gelman, *Bayesian Data Analysis*, 2nd ed. Chapman & Hall/CRC Texts in Statistical Science, 2003.