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. First we apply the algorithm to toy data generated from a Gaussian distribution, then we apply the algorithm to data obtained from a simulated physics experiment.

1 Introduction

PARTICLE physics experiments generally seek to estimate fundamental constants of nature. Depending on the choice of experimental settings, these constants can be measured with varying levels of uncertainty. Physicists would like to measure these constants with a minimum of uncertainty. Our project is designed to demonstrate a proof-of-concept for automating the identification of the experimental settings which minimize the uncertainty in the measurement of the quantities of interest in an experiment.

This ”robot physicist” system can be understood using the simple analogy of two experimenters, Alice and Bob. Alice runs real 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 as his prior 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

Three variables will be referenced frequently in this paper, so they bear special mention:

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

2 Robot Physicist Proof-of-Concept

2.1 Toy Black-box Simulator

To demonstrate the Robot Physicist algorithm we define a black-box simulator which generates data, X, such that X follows a univariate Gaussian distribution with mean θ and variance $2 + \cos(\phi)$. This proof-of-concept would hold for any black box, but this configuration was chosen for convenience, as ϕ is unconstrained.

2.2 Robot Physicist Algorithm

In the following pseudo-code we outline the key algorithmic steps and functions core to the Robot Physicist system (Figure 1)

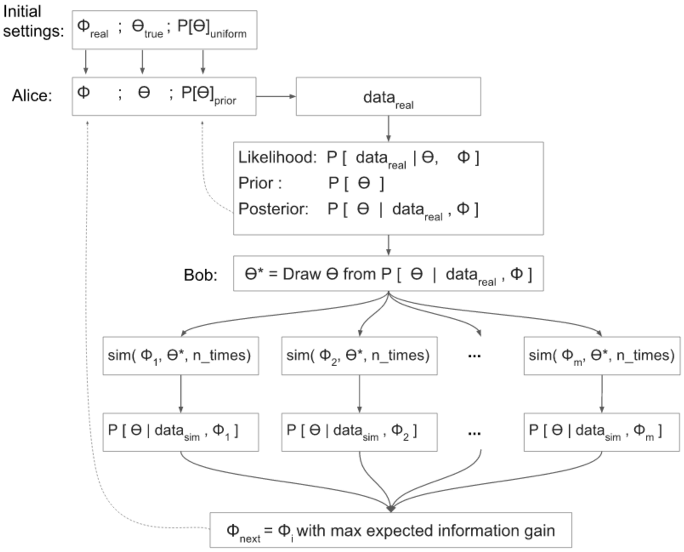

Fig. 1. Flowchart of the program

Algorithm 1: Robot Physicist Pseudo-code

```
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($P(\theta)$ × $P(X|\theta, \phi)$;

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

3 Algorithm Description

3.1 Generate a posterior distribution on theta

The data generated by the experiment, x, depends on θ, the constant of nature we seek to measure, as well as Φ, the settings of the experiment. $P(x|\theta, \Phi)$ is the likelihood (the probability of the data given the experimental settings and the constant of nature).

To estimate the likelihood, we generate an empirical pdf using a normalized histogram with samples generated from the black-box (real-experiment) using θ and ϕ.

Then, using Bayes theorem, we can compute the posterior (the probability distribution for the constant of interest given the observed data and the experimental settings) as follows:

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

Given an lower- and upper-bound for θ (which should be available for most physical constants of interest), we discretize θ using a suitable step-size within these bounds.

As the experimenter varies Φ, this distribution changes. The experimenter's goal is to find the value of ϕ that estimates θ with a minimum of uncertainty.

Once we find this posterior for an experimental setting, we hand this to the second experimenter who performs the optimize phi loop using a set of simulated experiments to identify the next candidate experimental setting.

3.2 Optimize phi

To find the best ϕ we define the expected information gain as a function of ϕ, and then we find the value of ϕ that maximizes it. The expected information gain (EIG) is defined as:

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

where H denotes entropy, $P(\theta)$ is Alice's posterior, and $P(\theta|x, \phi)$ is the posterior obtained by Bob for different experimental settings. Because we have an upper and lower bound on θ, we discretize the distribution of θ using a predetermined step size:

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

To compute this we need to sample from $P(x|\Phi)$. 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:

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

We approximate this integral with $P(x|\theta_{MAP}, \phi)$. Where θ_{MAP} is the maximum a posteriori estimate of Bob's prior. This makes the expected information gain reduce to this approximation:

$$EIG(\Phi) = H[P(\theta)] - \frac{\sum_{i=1}^{N} H[P(\theta|x_i, \Phi)]}{N} \tag{5}$$

where x_i is the data obtained from the i^{th} simulation.

For now, we use a greedy procedure in which we compute the information gain for a set of plausible ϕ 's (experimental settings) and we return the ϕ that maximizes the expected information gain.

4 Extension to a Real Physics Simulator

We connected our data pipeline to a real physics experiment simulation provided by Lukas Heinrich (https://github.com/lukasheinrich/higgs-mc-studies). We scanned values of the first dimension of ϕ, "sqrtshalf", and values of the simulator variable "Gf" (which is related to θ) to generate data.

There is substantial overhead associated with running a simulation on the real physics simulator. Generating 100 events takes 15 seconds, and generating 100K events take 100 seconds. To optimize the process, we run simulations in parallel and cache the results:

1) Caching: To avoid starting up the simulator to generate small samples, we cache results for 2,000 combinations of ϕ and θ. We ran the simulator for 10 values of ϕ and 200 values of θ.
2) Parallelization: To generate and cache the data we used five 8-core AWS compute-optimized instances. All 2,000 simulations were completed in 12 hours of wall time (i.e., 60 hours of computing time).

5 Results

5.1 Toy Model Results

We ran our toy model with $\theta_{true} = 1.0$ and we scanned θ in the range from -3 to 3 with 200 steps. Then we loop over the possible values of ϕ in the range $\phi = 2 + \cos(w)$ with w in $(0 - 2\pi)$.

In Figures 9 to 14 we plot the posterior of θ for different value of ϕ . As ϕ, the experimental settings, approaches the optimal value, approximately 3, the variance in our measure of the constant of interest decreases.

The information gain is plotted in Figure 8. We can see that the posterior gets better (with less variance and with mean value closer to θ_{true}) when ϕ approaches the value that corresponds to the maximum expected information gain.

5.2 Physics Experiment Results

We did not have access to data from a real physics experiment so we used data from a physics experiment simulation as a substitute. We ran the simulator with $Gf_{true} = 1.166390e - 05$ and $\phi_{real} = 45$ and we assumed the output of this simulation to be the real physics experiment outcome (the data that Alice generates).

We use the simulator to generate data for 10 values of ϕ in the range (40, 50), scanning θ from $5.83195e - 06$ to $1.749585e{-}05$ with 200 steps (Bob's experiments). The number of loops we used for our model are N_experiment= 20 and n_iter = 10. The posteriors we obtained are reported in Figures 9 to 18, and the average information gain is plotted in Figure 19

6 Potential Implementation Improvements

6.1 Likelihood-free inference

For multi-dimensional data, instead of using histograms we could use likelihood-free inference techniques to compute $P(\theta|X, \phi)$. For instance, this proof of concept can be extended to make use of a likelihood-free inference toolbox (https://github.com/diana-hep/carl) to estimate $P(\theta|X, \phi)$.

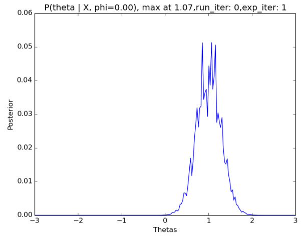

Fig. 2. Toy model: $\phi = 0$

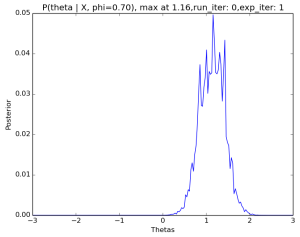

Fig. 3. Toy model: $\phi = 0.70$

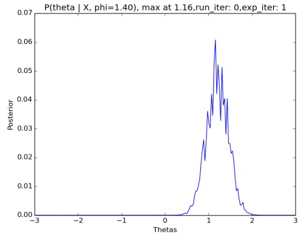

Fig. 4. Toy model: $\phi = 1.40$

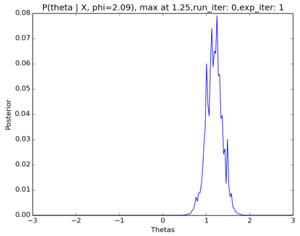

Fig. 5. Toy model: $\phi = 2.09$

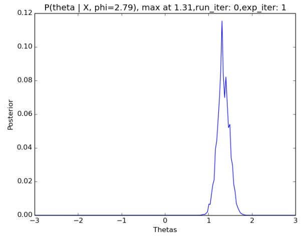

Fig. 6. Toy model: $\phi = 2.79$

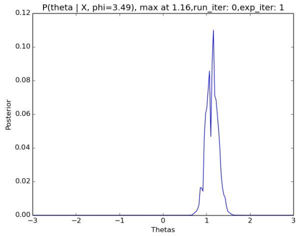

Fig. 7. Toy model: $\phi = 3.49$

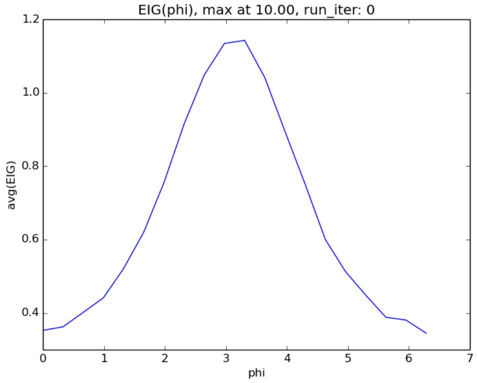

Fig. 8. Toy model: Expected Information Gain

6.2 Bayesian optimization

Currently we optimize over the experimental settings, ϕ, by choosing a range of values with a fixed step size. However, optimization over ϕ is a technique that very naturally fits into a Bayesian optimization framework because the function that computes the expected information gain is a very expensive function that includes many simulator calls.

6.3 Bob's choice of θ

Currently we compute the expected information gain using the maximum a posteriori estimate of θ. However, this could be replaced by computing the average information gain by sampling from the posterior distribution, or estimating the integral by monte carlo methods.

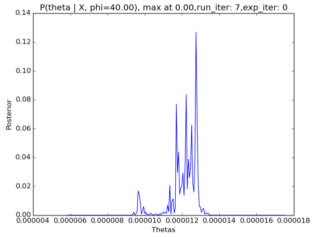

Fig. 9. Physics simulation: $\phi = 40$

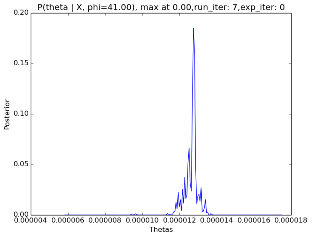

Fig. 10. Physics simulation: $\phi = 41$

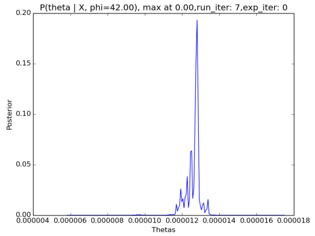

Fig. 11. Physics simulation: $\phi = 42$

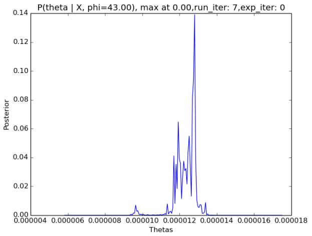

Fig. 12. Physics simulation: $\phi = 43$

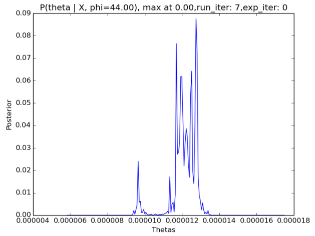

Fig. 13. Physics simulation: $\phi = 44$

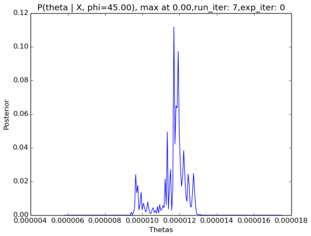

Fig. 14. Physics simulation: $\phi = 45$

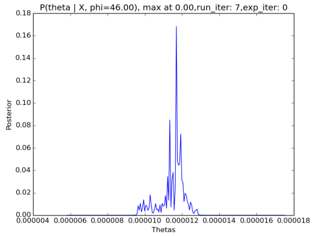

Fig. 15. Physics simulation: $\phi = 46$

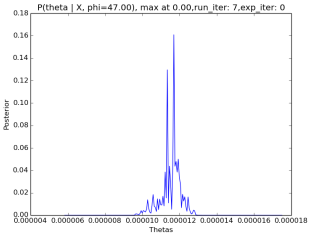

Fig. 16. Physics simulation: $\phi = 47$

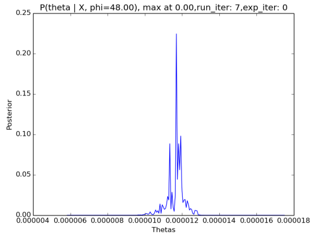

Fig. 17. Physics simulation: $\phi = 48$

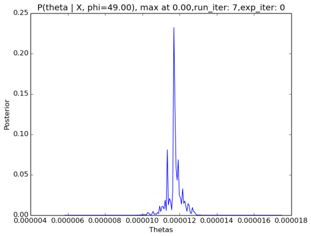

Fig. 18. Physics simulation: $\phi = 49$

7 Conclusion

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

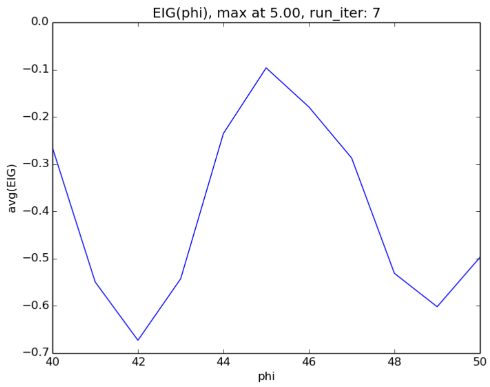

Fig. 19. Physics simulation: Expected Information Gain

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

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; Brenden Lake for helping us with Bayesian analysis; 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.

[2] L. Heinrich, *weinberg-exp*, https://github.com/lukasheinrich/weinberg-exp, 2016.