Bayesian Analysis Toolkit in Searches

Frederik Beaujean[1]*, Allen Caldwell*[1]*, Daniel Kollár*[2]*, Kevin Kröninger*[3]*, Shabnaz Pashapour*[3*]
[1] Max-Planck-Institut für Physik, [2] CERN,[3] Georg-August-Universität Göttingen
*Corresponding Author

Abstract
The Bayesian Analysis Toolkit, a software package for data analysis based on Bayes' theorem, is introduced. This toolkit takes advantage of Markov Chain Monte Carlo to find the full posterior probability distributions. The tool can easily be used for parameter estimation, limit setting and error propagation. Model comparison and goodness-of-fit estimation are realized in the package through well-established methods. In addition to a brief description of the Bayesian Analysis Toolkit, the use of this tool in searches is described in the example of Banff Challenge 2a problem 1.

1 Introduction

A comprehensive statistical interpretation is an essential part of any data analysis. Typically, one needs to compare model predictions with data, to draw conclusions on the validity of the model as a representation of the data and to extract values of parameters. It is not trivial to implement the required tools for such a task and usually individual researchers develop their own versions of these tools. Therefore, it is beneficial to have a set of common statistical tools and numerical algorithms that can be validated regularly and easily adapted to solve arbitrary problems.

One of the problems to be addressed is to search for the presence of a signal over some background. For example, in the current status of particle physics that the Higgs boson has not yet been observed and there is no clear direction on what nature has in store for us as the new physics, we need to have the ability to spot the smallest signals in a reliable way to continue our path in better understanding the workings of nature.

In this article, we describe the Bayesian Analysis Toolkit (BAT) [1, 2], its philosophy and functionalities as well as the use of this toolkit to address one of the Banff challenge problems [3].

2 The Bayesian Analysis Toolkit

The Bayesian Analysis Toolkit is a C++ software package to address statistical problems based on Bayes' theorem. It comes in form of a library working with ROOT [4], developed to provide the users with easy to implement methods. The tool has an interface with CUBA [5], MINUIT [6] and RooStats [7]. Bayes' theorem for a single model has the form

$$P(\vec{\lambda}|\vec{D}) = \frac{P(\vec{D}|\vec{\lambda})P_0(\vec{\lambda})}{\int P(\vec{D}|\vec{\lambda})P_0(\vec{\lambda})\,d\vec{\lambda}}\ , \tag{1}$$

i.e., the probability of parameter set $\vec{\lambda}$ given data $\vec{D}$, the *posterior* probability, is proportional to the probability of data given parameters, also known as the *likelihood*, times the initial probability for the parameters, the *prior* probability[1]. The denominator ensures the normalization of the posterior probability and is just the integral of the numerator over the allowed region of $\vec{\lambda}$. One can also interpret this formula as a learning rule stating that: The knowledge about the model and its parameters before the experiment, the prior, is updated using the probability of the new data for different values of the parameters, resulting in posterior knowledge.

The approach taken in the BAT technical development is to fulfill two main requirements:

[1] Throughout this article the term probability is used for both probability and probability density.

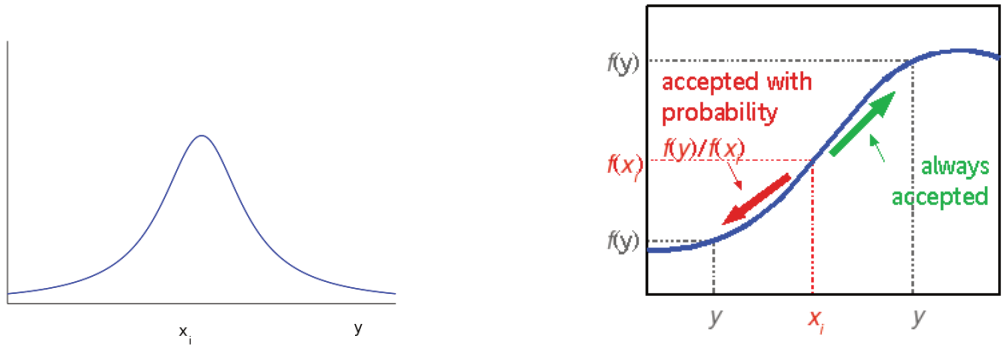

Fig. 1: The proposal function to pick the point to move to for the MCMC random walk (left), and a simple sketch to show the decision-making process in the MCMC process (right).

i) provide a flexible framework which allows formulation of arbitrary models,
ii) provide a reliable mapping of the full posterior probability density.

For each data analysis, there is a case specific part which includes the model and the data and the other part consists of the common tools. The user only needs to deal with the case specific issues and the rest is taken care of by the common tools in BAT. The user should create the model by defining the parameters, $\vec{\lambda}$, the likelihood, $P(\vec{D}|\vec{\lambda})$, and the priors, $P_0(\vec{\lambda})$, and read in the data. Common tasks such as normalization, mode finding, goodness-of-fit test, marginalization and presentation of the output in a nice format are handled by BAT functions. The key tool in BAT, allowing for the mapping of the posterior probability in multidimensional parameter space and the extraction of quantities of interest, is the Markov Chain Monte Carlo.

2.1 Markov Chain Monte Carlo

The feasibility of Bayesian inference has been revolutionized by the use of Markov Chain Monte Carlo (MCMC) (see e.g., [8, 9]). The MCMC can be used to obtain the posterior probability given by Eq. 1 which otherwise is generally a difficult task, specially for models with a large number of parameters. The MCMC can be employed to scan very complicated probability distributions in many dimensions through a random walk to points with higher probabilities in the allowed parameter space. The Metropolis algorithm [10] is the first and the most popular MCMC algorithm and is implemented in BAT. This procedure is followed to map out a function $f(\vec{x})$:

1. start at a random $\vec{x_i}$
2. generate a random point around $\vec{x_i}$, the proposal point, according to a proposal function,
3. calculate the values of the function at the current point, $\vec{x}_i$, and the proposal point, $\vec{y}$, and compare them:
   - if $f(\vec{y}) \geq f(\vec{x}_i)$, set $\vec{x}_{i+1} = \vec{y}$,
   - if $f(\vec{y}) < f(\vec{x}_i)$, set $\vec{x}_{i+1} = \vec{y}$ with probability $r = f(\vec{y})/f(\vec{x}_i)$,
   - if $\vec{y}$ is not accepted, stay where you are, $\vec{x}_{i+1} = \vec{x}_i$.
4. generate a new $\vec{y}$ around the new $\vec{x}$ (go to 2).

For an infinite number of steps, the $f(\vec{x_i})$ is gauranteed to converge to $f(\vec{x})$. However, for a finite number of steps, one has to check for convergence. Figure 1 shows the proposal function, a Cauchy distribution, and a schematic of how the MCMC works.

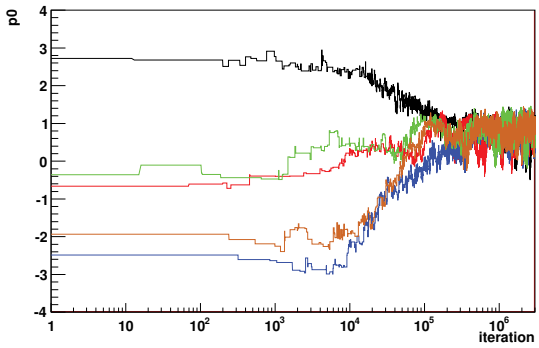

Fig. 2: Parameter value at each iteration for 5 Markov chains that shows the convergence of the 5 chains after aprroximately 10^5 iterations.

2.2 Convergence

To achieve the convergence of MCMC and find reasonable run parameters a *pre-run* is performed in BAT before the MCMC is used for the analysis of the posterior. In the pre-run phase, we use several chains in the parameter space in parallel. The steps in parameter space are done consecutively for each parameter and chain.

The set of steps from an update of the first parameter of the first chain to the last parameter of the last chain makes one iteration. The efficiency for accepting or rejecting new points is evaluated separately for each parameter and chain over a number of iterations. The proposal function is set to a Cauchy function by default and the width of the function is adjusted during the pre-run to match the required efficiency of the sampling. After a finite number of steps are taken, for example 1000 steps, we update the width of the proposal function to optimize performance, until an efficiency between 15% to 50% is reached for each parameter. Users can also define their own proposal function.

The convergence is determined based on the *R-value* [11] which should be about 1 once the convergence is reached. The R-value is the ratio of the current variance estimate to the within-sequence variance with a factor to account for the extra variance of the Student's t distribution. Figure 2 shows the parameter value, $p0$, in five different chains as a function of the iteration. Convergence, defined via the R-value, is reached after aprroximately 10^5 iterations.

2.3 Main Analysis Run and Functionality

The analysis run is performed for a defined number of iterations with run parameters found in the pre-run. At this stage all the scales are fixed and samples are collected for posterior analysis. At each step for each parameter a 1-dimensional (1-D) histogram is filled with the values of the parameter and the so-called marginalized distribution is obtained. This represents the posterior probability function of a single parameter of a model given data when all the other parameters are integrated over.

Similarly one can create 2-dimensional (2-D) histograms for every pair of parameters when all the other parameters are integrated over. This will show the correlation between the two parameters. The full output of the MCMC can be saved during the run for future analysis. Figure 3 shows an example of these distributions for a 6-parameter model. It enables you to compare all the 6 parameters and their correlations in one plot.

In addition to the posterior probability functions, BAT can be used for other statistical calculations. Evaluation of arbitrary functions of parameters is also implemented in BAT and typically is used for error propogation. Given that MCMC covers the full parameter space, the location of the global maximum is updated at each step. Therefore, a bi-product of the MCMC is the location of the global mode of the

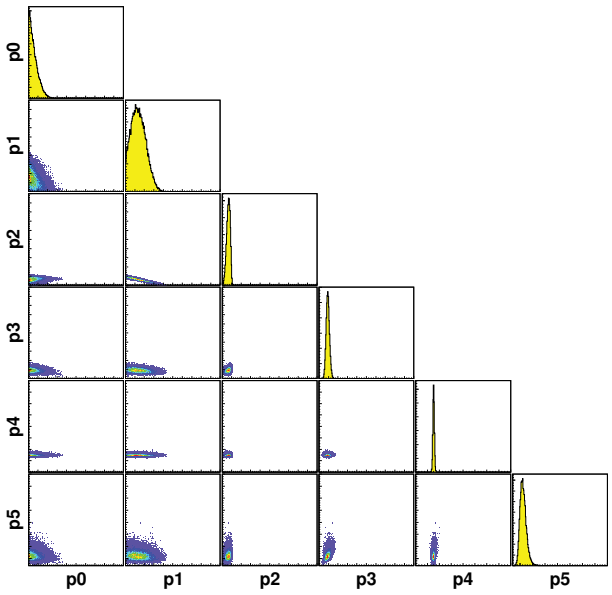

Fig. 3: An example plot : 1-D and 2-D marginalized distributions for a 6-parameter, p0 to p5, model, allowing one to easily examine the parameters and their correlations.

posterior probability. MCMC is not optimized for this task and as such the estimated mode is not accurate but it can be used as a good starting point for other minimization programs, e.g., MINUIT. A numerical integration over the posterior probability is also possible in BAT using the sampled mean algorithm with and without importance sampling. Alternatively, one can compile BAT with the CUBA library which allows the use of well-tuned routines for integration in many dimensions.

3 Signal Discovery

One of the most common tasks in data analysis is to look for the presence of a signal over some background. Usually the general form of background is known and there are predictions for the signal model. The task is to check for the presence of signal over the background given the data at hand. Here a method to search for signal using BAT is suggested. As an example the result of the Banff Challenge 2a Problem 1 is discussed.

A simple strategy to look for the signal is to define the models under investigation, a null hypothesis, $P_0(H_1)$, for background only case, a possible signal including the background, $P_0(H_2)$, such that the total probability of the two cases is one, $P_0(H_1) + P_0(H_2) = 1$. You also need to set the possible values for the parameters given a signal is present, $P(\mu, A, \sigma|H_2)$. Here, μ is the average value of the observed data, A is the rate for the background and σ is the width of the signal model. Variables μ, A, and σ are the parameters for the model. After defining the models, one can calculate the posterior probabilities for the hypothesis and check the validity of the models under investigations.

Once one decides on the choice of priors and probabilities, it is very straightforward to implement this simple strategy in BAT. One just needs to define the parameters and their ranges, the priors and the likelihood method and can read the data, all of which can be done by using the most basic class in BAT, `BCModel` class, or one of its inheritance. Afterwards, the functions available in BAT can be used to calculate the posterior probabilities and the p-value, to provide the marginalized 1D/2D distributions, the full MCMC output and to do many other tasks.

3.1 An Example - Banff Challenge 2a Problem 1

Several interesting statistical issues have been raised in the workshop at the Banff International Research Station on Statistical Issues Relevant to Significance of Discovery Claims [3]. These issues have been illustrated as specific examples to be addressed by the participants. One of these examples is to simulate the task of discovering a signal or new phenomena. The details of the Banff challenge can be found in [3].

In brief, two hypothesis have been considered, a background only case in the form of

$$B(x) = Ae^{-Cx} \,, \tag{2}$$

where x is the mark of the event and is between 0 and 1, $C = 10 \pm 0$, and the background rate is drawn from a truncated Gaussian distribution such that $A = 10000 \pm 1000$ and $A \geq 0$. The second hypothesis considers a signal, in addition to the above background, of the form

$$S(x) = De^{-(x-E)^2/2\sigma^2} \,, \tag{3}$$

where $D \geq 0$, and $\sigma = 0.03$ and, in the signals generated for the simulated data, the peak position E is between 0 and 1 exclusive of both sides. Participants are given 20K simulated datasets with a mixture of the two hypotheses to analyse and provide a yes-no decision whether a signal is to be claimed. The Type-I error should not be more than 0.01 and if a signal is claimed, the peak position and its 68% interval should also be reported. No prior was provided.

Our approach was to follow a Bayesian logic. A two-step decision procedure has been employed. We start with a background only fit to the data and calculate the p-value. If the p-value is greater than 1%, we discard the possibility of the signal presence, if it is less than 1%, we further analyse the data. We bin the data and choose our likelihood as the product of Poisson probabilities for each bin. We use BAT's fast Poisson p-value estimate corrected for the degrees of freedom [12] to get the p-value. For the background rate, we consider a truncated Gaussian distribution prior in a range of 0 to 20K. If we had made a signal discovery claim based on the p-value cut, we would have a Type-I error of 0.0138 ± 0.0009.

However, we do not make a claim based on p-value cut, instead we start with step two and calculate the probability of background only model, $P(B|D)$. Our prior for the background only model is chosen to be 0.95. In other words, we take a case where we have strong prior belief in the null hypothesis. Furthermore, we set the strict requirement that $P(B|D) < 0.001$ to claim evidence for the presence of a signal. Flat priors are assumed for E and D with respective ranges of 0 to 1 and 0 to 2000. The "Look Elsewhere Effect" is already taken care of with the choice of priors [13]. Our Type-I error for our final result is 0.0 which is an important result stating we have no false positive.

Using this strategy, 271 datasets were flagged as having signal. Figure 4 shows the measured peak position vs. the measured signal rate for the datasets claimed to have a signal. For the found signals, 63% of the measured peak positions and 50% of the signal rates fall within 68% of their true values.

The result will change by the choice of different priors, however, in the real experiment, you have to make a judgment based on how far out the new physics is believed to be.

4 Summary

The Bayesian Analysis Toolkit is introduced. The philosophy of its design, its functionalities and general characteristics are briefly discussed. The implementation and performance of Markov Chain Monte Carlo in BAT is described. As an example, the use of BAT in search for a signal is outlined and the result for the Banff Challenge 2a problem 1 is reported.

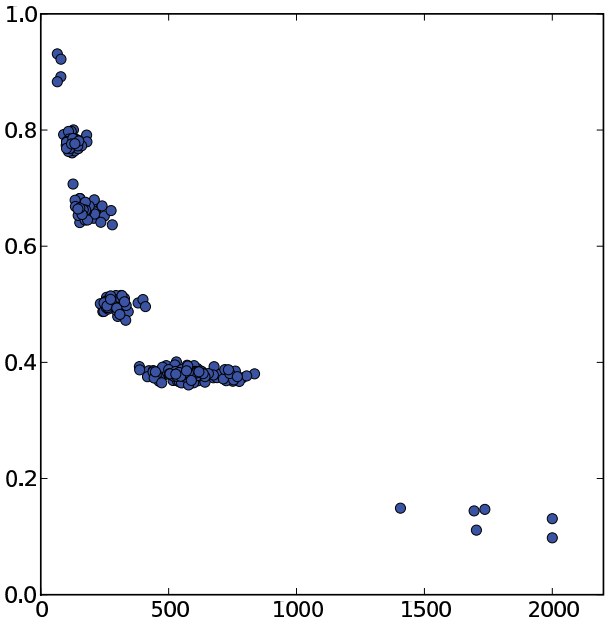

Fig. 4: The measured peak position, E, vs. the measured signal rate, D, for the datasets claimed to have a signal.

References

[1] A. Caldwell, D. Kollár and K. Kröninger, *BAT - The Bayesian Analysis Toolkit*, Comp. Phys. Comm. **180**, 2197 (2009).

[2] `http://www.mppmu.mpg.de/bat/`

[3] T. R. Junk, *Banff Challenge 2,* these Proceedings.

[4] R. Brun and F. Rademakers, *ROOT - An object oriented data analysis framework*, Nuclear Instruments and Methods in Physics Research **A**, 81 (1997).

[5] T. Hahn, *CUBA – a library for multidimensional numerical integration*, Comp. Phys. Comm. **168**, 78 (2005).

[6] F. James, *MINUIT - Function Minimization and Error Analysis*, CERN Program Library Long Writeup **D506** (1994-1998).

[7] W. Verkerke and D. Kirkby, *The RooFit toolkit for data modeling*, [arXiv:physics/0306116] (2003); L. Moneta *et al. The RooStats project*, [arXiv:1009.1003v2] (2011).

[8] S. Karlin and H. Taylor, *A first course in Stochastic processes*, Academic Press (1975).

[9] W.R. Gilks, S. Richardson and D.J. Spiegelhalter, *Markov Chain Monte Carlo in Practice*, Chapman (1996).

[10] N. Metropolis *et al. Equation of State Calculations by Fast Computing Machines*, J. Chem. Phys. **21**, 1087 (1953).

[11] A. Gelman and D.B. Rubin, *Inference from iterative simulation using multiple sequences*, Statistical Science **7**, 457 (1992).

[12] F. Beaujean *et al.*, *p-values for model evaluation*, Phys. Rev. D **83**, 012004 (2011).

[13] A. Caldwell, *Signal discovery in sparse spectra: a Bayesian analysis*, these Proceedings.