Robust Methods of Error Estimations and Bayesian Curve Fits

Bryan Clark

Carnegie Mellon University, Pittsburgh, Pa, 15213

Abstract

In this paper we explore a variety of different methods for robust error estimations while doing bayesian curve fitting. This exploration is done specifically with the example of Lattice QCD simulations. We show that quadratic approximations, numerical integration, and bootstrap methods all give similar results for the values and errors on parameters and examine some advantages/disadvantages of each method.

1. Introduction

One of the important links between taking data from experiment and establishing the parameters or validity of a theory is being able to fit the theory to the data. This procedure typically involves two pieces. First, one needs a model. Secondly, one then needs to find the parameters of that model that fit the data the best. In this paper, we will discuss the latter of these two steps.

A central concern is calculating the errors that are associated with these values. This is especially important in simulations where knowing the errors is integral in understanding how close your simulation has come to reproducing (or predicting) the empirical results. One example of this is in numerical QCD. In QCD simulations, the result is a series of gauge field configurations that result from a metropolis algorithm. Various quantities are computed from these configurations and the results are then averaged and taken as data. In this paper we examine "meson propagators" which are modeled by a function of the form

$$G(t) = \sum_i a_i^2 e^{-E_i t} \tag{1}$$

where the E_i are energies and the a_i are the amplitudes. In Section 2 we will discuss three different methods of establishing parameters and their errors. These methods will include calculating the values by numerical integration, bootstrap, and using a gaussian approximation. In Section 3 we will compare these three methods.

2. Error Analysis

Given a set of data D, we would like to find the most probable parameters T for our model. In other words, we want to maximize the probability $P(T|D)$. A standard approach is to minimize the chi squared between the data and a given theory: $\chi^2 = \sum_t \frac{(G_{mc}(t)-G_T(t))^2}{\sigma_G^2}$. The problem with this approach is that our theory has an infinite number of parameters. Since we are fitting to an energy spectrum, there are an infinite number of energy levels. Yet, you only have a finite amount of data to fit with. If the infinite number of energies were relevant in fitting the problem, the problem would not be tractable. For our specific

problem, though, we know a priori that only a finite number of energies are relevant. This is because we know that all the a_i are approximately the same, but the E_i grow steadily with i . Nonetheless, there are two pieces of information that we don't know a priori. We don't know how many parameters are going to be needed to fit the data. Secondly, we don't know how the effect of the terms that aren't fit will effect the uncertainty of the final errors.

One way to resolve these two issues is to calculate $P(T|D)$ from basic probability theory [1]. Bayes rule states that the probability of obtaining theory T given data D is

$$P(T|D) = \frac{P(D|T)P(T)}{P(D)}.$$

We know that $P(D|T)$ is equal to $e^{-\frac{1}{2}\chi^2}$. (This is the basis of the standard method for minimizing χ^2). The $P(D)$ is simply a normalization constant that can be ignored since it is independent of T. The factor $P(T)$ is the probability that the theory is correct before considering the data. In Bayesian analysis this is called the prior probability, because it is here we were introduce information that we have a priori about the probability of different theories. The only information that we have about the distribution of our parameters is its average value as well as a guess at its standard deviation. Given these values, it makes sense to choose a gaussian distribution as our prior. Now let us take the logarithm of this probability distribution to create a

$$\chi^2_{aug} = \chi^2_{old} + \sum \frac{(\omega - \omega_{th})^2}{\sigma^2}$$

where ω is the prior average and σ^2 is the standard deviation of each of your parameters. To calculate the value of some parameter E_1 one computes the average and standard deviation of the energy with respect to $P(T|D) \propto e^{-\chi^2_{aug}/2}$ by calculating

$$\langle E_1 \rangle = \frac{\int dE_1 dE_2 ... dE_n e^{-\chi^2_{aug}/2} E_1}{\int dE_1 dE_2 ... dE_n e^{-\chi^2_{aug}/2}} \tag{2}$$

$$\sigma^2_{E_1} = \langle E_1^2 \rangle - \langle E_1 \rangle^2.$$

These integrals are impossible to evaluate analytically – they are high dimensional integrals and they have complicated integrands. I have explored three approaches for approximating these integrals.

The first of these methods is to calculate the integral exactly using numerical techniques. I did this with the use of a Metropolis algorithm. The Metropolis algorithm starts with an initial set of parameters; I used an estimate of the minimum of χ^2_{aug}. From this point, it proceeds to try to take a step in the parameter space. If this step lowers the value of chi squared it takes the step. On the other hand, if it increases the value it only takes the step with probability $e^{-\delta\chi^2_{aug}/2}$. This algorithm generates random points in parameter space that have probability density proportional to $e^{-\chi^2_{aug}/2}$. The averages in equation 2 are then straight averages of the metropolis results. The metropolis algorithm is halted when the errors stop growing as the square root of the bin size.

The second method that we utilized was to assume that $\chi^2_{\rm aug}$ is approximately quadratic in the parameters in the region about the minimum of $\chi^2_{\rm aug}$. This makes the integrals in

equation 2 into gaussians and allows us to compute them analytically. This results in being able to calculate any function f of the parameters with the formulas

$$\langle f \rangle = f(w^*)$$

$$\sigma_f^2 = \sum_{ij} C_{ij} \frac{\partial f}{\partial \omega_i} \frac{\partial f}{\partial w_j} \bigg|_{\omega^*}$$

$$(C^{-1})_{ij} = \frac{1}{2} \frac{\partial^2 \chi^2_{aug}}{\partial \omega_i \omega_j} \bigg|_{\omega^*}$$

where ω^* is the minimum of χ^2_{aug} . [2]

Our third approach uses a bootstrap method. The minimization is repeated for a large number of bootstrapped copies of the data and the priors. The different minimums are distributed in parameter space approximately as $P(T|D)$ Then to get the distribution from this value, we bootstrap both the data and the priors that are used in the calculation of χ^2_{aug} . Bootstrapping the data is done by creating a new set of configurations by choosing a random set (with replacement) of the same size from the old set of configurations. The priors are bootstrapped by choosing a new mean for the prior randomly from a gaussian distribution that has means and standard deviations taken from the original prior. Note that in a typical bootstrap you only bootstrap the data. When you are using Bayesian analysis, though, it is important that you bootstrap the priors in order to come to correct conclusions.

The one limitation of the bootstrap approach is that it tends to allow exponentials to ocassionally drop out of the fit. The reason that this happens is that the combination of a low prior and a large σ can allow the value of an amplitude to touch 0 without harming χ^2 significantly. Since we are testing how many exponentials we need to fit, we will be fitting more exponentials then we need to and consequently χ^2 is also not significantly altered when an exponential drops out. Therefore sometimes the minimization will find some amplitude of 0. In these cases, we have eliminated these outliers from our error calculations as it is not an accurate representation of the error on this parameter.

3. Comparison

The actual data that we used to explore these different methods is the calculation of the mass of the Upsilon particle. This data was taken from a lattice gauge simulation. The model that was utilized is the one mentioned in equation 1. The specific parameterization we used to fit the data was one where

$$\zeta_{i\in\{1..n\}} = e^{a_i} \quad \zeta_{i\in\{n+1..2n\}} = e^{\sum_{j=1}^{i} E_j}$$

This was defined to prevent the amplitudes from going negative as well as keeping the energies appropriately ordered. We fit with $n \in \{4, 5, 6\}$ which corresponds to fitting with four, five, or six exponentials. For each of these, we ran the metropolis as well as the bootstrap until they had converged.

The first thing to examine is how closely the different methods agree. The different results that each method gets for the parameters and errors are demonstrated in figure 1. This figure is a plot of the ten parameters (five energy parameters, five amplitudes) associated

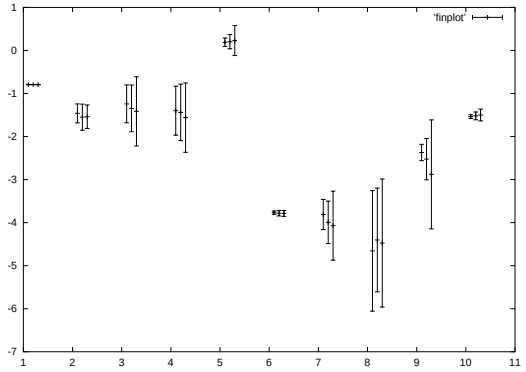

FIGURE 1. Comparison of Quadratic Approximation, Metropolis, Bootstrap for a fit of ten parameters. Each bunch of three is the average value and standard deviation of one parameter for the three methods in the above order.

with fitting $n = 5$ exponentials. Each group of three points represents one parameter with average values and errors for the quadratic approximation, the metropolis, and the bootstrap repsectively. Notice that for the first two energies and amplitudes the values and errors are almost identical.

Moreover, not only are the actual values that result fairly close, but the distributions also are extremely similar. In figure 2 , there is the distribution for the different energy levels and amplitudes with all three different methods. The line represents the curve from the quadratic approximation, the points represent the metropolis and the histogram represents the values from the bootstrap methods. Notice for the lower energy and amplitudes, the three distributions are very similar. This is especially true of the bootstrap and metropolis as they both track the tail of the distribution. For the higher energies and amplitudes the values are not as well determined. In this case, there is still correspondence between the different methods, but it is not as well correlated. This is effectively because these values have more room to wander without effecting chi squared significantly. Nonetheless we again have a situation where the bootstrap and metropolis agree significantly better then the curvature values.

The reason for this is that the errors based on curvature depend on the assumption that the nature of the integral at its minimum is well approximated by a quadratic function. We can demonstrate that in a number of cases this approximation is not valid. If this approximation were valid we could examine any two dimensional projection and that projection should look elliptical. Figure 3 demonstrates the contours that contain 66% and 95% of the points that a converged metropolis visited. This is for the 2 dimensional projection into the 3'rd energy parameter and 4'th amplitude parameter. The center of the box is the minimum of χ^2_{aug} and the box is scaled so that it is three σ in each direction where the σ is calculated by the local quadrature. This structure should mirror the actual structure of the integral. The quadratic approximately is clearly not a valid one in this case.

Another thing to be cognizant of is the metropolis takes a significant time to converge because it is attempting to calculate a high dimensional integral. This is because the high dimensional structure that results appears to have a variety of ridges that can cause the

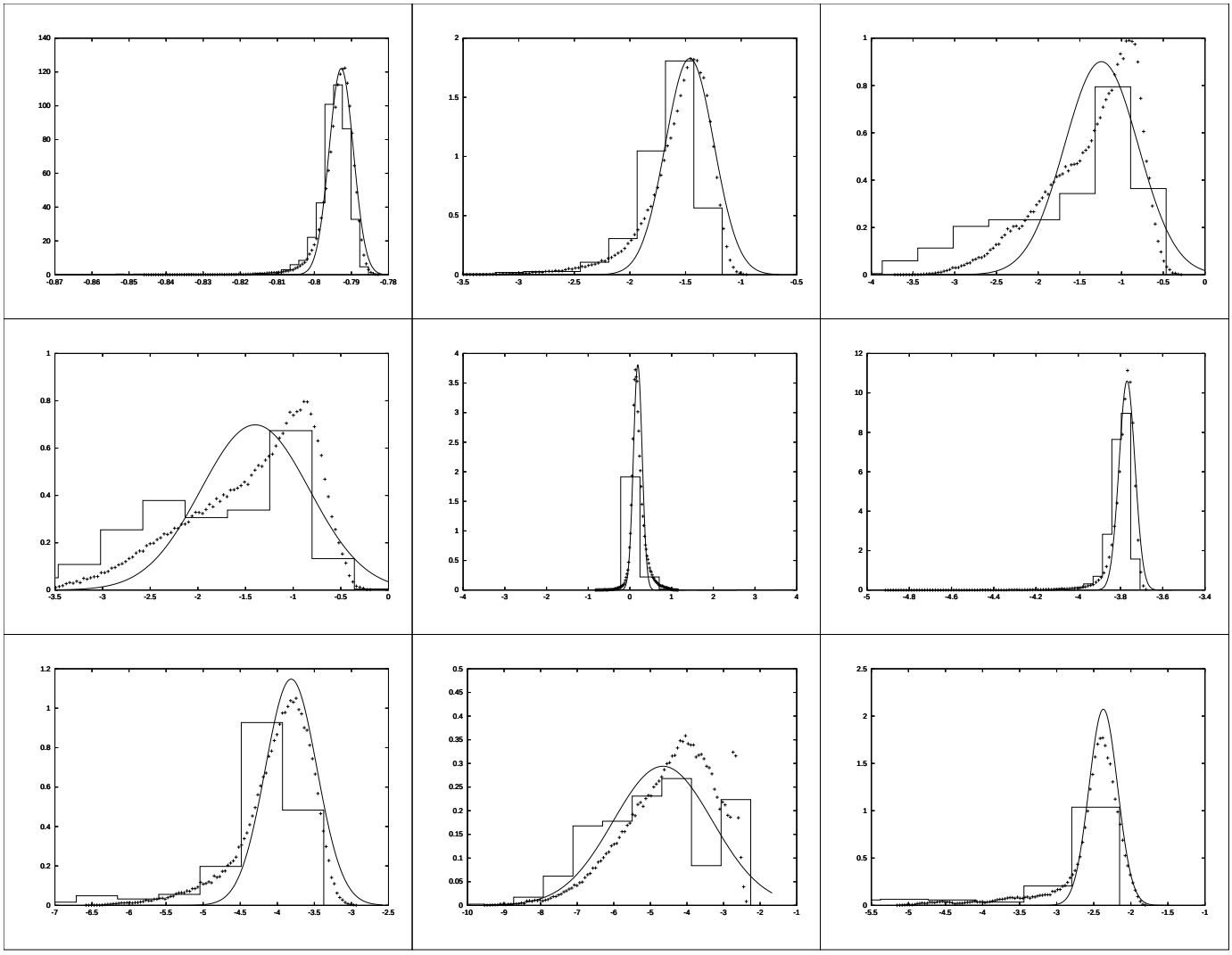

FIGURE 2. Normalized density of points for curvature, metropolis, and bootstrap

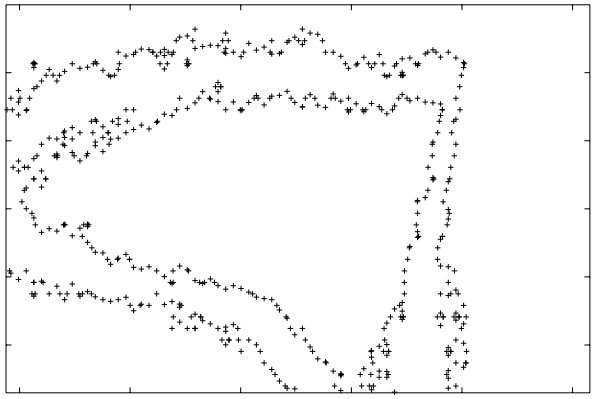

FIGURE 3. 2d projection of metropolis for 66% and 95% contours.

metropolis to get stuck for an extended period of time. In general, the less well constrained by the data a parameter tends to be, the more likely it is that the structure in that dimension is going to be highly non-regular and cause problems for the metropolis. For $n = 4$ we began seeing convergence at approximately 1,000,000 points. It took on the order of 6,000,000 for $n = 5$ to converge and approximately $50,000,000$ for $n = 6$ to converge. This is in sharp contrast to the bootstrap algorithm which scales better. This is because it appears that 1000 bootstraps successfully converges the values no matter how many exponentials there are. The reason for this significant distinction between these two methods is the bootstrap method is getting independent values every iteration, whereas in the metropolis algorithm the values are related to each other. Of course, the method that scales the best is utilizing the gaussian approximation as this only requires one run of the minimization routine in order to achieve the desired result.

Overall, we find that the errors calculated in different manners from bayesian analysis seem to match each other fairly closely. This is especially true of the bootstrap and the metropolis algorithms. In general, these two methods have more credence then the utilization of the guassian approximation because they do not need to assume that the value is quadratic. In the cases where the bootstrap and metropolis disagree, it is not clear which error estimate ought to be used. There are different approximations that go into each estimator and both sets of approximations seem valid. In general, it seems to make sense to use the bootstrap method as this method can be run faster.

Acknowledgments

I am pleased to acknowledge Professor Lepage of Cornell University for guiding me in the work on this Research Experience for Undergraduates project. This work was supported by the National Science Foundation REU grant PHY-0097595 and research grant PHY-9809799.

Footnotes and References

1. Sivia, D. S. Data Analysis: A Bayesian Tutorial. Oxford: Clarendon Press, 1996.
2. G.P. Lepage (private communication)