October 22, 2018, 17 August 2018

Extracting Excited States from Lattice Correlation Functions

Kimmy Cushman

1 Motivation

1.1 Correlation Functions

Lattice QCD is a numerical method for understanding strongly coupled gauge theories at energies where perturbation theory breaks down ($\alpha_s > 1$). We are interested in studying the structure of bound states in Nature, about which information is gained from scattering experiments. Lattice gauge theory is however computationally bound by our finite computing resources so we must perform lattice simulations of strong forces on a finite volume of space-time points, $V = L^3 \times N_t$ with finite lattice spacing, a. Physical parameters are calculated in the infinite volume $V \to \infty$ and continuum $a \to 0$ limits.

We are interested in computing the energy spectrum of QCD states, as well as matrix elements of these states.

For example, the pion π is the ground state of QCD with the quantum numbers (J^{PC}) $\pi^{\pm} = 0^{-\pm}, \pi^0 = 0^{-0}$, having $J = 0$ angular momentum and negative parity $P = -$ (a pseudoscalar meson) with $C = \pm 1$ or 0 charge. The CKM matrix of flavor mixing is of interest to the standard model (SM) because it contains four of the SM parameters, including one CP-violating phase, which is of interest to cosmologists as well. Lattice QCD can give estimates for the CKM matrix elements because weak decays involve these as flavor mixing factors. For example the rate of pion decay is proportional to the ud mixing as

$$\Gamma(\pi \to l\nu) = \frac{G_F^2 |V_{ud}|^2 f_\pi^2}{8\pi} m_\pi m_l^2 \Big(1 - \frac{m_l^2}{m_\pi^2}\Big)^2. \tag{1}$$

Thus, the lattice can compliment experimental measurements to reduce the uncertainty of SM parameters. Matrix elements and energy states are calculated on the lattice by analyzing correlation functions.

1.2 Excited states

In many cases, only the ground state of a particular operator are of interest. However, lattice computations are extremely expensive, and being able to extract information about excited states would allow us to squeeze out as much information as possible from our calculations.

1.3 Correlation functions

Consider a two point function of say, a pion, $\pi(t)$ and $\pi(0)$:

$$C_\pi(t) = \langle 0|\pi(t)(0)\pi(0)|0\rangle \tag{2}$$

$$= \sum_m^\infty \langle 0|\pi(0)|E_m\rangle \mathrm{e}^{-E_m t}\langle E_m|\pi(0)|0\rangle \tag{3}$$

$$\boxed{\Rightarrow C(t) = \sum_m^\infty a_m \, \mathrm{e}^{-E_m t}} \tag{4}$$

where we insert a complete set of eigenstates having the same quantum numbers as the piom. Note that the operators are necessarily simultaneously diagonalizable in this basis because they belong to the same symmetry sector of the Hamilton, or else the two point correlation would be zero. For example, we can't create a particle with the quantum numbers of a pion and then annihilate a proton.

1.4 Effective mass

It is "easy" to find the ground state energy, E_1, given correlation function data, $C(t)$:

$$C(t) = a_1 \mathrm{e}^{-E_1 t} + a_2 \mathrm{e}^{-E_2 t} + \cdots \tag{5}$$

$$= a_1 \mathrm{e}^{-E_1 t}\Big(1 + \frac{a_2}{a_1}\mathrm{e}^{-(E_2 - E_1)t} + \cdots\Big) \tag{6}$$

$$= a_1 \mathrm{e}^{-E_1 t}\Big(1 + \frac{a_2}{a_1}\Delta_2^t + \frac{a_3}{a_1}\Delta_3^t \cdots\Big) \tag{7}$$

where $1 > \Delta_2 > \Delta_3 > \cdots$. Thus, as time goes on, the contributions from the excited states become more and more negligible. Therefore,

$$\frac{C(t+1)}{C(t)} \approx \frac{a_1 \mathrm{e}^{-E_1(t+1)}}{a_1 \mathrm{e}^{-E_1 t}} \tag{8}$$

$$= \mathrm{e}^{-E_1} \tag{9}$$

$$\boxed{\Rightarrow E_1 = \log\Big(\frac{C(t)}{C(t+1)}\Big)} \tag{10}$$

The ground state energy is the found by taking the log of the ratio of the correlation function from consecutive time slices. As stated before, this approximation becomes more accurate at longer times. But the correlation function is falling exponentially so the singal to noise may also be plummeting. Therefore, the sought after so-called **"plateau"** is the region in time where the excited state contamination is small, but the error bars have not yet blown up. See Fig. 1 for an example.

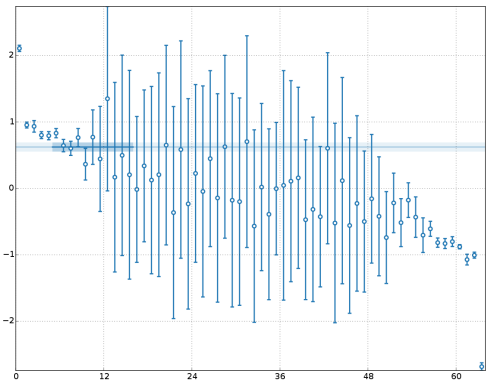

Figure 1: Effective mass plot m_{eff} vs. t with large errors increasing for $t \approx N_t/2$.

1.5 Black-box method

For a given correlation function, $C(t)$, consider modeling the function as a sum of M states. Then define

$$\begin{aligned} y_n(t) &\equiv C(t+n) \\ &= \sum_m^M a_m \mathrm{e}^{-E_m(t+n)} \\ &= \sum_m^M a_m \mathrm{e}^{-E_m t}\mathrm{e}^{-E_m n} \end{aligned} \tag{11}$$

$$\boxed{\Rightarrow y_n(t) = \sum_m^M A_m(t) z_m^n} \tag{12}$$

To extract M energies and M amplitudes, we need $2M$ data points, $y_0, \cdots, y_{2M-1}$. For a given starting time, t, these are a set of $2M$ equations which can be written

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{2M-1} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & \cdots & 1 \\ z_1 & z_2 & \cdots & z_M \\ z_1^2 & z_2^2 & \cdots & z_M^2 \\ \vdots & \vdots & \ddots & \vdots \\ z_1^{2M-1} & z_2^{2M-1} & \cdots & z_M^{2M-1} \end{pmatrix}}_{\text{Vandermode matrix}} \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_M \end{pmatrix} \tag{13}$$

Prony's method is to solve for the z's first algebraically, which is a non-linear problem, then go back and solve the linear Vandermonde system.

We solve for the z's which encode the energies by solving the $(2M+1) \times (2M+1)$ Hankel matrix determinant

$$0 = \begin{vmatrix} y_0 & y_1 & \cdots & y_{M-2} & y_{M-1} & 1 \\ y_1 & y_2 & \cdots & y_{M-1} & y_M & z \\ y_2 & y_3 & \cdots & y_M & y_{M+1} & z^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \underbrace{y_M \quad y_{M+1} \quad \cdots \quad y_{2M-2} \quad y_{2M-1}}_{\text{Hankel matrix}} & & & & & z^M \end{vmatrix} \tag{14}$$

The problem of solving for the energies is then to solve for the roots of a polynomial. This is a black-box method since we do not input any initial guesses to what the energies and amplitudes are, as with fitting to exponentials.

2 Why Clustering?

We solve the non-linear problem (Hankel matrix determinant) to find the z's and then the linear problem 12 ti find the A's.

Many calculations of a given $C(t)$ are done by making many lattice computations using different gauge configurations.(We get same physics from a different vacuum state, which we generate according to the SU(3) guage.

We have a set of $N > 100$ $\{z_m, A_m\}$ samples. **But our data is noisy**

How do we find the mean and error of the energies and amplitudes when the points are not colored red and blue?

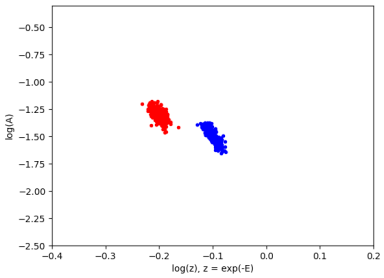

Figure 2: What we wished our data looked like

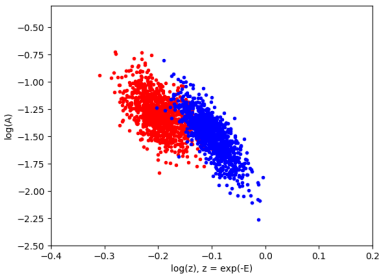

Figure 3: What our data might actually look like

3 Expectation Maximization Clustering

Given a Gaussian bivariate distribution of data points (like the two clusters above) define the mean μ, and the covariance matrix Σ:

$$\vec{\mu} = \langle \begin{pmatrix} x \\ y \end{pmatrix} \rangle \tag{15}$$

$$\Sigma_{xx} = \sigma_x^2 = \langle (x - \mu_x)^2 \rangle \tag{16}$$

$$\Sigma_{yy} = \sigma_y^2 = \langle (y - \mu_y)^2 \rangle \tag{17}$$

$$\Sigma_{xy} = \Sigma_{yx}^{\dagger} = \langle (x - \mu_x)(y - \mu_y) \rangle \tag{18}$$

Note that $\Sigma_{xy} \neq \sigma_x \sigma_y$ in general. The probability distribution of a bivariate Gaussian is given as

$$\boxed{p(\vec{x}) = \frac{1}{2\pi\sqrt{|\Sigma|}} \exp\big[-\frac{1}{2} (\vec{x} - \vec{\mu})^T \Sigma^{-1} (\vec{x} - \vec{\mu}) \big]} \tag{19}$$

3.1 The algorithm

We use initial cluster assignment by the magnitude of $z_1, z_2, \cdots z_M$ for each sample data set.

1. Assign points M initial clusters
2. Compute the mean and covariance matrix for each cluster
3. Reassign each point, $\vec{x}$ to the cluster than maximizes $p(\vec{x})$.
4. Repeat 2. and 3. until the process converges

Note that when we are considering maximizing the probability for a point to be in cluster 1 with μ_1 and Σ_1 versus cluster 2 with μ_2 and Σ_2, we can simplify the calculation by noting if the point $\vec{x}$ should be assigned to cluster 1, then

$$p_1(\vec{x}) > p_2(\vec{x}) \tag{20}$$

$$\Rightarrow \frac{1}{\sqrt{|\Sigma_1|}} \mathrm{e}^{-\frac{1}{2}(\vec{x} - \vec{\mu_1})^T \Sigma_1^{-1} (\vec{x} - \vec{\mu_1})} > \frac{1}{\sqrt{|\Sigma_2|}} \mathrm{e}^{-\frac{1}{2}(\vec{x} - \vec{\mu_2})^T \Sigma_2^{-1} (\vec{x} - \vec{\mu_2})}, \tag{21}$$

taking log of both sides,

$$-\frac{1}{2}(\vec{x} - \vec{\mu_1})^T \Sigma_1^{-1} (\vec{x} - \vec{\mu_1}) - \frac{1}{2}\log|\Sigma_1| > -\frac{1}{2}(\vec{x} - \vec{\mu_2})^T \Sigma_2^{-1} (\vec{x} - \vec{\mu_2}) - \frac{1}{2}\log|\Sigma_2| \tag{22}$$

$$\boxed{\Rightarrow d_1(\vec{x}) + \log|\Sigma_1| < d_2(\vec{x}) + \log|\Sigma_2|}, \tag{23}$$

where $d(\vec{x})$ is the distance between $\vec{x}$ and $\vec{\mu}$ in standard deviations. In a one dimensional distribution this would be $(x - \mu)/\sigma$.

The point is assigned to the cluster than minimizes the distance in standard deviations, with a penalty ($+\log|\Sigma_i|$) for a cluster being very large.

We use a variation of this algorithm that looks at each sample individually. For M states extracted from the correlations functions, we know we get M points going into M unique clusters.

3.2 Example

For more dramatic change in the clusters, I randomly initialized the clusters. The true distributions for the data are

$$\vec{\mu}_1 = \begin{pmatrix} -0.2 \\ -1.3 \end{pmatrix} \qquad \Sigma_1 = 10^{-3} \begin{pmatrix} 1 & -3 \\ -3 & 30 \end{pmatrix} \tag{24}$$

$$\vec{\mu}_2 = \begin{pmatrix} -0.1 \\ -1.5 \end{pmatrix} \qquad \Sigma_2 = 10^{-3} \begin{pmatrix} 1 & -5 \\ -5 & 40 \end{pmatrix} \tag{25}$$

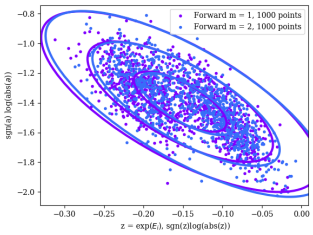

Figure 4

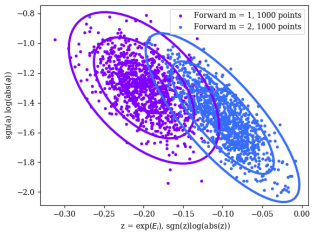

Figure 5

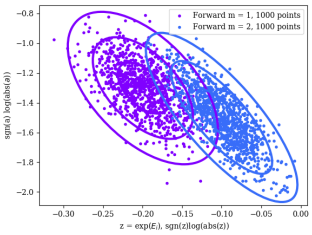

Figure 6

This data set converges with only 2 iterations of the modified algorithm. The means and standard deviations go through the following evolution

$$\vec{\mu}_1 = \begin{pmatrix} -0.154 \\ -1.393 \end{pmatrix} \rightarrow \begin{pmatrix} -0.199 \\ -1.300 \end{pmatrix} \rightarrow \begin{pmatrix} -0.200 \\ -1.301 \end{pmatrix} \tag{26}$$

$$10^3 \times \Sigma_1 = \begin{pmatrix} 3 & -8 \\ -8 & 41 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & -3 \\ -3 & 29 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & -3 \\ -3 & 29 \end{pmatrix} \tag{27}$$

$$\vec{\mu}_2 = \begin{pmatrix} -0.146 \\ -1.408 \end{pmatrix} \rightarrow \begin{pmatrix} -0.100 \\ -1.500 \end{pmatrix} \rightarrow \begin{pmatrix} -0.100 \\ -1.499 \end{pmatrix} \tag{28}$$

$$10^3 \times \Sigma_2 = \begin{pmatrix} 4 & -9 \\ -9 & 43 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & -5 \\ -5 & 36 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & -5 \\ -5 & 36 \end{pmatrix} \tag{29}$$

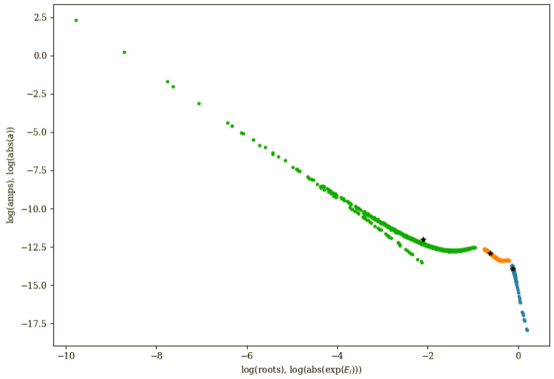

Figure 7: Example of data we are dealing with

3.3 Our Data

Clearly not Gaussian! The concept of "distance in standard deviations" relies on Gaussian data. What if we consider a different metric for determining which cluster to assign points to?

4 Tukey Depth

Tukey depth is a generalization of one-dimensional percentiles to multidimensional data.
From Tukey's original paper: 1974

1. "In the plane, an (i, j) line will be any directed line with $\geq i$ points to its left or on it and $\leq j$ points strictly to its left.
2. For any (i, j) the set of (i, j) lines is closed.
3. For any $i < n$ there is one and only one $(i, i-1)$ line in a given direction, one line of depth. Thus the $(i, i-1)$ lines form a closed curve of lines of depth i, the i-loop.
4. If $j < i - 1$, the set of (i, j) lines is finite.
5. The set of $(i, i-2)$ lines forms a closed polygon, the $(i - \frac{1}{2})$-polygon (all its sides belong to both the $(i-1)$-loop and the i-loop"

$$\vdots$$

"The midpoints of the segments cut off by the $(i - \frac{1}{2})$-polygon from the extensions of the sides of the $(i + \frac{1}{2})$ - polygon define a new, intermediate polygon, the i-polygon."

4.1 Our data in Tukey Depth

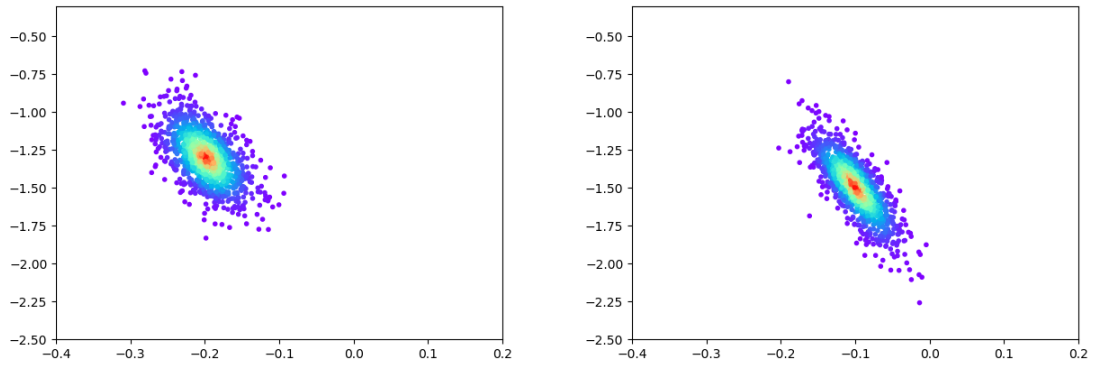

Figure 8: The same two clusters where each point is assigned a Tukey depth.

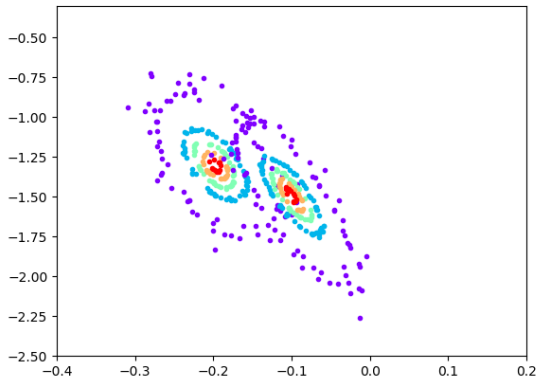

Figure 9: We can use the Tukey depth as a measure of fitness to a cluster.

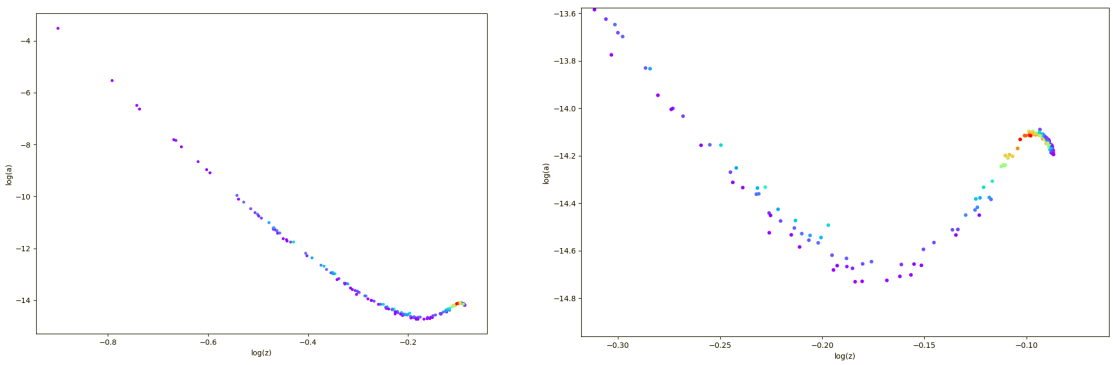

Figure 10: An example of non-Gaussian data where each point is assigned a Tukey depth.