Maximum entropy method for determination of density of states

Enzhi Li

January 6, 2017

1 Maximum entropy method

Maximum entropy method is widely used to extract the density of states (DOS) from the knowledge of Matsubara Green function, whether be it in imaginary time space or imaginary frequency space. It is a method that is based upon Bayesian analysis, and can give us the best possible inference given incomplete information. A detailed review of this method is already available in [1]. Here, I will give only a brief introduction

In this note, I will determine the DOS from the imaginary frequency Matsubara Green function. The spectral function, which is proportional to the DOS, is related with the Green function through

$$\begin{aligned} G(i\omega_n) &= \int_{-\infty}^{\infty} d\omega \frac{A(\omega)}{i\omega_n - \omega} \\ &:= KA(i\omega_n) \end{aligned} \tag{1}$$

Here, $\omega_n = \frac{(2n+1)\pi}{\beta}$ is a discrete variable, and ω is a continuous variable, K is the kernel function, and KA is a short-hand notation for the integration over ω. From the above equation, we can easily determine the value of $G(i\omega_n)$ once we know the spectral function $A(\omega)$. However, in reality, what we know is the imaginary frequency Green function, which can be computed from quantum Monte Carlo simulation, and what we want to extract is the spectral function that allows us to compare our numerical results with experiments. Direct inversion of Equation [1] is not viable, since the rapid decrease of the spectral function at large frequency region renders the Green function insensitive to the details of the spectral

function. Thus, infinitely many spectral functions may correspond to a single Green function. In order to solve this conundrum, maximum entropy (MaxEnt) method is devised. In the framework of MaxEnt, what we get is not the precise density of states corresponding to a Matsubara Green function, rather, it the most probable DOS that can be obtained from the existing incomplete and noisy information. In order to do this, we will use Bayesian inference, within which framework, our search of the most probable DOS can be formulated as, how to maximize the probability of getting some spectral function given the knowledge of the imaginary frequency Green function? This probability can be denoted as $p(A|\bar{G})$, and according to the Bayesian inference, we have the relation

$$p(A|\bar{G}) = \frac{p(A)p(\bar{G}|A)}{p(\bar{G})} \tag{2}$$

According to [1], $p(A) = \exp(\alpha S)$, and $p(\bar{G}|A) = \exp(-\frac{1}{2}\chi^2)$. From these, we define a quantity Q,

$$Q = \alpha S - \frac{1}{2}\chi^2. \tag{3}$$

It is clear that maximization of the probability $p(A|\bar{G})$ is equivalent to maximization of Q. Here, α is a real number that is introduced for annealing purpose, S is the entropy, and χ^2 is the difference between the actual value of Green function and the expected value of Green function. This formula can be understood as a generalization of the least square method. When α is zero, maximization of Q is equivalent to minimization of the difference between the real data and the fitted data. When $\alpha \to \infty$, Q is dominated by entropy, and the maximization of Q is precisely the maximization of entropy. We can interpret the α as temperature, and the $\frac{1}{2}\chi^2$ as the energy. With these interpretations, Q can be interpreted as the negative free energy. Thus, maximization of Q is just the minimization of the free energy. The entropy is defined as

$$S = -\int_{-\infty}^{\infty} d\omega A(\omega) \log \frac{A(\omega)}{D(\omega)}, \tag{4}$$

where, $D(\omega)$ is the default model, and χ^2 is defined as

$$\chi^2 \;=\; (\bar{G} - KA)^\dagger C^{-1} (\bar{G} - KA) \tag{5}$$

In order to understand this formula, some explanation of the Monte Carlo process is needed. The Matsubara Green function G is obtained from Monte Carlo measurements. In order to eliminate the correlations between two successive measurements, we have binned the data.

Assume that we want to make 1000 measurements. We will not make these measurements all at once. Rather, we will make, say 20 measurements in bin one, and another 20 measurements in bin two, etc, until finally we have 20 measurements in bin 50. We average the 20 measured Green functions in bin one, and call it $G^{(1)}$, and similarly in bin two, we get $G^{(2)}$, etc, until we get $G^{(50)}$. The $\bar{G}$ that appears in Equation [5] is the average of these 50 bin Green functions. That is, $G = \frac{1}{N}\sum_{i=1}^{N} G^i$, here N is the bin number. From the binned Green functions, we can calculate the covariance matrix C. The definition of the covariance matrix is

$$C_{mn} = \frac{1}{N(N-1)} \sum_{i=1}^{N} \Big(G^{(i)}(i\omega_m) - \bar{G}(i\omega_m) \Big)^* \Big(G^{(i)}(i\omega_n) - \bar{G}(i\omega_n) \Big) \tag{6}$$

From the definition, we see that the covariance matrix describes the correlations between adjacent Matsubara frequencies. It is easy to show that the matrix C is Hermitian, that is, $C^\dagger = C$. For sake of the simplicity of notations, we will denote $\xi = \bar{G} - KA$. Since ξ is a complex vector, we can rewrite it as $\xi = \xi_R + i\xi_I$. Here, ξ_R and ξ_I are the real and imaginary part of the vector ξ, respectively. In a similar fashion, the inverse of the covariance matrix can also be rewritten as $C^{-1} = C_R^{-1} + iC_I^{-1}$. With these notations, Equation [5] can be recast into the form

$$\begin{aligned} \chi^2 &= \xi_R^T C_R^{-1} \xi_R + \xi_I^T C_R^{-1} \xi_I \\ &= \begin{pmatrix} \xi_R^T & \xi_I^T \end{pmatrix} \begin{pmatrix} C_R^{-1} & 0 \\ 0 & C_R^{-1} \end{pmatrix} \begin{pmatrix} \xi_R \\ \xi_I \end{pmatrix} \end{aligned} \tag{7}$$

However, before we can do any numerical calculation, some more work still need to be done. Generally, the covariance matrix may possess exceedingly small eigenvalues, and thus, the covariance matrix may not be invertible. In order to avoid this problem, we will rotate the covariance into its diagonal form, and discard these exceedingly small eigenvalues. We know that C_R is the a real symmetric matrix, thus, it is easy to find an orthogonal matrix U that can rotate C_R^{-1} into a diagonal matrix. That is, we can rewrite Equation [7] as

$$\begin{aligned} \chi^2 &= \begin{pmatrix} \xi_R^T & \xi_I^T \end{pmatrix} \begin{pmatrix} U\Lambda^{-1}U^T & 0 \\ 0 & U\Lambda^{-1}U^T \end{pmatrix} \begin{pmatrix} \xi_R \\ \xi_I \end{pmatrix} \\ &= \begin{pmatrix} \xi_R^T U & \xi_I^T U \end{pmatrix} \begin{pmatrix} \Lambda^{-1} & 0 \\ 0 & \Lambda^{-1} \end{pmatrix} \begin{pmatrix} U^T\xi_R \\ U^T\xi_I \end{pmatrix} \\ &= \begin{pmatrix} \tilde{\xi}_R^T & \tilde{\xi}_I^T \end{pmatrix} \begin{pmatrix} \Lambda^{-1} & 0 \\ 0 & \Lambda^{-1} \end{pmatrix} \begin{pmatrix} \tilde{\xi}_R \\ \tilde{\xi}_I \end{pmatrix} \\ &= \tilde{\xi}^T \begin{pmatrix} \Lambda^{-1} & 0 \\ 0 & \Lambda^{-1} \end{pmatrix} \tilde{\xi} \end{aligned} \tag{8}$$

Here,

$$\begin{aligned}
\tilde{\xi}_R &= U^T \xi_R \\
&= U^T G_R - U^T K_R A \\
&= \tilde{G}_R - \tilde{K}_R A, \\
\tilde{\xi}_I &= U^T \xi_I \\
&= U^T G_I - U^T K_I A \\
&= \tilde{G}_I - \tilde{K}_I A, \\
\tilde{\xi} &= \begin{pmatrix} \tilde{\xi}_R \\ \tilde{\xi}_I \end{pmatrix}
\end{aligned} \tag{9}$$

The diagonal matrix Λ may contain exceedingly small eigenvalues, and thus its inverse may diverge. In order to avoid this divergence, we should discard these exceedingly small eigenvalues. To do this, we will cut off the matrix Λ. After the cut off, the dimension of the matrix would become smaller, and we denote this new matrix as Λ_c. Similarly, the matrix U and the vector $\tilde{\xi}$ will also be cut off, and we denote the new matrix and vector as U_c and $\tilde{\xi}_c$.

Now we can write Q as (Here, and from now on, for sake of simplicity, the subscript c has been removed. It should be understood that the matrices and vectors that appear in the formula below are all truncated matrices and vectors.)

$$\begin{aligned}
Q &= \alpha S - \frac{1}{2}\chi^2 \\
&= \alpha S - \frac{1}{2}\tilde{\xi}_R^T \Lambda^{-1} \tilde{\xi}_R - \frac{1}{2}\tilde{\xi}_I^T \Lambda^{-1} \tilde{\xi}_I
\end{aligned} \tag{10}$$

In the maximum entropy method, we will calculate the extremal values for Q, and the extremum is obtained by setting the gradient of Q to zero, that is, $\nabla_A Q = 0$. The gradient of Q with respect to $A(\omega_i)$ is (Here, the originally continuous variable ω has been discretized, and the index i is used to enumerate the discretized ω.)

$$\begin{aligned}
f(A(\omega_i)) &= \frac{\delta Q}{\delta A(\omega_i)} \\
&= -\alpha\Big(1 + \log\frac{A(\omega_i)}{D(\omega_i)}\Big)\Delta\omega \\
&+ \sum_{nm} \tilde{K}_R(\omega_n, \omega_i)\Delta\omega(\Lambda^{-1})_{nm}\Big(\tilde{G}_R - \tilde{K}_R A\Big)_m \\
&+ \sum_{nm} \tilde{K}_I(\omega_n, \omega_i)\Delta\omega(\Lambda^{-1})_{nm}\Big(\tilde{G}_I - \tilde{K}_I A\Big)_m
\end{aligned} \tag{11}$$

We will use Newton's method to calculate the roots of the equation $f(A(\omega_i)) = 0$. In Newton's method, we are to solve the iteration equation

$$A_{n+1}(\omega_i) = A_n(\omega_i) - \sum_j (J^{-1})_{ij} f(A_n(\omega_j)). \tag{12}$$

Here, we have defined the Jacobian matrix as

$$\begin{aligned} J_{ij} &= \frac{\delta f(A(\omega_i))}{\delta A(\omega_j)} \\ &= \frac{\delta^2 Q}{\delta A(\omega_i)\delta A(\omega_j)} \\ &= -\alpha\Delta\omega\frac{\delta_{ij}}{A(\omega_i)} \\ &\quad - \sum_{nm} \tilde{K}_R(\omega_n,\omega_i)\Delta\omega(\Lambda^{-1})_{nm}\tilde{K}_R(\omega_m,\omega_j)\Delta\omega \\ &\quad - \sum_{nm} \tilde{K}_I(\omega_n,\omega_i)\Delta\omega(\Lambda^{-1})_{nm}\tilde{K}_I(\omega_m,\omega_j)\Delta\omega \end{aligned} \tag{13}$$

It is easy to see that the Jaocbian matrix is actually the Hessian matrix for the function Q, with $A(\omega_i)$ as the variables. Moreover, the matrix J is negative definite, which guarantees that we can always find the maximum for the function Q. Thus, we can rewrite the Equation [12] as

$$-J\Big(A_{n+1}(\omega) - A_n(\omega)\Big) = f(A_n(\omega)) \tag{14}$$

Since $-J$ is positive definite, we can use conjugate gradient method to solve the above equation. The calculate of the inverse of Jaocbian matrix is not recommended for the solution of Equation [12] because of the instability of inverting matrices. The rule of thumb for the numerical solution of linear systems is that, you should avoid inverting matrices whenever possible. As a result of this, we adopt the conjugate gradient method for the solution of Equation [12]. The conjugate gradient method is fully described in wikipedia and will not be repeated here. The pseudocode for this method is displayed in Fig. [1.1].

When I run the program, I start from a very large α value. As has been noted above, we can interpret the α as temperature. Maximization of Q, which is equivalent to the minimization of "free energy", is easier to do at high temperature. It can be seen from the definition of Q that when $\alpha \to \infty$, the extremal values of Q will coincide with the extremal values of S. Setting the gradient of entropy to zero, we have

$$\frac{\delta S}{\delta A(\omega)} = -1 - \log\frac{A(\omega)}{D(\omega)} = 0 \tag{15}$$

The algorithm is detailed below for solving **Ax** = **b** where **A** is a real, symmetric, positive-definite matrix. The input vector $\mathbf{x}_0$ can be an approximate initial solution or **0**. It is a different formulation of the exact procedure described above.

$\mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0$
$\mathbf{p}_0 := \mathbf{r}_0$
$k := 0$
repeat
$\alpha_k := \frac{\mathbf{r}_k^\mathsf{T}\mathbf{r}_k}{\mathbf{p}_k^\mathsf{T}\mathbf{A}\mathbf{p}_k}$
$\mathbf{x}_{k+1} := \mathbf{x}_k + \alpha_k\mathbf{p}_k$
$\mathbf{r}_{k+1} := \mathbf{r}_k - \alpha_k\mathbf{A}\mathbf{p}_k$
if r_{k+1} is sufficiently small then exit loop
$\beta_k := \frac{\mathbf{r}_{k+1}^\mathsf{T}\mathbf{r}_{k+1}}{\mathbf{r}_k^\mathsf{T}\mathbf{r}_k}$
$\mathbf{p}_{k+1} := \mathbf{r}_{k+1} + \beta_k\mathbf{p}_k$
$k := k + 1$
end repeat
The result is $\mathbf{x}_{k+1}$

Figure 1.1: The pseudocode for conjugate gradient algorithm. From wikipedia.

Solving the above equation, we have $A(\omega) = e^{-1}D(\omega)$. Therefore, when α is very large, the spectral function we have should be very similar to the default model. I have used the Gaussian as the default model, and run the program by initializing the initial spectral function at small α from the calculated spectral function at large α. Running the program from large α to small α is called the annealing process.

2 DOS results

2.1 Test case

I am going to use the python program that I developed to calculate the DOS from the knowledge of imaginary frequency Green function. In order to test if the program is correct,

I assume that we have a spectral function, which is

$$A(\omega) = \frac{1}{2}N(\omega, -2.1, 0.4) + \frac{1}{2}N(\omega, 1.2, 0.5). \tag{16}$$

Here, $N(\omega, \mu, \sigma)$ means a normal distribution with mean value μ and standard deviation σ. From this spectral function, I can calculate the imaginary frequency Green function. I further added Gaussian noise to the Green function, and from the Green function, I extracted the spectral function, and compare the calculated spectral function with the original spectral function. The comparison is shown in Fig. 2.1.

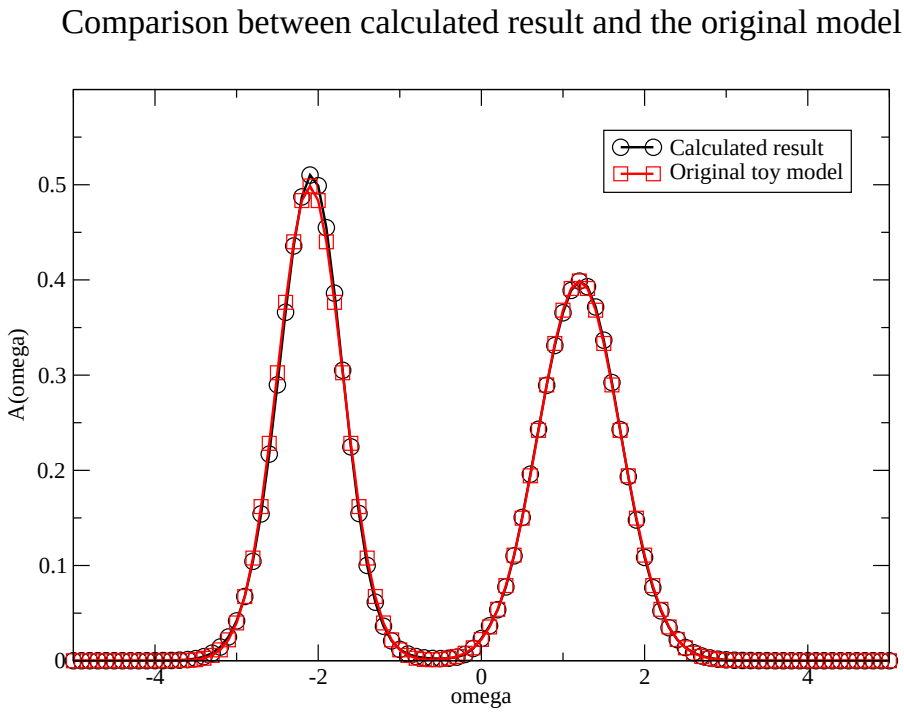

Figure 2.1: Comparison of the calculated spectral function with the original known spectral function.

It can be seen that the calculated result is almost the same as the original spectral function. Therefore, the program can generate the correct spectral function from the knowledge of the imaginary frequency Green function.

References

[1] Mark Jarrell and James E Gubernatis. Bayesian inference and the analytic continuation of imaginary-time quantum monte carlo data. *Physics Reports*, 269(3):133–195, 1996.