1 Solving the MBAR Equations

1.1 Minimization

As discussed previously [], there a number of mathematically equivalent ways of formulating these equations. For example, one can show that minimizing the following objective function is an equivalent formulation of MBAR:

$$h(f) = -\sum_{k}^{m} N_k f_k + \sum_{n}^{N} \log \sum_{k}^{m} N_K \exp(f_k - u_k(x_n))$$

The gradient and hessian of this equation are given by

$$\frac{\partial h}{\partial f_i} = N_i - \sum_{n}^{N} \frac{N_i \exp(f_i - u_i(x_n))}{\sum_k N_k \exp(f_k - u_k(x_n))}$$

$$\frac{\partial^2 h}{\partial f_i \partial f_j} = N_i \delta_{ij} \sum_{n} W_{ni} - N_i N_j \sum_{n} W_{ni} W_{nj}$$

Due to precision issues, this objective function–essentially a partition function–poses a moderate challenge for many packaged optimization routines. To reduce these issues, we have several tips. First, using a overflow-protected implementation of the *logsumexp*() function is critical. This means that any time a *logsumexp* operation is performed along an axis, the maximum values along that axis must be pre-subtracted out of the operation []. Alone, this trick is insufficient because the objective function also suffers from underflow. At the optimal value of f_i, the objective function can have values on the order of 10^5. This means that changes $\Delta h < 10^{-10}$ are often lost to underflow. To mitigate underflow, there are several solutions. First, one can use a stable sum–e.g. ordering the summands from least to greatest during summation. The stable sum can be used in several other MBAR calculations; however, we find that it is most sorely needed in the objective function, as this includes a sumation over both the i and n axes–the double summation is quite sensitive to Second, one can pre-condition the matrix $u_k(x_n)$ by first subracting out any vector: $u_k(x_n)^* = u_k(x_n) - b_n$. Adjusting b_n is mathematically (but not numerically) equivalent to working with $h(f_i) - c$ and often has values with smaller magnitude, allowing great precision. Finally, we point out that many of the widely-used minimization techniques (e.g. BFGS) attempt to numerically approximate the hessian matrix from repeated gradient evaluations. However, the precision challenges in MBAR may cause difficulty in these approaches.

1.2 Nonlinear Equations

In our experience, the most effective solution of MBAR involves the following set of nonlinear equations:

$$g_i(f) = f_i - \log \sum_n^N \frac{\exp[-u_i(x_n)]}{\sum_k^m N_k \exp[f_k - u_k(x_n)]}$$

First, we note that this equation can be used in a simple iterative scheme, as discussed in []. However, we find that using this nonlinear equation and its hessian (shown below) in the MINPACK solver HYBR provides a fast and robust solution to the MBAR equations.

$$\frac{\partial g_i}{\partial f_j} = \frac{1}{\sum_n^N W_{ni}}(-\delta_{ij} \sum_n^N W_{ni} + N_j \sum_n^N W_{ni} W_{nj})$$

2 Exp Space

Let

$$Q_{ij} = q_i(x_j)$$

and

$$R_{ij} = Q_{ij} N_i$$

Then

$$c_i = \sum_j \frac{Q_{ij}}{\sum_k R_{kj} c_k^{-1}}$$

3 Log Space

Let's work out an optimized form of the log-space calculation.

Let

$$\mu_{ij} = \log(N_i) - u_i(x_j)$$

Thus

$$f_i = -\log \sum_j \frac{\exp[-u_i(x_j)]}{\sum_k \exp[f_k + \mu_{kj}]}$$

$$f_i = -\log \sum_j \exp[-u_i(x_j) - \log \sum_k \exp[f_k + \mu_{kj}]]$$

Thus, we need to perform two logsumexp calculations:

$$s_j = -\log \sum_k \exp[f_k + \mu_{kj}]$$

$$f_i = -\log \sum_j \exp[-u_i(x_j) + s_j]$$

4 Minimization

Consider the objective function:

$$F = \sum_k^K N_k f_k - \sum_n^N \log \sum_k N_k \exp(f_k - u_k(x_n))$$

The partial derivatives are given by

$$\frac{\partial F}{\partial f_i} = N_i - \sum_n N_i \exp(f_i) \exp(-u_i(x_n)) \frac{1}{\sum_k N_k \exp(f_k - u_k(x_n))}$$

$$\frac{\partial F}{\partial f_i} = N_i - N_i \exp(f_i) \sum_n \exp(-u_i(x_n)) \frac{1}{\sum_k N_k \exp(f_k - u_k(x_n))}$$

Suppose we solve for the maximum of this equation:

$$\frac{\partial F}{\partial f_i} = 0$$

$$N_i = N_i \exp(f_i) \sum_n \exp(-u_i(x_n)) \frac{1}{\sum_k N_k \exp(f_k - u_k(x_n))}$$

$$\exp(-f_i) = \sum_n \exp(-u_i(x_n)) \frac{1}{\sum_k N_k \exp(f_k - u_k(x_n))}$$

$$f_i = -\log \sum_n \exp(-u_i(x_n)) \frac{1}{\sum_k N_k \exp(f_k - u_k(x_n))}$$

Thus, the maximization problem encodes the MBAR equations. Now, how does one best calculate the gradient?

$$\frac{\partial F}{\partial f_i} = N_i - \sum_n N_i \exp(f_i) \exp(-u_i(x_n)) \frac{1}{\sum_k N_k \exp(f_k - u_k(x_n))}$$

As we saw previously, we can decompose this into two *logsumexp* calculations–first the denominator, then the numerator. Again, this approach should be fairly robust as the logsumexp operation limits overflow error.