Gradient Descent for Logistic-Weighted Kernel-Based Point and Interval Estimation of Two-Parameter Systems

Will Nickols, Srihari Ganesh

August 7, 2022

1 Model

Let (y_1, y_2) be the true value for the two-dimensional continuous condition we are interested in. We want to create a probability density function $f(\hat{y_1}, \hat{y_2})$ that models the probability of the true crystallization condition being equal to some potential $(\hat{y_1}, \hat{y_2})$. We want to maximize the probability assigned to some small interval around the value of the true condition, $\int_{y_1(1-\delta)}^{y_1(1+\delta)} \int_{y_2(1-\delta)}^{y_2(1+\delta)} f(\hat{y_1}, \hat{y_2}) d\hat{y_1} d\hat{y_2}$ for some small δ, or, equivalently, we want to minimize the area of the fit density function that falls outside of that interval, $1 - \int_{y_1(1-\delta)}^{y_1(1+\delta)} \int_{y_2(1-\delta)}^{y_2(1+\delta)} f(\hat{y_1}, \hat{y_2}) d\hat{y_1} d\hat{y_2}$. This probability density can be created by applying a Gaussian kernel to a set of known crystallization conditions $\mathbf{x}$ from similar proteins that contained the condition of interest. With $(x_{1,i}, x_{2,i})$ as the value of the crystallization condition for protein i, p_i as the Mash p-value of the alignment between protein i and the target protein, $(h_{1,i}, h_{2,i})$ as the bandwidths of the kernel element for protein i, n_p as the total number of proteins with Mash p-values less than τ, $(\bar{x_1}, \bar{x_2})$ as the average of the crystallization conditions of all proteins excluding the protein of interest, and η as the standard deviation of the crystallization conditions of all proteins excluding the protein of interest, this density function can be written as follows.

$$\begin{aligned} f(\hat{y_1}, \hat{y_2}) = {} & \frac{1}{n_p + 1} \frac{1}{2\pi} \cdot \frac{1}{\eta_1 \eta_2} \exp\left[-\frac{\left(\frac{\hat{y_1} - \bar{x_1}}{\eta_1}\right)^2 + \left(\frac{\hat{y_2} - \bar{x_2}}{\eta_2}\right)^2}{2} \right] \\ & + \frac{1}{n_p + 1} \cdot \frac{1}{2\pi} \sum_i \frac{1}{h_{1,i} h_{2,i}} \exp\left[-\frac{\left(\frac{\hat{y_1} - x_{1,i}}{h_{1,i}}\right)^2 + \left(\frac{\hat{y_2} - x_{2,i}}{h_{2,i}}\right)^2}{2} \right] \text{ for } i \text{ such that } p_i < \tau \end{aligned} \tag{1}$$

Intuitively, this is a kernel density estimate weighing together the distribution of crystallization conditions for all proteins and the distribution of crystallization conditions for only proteins similar to the protein of interest. However, two issues arise: not all of these similar proteins are equally similar, so they should not be weighted equally, and the optimal bandwidths $(h_{1,i}, h_{2,i})$ of each term is unknown. Both of these issues can be solved simultaneously by allowing $(h_{1,i}, h_{2,i})$ to be a function of s_i, the sequence identity (specifically, 1 minus the Mash distance between protein i and the protein of interest). This function should be continuous and decreasing on $[0, 1]$ because more similar proteins should have a smaller bandwidth for their kernels, and the function should have a codomain of $(0, \infty)$ because all the weights should be positive, but some could be much larger than others. Therefore, the relationship between $(h_{1,i}, h_{2,i})$ and s_i will be given by

$$h_{j,i} = \frac{c_j\sigma(w_{j,1}s_i + w_{j,0})}{\int_0^1 \sigma(w_{j,1}x + w_{j,0})dx} = \frac{c_j\sigma(w_{j,1}s_i + w_{j,0})}{\frac{\ln\left(e^{-w_{j,1}} + e^{w_{j,0}}\right) - \ln(1+e^{w_{j,0}})}{w_{j,1}} + 1} \tag{2}$$

for $j \in \{1, 2\}$ where σ is the sigmoid function and c_j is a scaling value to be fit. Letting $\mathbf{x}$ be the vector of conditions of the similar proteins and $\mathbf{s}$ be the vector of sequence identities of the similar proteins, define the loss as

$$L((y_1, y_2), \mathbf{x}, \mathbf{s}, (\bar{x_1}, \bar{x_2}), (\eta_1, \eta_2), (\delta_1, \delta_2), \beta) = 1 - \int_{y_1(1-\delta)}^{y_1(1+\delta)} \int_{y_2(1-\delta)}^{y_2(1+\delta)} f(\hat{y_1}, \hat{y_2}) d\hat{y_1} d\hat{y_2} + \beta ||\boldsymbol{\eta} - \mathbf{c}||^2$$

with f as defined in equation 1 with its $(h_{1,i}, h_{2,i})$ as defined in equation 2. We choose to regularize $\mathbf{c}$ against $\boldsymbol{\eta}$ because a naive bandwidth should be about the standard deviation of the whole observed condition range, not 0. In practice, letting c_1 and c_2 vary often causes the model to break down because values of x_i close to y generate an extremely steep gradient for c_1 and c_2, pushing them very close to 0. By creating extremely sharp peaks of density at each x_i, this undermines the effort to create a smooth probability density and makes numerical integration essentially impossible. Thus, we will fix c_1 and c_2 at η_1 and η_2 respectively. While fixing these values forces the average bandwidth to be the standard deviation of all values for the condition, the function is capable of becoming much larger near 0 than near 1, and protein similarities near 0 often do not pass the p-value threshold for inclusion. Thus, in practice, bandwidths can become as small as necessary even with fixed c_1 and c_2. Still, for generality, we will treat both as variables.

(In the implementation of this model, all values y and x of the crystallization condition will be divided by their means to account for the considerable differences in scale between conditions while maintaining the general right skewedness and positive values of all conditions. When predicting values or ranges on the original scale, we can simply generate an estimate on this altered scale and multiply by the condition's mean.)

2 Gradient

The specified model enables the fitting of six parameters: $w_{1,0}$, $w_{1,1}$, c_1, $w_{2,0}$, $w_{2,1}$, and c_2. Let $(h_{1,i}, h_{2,i})$ be as described above. For $j \in \{1, 2\}$, let $\sigma_{j,i} = \sigma(w_{j,1}s_i + w_{j,0})$. Let U_j be the σ_j normalization term $\int_0^1 \sigma(w_{j,1}x + w_{j,0})dx = \frac{\ln\left(e^{-w_{j,1}} + e^{w_{j,0}}\right) - \ln(1+e^{w_{j,0}})}{w_{j,1}} + 1$. Let $d_{j,i} = y_j - x_{j,i}$. Let $z_{j,i} = \left(\frac{d_{j,i}}{h_{j,i}}\right)$. Let $m_j = e^{w_{j,0}} + e^{-w_{j,1}}$. Applying the chain rule, the sum rule for derivatives, and the fact that for $\sigma(x)$, $\frac{d\sigma}{dx} = \sigma(x)(1 - \sigma(x))$, we obtain the following:

$$
\begin{aligned}
\frac{\partial f(\hat{y_1},\hat{y_2})}{\partial w_{j,0}} &= \sum_i \frac{\partial f(\hat{y_1},\hat{y_2})}{\partial h_{j,i}} \frac{\partial h_{j,i}}{\partial w_{j,0}} \\
\frac{\partial f(\hat{y_1},\hat{y_2})}{\partial w_{j,1}} &= \sum_i \frac{\partial f(\hat{y_1},\hat{y_2})}{\partial h_{j,i}} \frac{\partial h_{j,i}}{\partial w_{j,1}} \\
\frac{\partial f(\hat{y_1},\hat{y_2})}{\partial c_j} &= \sum_i \frac{\partial f(\hat{y_1},\hat{y_2})}{\partial h_{j,i}} \frac{\partial h_{j,i}}{\partial c_j} \\
\frac{\partial f(\hat{y_1},\hat{y_2})}{\partial h_{1,i}} &= \frac{1}{(n_p+1)2\pi} \frac{\exp(-z_{2,i}^2/2)}{h_{2,i}} \frac{\exp(-z_{1,i}^2/2)(z_{1,i}^2-1)}{h_{1,i}^2} \\
\frac{\partial f(\hat{y_1},\hat{y_2})}{\partial h_{2,i}} &= \frac{1}{(n_p+1)2\pi} \frac{\exp(-z_{1,i}^2/2)}{h_{1,i}} \frac{\exp(-z_{2,i}^2/2)(z_{2,i}^2-1)}{h_{2,i}^2} \\
\frac{\partial h_{j,i}}{\partial w_{j,0}} &= \frac{c_j \sigma_{j,i}(1-\sigma_{j,i})}{U_j} - \frac{c_j \sigma_{j,i}(\frac{e^{w_{j,0}}}{m_j} - \frac{e^{w_{j,0}}}{1+e^{w_{j,0}}})}{w_{j,1} U_j^2} \\
\frac{\partial h_{j,i}}{\partial w_{j,1}} &= \frac{s_i c_j \sigma_{j,i}(1-\sigma_{j,i})}{U_j} - \frac{c_j \sigma_{j,i}[\frac{-w_{j,1}e^{-w_{j,1}}}{m_j} - (\ln(e^{-w_{j,1}} + e^{w_{j,0}}) - \ln(1+e^{w_{j,0}}))]}{w_{j,1}^2 U_j^2} \\
\frac{\partial h_{j,i}}{\partial c_j} &= \frac{h_{j,i}}{c_j}
\end{aligned}
\tag{3}
$$

However, we are actually interested in the integrals of these quantities, so applying the Leibniz integral rule gives the following:

$$
\begin{aligned}
\frac{\partial L((y_1,y_2),\mathbf{x},\mathbf{s},(\bar{x_1},\bar{x_2}),(\eta_1,\eta_2),(\delta_1,\delta_2),\beta)}{\partial w_{j,0}} &= -\int_{y_1(1-\delta)}^{y_1(1+\delta)} \int_{y_2(1-\delta)}^{y_2(1+\delta)} \frac{f(\hat{y_1},\hat{y_2})}{\partial w_{j,0}} d\hat{y_1} d\hat{y_2} \\
\frac{\partial L((y_1,y_2),\mathbf{x},\mathbf{s},(\bar{x_1},\bar{x_2}),(\eta_1,\eta_2),(\delta_1,\delta_2),\beta)}{\partial w_{j,1}} &= -\int_{y_1(1-\delta)}^{y_1(1+\delta)} \int_{y_2(1-\delta)}^{y_2(1+\delta)} \frac{f(\hat{y_1},\hat{y_2})}{\partial w_{j,1}} d\hat{y_1} d\hat{y_2} \\
\frac{\partial L((y_1,y_2),\mathbf{x},\mathbf{s},(\bar{x_1},\bar{x_2}),(\eta_1,\eta_2),(\delta_1,\delta_2),\beta)}{\partial c_j} &= -\int_{y_1(1-\delta)}^{y_1(1+\delta)} \int_{y_2(1-\delta)}^{y_2(1+\delta)} \frac{f(\hat{y_1},\hat{y_2})}{\partial c_j} d\hat{y_1} d\hat{y_2} + 2\beta(c_j - \eta_j)
\end{aligned}
\tag{4}
$$

3 Stochastic Gradient Descent and Model Updating

Because of the memory requirements involved in manipulating all the amino acid identity scores at once, we will use stochastic gradient descent to pick a protein at random, determine its amino acid identity against all the other proteins, compute its density function, and update the weights according to the loss. With a learning rate α, the update statements will be as follows:

$$
\begin{aligned}
w_{j,0} &\leftarrow w_{j,0} - \alpha \frac{\partial L((y_1,y_2),\mathbf{x},\mathbf{s},(\bar{x_1},\bar{x_2}),(\eta_1,\eta_2),(\delta_1,\delta_2),\beta)}{\partial w_{j,0}} \\
w_{j,1} &\leftarrow w_{j,1} - \alpha \frac{\partial L((y_1,y_2),\mathbf{x},\mathbf{s},(\bar{x_1},\bar{x_2}),(\eta_1,\eta_2),(\delta_1,\delta_2),\beta)}{\partial w_{j,1}} \\
c_j &\leftarrow c_j - \alpha \frac{\partial L((y_1,y_2),\mathbf{x},\mathbf{s},(\bar{x_1},\bar{x_2}),(\eta_1,\eta_2),(\delta_1,\delta_2),\beta)}{\partial c_j}
\end{aligned}
\tag{5}
$$

The partial derivative of the density function with respect to each parameter will be computed exactly using equation 3, but the Leibniz integrals in equation 4 will be approximated with a left Riemann sum and a $\Delta\hat{y_j}$ of $y_j/100$.

4 Expected value, mode estimation, and confidence intervals

By linearity of expectation, the expectation of f is simply

$$\frac{1}{n_p+1}\bar{x} + \frac{n_p}{n_p+1}\frac{1}{n_p}\sum_i x_i$$

Because we do not need extreme precision, the approximate mode of the distribution can be found by evaluating the PDF from $\min(\bar{x}, \min(\{x_i | x_i \in \mathbf{x}\}))$ to $\max(\bar{x}, \max(\{x_i | x_i \in \mathbf{x}\}))$ with a step size of the difference divided by 1,000 and recording the crystallization condition at the largest value of the PDF. Here, min indicates the element-wise minimum of the two-element vector of conditions (i.e. the minimum of $x_{j,i}$ over all i for $j = 1$ and $j = 2$ separately). Because the density of an individual kernel input decreases on all sides of its mean, the mode is guaranteed to be between these bounds. Because numeric integration over a large sum of variables is computationally costly and we do not need extreme precision, we can use the marginal density for each of the two elements of the condition to find an estimated 95% confidence interval from the 2.5$^{\text{th}}$ percentile to the 97.5$^{\text{th}}$ percentile of the kernel density by iterating over crystallization condition values from either end of their marginal distributions. For the 2.5$^{\text{th}}$ percentile, we begin iterating upwards from $\hat{y_j} = \max(\min(\text{condition value for all proteins}), \min(\Phi^{-1}(0.025)\eta + \bar{x_j}, \{\min(\Phi^{-1}(0.025)h_{j,i} + x_{j,i}) | x_{j,i} \in \mathbf{x}\}))$ and taking steps of the same size as for the mode until

$$\frac{1}{n_p+1}\Phi\left(\frac{\hat{y_j} - \bar{x_j}}{\eta_j}\right) + \frac{n_p}{n_p+1} \cdot \frac{1}{n_p}\sum_i \Phi\left(\frac{\hat{y_j} - x_{j,i}}{h_{j,i}}\right) \geq 0.025$$

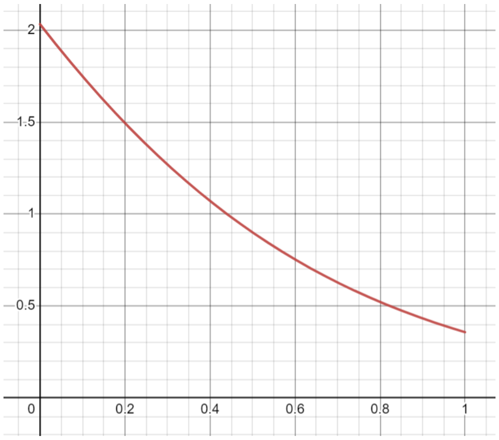
We take that $\hat{y_j}$ as the 2.5$^{\text{th}}$ percentile. Likewise, for the 97.5$^{\text{th}}$ percentile, we begin iterating downwards from $\hat{y_j} = \min(\max(\text{condition value for all proteins}), \max(\Phi^{-1}(0.975)\eta + \bar{x_j}, \max(\{\Phi^{-1}(0.975)h_{j,i} + x_{j,i}) | x_{j,i} \in \mathbf{x}\}))$ and taking steps of the same size until

$$\frac{1}{n_p+1}\Phi\left(\frac{\hat{y_j} - \bar{x_j}}{\eta_j}\right) + \frac{n_p}{n_p+1} \cdot \frac{1}{n_p}\sum_i \Phi\left(\frac{\hat{y_j} - x_{j,i}}{h_{j,i}}\right) < 0.975$$

We then take the $\hat{y_j}$ before the current one (the last one where the expression was greater than 0.975) as the 97.5$^{\text{th}}$ percentile. The proof sketch that the minimum of any term's 2.5$^{\text{th}}$ percentile is less than or equal to the 2.5$^{\text{th}}$ percentile of the marginal kernel density is analogous to the sketch given in the one-parameter system description. Further bounding the search range by the minimum and maximum observed values of the condition ensures that a protein with only a few distantly related proteins doesn't require a massive search space due to very large bandwidths in the kernel density.

5 Initialization

To achieve an initially plausible bandwidth scheme, the following initializations will chosen for $j \in \{1, 2\}$: $w_{j,0} = -1$, $w_{j,1} = -2, c_j = \eta_j$. The following image shows the bandwidths produced by these parameters (with $\eta_j = 1$) with more similar proteins having a lower bandwidth as expected.



The learning rate will start as $\alpha = 0.1$ and will decay by $1/(n' \cdot \text{(number of epochs)})$ after each update. If the weights are within a specified convergence radius for a specified number of iterations (0.1 and 20,000 by default), training for that condition's weights will end. Additionally, because the weight updating proceeds much faster than the Mash distance calculation, the Mash distances will be computed in parallel for a batch of proteins, and the weight updating will proceed sequentially. We fix $\delta = 0.1$. Finally, the Mash p-value threshold will be $\tau = 1/n' \approx 9 \cdot 10^{-6}$.