AST1501 Notes

1. Coordinate Transformation

a. Stream to Equatorial Coordinates

$$\begin{bmatrix} \cos(\phi_1)\cos(\phi_2) \\ \sin(\phi_1)\cos(\phi_2) \\ \sin(\phi_2) \end{bmatrix} = \begin{bmatrix} -0.4776303088 & -0.1738432154 & 0.8611897727 \\ 0.510844589 & -0.8524449229 & 0.111245042 \\ 0.7147776536 & 0.493068392 & 0.4959603976 \end{bmatrix} \times \begin{bmatrix} \cos(\alpha)\cos(\delta) \\ \sin(\alpha)\cos(\delta) \\ \sin(\delta) \end{bmatrix} \tag{1}$$

So if we assume the left hand side matric is A and the others are B and C, respectively, we can get C matrix by:

$$\begin{aligned} A &= BC \\ B^{-1}A &= (B^{-1}B)C \\ B^{-1}A &= C, \end{aligned} \tag{2}$$

therefore, we get:

$$x = y + z \tag{3}$$

b. Cylindrical to Stream coordinates

Coversion from cylindrical coordinates (R, z, ϕ) to cartesian coordinates (x, y, z):

$$\begin{aligned} x &= R\cos(\phi) \\ y &= R\sin(\phi) \\ z &= z \end{aligned} \tag{4}$$

Conversion from cartesian coordinates (x, y, z) to equatorial coordinates (δ, α, d):

$$\begin{aligned} d &= \sqrt{x^2+y^2+z^2} \\ \delta &= \arcsin(\frac{z}{d}) \\ \alpha &= \arctan(\frac{y}{x}) \end{aligned} \tag{5}$$

Finally, conversion from equatorial (δ, α, d) to stream coordinates (ϕ_1, ϕ_2) can be done using equation 1.

2. Likelihood Procedure

$$L(\text{modelorbit}) = P(V_c, q_\phi|\text{Data}) = \prod_i P(\text{data}_i|\text{model}) =$$
$$\prod_i (P(\phi_{2,i}|\phi_{1,i}), P(D_i|\phi_{1,i}), P(Vrad_i|\phi_{1,i}), P(\mu_i|\phi_{1,i})) \text{ (6)}$$

$$\ln(\mathcal{L}) = \sum_i P(\text{data}_i|\text{model}). \tag{7}$$

We can get the likelihood by computing this probablity. However, in Koposov et al. 2010, they marginalized over the oarameters rather than finding the likelihood. Marginalization can be done in the following way:

$$\text{P}(V_c, q_\phi|\text{data(D)}) = \int_7^{10} \text{P}(V_c, q_\phi|\text{D}, \text{R}_0)\text{P}(\text{R}_0)\text{dR}_0, \tag{8}$$

where D represents the data and $\text{P}(\text{R}_0)$ is given by:

$$\text{P}(\text{R}_0) = M(R_0|8.4, 0.42) = \frac{1}{2\pi} e^{\frac{-(R_0-8.4)^2}{2\times 0.42^2}}. \tag{9}$$

The value of our guess for R_0 being 8.4 with the error of 0.42 comes from Koposov et al. 2010. We should then do the same calculation with values of 8.2 for R_0 with the error of 0.1 .

We also need to optimize the initial points for calculating the orbit for each given V_c and q_ϕ. The steps for compouting the likelihood including the optimization is as follows:

1. Take an initial guess for (ϕ_1, ϕ_2) initial point (keep the ϕ_1 the same and change ϕ_2).

2. Convert the (ϕ_1, ϕ_2) to be in cylindrical coordintes
3. Compute potential and orbit for the specified V_c and $_\phi$
4. Find the likelihood value
5. Optimize orbit, which means maximize the the $\ln(\mathcal{L})$ (or minimize the χ^2 since $\chi_2 = -2\ln(\mathcal{L})$)
6. Get the initial position obtained from optimization
7. calculate the orbit with the obtained initial position
8. compute the likelihood from the calculated orbit
9. compute the marginalization
10. Do the above steps for each set of V_c and $_\phi$
11. Make a contour of likelihood

In order to optimize the likelihood, I should pass a function of one or more variables to the scipy.optimize() module. So I need to write a function of phi_2, d, $\mu_{\phi_1}, \mu_{\phi_2}andV_c$ which does find the likelihood value for a set value of ϕ_1. Then I can pass this function to the optimizer function to get the parameter values that maximize the likelihood or minimize the χ^2.

I made my likelihood function faster by eliminating the for loop existed in the previous version. It now takes arrays of data and model values and computes the likelihood for each set of parameter (such as $\phi_2, V_{\rm rad} and \mu$ using numpy.tile() function. The numpy.tile() generates an array that has the array you pass to it repeated n times. So in order to have the calculations correct, I made an array as shown in equation (**??**):

$$\begin{bmatrix} \text{data1} & \text{data2} & \cdots & \text{dataN} \\ \text{data1} & \text{data2} & \cdots & \text{dataN} \\ \text{data1} & \text{data2} & \cdots & \text{dataN} \\ \vdots & \vdots & \ddots & \vdots \\ \text{data1} & \text{data2} & \cdots & \text{dataN} \end{bmatrix}, \tag{10}$$

where the number of rows is equal to the length of model and the number of the columns is equal to the length of data.

Likewise, we can write down a similar matrix for the model values as in equation (**??**):

$$\begin{bmatrix} \mathrm{model\,1} & \mathrm{model\,1} & \cdots & \mathrm{model\,1} \\ \mathrm{model\,2} & \mathrm{model\,2} & \cdots & \mathrm{model\,2} \\ \mathrm{model\,3} & \mathrm{model\,3} & \cdots & \mathrm{model\,3} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{model\,N} & \mathrm{model\,N} & \cdots & \mathrm{model\,N} \end{bmatrix}, \tag{11}$$

where the number of rows is equal to the length of model and the number of columns is equal to the length of data. Then, in order for the matrices to have the same shape so that we will be able to do mathematical operations on them, we have to take the transpose of the model matrix. Finally, to compute the integral of each column in the final value of the tiled matrix, I used simps integration function including axis=0 as an argument so that it takes the integral of the columns and returns the values of the integra of each column as an array to avoid using for loops. Then we will be able to the likelihood calculations as normal.

3. General Notes

- The actions of stars in the cluster are not conserved (because the self-gravity of the cluster is important), but that the actions of stream members freeze once they are stripped.
- The angle difference between stars in a stream and the progenitor increases linearly with time.
-
-
-
-