A Probabilistic Approach to Finding Geometric Objects in Spatial Datasets of the Milky Way

Jon Purnell, Malik Magdon-Ismail, and Heidi Jo Newberg

Rensselaer Polytechnic Institute

Abstract. Data from the Sloan Digital Sky Survey has given evidence of structures within the Milky Way halo from other nearby galaxies. Both the halo and these structures are approximated by densities based on geometric objects. A model of the data is formed by a mixture of geometric densities. By using an EM-style algorithm, we optimize the parameters of our model in order to separate out these structures from the data and thus obtain an accurate dataset of the Milky Way.

1 Introduction

Recent surveys of the Milky Way halo have given astronomers a better idea of the distribution of stars in the galaxy and the location of structures that come from other nearby galaxies. In Newberg & Yanny [?,?], the data from the Sloan Digital Sky Survey (SDSS) has a distribution inconsistent with the power-law distribution that is commonly used and also indicates the presence of a tidal stream from the Sagittarius galaxy.

A great deal of effort is spent fitting various models to this large dataset. The need arises for a tool that can automatically extract the background distribution of the galaxy stars as well as find certain structures of astronomical importance. Also, such a tool can also yield more insight to how well a given distribution fits the Milky Way halo.

We formulate the problem of simultaneously identifying the galaxy stellar distribution and finding structures of astronomical interest as a *mixture density estimation* problem. The observed stellar density (from which the stars are "sampled") is a mixture of the parameterized densities representing the geometric objects and the background stars. We assume that the observed stellar distribution forms an *i.i.d* random sample from the mixture, and the task is to extract each of the mixture components.

Our Contributions. In this paper we focus on a single structure (a tidal stream) in a background, i.e. a mixture of two densities. We use an EM-style algorithm to optimize for the model parameters to determine the position, orientation and size (number of stars) of the stream, in addition to the parameters describing the distribution of the background stars. In the past, in order to obtain the background distribution, one had to be careful to "look" in a direction that would avoid the stream. By *simultaneously* obtaining both distributions, we avoid this complication, and can use all the data. As a result, our estimates should be more accurate.

We give experimental results on synthetic as well as real data, which indicate that this approach performs well at automatically and simultaneously extracting both the background and structure.

2 Mixture Model for the Galaxy

The first step is to define the probability distributions of stars in the galaxy and in the tidal stream (the geometric object of intent). The following distributions are chosen both for their close approximation to the true distribution of stars as well as for their analytic simplicity (integrability and invertibility).

2.1 Galaxy (Background) Stellar Density

The formula for the galactic, or background, distribution, P_b, is a generalized version of the Hernquist equation:

$$P_b(x, y, z) = \frac{1}{r^\alpha (r + r_o)^{3-\alpha+\delta}} \tag{1}$$

where $r = \sqrt{x^2 + y^2 + (z/q)^2}$

When $\alpha = 1$ and $\delta = 1$, our formula becomes the standard Hernquist equation. q controls the scaling of the galaxy model along the z-axis and r_o controls the density of stars near the origin.

2.2 Tidal Stream Density

To represent the stars in the tidal stream, we use a longitudinal elliptical density with a two-dimensional Gaussian cross-sectional density. This has the effect of "smearing" the stars along the ellipse. The ellipse is defined by three vectors that represent its displacement and two axes. Let $\mathbf{x}$ be a generic point on the ellipse. The parametric equation for the ellipse is then

$$\mathbf{x} = \mathbf{c} + \mathbf{a}\cos t + \mathbf{b}\sin t \;\; t \in [0, 2\pi] \tag{2}$$

In order to describe the probability density for the stream, consider $P_s(\mathbf{z})$, for a generic point $\mathbf{z}$. Let $\mathbf{x}^*$ be the closest point on the stream to $\mathbf{z}$, and suppose that $\mathbf{x}^* = \mathbf{c} + \mathbf{a}\cos t^* + \mathbf{b}\sin t^*$. We define a cross sectional basis $\{\mathbf{E}_1, \mathbf{E}_2\}$ for the stream at the point $\mathbf{x}^*$ by the two unit vectors

$$\mathbf{E}_1 = \frac{\mathbf{a} \times \mathbf{b}}{\| \mathbf{a} \times \mathbf{b} \|} \;\; \mathbf{E}_2 = \frac{\mathbf{E}_1 \times \dot{\mathbf{x}}^*}{\| \mathbf{E}_1 \times \dot{\mathbf{x}}^* \|} = \frac{\mathbf{E}_1 \times (-\mathbf{a}\sin t^* + \mathbf{b}\cos t^*)}{\| \mathbf{E}_1 \times (-\mathbf{a}\sin t^* + \mathbf{b}\cos t^*) \|} \tag{3}$$

where $-\mathbf{a}\sin\alpha + \mathbf{b}\cos\alpha$ is the tangent vector at $\mathbf{x}^*$. We can then write

$$\mathbf{z} = \mathbf{c} + \mathbf{a}\cos t^* + \mathbf{b}\sin t^* + x\mathbf{E}_1 + y\mathbf{E}_2, \tag{4}$$

where

$$x = (\mathbf{z} - \mathbf{c} - \mathbf{a}\cos t^* - \mathbf{b}\sin t^*) \cdot \mathbf{E_1} \tag{5}$$

$$y = (\mathbf{z} - \mathbf{c} - \mathbf{a}\cos t^* - \mathbf{b}\sin t^*) \cdot \mathbf{E_2} \tag{6}$$

We then have

$$P_s(\mathbf{z}) = \frac{1}{2\pi} \cdot \frac{1}{2\pi\sqrt{\det \Sigma}} e^{-\frac{1}{2}\mathbf{y}^T \Sigma^{-1} \mathbf{y}},$$

where $\mathbf{y} = \begin{bmatrix} x \\ y \end{bmatrix}$. This density corresponds to choosing t uniformly in $[0, 2\pi]$ along the ellipse, and then adding Gaussian cross-sectional noise with variance covariance matrix Σ. We will simplify this general model to assume that the cross-section is spherically symmetric, in which case we get the simpler density

$$P_s(\mathbf{z}) = \frac{1}{2\pi} \cdot \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2\sigma^2}(x^2+y^2)}.$$

σ corresponds to the "thickness" of the tidal stream.

2.3 Mixture Model Density

The mixture density is obtained by combining the galaxy and tidal stream densities. To combine the distributions, they must be both normalized within the region of interest and weighted by a mixing parameter ϵ. Normalization is achieved by dividing the distributions by their integral over the region of interest. These integrals can be calculated analytical for simple densities and by numerical integration for more complex densities. The final mixture density is given by:

$$P_m(\mathbf{x}) = \epsilon \frac{P_b^{'}(\mathbf{x})}{\int P_b^{'}} + (1-\epsilon) \frac{P_s^{'}(\mathbf{x})}{\int P_s^{'}} \tag{7}$$

where $P_b^{'}(\mathbf{x}) = \rho(\mathbf{x}) P_b(\mathbf{x})$ and $P_s^{'}(\mathbf{x}) = \rho(\mathbf{x}) P_s(\mathbf{x})$. $\rho(\mathbf{x})$ is an efficiency which corresponds to the probability that a star is detected given that it is at position $\mathbf{x}$. Thus the presence of stars at large distances in our dataset indicates that the density there must be higher because fewer of these stars are detected.

2.4 Normalization

In normalizing our probabilities, we calculate the integral of each distribution over the area of our dataset. In our physical application, the dataset is in a wedge shape that pivots at our Sun given by:

$$307 \leq \theta < 436 \quad -1.25 \leq \phi < 1.25 \quad 1.4 \leq r < 57.5 \tag{8}$$

The analytical solution to the normalization integrals is not easy. Also, if we were to change the distributions, we would have to re-calculate the solution. For these reasons, we choose a numerical integration approach.

The integrals are obtained by dividing the wedge into several smaller volume elements and computing the Rieman sum approximation to the integrals. The small volume elements we used are defined by:

$$\Delta\theta = \frac{436 - 307}{490} \approx 0.26 \tag{9}$$

$$\Delta\phi = \frac{1.25 + 1.25}{10} = 0.25 \tag{10}$$

$$\Delta r = \frac{57.5 - 1.4}{180} \approx 0.31 \tag{11}$$

Let A represent a generic small volume element, V_A its volume and $\mathbf{O}_A$ its center. Then the Rieman approximation to the integral is given by:

$$\int P_b' = \sum_A P_b'(\mathbf{O}_A)V_A \qquad \int P_s' = \sum_A P_s'(\mathbf{O}_A)V_A \tag{12}$$

2.5 Data Efficiency

The ratio of the number of stars in our dataset to the number of actual stars in the galaxy is called the efficiency of our dataset and is a function of distance from the sun, r_S, which is measured in parsecs. By comparing overlapping runs, and the number of matched stars, we can estimate the efficiency function. Figure 1 shows a plot of the efficiency versus g^* where:

$$g^* = 5\log_{10}(r_S/10) + 4.2 \tag{13}$$

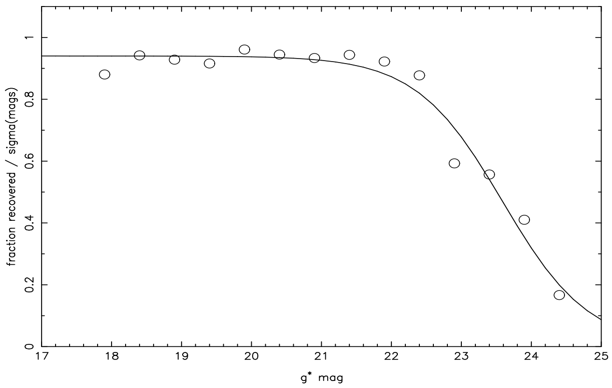

Fig. 1. Completeness of the dataset where the color is in the range of (0.1,0.3). The circles indicate the completeness estimated by comparing two overlapping runs. The solid line is the function fitted to these estimates

The solid line in Figure 1 is the result of fitting a function to these data points. This derived efficiency formula is given by:

$$\rho(g^*) = \frac{0.9402}{e^{1.6171(g^* - 23.5877)} + 1} \tag{14}$$

3 Maximum Likelihood Estimation

We use maximum likelihood density estimation to estimate the parameters in our model. The dataset is composed of the stars $\{\mathbf{x}_i\}_{i=1}^N$. The likelihood is given by $P[\{\mathbf{x}_i\}_{i=1}^N \mid Parameters] = \prod_i P_m(\mathbf{x}_i)$. Taking the logarithm and dividing by N, we then maximize $\mathcal{E}(Parameters) = \frac{1}{N}\sum_{i=1}^N \log P_m(\mathbf{x}_i)$.

4 Parameter Optimization

To maximize $\mathcal{E}$, we use a standard conjugate gradient optimization algorithm together with the line search algorithm [**?**] described in [**?**].

Each step taken in the conjugate gradient algorithm involves calculating a direction vector based on the gradient at the current parameter set, finding the maximum probability that can be obtained along this direction and then testing whether the stopping conditions have been met. We use a simple finite difference scheme to obtain the gradient.

$$\left(\frac{\partial \mathcal{E}}{\partial \mathbf{p}_i}\right)_{\pm} = \pm\left(\frac{\mathcal{E}(\mathbf{p} \pm h_i \mathbf{e}_i) - \mathcal{E}(\mathbf{p})}{h_i}\right) \tag{15}$$

where $\mathbf{e}_i$ is the standard unit vector.

Since $\mathcal{E}$ has varying sensitivities to the different parameters there is no single value for h that we can use. In fact, due to the normalization constants in our probability distributions, we may find ourselves near a non-differentiable cusp on the error surface. To get around this, we start with a sufficiently large value for h, for each parameter $\mathbf{p}_i$. We find $\mathcal{E}$ with that parameter at its current value, $\mathbf{p}_i$, and at the values of $\mathbf{p}_i \pm h$. If the signs of $\left(\frac{\partial \mathcal{E}}{\partial \mathbf{p}_i}\right)_+$ and $\left(\frac{\partial \mathcal{E}}{\partial \mathbf{p}_i}\right)_-$ are different, we know that we are near a local minimum or maximum and that our value for h is too large. In this case we decrease h by half and check the signs of the gradients again. This process continues until either the signs of the gradients agrees or until the value of h falls below a precision tolerance. If the signs agree, we use either gradient as shown in Equation (15). If h falls below the precision tolerances, we must be sufficiently close to a maximum or minimum and so we consider the gradient to be 0 with respect to $\mathbf{p}_i$.

4.1 Parameter Representation

There are two aspects of our parameters that can help us increase the efficiency of our algorithm. First is that some parameters are redundant and can be eliminated. The second aspect is that some parameters can only take on a particular range of values. Converting these parameters into variables without constrained ranges allows us to use more efficient unconstrained optimization techniques.

Reducing the Number of Parameters Our first reduction takes advantage of the orthogonality condition between $\mathbf{a}$ and $\mathbf{b}$. Since $\mathbf{a} \cdot \mathbf{b} = 0$ we can define $\mathbf{b}$ with a magnitude $d_{\mathbf{b}}$ and an angle $\theta_{\mathbf{b}}$ and remove one parameter. The second reduction is from the assumption that the cross-section of the stream is circular. So, we reduce both width values to just one parameter, σ.

Removing Parameter Constraints Since the constraints on our parameters are relatively simple bound constraints, we can optimize with respect to unconstrained parameters by explicitly incorporating the constraints in the objective function as follows. Suppose α is a parameter in $\mathcal{E}$, i.e. $\mathcal{E} = \mathcal{E}\ (\alpha)$ (we only show the α dependence). Suppose that α has a bound constraint $\alpha \in [A, B]$. We can write $\mathcal{E}(\alpha)$ in terms of an unconstrained parameter β by $\mathcal{E}(\alpha) \rightarrow \mathcal{E}(\alpha(\beta))$ where $\alpha(\beta) = A + (B - A)e^{-\beta^2}$. β now becomes an unconstrained parameter in the optimization and α can easily be obtained from β. Such bound constraints apply to $\epsilon \in [0, 1]$ and $q \in [0, 1]$. An unbound constraint of the form $\alpha \in [A, \infty)$ can also be incorporated using the mapping $\alpha(\beta) = A + \beta^2$. Such an unbound constraint applies to $\delta \in [0, \infty)$.

5 Results

We applied our algorithm to two datasets, one synthesized and one from a 2.5 degree thick wedge along the Celestial equator of the SDSS dataset. For each dataset we initialized the algorithm with a parameter set that was close to the optimal value but randomized. The algorithm then ran until the maximum component of the gradient was less that 0.002.

5.1 Synthetic Data

The synthetic data was generated using the exact model mixture density for a particular setting of the parameters. For each star generated, a random number determined whether it was to be a stream star or a background star. If the star is a stream star, three random numbers are generated to determine α, x, and y in (4). If the star is a background star, three random numbers are generated to determine the coordinates of the star, the z-axis component being multiplied by q to take squashness into account.

The synthetic data was generated with the following parameters:

$$\mathbf{c} = (6.9, 10.23, 0.166) \quad \mathbf{a} = (19.4, 9.8, 35.5) \quad \mathbf{b} = (18.5, -2.45, -9.43)$$
$$\sigma = 5.0 \quad q = 0.65 \quad r_o = 13.5 \quad \epsilon = -2.197$$

The algorithm ran for 19 iterations and ended with the following parameters and probability:

$$\mathbf{c} = (7.32, 6.43, -4.72) \quad \mathbf{a} = (22.26, 9.99, 33.08) \quad \mathbf{b} = (15.40, -9.64, -7.45)$$
$$\sigma = 3.32 \quad q = 0.78 \quad r_o = 13.65 \quad \epsilon = -2.78 \quad \mathcal{E} = -3.38358$$

Figure 2 shows the generated synthetic data. Figure 3 shows the separation.

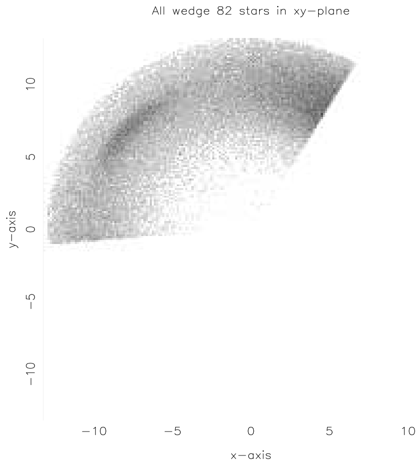

Fig. 2. Density plot of synthetic data in log space

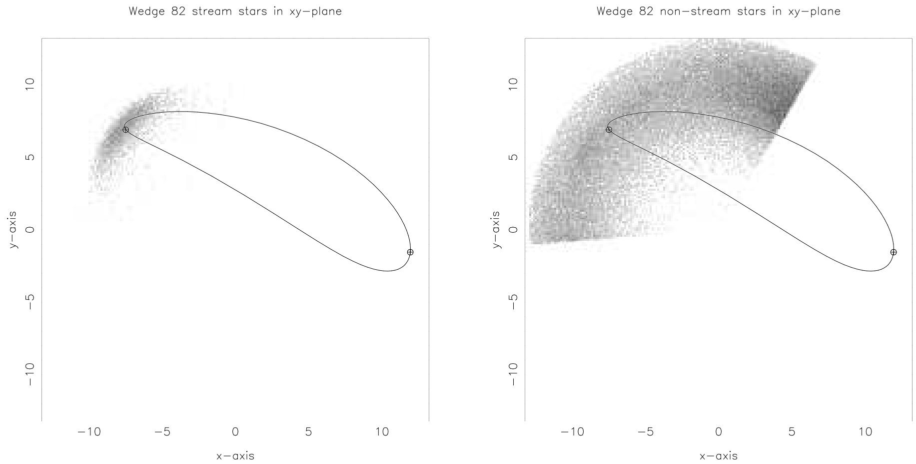

Fig. 3. Separation plot of the synthetic data. The left plot shows the stars labeled as stream stars and the right plot shows the stars labeled as background, or galactic, stars. The line indicates the ellipse of the stream and the circles indicate the point where the stream intersects the plane of the data

5.2 Real Data

With the real data, the algorithm ran for 10 iterations and ended with the following parameters and probability:

$$\mathbf{c} = (6.06, 12.85, -0.039) \quad \mathbf{a} = (19.49, 13.34, 35.70) \quad \mathbf{b} = (19.76, -4.26, -9.20)$$
$$\sigma = 6.21 \quad q = 0.71 \quad r_o = 14.43 \quad \epsilon = -2.38 \quad \mathcal{E} = -3.39434$$

Figure 4 shows a plot of the real data. Figure 5 shows the separation.

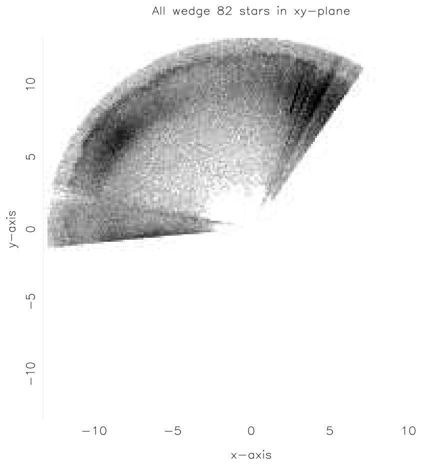

Fig. 4. Density plot of real data from SDSS

6 Conclusion

We have presented a probabilistic approach to finding geometric objects in spatial databases. From these results we can get a clear idea of the direction and size of the tidal stream as well as the distribution of stars in the halo of the Milky Way. Future plans for this work are to increase the number and types of structures that are included in the mixture model, and to scale it up to handle millions of stars. A further direction is to investigate non-parametric estimates of the background densities and resulting geometric structures.

With this approach, astronomers, for the first time, have a tool for extracting the stream from the entire collection of stars, and hence the ability to study separately the stream structure (without being obscured by galaxy structure.

7 Related Works

Previous research has used mixture models and EM algorithms for clustering in large databases [?,?]. Our techniques similarly uses mixture models but instead

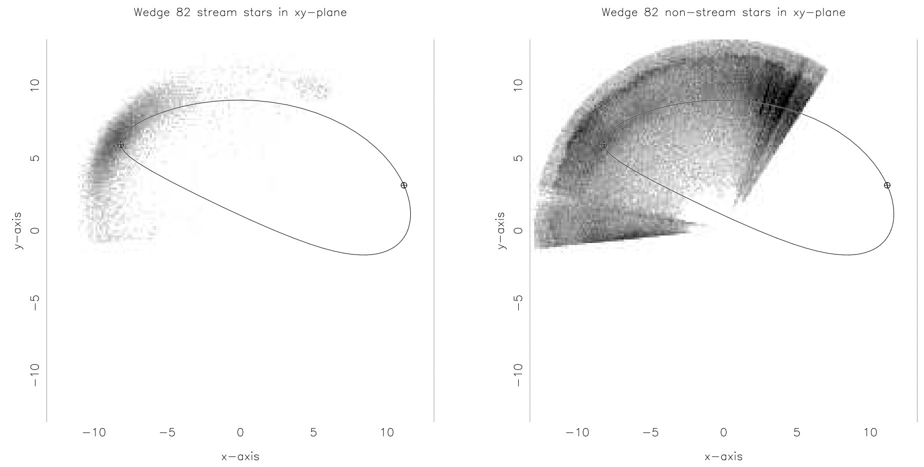

Fig. 5. Separation plot of real data. The left plot shows the stars labeled as stream stars and the right plot shows the stars labeled as background, or galactic, stars. The line indicates the ellipse of the stream and the circles indicate the point where the stream intersects the plane of the real data

use conjugate gradients to solve the maximum likelihood problem. Additionally, our task is not to cluster the data, but to find the parameter set that best fits our model to the data.

8 Acknowledgement

Initial work on this topic was done in May 2003 by Fred Liu. Also, various parts of the mathematics behind the algorithm were implemented by undergraduates at Rensselaer Polytechnic Institute. We would like to acknowledge them here: Adam Gennett, Warren Hayashi, Jim Wisniewski. Final results were obtained with the help of Nate Cole.