SUMMER PROJECT REPORT

The dilemma of Dolidze 28!

by

Susnata Chattopadhyay [1]

under Prof. U.S. Kamath

Indian Institute of Astrophysics, Bengaluru

May - July 2024

[1]Indian Institute of Science Education and Research, Kolkata. Email: sc22ms077@iiserkol.ac.in

Contents

Acknowledgements **0**

Abstract **0**

1 Introduction **1**
1.1 What are clusters? . . . 1
1.2 Classification . . . 1
1.3 Why are open clusters important? . . . 3
1.4 Telescopes . . . 4
1.5 Understanding Color Magnitude diagrams . . . 5

2 Techniques Implemented **8**
2.1 k Nearest Neighbours . . . 8
2.2 DBSCAN . . . 9
2.3 Field Star Decontamination . . . 10
2.4 Probabilistic distribution models . . . 11
2.5 χ^2 Minimization . . . 12
2.6 MCMC Fitting . . . 14

3 Methodology and Results **20**
3.1 Dolidze 28 in Literature . . . 20
3.2 Observational Data . . . 21
3.3 Statistical Validation . . . 23

Conclusions **25**

References **26**

ACKNOWLEDGEMENTS

I, Susnata Chattopadhyay (he/him), would like to express my sincere gratitude for getting the opportunity to participate in the summer project internship (`Ref No.IIA/AS/BGS/VSP/----/---`) at the Indian Institute of Astrophysics (IIA), Bengaluru on the title *'Photometric analysis of open cluster Dolidze 28'*. This project was funded by the VSP program. I am particularly thankful to Prof. Umanath Kamath for his invaluable guidance and mentorship throughout this journey. I would also like to extend my heartfelt gratitude to the Director of this institute for providing me this remarkable experience. A special note of thanks to Mr. Suman Bhattacharya (PhD, Christ's University), Mr. S.R Dhanush (PhD, IIA) and Ms. Harsha K H (BS-MS, IISER- Tirupati) for the endless fruitful discussions on this topic. I also thank my family and friends for their unwavering mental support and motivation, which kept me going during this project.

This research has utilized the data from the European Space Agency's (ESA) GAIA mission, processed by Gaia Data Processing and Analysis Consortium (DPAC) (https://www.cosmos.esa.int/web/gaia/dpac/consortium). Additionally, I have employed the SIMBAD database, operated by CDS in Strasbourg, France. Lastly, I acknowledge the VizieR online catalog, available since 1996, as detailed in a paper published in A&AS, 2000, 143, 23.

ABSTRACT

In our literature search for Cl Dolidze 28, we found a list of 9 references related to this object. We examined each catalogue to determine the instruments used for the survey and the criteria applied. This information is summarized in section 3.1. Following this, we conducted observational and statistical studies in sections 3.2 and 3.3, respectively, focusing on our cluster. These analyses prompt us to investigate whether the region identified as *'Dolidze 28'*, centered at RA = 276.370°, DEC = -14.650° (ICRS J2000), truly meets the criteria for classification as an open cluster.

Chapter 1

Introduction

1.1 What are clusters?

Upon closely observing a *good* night sky, we can see some small faint smudgy patches of objects which stand out against the black background. If you live in Northern Hemisphere, except the Andromeda galaxy, Triangulum galaxy and Orion Nebula (and clouds of course), these patches are called **star clusters**[1]. Our galaxy contains numerous such agglomerations of stars. The Beehive cluster, The Pleiades, The Hyades, Omega Centauri (NGC 5139) are a few famous naked eye star clusters.

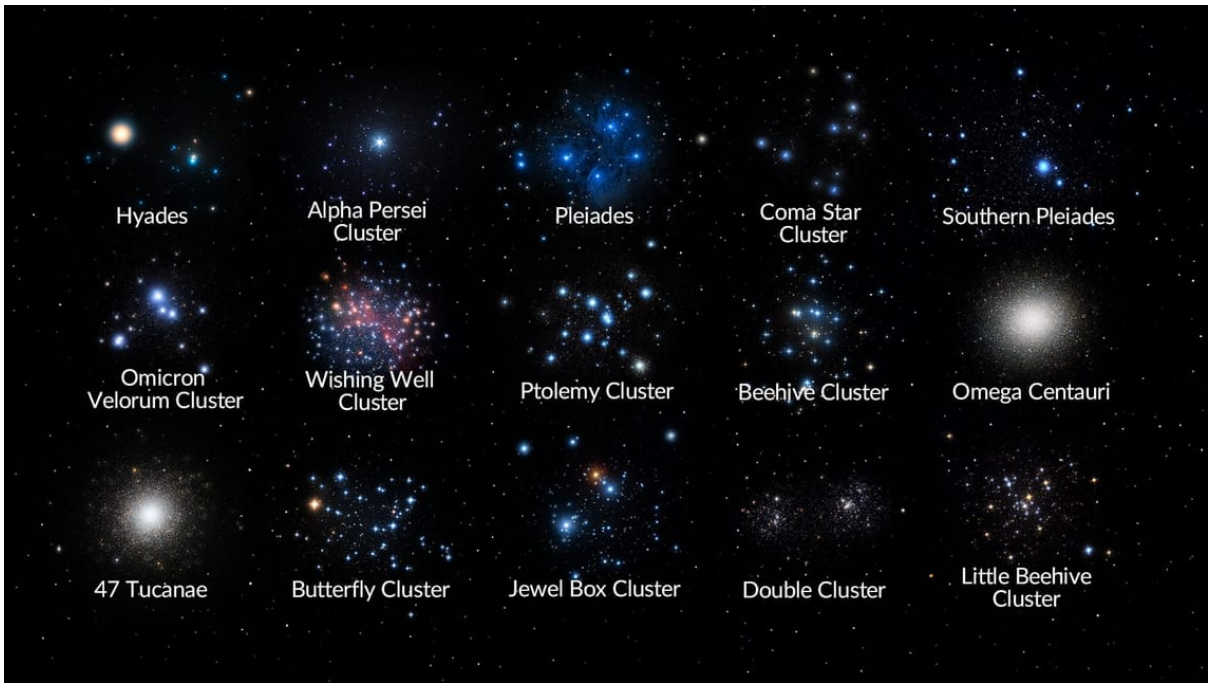

Figure 1.1: A must have bucket list star clusters on a stargazing session. Credit: Starwalk

1.2 Classification

Star clusters are known since millennia, after all they were easy to spot and observe. The first systematic cataloguing of clusters was first done by the great Italian astronomer (again!),

[1]Most of the content in this chapter is referred from the book *The Complex Lives of Star Clusters* by (Stevenson, 2015).

Galileo Galilei in 1610. However, some people, like the English naturalist Reverend John Michell, thought these clusters could be just a fluke alignment of stars in the line of sight. He calculated its probability but soon realized it had a probability of only 1 in 496,000. So it was concluded that these associations had to be physical and real! (Think of what makes a star cluster different from a multiple star system?)

Towards the end of the nineteenth century, as optical instruments became more superior, astronomers were able to observe these associations to fall in two distinct categories- some took a good spherical (globe- like) shape, while others had a rag-tag open appearance- lacking a defined boundary. As you might have already guessed, they are known as **Globular** and **Open** clusters. Our cluster Dolidze 28 is (*was?*) one such proposed open cluster (*or is it?*). Let us look at them more closely!

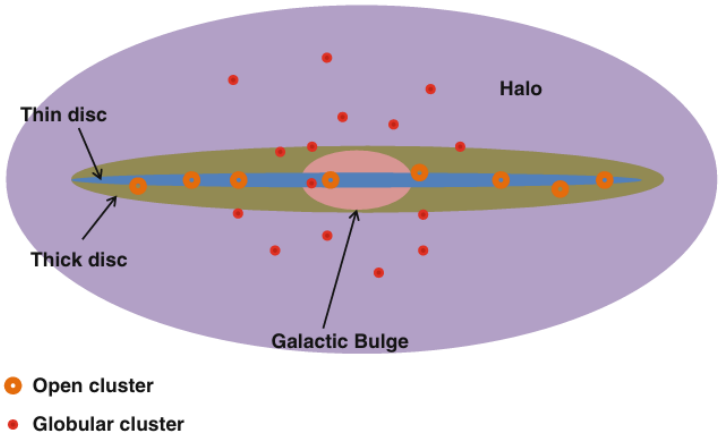

Figure 1.2: A simplified model of the Milky way. The globular clusters are present in the galactic halo, while open clusters are present in dense star forming regions of the galaxy. Credits: (Stevenson, 2015)

1.2.1 Globular Clusters

Globular clusters are the *senior citizens* of our galaxy. All the globular clusters were formed at roughly the same time during the formation of our galaxy in the galactic halo. To the eye, they consist of more luminous red and yellow stars (older). Their spectra showed weak absorption lines of *metals*[2]. This indicated that they had very few heavy elements compared to the Sun. Although the number of stars is a vague parameter to distinguish between open and globular, the later contains huge - lakhs to ten thousands of stars. The **Sawyer-Hogg system** is used to classify globular clusters based on the size and central density of these clusters.

Globular clusters exhibit varying distances and orbital paths. For instance, *M22* follow the general distribution of the halo, which is a vast collection of stars encircling the galaxy's core in sweeping orbits that occasionally traverse the galactic disk. Conversely, clusters like *47 Tucanae* orbit more closely in alignment with the thick galactic disk.

1.2.2 Open Clusters

Open clusters arise from a larger more diffuse affiliations called the *stellar associations*. They are loose groups of stars that began life in a single cloud of gas (called *molecular clouds*) and move through space altogether. Imagine a six-lane highway (as you will often find in Bengaluru), with lots of Autos going in the same direction at same speed. Sooner or later, you will notice some of those moving autos clump together and you can make out a group, while some of them

[2]in Astronomy all the elements heavier than hydrogen and helium are called metals!

gradually tend to drift apart, falling behind the group (cluster). This is exactly what happens but on a large scale with stars. Similarly our Sun is also thought to have begun life in some loose and long ago, abandoned association.

Open clusters (OCs) continue to form today within the galactic disc, recycling scraps of hydrogen and helium, along with traces of supernova remnants from previous stars. This process enriches them with metals. Due to their recent or ongoing star formation activity, they predominantly contain hot, young, blue stars and are less luminous (James Binney, 2021).

As our observational techniques and instruments advance, the distinction between open and globular clusters (GCs) becomes less clear, leading us to observe several clusters with mixed properties in distant galaxies, which lack counterparts in our Milky Way.

Trumpler Classification Scheme for OCs

1. The first number is a Roman numeral from I to IV indicating both the concentration of stars and how readily the cluster is distinguishable from the neighboring field of stars in the galaxy.

2. The second number is an Arabic 1–3, which describes the range in brightness of the stars within the cluster. Three indicates the largest range.

3. The third, is a letter *p, m* or *r*, which describes the overall metal richness of the cluster (poor, medium or rich).

4. Last, but not the least, some carry a final fourth letter *n* also if they have any associated nebulosity.

Example: Pleiades is `I3rnn`.

Although this classification system is superficially easy to grasp, putting it into practice is much more difficult.

1.3 Why are open clusters important?

Studying star clusters is a big advantage because we have all the stars roughly of the same age and composition, so it is a good place to understand how stars of different masses form and evolve. It also formed the test-beds of our theories and challenged the processes by which we knew stars evolved.

Open clusters are the jewels in the crowns of most star forming galaxy. They are sites of high speed action, because of their young, dynamic and unstable nature. By examining the dynamics and kinematics of our neardoor OCs, we can cleverly extrapolate it to far away galactic clusters. The forces that bring the downfall of these clusters, will tell the same tale as with Galaxies and every other stellar bound systems in the universe (Stevenson, 2015).

1.4 Telescopes

In this study, we primarily use data from 2 telescopic surveys

1.4.1 ESA Gaia

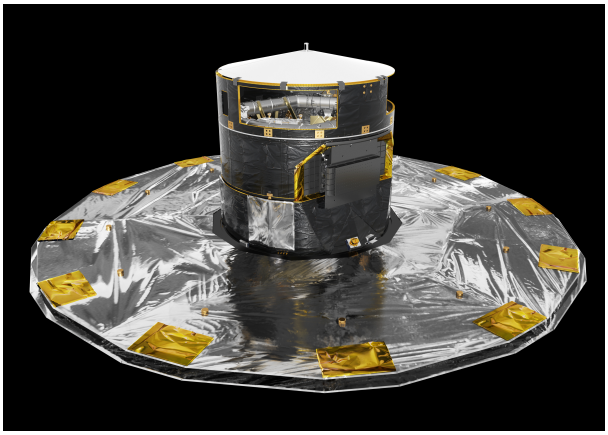

Figure 1.3: A representational image of Gaia satellite telescope. Credits: ESA

Gaia[3] is a satellite which was launched by the European Space Agency (ESA) for a mission to build largest, most precise three-dimensional map of our Galaxy by surveying nearly two billion objects. Gaia contains two optical telescopes that work with three science instruments that precisely measures the astrometry of stars, their velocities, and split their light into a spectrum for analysis. The GAIA DR3 (Data Release-3) provides us 5 parameter astrometric solutions for a star consisting of positions on the sky (α, δ) [`ra, dec`], `parallaxes`, Proper Motions $(\mu_\alpha \cos\delta, \mu_\delta)$ [`pmra, pmdec`] with a limiting magnitude $G \sim 21$ mag and a bright limit of $G \sim 3$ mag. The astrometric solution is accompanied with some new quality indicators, like RUWE, and source image descriptors. The Gaia G band ranges from (330-1050 nm)[Visible - Near IR], G_{BP} (330-680 nm), G_{RP} from (630-1050 nm). The corresponding photometric uncertainities are $\sim 6, 108, 52$ mmag respectively at G=20 mag. The median proper motion uncertainties are about $0.02 - 0.03$ mas/yr ($G = 9 - 15$), 0.07 mas/yr ($G = 17$), 0.5 mas/yr ($G = 20$), and 1.4 mas/yr at $G = 21$ mag [Gaia collaboration (Vallenari et al., 2023)].

1.4.2 2MASS

The 2MASS[4] or the *Two micron-all sky survey* conducted between 1997 and 2001 was an astronomical survey of the sky in infrared light. The 2MASS survey, was conducted by two automated 1.3-meter *cassegrain equatorial* telescopes located at Mt. Hopkins, Arizona (AZ), USA and CTIO, Chile, which employs a three-channel camera (256×256 array) equipped with HgCdTe detectors. This survey catalog provides photometric measurements in the J (1.235 μm), H (1.662 μm), and K_s (2.159 μm) bands for millions of galaxies and nearly half a billion stars. The catalog's sensitivity is 15.8 mag for J-band, 15.1 mag for H-band, and 14.3 mag for K_s-band at S/N of 10. (Skrutskie et al., 2006).

[3] https://www.esa.int/Science_Exploration/Space_Science/Gaia_overview
[4] https://irsa.ipac.caltech.edu/Missions/2mass.html

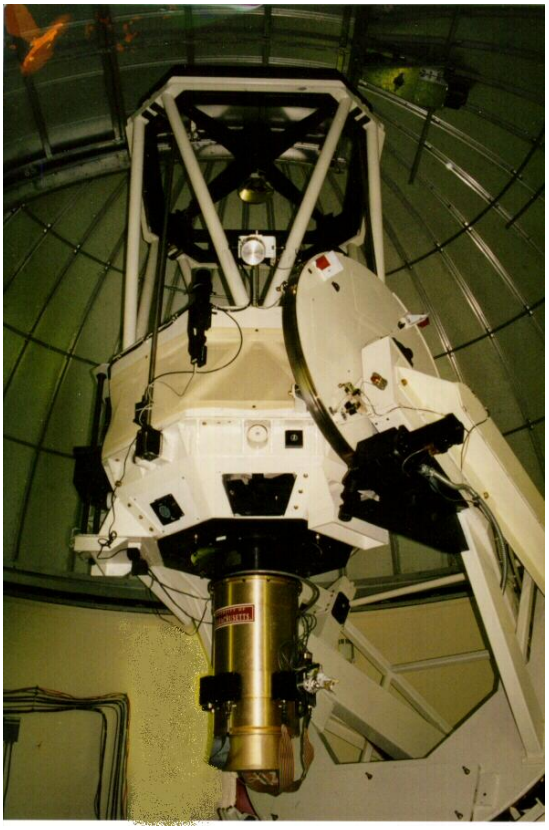

Figure 1.4: One of the 2MASS telescopes at Mt. Hopkins, USA. Credits: harvard.edu

1.5 Understanding Color Magnitude diagrams

1.5.1 What is it?

The theoretical **color magnitude** diagram is a plot of absolute magnitude ($\log \frac{L}{L_\odot}$) against T_{eff}; since color[5] is related to effective temperature (T_{eff}), each point on this diagram corresponds to a unique luminosity, effective temperature, color and stellar radius. The location of a star in CM diagram depends on:

- the rate with which it is generating energy in its core, through nuclear fusion
- structure of star
- age
- chemical composition

While in HR diagram, spectral class replaces color in the diagram.

1.5.2 Historical Background

In the early 20th century, Ejnar Hertzsprung and Henry Norris Russell discovered that, in a plot of color v/s luminosity, stars are not randomly scattered but are concentrated within tighty defined bands. These plots came to known as HR diagrams, after their inventor.

Since then, the HR diagram has been of profound importance to the development of our understanding of stellar evolution.

[5]The color of a star is measured by the ratio of the luminosity in two wavelength bands. For eg: $\frac{L_R}{L_V} = V - R = -0.01$ for *Sirius*.

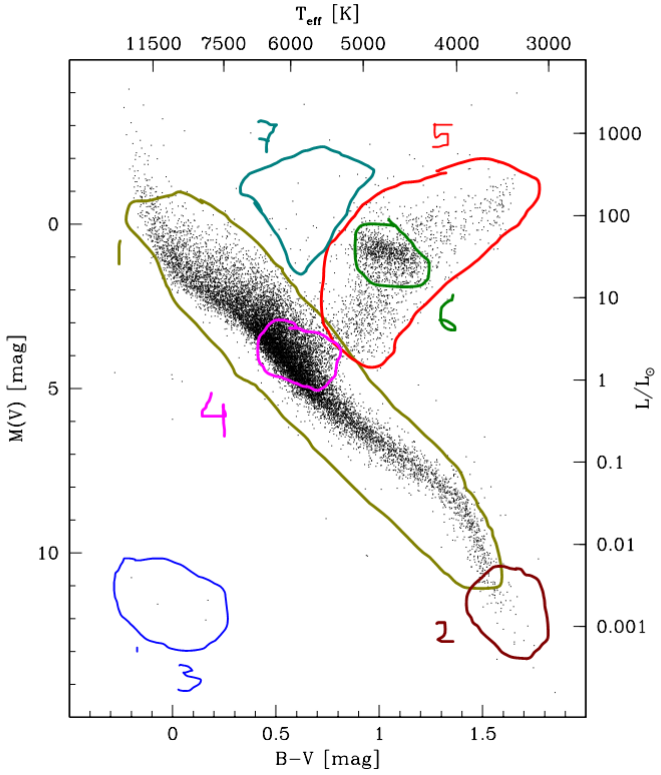

Figure 1.5: CM diagram for 10^4 nearby stars. Credits: (Binney & Tremaine, 2011)

1.5.3 Axes Description

In Fig 1.5, The absolute magnitudes are in the V band and the colors are based on B-V. The right axis shows the V band luminosity in solar units, and the top axis shows approximate values of the effective temperature.

1.5.4 Diagram Description

Close binary stars have been excluded, since the companion contaminates the color and magnitude measurements. Most of the stars fall on the main sequence, which runs from upper left to lower right. The red-giant branch runs upward and to the right from the main sequence. The red-clump stars form a prominent concentration in the middle of the red-giant branch. The giants are chosen from a larger volume than the other stars to enhance their numbers and thereby show the structure of the giant branch more clearly.

Between $M \simeq 0$ and $M \simeq 10$, a factor of 10^4 in luminosity, the radius varies only by a factor of 6, from $3R_\odot$ to $0.5R_\odot$

1.5.5 Explanation

Please refer to the annotated parts in Fig 1.5. This is a brief summary to interpret the CMD.

1. **Main Sequence (MS)**
   - Extends from $(B-V, M_V) \equiv (0,0) \Rightarrow (1.5, 11)$
   - Stars are burning hydrogen in their cores
   - Gravitational contraction is resisted by both thermal and radiation pressure. (**Hydrostatic Equilibrium**)

- Position of star remains fixed, throughout the stage.
- Also called **dwarf stars.**
- *Upper left* stars are $R\uparrow, L\uparrow, T\uparrow$ and blue $\Rightarrow$ **O** type.
- *Lower right* stars have $R\downarrow, L\downarrow, T\downarrow$ and red $\Rightarrow$ **M** type.

2. **Brown Dwarfs**
   - These objects have masses $m < 0.08 m_{\odot}$
   - Never ignite hydrogen in their cores, although they do ignite some lighter elements like Li and D.
   - visible mainly from the radiation they emit, as they contract and cool.
   - their luminosity depends on both mass and age.
3. **White Dwarfs**
   - dim, blue stars
   - small, $R \simeq 10^{-2}\ R_{\odot}$
   - Centered on $(B-V, M_V) \simeq (0, 14)$
   - exhausted their nuclear fuel and are gradually cooling to invisibility
   - White dwarfs are so dense that the electron gas in the interior of the star is degenerate.
   - Gravitational contraction is resisted by the Fermi energy of the star's cold, degenerate electron gas.
4. **Subgiant Branch (SGB)** It is the populated region around $(B-V, M_V) \simeq (0.7 \rightarrow 1,\ 4)$ which joins MS with RGB.
5. **Red Giant Branch (RGB)**
   - Stars that have exhausted hydrogen in their cores and are now burning hydrogen in a shell surrounding an inert helium core.
   - the red-giant branch is an evolutionary sequence: stars climb the giant branch from the main sequence to its tip (where their radii $\gtrsim 100 R_{\odot}$) stars like the Sun.
6. **Red Clump**
   - Centered on $(B-V, M_V) \simeq (1.1, 1)$
   - This feature arises from a later evolutionary stage, which happens to coincide with the red-giant branch for stars of solar metallicity.
   - Red clump stars have already ascended the red-giant branch to its tip and returned, settling at this prominent concentration, when they *start to burn helium* in their cores.
7. **Hertzsprung Gap** It is the region ($M_V \backsim 1$) devoid of stars, between MS and RGB.

Chapter 2

Techniques Implemented

2.1 k Nearest Neighbours

In our project we implement the `NearestNeighbour` function by scikit-learn [https://scikit-learn.org/stable/modules/neighbors.html]. As we shall see later, DBSCAN requires us to estimate 2 input parameters $\equiv (\min_{pts}, \epsilon)$. For a given data-set the optimum value of which is determined with the help of kNN algorithm.

In simple terms, kNN determines the class of a new data point based on the majority class among its k nearest neighbors. The overall steps are:

1. Choose the number k of the neighbours (preferably k should be odd).
2. Calculate the distance to it's neighbours with the new data point using a suitable metric.
3. Take the k nearest neighbours as per the calculated distance.
4. Among these k neighbors, count the number of the data points in each class/category.
5. Assign the new data point to that class/category for which the number of the neighbor is maximum.

The optimum value of ϵ is estimated by a **k-distance graph**. Where we plot the average distance of each point to it's k ($=\min_{pts}$) nearest neighbours, $[\epsilon_i = \frac{d_1+\cdots+d_k}{k}]$. Suppose $k = 40$, then to calculate the distance of each of the N points in the dataset:

```
import numpy as np
from sklearn.neighbors import NearestNeighbors
# Assuming X is your data
k = 40  # Adjust the value of k (min_samples) based on your dataset
nbrs = NearestNeighbors(n_neighbors=k).fit(X)
distances, indices = nbrs.kneighbors(X)

# Sort and plot the average distances to k nearest neighbours
dist_avg = np.sort(np.sum(distances,axis= 1)/k)
plt.plot(dist_avg)
minpts= k
eps = dist_avg[minpts-1]  #as array indexing starts from zero
```

The line `distances, indices = nbrs.kneighbors(X)` returns the array of distances (ϵ_i for i till N) and indices of the k nearest neighbors for each point in the dataset X. The distances array contains the distances of every point to each of its' neighbors till k (so the last column will contain the distance to it's kth neighbour, i.e `distances` is an $N \times k$ array), and the indices variable contains the indices of those neighbors. So, think about it, the farthest isolated point

from the dense regions will have the maximum value of ϵ, while the data point in the core region of a dense cluster will have least ϵ.

Then sort the distance arrays in ascending order, `dist_avg = np.sort(np.sum(distances, axis= 1)/k)` and plot it to obtain the k-Distance graph. The ideal value for ϵ is generally taken to be optimum at the distance value at the "knee point", or the point of maximum curvature. This is because this point represents the point of inflection between two states of data. On either side of this point the clustering behaves differently.

2.2 DBSCAN

Unlike `k-Means clustering` which partitions the data set into 1 or more clusters, DBSCAN (or Density-Based Spatial Clustering Of Applications With Noise) is based on the intuitive notion of clusters and *field noise*[1]. It requires two input parameters $(\epsilon, \min_{pts})$. The estimation of $\min_{pts}$ is dependent on the size of the data (better estimations will come with hit-and-trial), while determination of optimum value of ϵ is already discussed in the preceding section.

The algorithm by (Deng, 2020) of this method is discussed below:

1. Enter parameters $(\epsilon, \min_{pts})$.
2. The model reads each data point in X and Y and stores it as (x, y) - 2-D array.
3. Determines the **density core points** among this coordinates. It accomplishes this in the following way: Start with each point $P \equiv (x_p, y_p)$, in the scatter plot. Take an ϵ neighbourhood ball (circle if it's 2-D) around it. If the ball overlaps with $\geq \min_{pts}$ points, then it is marked as a core point. Proceed in this way to rest of dataset.
4. At the end we will get all the points marked into either *core* or *non-core* points.
5. Then, pick one core-point at random and *assign it cluster 1.* Similarly draw a circle around it. All the *core points* which overlaps within the neighbourhood are assigned cluster 1.
6. Continue in this way until no nearby core points are left. You have now got a cluster, with few non-core points surrounding it. Add the closest *non-core* points to the original core points to cluster 1. You now have your 1st cluster
7. For Core points which are farther than ϵ [Euclidean metric] than cluster 1, assign them cluster 2, repeat the same steps similarly. Notice that clusters are created sequentially.
8. The non-core points which are not labelled in any cluster are marked *field/ outliers.*
9. Hurray!

2.2.1 HDBSCAN

HDBSCAN refers to **Hierarchical** Density-Based Spatial Clustering of Applications with Noise (Campello et al., 2013). Unlike traditional DBSCAN which has a fixed ϵ value, HDBSCAN performs DBSCAN over varying ϵ values and integrates the result to find a clustering that gives the best stability over epsilon. This allows HDBSCAN to find clusters of varying densities, and be more robust to parameter selection.

[1] https://www.youtube.com/watch?v=RDZUdRSDOok

2.3 Field Star Decontamination

We follow the method as suggested by (Dhanush et al., 2024). Note that this method is not used to find any cluster member, but to rather remove the contaminants in the CMD (Color-Magnitude diagram) on the basis of their similar photometric property with that of nearby field regions. The number of cluster stars which we get from this method are the maximum that can be obtained statistically. (Because in highly crowded regions, a mere high density region can be mis-identified as a cluster, they might not be gravitation-ally bound). The algorithm is proceeded as follows:

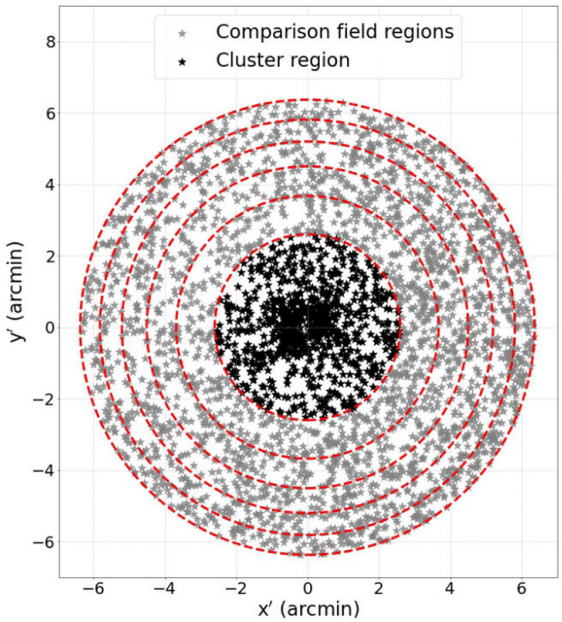

Figure 2.1: A schematic diagram of our model. Credits: (Dhanush et al., 2024)

1. Let the angular size of the cluster (obtained from literature) be d'. We query data with a radius of $r = d'/2$ around the cluster center.

2. Then select another radius R_1 such that the area of the annular region will be equal to that of inner circle. i.e
$$\pi r^2 = \pi(R_1^2 - r^2) \implies R_1 = \sqrt{2}r$$

3. Proceed in this way to get n such annular regions of equal area.
NOTE: We assume field star density to be uniform, so we take annulii of equal areas to ensure each `fieldn` to contain roughly equal number of stars.
$$\pi(R_n^2 - R_{n-1}^2) = \pi r^2 \ \forall n > 2$$
$$\implies R_n = r\sqrt{n+1}$$

4. Create different sets for the annular data like `field_1, field_2, ...` and inner circle as `clusters_can` (candidates).
NOTE: We have used the **Haversine formula (Robusto, 1957)** to calculate the angular distance between 2 points (RA, DEC) $\equiv (\alpha_1, \delta_1)$ & (α_2, δ_2) (in radians!)

$$\boxed{d = 2\arcsin\sqrt{\sin^2\frac{|\delta_1 - \delta_2|}{2} + \cos\delta_1\cos\delta_2\sin^2\frac{|\alpha_1 - \alpha_2|}{2}}}$$

5. Plot the $G - G_{BP-RP}$ of `field_n` and `cluster_can` side by side. Observe.

6. Take one data point at a time from `cluster_can`, and consider [magnitude, colour] bins with different sizes, starting with $[\Delta G, \Delta(G_{BP} - G_{RP})] = [0.02,\ 0.01]$ and iteratively increase (if no matches are found) up to a maximum of [0.5, 0.25] to search for the closest field star counterpart in the `field_1` CMD. Think how you can make this process faster!

7. Those with counterparts in the field CMD(s) are considered as field stars in the cluster region and are removed from the CMD. *The star and the counterpart are both removed* from `field_n` and `cluster_can` CMD!

8. If no matches are found proceed to the next annulus field region, and apply the same steps.

Stars that remained in the `cluster_can` Color-Magnitude Diagram after this statistical cleaning are most likely to be the cluster members and the CMD with such stars may be considered as a cleaned CMD(s).

2.4 Probabilistic distribution models

After we are done with clustering to select the preliminary cluster candidates, we proceed to implement the probability models as suggested by (Balaguer-Núnez et al., 1998). We retained only those stars as **most probable cluster candidates** which have a higher probability. We define $\Phi_c^{\mathbf{v}}$ as the velocity distribution of the cluster stars, $\Phi_f^{\mathbf{v}}$, velocity distribution for field stars, $\Phi_c^{\mathbf{r}}$, the surface-number-density of cluster members and $\Phi_f^{\mathbf{r}}$ to be a uniform distribution for the field stars.

$$\Phi_c^{\mathbf{v}} = \frac{1}{2\pi\sqrt{(\sigma_{\rm c}^2+\epsilon_{xi}^2)(\sigma_{\rm c}^2+\epsilon_{yi}^2)}}\exp\left\{-\frac{1}{2}\left[\frac{(\mu_{xi}-\mu_{x\rm c})^2}{\sigma_{\rm c}^2+\epsilon_{xi}^2}+\frac{(\mu_{yi}-\mu_{y\rm c})^2}{\sigma_{\rm c}^2+\epsilon_{yi}^2}\right]\right\} \tag{2.1}$$

$$\Phi_f^{\mathbf{v}} = \frac{1}{2\pi\left(1-\gamma^2\right)^{1/2}\left(\sigma_{x\rm f}^2+\epsilon_{xi}^2\right)^{1/2}\left(\sigma_{y\rm f}^2+\epsilon_{yi}^2\right)^{1/2}}\times \tag{2.2}$$

$$\exp\left\{-\frac{1}{2\left(1-\gamma^2\right)}\left[\frac{\left(\mu_{xi}-\mu_{x\rm f}\right)^2}{\sigma_{x\rm f}^2+\epsilon_{xi}^2}-\frac{2\gamma\left(\mu_{xi}-\mu_{x\rm f}\right)\left(\mu_{yi}-\mu_{y\rm f}\right)}{\left(\sigma_{x\rm f}^2+\epsilon_{xi}^2\right)^{1/2}\left(\sigma_{y\rm f}^2+\epsilon_{yi}^2\right)^{1/2}}+\frac{\left(\mu_{yi}-\mu_{y\rm f}\right)^2}{\sigma_{y\rm f}^2+\epsilon_{yi}^2}\right]\right\}$$

$$\Phi_c^{\mathbf{r}} = \frac{1}{2\pi r_{\rm c}^2}\cdot\exp\left\{-\frac{1}{2}\left[\left(\frac{x_i-x_{\rm c}}{r_{\rm c}}\right)^2+\left(\frac{y_i-y_{\rm c}}{r_{\rm c}}\right)^2\right]\right\} \tag{2.3}$$

$$\Phi_f^{\mathbf{r}} = \frac{1}{\pi r_{\rm max}^2} \tag{2.4}$$

$$\gamma = \frac{(\mu_{xi}-\mu_{xf})(\mu_{yi}-\mu_{yf})}{(\sigma_{xf}\sigma_{yf})} \tag{2.5}$$

NOTE:

- The axes x and y refer to RA and DEC respectively.
- (μ_{xi}, μ_{yi}) are the proper motions of the i-th star,
- (μ_{xf}, μ_{yf}) is the field proper motion center.
- $(\epsilon_{xi}, \epsilon_{yi})$ are the observed errors of the proper-motion components of the i-th star,

- $(\sigma_{xf}, \sigma_{yf})$ the field intrinsic proper motion dispersions and γ the correlation coefficient.
- $r_{\max}$ is taken as the maximum queried radius.
- r_c is taken as the half of angular size of the cluster.

2.4.1 σ_c calculation

σ_c is called **intrinsic proper motion dispersion** of the cluster stars. Open clusters usually have internal velocity spreads around 1 km/s or lower. In contrast, field disk-population stars typically show velocity spreads that are 10-20 times larger. This allows for the distinction between cluster members and non-members by observing the relative proper motions of stars in the field of an open cluster. However, only for relatively nearby clusters it is possible to determine the intrinsic cluster dispersion under the scatter imposed by the proper-motion measurement errors (Girard et al., 1989).

It is defined as (McNamara & Sanders, 1978):

$$\sigma_c^2 = \sigma_0^2 - \frac{1}{n}\sum_i \zeta_i^2 \tag{2.6}$$

where, n is the total number of *preliminary* cluster stars.

$$\zeta_i = \frac{\epsilon_{xi} + \epsilon_{yi}}{2}$$

$$\sigma_0^2 = \sqrt{\frac{\sum_i (\mu_i - \mu_c)^2}{n-1}}$$

where, μ_i is the proper motion of each star and μ_c is the mean cluster proper motion.
The distribution of all the stars can be described as follows:

$$\boxed{\Phi(i) = n_c \Phi_c^{\mathbf{r}} \Phi_c^{\mathbf{v}}(i) + n_f \Phi_f^{\mathbf{r}} \Phi_f^{\mathbf{v}}(i) = \Phi_c + \Phi_f}$$

where n_c, n_f are the normalized number of (*preliminary*) cluster and field stars.

$$n_c + n_f = 1$$

Thus the total probability is then computed as,

$$P_\mu(i) = \frac{\Phi_c(i)}{\Phi(i)}$$

2.5 χ^2 Minimization

Before implementing MCMC or any other fitting procedures, it is important for us to understand how to *quantitatively* measure the goodness of a fit. One such method is χ^2 test, which gives us a measure of how far our *Model* (M) / fit lies from the *data*(D) points. We are looking to minimize χ^2 with respect to our model parameters $(\theta = [\theta_1, \theta_2, \dots])$.
Let's say we have a data of N measurements (or variables, we are going to assume for now all are independent).

$$D_i = [x_i, y_i, \sigma_i]$$

where, σ_i is the uncertainty in the i^{th} measurement (of y_i). Generally we assume the data (y_i's) to be normally (= gaussian) distributed about the model, with the deviation σ_i. Then, (PDF of a Gaussian):

$$\boxed{P(D_i|M) = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\Delta Y_i^2/2\sigma_i^2}}$$

where, ΔY_i is $y_i - f(x_i, \theta)$ (termed as **residuals**). f is our model dependent on the set of parameters θ.

Total Probability The total probability of getting a set of data, D, given a model, M, is the product of the individual probabilities of each data point:

$$P(D \mid M) = \prod_i \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\Delta Y_i^2/2\sigma_i^2} \equiv \mathcal{L}$$

In technical terms, this probability function is called the **Likelihood** and is denoted by $\mathcal{L}$. However, it is often convenient to work with the log of exponential quantities, so we have

$$\ln \mathcal{L} = -\frac{1}{2}\left[\sum_i \ln 2\pi\sigma_i^2 + \sum_i \left(\frac{\Delta Y_i}{\sigma_i}\right)^2\right]$$

Notice that the first term does not depend on the model, so we can safely drop it

$$\ln \mathcal{L} \sim -\frac{1}{2}\sum_i \left(\frac{\Delta Y_i}{\sigma_i}\right)^2$$

$$\boxed{\chi^2 \equiv \sum_i \left(\frac{\Delta Y_i}{\sigma_i}\right)^2 = \sum_i \left(\frac{y_i - f(x_i, \theta)}{\sigma_i}\right)^2}$$

Thus, the (relative) probability of randomly drawing a specific sample of data given a model is related to the χ^2 of the data given the model:

$$\mathcal{L} \sim e^{-\chi^2/2}$$

which leads us to conclude

$$\chi^2 \text{ minimization} \iff \text{Maximum Likelihood } (\mathcal{L}) \text{ Estimation}$$

2.5.1 Bootstrapping

Suppose we have a collection of N (~ 1000) data points, $\mathbf{A} = [x_1, x_2, \ldots x_N]$, and we perform some statistics say ψ (some function like mean, median etc) to get some result $\psi(\mathbf{A})$. But what if another person takes a different set of measurements and repeats the same statistical analysis? Will they get the same result as ours? No. For this reason, we also want to provide a *confidence interval*—a range within which results may lie—giving us an idea of the randomness in nature.

Fortunately there is a way to *resample* (or generate) new set of data without approaching your colleagues!. There are 2 methods to achieve this- **Bootstrap** and **Jacknife method**.
In Bootstrap Method, the *resampler* basically selects one data point from **A with replacement** N times, to create another dataset **B** and applies the statistics to get $\psi(\mathbf{B})$. This is done $m(\sim 10000)$ times to generate the bootstrap distribution of size m, denoted by $\mathbf{y}$. The algorithm is:

```
def bootstrap_dist(sample, func, m=10000):
    dist = np.zeros(m)
    for i in range(m):
        resample = np.random.choice(sample, size = sample.size)
        #without replacement
        dist[i] = func(resample)
    return dist
```

The error in Bootstrap method is given by,

$$\sigma_{\psi(\mathbf{A})} \approx \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{x})^2}$$

where $\hat{x} = \psi(\mathbf{A})$ and $\mathbf{y} = [y_1, y_2, \dots y_m]$.

Generally the confidence interval is defined by the **interquartile range (IQR)**. We may report our measurement(median or q_2 or α) as $\alpha^{q_3-q_2}_{q_1-q_2}$.[2]

Jacknife method works like, given N data points, it calculates the statistic N times using subsets of the data excluding one data point each time ($N-1$ size).

2.6 MCMC Fitting

The **Markov chain Monte Carlo** (MCMC) fitting isn't quite about finding the absolute best fit for a model, but rather understanding the probability distribution of possible fits. It works by creating a kind of random walk through possible parameter values for your model. This walk is informed by how well each parameter set fits our data, with better fitting sets having a higher chance of being chosen for the next step.(This is called **Metropolis Hastings Algorithm**).

Imagine a rough landscape where valleys represent good fits and hills represent bad fits. MCMC stumbles around this landscape, favoring valleys but occasionally going uphill to explore. By taking many steps, MCMC builds a map of where good fits are likely to be found. This map, the **posterior** distribution, tells us not just the single "best" fit, but also how likely other possible fits are. This is crucial for understanding the uncertainty in our model's predictions. It is a good practice to exclude the initial *burn-in*[3] steps ($\sim$ 1000 steps).

We are going to use `emcee` [4] python package (Foreman-Mackey et al., 2013), for implementing markov chains.

In this report, we shall perform only the traditional fitting of King (1962) profile to compute the structural parameters of the cluster.

2.6.1 King Profile

In any fitting, we always aim to maximize the **posterior** probability distribution $[P(\theta|D, M)]$ given the **data** (D) and **model** (M). We need to estimate the best possible value of the set of **parameters**(θ) (say, m parameters), for which the posterior is maximum. It follows from Bayes theorem,

$$P(\theta|D, M) = \frac{P(D|\theta, M)P(\theta|M)}{P(D|M)} \tag{2.7}$$

$P(D|\theta, M)$ is called **likelihood** - denoted by $\mathcal{L}$. It is this quantity that we seek to maximize in a method called *maximum likelihood estimation* **(MLE)**.
$P(\theta|M)$ is known as the **prior** of the distribution, and is denoted by π. This function encodes any previous knowledge that we have about the parameters like previously studied results in literature, physically acceptable ranges (like non-negative values) etc.

The term in the denominator is called **evidence** of the distribution, which can be calculated separately through *nested sampling*. It is the total marginal likelihood,

$$P(D|M) = \int P(D|\theta, M)P(\theta|M)\mathrm{d}^m\theta$$

[2] q_n refers to n^{th} quartile, where $q_1, q_2, \& \ q_3$ refers to 25th, 50th (median) and 75th percentiles respectively. For eg: `q3 = np.percentile(data,75)`

[3] think of it as the initial steps, that the *walkers* took to reach near the point of minima from their starting point.

[4] check out the tutorials at [https://emcee.readthedocs.io/en/stable/] to have a better understanding

It is **important to note** that, an mcmc *sampler* cannot draw parameter samples from likelihood function. This is because a likelihood function is a probability distribution over datasets so, conditioned on model parameters, we can draw representative datasets but not parameter samples.

Now, let's try to learn the implementation. We have

Data (D)

$$D_i = [x_i, y_i, \sigma_i]$$

where, x_i is the position of the center of the radial bins and y_i is the corresponding number density of stars within the annular region. The error σ_i is defined as $1/\sqrt{n_i}$, where n_i is the star count in each bin.

```
cluster_center_ra = np.median(data['ra'])
cluster_center_dec = np.median(data['dec'])
data['r'] = angular_dist(cluster_center_ra,cluster_center_dec,data['ra'],data['
    dec']) #distances in arcmin using Havershine Formula

# Calculate the number of stars in each annulus
counts, radial_bins = np.histogram(data['r'], bins=15)

#error in each point = 1/rt
density_err =1/np.sqrt(counts)
# Calculate the area of each annulus
annulus_areas = np.pi * (radial_bins[1:]**2 - radial_bins[:-1]**2)

# Calculate the number density in each annulus
radial_density = counts / annulus_areas
```

Model (M)

$$f(r) = f_{bg} + \frac{f_0}{1 + (r/r_c)^2}$$

where, r_c, f_{bg}, and f_0 are cluster core radius, background density level and central density level respectively.

```
def king_model2(r,f0, fbg,rc):
    return fbg + f0/ (1+ (r/rc)**2)
```

Parameters

$$\theta = [f_0, f_{bg}, r_c]$$

Defining Likelihood

Generally the data is assumed to be normally (= Gaussian) distributed about the model.

$$\mathcal{L} \equiv \prod_i \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\Delta Y_i^2/2\sigma_i^2}$$

However, it is often convenient to work with the log of exponential quantities, so we have

$$\ln \mathcal{L} = -\frac{1}{2}\left[\sum_i \ln 2\pi\sigma_i^2 + \sum_i \left(\frac{\Delta Y_i}{\sigma_i}\right)^2\right]$$

```
def log_likelihood(theta, r, density, density_err):
    f0, fbg, rc = theta
    model = king_model2(r, f0, fbg, rc)
    sigma2 = density_err**2
    return -0.5 * np.sum((density - model)**2 / sigma2 + np.log( 2 * np.pi *
    sigma2))
```

Defining Prior

For better convergence of chains, we construct an **uniform prior** for r_c and f_0, while a **gaussian prior** for f_{bg}. The following code will make it clear as we define $\log \pi$.

```
def log_prior(theta):
    f0, fbg, rc = theta
    if 0 < f0 < 50 and 0 < rc < 20 and fbg >0 :
        s_bg = 1.5
        mu_bg = 0.41 #litrerature value
        log_prior_fbg = -0.5 * (((fbg - mu_bg) / s_bg)**2 + np.log( 2 * np.pi *
     s_bg))
        return 0.0 + log_prior_fbg
    return -np.inf
```

Constructing Posterior/ Total probability

As the evidence is not dependent on parameters, we can safely exclude it as a constant.

$$\log P \simeq \log \mathcal{L} + \log \pi$$

```
def log_probability(theta, r, density, density_err):
    lp = log_prior(theta)
    if not np.isfinite(lp):
        return -np.inf
    return lp + log_likelihood(theta, r, density, density_err)
```

2.6.2 Maximum Likelihood estimation

$$\begin{array}{c} \text{Maximizing posterior} \\ \downarrow \\ \text{maximizing } \log \mathcal{L} \\ \downarrow \\ \text{minimizing } -\log \mathcal{L} \end{array}$$

We will use `minimize` function from `scipy.optimize` library.

```
nll = lambda *args: -log_likelihood(*args)
# initial = np.array([m_true, b_true]) + 0.1 * np.random.randn(2) #initial
    guess
initial = np.array([15,0,2])
soln = minimize(nll, initial, args=(bin_center, radial_density , density_err))
f0_ml, fbg_ml, rc_ml = soln.x
```

Initialize Sampler

To reduce the burn-in steps drastically and ensure rapid convergence it is advisable to initialize the sampler in a tiny gaussian ball (`pos`) around our MLE estimates for the parameters.

```
ndim, nwalkers = 3, 24
pos = soln.x + 1e-4 * np.random.randn(nwalkers, ndim)
```

Here we are specifying the number of dimensions as 3 ($= m$) and number of walkers to be 24 (8 per parameters). Now think of the walkers as *fingers* crawling out of each *hands* on the manifold (surface) of each parameter in an m ($= 3$)-dimensional space, each walkers[5] progressing towards their optimum values, after each step in the parameter space.

```
# Specify the number of steps for thhe sampler to run
nsteps = 10000

sampler = emcee.EnsembleSampler(nwalkers, ndim, log_probability, args=(
    bin_center, radial_density , density_err))
sampler.run_mcmc(pos, nsteps, progress=True)

# Get the samples
samples = sampler.get_chain(discard=2000, thin=35, flat=True) #discard first
    2000 as burn in steps
```

The final parameters and their errors are

```
f0_mcmc = np.percentile(samples[:, 0], 50)
f0_err = (np.diff(np.percentile(samples[:, 0], [16, 84])) / 2)[0]
...
```

Visualize your results

Corner plots are one of the most useful ways to visualise your mcmc fitting. In our report we have used `corner.py` (Foreman-Mackey, 2016) module to create the corner plots.

The corner plot displays all 1 and 2-D projections of the posterior probability distributions of your parameters. This is beneficial as it swiftly reveals the covariances between parameters. To find the marginalized distribution for a parameter or a set of parameters using MCMC chain results, we project the samples onto that plane and create an N-dimensional histogram. Consequently, the corner plot presents the marginalized distribution for each parameter independently in the histograms along the diagonal, and the marginalized two-dimensional distributions on the other panels.

[5] **how to find the ideal number of walkers?** A general rule-of-thumb is to have $2 \times N_{\rm dim}$ walkers. More walkers improve the exploration of the parameter space and convergence of chains, but at the expense of computational burden.

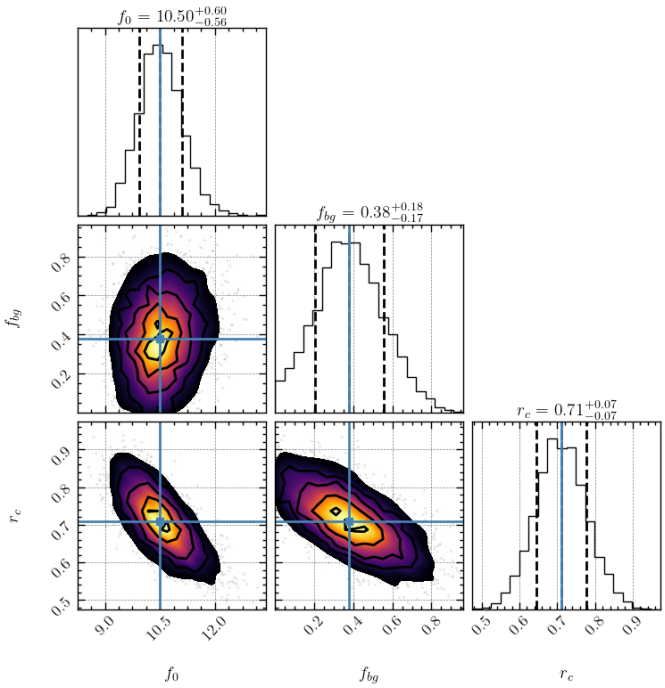

Figure 2.2: Closed ellipsoidal density plots indicates a successful convergence of all the parameters

The following code will reproduce the above plot:

```
import corner
import seaborn as sns
labels = [r"$f_0$", r'$f_{bg}$', r'$r_c$']
# Create a figure with a specified size
fig = plt.figure(figsize=(8,8))  # Adjust the figsize as needed

fig = corner.corner(
    samples, labels=labels,
    truths=[f0_mcmc,fbg_mcmc,rc_mcmc],
    label_kwargs={"fontsize": 14},
    quantiles= [0.16,0.5,0.84],
    show_titles= True,
    title_kwargs={'fontsize':14},
    fig = fig
);

# Overlay KDE plots with a color map
axes = np.array(fig.axes).reshape((3, 3))

for i in range(3):
    for j in range(i):
        ax = axes[i, j]
        sns.kdeplot(
            x=samples[:, j], y=samples[:, i],
            ax=ax, fill=True, cmap='inferno',
            levels=100, thresh=0.05
        )

# Adjust the tick label size
for ax in fig.get_axes():
    ax.tick_params(axis='both', labelsize=13)  # Set the desired tick label
    size here
#plt.savefig('corner_ngc4337_king.png', dpi=500)
```

2.6.3 Advantages of MCMC

Some of the reasons, why MCMC proves to be powerful in comparison to other methods are:

1. **Robustness in Complex Models:** MCMC handles non-linear and high-dimensional parameter spaces effectively, managing complex models where traditional methods may fail.

2. **Comprehensive Uncertainty Quantification:** Provides full posterior distributions and credible intervals, offering a detailed understanding of parameter uncertainties.

3. **Incorporation of Prior Information:** Naturally integrates prior knowledge about parameters, useful in cases with sparse or noisy data (Not possible in ordinary least squares fitting).

4. **Exploration of Multi-modal Distributions:** Capable of finding multiple peaks in the likelihood surface, ensuring thorough exploration of parameter spaces.

5. **Adaptability and Customization:** Customizable algorithms (e.g., Metropolis-Hastings, Gibbs sampling) and reliable convergence to true posterior distributions.

Chapter 3

Methodology and Results

3.1 Dolidze 28 in Literature

In 1960s, Russian astronomer Madona V. Dolidze at the Abastumani Astrophysical Observatory located in Georgia, created a catalog of 57 open star clusters based on his emission-line spectral surveys done with the observatory's 70cm f/3 Maksutov astrograph. These were called **Dolidze** objects. [1961a , Astronomy Circular AC 223, Dolidze M.V; Astronomicheskii Tsirkulyar, Vol. 382, p. 7-8 (1966)]. He marked Dolidze 28 as an suspected open cluster.

After thoroughly reviewing the recent literature on this cluster, we discovered that all studies focused on cluster objects identified either in the MWSC survey or the UCAC(5) Catalogue. In the MWSC survey, (Kharchenko et al., 2013) utilized the 2MAst and PPMXL surveys to develop a probabilistic pipeline for selecting unknown cluster candidates. And then mapped their colour magnitude diagrams to those isochrones of already studied clusters.

On the other hand, the UCAC(5)[1] (Zacharias et al., 2017) (United States Naval Observatory CCD Astrograph Catalog) was compiled by imaging the sky with a specially designed telescope equipped with a CCD camera. This catalog used a CCD that was photometrically calibrated using APASS DR2. Stars were considered in the UCAC catalog only if they were also present in the 2MASS data. It's worth noting that both APASS and 2MASS observations were ground-based, with 2MASS specifically operating in the infrared spectrum. This implies that these observations were subjected to a limiting magnitude of 12-14 mag only, indicating highly magnitude-limited observations.

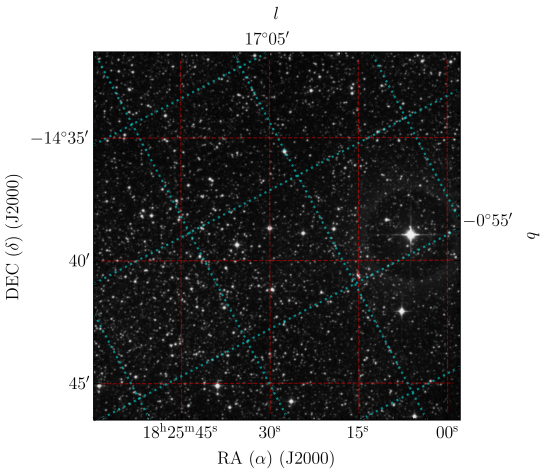

Figure 3.1: Identification map of cluster Dolidze 28 region taken from DSS

[1] http://tdc-www.harvard.edu/software/catalogs/ucac5.html#top

3.2 Observational Data

We have used ADQL to query our data (from https://gea.esac.esa.int/archive/) with the minimum possible relaxed restrictions.

```
SELECT *
FROM gaiadr3.gaia_source
WHERE CONTAINS(POINT(gaiadr3.gaia_source.ra,gaiadr3.gaia_source.dec),CIRCLE
    (276.370,-14.650,0.25))=1
AND abs(pmra_error)<0.5
AND abs(pmdec_error)<0.5
AND pmra IS NOT NULL
AND pmdec IS NOT NULL
AND parallax IS NOT NULL
AND parallax_over_error > 10
AND parallax >0
AND ruwe < 1.4;
-- gave 1502 rows
```

`ruwe` stands for **renormalized-unit-weight error**. This cut on `ruwe` ensures each data point is indeed a single star and not an unresolved binary!

We plot a vector motion diagram after a 3σ clipping on median PM to remove high PM foreground stars, to get the resulting diagram

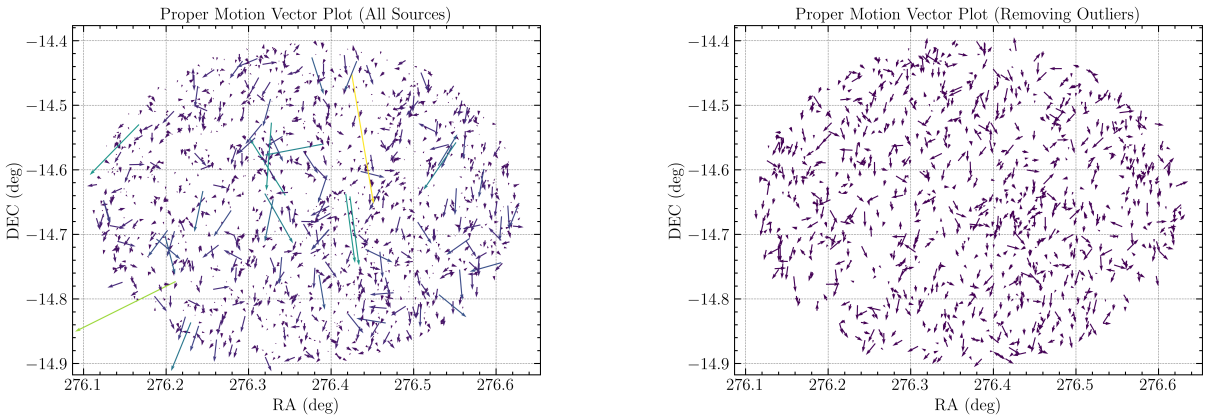

Figure 3.2: Left: Notice the blue, green and yellow coloured long arrows pointing in random directions. These are the foreground stars in our data set with high proper motion. These stars are easily removed by σ clipping. **Right:** We see that even after removing outliers, considerable number of stars having varying speeds in random directions, lacking the uniformity (even within field stars) which is expected from an open cluster region.

Next we perform DBSCAN clustering on Velocity Point Diagram (expecting all the cluster stars to have similar proper motion), followed by 3σ clipping about parallax median and implementing the probability models as suggested by (Balaguer-Núnez et al., 1998). We retained only those stars as **most probable cluster candidates** which have probability $> 90\%$. We report 421 stars as most probable. Note that, according to (Lindegren et al., 2021) the parallaxes have been corrected for the **gaia DR3 zero point offsets**[2], with the help of their code[3], which returns the estimated parallax zero-point offset given the ecliptic latitude, magnitude and color of any Gaia DR3 object.

[2] A calibration constant added to raw measurements to correct for systematic errors or biases inherent in the measurement device or procedure.

[3] https://gitlab.com/icc-ub/public/gaiadr3_zeropoint

$$M_G = G - 5 \log_{10} \frac{d}{10\ pc} - A_G$$

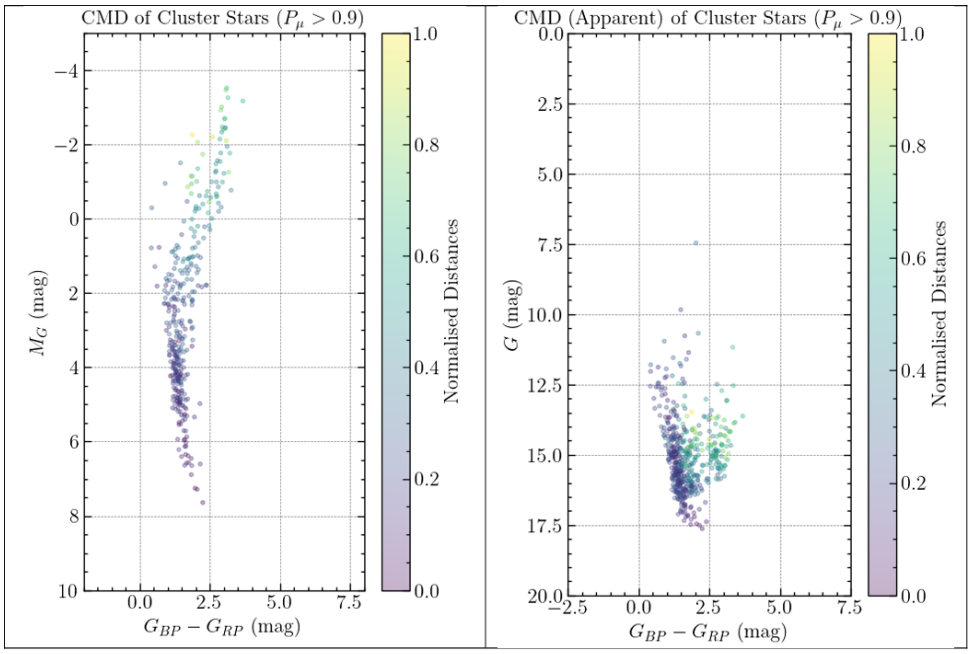

Figure 3.3: Left: Absolute Color Magnitude diagram (CMD) of *most probable* (mp) stars. **Right:** Apparent CMD of mp stars. A main sequence feature is visible with a faint red clump in both the plots. The presence of significant background contamination is obvious in these plots.

To ascertain whether it is consistent with other works in known open clusters, we perform a statistical validation which is discussed further in section 3.

3.3 Statistical Validation

3.3.1 Divide and Rule

We manually divide the apparent colour magnitude diagram [in Fig 3.3] (using TOPCAT software (Taylor, 2005)) into the Main Sequence (violet) and the clump of greenish points scattered to its right (possibility of red clump stars in the clusters) to analyse whether they also belong to our cluster. The number of members in each category is 215 (MS)and 196(others) respectively. Earlier, we had computed a distance of 1472.80 pc to the cluster (by inverting mean parallax). So we purposely clipped both these data sets between the 1st and 3rd quartiles (0.4821 and 0.7923 mas) respectively because this is where **statistically most of the distribution lies**.

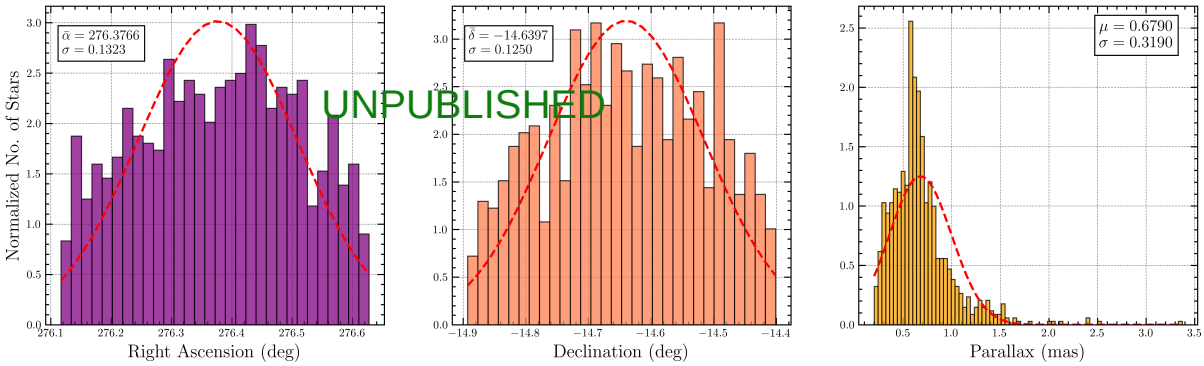

Figure 3.4: Cluster center and mean parallax was computed through gaussian fitting of the histograms. We fail to find any prominent peak in RA and DEC cluster density in histograms.

After clipping (the numbers are 136 and 42 respectively) we expect to see the proper motion velocities of cluster members (in both these data sets) to fall within a more concentrated region than before. But we see it is spread out over the same range as before clipping.

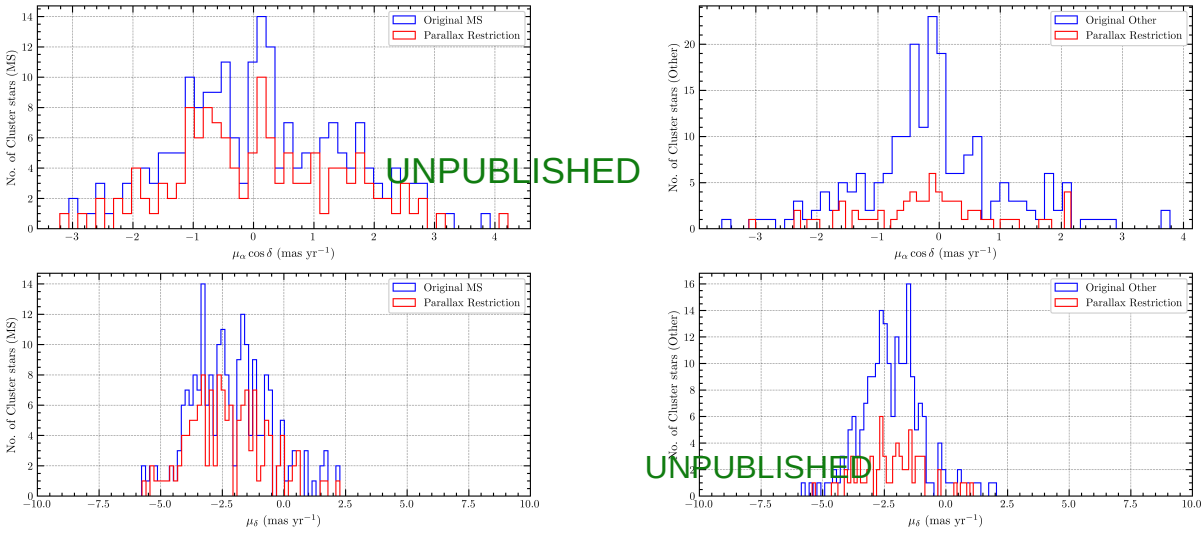

Figure 3.5: Depiction of the dispersion of proper motion before and after parallax restrictions in each of the two categories. **Left:** Main Sequence **Right:** Spread out Red clump region.

We consider a possible explanation for this existing contamination in CMD plots to be arising due to similar proper motion (magnitude) of background *field* stars alongwith cluster stars which is hindering our clustering based on proper motion alone. We now proceed to Field Star Decontamination.

3.3.2 **FSD**

After performing Field Star Decontamination (FSD) 3 times, we notice the resulting cluster numbers are reducing by half at each step - leaving us with too few stars at the end. This leads us to conclude a large fraction of our *identified* cluster stars have indeed the same properties as that of field regions, with no special microclustering observed in a rather broad Main sequence as well.

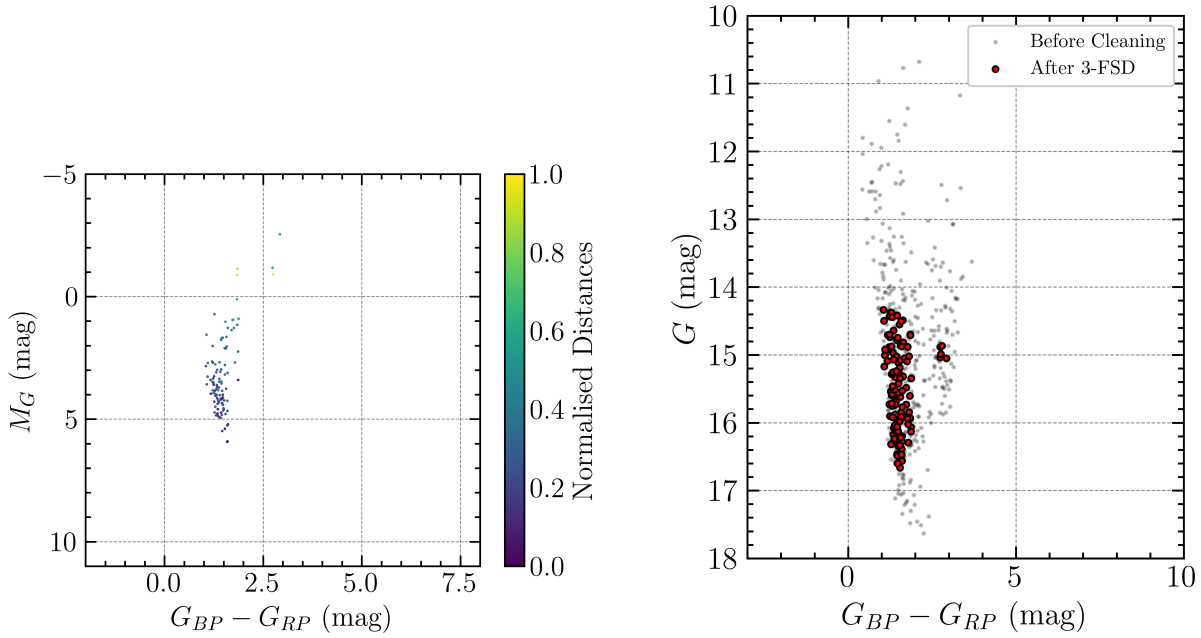

Figure 3.6: Left: Cleaned CMD of most probable cluster stars after 3 -FSD process. Higher FSD retain too few stars to make up any particular pattern in CMDs. Suggesting that a large fraction of our cluster stars indeed show similar photometric properties like nearby field regions. **Right:** A comparison plot to show the cluster members before and after 3-FSD cleaning.

CONCLUSIONS

After carefully studying the literature we find that there are enough reasons, particularly owing to Dolidze 28's position in the galactic disk, to conclude that it is an **outlier** in the *mass-scale* automated survey conducted by Kharchenko et al. (2013), (MWSC) and this **has not been validated** by any study since. Further the ground based catalogs, provided a **highly magnitude limited** observation, which leads to a high chance that these bright nearby **foreground stars** were the once which were classified as cluster objects.

We know that the most common stars in our galaxy are main sequence which are followed by slightly evolved red giant stars (forming red clump). This is to be particularly noted because, we had **reproduced** the same looking CMD and VPD even if we analyse the data from **from any other field region** (for example shifting 1° in RA). The huge loss of cluster stars during FSD process equally supports our suspicion, that the region of Dolidze 28 is **no less different than any other nearby field** (non cluster region). Also apart from all this, we also cannot find any significant **visual evidence** in either of DSS / 2MASS / allWISE/ PanSTARRS surveys. To support my claim, the statistical validation also provides a more robust argument alongwith a visual proof as well.

We try to determine a correlation between our results and Dolidze 28 for being an open cluster, but we find no evidence of the same. Thus my analysis indicates Dolidze 28 to be failing at every point to be a cluster.

References

Balaguer-Núnez, L., Tian, K., & Zhao, J. 1998, Astronomy and Astrophysics Supplement Series, 133, 387. https://doi.org/10.1051/aas:1998324

Binney, J., & Tremaine, S. 2011, Galactic dynamics, Vol. 13 (Princeton university press)

Campello, R. J., Moulavi, D., & Sander, J. 2013, in Pacific-Asia conference on knowledge discovery and data mining, Springer, 160–172

Deng, D. 2020, in 2020 7th International Forum on Electrical Engineering and Automation (IFEEA), 949–953, doi: 10.1109/IFEEA51475.2020.00199

Dhanush, S. R., Subramaniam, A., Nayak, P. K., & Subramanian, S. 2024, Monthly Notices of the Royal Astronomical Society, 528, 2274, doi: 10.1093/mnras/stae096

Foreman-Mackey, D. 2016, The Journal of Open Source Software, 1, 24, doi: 10.21105/joss.00024

Foreman-Mackey, D., Hogg, D. W., Lang, D., & Goodman, J. 2013, 125, 306, doi: 10.1086/670067

Girard, T. M., Grundy, W. M., López, C. E., & van Altena, W. F. 1989, Astronomical Journal (ISSN 0004-6256), vol. 98, July 1989, p. 227-243. Research supported by NSF., 98, 227. https://adsabs.harvard.edu/full/1989AJ.....98..227G

James Binney, M. M. 2021, Galactic astronomy (Princeton University Press)

Kharchenko, N., Piskunov, A., Roeser, S., Schilbach, E., & Scholz, R.-D. 2013, VizieR Online Data Catalog, J

Lindegren, L., Klioner, S., Hernández, J., et al. 2021, Astronomy & Astrophysics, 649, A2

McNamara, B., & Sanders, W. 1978, Astronomy and Astrophysics, 62, 259. https://ui.adsabs.harvard.edu/abs/1978A%26A....62..259M/abstract

Robusto, C. C. 1957, The American Mathematical Monthly, 64, 38. http://www.jstor.org/stable/2309088

Skrutskie, M., Cutri, R., Stiening, R., et al. 2006, The Astronomical Journal, 131, 1163

Stevenson, D. 2015, The Complex Lives of Star Clusters (Springer)

Taylor, M. B. 2005, in Astronomical data analysis software and systems XIV, Vol. 347, 29

Vallenari, A., Brown, A. G., Prusti, T., et al. 2023, Astronomy & Astrophysics, 674, A1

Zacharias, N., Finch, C., & Frouard, J. 2017, The Astronomical Journal, 153, 166