Insights into Open Star Clusters using Gaia DR3 Data
Presenter: Surinder Singh Chhabra
San Diego State University

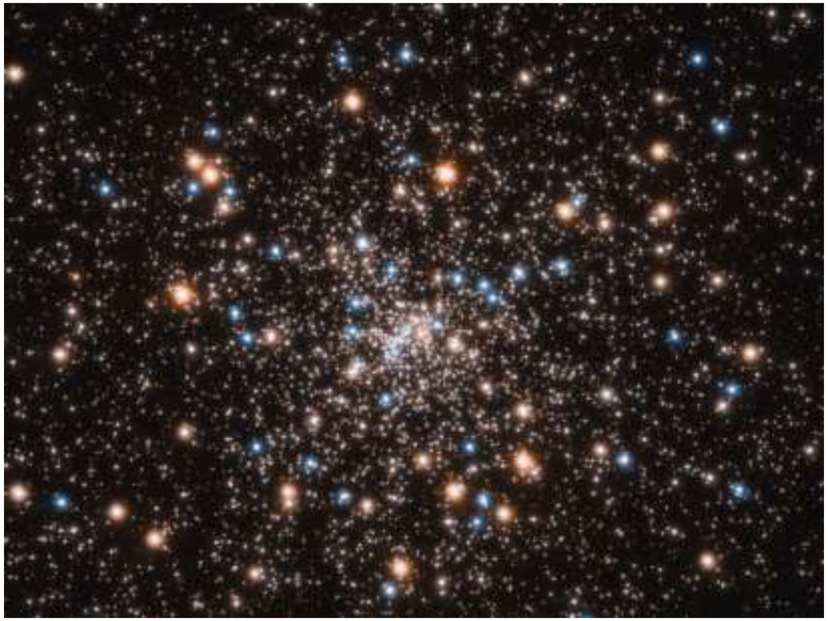
Gaia Cluster Analysis using K-Means and Bayesian Methods

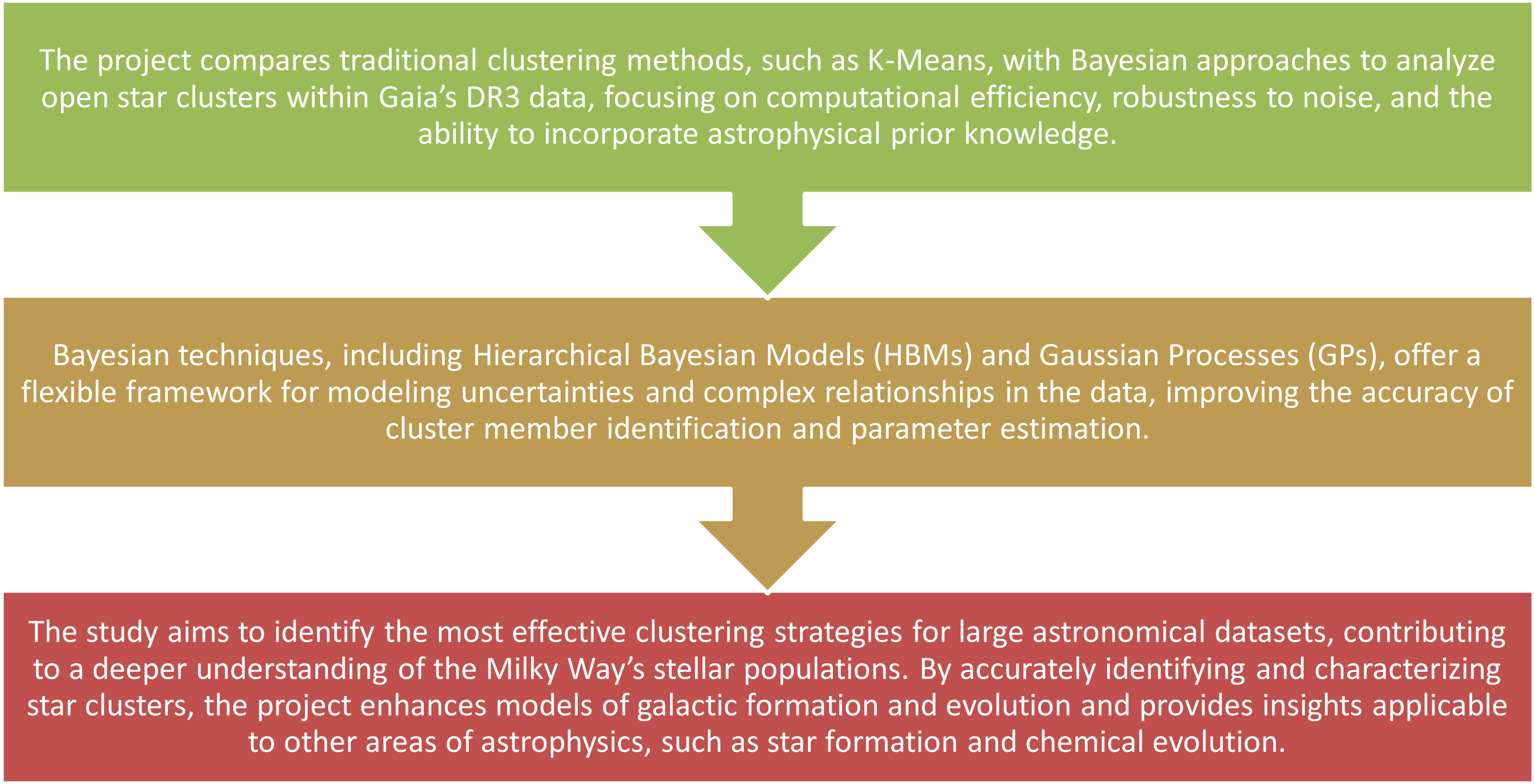
Abstract

The Gaia Cluster Analysis project uses advanced machine learning techniques, including K-Means clustering and Bayesian Gaussian Mixture Models, to analyze open star clusters from Gaia's DR3 data. By addressing challenges like noisy data and measurement uncertainties, it enhances cluster membership classification and distance estimation, contributing to the study of the Milky Way's galactic dynamics and stellar population.

Introduction

- Open star clusters, formed from the same molecular cloud, provide valuable insights into stellar evolution, galactic dynamics, and the structure of the Milky Way by sharing similar ages, chemical compositions, and kinematic properties.
- Gaia's Data Release 3 (DR3) revolutionizes star cluster research with high-precision astrometric and photometric data for over 1.3 billion stars, enabling detailed analyses of cluster properties such as distances, velocities, and member compositions.
- The vast size of Gaia’s dataset and measurement uncertainties pose challenges in identifying cluster members and estimating physical parameters, especially in the presence of overlapping clusters and field stars.





The project compares traditional clustering methods, such as K-Means, with Bayesian approaches to analyze open star clusters within Gaia’s DR3 data, focusing on computational efficiency, robustness to noise, and the ability to incorporate astrophysical prior knowledge.

Bayesian techniques, including Hierarchical Bayesian Models (HBMs) and Gaussian Processes (GPs), offer a flexible framework for modeling uncertainties and complex relationships in the data, improving the accuracy of cluster member identification and parameter estimation.

The study aims to identify the most effective clustering strategies for large astronomical datasets, contributing to a deeper understanding of the Milky Way’s stellar populations. By accurately identifying and characterizing star clusters, the project enhances models of galactic formation and evolution and provides insights applicable to other areas of astrophysics, such as star formation and chemical evolution.

Dataset Overview: Gaia DR3

- 1.8 billion celestial objects
- Astrometric, photometric, and spectroscopic features
- Key data: Parallax, proper motion, BP/RP photometry
- Improved accuracy and extended data over DR2

Methodologies

1. K-Means Clustering:

- Groups stars based on proximity using Euclidean distance.

- Iteratively optimizes cluster centroids.

2. Bayesian Gaussian Mixture Models (GMM):

- Probabilistic clustering with uncertainty handling.

- Uses the Expectation-Maximization algorithm for parameter estimation.

Data Preprocessing

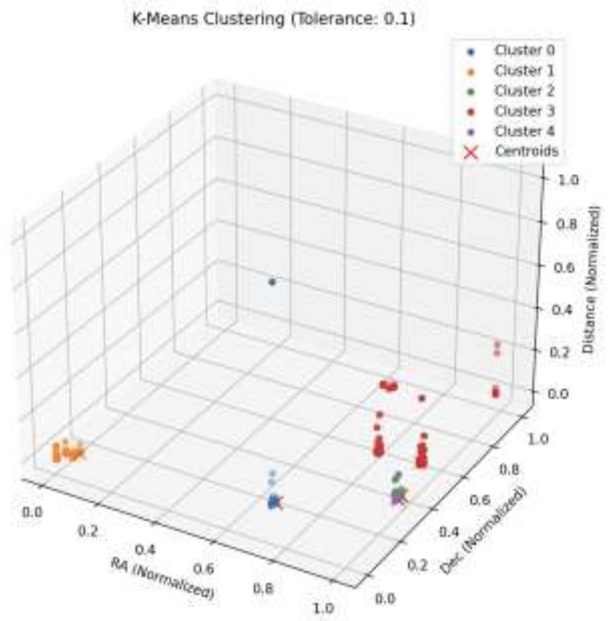
- Parallax converted to distance using inverse relation.

- Feature selection: Right Ascension (RA), Declination (Dec), and Distance.

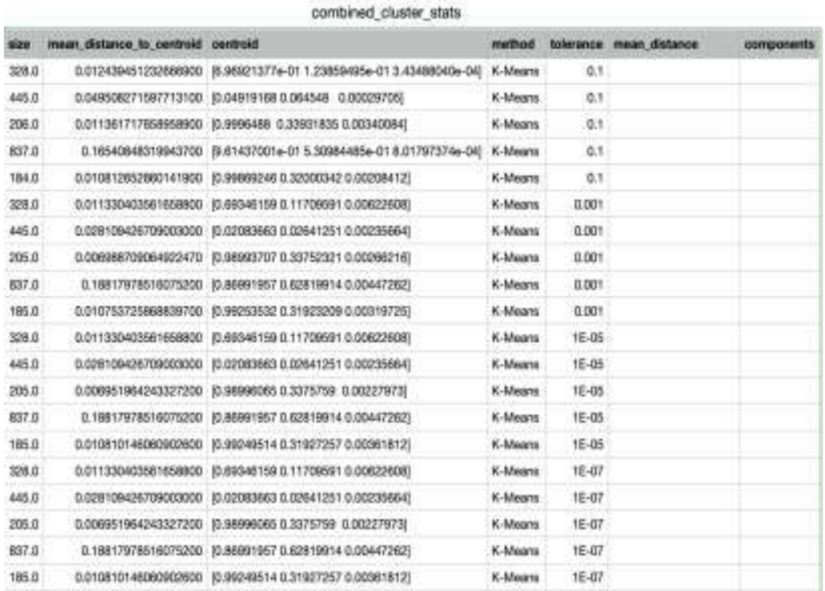
- Normalization ensures equal feature contribution during clustering.

Results: K-Means Clustering

- - Efficient for well-separated clusters.
- - Struggles with noisy data and overlapping clusters.
- - Best performance achieved with 1e-5 tolerance level.



Centroid and mean distance from centroids for k-means

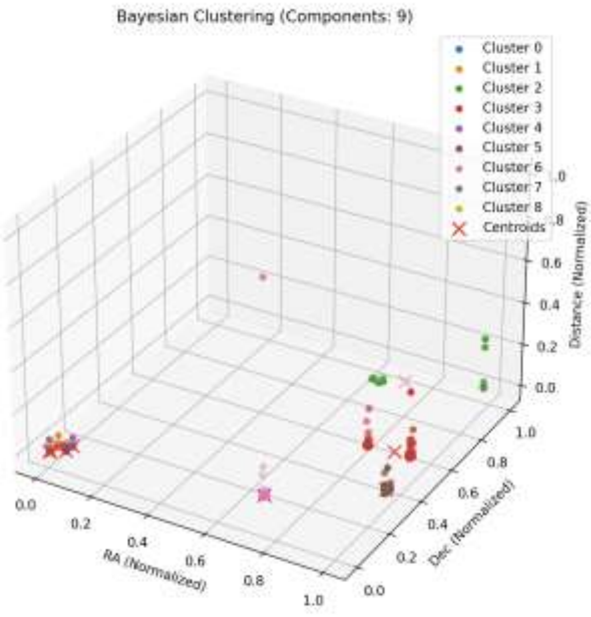


combined_cluster_stats

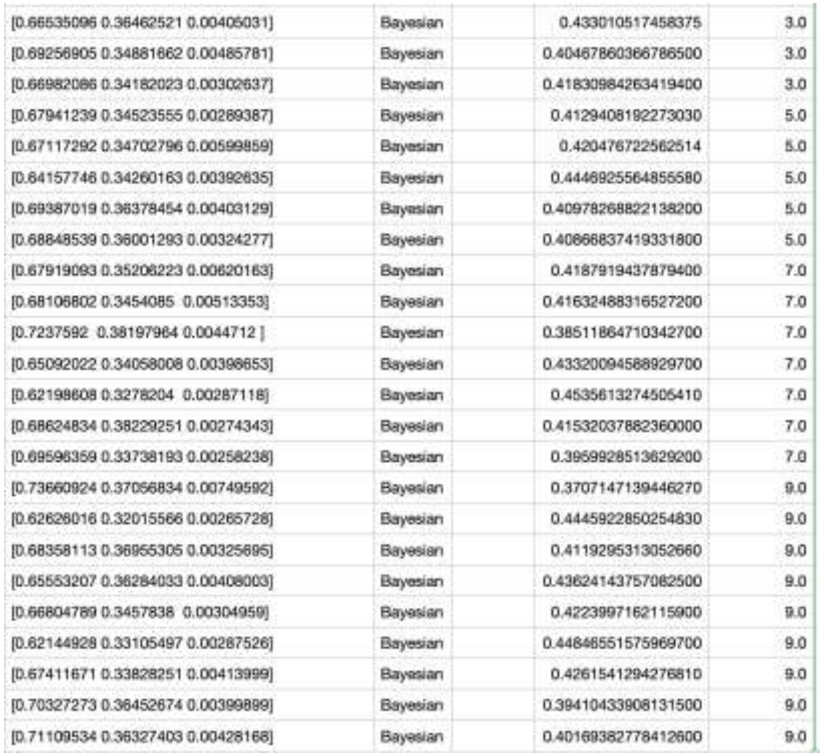
size	mean_distance_to_centroid	centroid	method	tolerance	mean_distance	components
328.0	0.012439451232666900	[8.96921377e-01 1.23859495e-01 3.43488040e-04]	K-Means	0.1		
445.0	0.049508271587713100	[0.04919168 0.064548 0.00029705]	K-Means	0.1		
206.0	0.011361717658958900	[0.9996488 0.33931835 0.00340084]	K-Means	0.1		
837.0	0.16540848319943700	[9.61437001e-01 5.30984485e-01 8.01797374e-04]	K-Means	0.1		
184.0	0.010812652880141900	[0.99869246 0.32000342 0.00208412]	K-Means	0.1		
328.0	0.011330403561658800	[0.69346159 0.11709691 0.00622608]	K-Means	0.001		
445.0	0.028109426709003000	[0.02083663 0.02641251 0.00235664]	K-Means	0.001		
205.0	0.006988709064922470	[0.98993707 0.33752321 0.00266216]	K-Means	0.001		
837.0	0.18817978516075200	[0.86991957 0.62819914 0.00447262]	K-Means	0.001		
185.0	0.010753725868839700	[0.99253532 0.31923209 0.00319725]	K-Means	0.001		
328.0	0.011330403561658800	[0.69346159 0.11709691 0.00622608]	K-Means	1E-05		
445.0	0.028109426709003000	[0.02083663 0.02641251 0.00235664]	K-Means	1E-05		
205.0	0.006951964243327200	[0.98996065 0.3375759 0.00227973]	K-Means	1E-05		
837.0	0.18817978516075200	[0.86991957 0.62819914 0.00447262]	K-Means	1E-05		
185.0	0.010810146060902600	[0.99249514 0.31927257 0.00361812]	K-Means	1E-05		
328.0	0.011330403561658800	[0.69346159 0.11709691 0.00622608]	K-Means	1E-07		
445.0	0.028109426709003000	[0.02083663 0.02641251 0.00235664]	K-Means	1E-07		
205.0	0.006951964243327200	[0.98996065 0.3375759 0.00227973]	K-Means	1E-07		
837.0	0.18817978516075200	[0.86991957 0.62819914 0.00447262]	K-Means	1E-07		
185.0	0.010810146060902600	[0.99249514 0.31927257 0.00361812]	K-Means	1E-07		

Results: Bayesian Gaussian Mixture Model

- - Handles uncertainties effectively.
- - Achieved highest log-likelihood with 5 components.
- - More computationally intensive compared to K-Means.



- **Centroid and mean distance from centroids for Bayesian**



[0.66535096 0.36462521 0.00405031]	Bayesian		0.433010517458375	3.0
[0.69256905 0.34881662 0.00485781]	Bayesian		0.40467860366786500	3.0
[0.66982086 0.34182023 0.00302637]	Bayesian		0.41830984263419400	3.0
[0.67941239 0.34523555 0.00289387]	Bayesian		0.4129408192273030	5.0
[0.67117292 0.34702796 0.00599859]	Bayesian		0.420476722562514	5.0
[0.64157746 0.34260163 0.00392635]	Bayesian		0.4446925564855580	5.0
[0.69387019 0.36378454 0.00403129]	Bayesian		0.40978268822138200	5.0
[0.68848539 0.36001293 0.00324277]	Bayesian		0.40866837419331800	5.0
[0.67919093 0.35208223 0.00620163]	Bayesian		0.4187919437879400	7.0
[0.68106802 0.3454085 0.00513353]	Bayesian		0.41632488316527200	7.0
[0.7237592 0.38197964 0.0044712]	Bayesian		0.38511864710342700	7.0
[0.65092022 0.34058008 0.00398653]	Bayesian		0.43320094588929700	7.0
[0.62198608 0.3278204 0.00287118]	Bayesian		0.4535613274505410	7.0
[0.68624834 0.38229251 0.00274343]	Bayesian		0.41532037882360000	7.0
[0.69596359 0.33738193 0.00258238]	Bayesian		0.3959928513629200	7.0
[0.73660924 0.37056834 0.00749592]	Bayesian		0.3707147139446270	9.0
[0.62626016 0.32015566 0.00265728]	Bayesian		0.4445922850254830	9.0
[0.68358113 0.36955305 0.00325695]	Bayesian		0.4119295313052660	9.0
[0.65553207 0.36284033 0.00408003]	Bayesian		0.43624143757082500	9.0
[0.66804789 0.3457838 0.00304959]	Bayesian		0.4223997162115900	9.0
[0.62144928 0.33105497 0.00287526]	Bayesian		0.44848551575969700	9.0
[0.67411671 0.33828251 0.00413999]	Bayesian		0.4261541294276810	9.0
[0.70327273 0.36452674 0.00399899]	Bayesian		0.39410433908131500	9.0
[0.71109534 0.36327403 0.00428168]	Bayesian		0.40169382778412600	9.0

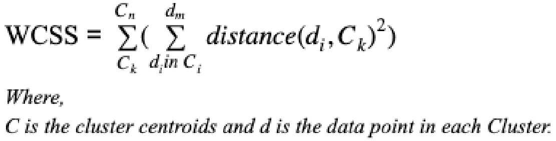
Comparative Analysis

- K-Means:

- - Computationally efficient.
- - Works well with simple datasets.

- Bayesian GMM:

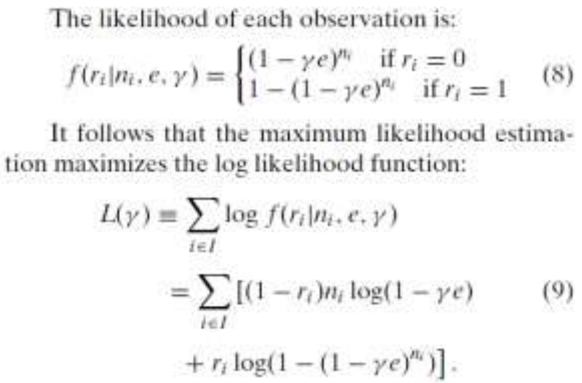
- - Robust against uncertainties.
- - Effective for complex datasets but computationally demanding.



$$\text{WCSS} = \sum_{C_k}^{C_n}\Big(\sum_{d_i\, in\, C_i}^{d_m} distance(d_i, C_k)^2\Big)$$

Where,

C is the cluster centroids and d is the data point in each Cluster.



The likelihood of each observation is:

$$f(r_i|n_i, e, \gamma) = \begin{cases}(1-\gamma e)^{n_i} & \text{if } r_i = 0\\ 1-(1-\gamma e)^{n_i} & \text{if } r_i = 1\end{cases} \qquad (8)$$

It follows that the maximum likelihood estimation maximizes the log likelihood function:

$$\begin{aligned} L(\gamma) &\equiv \sum_{i\in I} \log f(r_i|n_i, e, \gamma)\\ &= \sum_{i\in I} [(1-r_i)n_i \log(1-\gamma e) \qquad (9)\\ &\quad + r_i \log(1-(1-\gamma e)^{n_i})]. \end{aligned}$$

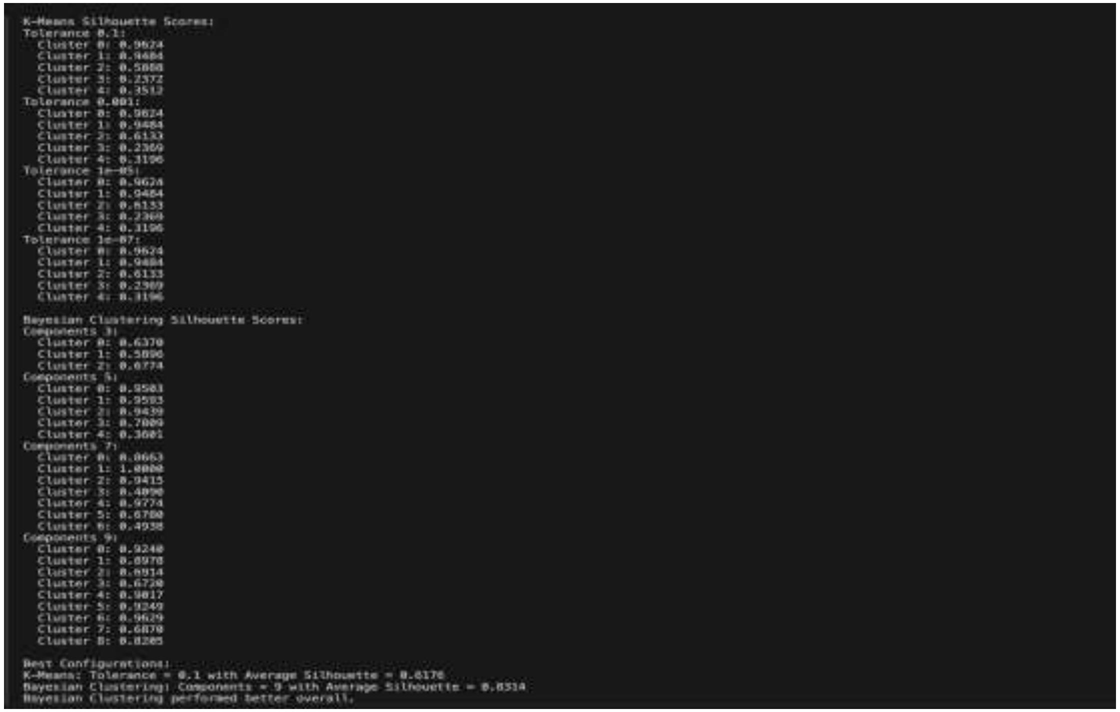
Output

```
(base) surindersinghchhabra@Mac 521 project % /opt/anaconda3/bin/python
"/Users/surindersinghchhabra/Desktop/521 project/main.py"
2024-12-09 16:31:02.760 python[29009:4229924] +[IMKClient subclass]: chose IMKClient_Legacy
2024-12-09 16:31:02.760 python[29009:4229924] +[IMKInputSession subclass]: chose
IMKInputSession_Legacy
K-Means clustering complete for tolerance 0.1. WCSS: 54.31338407113614
K-Means clustering complete for tolerance 0.001. WCSS: 38.316880653672946
K-Means clustering complete for tolerance 1e-05. WCSS: 38.31681661890941
K-Means clustering complete for tolerance 1e-07. WCSS: 38.31681661890941
Bayesian clustering complete for 3 components. Log-Likelihood: 9271.528747256441
Bayesian clustering complete for 5 components. Log-Likelihood: 15643.764318627638
/Users/surindersinghchhabra/Desktop/521 project/bayesian.py:42: RuntimeWarning: Degrees of
freedom <= 0 for slice
  cov_matrix = np.cov(cluster_points.T) + np.eye(d) * 1e-6  # Regularization term
/opt/anaconda3/lib/python3.12/site-packages/numpy/lib/function_base.py:2748: RuntimeWarning:
divide by zero encountered in divide
  c *= np.true_divide(1, fact)
/opt/anaconda3/lib/python3.12/site-packages/numpy/lib/function_base.py:2748: RuntimeWarning:
invalid value encountered in multiply
  c *= np.true_divide(1, fact)
Bayesian clustering complete for 7 components. Log-Likelihood: 18871.45219132887
Bayesian clustering complete for 9 components. Log-Likelihood: 20734.566195292766
Best K-Means configuration: Tolerance = 1e-05 with WCSS = 38.31681661890941
Best Bayesian (GMM) configuration: Components = 9 with Log-Likelihood = 20734.566195292766
Comparison of Methods:
K-Means WCSS: 38.31681661890941
Bayesian Log-Likelihood: 20734.566195292766
Bayesian (GMM) performed better for cluster identification.
```

Silhouette scores

$$s = \frac{b-a}{\max\left(a,b\right)}$$

- a(i): **Mean intra-cluster distance**.
- b(i): **Mean nearest-cluster distance**

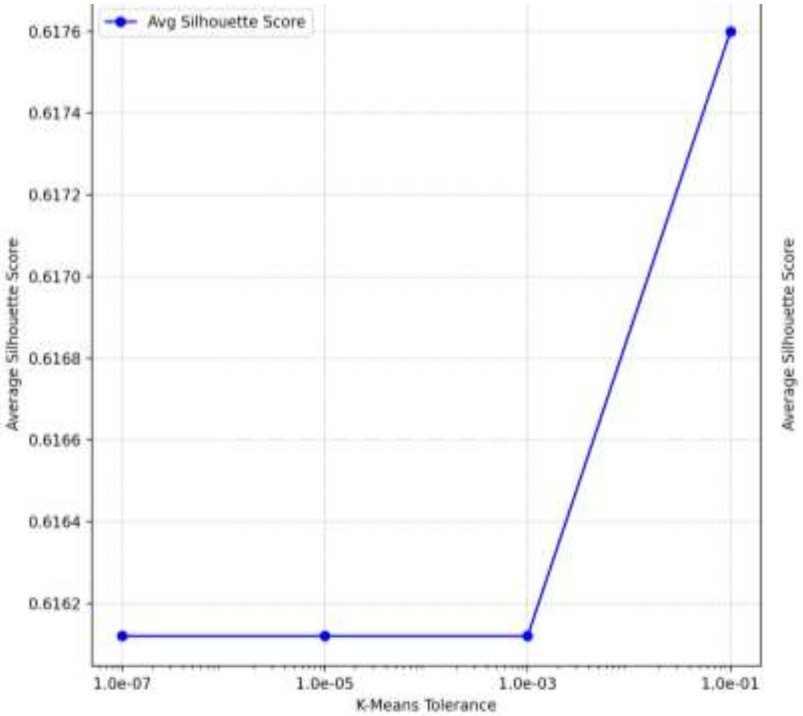


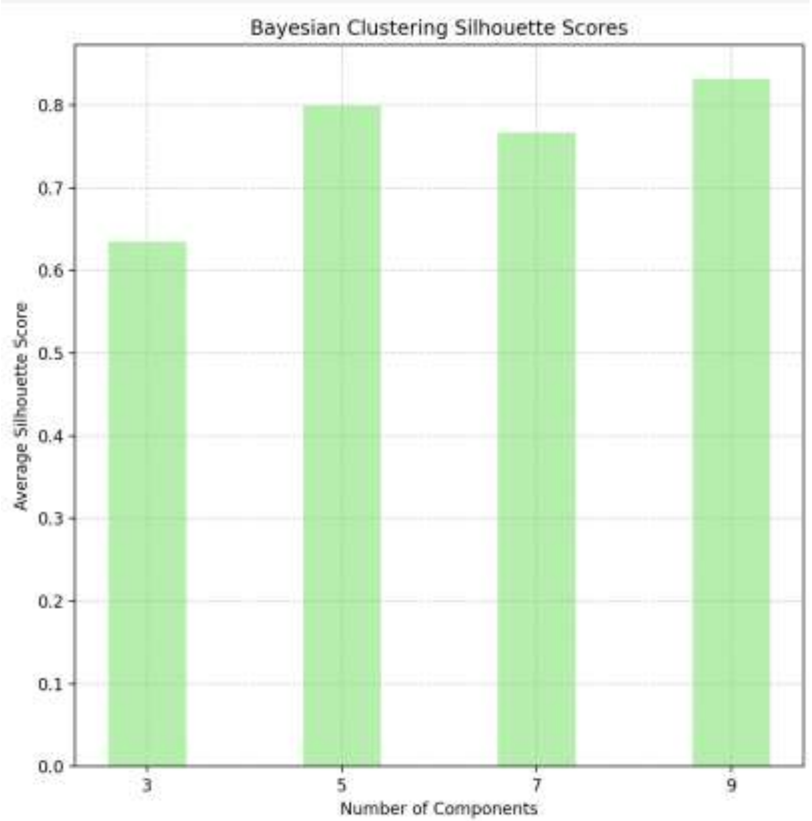
```
K-Means Silhouette Scores:
Tolerance 0.1:
  Cluster 0: 0.9624
  Cluster 1: 0.9484
  Cluster 2: 0.5888
  Cluster 3: 0.2372
  Cluster 4: 0.3512
Tolerance 0.001:
  Cluster 0: 0.9624
  Cluster 1: 0.9484
  Cluster 2: 0.6133
  Cluster 3: 0.2369
  Cluster 4: 0.3196
Tolerance 1e-05:
  Cluster 0: 0.9624
  Cluster 1: 0.9484
  Cluster 2: 0.6133
  Cluster 3: 0.2369
  Cluster 4: 0.3196
Tolerance 1e-07:
  Cluster 0: 0.9624
  Cluster 1: 0.9484
  Cluster 2: 0.6133
  Cluster 3: 0.2369
  Cluster 4: 0.3196

Bayesian Clustering Silhouette Scores:
Components 3:
  Cluster 0: 0.6370
  Cluster 1: 0.5896
  Cluster 2: 0.6774
Components 5:
  Cluster 0: 0.9503
  Cluster 1: 0.9593
  Cluster 2: 0.9439
  Cluster 3: 0.7009
  Cluster 4: 0.3601
Components 7:
  Cluster 0: 0.8663
  Cluster 1: 1.0000
  Cluster 2: 0.9415
  Cluster 3: 0.4090
  Cluster 4: 0.9774
  Cluster 5: 0.6700
  Cluster 6: 0.4938
Components 9:
  Cluster 0: 0.9240
  Cluster 1: 0.8978
  Cluster 2: 0.6914
  Cluster 3: 0.6720
  Cluster 4: 0.9017
  Cluster 5: 0.9249
  Cluster 6: 0.9629
  Cluster 7: 0.6870
  Cluster 8: 0.8285

Best Configurations:
K-Means: Tolerance = 0.1 with Average Silhouette = 0.6176
Bayesian Clustering: Components = 9 with Average Silhouette = 0.8314
Bayesian Clustering performed better overall.
```

Silhouette plots





Key Insights

- COMBINING K-MEANS AND BAYESIAN METHODS OFFERS A BALANCE BETWEEN EFFICIENCY AND ROBUSTNESS.

- PROVIDES NEW PERSPECTIVES ON GALACTIC DYNAMICS AND STELLAR POPULATIONS.

- DEMONSTRATES THE IMPORTANCE OF METHODICAL FLEXIBILITY IN DATA ANALYSIS.

Challenges and Future Work

- Challenges:

- - Scaling Bayesian methods for large datasets.
- - Computational costs and parameter tuning.

- Future Directions:

- - Explore Gaussian Processes for refined spatial modeling.
- - Extend methods to other astrophysical phenomena.

Conclusion

- Compared the effectiveness of K-Means and Bayesian methods.

- Highlighted contributions to understanding the Milky Way's stellar populations.

- Advanced methodologies for astronomical clustering.

References

1. Gaia Collaboration et al., 2023, Astronomy Astrophysics.

2. S. Lloyd, 1982, IEEE Transactions on Information Theory.

3. Astropy Collaboration, 2013, Astronomy Astrophysics.