INTRODUCTION, METHODS, AND MATERIALS

ALEC HEWITT

Introduction and Background

The Gaia database has data for over 1 billion stars. This is only roughly 1% of the stars contained in the Milky Way but there is still a plethora of data to explore some of the structures nature can produce. With this amount of data, we cannot rely on traditional programming methods to identify such patterns as this may take an incredibly long time. This is where machine learning comes in. The main software that will be used is FAISS (Facebook AI Similarity Search) which is a C++ library but also contains an interface in Python; Python is the language used in this project. The idea to apply machine learning to astronomical data for this project came from Maurice Garcia-Sciveres who saw an analogy between particle physics and astro-physics.

In particle physics, when particles collide they fly off into particle detectors which detect the positions of these particles at certain discrete points in space. To know the trajectories they must "connect the dots" to identify which particles came from where. This is a fairly simple task to connect the dots for a small number of particles, but once there are thousands of particles, such tasks become insurmountable. To deal with such a problem, they must figure out a way to systematically connect the dots. One such method is to take each dot and find the k-nearest neighbors of that dot, this turns into a combinatorics problem since we are finding every possible combination of dots and choosing the combinations that correspond to the smallest distances.

This can be a computationally intense problem. One way around this is to use a similarity search that returns the k-nearest neighbors of every particle; such a method requires machine learning. This project will use a similar technique to analyze data from Gaia eDR3 (2),(3). However, instead of "connecting the dots" of particle trajectories this project provides a tool to connect clusters, where the analogy of a particle in this project is a cluster. Such a tool could be used to locate and map out astronomical objects, such as stellar streams.

It is a trivial task to identify whether stars are similar in position, just look at the regions that are clustered. While these stars may contain similar properties, this project seeks to find clusters of stars in a similar but more subtle way. Gaia eDR3 has many useful properties that extend beyond just the position of the star. Only features within the Gaia database identified as "good" measurements and features that most stars possess are used. Once a dataset has been obtained, the similarity search can begin. This search involves using the Python package "FAISS" which stands for "Facebook AI Similarity Search".

A basic example of how this works is as follows: positions of a group of stars are given within a dataset, the objective is to find the clusters within this data set, i.e., the regions of the dataset that are the "clumpiest". FAISS has a function called "Inverted File Index" or "IVF". This function takes in the set of vectors, the number of clusters needed, and the number of data points desired for each cluster and spits out a list of clusters. This project will use a similar method except it will use higher dimensional vectors, however, this particular example can be useful as the human eye can identify such clusters and this can be used to ensure the methods are returning intuitively correct results.

The research group consists of Maurice Garcia-Sciveres, Xiangyang Ju, and Alec Hewitt. Maurice is the mentor of this project, Xiangyang is the associate mentor and Alec is the intern. The purpose of the research group is to have meetings at least once a week to brainstorm. During these meetings, the intern discusses what progress has been made, possible issues that were encountered, and what can be improved. The mentor and the associate mentor then evaluate the progress and determine whether the intern is on the right track or whether his methods and direction need to be modified. The results for that week are evaluated and a list of tasks is created to complete by the next meeting. The intern takes these tasks and attempts to complete them promptly and regular updates are given throughout the week.

Any details that were not covered in the meeting are expanded upon by the intern by using their best judgment. Questions regarding the tasks of the assignment are generally directed towards the associate mentor while questions regarding the overall direction of the project are directed towards the mentor.

Methods

Obtaining and Preprocessing the Dataset

The initial dataset was obtained through the Gaia archive using "ADQL" for the search, (1) contains many helpful examples on how this is done. Here we retrieve all features for stars with parallax>33.333333, which corresponds to a distance estimate (reciprocal of parallax) of $<$ 30 pc.

Once the dataset was obtained, some measurements are missing and are given by "Nan". For this analysis only features that have measurements for more than 92% entries are used to identify clusters. Although the positions are not used to identify clusters, they will be used to plot and connect clusters.

The data is further constrained by requiring:

$$\begin{cases} \text{abs('phot\_g\_mean\_flux\_over\_error')} > 3, \\ \text{abs('phot\_bp\_mean\_flux\_over\_error')} > 3, \\ \text{abs('phot\_rp\_mean\_flux\_over\_error')} > 3, \\ \text{abs('astrometric\_excess\_noise')} < 3, \end{cases} \tag{1}$$

and 'phot_proc_mode'==0. The corresponding positions were transformed from galactic coordinates to cartesian, in the galactic basis. The distributions for the accepted features were obtained and analyzed, to determine how to standardize these quantities. To standardize these quantities something similar to the standard deviation must be defined. The distributions for accepted features were not all gaussian so it was decided to use the fullwidth half max as a substitute for the standard deviation. The standardized quantities were thus determined to be $\frac{d_{ij}-\langle d_j \rangle}{\sigma_j}$, where d_{ij} is the ith entry of the jth feature and sigma is the full width half max for the jth accepted feature.

Obtaining Clusters

Before obtaining the clusters a crucial question to ask is what is the most natural number of clusters in the dataset? There are many solutions to this problem and one of them is known as "Silhouette clustering". The silhouette score is a number that represents how well the data is grouped into n clusters and ranges between -1 and 1, a value of 1 means that the data is grouped perfectly into those n clusters, so a higher value represents a better grouping. the data was grouped into clusters for the total number of clusters varying from 2 to 10 and the silhouette score for each grouping was calculated. The value of n corresponding to the highest silhouette score was used to group the data. (it was noticed that a value of n=2 always corresponded to the best number of clusters so this number of clusters was used moving forward)

To obtain the clusters of the dataset the python packages "faiss" and "sklearn" (4) were used. faiss's kmeans function was used and trained on the data. The centroids were then extracted where the number of centroids is given by the silhouette method. An inverted index was initialized using the "IndexIVFFlat" function from the faiss package and it was also trained on the data. This function sorts all data into n clusters, the purpose of using kmeans is to obtain the centroids. The data was sorted to the kmeans centroids using a similarity search on the centroids obtained from k-means. The number of neighbors specified was the length of the dataset. The number of neighbors is irrelevant if large enough since the index will not place more stars in the cluster than the cluster contains. The return of this algorithm is a list of clusters, each entry contains indices for all members of that cluster, the entire set is a partition of the original indices.

Identifying significant clusters

These indices were used to obtain corresponding positions of all stars in the group of clusters. For each cluster, a line of best fit was fitted to the cluster using svd or singular value decomposition. The goodness of fit was determined using the coefficient of determination or r^2 value. Recall that the r^2 value typically ranges from 0 to 1 with 1 signifies that the data is fitted well by a line. The most significant cluster is defined as the cluster with the highest r^2 score.

Obtaining new datasets

The line of best fit is then used to calculate the intersection points with the dataset boundaries as follows. Suppose the line of best fit is given by

$$\vec{r}(t) = \vec{r}_0 + \vec{v}t, \tag{2}$$

suppose the line intersects the boundaries of the dataset (which is a ball centered at $(0, 0, 0)$ in position space) at t_0, then the line must intersect the sphere at

$$(x, y, z) = (x_0 + v_x t, y_0 + v_y t, z_0 + v_z t) \tag{3}$$

Suppose the boundary of the dataset is given by

$$x^2 + y^2 + z^2 = r^2 \tag{4}$$

where r is the radius of the dataset. Inserting (x, y, z) into the equation of the sphere allows us to find t_0 and thus the corresponding intersection points (x, y, z).

These intersection points are then inverted back into galactic cordinates, i.e., $(x, y, z) \to (r, l, b)$. These points are used to obtain another dataset. Since there are two t_0 values let them be given by $t_{01} >= t_{02}$ and corresponding galactic coordinates of (r_0, l_{01}, b_{01}) and (r_0, l_{01}, b_{01}). The two new datasets are obtained through the Gaia archive by using the constraints:

$$\begin{cases} 16.6666 < \text{parallax} < 33.333333, \\ b_{01} - \theta < b < b_{01} + \theta, \\ l_{01} - \theta < b < b_{01} + \theta \end{cases} \tag{5}$$

for the first dataset and

$$\begin{cases} 16.6666 < \text{parallax} < 33.333333, \\ b_{02} - \theta < b < b_{02} + \theta, \\ l_{02} - \theta < b < b_{02} + \theta \end{cases} \tag{6}$$

for the second dataset. These datasets are extracted and the steps "Preprocessing the dataset", and "Obtaining clusters" are repeated (the direction of the project becomes fuzzy at this point).

Following the cluster

In these new data, lines of best fit are calculated for each cluster. To identify where the original significant cluster (seed) is headed, each of the lines from the new data are compared with the line corresponding to the seed. To extend the original cluster the new clusters must share certain similarities. First, they must be close in feature space and they must be aligned in the same direction in real space. One way to determine whether they are close in feature space is to throw the test cluster into the original dataset and see if it is placed in the original cluster as the seed. To determine whether they are aligned in the same direction, the direction of the lines of best fit can be determined via the dot product. Suppose the seed has line $\vec{r}_0(t) = \vec{b}_0 + \vec{v}_0 t$ and the test cluster has line $\vec{r}_1(t) = \vec{b}_1 + \vec{v}_1 t$, then if they are nearly aligned in the same direction then $\frac{\vec{v}_0 \cdot \vec{v}_1}{||\vec{v}_0||||\vec{v}_1||} \approx 1$

References

(1) https://gea.esac.esa.int/archive-help/adql/examples/index.html

(2) Gaia Collaboration et al. (2016b): The Gaia mission (provides a description of the Gaia mission including spacecraft, instruments, survey and measurement principles, and operations);

(3) Gaia Collaboration et al. (2020b): Gaia EDR3: Summary of the contents and survey properties.

(4) @articlescikit-learn, title=Scikit-learn: Machine Learning in Python, author=Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E., journal=Journal of Machine Learning Research, volume=12, pages=2825–2830, year=2011