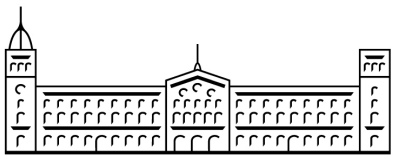

Treball final de grau

GRAU DE MATEMÀTIQUES

Facultat de Matemàtiques i Informàtica
Universitat de Barcelona

Advanced clustering techniques for the IDU cross-match in the ESA Gaia mission

---

Author: Ferran Torra Clotet

Advisor:	**Dra. Rosa Ma. Miró-Roig**
Co-advisor:	**Juan José González-Vidal and Marcial Clotet**
Made at:	**Departament de Matemàtiques i Informàtica**

Barcelona, **January 17, 2017**

Abstract

The cross-match (XM) is a sophisticated process that provides a link between every Gaia observation and the corresponding source in the catalogue. In this work, we describe a generalized method based on clustering analysis for a clustering stage of the Gaia XM, including additional parameters such as magnitude and proper motion. The performance of the implemented algorithm is assessed through real-case examples using Gaia data, and the successful results that were obtained demonstrates that the system behaves as expected.

Resum

El cross-match (XM) és un procés sofisticat que assigna cada observació de Gaia amb la font de llum del catàleg corresponent. En aquest treball, descrivim un mètode generalitzat basat en l'anàlisi de clústers pel XM, incloent paràmetres adicionals com la magnitud i el moviment propi. El rendiment de l'algoritme implementat s'ha avaluat en casos reals usant dades de Gaia, i els resultats satisfactoris obtinguts demostren que el sistema es comporta com s'esperava.

Agraïments

Primer de tot, agraeixo a la Dra. Rosa M. Miró-Roig, la seva dedicació i els oportuns comentaris en aquest projecte des del primer moment malgrat no fos un treball del seu camp.

També agraeixo a tot l'equip Gaia de l'IEEC-UB l'oportunitat d'haver pogut formar part de l'equip. En especial al Marcial, per haver-me iniciat en el XM de Gaia; i al Juanjo, per les hores que m'ha dedicat durant aquests mesos.

Finalment, dono les gràcies a la meva família i amics, pel seu suport i ajuda per arribar fins aquí. En especial, als meus pares que m'han donat les facilitats per dur a terme aquest treball.

Contents

Introduction

Motivation

Gaia is a mission by the European Space Agency (ESA) designed to make the largest, most precise three-dimensional map of our Galaxy by surveying a billion stars with an unprecedented precision in position and motion. A large european team of scientists and engineers known as DPAC (Data Processing and Analysis Consortium) is responsible for the processing of Gaia's data with the final aim of producing the Gaia Catalogue. A fundamental system of the Gaia data reduction process is the Intermediate Data Updating (IDU), a massive data processing system running at the Barcelona Supercomputing Center (BSC) which calibrates the instrument response and refines image parameters and cross-matching (XM) for all Gaia detections.

The IDU-XM process must provide a link between every observation and the corresponding source in the catalogue, resolving the conflicting situations. The current IDU-XM algorithm has an observation clustering stage which precedes the resolution against the catalogue. The current version of the algorithm only uses the position to cluster the observations. The main objective of this project is to study, describe and perform a comparative analysis of the clustering results when including additional dimensions in the algorithm such as the magnitude or proper motion.

Structure

This work is organized in three different parts.

First, we present an overview of the Gaia Mission and the data processing. Specifically, all the stages of the IDU-XM task are described and the purposed parameters to include are defined, as well.

Second, we develop the clustering stage of the XM task based on clustering analysis. In this chapter, we analyze the source model and get an algorithm which is consistent with this model.

Finally, we show the results obtained with our algorithm on realistic Gaia data.

Acronyms

The following table is a list of acronyms used in this document.

Acronym	Description
ESA	European Space Agency
HPM	High Proper Motion
IDU	Intermediate Data Update
MC	Match Candidate
MCG	Match Candidate Group
NN	Nearest Neighbor
NNC	Nearest Neighbor Chain
RNN	Reciprocal Nearest Neighbor
SSR	Sum of Squared Residuals
XM	Cross-Match

Chapter 1

The Gaia Mission

Gaia is a global astrometry mission of the European Space Agency (ESA) adopted by ESA's Science Programme Committee in October 2000 and was launched on 19 December 2013. This ambitious mission will accurately measure the positions and motions of a huge number of stars and galactic objects down to magnitude 20 with a precision of the angular measurements about 20 μas at magnitude 15. Therefore, Gaia will obtain a precise three-dimensional map of more than 1 billion stars of our Galaxy (approximately 1% of the stars populating the Milky Way) which will be a crucial tool to reveal the composition, formation and evolution of the Galaxy.

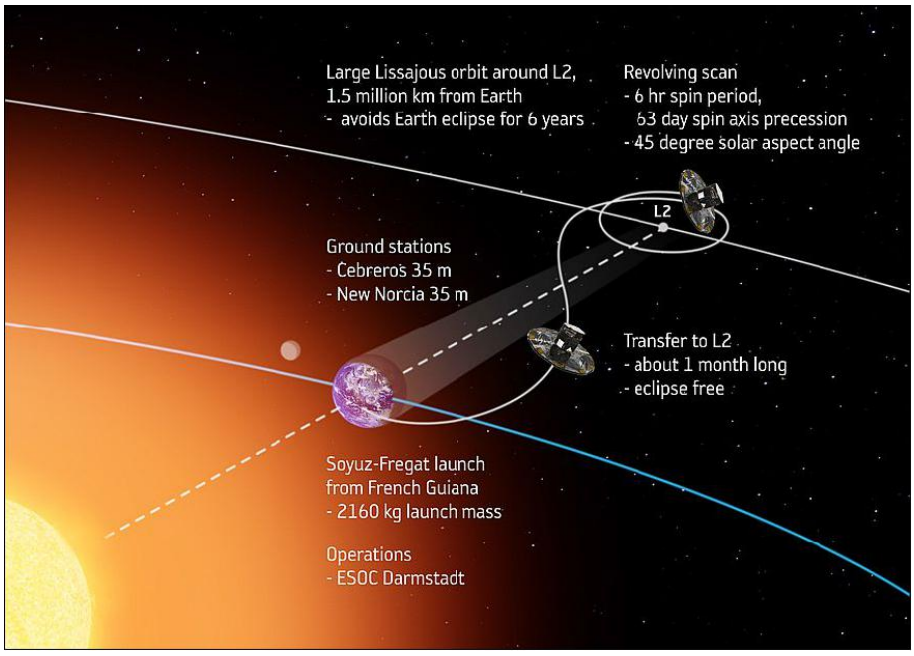

Figure 1.1: Overview of Gaia's launch and its operational orbit. Credit: ESA.

Gaia orbits around the L2 Lagrangian Point and scans the full sky because it is spinning on its own axis, which itself precesses at a fixed angle of 45 degrees with respect to the Sun-Satellite line as shown in Fig. 1.1. Thus, Gaia will observe each of the point-like sources from our Galaxy and beyond about 75 times over a five-year period recording the brightness, colour and position of each observation.

To fulfill its objectives, the Gaia spacecraft is composed of two main parts: the payload module and the service module.

- The service module contains electronic units to run the instruments, as well as the propulsion system, communication units and other essential components like monitoring systems.
- The payload module contains two telescopes which are combined onto a single focal plane with a total of almost 1 Gigapixels and physical dimensions of $0.5m \times 1m$.

The focal plane (see Fig. 1.2) is composed of 106 state-of-the-art Charge Coupled Devices (CCDs) which are served for the wave-front sensor (used to measure the optical quality of each telescope), the basic-angle monitor (used to measure fluctuations in the angle between the two telescopes), the Sky Mapper (which detects the incoming objects) and used for the three science instruments: the astrometric field (devoted to measuring stellar positions), the photometric (which obtains the colour information) and the radial velocity spectrometer (which measures the Doppler shift of absorption lines).

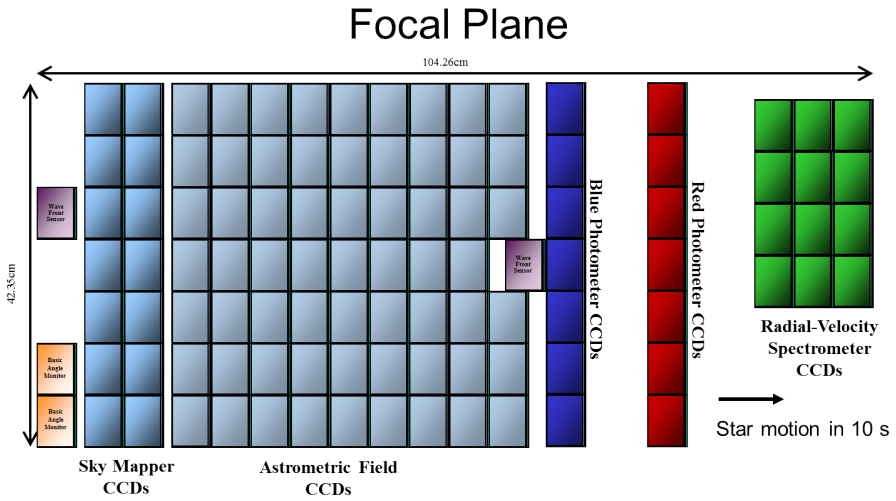

Figure 1.2: Schematic of the Gaia focal plane. Credit: ESA.

The data generated in the focal plane is transmited by low gain antena to one of Gaia's ground stations located in Cebreros (Spain), New Norcia (Australia) and Malargüe (Argentina) in order to process these data.

The Gaia mission will produce more than 100 Tb of raw data and it is estimated that it will consume a processing power of over 10^{21} flops (Floating-point Operations), therefore the efficient processing of the data generated by Gaia is a true challenge.

1.1 Ground-segment processing

The task of the data reduction is entrusted to a multidisciplinary team of more than 400 scientists and software engineers organised themselves in the Gaia Data Processing and Analysis Consortium (DPAC). This large pan-European team develops the data processing algorithms with the final objective of producing the Gaia Catalogue around 2021. The consortium is structured around nine specialist units known as Coordination Units (CUs) with each unit being responsible of a particular sub-system of the overall Gaia data processing system. In addition, each CU is supported by Data Processing Centers (DPCs) where the actual computer hardware for processing is available.

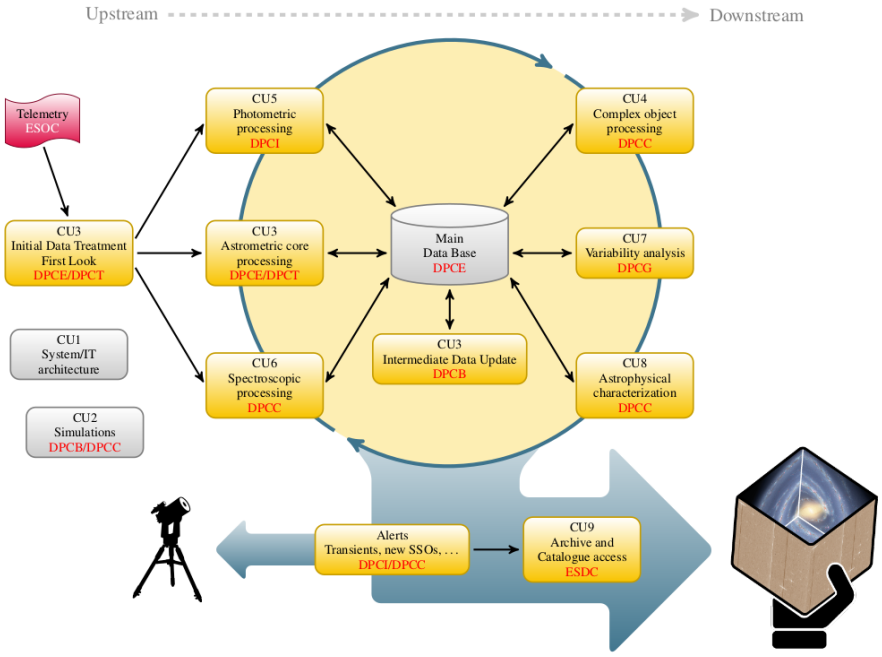

Figure 1.3: Overview of the structure of DPAC showing the data reduction process and its organization in CUs and DPCs (locations of the latter in red). Credit: ESA.

In particular, CU3 (core processing) covers the data processing chain all the way from the raw telemetry to the astrometric core solution. More specifically, CU3-UB group manages and develops the following software:

- The Initial Data Treatment (IDT), which processes all the raw telemetry coming from

Gaia in a real-time manner, transforming it to a ligher-level set of data including a satellite attitude, image parameters and a preliminary Cross-Match (XM).

- The Intermediate Data Updating (IDU) aims to refine intermediate astrometric data and the XM in a cyclic maner using the latest and most accurate calibrations and source catalogues. Moreover, some of the major Gaia calibrations tasks are included in IDU. Finally, it is involved in the main iterative process within the Gaia data reduction which also involves other cyclic systems like AGIS and PhotPipe.

Specifically, IDU is deployed in the Data Processing Center of Barcelona (DPCB) which uses Barcelona Supercomputing Center (BSC) resources and concretely the MareNostrum supercomputer, one of the most powerful supercomputers in Europe.

1.2 Cross-matching Gaia objects

Apart from the position coordinates, the gaia objects have other interesting quantities which may be used in the XM, such as the magnitude or the movement of the source.

Magnitude is a measure of brightness of a source or an observation and it is available from the Gaia data. The magnitude system uses a logarithmic scale where brighter objects have smaller magnitude than fainter ones.

In addition, since the stars are moving (see Fig. 1.4), the sources have to include quantities related with this motion.

More specifically, there is the proper motion which is an angular velocity of a star across the sky and it is generally measured in seconds of arc per year.

Note that the proper motion is not entirely intrinsic to the star because it includes the motion of the Sun. Despite of that, the name comes from that it is not the observed motion due to Earth's movement.

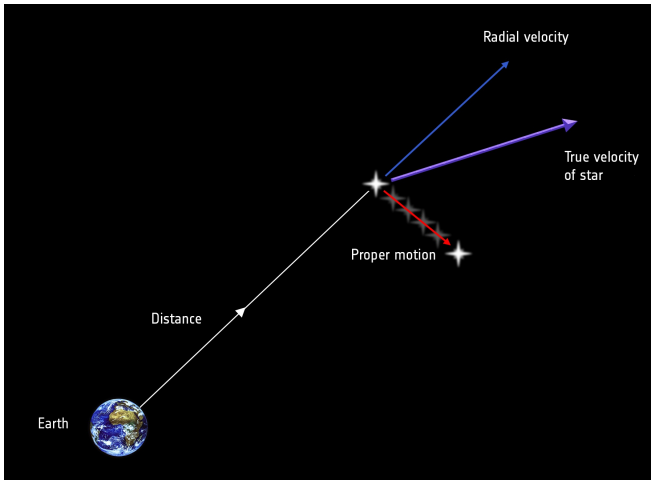

Figure 1.4: Illustration of the motion of a star. Credit: ESA.

Proper motion provides two of the three components of a star's velocity. The other one is the radial velocity which is the component in the direction of the radius between the star and the Sun, and it is measured with the radial velocity spectrometer.

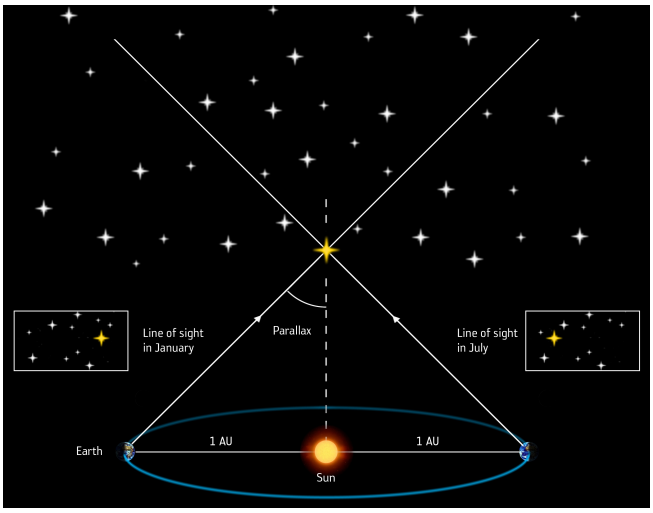

Figure 1.5: Illustration of the parallax between two observations which are separated by six months. Credit: ESA.

In addition, the star's motion includes the parallax which is the displacement in the apparent position of a star created by the relative motion between the Earth and the star.

Its measurement permits to determine the distance to the stars aplying trigonometry with the distance Sun-Earth as shown in Fig. 1.5.

Focusing on the IDU-XM task, the purpose of the XM is to provide the links of the individual observations with the corresponding source from the catalogue.

As mentioned above, during IDT a preliminary XM is carried out but the scientific quality of this XM may be deficient. Thus, the iterative task in IDU is necessary to get a consistent XM.

The inputs to the XM task are the following:

- Gaia Observations.
- Source catalogue.
- Calibrations and ephemeris.
- Spacecraft attitude.

And the outputs of the XM task are:

- Match Table: table with exactly one source for each detection.
- Ambiguous Match: table with all the possible source candidates for each detection.
- New Source Table: this table contains the new sources that have been created during the XM.

- Track Table: table describing the action updates applied to the sources including source deletion and new source creation from scratch or as a result of a splitting or merging process, represented in Fig 1.6.

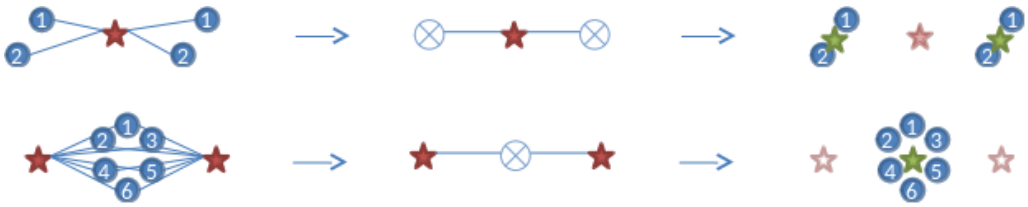

Figure 1.6: Representation of a splitting process (existing source split into split sources) at the top and a merging process (merged sources into a new one) at the bottom.

At the end of the mission, the number of detections will reach $\sim 10^{11}$. According to this huge number of detections, handling all this data in a single process is not a feasible approach. Therefore, the adopted solution in IDU-XM consists in the splitting of the task in three different stages (see Fig. 1.7):

- **Obs-Src Match**: this is a time ordered match stage where we identify all the possible matching sources for each individual detection. In this stage, we use the latest calibrations to compute the observation coordinates and then the sources are propagated to the observation epoch. Thus, all candidates sources are selected according to a pure distance criteria. Finally, the output of this task is a set of objects, which contains the basic detection parameters and all the candidates sources, the so-called Match Candidates (MCs). In other words, a MC is an observation with all the possible sources which it could be matched to.

- **Sky Partitioner**: the objective of this second stage is to create self contained groups of MC, the so-called Match Candidate Groups (MCGs). This is a spatial-based stage which provides an efficient spatial data arrangement because it avoids boundary effects. This stage acts as a bridge between the core time-based and the final XM resolution stage.

- **XM resolver**: this is the final stage which is a spatially based stage where all detections from a MCG are solved together, thus it takes into account all observations and sources of that region. Therefore, the XM is resolved and the final data are produced.

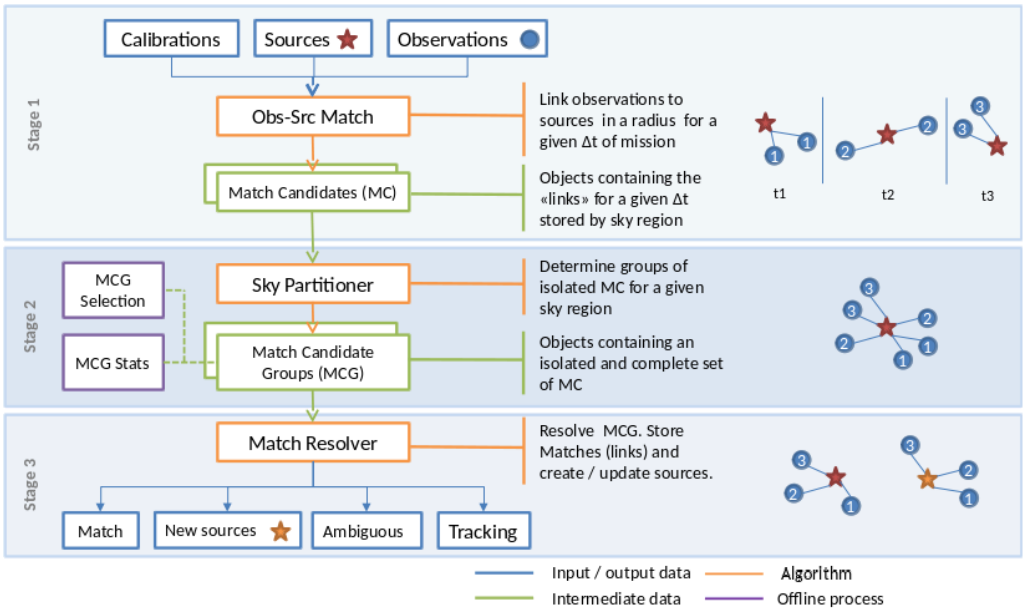

Figure 1.7: XM task overview.

More specifically, the XM resolver task is divided into three stages:

- **Clustering stage**: the purpose of this stage is to divide the observations in the MCG into smaller sets of observations (the so-called clusters[1]). The main of this stage is to produce clusters with all the observations that may correspond to the same source.
- **Cluster linking stage**: this task matches the generated clusters to candidate sources. Therefore, it creates a list of sources which might be assigned to the cluster, similar to the Obs-Src Match process but with clusters instead of single observations.
- **Conflict resolution stage**: this final step is intended to remove all conflicts between cluster-source links (see Fig. 1.6) and provide the final optimal resolution through an ad-hoc decision tree algorithm providing all the outputs mentioned above.

The following chapter describes the clustering stage algorithm given the requirements and restrictions. It provides an algorithm description that is suitable to be implemented as the clustering stage. Particularly, the match criteria in the proposed algorithm includes the magnitude and the movement of the source.

[1] A *cluster* is a set of objects tentatively associated with same label, in our case group of MCs.

Chapter 2

Generalized cross-matching based on clustering analysis

Cluster analysis aims to divide data into groups (the so-called clusters), where the objects in each cluster are similar between them and different from objects within other clusters.

First of all, we have to introduce a definition for the dissimilarity in accordance with the source model and promising for the cross-matching problem.

After that, we have to consider an algorithm to solve efficiently the XM task according to the source model which will have been explained.

Moreover, the XM should include methods to correct some possible conflictive cases as part of the post-process.

2.1 Source model

The aim of the cross-match is to link one source to each observation, so it first of all builds the clusters by the agglomeration of observations, and then it gets a one-to-one assignation between clusters and sources.

According to that, it is important to know the source model because it will establish how to make the cluster agglomeration; in other words, which agglomerative method has to be used and, accordingly, how is the proximity of each pair of objects defined.

The model pretends to be independent from other catalogues, so the input only consists of a set of observations and, therefore, the source model has to be interpreted as the number of observations in the cluster is increased. Following this premise, the agglomerative method has to be adapted to our type of data and it should be generalized to any linear model of arbitrary order.

In any case, in all methods of cluster analysis, the proximity between clusters has to be defined but, due to the hierarchical clustering procedures, it does not have to use a

distance, a less restrictive measure is enough.

Definition 2.1. *Let $\mathcal{C}$ be the set of all clusters, a dissimilarity on $\mathcal{C}$ is a function*

$$\Delta : \ \mathcal{C} \times \mathcal{C} \longrightarrow \mathbb{R},$$

and the following conditions are satisfied for all $C_i, C_j \in \mathcal{C}$,

1. $\Delta(C_i, C_j) \geq 0$ *and* $\Delta(C_i, C_i) = 0$,
2. $\Delta(C_i, C_j) = \Delta(C_j, C_i)$.

Note that a dissimilarity may not satisfy the triangle inequality.
Some hierarchical clustering methods use different types of dissimilarities,

- Single link: shortest distance between two individual members of the clusters,

$$\Delta(C_i, C_j) = \min \{d(O_i, O_j) \mid O_i \in C_i, \ O_j \in C_j\}.$$

- Complete link: farthest distance between two individual members of the clusters,

$$\Delta(C_i, C_j) = \max \{d(O_i, O_j) \mid O_i \in C_i, \ O_j \in C_j\}.$$

- Group average: average of the distances between all the individual members of the clusters,

$$\Delta(C_i, C_j) = \frac{1}{n_i n_j} \sum_{O_i \in C_i} \sum_{O_j \in C_j} d(O_i, O_j),$$

  where n_i (resp. n_j) is the number of observations in the cluster C_i (resp. C_j).

- Centroid: the dissimilarity is defined in terms of the distance between the cluster centers.

- Median: the dissimilarity is defined in the same idea as the centroid method but assuming that the clusters to be agglomerate are of equal size using the median.

- Ward's method: the dissimilarity is defined to minimize the increase of internal variance.

After comprehensive analysis of multiple clustering techniques a customised Nearest Neighbour Chain (NNC) algorithm using Ward's method was selected which builds upon a preliminary study conducted in [7].

Moreover, this method can be generalized to any linear model of arbitrary order such as the proper motion model.

2.1.1 Zeroth-order model

In this model, we consider that the coordinates of the observations do not depend on time. Therefore, our interest is on agglomerate the obseravtions with coordinates values more similar.

In addition, using Ward's method, the objective is to agglomerate the clusters with the minimum increase in information loss, which is defined by the sum of squared residuals (SSR)

In this assumption, the dissimilarity measure is defined as follow,

Definition 2.2. *Let C_i and C_j be two disjoint clusters, the dissimilarity between them is*

$$\Delta(C_i, C_j) = R(C_i \cup C_j) - R(C_i) - R(C_j), \tag{2.1}$$

where $R(C)$ is the sum of squared residuals in the cluster $C \in \mathcal{C}$.

This dissimilarity is clearly symmetric but to show that it is non-negative definite is suitable to express it in terms of the coordinates of the observations.

Therefore, let n be the number of components of the observed data and we denote O as a observation and C as a cluster (set of observations). Thus, we consider an $n-$vector called observed coordinates $\boldsymbol{x}(O) = (x_1(O), \ldots, x_n(O)) \in \mathbb{R}^n$ which components are the observed data and the $n-$vector $\boldsymbol{x}(C) = (x_1(C), \ldots, x_n(C)) \in \mathbb{R}^n$ as the corresponding coordinates of the cluster center.

Accordingly, the sum of squared residuals is

$$R(C) = \sum_{O \in C} \sum_{k=1}^{n} w_k (x_k(O) - x_k(C))^2, \tag{2.2}$$

where w_k are the weight factors which permit to include some coordinates that are not space coordinates as, for example, the magnitude.

In terms of these weights, we can define a dot product and a norm as follow,

Definition 2.3. *Let $x_i \in \mathbb{R}$ $(i = 1, \ldots, n)$ be the components of the observed data and w_i $(i = 1, \ldots, n)$ their weight factors. The weighted matrix*

$$\boldsymbol{W} = \begin{pmatrix} w_1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & w_n \end{pmatrix} \tag{2.3}$$

defines a dot product by

$$\boldsymbol{x} \cdot \boldsymbol{y} = \boldsymbol{x}\boldsymbol{W}\boldsymbol{y}, \qquad \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n. \tag{2.4}$$

In addition, this product allows us to define a norm,

$$\|\boldsymbol{x}\| = \sqrt{\boldsymbol{x} \cdot \boldsymbol{x}}. \tag{2.5}$$

Note that the properties of a dot product and a norm are satisfied because the matrix $\boldsymbol{W}$ is diagonal.

Therefore, (2.2) can be written,

$$R(C) = \sum_{O \in C} \|\boldsymbol{x}(O) - \boldsymbol{x}(C)\|^2 . \tag{2.6}$$

In this expression, the coordinates of the cluster center, $\boldsymbol{x}(C)$, are chosen to minimize the SSR.

Since the coordinates are independent, the minimum of the SSR corresponds to the minimum of all components. It is thus clear that, for each coordinate, the center corresponds to the mean value,

$$x_k(C) = \frac{1}{n} \sum_{O \in C} x_k(O), \tag{2.7}$$

where n is the number of observations in the cluster C.

Therefore, the cluster center is given by

$$\boldsymbol{x}(C) = \frac{1}{n} \sum_{O \in C} \boldsymbol{x}(O). \tag{2.8}$$

Since our interest is in the clusters agglomeration, we also write the cluster center in terms of two disjoint clusters C_i and C_j such that $C = C_i \cup C_j$,

$$\boldsymbol{x}(C) = \frac{n_i \boldsymbol{x}(C_i) + n_j \boldsymbol{x}(C_j)}{n_i + n_j} \tag{2.9}$$

where n_i and n_j are the number of observations in the clusters C_i and C_j respectively.

Using these expressions of the coordinates, we rewrite the SSR as

$$\begin{aligned}
R(C) &= \sum_{O \in C} \|\boldsymbol{x}(O) - \boldsymbol{x}(C)\|^2 = && (2.10)\\
&= \sum_{O_i \in C_i} \|\boldsymbol{x}(O_i) - \boldsymbol{x}(C)\|^2 + \sum_{O_j \in C_j} \|\boldsymbol{x}(O_j) - \boldsymbol{x}(C)\|^2 = && (2.11)\\
&= \sum_{O_i \in C_i} \|\boldsymbol{x}(O_i) - \boldsymbol{x}(C_i) + \boldsymbol{x}(C_i) - \boldsymbol{x}(C)\|^2 + \\
&\quad + \sum_{O_j \in C_j} \|\boldsymbol{x}(O_j) - \boldsymbol{x}(C_j) + \boldsymbol{x}(C_j) - \boldsymbol{x}(C)\|^2 = && (2.12)\\
&= \sum_{O_i \in C_i} \|\boldsymbol{x}(O_i) - \boldsymbol{x}(C_i)\|^2 + 2 \sum_{O_i \in C_i} (\boldsymbol{x}(O_i) - \boldsymbol{x}(C_i)) \cdot (\boldsymbol{x}(C_i) - \boldsymbol{x}(C)) + \\
&\quad + \sum_{O_i \in C_i} \|\boldsymbol{x}(C_i) - \boldsymbol{x}(C)\|^2 + \sum_{O_j \in C_j} \|\boldsymbol{x}(O_j) - \boldsymbol{x}(C_j)\|^2 + \\
&\quad + 2 \sum_{O_j \in C_j} (\boldsymbol{x}(O_j) - \boldsymbol{x}(C_j)) \cdot (\boldsymbol{x}(C_j) - \boldsymbol{x}(C)) + \sum_{O_j \in C_j} \|\boldsymbol{x}(C_j) - \boldsymbol{x}(C)\|^2 . && (2.13)
\end{aligned}$$

Note that the second and fifth term of (2.13) are equal to zero due to

$$\sum_{O\in C}(\boldsymbol{x}(O)-\boldsymbol{x}(C))=\sum_{O\in C}\boldsymbol{x}(O)-n\boldsymbol{x}(C)=0. \tag{2.14}$$

Moreover, using the following expression,

$$\begin{aligned}\sum_{O_i\in C_i}\|\boldsymbol{x}(C_i)-\boldsymbol{x}(C)\|^2 &= \sum_{O_i\in C_i}\left\|\boldsymbol{x}(C_i)-\frac{n_i\boldsymbol{x}(C_i)+n_j\boldsymbol{x}(C_j)}{n_i+n_j}\right\|^2= &(2.15)\\ &= \frac{n_j^2}{(n_i+n_j)^2}\|\boldsymbol{x}(C_i)-\boldsymbol{x}(C_j)\|^2, &(2.16)\end{aligned}$$

the SSR can be expressed in terms of the clusters C_i and C_j,

$$\begin{aligned}R(C) &= R(C_i)+R(C_j)+\frac{n_j^2n_i}{(n_i+n_j)^2}\|\boldsymbol{x}(C_i)-\boldsymbol{x}(C_j)\|^2+\\ &\quad+\frac{n_i^2n_j}{(n_i+n_j)^2}\|\boldsymbol{x}(C_i)-\boldsymbol{x}(C_j)\|^2= &(2.17)\\ &= R(C_i)+R(C_j)+\frac{n_in_j}{n_i+n_j}\|\boldsymbol{x}(C_i)-\boldsymbol{x}(C_j)\|^2. &(2.18)\end{aligned}$$

Therefore, the dissimilarity is non-negative definite,

$$\Delta(C_i,C_j)=\frac{n_in_j}{n_i+n_j}\|\boldsymbol{x}(C_i)-\boldsymbol{x}(C_j)\|^2. \tag{2.19}$$

However, this is not a distance because the triangle inequality is not satisfied.

Moreover, it may happen that two disjoint clusters have dissimilarity zero and these are not equal since the cluster center could be the same.

This method, in the same way as other methods, satisfies the Lance and Williams' recurrence formula,

Proposition 2.4. *Let C_i, C_j and C_k be three disjoint clusters, then the dissimilarity between the cluster C_k and the cluster $C_i\cup C_j$ formed by the aglomeration of clusters C_i and C_j is*

$$\Delta(C_i\cup C_j,C_k)=a_i\Delta(C_i,C_k)+a_j\Delta(C_j,C_k)+b\Delta(C_i,C_j)+c|\Delta(C_i,C_k)-\Delta(C_j,C_k)|,$$

where the parameters a_i, a_j, b and c depend on the method used.

In Ward's method, the parameter values are

$$\begin{aligned}a_l &= \frac{n_l+n_k}{n_i+n_j+n_k},\quad l=i,j, &(2.20)\\ b &= -\frac{n_k}{n_i+n_j+n_k},\qquad c=0. &(2.21)\end{aligned}$$

Proof. On the one hand, using (2.9),

$$\begin{aligned}
\Delta(C_i \cup C_j, C_k) &= \frac{n_k(n_i+n_j)}{n_i+n_j+n_k} \|\boldsymbol{x}(C_i \cup C_j) - \boldsymbol{x}(C_k)\|^2 = && (2.22)\\
&= \frac{n_k n_i^2}{(n_i+n_j+n_k)(n_i+n_j)} \|\boldsymbol{x}(C_i)\|^2 + \\
&\quad + \frac{n_k n_j^2}{(n_i+n_j+n_k)(n_i+n_j)} \|\boldsymbol{x}(C_j)\|^2 + \\
&\quad + \frac{n_k}{n_i+n_j+n_k} \|\boldsymbol{x}(C_k)\|^2 - \\
&\quad - 2\frac{n_k n_i}{n_i+n_j+n_k} \boldsymbol{x}(C_i) \cdot \boldsymbol{x}(C_k) - \\
&\quad - 2\frac{n_k n_j}{n_i+n_j+n_k} \boldsymbol{x}(C_j) \cdot \boldsymbol{x}(C_k) + \\
&\quad + 2\frac{n_k n_i n_j}{(n_i+n_j+n_k)(n_i+n_j)} \boldsymbol{x}(C_i) \cdot \boldsymbol{x}(C_j). && (2.23)
\end{aligned}$$

On the other hand,

$$\begin{aligned}
\frac{n_i+n_k}{n_i+n_j+n_k}\Delta(C_i, C_k) &= \frac{n_i n_k}{n_i+n_j+n_k} \|\boldsymbol{x}(C_i) - \boldsymbol{x}(C_k)\|^2 = && (2.24)\\
&= \frac{n_k n_i}{n_i+n_j+n_k} \|\boldsymbol{x}(C_i)\|^2 + \\
&\quad + \frac{n_k n_i}{n_i+n_j+n_k} \|\boldsymbol{x}(C_k)\|^2 - \\
&\quad - 2\frac{n_k n_i}{n_i+n_j+n_k} \boldsymbol{x}(C_i) \cdot \boldsymbol{x}(C_k), && (2.25)\\
\frac{n_j+n_k}{n_i+n_j+n_k}\Delta(C_j, C_k) &= \frac{n_j n_k}{n_i+n_j+n_k} \|\boldsymbol{x}(C_j) - \boldsymbol{x}(C_k)\|^2 = && (2.26)\\
&= \frac{n_k n_j}{n_i+n_j+n_k} \|\boldsymbol{x}(C_j)\|^2 + \\
&\quad + \frac{n_k n_j}{n_i+n_j+n_k} \|\boldsymbol{x}(C_k)\|^2 - \\
&\quad - 2\frac{n_k n_j}{n_i+n_j+n_k} \boldsymbol{x}(C_j) \cdot \boldsymbol{x}(C_k), && (2.27)\\
-\frac{n_k}{n_i+n_j+n_k}\Delta(C_i, C_j) &= -\frac{n_i n_j n_k}{(n_i+n_j+n_k)(n_i+n_j)} \|\boldsymbol{x}(C_i) - \boldsymbol{x}(C_j)\|^2 = && (2.28)\\
&= -\frac{n_i n_j n_k}{(n_i+n_j+n_k)(n_i+n_j)} \|\boldsymbol{x}(C_i)\|^2 - \\
&\quad - \frac{n_i n_j n_k}{(n_i+n_j+n_k)(n_i+n_j)} \|\boldsymbol{x}(C_j)\|^2 + \\
&\quad + 2\frac{n_i n_j n_k}{(n_i+n_j+n_k)(n_i+n_j)} \boldsymbol{x}(C_i) \cdot \boldsymbol{x}(C_j). && (2.29)
\end{aligned}$$

Now, it is clear that the sum of terms in (2.25), (2.27) and (2.29) is equal to the six

terms in (2.23). Therefore, we get the equality,

$$\Delta(C_i \cup C_j, C_k) = \frac{n_i + n_k}{n_i + n_j + n_k}\Delta(C_i, C_k) + \frac{n_j + n_k}{n_i + n_j + n_k}\Delta(C_j, C_k) - \frac{n_k}{n_i + n_j + n_k}\Delta(C_i, C_j). \tag{2.30}$$

□

2.1.2 Proper motion model

So far, we have supposed that the stars are fixed but they are moving relative to the Sun, so we should include the proper motion in the cluster analysis.

The motion of a source on the sky may be described by the coordinates on the unit sphere in three-dimensional space or by a two-dimensional model in the tangent plane of the unit sphere since each cross-matching area covers only a very small part of the sky. Thus, without loss of generality, we consider a single independent component of the coordinates and for any other direction the approach would be the same.

Accordingly, let $u(t)$ be any of the coordinate functions. A source model is linear if $u(t) = \sum_k a_k f_k(t)$ where $f_k(t)$ has the time information and a_k are the source astrometric parameters.

Observe that this model allows us to include more parameters than the proper motion such as the parallax.

For the inclusion of the proper motion, we define the following linear model

$$u(t) = u_0 + u_1 t \tag{2.31}$$

where, in this case, the source parameters a_k are u_0, the mean position, and u_1, the proper motion. The time functions are 1 and t respectively.

Therefore, in our case, the linear system in matricial form is

$$\boldsymbol{b} = \boldsymbol{A}\boldsymbol{u} + \boldsymbol{e} \tag{2.32}$$

where $\boldsymbol{b} = (b_1, \ldots, b_n)^T$ is a $n-$vector of observations, $\boldsymbol{u} = (u_0, u_1)^T$ is a $2-$vector of the source parameters, $\boldsymbol{e} = (e_1, \ldots, e_n)^T$ is a $n-$vector of observation errors, and $\boldsymbol{A}$ is a $2 \times n-$matrix with the time functions,

$$\begin{pmatrix} 1 & t_1 \\ 1 & t_2 \\ \vdots & \vdots \\ 1 & t_n \end{pmatrix}.$$

The least-squares formalism

The least-squares method is used for the estimation of the source parameters in the linear regression model. The procedure is based on minimizing the sum of squared residuals,

$$R_u(C) = \sum_{i=1}^{n} e_i^2 = \boldsymbol{e}^T\boldsymbol{e} = (\boldsymbol{b} - \boldsymbol{A}\boldsymbol{u})^T(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{u}). \tag{2.33}$$

First of all, we write $R_u(C)$ as follow

$$\begin{aligned} R_u(C) &= \boldsymbol{b}^T\boldsymbol{b} - \boldsymbol{b}^T\boldsymbol{A}\boldsymbol{u} - \boldsymbol{u}^T\boldsymbol{A}^T\boldsymbol{b} + \boldsymbol{u}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{u} = & (2.34)\\ &= \boldsymbol{b}^T\boldsymbol{b} + \boldsymbol{u}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{u} - 2\boldsymbol{u}^T\boldsymbol{A}^T\boldsymbol{b}. & (2.35)\end{aligned}$$

Since this expression is in terms of matrices, we need some results of differentiation of scalar matrices.

Definition 2.5. *If $f(\boldsymbol{X})$ is a real function of an $m\times n-$matrix $\boldsymbol{X} = (x_{ij})$, then the partial differential of f with respect to X is defined as the $m \times n-$matrix of partial differentials $\partial f/\partial x_{ij}$:*

$$\frac{\partial f(\boldsymbol{X})}{\partial \boldsymbol{X}} = \begin{pmatrix} \frac{\partial f(\boldsymbol{X})}{\partial x_{11}} & \cdots & \frac{\partial f(\boldsymbol{X})}{\partial x_{1n}} \\ \vdots & & \vdots \\ \frac{\partial f(\boldsymbol{X})}{\partial x_{m1}} & \cdots & \frac{\partial f(\boldsymbol{X})}{\partial x_{mn}} \end{pmatrix} \tag{2.36}$$

Lemma 2.6. *Let $\boldsymbol{x}$ be an $n-$vector, $\boldsymbol{y}$ be an $m-$vector and $\boldsymbol{A}$ be an $n \times m-$matrix, then*

$$\frac{\partial}{\partial \boldsymbol{x}}\left(\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{y}\right) = \boldsymbol{A}\boldsymbol{y}.$$

Proof.

$$\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{y} = \sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij}x_iy_j, \tag{2.37}$$

$$\frac{\partial}{\partial x_k}\left(\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{y}\right) = \sum_{j=1}^{m} a_{kj}y_j = \boldsymbol{a_k}^T\boldsymbol{y}, \tag{2.38}$$

where $\boldsymbol{a_k}$ is the k^{th} row vector of $\boldsymbol{A}$.

According to the definition,

$$\frac{\partial}{\partial \boldsymbol{x}}\left(\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{y}\right) = \begin{pmatrix} \frac{\partial}{\partial x_1} \\ \vdots \\ \frac{\partial}{\partial x_n} \end{pmatrix} \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{y} = \begin{pmatrix} \boldsymbol{a_1}^T \\ \vdots \\ \boldsymbol{a_n}^T \end{pmatrix} \boldsymbol{y} = \boldsymbol{A}\boldsymbol{y}. \tag{2.39}$$

□

Lemma 2.7. *Let $\boldsymbol{x}$ be an $n-$vector, and let $\boldsymbol{A}$ be a symmetric $n \times n-$matrix, then*

$$\frac{\partial}{\partial \boldsymbol{x}} \left(\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{a} \right) = 2\boldsymbol{A}\boldsymbol{x}.$$

Proof.

$$\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{y} = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j, \tag{2.40}$$

$$\frac{\partial}{\partial x_k} \left(\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} \right) = \sum_{\substack{j=1 \\ (j \neq k)}}^{n} a_{kj} x_j + \sum_{\substack{i=1 \\ (i \neq k)}}^{n} a_{ik} x_i + 2a_{ii} x_i = 2 \sum_{j=1}^{n} a_{kj} x_j = 2\boldsymbol{a_k}^T \boldsymbol{x}, \tag{2.41}$$

where $\boldsymbol{a_k}$ is the k^{th} row vector of $\boldsymbol{A}$.

According to the definition,

$$\frac{\partial}{\partial \boldsymbol{x}} \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} = \begin{pmatrix} \frac{\partial}{\partial x_1} \\ \vdots \\ \frac{\partial}{\partial x_n} \end{pmatrix} \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} = 2 \begin{pmatrix} \boldsymbol{a_1}^T \\ \vdots \\ \boldsymbol{a_n}^T \end{pmatrix} \boldsymbol{x} = 2\boldsymbol{A}\boldsymbol{x}. \tag{2.42}$$

□

Using now the differentiation of matrices we obtain the following result.

Proposition 2.8. *The least-squares estimator satisfies the normal equations,*

$$\boldsymbol{A}^T \boldsymbol{A} \widehat{\boldsymbol{u}} = \boldsymbol{A}^T \boldsymbol{b}. \tag{2.43}$$

Proof.

$$\frac{\partial R_u(C)}{\partial \boldsymbol{u}} = 2\boldsymbol{A}^T \boldsymbol{A} \boldsymbol{u} - 2\boldsymbol{A}^T \boldsymbol{b}, \tag{2.44}$$

$$\frac{\partial^2 R_u(C)}{\partial \boldsymbol{u}^2} = 2\boldsymbol{A}^T \boldsymbol{A}. \tag{2.45}$$

A condition for minimisation is that the first derivative must be equal to zero, which gives the following equation (the so-called normal equation)

$$\boldsymbol{A}^T \boldsymbol{A} \widehat{\boldsymbol{u}} = \boldsymbol{A}^T \boldsymbol{b}. \tag{2.46}$$

□

Note that if there is only one observation in the cluster, this method is not necessary, so we only be taken into account the other cases.

Lemma 2.9. *If there is more than one observation in the cluster C ($n \geq 2$), then $\boldsymbol{A}^T\boldsymbol{A}$ is positive definite and nonsingular.*

Proof. It suffices to show that $\text{rank}\boldsymbol{A} = \text{rank}\boldsymbol{A}^T\boldsymbol{A}$ because $\text{rank}\boldsymbol{A} = 2$.

Thus,

$$\boldsymbol{x} \in \ker\boldsymbol{A} \Rightarrow \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0} \Rightarrow \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0} \Rightarrow \boldsymbol{x} \in \ker\boldsymbol{A}^T\boldsymbol{A},$$

$$\boldsymbol{x} \in \ker\boldsymbol{A}^T\boldsymbol{A} \Rightarrow \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0} \Rightarrow \boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} = (\boldsymbol{A}\boldsymbol{x})^T(\boldsymbol{A}\boldsymbol{x}) = \boldsymbol{0} \Rightarrow \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0} \Rightarrow \boldsymbol{x} \in \ker\boldsymbol{A}.$$

This gives $\ker\boldsymbol{A} = \ker\boldsymbol{A}^T\boldsymbol{A}$ and, of course, this implies $\text{rank}\boldsymbol{A} = \text{rank}\boldsymbol{A}^T\boldsymbol{A}$, which is what had to be shown.

In addition, $\boldsymbol{A}^T\boldsymbol{A}$ is positive definite since $\boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} = (\boldsymbol{A}\boldsymbol{x})^T(\boldsymbol{A}\boldsymbol{x}) > 0$.

□

Corollary 2.10. *The normal equation* (2.46) *has an unique solution which minimizes the sum of squared residuals,*

$$\widehat{\boldsymbol{u}} = \boldsymbol{N}^{-1}\boldsymbol{h}, \tag{2.47}$$

where $\boldsymbol{N} = \boldsymbol{A}^T\boldsymbol{A}$ is the normal matrix and $\boldsymbol{h} = \boldsymbol{A}^T\boldsymbol{b}$.

Proof. Since $\boldsymbol{A}^T\boldsymbol{A}$ is positive definite, the second-order condition is satisfied. In addition, the inverse matrix exists, so we can isolate the estimator $\widehat{\boldsymbol{u}}$. □

Finally, the minimum of the sum of squared residuals can be written as,

$$R_u(C) = \boldsymbol{b}^T\boldsymbol{b} + \widehat{\boldsymbol{u}}^T\boldsymbol{N}\widehat{\boldsymbol{u}} - 2\widehat{\boldsymbol{u}}^T\boldsymbol{h} = \boldsymbol{b}^T\boldsymbol{b} + \widehat{\boldsymbol{u}}^T\boldsymbol{N}\widehat{\boldsymbol{u}} - 2\widehat{\boldsymbol{u}}^T\boldsymbol{N}\widehat{\boldsymbol{u}} = \boldsymbol{b}^T\boldsymbol{b} - \widehat{\boldsymbol{u}}^T\boldsymbol{N}\widehat{\boldsymbol{u}}. \tag{2.48}$$

Clusters agglomeration

Our interest is to agglomerate two different clusters. Therefore, we consider hereafter the terms in (2.32) according to the terms of the two disjoint clusters C_i and C_j,

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A_i} \\ \boldsymbol{A_j} \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} \boldsymbol{b_i} \\ \boldsymbol{b_j} \end{pmatrix}, \quad \boldsymbol{e} = \begin{pmatrix} \boldsymbol{e_i} \\ \boldsymbol{e_j} \end{pmatrix}. \tag{2.49}$$

In consequence, the normal matrix for the agglomerated cluster is

$$\boldsymbol{N} = \boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{N}_i + \boldsymbol{N}_j, \tag{2.50}$$

and therefore, we rewrite the normal equation (2.46) as follow

$$\boldsymbol{N}\widehat{\boldsymbol{u}} = (\boldsymbol{N}_i + \boldsymbol{N}_j)\widehat{\boldsymbol{u}} = \boldsymbol{h}_i + \boldsymbol{h}_j = \boldsymbol{N}_i\widehat{\boldsymbol{u}}_i + \boldsymbol{N}_j\widehat{\boldsymbol{u}}_j. \tag{2.51}$$

This gives the solution

$$\widehat{\boldsymbol{u}} = (\boldsymbol{N}_i + \boldsymbol{N}_j)^{-1}(\boldsymbol{N}_i\widehat{\boldsymbol{u}}_i + \boldsymbol{N}_j\widehat{\boldsymbol{u}}_j), \tag{2.52}$$

and its SSR is

$$R_u(C) = \boldsymbol{b}_i^T \boldsymbol{b}_i + \boldsymbol{b}_j^T \boldsymbol{b}_j - \widehat{\boldsymbol{u}}^T \boldsymbol{N} \widehat{\boldsymbol{u}}. \tag{2.53}$$

In our case, we define a dissimilarity measure in $u-$direction similar as Definition 2.2.

Definition 2.11. *Let C_i and C_j be two disjoint clusters, the dissimilarity in $u-$direction between them is*

$$\Delta_u(C_i, C_j) = R_u(C_i \cup C_j) - R_u(C_i) - R_u(C_j). \tag{2.54}$$

From this definition, it is clear that the dissimilarity is the penalty in the SSR when agglomerating two clusters. Moreover, if there is only one observation in cluster C_i or C_j, its SSR is equal to zero and it simplifies the expression (2.54). Therefore,

$$\begin{aligned}
\Delta_u(C_i, C_j) &= R_u(C_i \cup C_j) - R_u(C_i) - R_u(C_j) = && (2.55)\\
&= \widehat{\boldsymbol{u}}_i^T \boldsymbol{N}_i \widehat{\boldsymbol{u}}_i + \widehat{\boldsymbol{u}}_j^T \boldsymbol{N}_j \widehat{\boldsymbol{u}}_j - \widehat{\boldsymbol{u}}^T \boldsymbol{N} \widehat{\boldsymbol{u}} = && (2.56)\\
&= \widehat{\boldsymbol{u}}_i^T \boldsymbol{N}_i \widehat{\boldsymbol{u}}_i + \widehat{\boldsymbol{u}}_j^T \boldsymbol{N}_j \widehat{\boldsymbol{u}}_j - (\widehat{\boldsymbol{u}}_i^T \boldsymbol{N}_i + \widehat{\boldsymbol{u}}_j^T \boldsymbol{N}_j) \boldsymbol{N}^{-1} (\boldsymbol{N}_i \widehat{\boldsymbol{u}}_i + \boldsymbol{N}_j \widehat{\boldsymbol{u}}_j) = && (2.57)\\
&= \widehat{\boldsymbol{u}}_i^T (\boldsymbol{N}_i - \boldsymbol{N}_i \boldsymbol{N}^{-1} \boldsymbol{N}_i) \widehat{\boldsymbol{u}}_i - \widehat{\boldsymbol{u}}_i^T \boldsymbol{N}_i \boldsymbol{N}^{-1} \boldsymbol{N}_j \widehat{\boldsymbol{u}}_j - \\
&\quad - \widehat{\boldsymbol{u}}_j^T \boldsymbol{N}_j \boldsymbol{N}^{-1} \boldsymbol{N}_i \widehat{\boldsymbol{u}}_i + \widehat{\boldsymbol{u}}_j^T (\boldsymbol{N}_j - \boldsymbol{N}_j \boldsymbol{N}^{-1} \boldsymbol{N}_j) \widehat{\boldsymbol{u}}_j = && (2.58)\\
&= \widehat{\boldsymbol{u}}_i^T \boldsymbol{N}_i \boldsymbol{N}^{-1} \boldsymbol{N}_j (\widehat{\boldsymbol{u}}_i - \widehat{\boldsymbol{u}}_j) - \widehat{\boldsymbol{u}}_j^T \boldsymbol{N}_j \boldsymbol{N}^{-1} \boldsymbol{N}_i (\widehat{\boldsymbol{u}}_i - \widehat{\boldsymbol{u}}_j). && (2.59)
\end{aligned}$$

where from (2.58) to (2.59) the following expression has been used,

$$\boldsymbol{N}_i - \boldsymbol{N}_i \boldsymbol{N}^{-1} \boldsymbol{N}_i = \boldsymbol{N}_i \boldsymbol{N}^{-1} (\boldsymbol{N} - \boldsymbol{N}_i) = \boldsymbol{N}_i \boldsymbol{N}^{-1} \boldsymbol{N}_j. \tag{2.60}$$

From this, using

$$\begin{aligned}
\boldsymbol{N}_i \boldsymbol{N}^{-1} \boldsymbol{N}_j &= \boldsymbol{N}_i \boldsymbol{N}^{-1} (\boldsymbol{N} - \boldsymbol{N}_j) = \boldsymbol{N}_i - \boldsymbol{N}_i \boldsymbol{N}^{-1} \boldsymbol{N}_i = && (2.61)\\
&= \boldsymbol{N}_i - (\boldsymbol{N} - \boldsymbol{N}_j) \boldsymbol{N}^{-1} \boldsymbol{N}_i = \boldsymbol{N}_j \boldsymbol{N}^{-1} \boldsymbol{N}_i, && (2.62)
\end{aligned}$$

the dissimilarity can be expressed as

$$\Delta_u(C_i, C_j) = (\widehat{\boldsymbol{u}}_i - \widehat{\boldsymbol{u}}_j)^T \boldsymbol{N}_i (\boldsymbol{N}_i + \boldsymbol{N}_j)^{-1} \boldsymbol{N}_j (\widehat{\boldsymbol{u}}_i - \widehat{\boldsymbol{u}}_j). \tag{2.63}$$

From this expression, the two conditions of the definition 2.1 are verified using some results of positive definite matrices.

Lemma 2.12. *A symmetric real matrix $\boldsymbol{A}$ is positive definite if and only if all the eigenvalues are positive.*

Proof. Suppose $\boldsymbol{x} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}$ is an eigenvector of $\boldsymbol{A}$ with corresponding eigenvalue $\lambda \in \mathbb{R}$ such that $\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{x}$. Then,

$$\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} = \lambda \boldsymbol{x}^T \boldsymbol{x} > 0 \Rightarrow \lambda > 0. \tag{2.64}$$

Hence, all the eigenvalues are positive.

Reciprocally, since $\boldsymbol{A}$ is symmetric, there exists an orthogonal matrix $\boldsymbol{Q}$ such that $\boldsymbol{A} = \boldsymbol{Q}^T\boldsymbol{\Lambda}\boldsymbol{Q}$ where $\boldsymbol{\Lambda}$ is a diagonal matrix with the eigenvalues. Therefore, for all $\boldsymbol{x} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}$,

$$\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{x}^T\boldsymbol{Q}^T\boldsymbol{\Lambda}\boldsymbol{Q}\boldsymbol{x} = (\boldsymbol{Q}\boldsymbol{x})^T\boldsymbol{\Lambda}(\boldsymbol{Q}\boldsymbol{x}) > 0. \tag{2.65}$$

□

Lemma 2.13. *Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be two real $n \times n$ positive definite matrices. The following properties hold:*

(i) Every positive definite matrix is invertible and its inverse is also positive definite.

(ii) There exists a unique positive definite matrix $\boldsymbol{A}^{1/2}$ such that $\left(\boldsymbol{A}^{1/2}\right)^2 = \boldsymbol{A}$.

(iii) The sum $\boldsymbol{A} + \boldsymbol{B}$ is a positive definite matrix.

(iv) The product $\boldsymbol{A}\boldsymbol{B}\boldsymbol{A}$ is a positive definite matrix.

(v) All the eigenvalues of $\boldsymbol{A}\boldsymbol{B}$ are positive.

(vi) Let $\boldsymbol{C}$ be an other real $n \times n$ positive definite matrix, all the eigenvalues of $\boldsymbol{A}\boldsymbol{B}\boldsymbol{C}$ are positive.

Proof. (i) Since $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} > 0$ for all $\boldsymbol{x} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}$, then $\boldsymbol{A}\boldsymbol{x} \neq \boldsymbol{0}$ for all $\boldsymbol{x} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}$. Hence, $\ker A = \{\boldsymbol{0}\}$ and the matrix A is invertible.

In addition, if we define $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$, then

$$\boldsymbol{y}^T\boldsymbol{A}^{-1}\boldsymbol{y} = \boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{A}^{-1}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} > 0, \quad \forall \boldsymbol{y} \in \mathbb{R}^n. \tag{2.66}$$

(ii) Consider the decomposition $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T$, where $\boldsymbol{\Lambda}$ is a diagonal matrix (having the eigenvalues of $\boldsymbol{A}$ on the diagonal) and $\boldsymbol{Q}$ is an orthogonal matrix.

The positive definite square root of $\boldsymbol{A}$ is the matrix $\boldsymbol{A}^{1/2} = \boldsymbol{Q}\boldsymbol{\Lambda}^{1/2}\boldsymbol{Q}^T$ where $\boldsymbol{\Lambda}^{1/2}$ is the square root of the diagonal matrix (i.e. having the positive square root of the eigenvalues on the diagonal). Indeed,

$$\boldsymbol{A}^{1/2}\boldsymbol{A}^{1/2} = \left(\boldsymbol{Q}\boldsymbol{\Lambda}^{1/2}\boldsymbol{Q}^T\right)\left(\boldsymbol{Q}\boldsymbol{\Lambda}^{1/2}\boldsymbol{Q}^T\right) = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T = \boldsymbol{A}. \tag{2.67}$$

The uniqueness comes from the uniqueness of the positive square root of the eigenvalues.

(iii) The sum is symmetric since $(\boldsymbol{A}+\boldsymbol{B})^T = \boldsymbol{A}^T + \boldsymbol{B}^T = \boldsymbol{A}+\boldsymbol{B}$. In addition,

$$\boldsymbol{x}^T(\boldsymbol{A}+\boldsymbol{B})\boldsymbol{x} = \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{B}\boldsymbol{x} > 0, \quad \forall \boldsymbol{x} \in \mathbb{R}^n. \tag{2.68}$$

(iv) The product $\boldsymbol{ABA}$ is symmetric, $(\boldsymbol{ABA})^T = \boldsymbol{A}^T\boldsymbol{B}^T\boldsymbol{A}^T = \boldsymbol{ABA}$ and

$$\boldsymbol{x}^T\boldsymbol{ABA}\boldsymbol{x} = (\boldsymbol{A}\boldsymbol{x})^T\boldsymbol{B}(\boldsymbol{A}\boldsymbol{x}) > 0, \quad \forall \boldsymbol{x} \in \mathbb{R}^n. \tag{2.69}$$

(v) Suppose $\boldsymbol{AB}$ has eigenvalue λ, thus there exists a eigenvector $\boldsymbol{x} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}$ such that $\boldsymbol{ABx} = \lambda\boldsymbol{x}$.

Then, $\boldsymbol{x}^T\boldsymbol{BABx} = \lambda\boldsymbol{x}^T\boldsymbol{Bx}$, and, using (iv), the product $\boldsymbol{BAB}$ is positive definite. Hence, aplying Lemma 2.12, $\lambda = \frac{\boldsymbol{x}^T\boldsymbol{BABx}}{\boldsymbol{x}^T\boldsymbol{Bx}} > 0$.

(vi) Let $\bar{\boldsymbol{A}} = \boldsymbol{B}^{\frac{1}{2}}\boldsymbol{A}\boldsymbol{B}^{\frac{1}{2}}$ and $\bar{\boldsymbol{C}} = \boldsymbol{B}^{\frac{1}{2}}\boldsymbol{C}\boldsymbol{B}^{\frac{1}{2}}$, which are positive definite matrices by (iv). Moreover, $\bar{\boldsymbol{A}}\bar{\boldsymbol{C}} = \boldsymbol{B}^{\frac{1}{2}}\boldsymbol{ABCB}^{\frac{1}{2}}$. Thus, by (v), $\boldsymbol{B}^{\frac{1}{2}}\boldsymbol{ABCB}^{\frac{1}{2}}$ has positive eigenvalues and, in consequence, $\boldsymbol{ABC}$ as well.

□

Corollary 2.14. *The dissimilarity Δ_u is non-negative.*

Proof. The normal matrices in (2.63) are positive definite as we had shown in Lemma 2.9. Therefore, using the properties in Lemma 2.13, all the eigenvalues of the matrix $\boldsymbol{N}_i(\boldsymbol{N}_i+\boldsymbol{N}_j)^{-1}\boldsymbol{N}_j$ are positive. Therefore, since this matrix is symmetric (from (2.61) and (2.62)) and using Lemma 2.12, it is also a positive definite matix.

Hence, $\Delta_u(C_i, C_j) \geq 0$ for all $C_i, C_j \in \mathcal{C}$. □

Note that this dissimilarity does not satisfy the Lance and Williams' recurrence formula. A similar proof of Proposition 2.4 gets a relation where the parameters depends on the determinants. Nevertheless, since the sum of determinants is not the determinant of the sum, then there appear some extra terms.

Prior information

We wish to add prior knowledge of proper motion corresponding to an observation with proper motion equal to zero with a fairly large error (i.e, $u_1 = 0 + \sigma_1$). If we multiply the last equation by a parameter L, it can be expressed as $Lu_1 = 0 + L\sigma_1$. Thus, we can interpret this prior information as an additional observation and we rewrite the terms in (2.32) as

$$\boldsymbol{b}^* = \begin{pmatrix} 0 \\ b_1 \\ \vdots \\ b_n \end{pmatrix}, \qquad \boldsymbol{A}^* = \begin{pmatrix} 0 & L \\ 1 & t_1 \\ \vdots & \vdots \\ 1 & t_n \end{pmatrix}.$$

The parameter L must have dimension of time and it corresponds to an error, thus the proposal is to compute it as $L = \sigma_u/\sigma_1$, where σ_u is the precision of an observation.

Note that the prior allows to analyze clusters with only one observation because $\text{rank}\boldsymbol{A^*} = 2$ in all cases.

The resolution using the least-squares formalism must be the same as explained above, so the solution will be such that (2.47):

$$\widehat{\boldsymbol{u}}^* = (\boldsymbol{N}^*)^{-1}\boldsymbol{h}^*, \tag{2.70}$$

where $\boldsymbol{N^*} = (\boldsymbol{A^*})^T\boldsymbol{A^*}$ is the normal matrix and $\boldsymbol{h^*} = (\boldsymbol{A^*})^T\boldsymbol{b^*}$.

In addition, we can express the terms in (2.70) as a function of the terms without prior information as

$$\boldsymbol{N^*} = \boldsymbol{N} + \boldsymbol{\Lambda}, \quad \boldsymbol{h^*} = \boldsymbol{h}. \tag{2.71}$$

where $\boldsymbol{\Lambda}$ is the prior information matrix,

$$\boldsymbol{\Lambda} = \begin{pmatrix} 0 & 0 \\ 0 & L^2 \end{pmatrix}.$$

It is also important to mention that when we agglomerate two clusters, we construct a cluster with the observations of the two clusters and the prior information (but only once). Therefore,

$$\boldsymbol{A^*} = \begin{pmatrix} 0 & L \\ \boldsymbol{A_i} & \\ \boldsymbol{A_j} & \end{pmatrix}, \quad \boldsymbol{b^*} = \begin{pmatrix} 0 \\ \boldsymbol{b_i} \\ \boldsymbol{b_j} \end{pmatrix}, \quad \boldsymbol{e^*} = \begin{pmatrix} L\sigma_1 \\ \boldsymbol{e_i} \\ \boldsymbol{e_j} \end{pmatrix}. \tag{2.72}$$

In the same way as (2.50), the normal matrix for the agglomerated cluster is

$$\boldsymbol{N^*} = (\boldsymbol{A^*})^T\boldsymbol{A^*} = \boldsymbol{N} + \boldsymbol{\Lambda} = \boldsymbol{N_i} + \boldsymbol{N_j} + \boldsymbol{\Lambda} = \boldsymbol{N_i^*} + \boldsymbol{N_j^*} - \boldsymbol{\Lambda}, \tag{2.73}$$

and the solution is

$$\widehat{\boldsymbol{u}}^* = (\boldsymbol{N_i^*} + \boldsymbol{N_j^*} - \boldsymbol{\Lambda})^{-1}(\boldsymbol{N_i^*}\widehat{\boldsymbol{u}}_i^* + \boldsymbol{N_j^*}\widehat{\boldsymbol{u}}_j^*). \tag{2.74}$$

From now on, we will omit the symbol * and, unless otherwise indicated, we will consider the case with the prior knowledge.

About the dissimilarity, we may observe that the agglomerated cluster is not really the agglomeration of the clusters C_i and C_j because we only have prior information once. Thus, the dissimilarity will not be strictly the penalty in the SSR and we have to add new terms related to the prior information.

Definition 2.15. *Let C_i and C_j be two disjoint clusters, the dissimilarity including the prior in $u-$direction between them is*

$$\begin{aligned} \Delta_u^{(L)}(C_i, C_j) \;=\; & R_u(C_i \cup C_j) - R_u(C_i) - R_u(C_j) + \boldsymbol{u}_i^T\boldsymbol{N}_i\boldsymbol{N}^{-1}\boldsymbol{\Lambda}\boldsymbol{u}_i + \\ & + \boldsymbol{u}_j^T\boldsymbol{N}_j\boldsymbol{N}^{-1}\boldsymbol{\Lambda}\boldsymbol{u}_j + \widehat{\boldsymbol{u}}_j^T(\boldsymbol{N}_i\boldsymbol{N}^{-1}\boldsymbol{N}_j - \boldsymbol{N}_j\boldsymbol{N}^{-1}\boldsymbol{N}_i)\widehat{\boldsymbol{u}}_i. \end{aligned} \tag{2.75}$$

From this definition, the dissimilarity is the penalty in the SSR plus three terms caused by the prior.

More specifically, the first term is interpreted as the correction in $R_u(C_i)$, the second corresponds to the correction in $R_u(C_j)$, and the third is the cross correlation in the agglomeration.

In addition, if we develop this expression, we will recover the same expression as (2.63),

$$\begin{aligned}
\Delta_u^{(L)}(C_i,C_j) &= R_u(C_i\cup C_j)-R_u(C_i)-R_u(C_j)+\boldsymbol{u}_i^T\boldsymbol{N}_i\boldsymbol{N}^{-1}\boldsymbol{\Lambda}\boldsymbol{u}_i+ \\
&\quad +\boldsymbol{u}_j^T\boldsymbol{N}_j\boldsymbol{N}^{-1}\boldsymbol{\Lambda}\boldsymbol{u}_j+\widehat{\boldsymbol{u}}_j^T(\boldsymbol{N}_i\boldsymbol{N}^{-1}\boldsymbol{N}_j-\boldsymbol{N}_j\boldsymbol{N}^{-1}\boldsymbol{N}_i)\widehat{\boldsymbol{u}}_i= && (2.76)\\
&= \widehat{\boldsymbol{u}}_i^T\boldsymbol{N}_i\widehat{\boldsymbol{u}}_i+\widehat{\boldsymbol{u}}_j^T\boldsymbol{N}_j\widehat{\boldsymbol{u}}_j-\widehat{\boldsymbol{u}}^T\boldsymbol{N}\widehat{\boldsymbol{u}}+\boldsymbol{u}_i^T\boldsymbol{N}_i\boldsymbol{N}^{-1}\boldsymbol{\Lambda}\boldsymbol{u}_i+ \\
&\quad +\boldsymbol{u}_j^T\boldsymbol{\Lambda}\boldsymbol{N}^{-1}\boldsymbol{N}_j\boldsymbol{u}_j+\widehat{\boldsymbol{u}}_j^T(\boldsymbol{N}_i\boldsymbol{N}^{-1}\boldsymbol{N}_j-\boldsymbol{N}_j\boldsymbol{N}^{-1}\boldsymbol{N}_i)\widehat{\boldsymbol{u}}_i= && (2.77)\\
&= \widehat{\boldsymbol{u}}_i^T\boldsymbol{N}_i\widehat{\boldsymbol{u}}_i+\widehat{\boldsymbol{u}}_j^T\boldsymbol{N}_j\widehat{\boldsymbol{u}}_j+\boldsymbol{u}_i^T\boldsymbol{N}_i\boldsymbol{N}^{-1}\boldsymbol{\Lambda}\boldsymbol{u}_i+ \\
&\quad +\boldsymbol{u}_j^T\boldsymbol{N}_j\boldsymbol{N}^{-1}\boldsymbol{\Lambda}\boldsymbol{u}_j-(\widehat{\boldsymbol{u}}_i^T\boldsymbol{N}_i+\widehat{\boldsymbol{u}}_j^T\boldsymbol{N}_j)\boldsymbol{N}^{-1}(\boldsymbol{N}_i\widehat{\boldsymbol{u}}_i+\boldsymbol{N}_j\widehat{\boldsymbol{u}}_j)+ \\
&\quad +\widehat{\boldsymbol{u}}_j^T(\boldsymbol{N}_i\boldsymbol{N}^{-1}\boldsymbol{N}_j-\boldsymbol{N}_j\boldsymbol{N}^{-1}\boldsymbol{N}_i)\widehat{\boldsymbol{u}}_i= && (2.78)\\
&= \widehat{\boldsymbol{u}}_i^T(\boldsymbol{N}_i-\boldsymbol{N}_i\boldsymbol{N}^{-1}\boldsymbol{N}_i+\boldsymbol{N}_i\boldsymbol{N}^{-1}\boldsymbol{\Lambda})\widehat{\boldsymbol{u}}_i-\widehat{\boldsymbol{u}}_i^T\boldsymbol{N}_i\boldsymbol{N}^{-1}\boldsymbol{N}_j\widehat{\boldsymbol{u}}_j- \\
&\quad -\widehat{\boldsymbol{u}}_j^T\boldsymbol{N}_i\boldsymbol{N}^{-1}\boldsymbol{N}_j\widehat{\boldsymbol{u}}_i+\widehat{\boldsymbol{u}}_j^T(\boldsymbol{N}_j-\boldsymbol{N}_j\boldsymbol{N}^{-1}\boldsymbol{N}_j+\boldsymbol{N}_j\boldsymbol{N}^{-1}\boldsymbol{\Lambda})\widehat{\boldsymbol{u}}_j= && (2.79)\\
&= \widehat{\boldsymbol{u}}_i^T\boldsymbol{N}_i\boldsymbol{N}^{-1}\boldsymbol{N}_j(\widehat{\boldsymbol{u}}_i-\widehat{\boldsymbol{u}}_j)-\widehat{\boldsymbol{u}}_j^T\boldsymbol{N}_i\boldsymbol{N}^{-1}\boldsymbol{N}_j(\widehat{\boldsymbol{u}}_i-\widehat{\boldsymbol{u}}_j). && (2.80)
\end{aligned}$$

where from (2.76) to (2.77) the following expression has been used,

$$\boldsymbol{u}_j^T\boldsymbol{N}_j\boldsymbol{N}^{-1}\boldsymbol{\Lambda}\boldsymbol{u}_j=(\boldsymbol{u}_j^T\boldsymbol{N}_j\boldsymbol{N}^{-1}\boldsymbol{\Lambda}\boldsymbol{u}_j)^T=\boldsymbol{u}_j^T\boldsymbol{\Lambda}\boldsymbol{N}^{-1}\boldsymbol{N}_j\boldsymbol{u}_j. \tag{2.81}$$

In addition, from (2.79) to (2.80),

$$\begin{aligned}
\boldsymbol{N}_i-\boldsymbol{N}_i\boldsymbol{N}^{-1}\boldsymbol{N}_i+\boldsymbol{N}_i\boldsymbol{N}^{-1}\boldsymbol{\Lambda} &= \boldsymbol{N}_i\boldsymbol{N}^{-1}(\boldsymbol{N}-\boldsymbol{N}_i+\boldsymbol{\Lambda})=\boldsymbol{N}_i\boldsymbol{N}^{-1}\boldsymbol{N}_j, && (2.82)\\
\boldsymbol{N}_j-\boldsymbol{N}_j\boldsymbol{N}^{-1}\boldsymbol{N}_j+\boldsymbol{\Lambda}\boldsymbol{N}^{-1}\boldsymbol{N}_j &= (\boldsymbol{N}-\boldsymbol{N}_j+\boldsymbol{\Lambda})\boldsymbol{N}^{-1}\boldsymbol{N}_j=\boldsymbol{N}_i\boldsymbol{N}^{-1}\boldsymbol{N}_j. && (2.83)
\end{aligned}$$

And finally, dissimilarity can be expressed as

$$\Delta_u^{(L)}(C_i,C_j)=(\widehat{\boldsymbol{u}}_i-\widehat{\boldsymbol{u}}_j)^T\boldsymbol{N}_i(\boldsymbol{N}_i+\boldsymbol{N}_j-\boldsymbol{\Lambda})^{-1}\boldsymbol{N}_j(\widehat{\boldsymbol{u}}_i-\widehat{\boldsymbol{u}}_j). \tag{2.84}$$

Therefore, we have defined a new dissimilarity, adapted to the new agglomeration form, in complete analogy with (2.54).

Moreover, it also introduces a small variation in the SSR,

$$\begin{aligned}
R_{C_i\cup C_j}(\widehat{\boldsymbol{u}}) &= R_u(C_i)+R_u(C_j)-\Delta_u^{(L)}(C_i,C_j)-\boldsymbol{u}_i^T\boldsymbol{N}_i\boldsymbol{N}^{-1}\boldsymbol{\Lambda}\boldsymbol{u}_i- \\
&\quad -\boldsymbol{u}_j^T\boldsymbol{N}_j\boldsymbol{N}^{-1}\boldsymbol{\Lambda}\boldsymbol{u}_j-\widehat{\boldsymbol{u}}_j^T(\boldsymbol{N}_i\boldsymbol{N}^{-1}\boldsymbol{N}_j-\boldsymbol{N}_j\boldsymbol{N}^{-1}\boldsymbol{N}_i)\widehat{\boldsymbol{u}}_i. && (2.85)
\end{aligned}$$

2.1.3 Magnitude criterion

The magnitude should be taken into account because in crowded areas may be ambiguity on the proximity in the position coordinates, thus the cross-match could significantly improve by adding this parameter.

As we have seen above, the norm from Definition 2.3 includes weight factors. If we only use the position coordinates, these weights are equal and the norm will be Euclidean. But since the inclusion of the magnitude requires a factor to make a magnitude error comparable with an error in position, we have to introduce a weight factor.

Hence, we consider the weight factor $w_m = \frac{\sigma_{pos}^2}{\sigma_m^2}$.

These errors may change according to the source magnitude, but since our input is only the observed data, the weight factor has to be initialized on $w_m = 0.36\frac{arcsec}{mag}$, which is calibrated properly using the assumptions of $\sigma_{pos} \sim 0.3 arcsec$ and $\sigma_m \sim 0.5 mag$. These values are the 90^{th} percentile from the source match diagnostic obtained from scientific validation and statistical analysis performed on operational Gaia data.

Despite that, the inclusion of the magnitude can create several clusters in the same position for variable stars. The first solution that comes to mind is to apply a linear model as we do for the inclusion of the proper motion but the magnitudes may change in an unpredictable way whereas the position of a source changes slowly and predictably. Therefore such cases should be detected and corrected in a post-processing.

2.1.4 Generalized source model

The zeroth-order model analyzed above in Section 2.1.1 includes all the coordinates but the proper motion model (Section 2.1.2) only includes a single independent coordinate. Therefore, using the same argument as the zeroth-order model, the SSR is

$$R(C) = \sum_{k=1}^{n} w_k R_k(C). \tag{2.86}$$

where $R_k(C)$ is the SSR according to the k-coordinate model.

Since the coordinates are independent, the minimum of the SSR corresponds to the minimum of all components. In consequence, the total dissimilarity is the sum of the dissimilarities in each coordinate.

Definition 2.16. *Let C_i and C_j be two disjoint clusters, the global dissimilarity between them is*

$$\Delta(C_i, C_j) = \sum_{k=1}^{n} w_k \Delta_k(C_i, C_j). \tag{2.87}$$

The proper motion model in the cluster analysis contains a problem in the computation of the dissimilarity between two observations because the dissimilarity is always zero and therefore it will be possible to match any two observations perfectly.

Hence, we have discussed the specification of a prior knowledge which could solve this problem. Despite that, the inclusion of the prior introduces an error (especially in high proper motion sources) caused by the assumption $u_1 = 0$ and the definition of the dissimilarity is more complicated due to prevent the weight of the prior information to increase during agglomeration.

Note that, using the prior information method, initially the observations with less dissimilarity are the closest ones. Therefore, applying this reasoning, the solution proposed consists to use the zeroth-order model for the agglomeration of clusters with few observations and, when the estimation of the proper motion is good enough, to use the proper motion model in the position coordinates.

In our case, the used coordinates are the right ascension α and the declination δ.

Therefore, the global dissimilarity that we use is the following:

Definition 2.17. *Let C_i and C_j be two disjoint clusters, the global dissimilarity between them is*

$$\Delta(C_i, C_j) = \begin{cases} \frac{n_i n_j}{n_i + n_j} \left\| \boldsymbol{x}(C_i) - \boldsymbol{x}(C_j) \right\|^2, & n_i + n_j \leq 3, \\ \Delta_\alpha(C_i, C_j) + \Delta_\delta(C_i, C_j) + w_m \frac{n_i n_j}{n_i + n_j} \left(m(C_i) - m(C_j) \right)^2, & n_i + n_j > 3, \end{cases} \tag{2.88}$$

where Δ_α and Δ_δ are considered using the proper motion model.

The reason that we consider the zeroth-order model until the agglomeration of 3 observations is due to that it has to be a number greater than 2 but if it is too great, the agglomeration does not occur for high proper motion stars. Therefore, the optimal value is considered to be 3.

2.2 Nearest Neighbor criteria

We consider the nearest neighbor criteria for cluster analysis because the nearest neighbor chain (NNC) algorithm is the preferred for Ward's dissimilarity (see [8]) and, most importantly, the modified NNC algorithm for IDU-XM task in previous versions is the most suitable algorithm for the IDU-XM task (see [3]).

Furthermore, other modifications have to be introduced to implement the algorithm with the source model purposed in Section 2.1.

2.2.1 Nearest neighbor chain

Definition 2.18. *Let $\mathcal{C}$ be a set of clusters, the nearest neighbor of a cluster $C_i \in \mathcal{C}$, denoted by $NN(C_i)$, is a distinct cluster $C_j \in \mathcal{C}$ such that*

$$\Delta(C_i, C_j) = \min_{C_k \in \mathcal{C} \setminus \{C_i\}} \Delta(C_i, C_k). \tag{2.89}$$

To make the nearest neighbor unique we choose the cluster C_j with the minimum identifier in case of ties.

If $C_j = NN(C_i)$ and $C_i = NN(C_j)$, the clusters C_i and C_j are called reciprocal nearest neighbors (RNNs).

Definition 2.19. *The nearest neighbor graph is the directed graph $\langle \mathcal{C}, E \rangle$ where the set of directed edges is $E = \{\langle C_i, NN(C_i)\rangle \mid C_i \in \mathcal{C}\}$.*

Proposition 2.20. ***(Reducibility property)*** *Let Δ be the Ward's dissimilarity (see Definition 2.2), and C_i, $C_j \in \mathcal{C}$ two reciprocal nearest neighbors, then there exists $\rho > 0$ such that for any other cluster $C_k \in \mathcal{C}$,*

$$\left.\begin{array}{rcl} \Delta(C_i, C_j) & < & \rho \\ \Delta(C_i, C_k) & > & \rho \\ \Delta(C_j, C_k) & > & \rho \end{array}\right\} \Longrightarrow \Delta(C_i \cup C_j, C_k) > \rho. \tag{2.90}$$

Proof. The proof can be done by just using the Lance-Williams' recurrence formula (Proposition 2.4),

$$\begin{aligned} \Delta(C_i \cup C_j, C_k) \quad > \quad & \frac{n_i + n_k}{n_i + n_j + n_k}\rho + \frac{n_j + n_k}{n_i + n_j + n_k)}\rho - \\ & - \frac{n_k}{n_i + n_j + n_k}\rho = \rho. \end{aligned} \tag{2.91}$$

□

Proposition 2.21. *Let C_i and C_j be two reciprocal nearest neighbors. If they are agglomerated and the reducibility property (Proposition 2.20) holds, then the nearest neighbor graph has to be updated only for those clusters which had C_i or C_j as nearest neighbor.*

Proof. Let C_k be an arbitrary cluster (different of C_i and C_j) in the nearest neighbor graph. It is enough to show that $\Delta(C_k, C_i \cup C_j) \geq \Delta(C_k, NN(C_k))$ because the only change is the agglomeration of C_i and C_j.

Note that, $\Delta(C_k, NN(C_k)) \leq \Delta(C_k, C_i)$ and $\Delta(C_k, NN(C_k)) \leq \Delta(C_k, C_j)$, so

- if $\Delta(C_k, NN(C_k)) > \Delta(C_i, C_j)$ then, using the reducibility property (Proposition 2.20) with $\rho = \Delta(C_k, NN(C_k))$,

$$\Delta(C_k, C_i \cup C_j) > \Delta(C_k, NN(C_k)), \tag{2.92}$$

- otherwise, let $\rho = \Delta(C_i, C_j)$ such that $\rho \leq \Delta(C_i, C_k)$ and $\rho \leq \Delta(C_j, C_k)$. Using again the reducibility property (Proposition 2.20), we get

$$\Delta(C_k, C_i \cup C_j) \geq \Delta(C_i, C_j) \geq \Delta(C_k, NN(C_k)). \tag{2.93}$$

□

Observe that, the equality (2.93) only holds when all the dissimilarities between C_i, C_j and C_k are equal.

Definition 2.22. *Let $C_i \in \mathcal{C}$ be an arbitrary cluster, a nearest neighbor chain from C_i is a directed path leading from C_i.*

Note that the maximum number of clusters in a nearest neighbor chain is $|\mathcal{C}|$ and, in this case, is the complete nearest neigbor graph.

Proposition 2.23. *The nearest neighbor chain satisfies the following properties:*

(i) Dissimilarities between consecutive clusters are not increasing.

(ii) The nearest neighbor chain cannot contain a $n-$cycle with $n > 2$.

(iii) The final two clusters of a nearest neighbor chain are an RNN pair.

Proof. (i) Let C_i, $C_j = NN(C_i)$ and $C_k = NN(C_j)$ be a subset of clusters in the nearest neighbor chain. Assume for contradiction that $\Delta(C_k, C_j) > \Delta(C_j, C_i)$. But we thus get $NN(C_j) \neq C_K$.

(ii) Let C_i, $C_j = NN(C_i), \ldots, C_k, NN(C_k) = C_i$ be a subset of clusters in the nearest neighbor chain. By property (i), $\Delta(C_i, C_k) < \Delta(C_i, C_j)$, thus $NN(C_i) \neq C_j$ and this is a contradiction.

(iii) Let $C_{nn(k)} \in \mathcal{C}$ be the last point of a nearest neighbor chain. This leads that its nearest neighbor has to be other cluster of the nearest neighbor chain, but from (ii), it has to be the predecessor of $C_{nn(k)} \in \mathcal{C}$, denoted by $C_{nn(k-1)}$. In other words, $NN(C_{nn(k)}) = C_{nn(k-1)}$ and, by definition, $NN(C_{nn(k-1)}) = C_{nn(k)}$.

□

By building a nearest neighbor chain, we have the so-called nearest neighbor chain algorithm, which performs the following steps,

Step 1. Let $C_i \in \mathcal{C}$ be an arbitrary cluster, get the nearest neighbor chain from C_i.

Step 2. Let C_j and C_k be the final two clusters of the nearest neighbor chain (the RNN pair), agglomerate them and replace with $C_{\min\{j,k\}} = C_j \cup C_k$.

Step 3. If $C_i \neq C_j$, carry on the nearest neighbor chain from the predecessor of C_j. Otherwise, return to step 1.

Step 4. Go to step 2 until there is more than one cluster in the set of clusters.

Definition 2.24. *Let $\mathcal{C}_0$ be a set of disjoint clusters, a stepwise dendrogram for $\mathcal{C}_0$ is a $(|\mathcal{C}_0| - 1) \times 3-$matrix whose i^{th} row is the triple $(C_{i_1}, C_{i_2}, \Delta(C_{i_1}, C_{i_2}))$ such that $C_{i_1}, C_{i_2} \in \mathcal{C}_i$ are reciprocal nearest neighbors and $\mathcal{C}_{i+1}$ is recursively defined as $\mathcal{C}_{i+1} = (\mathcal{C}_i \setminus \{C_{i_1}, C_{i_2}\}) \cup \{C_{i_1} \cup C_{i_2}\}$.*

Observe that a nearest neighbor chain produces a stepwise dendrogram. So now, our interest is to prove the uniqueness of the stepwise dendrogram (up to row order), in other words, to demonstrate that the reciprocal nearest neighbors do not depend on the order of the clustering steps.

Proposition 2.25. *Let $C_i, C_j \in \mathcal{C}$ be two reciprocal nearest neighbors and $C_k, C_l \in \mathcal{C}$ be two other reciprocal nearest neighbors. If Δ is the Ward's dissimilarity, then $\Delta(C_i \cup C_j, C_k \cup C_l)$ is independent of whether C_i, C_j are agglomerated first and then C_k, C_l or the other way round.*

Proof. Using the Lance Williams' recurrence formula (Proposition 2.4),

$$\begin{aligned}
\Delta(C_i \cup C_j, C_k \cup C_l) &= \frac{n_i + n_k + n_l}{n_i + n_j + n_k + n_l}\Delta(C_i, C_k \cup C_l) + \\
&+ \frac{n_j + n_k + n_l}{n_i + n_j + n_k + n_l}\Delta(C_j, C_k \cup C_l) - \\
&- \frac{n_k + n_l}{n_i + n_j + n_k + n_l}\Delta(C_i, C_j) = \qquad (2.94) \\
&= \frac{n_i + n_k}{n_i + n_j + n_k + n_l}\Delta(C_i, C_k) + \frac{n_i + n_l}{n_i + n_j + n_k + n_l}\Delta(C_i, C_l) + \\
&+ \frac{n_j + n_k}{n_i + n_j + n_k + n_l}\Delta(C_j, C_k) + \frac{n_j + n_l}{n_i + n_j + n_k + n_l}\Delta(C_j, C_l) - \\
&- \frac{n_i + n_j}{n_i + n_j + n_k + n_l}\Delta(C_k, C_l) - \\
&- \frac{n_k + n_l}{n_i + n_j + n_k + n_l}\Delta(C_i, C_j), \qquad (2.95)
\end{aligned}$$

where n_α for $\alpha = i, j, k, l$ is the number of observations in the cluster C_α.

Due to the symmetry, the dissimilarity between $C_i \cup C_j$ and $C_k \cup C_l$ does not depend on the order of the agglomeration. □

Proposition 2.26. *Let $\mathcal{C}$ be a set of disjoint clusters and let Δ be the Ward's dissimilarity. Let $C_\alpha, C_\beta \in \mathcal{C}$ be two reciprocal nearest neihbors, define $\mathcal{C}_0 = (\mathcal{C} \setminus \{C_\alpha, C_\beta\}) \cup \{C_\alpha \cup C_\beta\}$ and let $D_0 = ((C_{i_1}, C_{i_2}, \Delta(C_{i_1}, C_{i_2}))_{i=0,\dots,|\mathcal{C}_0|-1})$ be a sorted stepwise dendrogram for $\mathcal{C}_0$, i.e., $\Delta(C_{i_1}, C_{i_2}) \leq \Delta(C_{i+1_1}, C_{i+1_2})$ for all $i = 0, \dots, |\mathcal{C}_0| - 2$.*

Assuming that it exists $j \in \mathbb{N}$ such that $\Delta(C_{i_1}, C_{i_2}) < \Delta(C_\alpha, C_\beta)$ for all $i < j$ and $\Delta(C_{i_1}, C_{i_2}) \geq \Delta(C_\alpha, C_\beta)$ for all $i \geq j$, then the following matrix D is a sorted stepwise dendrogram for $\mathcal{C}$,

$$D = \begin{pmatrix} C_{0_1} & C_{0_2} & \Delta(C_{0_1}, C_{0_2}) \\ \vdots & \vdots & \vdots \\ C_{(j-1)_1} & C_{(j-1)_2} & \Delta(C_{(j-1)_1}, C_{(j-1)_2}) \\ C_\alpha & C_\beta & \Delta(C_\alpha, C_\beta) \\ C_{j_1} & C_{j_2} & \Delta(C_{j_1}, C_{j_2}) \\ \vdots & \vdots & \vdots \\ C_{(|\mathcal{C}_0|-1)_1} & C_{(|\mathcal{C}_0|-1)_2} & \Delta(C_{(|\mathcal{C}_0|-1)_1}, C_{(|\mathcal{C}_0|-1)_2}) \end{pmatrix}. \tag{2.96}$$

Proof. By the reducibility property (Proposition 2.20), it is clear that C_α and C_β are kept as reciprocal nearest neighbors after the agglomeration between other reciprocal nearest neighbors. Furthermore, the first j rows of D cannot include $C_\alpha \cup C_\beta$ because the dissimilarity between $C_\alpha \cup C_\beta$ and any other cluster is greater than $\Delta(C_\alpha, C_\beta)$.

Hence, the first j rows of D are valid for a sorted stepwise dendrogram for $\mathcal{C}$.

Afterward, it suffices to show that the dissimilarities are the same in the set $\mathcal{C}_0$ after j agglomerations (in sorted order) and in the set $\mathcal{C}$ after $j+1$ agglomerations (in sorted order) because the partitions of the original set are obviously equal.

On the one hand, the dissimilarity between a distinct cluster $C_k \neq C_{i_1} \cup C_{i_2}$ and $C_\alpha \cup C_\beta$ (resp. $C_{i_1} \cup C_{i_2}$) is independent of the agglomeration of C_{i_1} and C_{i_2} (resp. C_α and C_β) by definition. On the other hand, the dissimilarity $\Delta(C_\alpha \cup C_\beta, C_{i_1} \cup C_{i_2})$ does not depend on the order of the agglomeration by Proposition 2.25. This completes the proof. □

Corollary 2.27. *The resulting hierarchy does not depend on the order of the agglomeration.*

Proof. By Proposition 2.26, the rows in a stepwise dendrogram are independent of the agglomeration, thus the algorithm produces the same cluster agglomerations in all cases. □

Finally, note that, as mentioned in [9], the algorithm yields in $O(N^2)$ time and $O(N)$ space if the previous properties are satisfied.

2.2.2 Modified nearest neighbor chain for IDU XM resolver

The dissimilarity in the generalized source model (Section 2.1.4) does not satisfies the reducibility property, therefore the chain has to be restarted in each agglomeration and the final clusters can depend on the order of the agglomeration. It makes sense because initially we do not know information about the proper motion, and it depends on the agglomerations.

Moreover, we must take into account that our objective is to get the optimal clusters but in the algorithm described above, the iteration in step 4 ends when there is only a single cluster, therefore we have to stop the cluster agglomeration for the optimum solution.

Stopping rule

For the XM task, the agglomeration only makes sense while the dispersion of residuals within the clusters is below a given limit. This dispersion is measured by the variance $\sigma^2(C) = R(C)/n$ and the limit has to depend on Gaia observation error and the model error caused by not including the parallax.

Therefore, the limit is

$$\sigma^2_{lim} = \sigma^2_{pos} + \sigma^2_{par}, \tag{2.97}$$

where $\sigma_{pos} = 0.3 arcsec$ is the Gaia observation error and σ_p is the parallax error which is configured as:

$$\sigma_{par} = \begin{cases} A, & A > \sigma_t \\ \sigma_{p0}, & \text{otherwise} \end{cases} \tag{2.98}$$

where A is the cluster amplitude error, $\sigma_{p0} = 0.5 arcsec$ and $\sigma_t = 0.2 arcsec$.

On one hand, A is calculated as the maximum dispersion from the mean in each direction:

$$A = \sqrt{A^2_\alpha + A^2_\delta}, \quad A_u = \max\{b_j - u(t_j) | j = 1, \ldots, n\}, \ u = \alpha, \delta. \tag{2.99}$$

On the other hand, the parameters $\sigma_{p0} = 0.5 arcsec$ and $\sigma_t = 0.2 arcsec$ are configured with maximum parallax sources.

The parameter σ_{p0} corresponds to the maximum parallax values, thus initially the algorithm uses this value because we do not know the correct value and if the maximum cluster error is bigger than a threshold, denoted by σ_t, we consider that the maximum cluster error is a value that we can take into account.

Implemented features

Definition 2.28. *A pair of observations (O_i, O_j) is incompatible when they are from the same scan. Otherwise, a pair is considered compatible.*

Therefore, two detections have to be separated at least 106.5 minutes in order to be compatible. This value is extracted from the separation between the two telescopes and the spin rate of Gaia.

Definition 2.29. *Two disjoint clusters C_i and C_j are NN-compatible if the pairs of observations (O_{i_k}, O_{j_l}) are compatible for all $O_{i_k} \in C_i$ and $O_{j_l} \in C_j$.*

Therefore, the implementation includes the condition that two clusters can be nearest neighbors if and only if they are NN-compatible.

Note that, it forces observations in the same scan into different clusters.

Algorithm

Including these considerations, the modified algorithm performs the following steps,

Step 1. Initialize $\mathcal{C}$ with n non-finished clusters $C_i = \{O_i\}$, one for each observation.

Step 2. Let $C_i \in \mathcal{C}$ be an arbitrary non-finished cluster, get the nearest neighbor chain from C_i only until a cluster $C_j \in \mathcal{C}$ such that $\sigma^2(C_j \cup NN(C_j)) > \sigma^2_{lim}$.

Step 3. Let C_k and C_l be the final two clusters of the nearest neighbor chain. If $\sigma^2(C_k \cup C_l) \leq \sigma^2_{lim}$, agglomerate them and replace with the cluster $C_{\min\{k,l\}} = C_k \cup C_l$. Otherwise, consider C_l as a finished cluster and return to step 2.

Step 4. If $C_i \neq C_k$, get the nearest neighbor chain from $C_{\min\{k,l\}}$ only until a cluster C_j such that $\sigma(C_j \cup NN(C_j)) > \sigma^2_{lim}$. Otherwise, return to step 2.

Step 5. Go to step 3 until only one non-finished cluster remains in $\mathcal{C}$.

In the step 4, the nearest neighbor chain is restarted because the reducibility property does not hold. Then, the chain begins from the agglomerated cluster because it is preferred than an arbitrary cluster due to the assumption that the agglomeration is correct.

Despite of this, the above algorithm depends on the order of the agglomeration and the final clusters may not be completely correct, therefore a post-processing should be taken into account to correct it.

Finally, note that the time is incremented with the inclusion of these modifications. Nevertheless, this increment is not relevant because the previous stages provide MCGs with a number of MC not really huge.

More schematically, this algorithm is represented in the following flowchart,

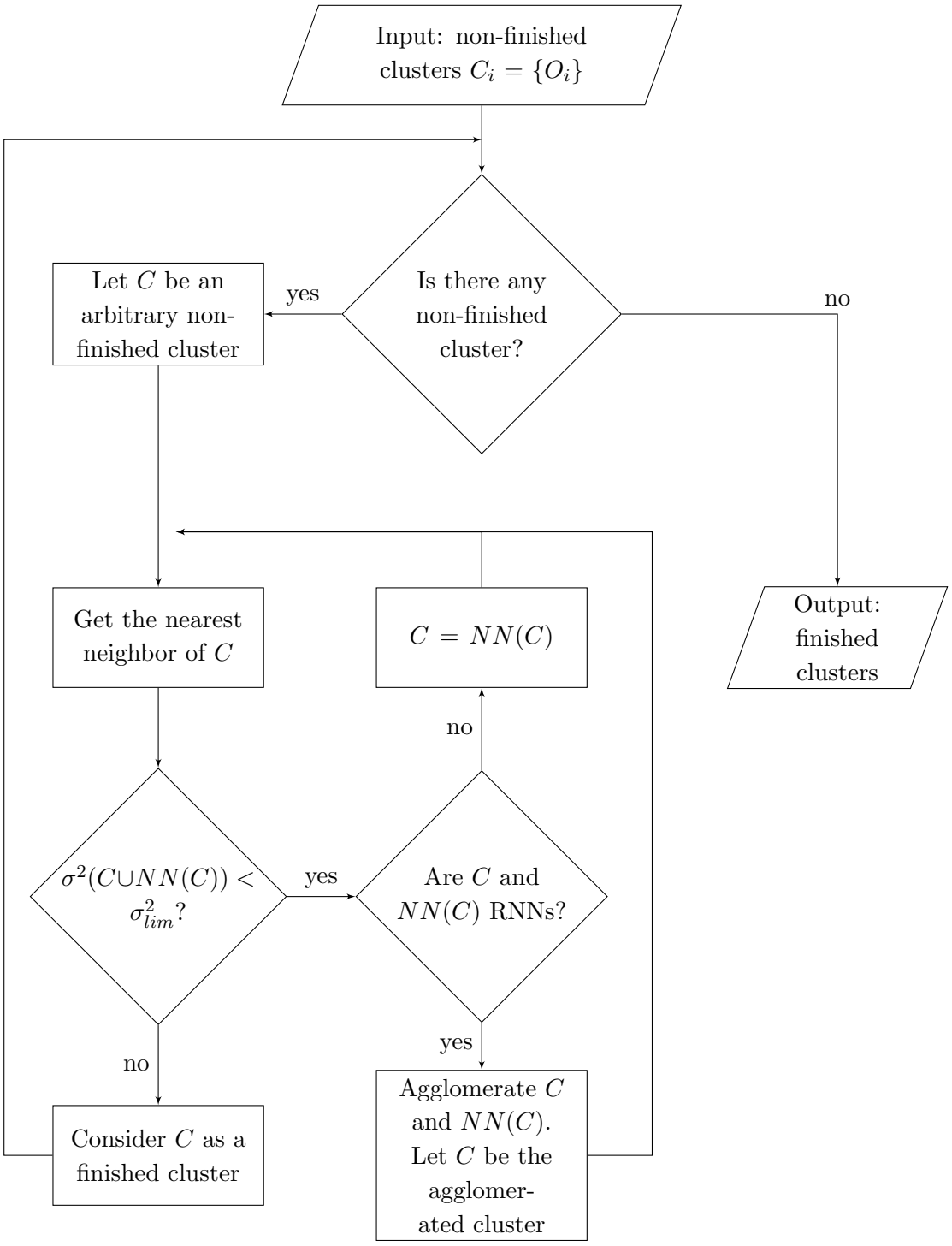

Figure 2.1: Flowchart for the modified NNC algorithm.

2.3 Post-processing algorithm

The post-processing algorithm has to detect and correct wrong cases due to the dependence of the order of the agglomeration.

These cases could be, but not only,

- a HPM star which is separated in two or more clusters,
- a HPM cluster with observations of several real sources,
- a high parallax source which is separated in two or more clusters,
- several clusters in crowded areas.

In all cases, there exist clusters with small number of observations. Thus, the clusters whose number of observations is at least the scans number divided by 2 are considered as well-finished clusters ($\mathcal{F}$), and the other clusters are considered as pending clusters ($\mathcal{P}$),

$$\mathcal{F} = \left\{ C \in \mathcal{C} \mid n > \frac{n(scans)}{2} \right\}, \tag{2.100}$$

$$\mathcal{P} = \left\{ C \in \mathcal{C} \mid n \leq \frac{n(scans)}{2} \right\}, \tag{2.101}$$

where $n(scans)$ is the number of scans.

Furthermore, if the number of finished clusters is less than the ratio $\left[\frac{|\mathcal{C}|}{n(scans)}\right]$, we will add the biggest pending clusters to reach this ratio, and we will remove them from pending clusters. Thus, $|\mathcal{F}| \geq \left[\frac{|\mathcal{C}|}{n(scans)}\right]$.

The method consists in breaking up the pending clusters into initial clusters and joining them to the finished clusters, if possible.

Since a source may not be observed in all scans, there can exist pending clusters which are correct. In this case, the initial clusters from this valid cluster cannot join with any finished cluster and we will have to consider it as a well-finished cluster.

Specifically, the algorithm performs the following steps,

Step 1. Break up all the clusters in $\mathcal{P}$ into initial clusters i.e., one for each observation.

Step 2. While there is an initial cluster C still pending,

(a) get the nearest neighbor of C from $\mathcal{F}$, i.e., $NN(C) \in \mathcal{F}$.

(b) If $\sigma^2(C \cup NN(C)) < \sigma^2_{lim}$, agglomerate the two clusters and remove C from the initial clusters.

(c) Otherwise, consider C as a non-finished cluster and remove it from the initial clusters.

Step 3. Do the modified NNC algorithm (Section 2.2.2) with the set of non-finished clusters.

Although each initial cluster can be agglomerated with other clusters which we know their shapes (in step 2), for the initial clusters that they cannot be merged into well-finished clusters, we recover the cluster from the modified NNC algorithm again (in step 3).

More schematically, this algorithm is represented in the following flowchart,

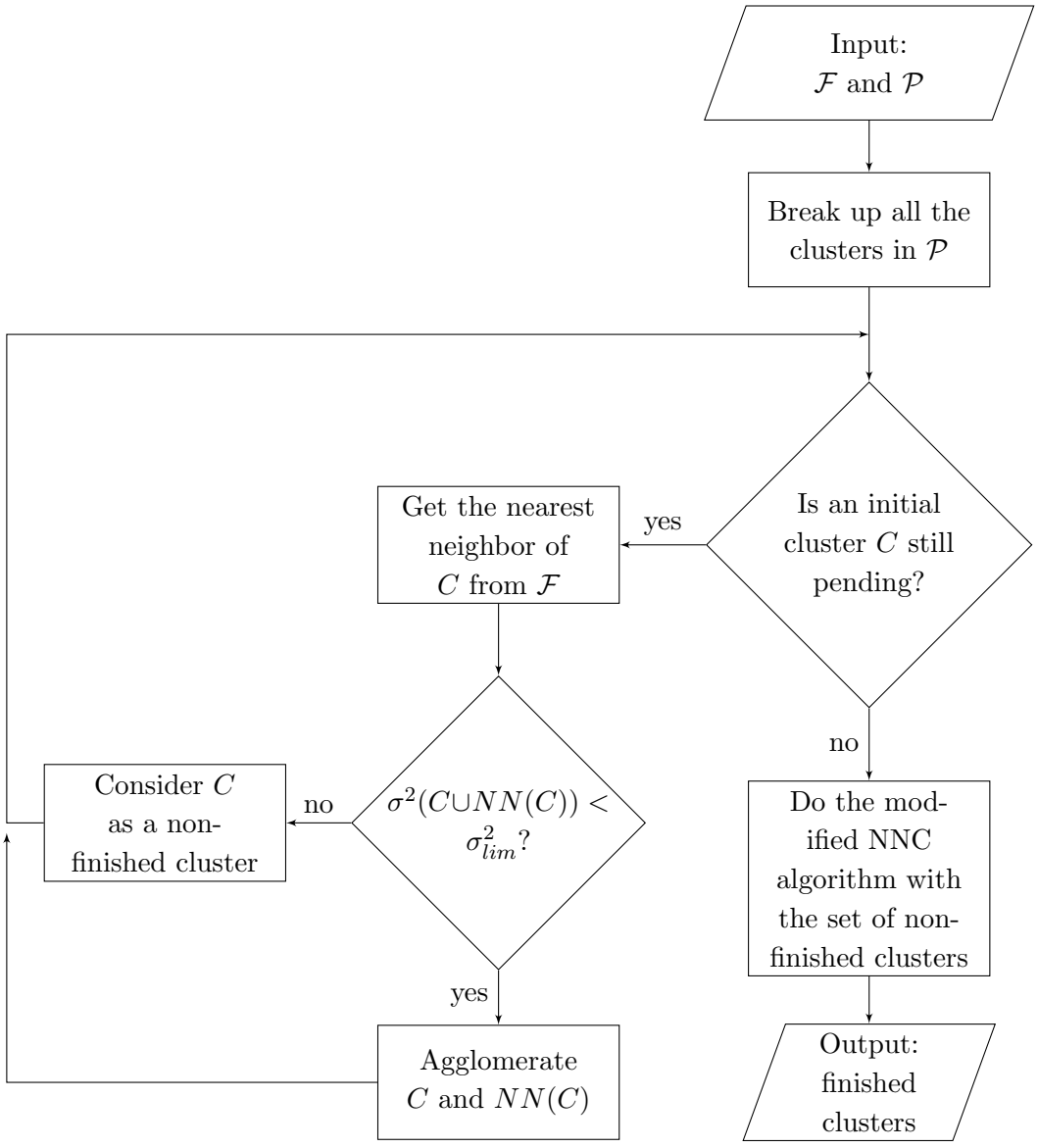

Figure 2.2: Flowchart for the post-processing algorithm.

Chapter 3

Explorative tests of the clustering-based cross-matching using Gaia data

The parameters and the implementation details of the algorithm explained in Chapter 2 have been configured using realistic Gaia data. Therefore, the purpose of this chapter is to assess some interesting realistic cases to show that the algorithm works as expected. To do so, we selected a set of high proper motion sources (HPM) and a set of high parallax sources (see Table 3.6.1 and Table 3.6.2 from [4].). Moreover, some crowded areas are analyzed as well.

Note that the Gaia observations may correspond to a valid source observations, but they may also correspond to spurious detections[1] not properly filtered. It is important to emphasize that the IDU-XM resolver task does not perform any kind of detection classification, which has been performed in previous stages. Thus we suppose that all the incoming detections are valid.

In the following figures, the observation size is according to the magnitude of the observation, i.e., a brighter observation is represented with a greater dot than a faint observation. In addition, the observations matched with an input source are represented with full blue dots, and the other ones are represented with empty blue dots.

3.1 Selected source cases

A correct inclusion of the proper motion was one of the main objectives of this work. In addition, as mentioned above in section 2.2.2, the stopping rule depends on the parallax

[1] the stars brigther than 15 magnitude create spikes in their near environment that the on-board detection system considers as new sources, the so-called *spurious detecions*.

error. Therefore, some real cases of HPM and high parallax sources are developed in this section.

The selected sources are sorted by decreasing proper motion.

3.1.1 HIP 87937 (Barnard's Star)

The main features of HIP 87937 are presented in Table 3.1 and the result of the clustering stage is shown in Figure 3.1.

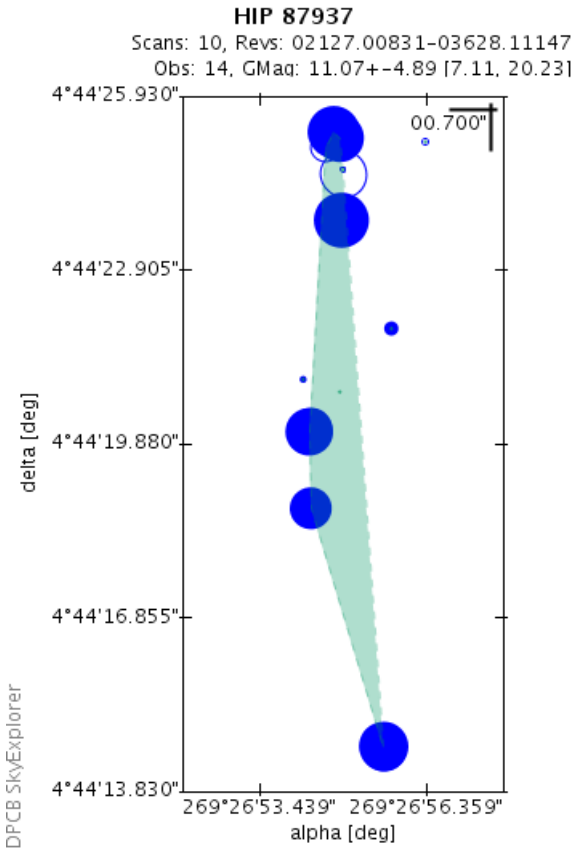

Figure 3.1: Clustering stage of IDU-XM Resolver around HIP 87937 with blue dots for observations and green areas for finished clusters.

The global algorithm sets the observations in 4 different finished clusters. Only one of them (with 10 observations) corresponds to the HPM source, as expected.

Therefore, the algorithm matches successfully all the Barnard's observations to the expected source.

Feature	Value
α	269.454°
δ	4.668°
Visual magnitude V	9.54 GMag
Total proper motion	10357.70 mas/year
Parallax	549.01 mas

Table 3.1: Main features of the input source from [4].

3.1.2 HIP 24186 (Kapteyn's Star)

The main features of HIP 24186 are presented in Table 3.2 and the result of the clustering stage is shown in Figure 3.2.

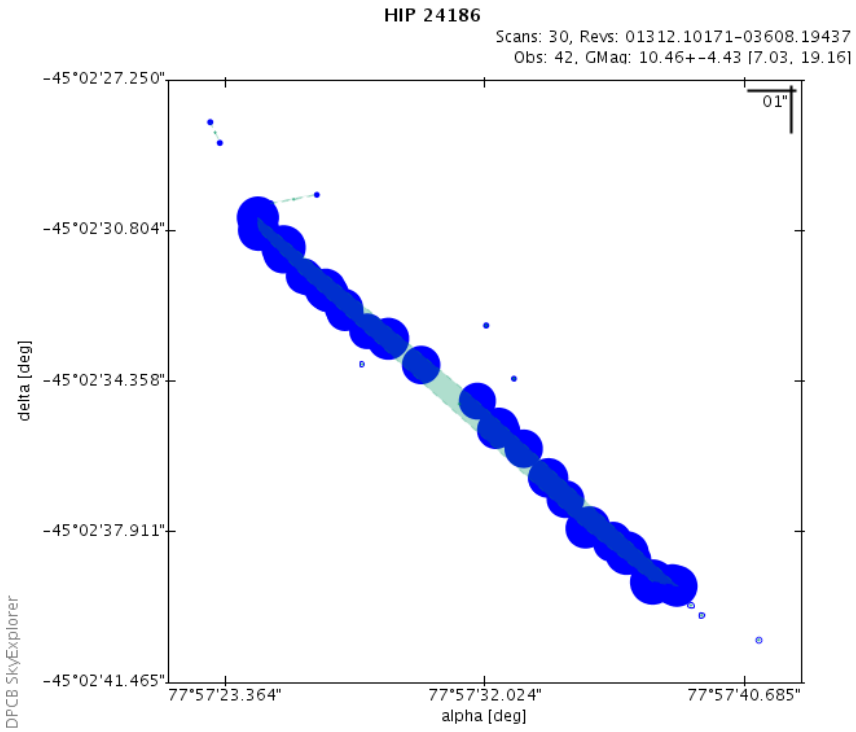

Figure 3.2: Clustering stage of IDU-XM Resolver around HIP 24186 with blue dots for observations and green areas for finished clusters.

The global algorithm sets the observations in 9 different finished clusters. Only one of them (with 29 observations) corresponds to the HPM source, as expected.

Therefore, the algorithm matches successfully all the Kapteyn's observations to the expected source.

Feature	Value
α	77.897°
δ	−45.004°
Visual magnitude V	8.86 GMag
Total proper motion	8670.50 mas/year
Parallax	255.26 mas

Table 3.2: Main features of the input source from [4].

3.1.3 HIP 70890 (α Cen C)

The main features of HIP 70890 are presented in Table 3.3 and the result of the clustering stage is shown in Figure 3.3.

Feature	Value
α	217.449°
δ	−62.681°
Visual magnitude V	11.01 GMag
Total proper motion	3852.99 mas/year
Parallax	772.33 mas

Table 3.3: Main features of the input source from [4].

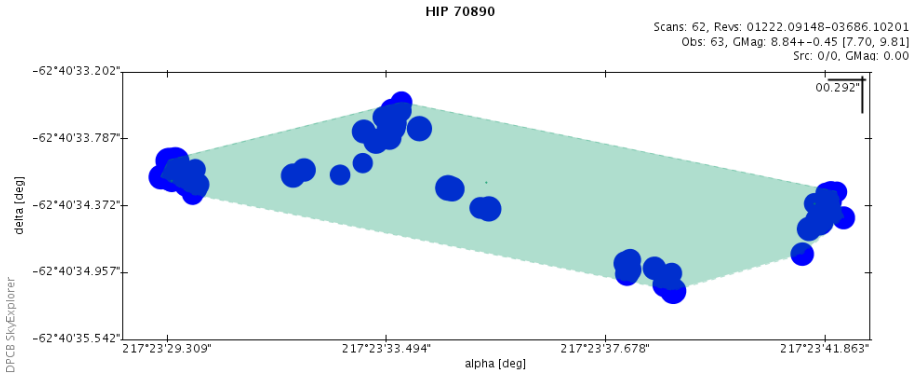

Figure 3.3: Clustering stage of IDU-XM Resolver around HIP 70890 with blue dots for observations and green areas for finished clusters.

Observe that only one finished cluster corresponds to the HPM source with 62 observations, as expected.

Therefore, the algorithm matches successfully all the observations to the expected source.

3.1.4 HIP 36208 (Luyten's Star)

The main features of HIP 36208 are presented in Table 3.4 and the result of the clustering stage is shown in Figure 3.4.

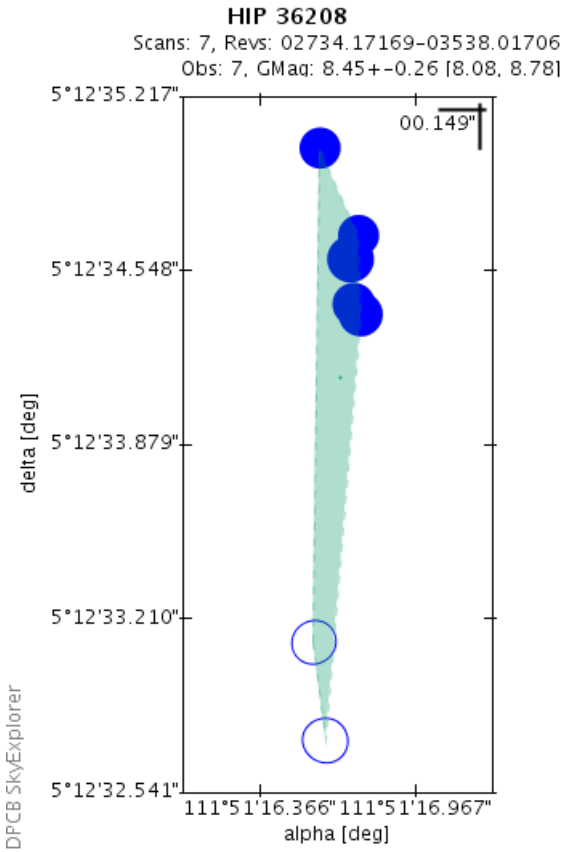

Figure 3.4: Clustering stage of IDU-XM Resolver around HIP 36208 with blue dots for observations and green areas for finished clusters.

Feature	**Value**
α	111.851°
δ	5.235°
Visual magnitude V	9.84 GMag
Total proper motion	3738.16 mas/year
Parallax	263.26 mas

Table 3.4: Main features of the input source from [4].

Observe that only one finished cluster corresponds to the HPM source with 7 observa-

tions, as expected.

Therefore, the algorithm matches successfully all the observations to the expected source.

3.1.5 HIP 74234

The main features of HIP 74234 are presented in Table 3.5 and the result of the clustering stage is shown in Figure 3.5.

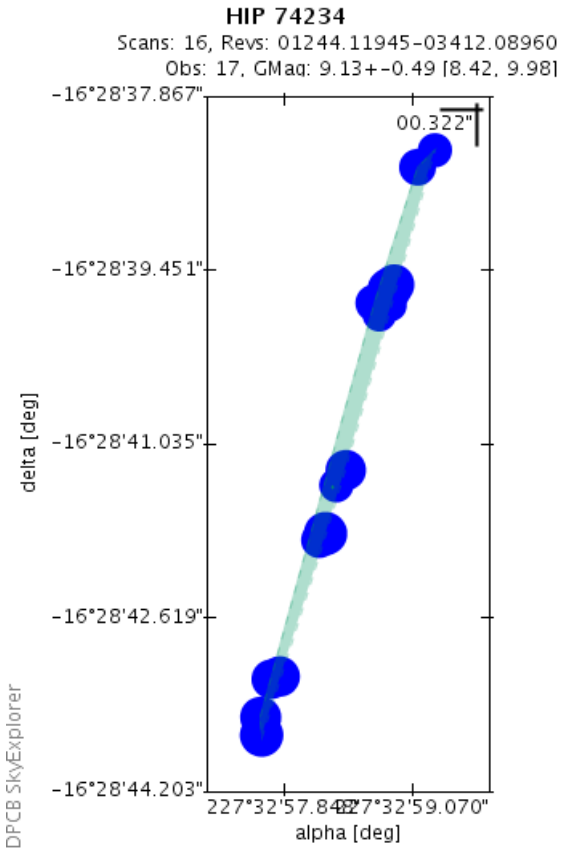

Figure 3.5: Clustering stage of IDU-XM Resolver around HIP 74234 with blue dots for observations and green areas for finished clusters.

Observe that only one finished cluster corresponds to the HPM source with 16 observations, as expected.

Therefore, the algorithm matches successfully all the observations to the expected source.

Feature	Value
α	227.557°
δ	−16.454°
Visual magnitude V	9.44 GMag
Total proper motion	3681.49 mas/year
Parallax	33.68 mas

Table 3.5: Main features of the input source from [4].

3.1.6 HIP 3829 (Van Maanen 2)

The main features of HIP 3829 are presented in Table 3.6 and the result of the clustering stage is shown in Figure 3.6.

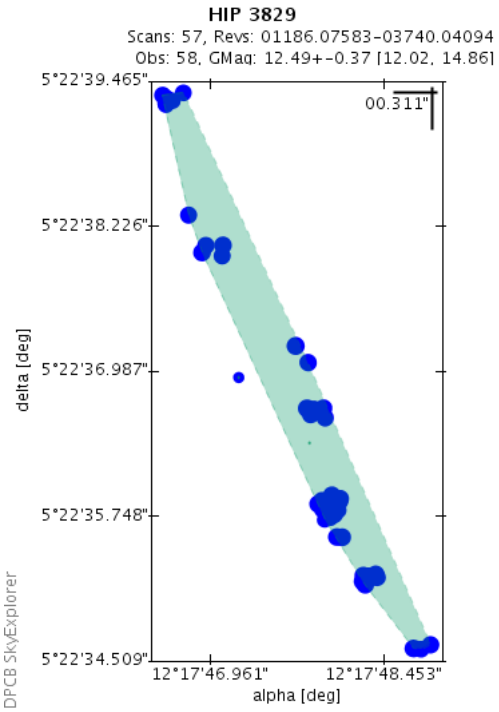

Figure 3.6: Clustering stage of IDU-XM Resolver around HIP 3829 with blue dots for observations and green areas for finished clusters.

The global algorithm sets the observations in 2 different finished clusters. Only one of them (with 57 observations) corresponds to the HPM source, as expected.

Therefore, the algorithm matches successfully the observations to the expected source.

Feature	Value
α	12.288°
δ	5.395°
Visual magnitude V	12.37 GMag
Total proper motion	2977.84 mas/year
Parallax	226.95 mas

Table 3.6: Main features of the input source from [4].

3.2 Tests in other selected areas

It is also interesting to analyze MCGs where the distance between sources is comparable to motions of HPM sources. These tests are shown in Figs. 3.7-3.9 and, according to the results of these tests, the algorithm's behaviour in crowded areas may also be benign.

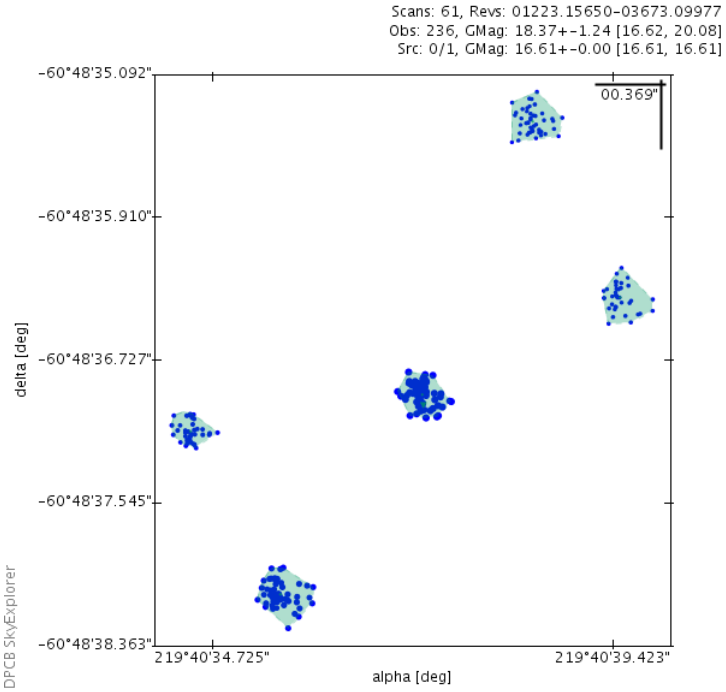

Figure 3.7: Clustering stage of IDU-XM Resolver with blue dots for observations and green areas for finished clusters. This MCG has 236 observations in 61 scans, and the algorithm sets the observations in 5 different finished clusters.

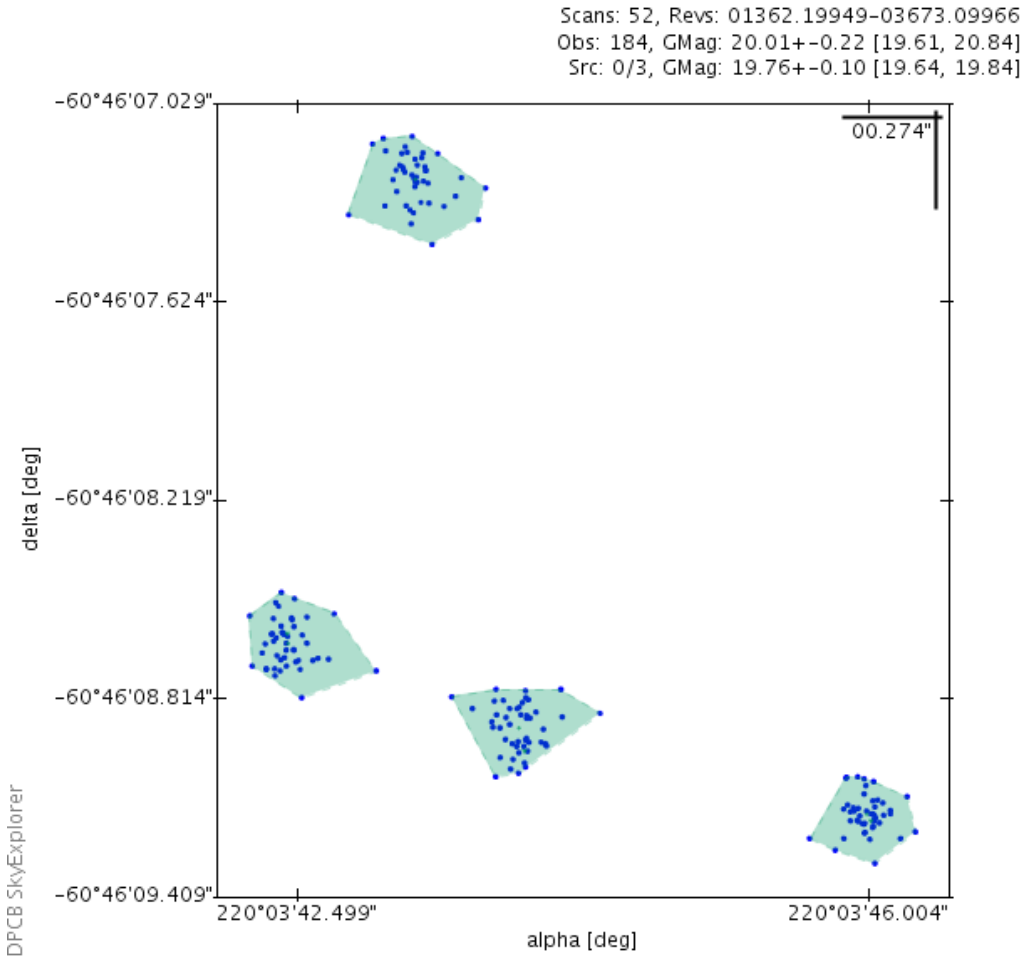

Figure 3.8: Clustering stage of IDU-XM Resolver with blue dots for observations and green areas for finished clusters. This MCG has 184 observations in 52 scans, and the algorithm sets the observations in 4 different finished clusters.

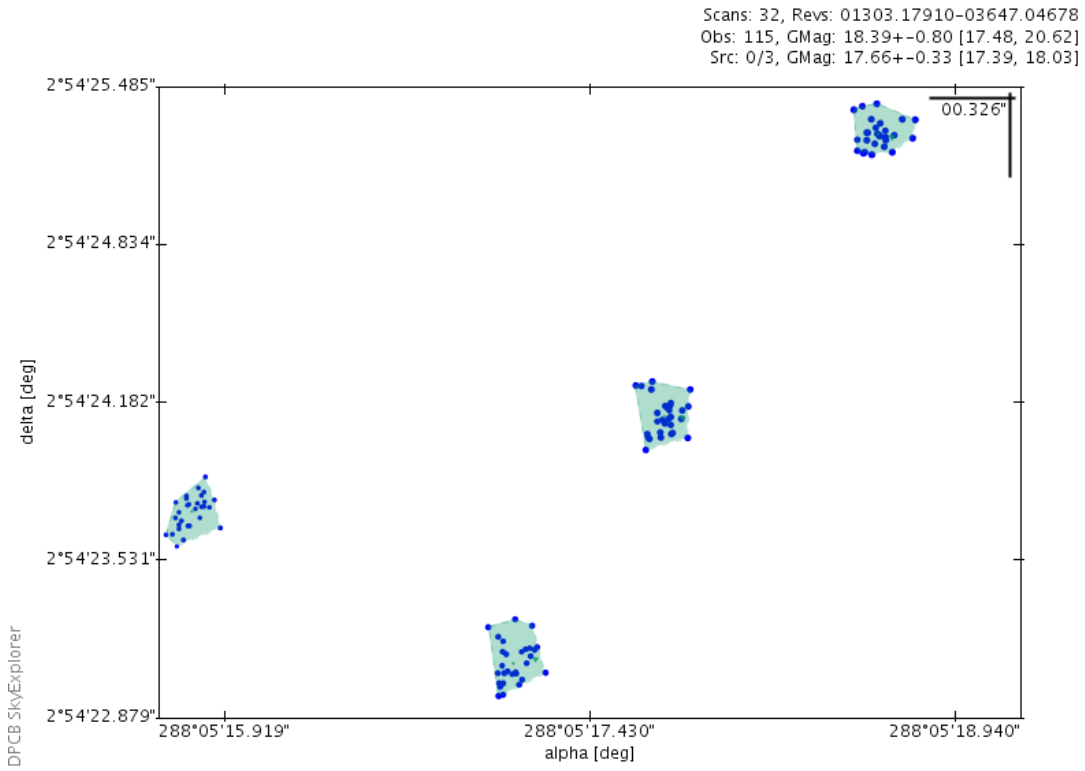

Figure 3.9: Clustering stage of IDU-XM Resolver with blue dots for observations and green areas for finished clusters. This MCG has 115 observations in 32 scans, and the algorithm sets the observations in 4 different finished clusters.

Conclusions

Since the beginning of its operational activities, Gaia detects a huge number of celestial objects including solar system objects and stars of our Galaxy -the Milky Way- and beyond, thus requiring an efficient data reduction strategy in order to provide the targeting accuracy of the final Gaia catalogue, expected in the early 2020s. In particular, one of the key tasks of the Gaia data reduction is the cross-matching (XM) of Gaia objects, which is aimed at providing a link between every Gaia observation and a source in a reference catalogue.

To do this, the adopted approach consists in the splitting of the Gaia XM in three different tasks: Observation to Source Match, Sky Partitioner and Match Resolver. Specifically, the Match Resolver task is divided in three stages (clustering, cluster linking, and conflict resolution) with the aim of offering an optimal XM resolution.

In this project, we have designed, developed and implemented a novel generalization of the clustering stage of the Match Resolver task. Concretely, this approach is based on a clustering analysis technique, namely the Nearest Neighbor Chain (NNC).

We have shown that the source model may be generalized to accommodate additional source parameters and that the NNC algorithm may be adapted to it. As a result, a suitable cluster analysis method has been identified with the inclusion of the proper motion and the magnitude of the sources from the observations.

In addition, in the model developed in this project, the stopping rule is dynamically computed during the clustering agglomeration process and it depends only on the observations of the specific cluster being processed, in contrast to other hierarchical methods where, in general, the stop criteria is fixed to a constant value under user intervention.

During the development of this project, we noticed that in the proper motion model the dissimilarity between two observations (i.e., clusters with only one observation) is always zero and therefore it would be possible to match any two observations. Thus, although the inclusion of the motion in the source model, initially we have had to agglomerate the observations in terms of position in order to retrieve then a better estimate of the motion of the source.

Moreover, the dissimilarity in the generalized source model does not satisfy the re-

ducibility property, and therefore the stack of observations within the NNC algorithm needs to be reset in each agglomeration iteration, which implies that the modified algorithm depends on the order of the agglomeration. To solve this, it has been considered, designed and implemented a post-processing to correct clusters which may not be coherent as a final cluster. This post-analysis allows to provide consistent and optimal sets of clusters.

As a future work, the inclusion of the parallax as well as other kind of source parameters may be taken into account. Specifically in the case of including the parallax in the source model, the current stopping rule may be updated thus not requiring any thresholding dependency in order to accommodate the displacements in the apparent position of a given object viewed along different lines of sight.

Conclusively, the results of the proposed generalization of the clustering stage for cross-matching Gaia objects developed in this project are consistent as well as promising in terms of performance. Moreover, the implemented algorithm provides an optimal resolution for observations from high proper motion sources.

Bibliography

[1] BIERMAN, G.:*Factorization methods for discrete sequential estimation*, Academic Press, New York, 1977.

[2] CASTAÑEDA, J.: *High performance computing of massive Astrometry and Photometry data from Gaia*, PhD thesis, UB, 2015.

[3] CLOTET, *et al.*: *Cross-matching algorithm for the Intermediate Data Updating system in Gaia*, in Highlights of Spanish Astrophysics IX, Bilbao, 2016.

[4] ESA: *The Hipparcos and Tycho Catalogues*, Vol. 1, ESA SP-1200, 1997.

[5] EVERITT, B.: *Cluster analysis*, 3rd. ed., Arnold, London, 1995.

[6] FABRICIUS, C. *et al.*: Astronomy & Astrophysics **595** A3, 2016.

[7] LINDEGREN, L.: Gaia Data Processing and Analysis Consortium (DPAC) technical note GAIA-LL-060, 2005.

[8] MÜLLNER, D.: *Modern hierarchical, agglomerative clustering algorithms*, Preprint in ArXiv: `arXiv:1109.2378v1`, 2011.

[9] MURTAGH, F.: The Computer Journal, **26**(4), 354-359, 1983.

[10] RAO, C.R.; TOUTENBURG, H.; SHALABH; HEUMANN, C.: *Linear models and generalizations: least Squares and alternatives*, Springer, New York, 2008.