Highlights on Spanish Astrophysics IX, Proceedings of the XII Scientific Meeting of the Spanish Astronomical Society held on July 18 – 22, 2016, in Bilbao, Spain. S. Arribas, A. Alonso-Herrero, F. Figueras, C. Hernández-Monteagudo, A. Sánchez-Lavega, S. Pérez-Hoyos (eds.)

Cross-matching algorithm for the intermediate data updating system in Gaia

M. Clotet, J.J. González-Vidal, J. Castañeda, N. Garralda, J. Portell, C. Fabricius, and J. Torra

Institut de Ciències del Cosmos, Universitat de Barcelona (IEEC-UB), Martí i Franquès 1, E08028 Barcelona, Spain.

Abstract

Cross Matching (XM) is an inherently difficult problem in astronomy. The assignation of which detection belongs to a given source is a complex issue that has deep implications in further usages of the data. Gaia provides a massive amount of new observations every day which must be linked to sources so that further data reduction can take place. The XM in Gaia provides a consistent match between observations and sources in the working catalogue for subsequent data reduction processes.
The system in charge of performing the XM in Gaia is designed in three stages. First the input observations are processed by time in order to compute the sky coordinates and obtain the preliminary source candidates for each individual detection. Then, a second task groups the results to determine isolated groups of detections, avoiding boundary issues. Finally, the relations between the observations and corresponding sources are provided.

1 Introduction

Gaia is an astrometric space mission of the European Space Agency (ESA)[3]. The mission will measure the positions, motions and parallaxes of more than 1 billion stars in our Galaxy and beyond up to the 20th magnitude. The data volume produced is expected to require more than one petabyte of disk storage. The first Gaia Data Release (Gaia DR1)[4] required the processing of more than 40TB of data and 30 billion observations. The latest estimations predict that the Gaia data processing will require more than 10^{21} flops [7].

The mission ground segment is formed by six Data Processing Centres (DPCs), managed by the Data Analysis and Processing Consortium (DPAC). The data reduction is formed by several systems which process the raw astrometric, photometric and spectrometric measurements downloaded from the spacecraft. The systems are run cyclically considering all the accumulated data since the beginning of the mission. Each iteration or Data Reduction

Cycle (DRC) uses the latest calibrations and improves the scientific quality of the previous results [3].

Between all the systems in DPAC, the Intermediate Data Updating (IDU) is one of the most demanding in terms of data volume managed and required processing power. One of the IDU tasks, the Cross-Match (IDU-XM), aims at providing an updated XM table between each observation and a source in the Gaia catalogue. Without this task most downstream data processing systems would not be able to produce the expected results or reach the envisaged accuracy.

2 Cross-Match in Gaia

In DPAC there are two systems that provide a XM between observations and sources in the catalogue. On a daily basis the Initial Data Treatment (IDT) system performs a preliminary XM that is used by all the daily systems [2]. IDT must process the data received from the satellite in near real time, so the data might not be complete due to the downlink priority scheme when it is processed. Therefore, the XM resolution of some sky regions, complex cases or high proper motion sources might be poor.

In every DRC IDU recomputes the XM using the latest updates on attitude, calibrations, etc. Unlike IDT, all the detections are available beforehand. Moreover, there are less restrictions on computing requirements which allow more complex algorithms to be used. The IDU-XM is used by all the cyclic processing software in DPAC and is the base upon which the Gaia data releases are built.

The IDU-XM process assumes that all the incoming detections are valid as they have been filtered beforehand [5]. Technically the task is implemented in three stages:

- Obs-Src Match: In this first stage, the input observations are processed by time in order to compute the detection sky coordinates and obtain the preliminary source candidates for each individual detection. First the observation coordinates are computed. Next relevant catalogue sources are propagated to the observation epoch and a radius search is performed. All sources that fall within the preconfigured distance criterion become *source candidates* for this observation.
- SkyPartitioner: In this stage the observations are grouped according to their source candidates. The aim is to determine isolated groups of observations, located in a fairly small and limited sky region. The resulting groups contain all the observations which are interconnected by their source candidates. By design, any given observation or source will only appear in one group. This process provides a convenient spatial data arrangement which solves boundary issues. As any two separate groups do not share any observation or source they can be processed independently.
- XM resolver: In this final stage the XM for each group is resolved. The process takes into account all the observations and sources in the group. This stage will resolve conflicting scenarios – one observation linked to two sources for example – with the aim of providing the correct match for each observation.

3 IDU-XM resolution algorithm

The XM resolution in Gaia DR1 was performed by a simple algorithm which solved conflicts between sources and observations individually. Observations were processed by pairs and a new source was created if required.

This algorithm is extremely simple and fast but it presents certain problems. For instance, the number of new sources is not minimised. This allows the observations to scatter between several unnecessary new sources. Given all the limitations a new algorithm was developed. This new algorithm is divided into three different steps: clustering, cluster-source linkage and resolution.

3.1 Clustering

The first step is to group observations into smaller sets. The aim is to cluster together all the observations belonging to the same source. After comprehensive analysis of multiple clustering techniques a customised Nearest Neighbour Chain (NNC) algorithm was selected which builds upon a preliminary study conducted by Lennart Lindegren.

The NNC algorithm works by finding pairs of mutual nearest neighbours to be merged [6] [1]. The required memory grows linearly with the number of elements to be clustered and the time is linear with respect to the number of distances between pairs of points [8]. The algorithm only requires the observations as input, a definition of distance between observations, and a stop criterion.

Our distance definition includes the observation sky coordinates as well as the observation magnitude with a calibrated weight. The magnitude is included to improve the clustering performance in complex cases. The dissimilarity is used as a stop criterion for the cluster agglomeration. In particular, Ward's minimum variance method which defines the dissimilarity of a given cluster as the sum of the squared residuals (SSR) with respect to the cluster centre, see Eq. 1.

$$R(C) = \sum_{O \in C} ||x(O) - x(C)||^2 \tag{1}$$

If the original clusters have SSR $R(C_i)$ and $R(C_j)$ respectively, Eq. 2 shows the SSR for the agglomerated cluster.

$$R(C) = R(C_i) + R(C_j) + \frac{n(C_i)n(C_j)}{n(C_i) + n(C_j)} ||x(C_i) - x(C_j)||^2 \tag{2}$$

The third term of Eq. 2 is taken as the measure of the dissimilarity between clusters. Therefore, $R(C)$ and $n(C)$ are monitored as the agglomeration proceeds. We introduce a rule to only allow an agglomeration if the resulting internal variance $R(C)/n(C)$ is below a predefined limit. This stop criterion, has been carefully calibrated for the specific IDU-XM scenario.

3.2 Cluster-Source linkage

The second step of the resolution task creates links between clusters and candidate sources within the group. The objective is, for each cluster, to create a list of existing sources that might be assigned to it. This resembles conceptually the Obs-Src Match process but with clusters instead of single observations.

We use the source candidates from each observation in the cluster as possible source candidates for the cluster. The advantage is that these links have been computed with the source positions propagated to each observation epoch. This improves the results and allows clusters to be linked to high proper motion (HPM) candidate sources consistently.

3.3 Resolution

The third step is the actual resolution of all conflicts between cluster-source links. A given source can only be assigned to one cluster. Therefore, if a cluster has links to multiple sources, or vice versa, these are considered conflicting. The aim is to provide the final resolution which must be a unique assignment between clusters and sources, and therefore between observations and sources.

When analysing a given cluster, each conflict for the current cluster can be resolved in multiple ways. One option is to keep the link, assuming it is correct and thus removing all the links to this source for all other clusters. Another choice is to break the link, and analyse if there are remaining conflicts for the current cluster. If there are none we can continue with the next cluster.

There are two other options available to resolve conflicts. The source merging process shown in Fig. 1 and the source splitting process shown in Fig. 2.

Figure 1: Merge process. Blue circles represent observations, crossed circles clusters, red stars existing sources in the catalogue and green stars new sources created by the merging process. The semi-opaque red sources represent deprecated sources.

The clusters in a group are processed sequentially removing conflicts one by one. Each conflict resolution affects the options available for subsequent clusters. This creates a *decision tree* where the leaves contain all the possible resolutions of the initial scenario.

The problem in resolving the conflicts between cluster and source assignation is that the order and how conflicts are solved affects the final resolution.

The processing iterates the clusters and analyses conflicts one by one. This requires an established cluster sort order. Figure 3 shows how the final resolutions obtained differ depending on the first cluster that is analysed (top left or right branches).

Figure 2: Split process. Blue circles represent observations, crossed circles clusters, red stars existing sources in the catalogue and green stars new sources created by the split process. The semi-opaque red sources represent deprecated sources.

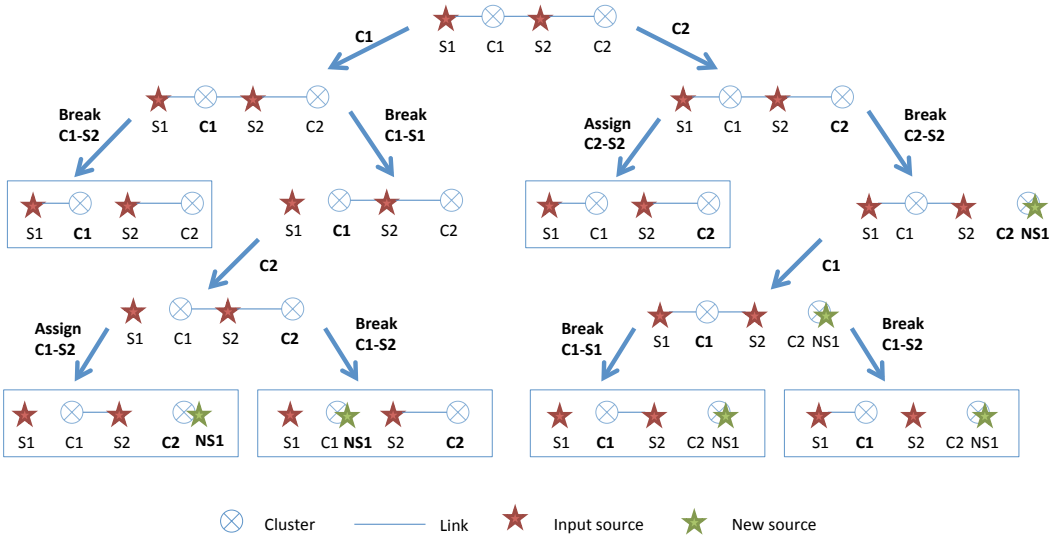

Figure 3: Decision tree example. Bold text indicates which cluster is being analysed in each step. Each blue box represents a valid final solution.

Up to a certain complexity level it is possible to explore all possible resolution order permutations which for n clusters are $n!$. The algorithm establishes limits to the number of permutations explored. However, in general the number of clusters in a group will be rather small. We also performed an in-depth analysis of the order in which the clusters will produce the optimal resolution. The algorithm explores these permutations first which speeds up the whole resolution.

To keep the best resolution a metric has to be defined. The metric must allow resolutions to be compared in order to decide whether one is considered better or worse than another. This allows the algorithm to determine the *best* resolution. However, a metric is also useful to limit the number of tree branches visited during the decision tree traversal. If a partial resolution is already *worse* than the previous *best* resolution found the algorithm will skip this branch and continue with the next one. In the example shown in Fig. 3, the system would skip processing the right-most branch (where a new source is created for C2) if we previously found one of the other solutions that does not require any new source.

At the end of the tree traversal the algorithm has kept the best resolution found and can proceed to generate a table containing for each observation the corresponding source, namely the Match table.

4 Conclusion

A novel XM resolution algorithm has been described here with the aim of improving the previous IDU-XM algorithm. We propose a resolution stage based on the clustering of observations using a tailored NNC algorithm, followed by a conflict resolution stage. The conflict resolution strategy explores all the possible resolutions to find the best one according to a given set of criteria defined beforehand. The algorithm has shown an excellent performance with both simulated and real data scenarios offering an optimal resolution in almost any situation.

Acknowledgments

This work was supported by the MINECO (Spanish Ministry of Economy) - FEDER through grant ESP2014-55996-C2-1-R and MDM-2014-0369 of ICCUB (Unidad de Excelencia *María de Maeztu*)

References

[1] Bruynooghe, M. 1977. Statistique et analyse des données, 2(3), pp.24-42.

[2] Fabricius, C., Bastian, U. Portell, J., et al. 2016, A&A, in press (Gaia SI).

[3] Gaia Collaboration (Prusti et al.) 2016, A&A, in press (Gaia SI).

[4] Gaia Collaboration (Brown et al) 2016, A&A, in press (Gaia SI).

[5] Garralda, N., Fabricius, C., Castañeda, J., Portell, J., Clotet, M., González-Vidal, J.J. and Torra, J. 2016. in Highlights of Spanish Astrophysics IX (this volume)

[6] Juan, J. 1982, Les cahiers de l'analyse des données, 7(2), pp.219-225.

[7] Mignard, F., Bailer-Jones, C., Bastian, U., et al. 2007, Proceedings of the International Astronomical Union, 3(S248), pp.224-230.

[8] Murtagh, F. 1983. The Computer Journal, 26(4), pp.354-359.