73rd International Astronautical Congress, Paris, France. 18-22 September 2022.

IAC-22-A6.9.1

Applying Graph-based Clustering to Tracklet-Tracklet Correlation

Franziska Griese[a], Kathrin Rack[a*], Melanie Schmidt[b], Daniel Schmidt[b], Simon Schmitz[a], Benjamin Hofmann[c], Thomas Schildknecht[d], Hauke Fiedler[c]

[a] *Department HPC, Institute for Software Technology, German Aerospace Center, Linder Höhe, 51147 Köln,* Germany, Franziska.Griese@dlr.de, Kathrin.Rack@dlr.de, Simon.Schmitz@dlr.de

[b] *Heinrich Heine University Düsseldorf, Universitätsstraße 1, 40225 Düsseldorf, Germany,* mschmidt@hhu.de, dschmidt@hhu.de

[c] *Space Flight Technology department (GSOC), German Aerospace Center, Münchener Straße 20, 82234 Weßling, Germany,* Benjamin.Hofmann@dlr.de, Hauke.Fiedler@dlr.de

[d] *Astronomical Institute, University of Bern, Sidlerstrasse 5, 3012 Bern, Switzerland,* Thomas.Schildknecht@aiub.unibe.ch

*Corresponding Author

Abstract

In order to identify new resident space objects from observations like tracklets, well-known algorithms like the tracklet-tracklet correlation are applied, which estimate whether a pair of tracklets might belong to the same resident space object. Subsequently, an orbit determination has to be performed for every pair to confirm the object. This procedure is known to be time consuming. We will show that an interposed clustering analysis enhances the computational speed of the whole process by reducing the number of false associations and thus the number of needed validations. Cluster analysis is a commonly used machine learning technique for grouping objects. It has been shown to be very successful in many fields. Starting from other research in the field of tracklet association, we use Markov Clustering, a graph-based clustering algorithm. We use a large observation data set provided by SMARTnet, which was split into subsets for training and testing. An important part of our work is the development of evaluation techniques that are suited for our task. Classical evaluation techniques often do not fulfil our requirements, because tracklets of the same object may coexist in more than one connected component of the graph that is clustered. That is the case because the edges of the graph result from the tracklet-tracklet correlation. In such a case, it is impossible to obtain a cluster containing all tracklets of one object. These scenarios are not considered in the established evaluation methods of clustering results. We present modifications of methods which allow for efficient evaluation of the clustering results and optimization of the cluster analysis for object identification. Furthermore, we show that our training results in a successful clustering for diverse test data. The whole process is realized in a data management and processing system for orbital objects called "Backbone Catalogue of Relational Debris Information" (BACARDI).

Keywords: Markov-Clustering, Tracklet-Association, Python, BACARDI, Clustering-Evaluation, Space Debris

1. Introduction

The rapidly growing number of resident space objects poses an increasing threat to manned and unmanned missions, and especially to operated satellites [1]. It is therefore imperative to observe and track the object population in Earth's orbit as comprehensively as possible. To achieve this, it is a prerequisite to gather information about new resident space objects that are not already catalogued. To move things forward, a data management and processing framework for observation data of resident space objects, called Backbone Catalogue of Relational Debris Information (BACARDI)[2], is developed and maintained by the German Aerospace Center (DLR).

The result of observations we are interested in here are short arcs, called tracklets. However, for a single tracklet, the corresponding state vector is not accurate enough to predict the object's position or to keep this object in a data base. Therefore, after importing tracklets into BACARDI, it is first tried to associate the tracklets to already catalogued objects [3]. If this fails, the tracklets are used to identify new objects. To calculate a meaningful initial orbit for a new object, we need at least two tracklets of the same object. To identify which tracklets belong to the same object, it is state of the art to

test pairwise with other uncorrelated tracklets [4]. We will refer to this process as tracklet-tracklet correlation (TTC). For every associated pair, an orbit determination is performed in order to verify if the tracklet-pair belongs to the same object.

However, the TTC has some disadvantages, e.g. for larger time spans between measurements we obtain wrong correlations, which has also been observed elsewhere [5]. It follows that huge computational resources would be occupied in order to examine a huge number of false associations.

Based on the results of the TTC, first attempts exist to improve this process by Markov clustering [5]. The Markov clustering algorithm has already been successfully used for a broad range of applications, e.g. grouping of protein sequences [6, 7] but also identifying galaxy groups [8].

As in the work of Yanez et al. [5], we will apply Markov Clustering to find clusters of pairwise correlated tracklets, which ideally belong to the same object and thus reduce the number of false associations. As a consequence, the orbit determination only has to be called for every cluster, thus reducing the computational time. Furthermore, a number of more than two tracklets improves the quality of the orbit determination, and objects might even be identified if a small fraction of tracklets in the cluster belongs to another object.

Moreover, in this work we will show how to modify typical evaluation methods in order to evaluate the quality of the clustering for this kind of application. Furthermore, we will apply a training and testing strategy to determine the parameters of the Markov clustering. For all this, we will use a large data set provided by SMARTnet, a global telescope network jointly operated by DLR's German Space Operations Center (GSOC) and the Astronomical Institute of the University of Bern (AIUB) and supported by international partners [9]. We utilize data from several months and several sensors and correlate measurements with a time interval of up to ten days between them in order to model a procedure that is realistic for application.

2. Methods

This section is a recap of literature knowledge. We will shortly introduce the tracklet-tracklet-correlation, Markov clustering method and the evaluation methods.

2.1 Tracklet-Tracklet Correlation

The input for the clustering process is the result of a pairwise association of tracklets, also called tracklet-tracklet correlation (TTC) [4]. From telescope measurements, we know the right ascension α and declination δ and their time derivatives $\dot{\alpha}$ and $\dot{\delta}$ of two tracklets i and j at the times t_i and t_j with $i \neq j$. The aim of the TTC is to identify which pairs of tracklets belong to the same object. Therefore, a boundary value formulation is used to evaluate a hypothesized orbital state of a pair of tracklets and an optimization scheme should find the best fitting orbits. Every hypothesized orbital state is defined by the number of half-revolutions, the direction of its orbital motion around Earth, and the positions of the potential object relative to the Earth. For every hypothesized orbital state, Lambert's problem [10, 11] is solved, which gives the estimated velocities of the object $\tilde{\mathbf{v}}_i$ and $\tilde{\mathbf{v}}_j$. The calculated velocities are compared to the measured velocities $\mathbf{v}_i$ and $\mathbf{v}_j$ by the evaluation function

$$L = (\dot{\mathbf{z}} - \tilde{\dot{\mathbf{z}}})^T (C_{\dot{\mathbf{z}}} + C_{\tilde{\dot{\mathbf{z}}}})(\dot{\mathbf{z}} - \tilde{\dot{\mathbf{z}}}), \tag{1}$$

with $\dot{\mathbf{z}} = (\dot{\alpha}_i, \dot{\delta}_i, \dot{\alpha}_j, \dot{\delta}_j)$ the angular rates calculated from the velocity and the corresponding covariances C. The tilde $\tilde{}$ marks the estimated results from Lambert's problem. This function is on the one hand used as the loss function for the optimization scheme and on the other hand used to decide which tracklets will be considered for the cluster analysis. Pairs with a loss value L below a threshold L^* will be considered, the others will not.

2.2 Markov Clustering

The Markov Clustering Algorithm (MCL) is a graph-based clustering method invented by van Dongen [12].

A weighted graph $G = (V, E, w)$ is an abstract structure composed of a finite number of edges E, which connect two vertices V, also called nodes, with weight w. However, we use $w = 1$ throughout this work, and vertices correspond to tracklets. Two vertices are connected via an edge, if the loss value L of the corresponding tracklet pair is smaller than the threshold value L^*. Since there are no directed relations between two vertices, we speak of an undirected graph.

The advantage of graph-based clustering algorithms is that a similarity measure is not needed between every pair of data points. This is important in our case, because from the TTC we do not obtain a loss value L for every pair. This is due to several constraints, e.g., the algorithm to solve Lambert's problem might not converge or the threshold for the loss value L^*_{TTC} or the time interval $\Delta t_{\max,TTC}$ is exceeded. Thus, we need a clustering algorithm that deals with this appropriately. The graph-based clustering translates a missing loss value to a missing edge. Furthermore, benchmark tests have shown that graph-based algorithms can compete with other clustering methods in terms of performance [13].

MCL partitions a graph by simulating random walks. The idea is that a random walk stays

mainly within dense subgraphs and just rarely walks through sparse connections between dense regions. Algebraic matrix operations eliminate the inter-cluster connections and enhance the intra-cluster connections. Algorithm 1 describes the MCL.

Algorithm 1 Markov Clustering

Input: graph G, self-loop parameter $c \geq 0$, expansion $e > 1$, inflation $l > 1$.
Output: clustering of G.

1: Generate adjacency matrix A from G and normalize columns.
2: Add self-loops to A as: $A = A + cI$, with the unity matrix I.
3: Normalize columns of A.
4: **while** termination criterion not fulfilled **do**
5: Expand: $A = A^e$.
6: Inflate: $a_{i,j} = a_{i,j}^l \forall a_{i,j} \in A$ then normalize every column of A.
7: Prune: Remove every entry of A smaller than threshold (0.001) but leave maximum of row.
8: **end while**
9: Identify weakly connected components of the graph associated with A that represent clusters.

For the MCL, we need three parameters: the self-loop parameter, the inflation, and the expansion. To avoid negative eigenvalues in the adjacency matrix, which correspond to oscillating behaviour [12], we use self-loops with edge weight 1 and thus set the self-loop parameter $c = 1$. The expansion is responsible for ensuring that distant vertices can be reached in a random walk. The inflation amplifies the differences between the matrix elements and thus affects the granularity of the solution. We will set the parameters inflation and expansion in Section 5.

For the graph implementations, we use the Python package `networkx` [14] and for the MCL implementation we use the package from Allard et al. [15].

2.3 Evaluation methods

In order to find the best parameter set, we need to evaluate the results of the clusterings. Since we know which tracklets belong to which object, we can use this ground truth to compare it to the clustering results. Here, we will use three evaluation methods, the adjusted Rand Index (ARI), the adjusted Mutual Information (AMI), and the pairwise F-score (PFS). The former methods are typically used in the field of multi-class clustering evaluation. The PFS is based on the classical F-score [16], which is typically used for binary classification problems, and will be modified in this work. They are all symmetric. For ARI and AMI, we use the implementation of the `scikit-learn` Python package [17, 18], and for PFS we use the pair confusion matrix provided in the same package.

For the ARI and the PFS, the data points, here the tracklets, are evaluated pairwise. Every pair is grouped into the categories shown in Table 1.

Table 1: Confusion matrix for pairwise treatment of data points.

Ground truth	Predicted clustering: same cluster	Predicted clustering: different clusters
same cluster	true positive (TP)	false negative (FN)
different clusters	false positive (FP)	true negative (TN)

The (unadjusted) Rand Index (RI) is a measure of similarity [19] and is given by

$$\mathrm{RI} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}} \in [0, 1], \qquad (2)$$

where TP, TN, FP, FN are the number of true positives, true negatives, false positives, and false negatives, respectively (see also Table 1).

However, for random labeling, the score is increasing when the number of clusters is approaching the number of data points, which is shown in Figure 1. Vinh et al. [20] showed that a correction is important if the number of data points is small compared to the number of clusters. This is the case in our application, because we usually measure just a few tracklets per object.

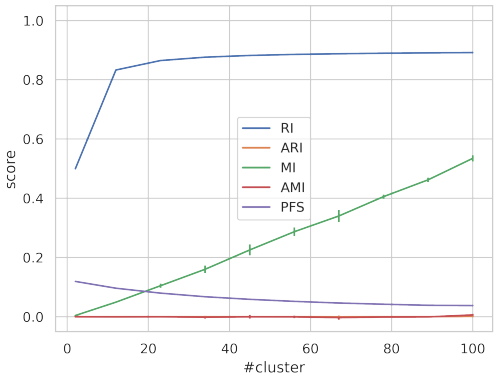


Figure 1: Scores of evaluation methods of a clustering with random uniform labeling against a reference assignment of 10 clusters. Number of data points is fixed to 1000. Scores of ARI and AMI are similar. Figure is based on [17].

For the RI, this effect is corrected using the expected RI for random labeling $E[\mathrm{RI}]$ [21, 22]. The

so-called *adjusted Rand Index* (ARI) is then given as

$$\mathrm{ARI} = \frac{\mathrm{RI} - E[\mathrm{RI}]}{\max(\mathrm{RI}) - E[\mathrm{RI}]} \in [-1, 1], \tag{3}$$

with $max(\mathrm{RI})$ being the maximum RI. Thus, for random labelling, the score is close to zero and, for similar clusterings, it is close to 1, cmp. Fig. 1.

The *Adjusted Mutual Information (AMI)* is based on basic concepts of information theory. The (unadjusted) Mutual Information (MI) measures the amount of information two compared clusterings share. The more information they share, the more similar are the clusterings.

Let U and V be two clusterings on N similar data points, then the *Mutual Information* (MI) [23] of U and V is

$$\mathrm{MI}(U, V) = \sum_{i=1}^{|U|} \sum_{j=1}^{|V|} P(i, j) \log\left(\frac{P(i, j)}{P(i)P'(j)}\right), \tag{4}$$

with $P(i) = |U_i|/N$ and $P'(j) = |V_j|/N$ the probability that a random data point is in the cluster U_i and V_i, respectively. Furthermore, $P(i, j) = |U_i \cap V_j|/N$ is the probability that a random data point is part of the cluster U_i and V_j.

For MI, the score also depends strongly on the number of data points and clusters. Therefore, the expectation value of MI $E[\mathrm{MI}]$, for details see [20], is again used to adjust the result. The adjusted Mutual Information (AMI) [20] is given by

$$\mathrm{AMI} = \frac{\mathrm{MI} - E[\mathrm{MI}]}{\mathrm{mean}(H(U), H(V)) - E[\mathrm{MI}]} \in [-1, 1], \tag{5}$$

with $H(U) = -\sum_{i=1}^{|U|} P(i) \log(P(i))$, $H(V) = -\sum_{j=1}^{|V|} P'(j) \log(P'(j))$ the entropy of U and V, respectively. An AMI score close to zero is obtained for independent clusterings, a score close to one means the clusterings show significant similarities, and a score of exactly one is obtained for identical clusterings.

In this work, we also use a modification of the classical *F-score* [16]. The F-score is typically used to measure the accuracy of a binary classification. If the truth label of every predicted cluster is known, a one-versus-all approach can be used for multi-class classification problems.

Since we do not know which object should be represented by which cluster, we use the pairwise F-score (PFS), which counts true positive, false negative, and false positive pairs.

The F-score combines the precision (P) and the recall (R), also called the sensitivity of the model, given by

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \in [0, 1], \tag{6}$$

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \in [0, 1]. \tag{7}$$

Precision is the proportion of pairs correctly classified as positive out of all pairs classified as positive. Recall is the probability of classifying a positive pair correctly as a positive pair.

The F-score weights recall β times higher than precision and is therefore defined by

$$\mathrm{F}_\beta = \frac{(1 + \beta^2)\mathrm{PR}}{\beta^2 \mathrm{P} + \mathrm{R}}, \tag{8}$$

with $\beta \geq 0$ [24].

For our problem here, it is most important that two tracklets that belong to different objects also belong to different clusters. Therefore, we use $\beta = 1/2$.

Compared to ARI and AMI, true negative pairs are not taken into account, and therefore the pairwise F-score (PFS) does not show any dependence between number of clusters and number of data points, see Fig. 1.

3. Data processing

The way our Python package works can be broken down into the steps as shown in Figure 2. First, the TTC, which is realized as a separate Python package, has to be run. One part of the package prepares the data for creating a graph from it. An optional pre-processing step can manipulate the data, e.g., depending on the threshold of the loss value L^* or time differences between measurements. Afterwards, the graph is created.

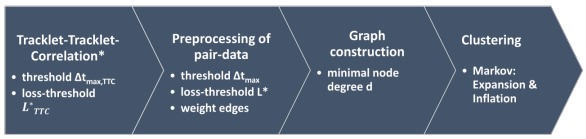
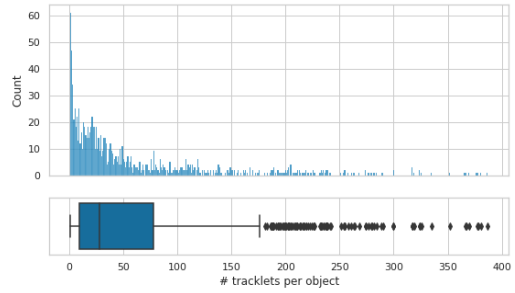


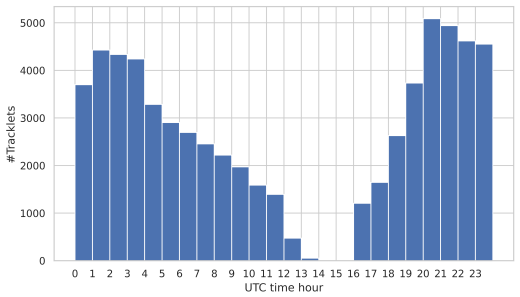
Figure 2: Process of tracklet clustering. From the TTC we obtain the data. Thereby, a threshold $\Delta t_{\mathrm{max},TTC}$ and loss threshold L^*_{TTC} can be defined. The asterisk * implies that the TTC is an external Python package. The data is then pre-processed for the graph construction. In this step, the thresholds for loss L^*_{TTC} and time interval Δt_{max} can be specified. Next, the graph is constructed with a defined minimal node degree d. Following that, the clustering is performed.

Figure 3 gives an impression of our data set we obtained from the telescope network SMARTnet [9]. It consists of 64,155 tracklets, which have been created from observations in the period from April 2017 to December 2019. Figure 3a shows the number of tracklets per object. Within the data set we have 1107 objects, of which 61 objects consist of just one tracklet and thus are unusable for an initial orbit determination. Figure 3b shows that the data set does not cover 24h of UTC-time. This is

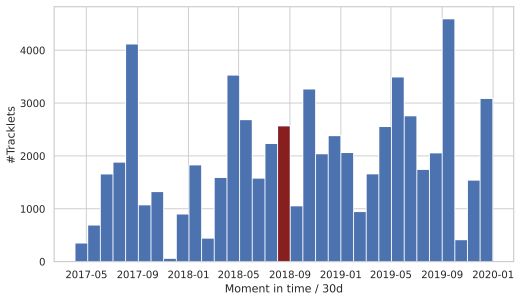
due to the local distribution of contributing telescope stations and the fact that observations are only performed at night. From Figure 3c, we can see that the number of tracklets per 30-day time interval varies a lot.



(a) Histogram of number of tracklets per object with corresponding box-plot. It shows that 50% of the objects contain between 10 and 78 tracklets. The median is 28.



(b) Number of tracklets per hour of UTC-time.



(c) Number of tracklets per time interval of 30 days. The month August 2018 (coloured in red) is taken for the training.

Figure 3: Statistical analysis of whole data set.

In the following, we use the tracklets of the month of August 2018 as training data set. We choose this month because it contains a total of 2615 tracklets relatively evenly distributed over the time interval. These tracklets belong to 404 different objects. In addition, the month contains some objects that were rarely observed and some that have been observed frequently (up to 50 times). This means that the variety and complexity of possible tracklets will be maximized. The median value of the number of tracklets per object is four. Multiple tracklets per object are important because we need at least two tracklets per predicted cluster to determine an initial orbit. Since the number of tracklets compared to the number of objects is small, it is important to use the for chance adjusted evaluation methods ARI and AMI as already stated in section 2.3.

Figure 4 shows the association results for the training data set with the loss threshold $L^* = 0.1$ and the maximum time difference of tracklet pairs $\Delta t_{\max} = 10\,\mathrm{d}$. Similar to the results in the work of Yanez et al. [5], we observe mixed clouds of false and correct associations, especially for larger time differences Δt between the measurements. The clouds are due to the fact that the observations do not cover 24h of UTC-time, see Fig. 3b. Since our data set is much larger than the data set used by Yanez et al. [5], it is more pronounced. Furthermore, we consistently observe in Figure 4 that the result for the TTC is best for small Δt.

In order to choose a good threshold value L^*, we look at the percentage of false associations for tracklets with a maximum time difference of $\Delta t_{\max} = 0.5\,\mathrm{d}$ depending on L^*. These tracklet pairs have been observed during the same night and thus lead to more reliable TTC results than tracklet pairs with larger time distances. As shown in Figure 5, the percentage of false associations starts to drastically increase from $L^* = 0.1$. Thus, we chose $L^* = 0.1$. With this and $\Delta t_{\max} = 10\,\mathrm{d}$, we have 75% false associations for the training data set. Note that Yanez et al. [5] chose $L^* = 1$ and had 31.5% false associations for $\Delta t_{\max} = 2.5\,\mathrm{d}$. With our data set, and using $\Delta t_{\max} = 2.5\,\mathrm{d}$ and $L^* = 1$, we obtain about 56% false associations, and with our selected threshold value of $L^* = 0.1$, we obtain about 32.2% false associations, close to the observation of Yanez et al. [5].

Since some space objects are observed infrequently due to various observation planning strategies, we set the maximum time distance of tracklet pairs to $\Delta t_{\max} = 10\,\mathrm{d}$ to increase the probability of observing at least the required two tracklets.

In order to reduce the number of false associations before the clustering, we try to remove the nodes with a low degree. Such nodes, being part of a sparse subgraph, are expected to correlate more strongly with a wrong association than a node in a dense subgraph. Removing all nodes from the graph that have a degree smaller than various thresholds $d_{\min}$ for our training data set, we see that the fraction of wrong associations increases, compare Figure 6. A look at Figure 7, which shows an example graph with 2093 nodes and 22 032 edges, shows that the graph consists of one dense part with 1162 vertices and many sparse subgraphs. Note that the positions of the nodes are generated with the spring layout provided by `networkx` [14]. In Figure 7b, we can observe that dense subgraphs mostly contain correct associations. However, many sparse

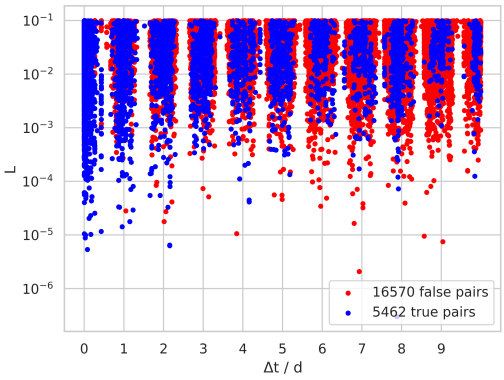


Figure 4: Loss values L of tracklet pairs in log scale as a function of the time interval Δt in days between the two measurements. Here, we set $\Delta t_{\max} = 10\,\mathrm{d}$ and $L^* = 0.1$.

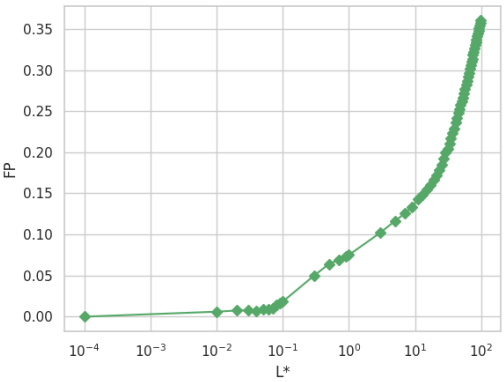


Figure 5: Fraction of false positives (FP) at different threshold values L^*. Maximum time difference of two tracklets is set to $\Delta t_{\max} = 0.5\,\mathrm{d}$.

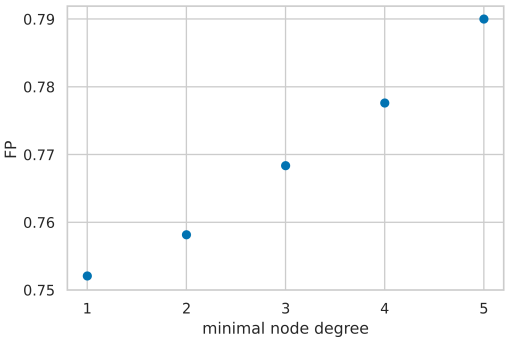
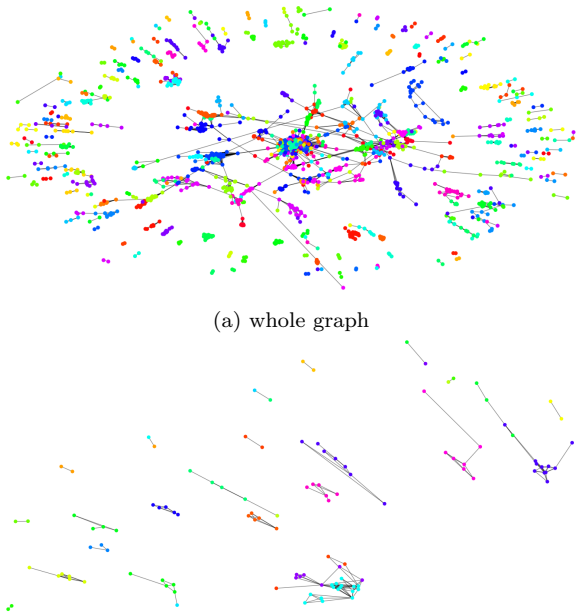


Figure 6: Fraction of false positives (FP) contained in the graph for different minimum node degrees $d_{\min}$.



(a) whole graph

(b) section of graph

Figure 7: Graph for the training data set. Vertices of the same colour belong to tracklets of the same object.

graphs also contain correct associations. Thus, by removing nodes with a degree $d > 1$, we also remove sparse subgraphs and with it increase the fraction of wrong associations. Therefore, we chose $d_{\min} = 1$, as with this we only remove isolated nodes.

4. Clustering evaluation

Combining the TTC with a subsequent graph clustering will lead to graphs that normally depart in different connected components, compare Fig. 7. This is due to the thresholds $\Delta t_{\max}$ and L^* introduced above. It is likely that different connected graph components contain tracklets from the same object and thus, it is impossible for the clustering algorithm to cluster all tracklets of one object together. However, if the tracklets t_i^o and t_j^o, which belong to the same object o but are in different connected components, belong to different clusters C_i and C_j with $i \neq j$, then the usual evaluation methods will count the pair (t_i^o, t_j^o) as a false negative pair, compare Table 1 for ARI and PFS. This has an unwanted negative impact on the evaluation of the clustering result. For the AMI the evaluation score also decreases, because the shared information between the clustering and the ground truth decreases as well. Note that ground truth currently means that all tracklets of one object are in one cluster.

Our aim for an application in BACARDI is to remove as many false associations from the tracklet-tracklet correlation while losing as few correct associations as possible. Therefore, we need evaluation methods that do not penalize the algorithm for the case described above. Moreover, this is important to be able to compare the clustering results from different data sets. Without modifications, it is possible that a clustering C of a graph G, which has

several false associations, achieves a higher evaluation score than a clustering C' of a graph G', which contains only true associations but decomposes into several components.

Therefore, we evaluate different modifications of the evaluation methods that do not penalize the algorithm if tracklets of an object are located in different subgraphs.

Modification 1 *Given a graph G, which consists of k connected graph components, and let $s_1, ..., s_k$ be the scores of an evaluation method of the single connected components $1, ..., k$. Then we define the score of the evaluation method as the mean of the subscores*

$$S = \frac{1}{k} \sum_{i=1}^{k} s_i.$$

In Modification 1, node pairs from different components are not considered and thus, the influence of equal objects in different components is eliminated. In Figure 8, we see that varying the inflation has almost no influence on the score. A higher inflation results in more clusters. Due to the unequal size of the connected components, and since each component contributes equally to the score, changes in the clustering of the large connected component have hardly any influence on the score. Therefore, Modification 1 is not useful for us.

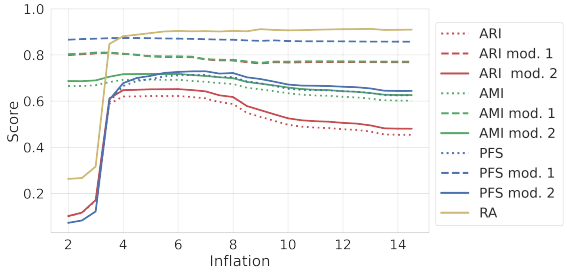


Figure 8: Comparison of evaluation methods with different modifications for the Markov-Clustering of the training data set. The respective score is shown as a function of the inflation l. The Expansion is fixed to $e = 6$. Scores of ARI and AMI with Modification 1 are similar.

We are also testing another modification of the evaluation methods, which does not penalize finding two nodes that belong to the same object but occur in different connected components.

Modification 2 *We define the modified ground truth clustering $\bar{C}_{truth}$ of the graph $G = (V, E, w)$ as the clustering $\bar{C}_{truth} = \{\bar{C}_1, \bar{C}_2, ...\}$ that, for nodes u, v, which belong to the same object and graph component, yields that $u, v \in \bar{C}_i$ else $u \in \bar{C}_i$ and $v \in \bar{C}_j$ with $i \neq j$.*

Modification 2 means that instead of having all tracklets of one object in one ground truth cluster, we split this ground truth cluster into several clusters, depending on the connected graph components this object is split up into in the initial graph. Thus, for ARI and PFS, the pair (t_i^o, t_j^o) with $t_n^o \in C_n$ and $n \in [i, j]$ is now a true negative instead of a false negative pair. For all evaluation methods introduced in 2.3, no punishment for splitting up into different components is observed. However, true negative pairs increase the score for ARI and AMI and thus, there is an influence of the number of objects in different graph components on the score.

Since true negative pairs do not influence the PFS, we have successfully eliminated the influence of the decay in this method with Modification 2. As a result, we can compare clusterings of different graphs with the PFS modified by Modification 2.

However, although Modification 2 for ARI and AMI is difficult for the comparison of clusterings of different data sets, it is possible to use it to compare the clustering results of the same data set, e.g., for different clustering parameters. This is because, starting with the same graph, the number of true negative pairs due to equal objects in different components leads to the same increase of the score for all of the clusterings. However, to compare different data sets, we can use the F-score with Modification 2, where true negatives are not considered. In the following, we will use Modification 2.

Additionally, for evaluation of clusterings, we use the fraction of right associations (RA) inside the clusters, which will be calculated as

$$RA = \frac{|E_{\text{eq}}|}{|E_{\text{eq}}| + |E_{\text{neq}}|}, \tag{9}$$

where $|E_{\text{eq}}|$ are the number of edges which connect nodes of the same object inside the clusters and $|E_{\text{neq}}|$ are the number of edges connecting nodes of different objects inside the clusters.

5. Tracklet-Clustering results

Knowing the evaluation methods, we now look at our training data set and search for the best values for the expansion e and the inflation l.

Figure 9 shows the results of MCL with different parameter sets applied to the training data. We see for MCL a broad range of parameters leading to good results, which shows the stability of this algorithm. Due to Fig. 9, we decide on values for the parameters expansion $e = 6$ and inflation $l = 6$.

In Table 2 we can see the reached scores of the different evaluation methods. We find that from the initial 75% false associations the number has reduced to about 10%. Thereby, we still have 3975 correct associations from an initial value of 5462

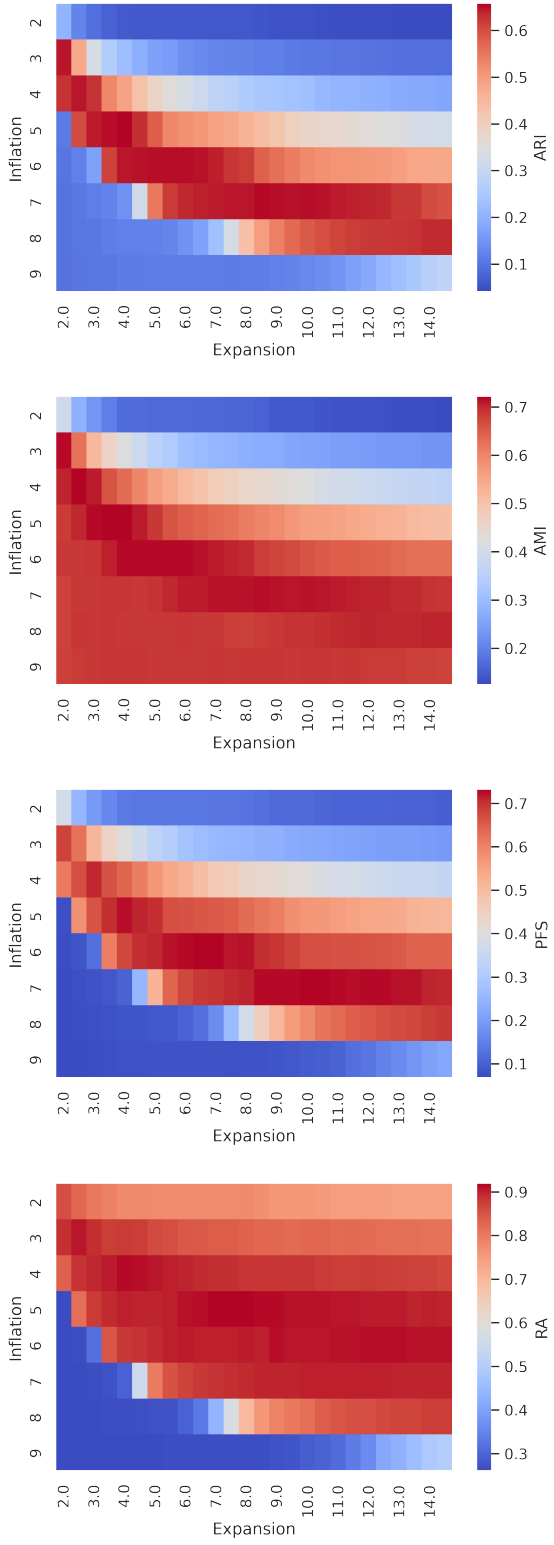


Figure 9: Markov Clustering Algorithm (MCL) results for different parameter sets of expansion and inflation evaluated with ARI, AMI, PFS, and RA. The colour code shows the score. Note the score range is different for the four diagrams.

correct associations. However, from the 16 570 false associations we just have 456 left.

Table 2: Clustering result of training data set for expansion $e = 6$ and inflation $l = 6$.

	Markov
ARI	0.64
AMI	0.71
PFS	0.73
RA	0.89

In order to estimate how many "new" space objects we would recognize, we assume that the orbit determination will work if a cluster contains at least two tracklets and all of them belong to just one object. The training data set contains a total of 404 objects. Tracklets not associated with any other tracklet by the TTC are not considered in the constructed graph. As a consequence the training data set graph includes a total of 300 space objects. The clustering result contains in total 788 clusters of which 300 clusters include more than one tracklet. 182 of these contain just tracklets from one object each. These clusters include 143 different objects and thus, we would find 143 new space objects. Consequently, 39 clusters represent an object already represented by another cluster. Initial orbit determination even works with some wrong tracklets included. Therefore, if we assume that just 75% of the tracklets have to belong to one object, we would have found 168 new space objects. Furthermore, for just applying the TTC we would need to perform orbit determinations for 22 032 correlated pairs. Whereas, after clustering the orbit determination has to be performed just for the 300 clusters that contain more than one tracklet.

After optimizing the parameters with the training data set, they have to be tested for an independent data set. For testing, we use the remaining available months from April 2017 to December 2019, evaluate the clustering results with the fixed parameters, and compare the evaluation results.

Figure 10 shows that for almost every score 25% of the test data is performing better than the training data. However, for ARI, AMI, and PFS, the median is worse than the results from training, whereas RA is better. On average, just 23% of the false associations but more than 68% of the true associations remain. Furthermore, the maximum number of false associations remaining is about 74%. It follows that even in the worst case 26% of the false associations are still filtered out. This shows that MCL is successfully filtering the false associations from TTC. Furthermore, it has to be noted that the data sets of all months vary a lot in their characteristics. Figure 3c gives an impression of the variability. We can assume that not only the num-

ber of tracklets differ per month but also the implied graph structure, for example, the number of connected components and the number of tracklets per component. Taking these considerations into account, we can expect the Markov Clustering to be good enough for our application.

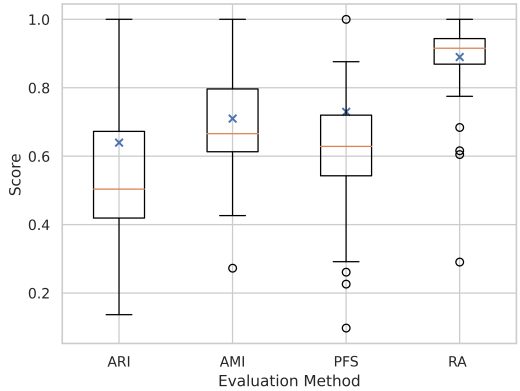


Figure 10: Statistics of MCL results on monthly data sets for the modified evaluation methods ARI, AMI, PFS, and RA. The orange lines show the medians, and blue crosses mark the scores of the training data set.

6. Outlook

We applied the Markov Clustering to a large and real data set, which we split up into a training set and a test set. As is common in the field of machine learning, we searched for the inflation and expansion parameters in a training data set and examined the applicability on the test data set. By using evaluation methods that we have adjusted to this special problem, we have shown that Markov Clustering is a very good method for finding new objects from tracklets, even if the foregoing tracklet-tracklet correlation contains many wrongly associated pairs and the clustered data varies a lot in its characteristics.

In a next step it would be interesting to investigate if other graph-based clustering methods lead to better results. Especially agglomerative clustering methods might be interesting to gain more insights. Furthermore, other weight functions for the edges might be useful, e.g., those that are dependent on the loss value, because lower loss values might be an indication for a better fitting. Additionally, applying a hyperparameter optimization might help to improve the results even more. Moreover, it would be interesting to analyze how the parameters of the graph, e.g., number of edges and connected components, influence the clustering result.

Taking all of the above into account, we will

- perform hyperparameter optimization,
- try different clustering algorithms, especially agglomerative ones,
- try different weight functions, and
- analyze what influences the clustering results.

Acknowledgements

The authors thank all former and current BACARDI team members, as well as the members of the DLR Departments SC-IVS, SC-HPC, and the SSA group within RB-RFT for fruitful discussions and support.

References

[1] ESA Space Debris Office. *ESA'S ANNUAL SPACE ENVIRONMENT REPORT*. LOG 6.0. GEN-DB-LOG-00288-OPS-SD. Robert-Bosch-Strasse 5, D-64293 Darmstadt, Germany: ESA UNCLASSIFIED - Releasable to the Public ESA ESOC, Apr. 2022.

[2] Martin Stoffers et al. "BACARDI: A System to track Space Debris". In: *ESA NEO and DEBRIS DETECTION CONFERENCE - EXPLOITING SYNERGIES* -. Feb. 2019. URL: https://elib.dlr.de/126572/.

[3] Carolin Fruh et al. "Catalogue correlation of space debris objects". In: *Fifth European Conference on Space Debris*. Volume 672. 2009, page 8.

[4] J.A. Siminski et al. "Short-arc tracklet association for geostationary objects". In: *Advances in Space Research* 53.8 (2014), pages 1184–1194. DOI: https://doi.org/10.1016/j.asr.2014.01.017.

[5] Carlos Yanez et al. "Optical measurements association using Optimized Boundary Value IOD coupled with Markov Clustering algorithm". In: June 2017.

[6] Anton Enright, S. Dongen, and C.A. Ouzounis. "An efficient algorithm for large-scale detection of protein families". In: *Nucleic acids research* 30 (May 2002), pages 1575–84. DOI: 10.1093/nar/30.7.1575.

[7] James Vlasblom and Shoshana J Wodak. "Markov clustering versus affinity propagation for the partitioning of protein interaction graphs". In: *BMC bioinformatics* 10.1 (2009), pages 1–14.

[8] L. Stothert, P. Norberg, and C.M. Baugh. "A new approach to finding galaxy groups using Markov Clustering". In: *Monthly Notices of the Royal Astronomical Society: Letters* 485.1 (2019), pages L126–L130.

[9] Hauke Fiedler et al. "SMARTnet - Evolution and Results". In: *IAC Bremen*. 2018. URL: https://elib.dlr.de/123984/.

[10] RC Blanchard and ER Lancaster. *A unified form of Lambert's theorem*. 1969.

[11] Dario Izzo. "Revisiting Lambert's problem". In: *Celestial Mechanics and Dynamical Astronomy* 121.1 (2015), pages 1–15.

[12] Stijn Dongen. "Graph Clustering by Flow Simulation". In: *PhD thesis, Center for Math and Computer Science (CWI)* (May 2000).

[13] Andrea Lancichinetti and Santo Fortunato. "Community Detection Algorithms: A Comparative Analysis". In: *Physical review. E, Statistical, nonlinear, and soft matter physics* 80 (Nov. 2009), page 056117. DOI: 10.1103/PhysRevE.80.056117.

[14] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. "Exploring Network Structure, Dynamics, and Function using NetworkX". In: *Proceedings of the 7th Python in Science Conference*. Edited by Gaël Varoquaux, Travis Vaught, and Jarrod Millman. Pasadena, CA USA, 2008, pages 11–15.

[15] Guy Allard, Mounir Mallak, and Jona Harris. *Markov Clustering*. Dec. 2018. URL: https://github.com/GuyAllard/markov_clustering (visited on 07/15/2022).

[16] C. J. van Rijsbergen. "Information Retrieval". In: 2nd. 1979.

[17] scikit-learn developers. *2.3. Clustering*. URL: https://scikit-learn.org/stable/modules/clustering.html (visited on 07/15/2022).

[18] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pages 2825–2830.

[19] William Rand. "Objective Criteria for the Evaluation of Clustering Methods". In: *Journal of the American Statistical Association* 66 (Dec. 1971), pages 846–850. DOI: 10.2307/2284239.

[20] Nguyen Vinh, Julien Epps, and James Bailey. "Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance". In: *Journal of Machine Learning Research* 11 (Oct. 2010), pages 2837–2854.

[21] Douglas Steinley. "Properties of the Hubert-Arabie Adjusted Rand Index". In: *Psychological methods* 9 (Sept. 2004), pages 386–96. DOI: 10.1037/1082-989X.9.3.386.

[22] Lawrence Hubert and Phipps Arabie. "Comparing partitions". In: *Journal of classification* 2.1 (1985), pages 193–218.

[23] Marina Meilă. "Comparing clusterings—an information based distance". In: *Journal of Multivariate Analysis* 98.5 (2007), pages 873–895. ISSN: 0047-259X. DOI: https://doi.org/10.1016/j.jmva.2006.11.013.

[24] R. Baeza-Yates and Berthier Ribeiro-Neto. "Modern Information Retrieval". In: 2011, pages 327–328.