Proceedings of the Annual General Donald R. Keith Memorial Conference
West Point, New York, USA
April 29, 2021
A Regional Conference of the Society for Industrial and Systems Engineering

Co-Travel Algorithm Development: Initial Steps

Aaron Jones, Adam McElligott, Bonnie Romeo, Eric Whiteman, and Erin Williams

United States Military Academy
Department of Systems Engineering
West Point, NY

Corresponding author's Email: erin.williams@westpoint.edu

Author Note: Cadets Jones, McElligott, Romeo and Whiteman are Cadets at the United States Military Academy. The Cadets completed this project under supervision of advisor MAJ Erin Williams, an instructor in the Department of Systems Engineering. The USMA Cadet-led Data Science team worked in partnership with Accenture, a leading innovation company, working towards evolving intelligence capabilities in certain contracted military capacities.

Abstract: The co-travel algorithm seeks to utilize Open-Source Intelligence (OSINT) to systematically compute and identify high value targets using advertising technology data sets. The OSINT discipline is a quickly evolving type of intelligence that could grant a significant advantage to ground combatants. Co-travel refers to devices moving together. It can establish critical relationships between individuals. The co-travel algorithm can be separated into four main functions: pattern of life, anomaly detection, device correlation, and co-traveler. These functions will assist the algorithm in enabling Army OSINT Enterprise to ultimately track persons of interest and uncover locations of interest. The Cadet Data Science Team uses k-means clustering and density-based spatial clustering application with noise (DBSCAN) to establish a pattern of life and detecting anomalies within an OSINT data set. The team also produces a functional R Shiny application to enable intelligence analysts to quickly analyze geotemporal dataset and visualize results. While the full Co-Travel algorithm remains unfinished, this report details the progress that was made by the Cadet Data Science team over the course of an academic year.

Keywords: Open-Source Intelligence, Pattern of Life, Anomaly Detection, Device Correlation, Co-Travel Algorithm

1. Introduction

The Cadet Data Science team has been working for U.S. Army Intelligence and Security Command (INSCOM) in conjunction with a contracted team from Accenture. INSCOM executes mission command of operational intelligence and security forces; conducts, synchronizes, and integrates worldwide multi-discipline and all-source intelligence and security operations; and delivers linguist support and intelligence related advanced skills training, acquisition support, logistics, communications, and other specialized capabilities in support of Army, Joint, and Coalition commands and the U.S. Intelligence Community (INSCOM, 2020). The Cadet Data Science team, with assistance and oversight from Accenture, has provided INSCOM with a tool that will enhance mission accomplishment.

The Cadet Data Science team has researched and worked with Accenture in developing OSINT tools, powered by traditional components of a co-travel algorithm. A co-travel algorithm computes the date, time, and location of devices over a window of space and time, and then looks for other devices that were seen in the same window.

When broken down its main components, the co-travel algorithm consists of establishing pattern of life, anomaly detection, device correlation, and the co-traveler. Establishing pattern of life is the tracking of high value target persons to the intelligence community. Anomaly detection distinguishes outliers in population behavior to assess threats or items of interest. Device correlation identifies possible relationships and networks between devices in each population. In this instance, device correlation ranges in the space and time for whether the devices that crosses paths are considered a correlation. Finally, the co-traveler identifies two or more devices with similar correlations that have been seen at significantly different locations within a given space and time, to be assessed as different devices belonging to one individual.

While the final end-state is to have a fully functioning co-travel algorithm, time constraints limited the availability of the Cadet Data Science Team and the partnership with Accenture. This report will discuss establishing pattern of life and anomaly detection, as they pertain the development of a co-travel algorithm. Put into application, these two tools will be utilized by end-users, American Soldiers, to analyze and exploit OSIF.

2. Background

OSINT is intelligence produced from publicly available information (Section 931, Public Law 109-163). OSINT is derived from Open-Source Information (OSIF), OSIF includes data that can be put together, sourced from widely disseminated information. This may include newspapers, books, and radio broadcasts, among other sources. In short, OSIF is publicly available unclassified information, while OSINT is data by way of applying, processing, and exploiting the unclassified information to validate information as relevant, accurate, and actionable for use by the consumers (Williams & Blum, 2018).

In this case, the data being analyzed is geospatial and temporal in nature. In order to successfully evaluate this data, the Cadet Data Science team focused on spatiotemporal analysis techniques such as trajectory mining. Trajectory mining aims to observe and synthesize location data over time in a manner that can be interpreted into a pattern of life for the observed entity. Trajectory data itself is the observed multidimensional data that is derived from movement, or the lack thereof, of the subject. Trajectory data mining is the mass collection and refinement of that data (Zheng, 2015). Trajectory mining uses clustering techniques to show indications of patterns, but its primary objective is deriving speed of travel, direction of travel, and time spent traveling.

Trajectory mining relies on multidimensional data analysis, which is the manipulation and analysis of different levels of dimensional objects where data is organized into meaningful hierarchies. Multidimensional data analysis enables data clustering. This divides data into various groups, or clusters, for the purposes of summarization or gaining a better understanding of a data set. Several clustering methods were evaluated, and k-means clustering, and density-based spatial clustering of application with noise (DBSCAN) were identified as most useful.

K-means clustering is an incremental approach to grouping data in which the number of clusters is varied, by a local user, to offer a deterministic, optimized solution (Likas, 2006). The algorithm partitions the data into *k* clusters with a specific cluster centroid, or mean. The approach aims to minimize the Euclidean distance from the cluster centroid to each data point (Tarpey, 2007). Once the Euclidean distances have been minimized, the resulting *k* clusters represent groups of associated data. The number of clusters is manually assigned, although a range of values can be tested. Advantages of k-means clustering include the avoidance of errors within the clustering groups populated, which can link grouping errors to a dataset. Disadvantages of k-means clustering include the potential impacts that outlier data points have on the clustering of the remainder of the data set.

DBSCAN is an unsupervised learning method that attempts to identify a larger picture, as it pertains to the structure of the data. DBSCAN is particularly effective for analysis where clusters tend to be of arbitrary shapes and is more efficient at detecting outliers than k-means clustering. DBSCAN looks for relationships between raw data points to outline arbitrary clusters and clusters with noise, outliers (Yıldırım, 2020). DBSCAN also has the advantage of not requiring you to list the number of clusters to use it. It uses a function to measure the difference between values, as well as some instructions on what constitutes a "near" distance to run effectively. Figure 1 shows the difference in output of both DBSCAN and k-means clustering, at DBSCANs clusters run across several different distributions yielding more rational outcomes.

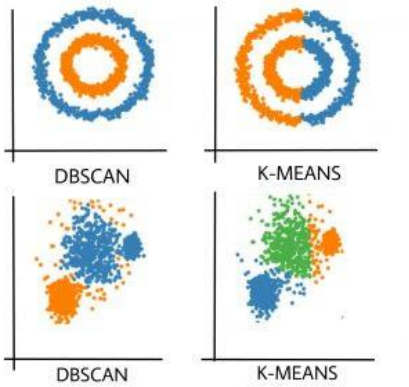

Figure 1: DSCAN Analysis (Mamidi, 2019)

ISBN: 97819384962-0-2

3. Methodology

3.1 Process

The creation of co-travel algorithm requires complex analysis relying on manipulation of high quantity data sets. Typically, algorithms are trained on data that is already classified. The data comes with some sort of classification tag, which the algorithm utilizes to determine the best parameters to use in model creation. The model is initially created with a training set of data. Then, the model is run on a test set of data and scored for how accurately it predicts the value or class of that test data. In this case, the Cadet Data Science team is at the forefront of analyzing and testing the effectiveness of a co-travel methods and therefore, lacked a training and test data set to verify the efficiency. Traditional methods of intelligence analysis can be validated through peer reviews and replication abilities to prove usefulness as an intelligence capability. Although the primary methods used are not methodical to traditional intelligence analysis, they are comprehensively studied and trusted within the mathematical analysis field of study. Thus, the algorithms should provide an accurate means of analyzing location data density and distribution.

To formulate the best algorithm, the Cadet Data Science team employed several different types of clustering techniques to assist in data interpretation and manipulation. Trajectory mining was used as a basis for identification of different clustering techniques to identify algorithms that would works best with a high-dimensional data. Research was focused on basic cluster analysis, clustering, principal component analysis (PCA), density based spatial clustering of applications with noise (DBSCAN), and autoencoders. Ultimately, k-means clustering and DBSCAN were selected for algorithm development because initial research indicated that the fundamental patterns and summarization of the data set demanded cluster analysis to assist in the identification of persons of interest, rather than methods of reducing dimensionality. These methods will be discussed in depth, relative to the timeline of application within the project.

3.2 Data Cleaning

In a data set, the information supplied must be manipulated before it can be analyzed. This ensures the high-quality information moving forward. The data set for this project was provided by Accenture. The data set contained many functional issues which initially prohibited algorithm development. The first step in creating useable data was removing the unusable rows and columns. To be classified as an unusable, a column was missing more than twenty-five percent of the associated input values. By removing these black data entries, memory is only allocated towards useful information This is critical to this project given that the high quantity of rows in the set varies and greatly changes the computational runtime. Data was classified as useable when the row input values were greater than seventy-five percent complete, contained information vital to analyzing the geolocation and timeline of the co-traveler, or was a unique device identification that could be used to link potential persons of interest to corresponding devices.

The final steps were focused about ensuring that the data remaining was useable in a coding language, ensuring that all the data was formatted in line with standard practice. This included the removal of additional letters and symbols in latitude and longitude values as well as the removal of duplicate data entries within the rows of the data. The data also contained time and date information regarding each device entry, relative to the original time zone that the device was utilized in. To mitigate inconsistencies across time, all time zone information was converted to coordinated universal time. After the dataset was thoroughly cleaned, the Cadet Data Science team was able to move forward with manipulation and algorithm development.

3.3 K-Means Clustering

The Cadet Data Science team first approached clustering analysis through the application of a k-means clustering algorithm. Figure 1 showcases the process of k-means clustering. First, the algorithm splits the data points into groups determined by location. Next, the groups are further broken down into groups based on location. The inability to cluster data of varying quantities and the algorithm's dependence on initial values means that the algorithm may fail to adequately adjust the clusters when varied by a local user.

When applied by the Cadet Data Science team, k-means clustering was determined to be least effective given an inability to work with high quantity data. When used on the provided dataset, k-means clustering struggled to cluster data points efficiently, resulting in many larger identifiable areas. It was apparent that when applied higher quality data, when the data explicitly lacked points that exhibited a circular pattern, indicating that k-means clustering struggled to operate to its full

capacity. Given the initial results, the Cadet Data Science team worked to investigate other clustering techniques. The full analysis of this clustering technique, relative to others, will be discussed in Section 4.

3.4 Density Based Spatial Clustering of Applications with Noise (DBSCAN)

DBSCAN is a clustering algorithm for grouping similar looking data points. DBSCAN begins by choosing a point, at random, from the collection of data points. The chosen point is tested to see if it is a core point. A point is a core point if it includes at least the number of minimum points in its epsilon-neighborhood to form a dense cluster. Next, DBSCAN works to find any associated components from the core points while disregarding non-core points. The non-core points are assigned to the nearest cluster in its epsilon-neighborhood and are considered border points. The points that remain unassigned to a core point are considered noise. The algorithm uses DBSCAN to analyze and separate data by its specific user identification number and then create the arbitrary clusters for each device separately. This will be particularly useful for developing the device correlation and co-traveler algorithms in future phases of the project.

4. Model Results

Application of DBSCAN methodology enables the discovery of definitive clusters of mobile device user location and activities. The generated clusters have assisted in the identification of pattern of life and predicting the locations users are operating in. When comparing k-means clusters with DBSCAN, it is evident that the latter method is more reliable in producing clusters that are more likely to be accurate representations of frequented movements. This is due to the nature of each method, with k-means requiring a set number of clusters to find, as opposed to DBSCAN utilizing input data to identify its own number of clusters. Additionally, DBSCAN has higher affinity for noisy data, while k-means clustering is sensitive to noise and outlier data points because a small number of such data can substantially influence the mean value (Chakraborty, 2014). For the type of data that was being analyzed, extremely dense points within cities, the decision to proceed with an algorithm that was better equipped to handle this type of data was made.

K-means clustering allows the user to identify a centroid for each cluster that is established, afterwards establishing a Euclidian distance from the cluster centroid to each point around its respective cluster. This process helps to determine points that are “least” likely to belong to that clusters by finding points with the greatest distance to their cluster centroid. Then the process will enable the ranking of points, based off the observed Euclidian distance by user or location, yielding insight to potential points of concern. The Euclidian distance metric primarily used in k-means clustering. Whereas a DBSCN methodology runs into issues given the lack of convexity in some clusters established. It may be necessary to use a blend of methodologies to analyze the data, first looking at the DBSCAN to identify clusters themselves and determine the number of said clusters, then utilize k-means clustering output to verify these clusters and its associated points of interest based upon the Euclidian distance metric from k-means clustering.

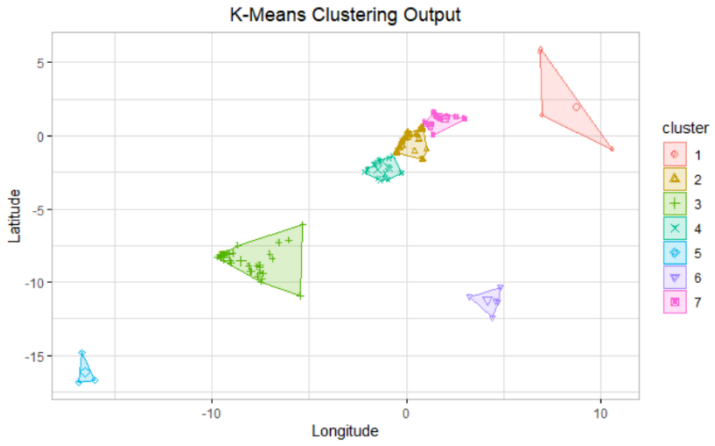

Figure 2: K-Means Clustering

Figure 2 depicts the output of k-means clustering of a subset of the full data frame, specifically points that fall within the Dubai Time Zone. The number of clusters chosen for this iteration of the k-means clustering analysis was seven, in order to have an identical number of clusters as the DBSCAN output, shown below. In this iteration of the clustering, all points were grouped within a cluster, showing no outliers predicted.

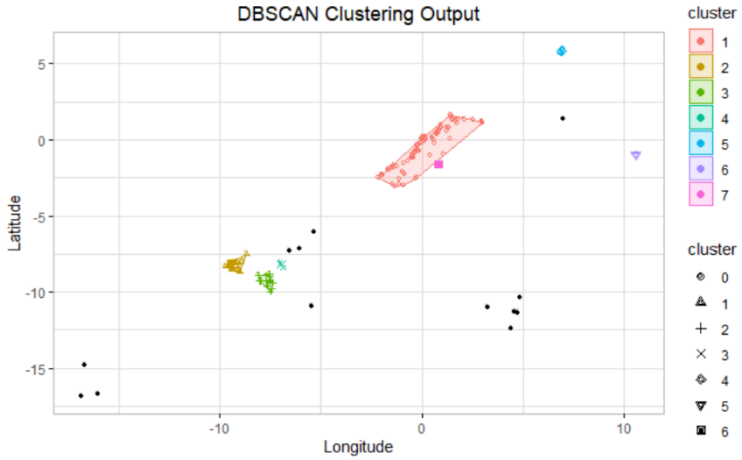

Figure 3: DBSCAN Plot

Figure 3 depicts the DBSCAN output run on the same subset of data points, Dubai Time Zone, and the corresponding results. In this scenario, the algorithm resulted in seven clusters of varying size, shape, and density. In comparison to the k-means clustering output, the clusters identified are different, despite being run on the same dataset. Also different from the k-means clustering output, the DBSCAN shows many more outliers (14) than k-means clustering (0). Examining the chart, it appears that DBSCAN is much more selective in which points it chooses to include in clusters, favoring those that are very tightly grouped, as opposed to points that are not as tightly group, but rather are just in the same general vicinity.

When employing a joint, k-means clustering DBSCAN, methodology the Cadet Data Science team was successful in identifying outlying points of interest. However, a lack of initial training data makes it difficult to gauge long term success in extrapolated application.

5. User Interface, R Shiny Application

The Cadet Data Science team created a user interface that allows end users, Army Soldiers, to access the algorithm readily and remotely through an internet-based application. The ability to access clustering operations within the algorithm will encourage the rapid analysis of information and report findings regarding a particular dataset. This is a task that would previously take days, if attempted to complete on a localized, individual level whereas the application can accomplish the analysis in minutes. The user interface application was developed using the Shiny package for R, which allows code to be processed into a browser-based application, using the code as a method of analyzing any user specific input. The application allows local users to upload a comma-separated values data file, first cleaning the data. This step is dependent on what the local user chooses to include or omit from the data set, varying the production of the clustering algorithms. Given the output of clusters, the application can display the corresponding clustered data points overlayed on a map of the region in question. Through the graphic features and resulting data, the identification of potential high target individuals. The main features of the application include the ability to view location of device identifiers, view clusters as associated with device identifiers, and overlay clusters on a map to detect co-travel patterns. The application has been presented to the Accenture team and INSCOM Deputy Commander and after security clearance specific modification will be fully deployed for INSCOM mission capabilities.

6. Future Work

In order to expand on the original goals of this project, the Cadet Data Science teams has identified potential courses of action that may lead to additional extrapolation of the co-traveler algorithm capabilities. Once the co-traveler is fully developed, integration with additional datasets, expanded clustering analysis, and further integration of the algorithm into current user interface, shiny application, are critical to expanding the scope of the algorithm. While the initial model was built and analyzed referencing moderately sized data set, it is likely that future datasets must accommodate millions of observations, rather than thousands. It is important to ensure that the framework of the algorithm is robust, enabling the computational ability of such a large amount of information. It is necessary that the end user can fully understand the processing power needed to run the algorithm to best identify the relationship between data size and power, breaking down the data as needed. Finally, while the study primarily focused on the implementation of DBSCAN to achieve cluster identification, there are many other methods of clustering that may be able to achieve a similar level of clustering. Alternate techniques such as partitioning around medoids, gaussian mixture models, and spectral clustering can also identify clusters; with each methodology having various benefits and drawbacks associated. Additional research may show a method that proves to be more precise in cluster identification, yielding better identification of potential high target persons.

7. References

Chakraborty, S., N. K. Nagwani and Lopamudra Dey. "Performance Comparison of Incremental K-means and Incremental DBSCAN Algorithms." *ArXiv* abs/1406.4751 (2014): n. page 14.

Likas, A., Vlassis, N., & Verbeek, J. J. (2002, March 23). The Global K-Means Clustering Algorithm. *Pergamon: The Journal of the Pattern Recognition Society*, 451-461. doi: S0031

Mamidi, A. (2019, November 17). DBSCAN -Density-Based Spatial Clustering of Application with Noise. Retrieved March 16, 2021, from https://www.abhishekmamidi.com/2019/11/dbscan-density-based-spatial-clustering-of-applications-with-noise.html

Public Law 109-163. (2006). Retrieved October 13, 2020, from htttps://www.govinfo.gov/app/details/PLAW-109publ163

Tarpey, T. (2007). Linear Transformations and the k-Means Clustering Algorithm. *The American Statistician, 61*(1), 34-40. doi:10.1198/000313007x171016

U.S. Army Intelligence & Security Command. INSCOM. (2020). https://www.inscom.army.mil/.

Williams, H., & Blum, I. (2018, May 17). Defining Second Generation Open Source Intelligence (OSINT) for the Defense Enterprise. Retrieved October 05, 2020, from https://www.rand.org/pubs/research_reports/RR1964.html

Yıldırım, S. (2020, April 22). DBSCAN Clustering Explained. Retrieved February 18, 2021, from https://towardsdatascience.com/dbscan-clustering-explained-97556a2ad556

Yu Zheng. 2015. Trajectory data mining: An overview. ACM Trans. Intell. Syst. Technol. 6, 3, Article 29 (May 2015), 41 pages. doi:10.1145/2743025