Automating Clustering Analysis of Ivory Coast Mobile Phone Data

Deriving Decision Support Models for Community Detection and Sensemaking

Thomas J. Klemas
MIT Lincoln Laboratory
tklemas@alum.mit.edu

Steve Chan
Network Science Research Centre
Swansea University
stevechan@post.harvard.edu

Abstract—**Sensemaking involves numerous levels of processing and logic in order to achieve automated decision support. Many of these concepts derive from the realm of pattern recognition. The data under consideration frequently is observed in a noisy environment and so one of the first steps involves preprocessing the data to suppress noise and isolate the data signal. Patterns within the data are often used to improve signal detection and aid identification of the data in the quest to produce actionable information. A critical step of making sense from raw or partially processed data and other aspects of decision support is to organize information, which frequently involves grouping, partitioning, or clustering objects. However, there is typically an assumption that structure exists within the data, and the number of clusters is a required parameter for many of the clustering algorithms. A common approach to determine the best number of clusters is to iterate across a set of potential values the for number of clusters and evaluate the quality of the resulting clusters using some metric. In this paper, we present an automated approach to detect structure and improve automation of clustering algorithm parameters. We apply our approach to analyze a complex, dynamic multiple edge set network that was used to model call data from the Ivory Coast compiled from France Telecom/Orange anonymized call records over a 5 month period.**

Keywords - Sensemaking; adaptive clustering; spectral clustering; network theory; silhouette; k-means; unsupervised; partitioning; proximity measure; similarity measure; decision support; iterative; randomized singular value decomposition.

I. Introduction

When data sets are extremely large, very complex, or the data is changing rapidly, analysis requirements reach a level beyond which humans are unable to consider the full scope of the data and lack the capacity to keep up, derive insights, and make decisions. In today's world of increasingly smart and interconnected systems, more and more sensors are deployed in civilian, medical, industrial, and military systems and these systems are frequently networked in some manner to allow programming (automation), monitoring, and remote control, to facilitate software updates, to enable interactivity, and related objectives. Accompanying the rapid rise of networked sensors is a flood of available new data. However, to maximize the value of this data it is critical to attach appropriate labels, meta data, and links that enable combining and synchronization of this data with other suitable data for analysis.

The ultimate objective for Sensemaking technologies is to make sense of the raw data. Automated decision support and Sensemaking tools apply machine learning, pattern recognition, expert logic, and other algorithms in order to detect and identify patterns in the mass of data that contain information that supports and enables decisions. Typically, there are many steps required before one is able discern actionable information from the raw data. These steps may include numerous preprocessing routines that may eliminate data outliers that have the potential to distort decision making algorithms, normalize the data to improve sensitivity, inserting values for missing data items, and similar manipulations to “clean up” the data in preparation for subsequent mainstream processing stages. Depending on the specific methods to be used, feature vectors will be computed from the raw data and metrics will be calculated from the feature vectors to support various classification decisions.

Our objective is targeted to detect and identify important but non-evident structural groupings, develop insights based on the structure to resolve community clusters. In this paper, as in the previous paper [5], we focus on pattern recognition algorithms that provide a mechanism for grouping objects detected in the data channel based on features vectors or measurable quantities of interest that are selected to help distinguish different objects. When training data is available, the grouping of objects is often called partitioning. When no training data is available, grouping of objects is termed clustering. Classical methods for clustering include *k*-means, spectral, Kerninghan-Lin, and other algorithms. A variety of proximity measures can be used to determine whether data points and corresponding objects share either similarity or dissimilarity, based on feature vector values in 1 or more dimensions. Examples include Euclidean distance, silhouette values, Pearson coefficients, Saltine’s cosines, or other proximity measures [1]. In particular, we will examine unsupervised clustering techniques that group data without the benefit of training data containing truth information.

This research will present and explore performance of an automated approach to detect if structure is present in the data and also to select a number of clusters and cluster objects from new, unknown data sets. If no data is present within a data set the various clustering algorithms can produce unusual, potentially nonsensical results. We adapted methods, based on spectral decomposition, to achieve clustering in a multiple edge set network that we generate to model the Ivory Coast France Telecom/Orange call records.

In our previous work [5], we observed that the evolutionary approach that we adopted to model the drift of parameters of the associated proximity measures required either careful selection of parameters [2] [3] [6] or iterative solves to choose a parameter value, such as number of clusters. Related to this issue, it is important to determine that structure exists before applying clustering algorithms or risk nonsensical results when attempting to interpret the results of clustering analysis. Additionally, we observed that solves were computationally intensive, so in this work we explore an approach to automate detect structure and improve parameter selection. Also in this paper, although it is obviously not the primary focus of this work, we illustrate the steps involved to apply randomized [4] hybrid methods to accelerate clustering algorithms in our problem space. This sort of computational efficiency becomes increasingly important especially when contemplating community detection in much larger countries or regions of the world.

The remainder of this manuscript is arranged as described herein. Section II describes the technical details of our clustering algorithms and how they accomplish analysis of a multiple edge set network. Section III provides a brief description of the data set and also outlines how the key data elements are aligned as inputs to the analysis. Section IV described the performance and provides results of applying our automated clustering approach to the multiple edge set network data modeled from the Ivory Coast France Telecom/Orange call records. Section V offers our conclusions. Finally, the acknowledgment and reference sections complete the manuscript.

II. Technical Details

We start by reviewing the fundamental clustering methods and the notations that we will be using throughout the manuscript. First of all, we will model sub-prefectures in which our callers access cell towers to make and receive calls as nodes and the call records between 2 sub-prefectures as result of a call between 2 callers (one in each sub-prefecture), as an edge. In our case, we also were able to construct travelers from the call data as we observed callers that switched cell towers and even sub-prefectures as they traveled by car, bus, train, or airplane. Thus, our nodes have traveler edge connections between them as well, as another type of edge set interconnecting our social network.

$$G = (V, E_1, E_2) \qquad (1)$$

In the graph G, the V nodes represent sub-prefectures, the E_1 edges represent calls between sub-prefectures, and the E_2 edges represent travelers between sub-prefectures Furthermore, for completeness, the additional induced graph relating the various cell towers and sub-prefecture centers by geographical distance should really be incorporated into this graph as well, but for now we ignore this layer. For the sake of simplicity, we will represent the travel with an undirected edges connecting the graph model. Since the callers use cell towers that are distributed geographically within sub-prefectures of the Ivory Coast, our algorithms incorporate a mapping layer to translate between cell towers and sub-prefectures As our goal is to detect hidden structure withing the call data that may correspond to communities, our notation provides corresponding terms. The term S_i defines a community or cluster of nodes, in this case sub-prefectures, that are disjoint to all other communities. Thus, a vertex can exist in only one community.

$$V = \bigcup S_i,\ \forall\ i,j,i\neq j,\ S_i \cap \mathrm{S}_{\mathrm{j}} = \varnothing \qquad (2)$$

Following well established methods [1], by selecting a feature vector, f_a, in this case the accumulated calls between sub-prefecture a and every other sub-prefecture, and choosing a proximity measure, in this case the euclidean distance between two feature vectors, f_a and f_b, we can utilize this dissimilarity metric to compare two sub-prefectures on the basis of the associated call records. Furthermore, extending this concept to the entire set of sub-prefectures, we can applying a variety of pattern recognition and network science techniques to attempt to cluster the sub-prefectures based on the call records, as well. In this research, we employed several clustering approaches, derived from k-means, spectral decomposition, and aggregation, and developed modifications aimed to enable improved automation, improve computational efficiency, improved ease of implementation, and facilitate comparison study of clustering performance,

First, we briefly review these approaches to augment our notation prior to modification and enhancement of the algorithms. The classical k-means algorithm [1] requires a parameter, k, which specifies the number of clusters into which the objects, in our case sub-prefectures, should be grouped. Then the algorithm, randomly selects k centroids in the space in which feature vectors reside. The objects are then clustered into the cluster with the nearest centroid using the similarity measure and centroids are recomputed. This process continues iteratively until the cluster centroids converge or cease to change.

To effectively utilize k-means and the other algorithms to cluster the caller records, based on selected features, as an precursor aid to facilitate community detection, it is typical to iteratively solve for a new clustering of the system and determine the best number of clusters based on a suitable metric. In our previous paper, we adopted silhouette values [8] as such a metric to facilitate choice of the number of clusters. The silhouette value concept was constructed to characterize the degree of community structure that is present in a clustering induced from a set of interrelated objects, such as the sub-prefectures of the Ivory Coast. Briefly reviewing the mechanics of this approach, the silhouette function is defined as:

$$silhouette(i) = (b(i) - a(i))/max(a(i),b(i)) \qquad (3)$$

the value *a(i)* represents the intra-cluster dissimilarity of sub-prefecture *i* or, in other words, the mean value of a chosen dissimilarity measure for the sub-prefecture *i* with respect to the other sub-prefectures that are members of the same cluster. The value *b(i)* represents the smallest average dissimilarity between sub-prefecture *i* and the clusters of which it is not a member. For this research, we adopt euclidean distance between two feature vectors as the proximity metric, in this case a measure of dissimilarity between the two nodes. A silhouette value is assigned to the entire clustering, as well,

$$silhouette(k) = mean_i(silhouette(i)) \qquad (4)$$

which is simply the mean value of the silhouette values of each node or sub-prefecture Using these definitions, the silhouette values will vary between 1, indicating high degree of community structure and -1, which suggests the absence of community structure.

Next, we describe how clustering can be accomplished using classical spectral methods for graph decomposition. The adjacency matrix, **A**, indicates a measure of the amount and duration of calls that were exchanged between each pair of particular sub-prefectures, forming connections between the corresponding nodes in the graph where inter-sub-prefecture calling was recorded in the data set. If the call pairing vectors that arise from the columns of the adjacency matrix are compared with a "proximity" measure (in particular a similarity measure) then it is possible to determine the extent to which nodes share common connectivity patterns. Thus, illustrating this concept, the similarity matrix, **W**, which is the target for the spectral decomposition, is computed from the adjacency matrix simply as the inner product of the column vectors forming the adjacency matrix,

$$W_{ij} = a_i^T a_j \qquad (5)$$

Thus, the entries indicate similarity between columns of the adjacency matrix and highlights node pairs with similar connectivity patterns. Traditional spectral clustering methods involves computing the singular value decomposition (SVD) of a matrix related to the similarity matrix, such as the Laplacian matrix,

$$L = D - A \qquad (6)$$

and the matrix is decomposed as described in equation (7).

$$L = U \Sigma V^T \qquad (7)$$

There are schemes that determine clusters based on the eigenvectors corresponding to largest or smallest eigenvalues of the Laplacian matrix and related systems. However, we have selected a technique described in [7] in which the first K eigenvectors are retained $\left[U_{1,} U_{2,} \dots, U_K\right]$ and then the rows of the retained K eigenvectors are partitioned in K clusters using the k-means algorithm. Since K is a parameter that needs to be selected, the silhouette values can be used to select a suitable value of K. However, this approach requires repeated iteration to compute the clusters that correspond to each value of K and determine the resulting silhouette values for each clustering. The clusterings are compared by silhouette values to determine the optimal number of clusters.

In this paper, we describe results from a method that we used to improve the overall approach automation by obviating the need for extensive clustering iterations to determine the best number of clusters, k. While many metrics can be used to select the best number of clusters, the silhouette metric [8], described previously, was used in this research, because its definition best captured our goals for clustering. As one might surmise, since we are using a hybrid spectral algorithm to accomplish clustering, the basis for selecting the k parameter derives from analysis of the larger singular values associated with the singular vectors that contribute the most towards defining the nodes sharing the most similar communication patterns in the graph subspace.

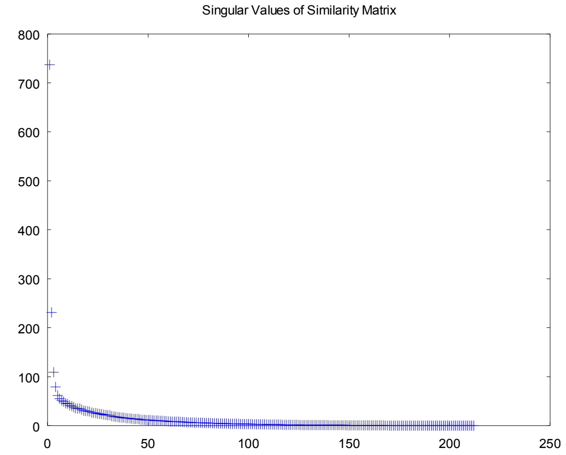

Figure 1: Singular Values for 255 by 255 Similarity Matrix indicate structure exists within sub-prefecture caller communication records.

We have developed a dual filtering analysis engine that attempts to detect the last large gap in the singular values and then determine the corresponding index of the singular value preceding the identified gap, in order to select this index value as an upper bound on the k parameter, thereby significantly reducing the number of iterations to kth clustering and the few clusterings preceding it. Silhouette values or a similar metric can be used to choose the optimal value among the few that are explored. We note that this singular value gap analysis approach, based on the described dual-filtering method, requires a bounding parameter for the maximum number of clusters expected, but this does not necessitate additional clustering iterations and we expect that in many application, as in ours, this required parameter does not decrease utility of the method relative to the primary objectives. In testing, as we will show subsequently in the results section, the clustering decisions seem to compare favorably with the typical approach based on the iterative computation of cluster quality metrics, such as silhouette values.

Finally, to reduce the cost of computing the SVD, it is possible to use a randomized technique to decrease the overall computations. The technique we selected and implemented, [4], involves stimulation of the system input by multiplying the similarity matrix by a random matrix, Θ , with a reduced number of columns to elicit the corresponding output matrix, as described by $W*\Theta=Y$. Then, by computing an orthogonal decomposition or QR factorization of the output matrix, $Y=Q*R$, and multiplying the original matrix by Hermitian of Q, $C=Q^H*W$. Computing a reduced subset of the SVD of C $C\simeq\tilde{U}*\tilde{\Sigma}*\tilde{V}^H$, is much less expensive, and we can approximate A as $W\simeq Q*Q^H*W$. Thus, the singular value decomposition of A can also be approximated $W\simeq Q*\tilde{U}*\tilde{\Sigma}*\tilde{V}^H$ and $U\simeq Q*\tilde{U}$. The overall cost of this SVD is significantly less expensive, $O(N^2)$, and by using a lower cost SVD, spectral clustering methods are significantly more feasible for numerous applications with extremely large numbers of nodes.

An added improvement to spectral clustering methods, developed in [7], is to use K-means to cluster the row space of the singular vectors U. The theory behind this approach is developed nicely in that paper. We incorporated the same technique into our approach in order to compute spectral clustering, and we explore its performance relative to the traditional approaches.

III. Application Space

The cell tower call record data used in this research is described in the document prepared by Blondel and Esch et al [9]. The records consist of several categories of differing types. Throughout this research, we primarily analyzed the first and second subsets within the Data for Development (D4D) data record collection. In the first subset of call data records, cell tower to cell tower connections were accumulated for each hour, including frequency and duration attributes for each pairing. The subsequent subset of data was comprised of sub-prefecture indexing information for a random sample of 500,000 individual callers (as opposed to pairs) over a limited 2 week period. Finally, the third and last subset of the data contained call records for a smaller number of 50,000 individual records that endured over the entire 5 month D4D data collection period. Our research focused on the the second subset as the source of traveler information. Additional data files included information that specified center locations of sub-prefectures and locations for antennas. With this auxiliary data it is possible to map antennas to closest center locations for sub-prefectures, and, coupled with the file registering the locations of the sub-prefecture centers, the combination enables graphical result data plots revealing geographical trends.

IV. Results

To evaluate the efficacy of our algorithms we developed feature vectors that captured the content of sub-prefecture call records between February 7, 2012 and Feb 14, 2012. Our feature vectors comprised the cell tower connection data between callers that was recorded during this period and is mapped to corresponding sub-prefectures to which they belong geographically, of the 255 total sub-prefectures Thus, with this approach, our feature vector is 255 element vector, γ_m, in which $\gamma_m(n)$ is the complete extent of calls between a pair of sub-prefectures indexed by m and n accumulated over the recording duration. Next, this sub-prefecture connection data is accumulated between each pair of sub-prefectures by stacking the columns γ_m to generate an adjacency matrix with elements γ_{mn}, representing the degree of connectivity between each of the associated pairs of sub-prefectures The adjacency matrix contains 255 rows and 255 columns.

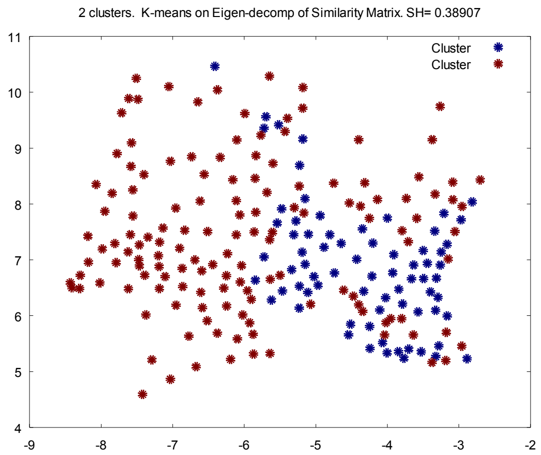

Figure 2: Clustering into 2 groups by applying K-means algorithm to similarity matrix

Then, in the final step of the setup stage, the similarity matrix is constructed from the adjacency matrix such that the value of each element is the selected proximity measure computed for the feature vectors of the sub-prefecture pair corresponding to the indices of the associated element of the similarity matrix. Thus, in this fashion, an appropriate similarity matrix, also containing 255 rows and 255 columns, is generated for clustering analysis using the algorithms referenced and described earlier. We would like to note that while the population density within the Ivory Coast has geographical dependence we were not able to obtain sub-prefecture specific population data to utilize to normalize our feature space relative to population.

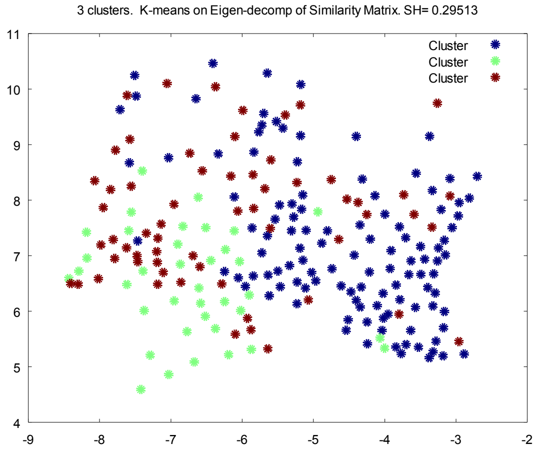

Figure 3: Clustering into 3 groups by applying K-means algorithm to similarity matrix. Note the decreasing silhouette score relative to the 2 group clustering.

As discussed earlier in section II, the spectral algorithm involves analysis of the singular values and vectors of the similarity matrix. In figure 1, we see the spectral value distribution for the similarity matrix constructed as described above. Note the steep drop-off in singular values as well as the rapid decrease in gaps between subsequent singular values.

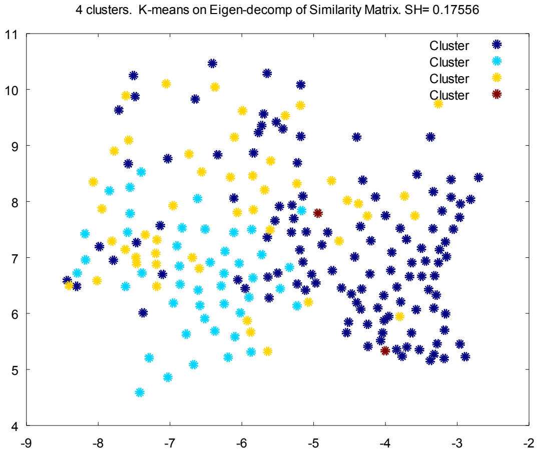

Figure 4: Clustering into 4 groups by applying K-means algorithm to similarity matrix. Note the decreasing silhouette score relative to the 2 and 3 group clusterings.

These observed phenomena are associated with the degree of structure within the data. Our dual-filtering algorithm analyzes the singular value gaps to determine suitable bounds on the number of clusters, k, and thereby significantly reduce the iterations required to generate strong clusterings comprised of the tightest possible clusters dependent on the selected clustering algorithm. Once we have selected the number of groups in which we wish to cluster the data, the algorithm for generating the clusters can be utilized in a straightforward manner as will be shown in our subsequent figures.

Figure 2 plots large points at the geographical centroids of each of the sub-prefectures that contain call data within the period of our study, so although there is some slight skewing of the relative scale of the axes, the points still align fairly well with a current political map of the Ivory Coast. The 2 colors, red and blue, in figure 2 are used to distinguish the 2 groups into which the sub-prefectures were clustered based on the call records. The silhouette score of approximately .40 indicates a significant degree of community structure. Silhouette values range between minus 1 and positive 1, whereby negative silhouette values indicate lack of structure and positive silhouette values indicate presence of structure within the analyzed data set.

As we see in figure 3, the 3-group clustering of sub-prefectures computed from the recorded caller data has a lower silhouette value than the previous figure depicting the 2-group clustering with parameter number of clusters selected automatically by the dual-filtering approach.

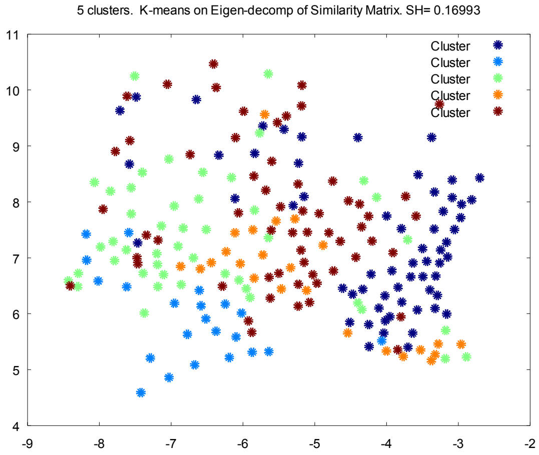

Figure 5: Clustering into 5 groups by applying K-means algorithm to similarity matrix. Note the decreasing silhouette score relative to the 2, 3, and 4 group clusterings.

This indicates that the structure present to the data is more indicative of a 2-group clustering than a 3-group clustering. In fact, the subsequent figures 4-6, each present an increasing number of groups, while the corresponding silhouette values decrease. These facts suggest that the structure present in the data best matches a 2-group clustering.

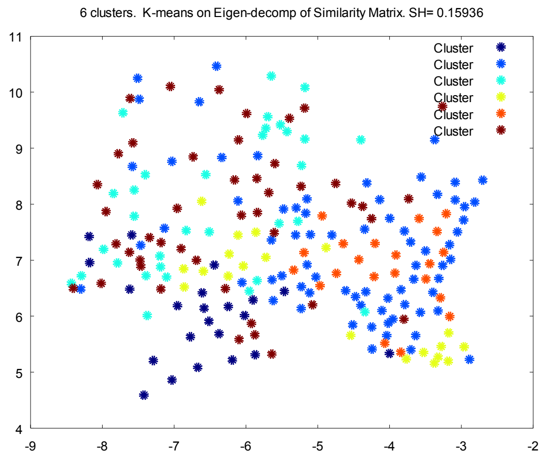

Figure 6: Clustering into 6 groups by applying K-means algorithm to similarity matrix. Note the decreasing silhouette score relative to the 2, 3, 4, and 5 group clusterings.

Other observations become apparent from examining the sequence of six figures generated from the results of the spectral decomposition based clustering algorithms. For one, the quality of the clusterings, measured in silhouette values, falls rapidly with the clusterings corresponding to 3 and then 4 clusters, but the rate of deterioration in quality slows dramatically between the 4th and 6th clusterings. This behavior matches the eigenvalue curve in figure 1. Also, we note that the region in and around Abidjan is differentiated from the other sub-prefectures in almost every one of the figures. This arises from the fact that Abidjan is the largest city by population within the Ivory Coast nation. Finally, every one of the clusterings (all of the figures) has positive silhouette values, revealing that the even with the differing number of groups in each clustering, there is evidence for corresponding structure hidden in the sub-prefecture call records, even if the degree of that particular structure (number of clusters) varies between clusterings.

V. Conclusion

In this research, we have explored the efficacy of using spectral information, revealed by singular value decomposition, to detect clustering structure and guide parameter selection for automated clustering algorithms. We utilized an independent measure, in the form of silhouette values, to characterize the quality of the clusterings generated by the spectral decomposition. The results support the conclusion that the singular value decomposition can aid in determining the presence of structure and selecting appropriate clustering parameters. As a result, these concepts seem like good candidates to improve automated clustering.

Acknowledgment

The authors would like to thank the Sensemaking/PACOM Fellowship and Swansea University's Network Science Research Center for challenging us and others to engage in researching this topic and also for the opportunity to examine this data set. Also, the authors would like to thank Daniel Rajchwald for his previous efforts to develop software input and preprocessing routines, which we reused, to prepare the data for analysis. Finally, the authors are very grateful to, Jan-Kees Buenen and Stef van den Elzenhave for advice regarding this paper.

References

[1] M. Newman, Networks, An Introduction. Oxford : Oxford University Press, 2010.

[2] Y. Chi, X. Song, D. Zhou, K. Hino, and B. Tseng, “Evolutionary spectral clustering by incorporating temporal smoothness,” in KDD Proceedings of 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Aug 2007, pp. 1-5.

[3] P. Grindrod and D. Higham, “Evolving graphs: Dynamical models, inverse problems, and propagation,” in Proceedings of the Royal Society A, 2009, pp. 753-770.

[4] N. Halko, P.G. Martinsson, J. Tropp, “Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions” arXiv.org report 0909.4061, 22 Sep 2009 .

[5] T. Klemas and D. Rajchwald, “Evolutionary Clustering Analysis of Multiple Edge Set Networks used for Modeling Ivory Coast Mobile Phone Data and Sensemaking”, Data Analytics 2014, Third International Conference on Data Analytics.

[6] H. Jo, R. Pan, and K. Kaski, “Emergence of bursts and communities in evolving weighted networks,” PLOS ONE, 2011, pp. 1-3.

[7] A. Ng, M. Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm,” Advances in Neural Information Processing Systems (NIPS), vol. 14, 2002, pp. 1-6.

[8] P. Rousseeuw, “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis,” Computational and Applied Mathematics , vol. 20, 1987, pp. 53-65.

[9] V. Blondel, M. Esch, C. Chan, F. Clerot, P. Deville, and E. Huens, “Data for Development: The d4d challenge on mobile phone data,” arXiv:1210.0137v1 [cs.CY], Sep 2012, pp. 5-9.