Identifying user habits through data mining on call data records

Article *in* Engineering Applications of Artificial Intelligence · September 2016

DOI: 10.1016/j.engappai.2016.05.007

4 authors, including:

Filippo Maria Bianchi
University of Tromsø
43 PUBLICATIONS **180** CITATIONS
SEE PROFILE

Antonello Rizzi
Sapienza University of Rome
167 PUBLICATIONS **1,090** CITATIONS
SEE PROFILE

Corrado Moiso
Telecom Italia
102 PUBLICATIONS **799** CITATIONS
SEE PROFILE

Identifying user habits through data mining on call data records

Filippo Maria Bianchi*[1], Antonello Rizzi[2], Alireza Sadeghian[3], and Corrado Moiso[4]

[1]Machine Learning Group, UiT the Arctic University of Norway, Hansine Hansens veg 18, 9019 Tromsø
[2]Department of Information Engineering, Electronics and Telecommunications (DIET), "Sapienza" University of Rome, Via Eudossiana 18, 00184 Rome, Italy
[3]Department of Computer Science, Ryerson University,350 Victoria Street, Toronto, ON M5B 2K3, Canada
[4]Future Centre Department, in Telecom Italia, via Reiss Romoli 274, 10148 Torino, Italy.

Abstract

In this paper we propose a framework for identifying patterns and regularities in the pseudo-anonymized Call Data Records (CDR) pertaining a generic subscriber of a mobile operator. We face the challenging task of automatically deriving meaningful information from the available data, by using an unsupervised procedure of cluster analysis and without including in the model any *a-priori* knowledge on the applicative context. Clusters mining results are employed for understanding users' habits and to draw their characterizing profiles. We propose two implementations of the data mining procedure; the first is based on a novel system for clusters and knowledge discovery called LD-ABCD, capable of retrieving clusters and, at the same time, to automatically discover for each returned cluster the most appropriate dissimilarity measure (local metric). The second approach instead is based on PROCLUS, the well-know subclustering algorithm. The dataset under analysis contains records characterized only by few features and, consequently, we show how to generate additional fields which describe implicit information hidden in data. Finally, we propose an effective graphical representation of the results of the data-mining procedure, which can be easily understood and employed by analysts for practical applications.

1 Introduction

Thanks to the popularity and wide diffusion of cellular phones, a huge quantity of mobile devices are moving everyday with their human companions, leaving tracks of theirs movements and their everyday habits. Mobile phones are becoming pervasive in both developed and developing countries and they can be a precious source of data and information, with a significant impact on research in behavioral science [7, 34].

A Call Data Record (CDR) is a data structure storing relevant information about a given telephonic activity involving an user of a telephonic network. A CDR usually contains spatial and temporal data and it can carry other additional useful information. Population census have been widely used in the past for keeping track of the demography and geographical movements of the population. Nowadays, due to short term and everyday mobility, more flexible methods such as various registers and indirect databases are employed: CDRs represent an optimal candidate in this sense. One of their main advantage is that they offer a statistically accurate representation of the distribution of people in an area and they can be used to track large and heterogeneous groups of people. Since CDRs evolve accordingly to the changes of users behavior, the information they carry "automatically" updates over time. Telecom operators continuously gather a huge quantity of CDRs, from which it is possible to extract additional information with low additional costs and generate valuable datasets. Analyses of CDR data can be successfully employed in many different fields, like monitoring the network, adaptation of supplied services (e.g., customers' billing, network planning), understanding of the economic level of a certain area, performing socioeconomic studies oriented to marketing and to build social networks [20]. For example, once the relationship between behavior, response, risk or other attributes is established, targeted offers of appropriate products or services can be addressed to specific customers by the telephone companies. Mobile positioning is

*filippo.m.bianchi@uit.no

a valuable source of information for investigating the spatial dynamics of human communities, but the number of published studies on this topic is still poor, mainly because of problems concerning limited access to such data and privacy issues. Localization procedures relying on mobile positioning generally provide less accurate information than the GPS (Global Positioning System), but the latter needs to be turned on to register the position, with a consequent increment of battery consumption. The wide diffusion of mobile equipment, in addition to the widespread installation of radio transmitters in both urban and rural areas, makes such positioning techniques very appealing for many case-based reasoning applications [37]. The cellular network consists in a set of base stations formed by one tower and several directional antennae. The radio coverage of a single antenna represents a cell, whose size is not fixed in the whole network. Mobile phones can be seen as a wide-area sensor network, whose measurements can be integrated with heterogeneous sources [13]. GSM (Global System for Mobile communications) is a mobile network based on the communication with an antenna covering a local area; the active connection to a certain antenna represents a spatio-temporal information which can be used for tracking the activity of a user in a GSM covered area. An effective approach for the analysis of CDR is offered by data mining techniques based on pattern recognition and machine learning procedures, which in the last years have been successfully employed in many different fields [47, 9, 56, 50].

A generic CDR relative to a telephone activity of a subscriber in a mobile network contains the identifiers of the two parties (the one which issued the call and the one which received it), the personal data of the user (name, age, sex, residence address), the coordinates of the cell which served the call, the time when the activity is registered, information on the mobile devices and on the telephonic plan. However, many of these fields are obfuscated or deleted from the publicly available records, in order to protect the privacy of the subscribers.

In fact, digital traces left by mobile phones reveal personal, often sensitive, information about their users. CDR analysis must be ruled according to the privacy of the national regulatory framework. Specifically, CDRs can be collected and processed by a mobile operator to implement the features necessary to deliver the mobile service (e.g., billing, customer care, network operation and planning). Any further usage must be either explicitly authorized by customers (e.g., through consent) or by the privacy authority. In general, the processing of anonymized CDRs (i.e., records in which the personal identification information of the referenced people is removed) is allowed, as these records are no longer personal data. Moreover, under some conditions, there is greater flexibility in processing and exploiting pseudoanonymised CDRs (i.e. records in which the ID of the referenced people is replaced with a code, often obtained through one-way cryptography): in particular, the pseudoanymization procedure must prevent the re-identification of a person from the analysis of the pseudoanonymized records. These conditions set some constraints on the datasets that a mobile operator can analyze or transfer to third parties. Therefore, also in case of pseudoanonymized records, the dataset must be pre-processed in order to reduce probability of re-identification. A common procedure consists in decreasing time resolution or increasing space granularity, so that the data collection never spans long time periods or the spatial information is not detailed. Typical examples are datasets of CDRs with high spatial resolution containing records of users which are monitored for a short period of time, or datasets where user activities tracked for longer time intervals usually come with a lower space resolution. This latter is the case considered for this study and it will be discussed in details in Sect. 3.

In this paper we propose a data-mining procedure for automatically identifying the recurrent patterns in the telephonic activity of mobile network users, in order to understand and describe their habits. We design an inference system that uses a cluster-based approach to discover regularities among data. Cluster analysis can be framed into unsupervised learning, which is the task of identify hidden structures in unlabeled data. As in our case of study, the ground truth of the expected result is unknown and there is no error or reward signal to evaluate a potential solution. Cluster-based approaches have been successfully applied for the discovery of new concepts in streams of data [46, 30]. However, the outcome of the clustering procedure is strongly influenced by the dissimilarity measure adopted and, in general, it depends on a set of configuration parameters. The procedure of tuning such parameters could be difficult and it may requires *a-priori* information not always available. As the core inference engine for our application, we use the LD-ABCD algorithm, which has been recently developed by the authors of this paper. LD-ABCD implements a novel cluster-analysis procedure, which has been presented in [8]. In order to validate the effectiveness of the proposed approach, the algorithm has previously been tested on synthetic and benchmarking datasets, where the ground-truth was known. In this work, we integrate our newly-designed system into a larger framework, which has been specifically designed to deal with a novel

real-world case of study. Since we do not possess a supervised information on the results, we evaluate the performances of our system through a comparison with a well-established subspace clustering algorithm. The main contributions of this work are summarized in the following.

1. We propose a cluster-based approach for retrieving multiple groups of CDRs, which are similar according to different subsets of features. We do not make assumptions in advance on which characteristics should be taken into account for identifying clusters, or on the total number of clusters. Moreover, each cluster is characterized by its own dissimilarity measure parameters, according to the concept of *local metric learning*. We interpret the well-defined clusters that have been identified as relevant patterns in the activity of a given user. Such patterns are used to generate a *digital fingerprint* representing user's habits, in terms of telephonic activity, geographical movements, time periods when daily communication activities are more frequent, most visited places (home, workplace etc..) and a social profiling. These fingerprints can be employed for different purposes, like profiling and definition of classes of users, depending on the specific application. With respect to other works focused on the analysis of CDR, we propose a new framework based on a complex data mining and knowledge discovery procedure. We show how meaningful patterns can be extracted and used to characterize a user, preserving his privacy and without making any *a-priori* assumption on the nature of the data.

2. When data characterized by an high number of distinct features are available, many informative, significant and useful information can be easily derived, more complex analysis can be performed and non-trivial relationships among data can be discovered. However, in our work we process a dataset of pseudo-anonymized CDRs where each entry contains only a limited number of attributes. The problem we face is challenging since it seems that, at a first glance, only naive regularities in the data can be retrieved. In this paper we show how to extract implicit information from the data and we use them for identifying hidden frequent patterns which lead to meaningful results and considerations.

3. We propose an effective method of visualization, which encodes data and information into visual objects. Our main goal is to communicate information clearly and effectively through graphical tools, in order to express and to quantify the results , through visual human interfaces [22].

The remainder of the paper is organized as follows: in Sect. 2 we review some relevant works and applications focused on the analysis of CDRs. In Sect, 3 we present the dataset considered for the analysis, discussing the representation of the data and how implicit information contained in the CDRs can be extracted. In Sect. 4 we propose a framework that can be used for discovering relevant patterns in the data by relying on a procedure of cluster analysis and we show the obtained results in Sect. 5. Finally, in Sect 6 we draw our conclusion and we discuss future works and improvements.

2 Works on Call Data Records

In this section we review some relevant works which leverage the information contained in CDRs for a multitude of different applications.

CDRs can be effectively used for understanding the interactions between users and to define a network of social relationships among them. In [21] the authors use CDRs information to provide insights into the relational dynamics of individuals, demonstrating that it is possible to accurately infer friendships. They proved that calls between friend dyads have distinctive temporal and spatial patterns. In [16], the authors use CDRs to study the average collective behavior at large scales, focusing on the occurrence of anomalous events.

One of the largest field of application of CDRs is the prediction of the movements of the users according to their calls. By predicting their movements, is it possible to understand the life-style of the users and to provide useful information to the telephonic companies, in order to tailor targeted phone plans for the customers. CDRs are used for understanding which are the most busy areas and times during the day, to predict the movements of the people and promote a better transportation service [6]. In [15] mobile data are used for monitoring in real-time the traffic conditions and pedestrian distribution. In a study performed by the German telephone company T-Mobile, mobile data are used for tracking

trajectories of vehicles on the streets [42]. A classification of the users based on their movements is proposed in [23], where the customers are assigned to three predefined classes (resident, commuters and visitors), depending on their movements in certain areas. The raw CDRs are initially aggregated on both spatial and temporal basis: for each user the authors build a matrix containing information on the number of calls which took place on a given area in three different periods of the day. Raw data can be aggregated directly by the telephone operator, in order to generate a compact representation which is easy to manage and above all is anonymized, so that the privacy of the users can be preserved. On the aggregated data they perform a clustering procedure using Self Organizing Maps and the obtained clusters are labeled with the most frequent class. Such clusters can be used to describe the habits of the users in a given area. In [45] the limits on the predictability of the movements of the people are studied. The authors used CDRs gathered from 50.000 users in a time-window of 3 months of activity and they propose three different measurements of entropy for analyzing the predictability of their movements: the random entropy, the temporal-uncorrelated entropy and the actual entropy. They claim to be able to predict 93% of the users movements at best and 80% in the worst case. In [25], authors build an Inhomogeneous Continuous-Time Markov (ICTM) model using both temporal and spatial information, for predicting the next position of the user in a given time lapse. The results show that the ICTM model is able to forecast the correct time interval with an error of 45 minutes and the next location with a 67% accuracy. Also in [24] the authors suggest using spatio-temporal data for predicting users position, employing 10 different types of models based on Bayesian rules or Markov models. The best forecast accuracy is achieved with a model called HPY (Hierarchical Pitman-Yor) Prior Hour-Day Model, which is able to correctly localize the users with an accuracy of 50%. In [38] the urban cells are clustered together, in order to avoid repeated switches of users between adjacent cells and to force them to assume the same size of the rural cells. Successively, a data mining procedure is performed on the sequences of the movements, aiming to predict whether a user will change its position or not. In order to increase the accuracy, temporal information are also included. The CDRs are then represented with one-day-long sequences and 3 different problems are considered: (i) predicting the first time interval when the user will change his position and where he will move; (ii) predicting only the next movement; (iii) predicting the next movement and his next telephonic activity.

Other applications leverage CDRs to promote analysis, diagnosis and prevention in organizational, social and security contexts, like tracking population movements in emergency conditions to improve government alert communications. Social applications of the CDRs involve statistics on the population, performed through a correlation of mobile data with poverty indexes and GNP (Gross National Product) in order to draw poverty maps that can be used for orienting actions and programs for development in the most needing regions [44]. Correlating mobile data with information on the diffusion of epidemics and diseases, allows the definition of efficient models for preventing and containing the spreading of epidemics. Such models can be employed for informing population, optimizing medical resources and planning vaccinations [36].

CDRs can also be used for profiling or for understanding the behaviors of one or more users. An interesting application of CDRs is the analysis and prediction of the lifestyle of the users. In these studies, the privacy of the users must be protected, their identities must remain unknown and not be accessible from outside. Those aspects are treated in [49], where it is proposed a strategy for increasing the security of the data, preserving data usability and their semantic value. In [27] the authors propose a model for grouping users according to their similarity in calling patterns and for classifying new call records. The procedure consists in (1) extracting from the CDRs the attributes which better characterize each user; (2) applying a clustering algorithm to identify common behavioral patterns among the costumers, trying to balance the dimension of clusters, each one representing a different state; (3) generating a transition matrix among the states which must be sparse, so that a user can reach only a limited subset of different states each time; (4) using such model to classify new call records. The model reflects the habits of the user and it is uncorrelated with his identity, meaning that it can be exported and re-used. The model can also be dynamically updated as long as new data are collected, evolving as the life-style of the user changes. Such model can be used by telephone companies for offering products to the user, according to their current and potential future states. In [28] the authors classify the users in a set of pre-determined classes by relying on the CDRs. Only the fields of the CDR which discriminate better the users are selected. In [4], CDRs are used to identify the areas most visited by the users, called "anchor-points" (AP), which are commonly the working and the living place along with the locations where secondary activities take place with a given frequency (e.g. the gym). The procedure for identify

AP is the following: (1) for each user they identify the cells from where more than 2 calls are issued on a monthly basis (Regular Cells - RC); (2) the two RC with the highest number of calls are tagged as home and work place. Work place is discerned from home place depending on the timing of the calls; (3) they process cases of users with more than one work or home AP, which could live or work on borders of adjacent cells; (4) for users whose activity is mainly registered in a single cell – meaning, for example, that they live and work in the same area – a multi-functional task AP is defined; (5) the remaining cells are classified as secondary AP.

3 Dataset Description

In this work we analyze a dataset of the Orange telephone data published for the "Data for Development" (D4D) challenge [14], which is an open collection of CDRs, containing anonymous calling events of Orange's mobile phone users in Ivory Coast. More information on the challenge are available on the website `http://www.d4d.orange.com`. The data consist in anonymized mobile phone calls and SMS that have been gathered in the period that spans form December 1, 2011 to April 28, 2012 and they are arranged in four different datasets. We analyze the CDRs relative to individual trajectories of 50,000 randomly selected customers, gathered over the entire observation period. Each CDR contains the time of the call and the location, expressed as the identifier of the prefecture, from where the call has been issued. Even if the spatial resolution in the records is low, the users are tracked for a time interval which is sufficiently long to identify meaningful patterns in their activity. This aspect is fundamental, since analyzing data concerning the activities of a user for an extended time interval allows a better detection of the regularities in his behavior and the profile can be drawn with higher accuracy.

A CDR in this dataset has the following structure: {`user_id`, `conn_datetime`, `subref_id`}, where `user_id` is the anonymized identifier of the user, `conn_datetime` is the date and the time of the registered telephonic activity and `subref_id` is the identifier of one of the 255 sub-prefectures in the country.

3.1 Data preprocessing and representation

The data that we consider contain only two fields, which are the temporal and the spatial information relative to each registered telephonic activity. However, it is possible to extract useful implicit information contained in the dataset, adding additional features. The the resulting final structure of an element in the dataset is: { `subref_id`, `week_day`, `work_day`, `conn_time`, `day_period`, `prev_call` }. In the following, we provide a description for each field in the augmented records.

- `subref_id` is preserved as in the original dataset. Since the values are unique identifiers, we consider the domain for this field to be discrete (nominal).
- `week_day` is generated from the field `conn_datetime` and it represents the day of the week where the connection has been registered. The possible values are: { `Mon`, `Tue`, `Wed`, `Thu`, `Fri`, `Sat`, `Sun` }. They are drawn from a nominal domain.
- `work_day` is a Boolean value derived from `conn_datetime` and it distinguishes working days from weekend days (0 for Saturday and Sunday, 1 for the other days). Also in this case the domain is nominal.
- `conn_time` is the time, expressed in hours and minutes (`HH:MM`), when the telephonic activity is registered. This is a continuous circular domain in [`00:00`, `23:00`] with resolution of 1 hour.
- `day_period` is the period of the day when the call was issued. In particular, activities registered in the time interval [`07:00`, `13:00`] are considered morning activities, the ones in [`14:00`, `19:00`] are afternoon activities and the ones in [`20:00`, `06:00`] are evening/night activities. We have then a nominal domain with the values { `Mor`, `Aft`, `Eve` }.
- `prev_call` is the elapsed time (in minutes) from the previous call and it belongs to an ordinal domain in $\mathbb{N}^+$.

We represent the CDR with a data structure called *sectioned vector* [33], which can be defined as a n-dimensional vector $\mathbf{u}$ composed of the concatenation of s opportune sub-vectors, called *sections*. The sectioned vector is a flexible data structure, which allows to represent an object whose components belong to different domains, tailoring the definition of a proper dissimilarity measure for each section. The number and size of the sections constitute the structure of the sectioned vector:

$$\mathbf{u} = (\mathbf{u}_0; \mathbf{u}_1; \ldots; \mathbf{u}_{s-1}), \text{ with } \sum_{i=0}^{s-1} \dim(\mathbf{u}_i) = n. \tag{1}$$

Other existing alternatives to represent data containing values coming from heterogeneous domains are labeled sequences and graphs. These data structure also carry topological information on data, but they are more difficult to handle and the procedures used for evaluating their dissimilarity are usually complex, less accurate and more demanding in terms of computational resources [10, 11]. In many cases data are not characterized by important topological information, i.e. it is only relevant the numerical value of each component and not their spatial/temporal organization. This coincides with our case of study, for which it is recommended to use sectioned vectors, avoiding more complex data structure.

We now introduce a generalized dissimilarity measure for sectioned vectors of $\mathbb{R}^n$. An highly flexible dissimilarity measure is required when the components have heterogeneous meaning and the use of a simple Minkowski metric is not so obvious [35, 40]. Let $\mathbf{u}^{(a)}, \mathbf{u}^{(b)}$ be two sectioned vectors with the same structure, and let $d_i(\mathbf{u}_i^{(a)}, \mathbf{u}_i^{(b)})$ be a dissimilarity measure for the i-th section. We can define the dissimilarity measure for the whole vectors $\mathbf{u}^{(a)}$ and $\mathbf{u}^{(b)}$ as:

$$d(\mathbf{u}^{(a)}, \mathbf{u}^{(b)}, \mathbf{m}) = \sum_{i=0}^{s-1} \mathbf{m}(i) d_i(\mathbf{u}_i^{(a)}, \mathbf{u}_i^{(b)}), \tag{2}$$

being $\mathbf{m}$, in general, a real-typed vector in the unitary hypercube in $\mathbb{R}^s$, which represents the parameter configuration (PC) of the dissimilarity measure $d(\cdot, \cdot, \mathbf{m})$. Each entry is a weight that tunes the importance of the respective component in the structured vectors for the computation of the total dissimilarity. For example, if $\mathbf{m}(1)$ assumes an high value (close to 1) it means that the difference between the values in the first section of two vectors $\mathbf{u}_i^{(a)}$ and $\mathbf{u}_i^{(b)}$ highly influences the degree of their total dissimilarity.

In our dataset, the content of the sections belong to three types of domain: ordinal, nominal and ordinal circular. Consequently, we use three different dissimilarity functions, which are introduced in the following. For what concerns the ordinal domain from which the values of `prev_call` are drawn, we use the normalized Manhattan distance. The decision of using the Manhattan distance was maturated after having tried different type of dissimilarity measures, among which the Euclidean distance. In our preliminary experiments we observed that there were not significant improvement (or in general changes) in the quality of the results obtained. On the other hand, there was a tangible increment in the computational time. The evaluation of the dissimilarity measure is repeated many times during the execution of the algorithm, which benefits from the simplicity of the Manhattan distance, which requires performing only sums and multiplications with respect to exponential operations. The Manhattan distance is defined as follows:

$$d_M(\mathbf{u}_i^{(a)}, \mathbf{u}_i^{(b)}) = \frac{\sum_{j=0}^{\dim(\mathbf{u}_i)-1} \left| u_{ij}^{(a)} - u_{ij}^{(b)} \right|}{\max(\mathbf{u}_i) - \min(\mathbf{u}_i)}. \tag{3}$$

For comparing the values of `subref_id`, `week_day`, `work_day` and `day_period` that belongs to a nominal domain, we use the Delta dissimilarity function, defined as:

$$d_\Delta(\mathbf{u}_i^{(a)}, \mathbf{u}_i^{(b)}) = \begin{cases} 0 & \text{if } \mathbf{u}_i^{(a)} = \mathbf{u}_i^{(b)}, \\ 1 & \text{otherwise.} \end{cases} \tag{4}$$

Finally, for what concerns the values of the attribute `conn_time`, it is required a dissimilarity that operates on an ordinal circular domain. Specifically, we defined the following function:

$$
\begin{aligned}
& d_C(\mathbf{u}_i^{(a)}, \mathbf{u}_i^{(b)}) = \frac{\min(t_1, t_2)}{12}, \text{with} \\
& t_1 = \|\mathbf{u}_i^{(a)} - \mathbf{u}_i^{(b)}\|, \\
& t2 = \min(\|24 - \mathbf{u}_i^{(b)} + \mathbf{u}_i^{(a)}\|, \|24 - \mathbf{u}_i^{(a)} + \mathbf{u}_i^{(b)}\|).
\end{aligned} \tag{5}
$$

In order to stress the importance of a section in the computation of the dissimilarity measure, in this work we use PCs with Boolean weights, i.e. $\mathbf{m} \in 0, 1^6$. Even if defining $\mathbf{m}$ in the continuous interval [0,1] provides an higher degree of flexibility, we opted for Boolean weights to facilitate the interpretation of the features which are selected to be relevant.

Defining $\mathbf{m}$ in advance is a difficult task which demands *a-priori* information on the data and on the applicative context. If, as in our case, prior knowledge is not available and the objective is to identify *any* existing regularity in the data, it is required a strategy for effectively determining the value of weights. This can be done through a data mining procedure, which deals with the problem of metric learning [43, 52, 53, 55, 17] and the concept of local metrics [32, 40, 41, 5, 26].

The focus of our work is to determine multiple instances of $\mathbf{m}$, which allow to retrieve meaningful regularities among data, necessary to build *digital fingerprints* of the users. We use a data mining algorithm that has been designed to automatically determine a specific dissimilarity measure for each cluster, implementing a *local* distance learning approach.

4 Approaches for the analysis

In this section, we present a framework that relies on two different cluster-based approaches [29, 51, 19] to retrieve relevant information in the dataset of CDRs.

When data are represented by structured vectors and the PC $\mathbf{m}$ of the dissimilarity measure is the collection of Boolean weights that determine the importance of each feature in the evaluation of the similarity of two elements, the task of setting $\mathbf{m}$ can be seen as the selection of the relevant features in each cluster. In this case, tuning the values in $\mathbf{m}$ can be straightforwardly connected to the problem of *subspace clustering*, which retrieves the subset of dimensions to be considered in the clustering problem, by removing irrelevant and redundant dimensions. While traditional clustering algorithms consider all the dimensions in the attempt of collecting as much information as possible, in high dimensional data many of the dimensions are often irrelevant. Irrelevant dimensions can misguide the clustering procedures by hiding clusters in noisy data. Unlike feature selection methods which examine the dataset as a whole, both the approaches that we present in this section localize their search and are able to uncover clusters that exist in multiple, possibly overlapping subspaces (local metrics learning). Clusters found in lower dimensional spaces also tend to be more easily interpretable, allowing the user to better conceive further analysis.

This procedure is also highly related with feature selection and dimensionality reduction techniques [57]; it can be considered as an extension that attempts to find clusters in different subspaces of the same dataset. Just like the feature selection procedure, the algorithms that we present require a search method and an evaluation criteria and they must somehow limit the scope of the search, to consider different subspaces for each cluster.

We used a data visualization technique, based on charts of various types, to show patterns and relationships in the data for one or more variables. Through an effective visualization, users can easily analyze data, discovering interesting insights. The visualization method that we propose in Sect. 5 makes complex data more accessible, understandable and usable by a larger class of users, not necessarily experts in data mining and graphics reading.

The first algorithm that we discuss in Sect. 4.1, is a novel multi-agent system for cluster and knowledge discovery called LD-ABCD [8], which has been recently developed by the authors of this paper. LD-ABCD identify well-defined clusters in the dataset using different configurations of the dissimilarity measure and it provides a semantic characterization of each identified cluster. The second method that we consider as an alternative implementation of our framework, is PROCLUS, discussed in Sect. 4.2, a subspace clustering method that identifies the most descriptive dimensions among the elements in each cluster.

4.1 Cluster Discovery by Dissimilarity Function Adaptation

LD-ABCD is a multi-agent algorithm designed to automatically discover relevant regularities in a given dataset, determining at the same time a set $\mathcal{M}$ of PCs of the adopted parametric dissimilarity measure, yielding compact and separated clusters in the data. Each agent operates independently on a suitable weighted graph, which is used to represent the data. The graph is fully connected, each node corresponds to an element in the dataset and the edges are labeled with a value proportional to the similarity of the two connected elements. The weights on the graph depend on the specific PC $\mathbf{m}_j$ of the dissimilarity measure adopted by the j-th agent. A new PC is iteratively selected by an agent j to construct its own instance G_j of the weighted graph.

The clusters discovery procedure is implemented by means of multiple Markovian random walks (RW), which are performed independently, at the same time, by several agents on the different instances of the weighted graph. The behavior of the RW is thus dependent on the PC selected by the agent. During a RW, an agent searches and takes decisions autonomously for one cluster at a time. A suitable on-line mechanism is designed to decide whether a set of elements visited (i.e., "walked upon") by an agent should be accepted as a meaningful cluster or rejected. Specifically, the set of nodes visited by the agent during the random walk is a subgraph c_i that represent a potential cluster. Its quality is evaluated with a measures called Cluster Quality (CQ) that relies on the concept of the graph conductance [31] and that depends also from the configuration $\mathbf{m}_j$ of the dissimilarity measure currently adopted by an agent, such that CQ is a function of the pair $\langle c_i, \mathbf{m}_j \rangle$. A cluster is accepted by an agent if its CQ value is greater than a threshold τ_{CQ}, which controls the overall quality of the solutions returned by the algorithm.

An edge $e_{kl} \in \mathcal{E}$ is labeled with a weight, $w(e_{kl}; \mathbf{m}_j) \in [0, 1]$, which depends on the dissimilarity $d(x_k, x_l; \mathbf{m}_j)$ according to the following relationship:

$$w(e_{kl}; \mathbf{m}_j) = \exp(d(x_k, x_l; \mathbf{m}_j) \cdot \tau_{\text{exp}}) \tag{6}$$

where τ_{exp} is a parameter used to magnify the edge weights between similar elements, making less likely the unwanted transitions to vertices connected by low weights. A correct setting of τ_{exp} is crucial, since it affects the behavior of the RW. An heuristic approach proposed in [8], consists in generating a weighted graph for every value of τ_{exp} in an interval $[\tau_{\text{exp}}^{min}, \tau_{\text{exp}}^{max}]$ and then performing a sufficiently high number of RWs to retrieve clusters. The optimal τ_{exp} is evaluated in function of the average CQ and cardinality of the clusters returned.

As long as the execution of LD-ABCD proceeds, an agent might find a set $\mathcal{C}$ of very similar (or even equal) clusters using different PCs, in the sense that they may overlap significantly. These clusters are merged into a *meta-cluster* $\hat{c}$, which contains the elements which appear more frequently in the clusters in $\mathcal{C}$. All the PCs used to discover the clusters in $\mathcal{C}$ are considered *equivalent* with respect to $\hat{c}$ and are inserted in a list $\mathcal{L}$ associated to the meta-cluster. The CQ value of each PC $\mathbf{m}_j \in \mathcal{L}$ is then recomputed on $\hat{c}$ and the higher its value, the better $\mathbf{m}_j$ characterizes the elements in $\hat{c}$. A parameter ϑ is used as a threshold for the distance between two clusters, in terms of percentage of elements commonly shared, for defining whether two clusters should be considered similar and included in the same meta-cluster; notably, ϑ can be thought as the "radius" of the meta-cluster.

The final output of the algorithm is a collection of meta-clusters, which may be (partially) overlapped and which may not cover the entire dataset. Each meta-cluster is associated with the list of PCs $\mathcal{L}$, representing a consistent and interpretable semantic characterization of the meta-cluster, since it defines the set of dissimilarity measures according to which the elements in the meta-cluster as indistinguishable. An important remark is that we forced each PC to assume at least two values different from 0, in order to avoid trivial solutions containing clusters composed of elements which are similar only with respect to a single feature.

Thanks to the multi-agent architecture, the algorithm possess high scalability and a distributed implementation is straightforward. LD-ABCD can process any type of data (vectors, graphs, sequences, etc..), given a suitable (parametric) dissimilarity measure for comparing the elements. In particular, the algorithm can process data that are defined in non-metric spaces, greatly increasing the scope of problems that can be treated. Unlike the k-clustering paradigm, LD-ABCD does not require to specify in advance the desired number of clusters to be returned. This feature is very important in our case of study, since we do not possess any *a-priori* information on the dataset. Another significant feature of LD-ABCD, which is relevant for our application, is the provided powerful semantic characterization of the identified clusters, allowing important analyses on the content of the results returned.

4.2 Subspace Clustering

The alternative procedure that we consider for comparing performances of our framework is based on a subspace clustering algorithm. Subspace clustering algorithms can be divided in two groups, the top-down search and bottom-up search methods, which are distinguished by their approach to identify subspaces [39].

The bottom-up search methods try to reduce the search space, taking advantage of the downward closure property of density: if there are dense units in k dimensions, there are also dense units in all $k-1$ dimensional projections. These methods first create an histogram for each dimension and then select those bins whose densities are above a given threshold. Candidate subspaces in higher dimensions can then be formed using only those dimensions that contain dense units, dramatically reducing the search space. The algorithm proceeds until no more dense units are found. Adjacent dense units are then combined to form clusters. These algorithms, of which CLIQUE [3] is one of the most famous representative, have been conceived to be applied on strictly real-typed domains, where a notion of proximity can be easily defined. In our case, we use sectioned vectors containing values drawn from circular and discrete domains, where the concept of density cannot be defined. This prevents to map similar values of the domain to close position on a grid representation. For this reason, applying bottom-up search methods to our problem is not feasible and we decided to consider the top-down approach.

The top-down subspace clustering approaches start by finding an initial approximation of the clusters in the full feature space with equally weighted dimensions. Successively, to each dimension is assigned a weight for each cluster. The updated weights are then used in the next iteration to regenerate the clusters. This approach requires to repeat multiple iterations of the clustering algorithm, considering the full set of dimensions, in order to converge to the optimal solution.

Among all the top-down approaches, we selected the PROCLUS algorithm [2], which is one of the first developed and probably the most famous. The algorithm consists in three different steps called initialization, iteration, and cluster refinement, during which the clustering is iteratively improved. In the initialization step PROCLUS samples the data, then it selects with a greedy strategy a set $\mathcal{M}$ of representatives, called medoids, to be as much spread as possible in the full dimensional space. The medoids represent the pool of the candidates for the cluster representatives. In the iteration phase, a random set of k medoids are selected from $\mathcal{P}$ and for each medoid p_i a neighborhood is generated, consisting in all the points whose distance from p_i is less than or equal to the distance from p_j, being p_j the closest medoid to p_i. For each medoid, the algorithm selects the set of dimensions along which the distances of the elements in the neighborhood are the smallest. The total number of dimensions associated to medoids must be $k \cdot l$, where l is an input parameter that selects the average dimensionality of the subspaces for each cluster. Once the subspaces have been selected for each medoid, all the elements in the dataset are assigned to their closest medoid, according to the average Manhattan segmental distance, which considers in the computation of the dissimilarity from each medoid only the selected dimensions. The medoid of the cluster with the least number of points is discarded along with any medoids associated with fewer than $(\frac{N}{k}) \cdot \texttt{minDev}$ points, being N the total number of patterns and `minDev` another input parameter. At the end of each iteration, discarded medoids are replaced with new ones, randomly chosen from $\mathcal{P}$, and it is checked if clustering has improved. In the refinement phase, PROCLUS computes new dimensions for each medoid based on the clusters formed (rather than on the neighborhood) and then it reassigns points to the medoids, removing outliers.

Due to the use of sampling, PROCLUS is faster than many other subspace clustering algorithms, especially on larger datasets. On the other hand, one of the main drawback of the algorithm is its strong dependence on the parameters k and l which, in many cases, can be hard to be set in advance, since they require an adequate knowledge of the problem and of the dataset at hand. Another drawback is due to the bias toward clusters that are hyper-spherical in shape. Additionally, since the average number of dimensions is given, the number of selected dimension in each cluster will be similar. It is also important to notice that PROCLUS creates a partition of the dataset and, possibly, an additional group of outliers. This means that each instance is assigned to only one cluster (or to the outliers group). This is a critical difference with respect to the procedure implemented in LD-ABCD, which does not form a proper partition, allowing the generation of overlapping clusters, meaning that an element can be assigned to one cluster, more cluster or no clusters at all.

In our experiments we used PROCLUS configured with the dissimilarity measure defined in Eq. 2, which can be considered a generalization of the average Manhattan segmental distance.

5 Experiments and Results

In this section, we first analyze the set of CDRs relative to a specific user. Successively, we process the dataset with our data-mining procedure and we discuss the results obtained by the two different implementations, proposed in Sect. 4.

The original dataset presented in Sect. 3 contains CDRs relative to 50,000 users. In our experiments we processed the data relative to the calls of more than 100 different users, which have been randomly selected. For most of them it was possible to identify clear and distinct patterns, while for others we did not obtained meaningful results, mainly because of irregular telephonic activity or for the limited number of the calls issued by the user. In the following, we show an example of how an analysis of the CDRs relative to a given user can be performed using the proposed methodology.

To visualize the content of the CDRs relative to a given user, we use 4 different charts that show how the CDRs are distributed according to the accounted features. In particular we have:

1. An histogram describing the number of calls done by the user from different prefectures. Each bin is associated with one of the prefectures from where calls were issued and the height of the bars is proportional to the number of calls done in that prefecture.

2. An histogram that describes the distribution of the values contained in the field `prev_call`. Each bin of the histogram represents the time elapsed from the previous call and its height is proportional to the number of CDRs whose value `prev_call` falls in that interval.

3. An histogram that represents the distribution of the calls of the user among the 7 days of the week, according to the field `week_day`.

4. An histogram that represents the distribution of the calls of the user among the 3 periods of the day, according to the field `day_period`.

For the sake of conciseness, among all the user that we have processed, we show the results relative to two particular users, characterized by a sufficiently high number of calls, for which we identified meaningful regularities. The identifier of a given user coincides with its order of appearance in the original dataset of 50,000 different customers. The considered users are the 4-th and the 6014-th in the dataset, the number of CDRs for these users are 1453 and 1003, respectively, and their content can be described through the 4 charts that we have previously defined. In Fig. 1 the values relative to the CDRs of User 4 are depicted using histograms where the value of each bin is normalized with respect to the total number of CDRs. Note that, in order to make the visualization more concise, we do not report the histograms concerning the fields `work_day` and `conn_time`, which we retained to be the least interesting attributes to be visualized.

From Fig. 1(a), it is easy to observe that User 4 performs calls from 5 different prefectures, namely the ones associated with the identifiers 60, 61, 64, 138 and 198 in the dataset, and that most of the calls are issued from prefecture 60, which is likely the one where the person spends most of the time. Note that this spatial information can be easily retrieved from the raw data in the dataset during a pre-processing step. For what concerns the temporal component instead, it is difficult to identify distinct patterns from the original raw data, while they emerge more clearly from the dataset augmented by expliciting the content in the temporal attribute. The distribution of the values contained in such fields is displayed in the 3 remaining histograms. Fig.1(b) depicts an histogram of the distribution of the calls in term of the number of minutes elapsed from the previous call. Every bar represents the number of calls whose time elapsed from the previous call is less than the bin label. The last bin contains all the calls issued after more than 1 day (24 hours) from the previous one. As we can see, the distribution among every bin is well balanced, with the exception of the bin relative to the calls issued after 12 hours, which contains less entries. In 1(c) we can observe the distribution of the calls along the days of the week. Even in this case, the number of calls are pretty well distributed with the exception of Sunday, when the activity of the user is very low. Analogously, 1(d) represents the distribution of the calls among the 3 periods in which the day is split: a pattern that clearly emerges is that the user calls much more frequently in the morning and in the afternoon, rather than in the evening.

From this first analysis we can draw some initial consideration. For example we can assume that User 4 uses its phone mostly for work calls, which reasonably occur more frequently in the morning and during

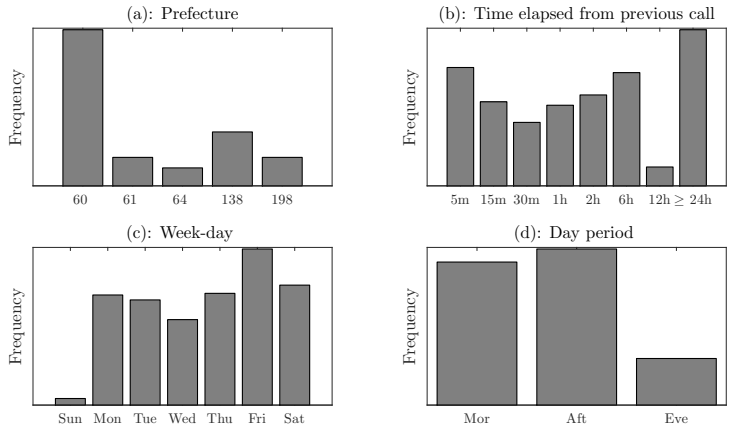


Figure 1: These charts represent the distribution of the values in the CDRs of User 4. The chart (a) represents the distribution of the calls among the prefectures visited by the user, the taller the bar in the histogram, the higher the number of calls issued from that prefecture. In (b) we can observe an histogram of the distribution of the calls, which are grouped according to the time elapsed from the previous call. In (c) and (d) the histograms show the total number of calls which are done in the 7 days of the week and in the 3 periods of the day respectively.

the work days. In the next sections we perform a more detailed analysis using the two cluster-based algorithms LD-ABCD and PROCLUS.

5.1 Analysis Results with LD-ABCD

The solution returned by LD-ABCD consists in a set of meta-clusters and in the list of associated PCs. The first step that must be performed before executing the algorithm is the heuristic estimation of the critical parameter τ_{exp}, according to the procedure described in [8]. We tried all the values of τ_{exp} in the interval $[1, 120]$, executing the algorithm 150 different times for each value and we have evaluated the average CQ obtained and average size of the clusters found (see Fig. 2). We noticed that in many dataset of CDRs related to different users, the optimal value of τ_{exp} falls in the interval $[15, 25]$, for which the average size of the clusters stops to decrease and their average CQ stop to increase.

Concerning the threshold which regulates the minimum allowed CQ value, for low values of τ_{CQ} a larger number of clusters is returned and some of them are characterized by a lower quality, in terms of compactness and separability from the remainder of the dataset. This setup is more conservative since less clusters are discarded, but, at the same time, it demands more computational resources because an higher amount of information must be processed in the successive steps of the procedure. For our analysis, we are more concerned about the quality of the result, rather than on computational efficiency. For this reason we set $\tau_{\text{CQ}} = 0.8$, a rather low value, meaning that only the clusters whose CQ is lower than 0.8 are discarded in the searching procedure and are not aggregated to any meta-cluster.

The last parameter to be set is the maximum allowed radius of the meta-clusters $\vartheta \in [0, 1]$: high values of ϑ generates less but larger meta-clusters, while for lower values the number of meta-cluster returned is higher, but their dimension is smaller. In our experiments it was useful to retrieve a large number of meta-clusters, in order to identify more clearly the most dense regions of the cluster space, which represent the most recurrent patterns among the CDRs of a given user. Through a trial-and-error approach, we found that setting $\vartheta = 0.2$ generates a number M of meta-clusters which is sufficiently high for the purpose of our analysis. Each meta-cluster $\hat{c}_i$ returned by LD-ABCD is represented with the Boolean vector μ_i of N elements, where N is the number of CDRs in dataset: if the $j-$th component of the vector $\mu_i(j) = 1$ it means that the j-th CDR in the dataset belongs to the cluster, otherwise if $\mu_i(j) = 0$, the j-th CDR is not in $\hat{c}_i$. In order to visualize the results, we applied a PCA (Principal Component Analysis) to the $M \times N$ matrix $\hat{\mathcal{C}}$ containing all the meta-clusters and we retrieved the first 3 principal components, that is, the representation of $\hat{\mathcal{C}}$ in the first 3 dimensions of the principal component space. In this way, we obtained a $M \times 3$ matrix where each row corresponds to a meta-cluster that can be visualized in a 3-dimensional space, as depicted in Fig. 3. Each meta-cluster in the plot is colored

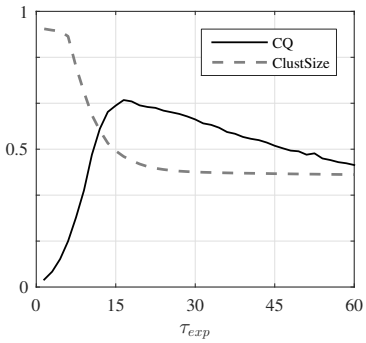


Figure 2: The graphic shows how the average CQ and size of the clusters found in the random walk change, as the edges of the considered graph vary according to the different values assumed by τ_{exp}. The parameter controls the weight of the edges, by modifying the values on their labels. High values of τ_{exp} force the walker to visit only very similar nodes, which produce small and compact clusters. On the other hand, a lower value of τ_{exp} allows the walker to move on larger sets of nodes, with a consequent discovery of wider clusters, which however are less compact and separated from the remainder of the dataset, and consequently they are characterized by a lower CQ value.

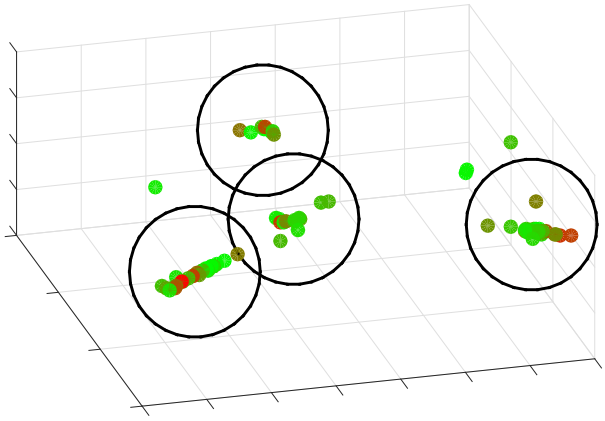


Figure 3: Plot of the first 3 principal components resulting from a PCA on $\hat{\mathcal{C}}$, relative to user 4. Each dot represents a meta-cluster and it is possible to identify 4 different dense regions, which are marked with a black circle. The color of a dot represents the CQ value of the meta-clusters: bright red tonalities correspond to high CQ values.

according to its CQ value: a color close to green means that its CQ is low, while red represents the clusters with high CQ values. Note however that all the meta-clusters are formed by clusters whose CQ will be higher than the threshold τ_{CQ} and so, even the green dots, represent clusters with a relatively high CQ.

From the Fig. 3 we can observe that there is a set $\mathcal{R}$ of dense regions, which have been verified to be formed by meta-clusters whose content is similar, i.e. the sets of the elements of the dataset that they represent are strongly overlapped. In this case, we identified 4 dense regions, which are marked in the figure with a black circle. From each region $r \in \mathcal{R}$ we selected the meta-cluster $\hat{c}_r$ with the highest CQ value, (the dot with the brightest red color) and we consider it the representative of the pattern described by the region. The list $\mathcal{L}_r$ of PCs associated to $\hat{c}_r$ represents the sets of features according to which the elements contained in $\hat{c}_r$ are similar to each other. Each pair $\langle \hat{c}_r, \mathcal{L}_r \rangle$ represents then a recurrent pattern r in the dataset. In Fig. 4 we report for each one of the 4 patterns a set of pie charts representing the distribution of the values in `subref_id`, `week_day`, `work_day`, `conn_time`, `day_period` and `prev_call` in the meta-cluster. In the figure, each pie chart represents the values distribution for an attribute. Note that for a more clear interpretation, we labeled only the largest slice, which represent the most frequent value assumed by the attribute.

(a) Meta-cluster $\hat{c}_1$

(b) Meta-cluster $\hat{c}_2$

(c) Meta-cluster $\hat{c}_3$

(d) Meta-cluster $\hat{c}_4$

Figure 4: The six pie charts in each row of this figure represent the distribution of the values in the attributes `subref_id`, `week_day`, `work_day`, `conn_time`, `day_period` and `prev_call` in each meta-cluster found in CDRs of User 4. Each slice represents the frequency of the values for that attribute. In each chart we report the name of the most frequent value.

In the following, we discuss in detail the patterns identified for user 4, according to the content of the meta-clusters.

The first pattern is represented by the meta-cluster $\hat{c}_1$ and the list of PCs $\mathcal{L}_1$, whose content is described in Tab. 1.

Table 1: The table reports the PCs contained in the list $\mathcal{L}_1$ associated to the meta-cluster $\hat{c}_1$ and the average value in each component. Each entry concerns the features `subref_id`, `week_day`, `work_day`, `conn_time`, `day_period` and `prev_call` respectively. A value equal to 1 means that the feature is considered relevant in the cluster, 0 otherwise.

	Values					
	`subref_id`	`week_day`	`work_day`	`conn_time`	`day_period`	`prev_call`
PC_1	1	1	1	0	1	0
PC_2	1	1	1	1	0	0
PC_3	1	1	1	0	1	1
PC_4	1	1	1	0	0	0
PC_5	0	1	1	0	1	0
PC_6	1	0	1	0	0	1
PC_7	1	1	1	0	0	0
Avg	**0.86**	**0.86**	**1**	**0.14**	**0.42**	**0.28**

As we can see from Tab. 1, the most selected features are `subref_id`, `week_day` and `work_day`. This means that the elements in $\hat{c}_1$ are mostly similar according to these attributes, as confirmed by the charts in Fig. 4a. Almost every call comes from the prefecture 60 during Saturday, which is a weekend day. Even if most of the calls are done in the morning and the most frequent time interval from the previous call is 6h, there is an high variance in the values of the attributes as it can be seen from the pie chart, hence the features `conn_time`, `day_period` and `prev_call` are often neglected, by being set to 0 in the related weights of the PCs.

For what concerns the remaining patterns in the CDRs of User 4, the second meta-cluster in Fig. 4b highlights the habit of User 4 of performing calls on Tuesday, mostly in the afternoon, from prefecture 60. The third pattern in Fig. 4c is related to the calls done in the weekend evening, mostly Saturday from

prefecture 60. Finally, the 4-th meta-cluster in Fig. 4d represents a recurrent pattern in the telephonic activity of the user, who often calls on Thursday from prefecture 60, mostly in the afternoon.

We can appreciate an important feature of LD-ABCD, by noticing that there are no patterns related, for example, to calls done on Monday, Wednesday or Friday, nor calls done in the evening. This is because there are no strong regularities in the calls involving these days or this period *in conjunction* with other features. In fact, we recall from Sect. 4.1 that in LD-ABCD at least 2 elements in the PC must be different from 0, in order to prevent the generation of clusters containing elements similar only with respect to a single feature. For this reason, trivial clusters of CDRs sharing only the day of the week are never considered.

In the following we report another example of analysis, relative to user 6014. Like in the previous case, we plot a visual representation of the meta-clusters using the PCA (see Fig. 5) and we identify the most dense regions, which are 3 in this case. From each region we select the meta-clusters with highest CQ value (brightest red color) and we consider them as the representatives of the related 3 recurrent patterns. Again, to describe the content of each meta-cluster and, consequently, the semantic of the related pattern, we use the pie charts in Fig. 6.

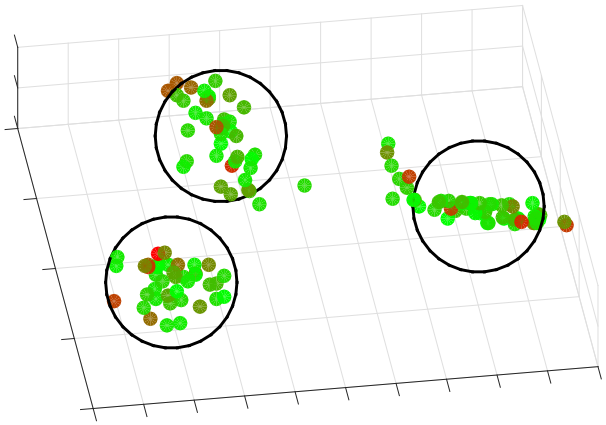
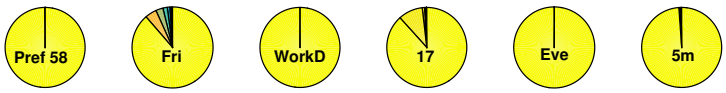


Figure 5: Plot of the first 3 principal component scores, resulting from a PCA on the meta-clusters found for user 6014. In this case there are three dense regions, marked with a black circle.

In this case, the first meta-cluster in Fig. 6a represents the habit of User 6014 of calling from prefecture 58 in the evening of working days after 5 minutes from the previous call, mostly at 17:00 on Friday. The second pattern in Fig. 6b highlights that the user issues calls in the weekends, during the morning, from prefecture 58, mostly Saturday. Finally, the third meta-cluster in Fig. 6c shows that the user often calls on Monday morning from prefecture 58, mostly after more than 24 hours from the



(a) Meta-cluster $\hat{c}_1$

(b) Meta-cluster $\hat{c}_2$

(c) Meta-cluster $\hat{c}_3$

Figure 6: The set of the 6 pie charts of the meta-clusters that represent the 3 relevant patterns among the CDRs of User 6014.

previous call.

We conclude with a consideration on the stability of the results found by LD-ABCD. For each user, we repeated several times the clusters mining procedure and we observed that the number and average quality of the meta-clusters returned in different runs is always the same. This is a valuable result, since the algorithm possess a strong stochastic component, due to the nature of the random walk and the probabilistic selection of new PCs. If the results returned can be repeated it means that the solution is stable and complete, in the sense that most of the recurrent patterns are identified in each run of the algorithm.

5.2 Analysis Results with PROCLUS

In the case of study under consideration, which is the identification of relevant and recurrent patterns among the CDRs of a given user, there is not a *ground truth* to which we can refer for evaluating the quality of a solution. Thereby, in order to appraise the effectiveness of the knowledge discovery system based on the LD-ABCD algorithm, we perform a comparison with the results returned by the procedure based on PROCLUS on the same dataset. In particular, we compare the patterns found by the two algorithms.

As described in Sect. 4.2, PROCLUS requires the user to specify the number k of clusters to be searched and the average number of dimensions l that must be identified in each cluster. A correct tuning of these parameters requires an *a-priori* knowledge of the problem and of the dataset, which we do not possess. For this reason and for making a more significant comparison between the two algorithms, we set k equal to the number of dense regions found by LD-ABCD in the space of the meta-clusters, after having applied the PCA dimensionality reduction. The parameter l is estimated by considering the average number of features that have been selected by LD-ABCD, i.e. how many features are, in average, set equal to 1 in the PCs associated to each meta-cluster. For what concerns the parameter `minDev` that controls the number of points in a cluster, we followed the recommendations provided in [2] by the authors of the procedure.

According to the results of LD-ABCD relative to User 4, we set $k = 4$ and $l = 5$. Like for LD-ABCD, the result of PROCLUS is not deterministic because of the stochastic nature of the initialization procedure. However, it has been proven that in datasets with well defined clusters, each one with a specific set of characterizing dimensions, the results returned in different runs were stable and very similar [54].

We performed a total of 200 different runs on the dataset of the CDRs of User 4, using the same values for the parameters k and l, and we analyzed the results obtained.

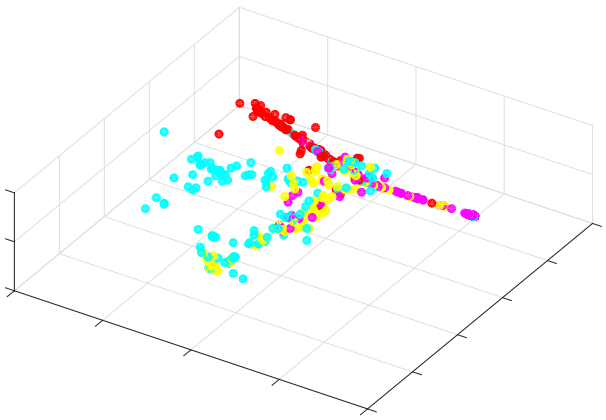


Figure 7: The first 3 principal components, resulting from a PCA on the 4 clusters returned by PROCLUS in 200 different runs. For each run, we plotted the 4 clusters found using the same set of colors. Dots of the same colors represent the set of clusters, each one coming from a distinct run, which are the most similar in terms of constituent elements.

As in the case of LD-ABCD, in Fig. 7 we plotted the first 3 principal component scores and we colored differently each one of the 4 clusters returned by a given run, assigning the same color to the

most similar clusters among different runs. More specifically, we retrieve the 4 clusters returned by the first run and we assign them 4 different colors: yellow, red, purple and turquoise. Then, in the successive runs we matched the 4 clusters returned with the ones found in the first run. We paired each new cluster with the most similar cluster (in terms of percentage of shared elements) from the first run using a Best Match First heuristic [18, 12] and we assigned the same 4 colors to the new clusters, according to this match. As we can see from Fig. 7, even if clusters of the same color (which are the most similar with respect to the matching) are mostly located in the same areas, we cannot distinguish 4 distinct dense groups and, especially in the middle, the clusters are strongly overlapped and mixed. This means that in different runs PROCLUS is not able to identify the same set of clusters, like LD-ABCD does.

Another symptom of the instability of the solution returned is the high variance of the number of outliers that the algorithm finds in different runs. In Fig. 8 we plotted the percentage of outliers found in each run and the average value. We can observe that there is a very high variance in the number of elements that the algorithm recognizes as outliers. If the composition of the outliers set changes significantly in each run, it means that the structure of the remaining clusters varies as well. Thus, there is not a stable, unique solution identified by the algorithm.

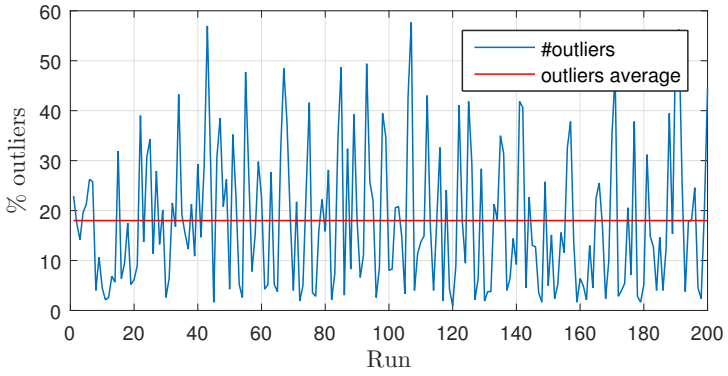


Figure 8: Number of outliers, expressed as a percentage of the total number of elements in the dataset, identified by PROCLUS in 200 different runs; in red is plotted the average value. As we can observe the number of outliers varies considerably from run to run, and in some cases it reaches very high values, up to 60 % of the elements in the dataset.

Several clusters found by PROCLUS results to be accurate, in terms of the similarity of the attributes of the elements in each cluster. However such clusters are generally small and they are returned in runs where an high percentage of elements are classified as outliers. In larger cluster instead, the values in the attributes of the contained elements are generally more heterogeneous, even if the PROCLUS selects such attributes as relevant dimensions in the cluster. As an example, in the first row of Fig. 9 we depict

(a) Meta-cluster A

(b) Meta-cluster B

Figure 9: Two different clusters identified by PROCLUS. In the first row there are the pie charts relative to a cluster A of large size and, as we can observe, the attributes on the different values are very heterogeneous among the elements in the cluster. In the second row a cluster B is represented with graphics that show a higher accuracy on the attributes, but the size of this cluster is significantly smaller.

the pie charts relative to a large cluster A, whose dimension is comparable in size with the clusters found by LD-ABCD. However, as we can see from the charts, the values of different attributes among

the elements in cluster A are very heterogeneous. On the second row instead, the pie charts represent a cluster B which contains elements with more homogeneous values on the relevant attributes, but the dimension of cluster B is much smaller, since it contains approximately 7% of the total CDRs and it comes from a run with a very high percentage of outliers, which is 23% of the entire dataset.

6 Conclusions and Future Works

In this paper we applied our recently developed knowledge discovery algorithm LD-ABCD, as the core engine for a data mining analysis on CDRs. The objective of this work is to build profiles of the users, which can be employed by telecommunication companies for monitoring and understanding the behaviors of their subscribers. A suitable characterization of the customer base allows to use with greater effectiveness the information resources of a telecom operator, to develop marketing strategies and for tailoring telephonic plans which better suit the needing of the users. Furthermore, a series of applications can be designed for monitoring the activity of a user, focused on the identification of anomalous and suspicious behaviors, which are the ones which differs from the usual subscribers' patterns of habits and that could compromise his privacy or security.

We analyzed the CDRs from the dataset of the D4D challenge, which contains only two type of values, concerning the time and the geographical position from where the calls were issued. We showed how additional features can be derived from these original attributes and how regularities can be extracted from a dataset which apparently contain a very limited quantity of information.

LD-ABCD is an agent-based algorithm that identify regularities and recurrent patterns among data. When applied to a dataset of CDRs, it can be used to identify habits in telephonic activity of the users, in order to create their "digital-fingerprints". One of most important features of the LD-ABCD, in our applicative context, is the possibility of identifying multiple parameter configurations, which highlight the characteristics of patterns within a cluster that are considered to be discriminative. Such configurations represent the key for interpreting and characterizing semantically the regularities found in the dataset. Another important advantage of LD-ABCD, with respect to other approaches to cluster analysis, is that the number k of clusters to be identified is not an input parameter of the algorithm, but it is automatically identified during the discovery procedure. This is particularly useful when there are no information on the number of possible clusters to be identified in the data, like in our case of study.

We compared the results of the knowledge discovery system based on LD-ABCD with an alternative implementation based on PROCLUS, the well-known subspace clustering algorithm capable of identifying clusters in a subset of the original feature-space. Both LD-ABCD and PROCLUS share the important characteristic of being able to identify a local metric for each retrieved cluster. We discussed the results of the analysis considering the CDRs of a specific user, applying the knowledge discovery systems based on LD-ABCD and PROCLUS and we showed how LD-ABCD is capable of identifying a set of patterns which can be semantically characterized. The result returned by LD-ABCD demonstrated to be stable and reliable, even when the nature of the data is completely unknown and when the presence of well-defined clusters is not clear. This is also a consequence of the low sensitivity of LD-ABCD to different settings of the configuration parameters (τ_{exp}, τ_{CQ} and ϑ), compared to the case of PROCLUS, where different choices of k and l could significantly modify the results. Additionally, as we discussed in Sect. 5, the most critical parameter τ_{exp} can be easily tuned, using an heuristic procedure, which demonstrated to be effective in different contexts [8].

On the other side, even if LD-ABCD can be easily implemented on a distributed computing network, due to the multi-agent architecture, the main disadvantage of the procedure is the high computational resources required, in terms of both space and time. For this reason, when the nature of the problem faced is simpler or when prior information on the problem are provided, many other alternatives can be considered for implementing the core engine of a data mining procedure. The difficulty of the considered problem has been confirmed by the results returned by PROCLUS, which was not able to identify a stable set of clusters among the data. In fact, from the experiments we observed that PROCLUS, in different execution of the clustering procedure, was not able to identify the same set of clusters, showing an high variance and instability in the results returned.

The results obtained confirm the effectiveness of the LD-ABCD and they encourage further applications. In a future work, we firstly plan to process the CDRs of every user in the dataset, searching for common patterns and regularities in order to group together users characterized by a similar profile.

Then, in a second step, with the identified clusters of users we aim to define specific classes, which can be analyzed and used for describing some general, common behaviors that allow a better understanding of the habits of the customer base of a telecommunication company. Important information can be assessed concerning the geographical location of the user and a study can be conducted on how the habits change in different areas. Finally, we plan to consider additional datasets relative to a telecommunication network, which are characterized by a large amount of entries and a higher number of features (for example, those included in the Telecom Italia BigData Challenges [1, 48]).

References

[1] Telecom Italia. Telecom Italia big data challenge 2014, 2014. Accessed: 2015-11-16.

[2] C. C. Aggarwal, J. L. Wolf, P. S. Yu, C. Procopiuc, and J. S. Park. Fast algorithms for projected clustering. *SIGMOD Rec.*, 28:61–72, June 1999.

[3] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. *Automatic subspace clustering of high dimensional data for data mining applications*, volume 27. ACM, 1998.

[4] R. Ahas, S. Silm, O. Järv, E. Saluveer, and M. Tiru. Using mobile positioning data to model locations meaningful to users of mobile phones. *Journal of Urban Technology*, 17(1):3–27, 2010.

[5] M. Bereta, W. Pedrycz, and M. Reformat. Local descriptors and similarity measures for frontal face recognition: A comparative analysis. *Journal of Visual Communication and Image Representation*, 24(8):1213–1231, 2013.

[6] M. Berlingerio, F. Calabrese, G. Di Lorenzo, R. Nair, F. Pinelli, and M. L. Sbodio. AllAboard: a system for exploring urban mobility and optimizing public transport using cellphone data. In *Machine Learning and Knowledge Discovery in Databases*, pages 663–666. Springer, 2013.

[7] D. M. Berry. The computational turn: Thinking about the digital humanities. *Culture Machine*, 12(0), 2011.

[8] F. Bianchi, E. Maiorino, L. Livi, A. Rizzi, and A. Sadeghian. An agent-based algorithm exploiting multiple local dissimilarities for clusters mining and knowledge discovery. *Soft Computing*, pages 1–23, 2015.

[9] F. M. Bianchi, E. De Santis, A. Rizzi, and A. Sadeghian. Short-term electric load forecasting using echo state networks and PCA decomposition. *IEEE Access*, 3:1931–1943, Oct. 2015.

[10] F. M. Bianchi, L. Livi, and A. Rizzi. Matching of time-varying labeled graphs. In *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pages 1–8. IEEE, 2013.

[11] F. M. Bianchi, L. Livi, A. Rizzi, and A. Sadeghian. A Granular Computing approach to the design of optimized graph classification systems. *Soft Computing*, 18(2):393–412, 2014.

[12] F. M. Bianchi, S. Scardapane, A. Rizzi, A. Uncini, and A. Sadeghian. Granular computing techniques for classication and semantic characterization of structured data. *Cognitive Computation*, Dec 2015.

[13] F. M. Bianchi, S. Scardapane, A. Uncini, A. Rizzi, and A. Sadeghian. Prediction of telephone calls load using Echo State Network with exogenous variables. *Neural Networks*, 71:204–213, 2015.

[14] V. D. Blondel, M. Esch, C. Chan, F. Clérot, P. Deville, E. Huens, F. Morlot, Z. Smoreda, and C. Ziemlicki. Data for development: the D4D challenge on mobile phone data. *CoRR*, abs/1210.0137, 2012.

[15] F. Calabrese, M. Colonna, P. Lovisolo, D. Parata, and C. Ratti. Real-time urban monitoring using cell phones: A case study in rome. *Intelligent Transportation Systems, IEEE Transactions on*, 12(1):141–151, 2011.

[16] J. Candia, M. C. González, P. Wang, T. Schoenharl, G. Madey, and A.-L. Barabási. Uncovering individual and collective human dynamics from mobile phone records. *Journal of Physics A: Mathematical and Theoretical*, 41(22):224015, 2008.

[17] C.-C. Chang. A boosting approach for supervised Mahalanobis distance metric learning. *Pattern Recognition*, 45(2):844–862, 2012.

[18] G. Del Vescovo and A. Rizzi. Automatic classification of graphs by symbolic histograms. In *Granular Computing, 2007. GRC 2007. IEEE International Conference on*, pages 410–410, Nov 2007.

[19] K. Demirli and P. Muthukumaran. Higher order fuzzy system identification using subtractive clustering. *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, 9(3, 4):129–158, 2000.

[20] T. H. Duong, N. T. Nguyen, and G. S. Jo. Constructing and mining a semantic-based academic social network. *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, 21(3):197–207, 2010.

[21] N. Eagle, A. S. Pentland, and D. Lazer. Inferring friendship network structure by using mobile phone data. *Proceedings of the National Academy of Sciences*, 106(36):15274–15278, 2009.

[22] U. M. Fayyad, A. Wierse, and G. G. Grinstein. *Information visualization in data mining and knowledge discovery*. Morgan Kaufmann, 2002.

[23] B. Furletti, L. Gabrielli, C. Renso, and S. Rinzivillo. Analysis of gsm calls data for understanding user mobility behavior. In *Big Data, 2013 IEEE International Conference on*, pages 550–555. IEEE, 2013.

[24] H. Gao, J. Tang, and H. Liu. Mobile Location Prediction in Spatio-Temporal Context. *the Procedings of Mobile Data Challenge by Nokia Workshop at the Tenth International Conference on Pervasive Computing*, June 2012.

[25] G. Gidófalvi and F. Dong. When and where next: individual mobility prediction. In *Proceedings of the First ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems*, pages 57–64. ACM, 2012.

[26] M. Gönen and E. Alpaydın. Multiple kernel learning algorithms. *The Journal of Machine Learning Research*, 12:2211–2268, 2011.

[27] P. Grindrod. Infering behavior-based lifestyle categorizations based on mobile phone usage data, Mar. 15 2013. US Patent App. 13/841,852.

[28] D. Hu, F. Sun, L. Tu, and B. Huang. We know what you are–a user classification based on mobile data. In *Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing*, pages 1282–1289. IEEE, 2013.

[29] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[30] N. Japkowicz. *Concept-learning in the absence of counter-examples: an autoassociation-based approach to classification*. PhD thesis, Rutgers, The State University of New Jersey, 1999.

[31] R. Kannan, S. Vempala, and A. Vetta. On clusterings: Good, bad and spectral. *Journal of the ACM (JACM)*, 51(3):497–515, 2004.

[32] S.-W. Kim and R. P. W. Duin. A Combine-Correct-Combine Scheme for Optimizing Dissimilarity-Based Classifiers. In E. Bayro-Corrochano and J.-O. Eklundh, editors, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, volume 5856 of *LNCS*, pages 425–432. Springer Berlin Heidelberg, 2009.

[33] J. Komorowski and J. Zytkow. *Principles of data mining and knowledge discovery*. Springer, 1997.

[34] D. Lazer, A. S. Pentland, L. Adamic, S. Aral, A. L. Barabasi, D. Brewer, N. Christakis, N. Contractor, J. Fowler, M. Gutmann, et al. Life in the network: the coming age of computational social science. *Science (New York, NY)*, 323(5915):721, 2009.

[35] B. Li, E. Chang, and C.-T. Wu. DPF-a perceptual distance function for image retrieval. In *Proceedings of the 2002 International Conference on Image Processing.*, volume 2, pages II–597. IEEE, 2002.

[36] A. Lima, M. De Domenico, V. Pejovic, and M. Musolesi. Exploiting cellular data for disease containment and information campaigns strategies in country-wide epidemics. *arXiv preprint arXiv:1306.4534*, 2013.

[37] E. Olsson, P. Funk, and N. Xiong. Fault diagnosis in industry using sensor readings and case-based reasoning. *Journal of Intelligent & Fuzzy Systems*, 15(1):41–46, 2004.

[38] M. Özer. *PREDICTING THE LOCATION AND TIME OF MOBILE PHONE USERS BY USING SEQUENTIAL PATTERN MINING TECHNIQUES*. PhD thesis, Middle East Technical University, 2014.

[39] L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: a review. *ACM SIGKDD Explorations Newsletter*, 6(1):90–105, 2004.

[40] W. Pedrycz. Proximity-Based Clustering: A Search for Structural Consistency in Data With Semantic Blocks of Features. *IEEE Transactions on Fuzzy Systems*, 21(5):978–982, 2013.

[41] S. Queiroz, F. d. A. T. de Carvalho, and Y. Lechevallier. Nonlinear multicriteria clustering based on multiple dissimilarity matrices. *Pattern Recognition*, 46(12):3383–3394, 2013.

[42] J. Schlaich, T. Otterstätter, and M. Friedrich. Generating trajectories from mobile phone data. In *Proceedings of the 89th Annual Meeting Compendium of Papers, Transportation Research Board of the National Academies*, 2010.

[43] C. Shen, J. Kim, F. Liu, L. Wang, and A. van den Hengel. Efficient Dual Approach to Distance Metric Learning. *IEEE Transactions on Neural Networks and Learning Systems*, 25(2):394–406, Feb 2014.

[44] C. Smith, A. Mashhadi, and L. Capra. Ubiquitous sensing for mapping poverty in developing countries. *Paper submitted to the Orange D4D Challenge*, 2013.

[45] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010.

[46] E. J. Spinosa, A. P. de Leon F. de Carvalho, and J. a. Gama. Olindda: A cluster-based approach for detecting novelty and concept drift in data streams. In *Proceedings of the 2007 ACM Symposium on Applied Computing*, SAC '07, pages 448–452, New York, NY, USA, 2007. ACM.

[47] R. Taormina and K. Chau. Neural network river forecasting with multi-objective fully informed particle swarm optimization. *J. Hydroinform*, 17:99–113, 2015.

[48] Telecom Italia. Telecom Italia big data challenge 2015, 2015. Accessed: 2015-11-16.

[49] S. van den Elzen, J. Blaas, J. J. van Wijk, and R. Spousta. Exploration and Analysis of Massive Mobile Phone Data: A Layered Visual Analytics approach. to apper, May 2013.

[50] W.-c. Wang, K.-w. Chau, D.-m. Xu, and X.-Y. Chen. Improving forecasting accuracy of annual runoff time series using arima based on eemd decomposition. *Water Resources Management*, 29(8):2655–2675, 2015.

[51] R. R. Yager and D. P. Filev. Generation of fuzzy rules by mountain clustering. *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, 2(3):209–219, 1994.

[52] L. Yang, R. Jin, L. Mummert, R. Sukthankar, A. Goode, B. Zheng, S. C. H. Hoi, and M. Satyanarayanan. A Boosting Framework for Visuality-Preserving Distance Metric Learning and Its Application to Medical Image Retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):30–44, Jan 2010.

[53] X. Yin, T. Shu, and Q. Huang. Semi-supervised fuzzy clustering with metric learning and entropy regularization. *Knowledge-Based Systems*, 35:304–311, 2012.

[54] M. L. Yiu and N. Mamoulis. Frequent-pattern based iterative projected clustering. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 689–692. IEEE, 2003.

[55] H. Zhang, J. Yu, M. Wang, and Y. Liu. Semi-supervised distance metric learning based on local linear regression for data clustering. *Neurocomputing*, 93:100–105, 2012.

[56] J. Zhang, K.-W. Chau, et al. Multilayer ensemble pruning via novel multi-sub-swarm particle swarm optimization. *J. UCS*, 15(4):840–858, 2009.

[57] S. Zhang and K.-W. Chau. Dimension reduction using semi-supervised locally linear embedding for plant leaf classification. In *Emerging Intelligent Computing Technology and Applications*, pages 948–955. Springer, 2009.