Social Network Analysis and Mining manuscript No.
(will be inserted by the editor)

Dynamic Communities in Evolving Customer Networks: An Analysis using Landmark and Sliding Windows

Márcia Oliveira · Américo Guerreiro ·
João Gama (✉)

Received: date / Accepted: date

Abstract The widespread availability of Customer Relationship Management applications in modern organizations, allows companies to collect and store vast amounts of high-detailed customer-related data. Making sense of these data using appropriate methods can yield insights into customers' behaviour and preferences. The extracted knowledge can then be explored for marketing purposes. Social Network Analysis techniques can play a key role in business analytics. By modelling the implicit relationships among customers as a social network, it is possible to understand how patterns in these relationships translate into competitive advantages for the company. Additionally, the incorporation of the temporal dimension in such analysis can help detect market trends and changes in customers' preferences. In this paper, we introduce a methodology to examine the dynamics of customer communities, which relies on two different time window models: a landmark and a sliding window. Landmark windows keep all the historical data and treat all nodes and links equally, even if they only appear at the early stages of the network life. Such approach is appropriate for the long term analysis of networks, but may fail to provide a realistic picture of the current evolution. On the other hand, sliding

M. Oliveira
FEP, School of Economics and Business, University of Porto and LIAAD/INESC TEC
E-mail: mdbo@inescporto.pt
Rua Dr. Roberto Frias, 4200-464 Porto, Portugal

A. Guerreiro
FEP, School of Economics and Business, University of Porto
E-mail: americo_guerreiro@hotmail.com
Rua Dr. Roberto Frias, 4200-464 Porto, Portugal

J. Gama (✉)
FEP, School of Economics and Business, University of Porto and LIAAD/INESC TEC
E-mail: jgama@fep.up.pt
Rua Dr. Roberto Frias, 4200-464 Porto, Portugal
Tel.: +351 22 209 4057
Fax: +351 222 094 050

windows focus on the most recent past thus allowing to capture current events. The application of the proposed methodology on a real-world customer network suggests that both window models provide complementary information. Nevertheless, the sliding window model is able to capture better the recent changes of the network.

Keywords Customer Networks · Dynamic Community Mining · Social Network Analysis · Time Window Models

1 Introduction

The scientific and technological advances of the last decades permeated virtually every facet of our everyday lives, revolutionizing the way how people interact, communicate, work, buy and access information. These advances also shaped the market and, as a consequence, how business organizations operate and relate with their customers. The proliferation of competitors and the weakening effectiveness of traditional marketing promotional campaigns had led companies to evolve from product-centered strategies to customer-centered strategies. The widespread availability of *Customer Relationship Management* (CRM) applications in modern organizations, allowed companies to collect and store vast amounts of high-detailed customer-related data (e.g. purchasing habits, values of proposals, demographic variables). Making sense of these data using appropriate methods can yield insights into customers' behaviour and preferences. The extracted knowledge can then be used to support the redesign of marketing promotions tailored to each individual customer, or to a group of customers showing similar purchasing behaviour/preferences. This kind of analysis can be effectively performed using *Social Network Analysis* (SNA) techniques, by modelling the implicit relationships among customers as a social network. We define *customer network* as a finite set of customers who are linked to each other if they bought at least one similar product during a given timeframe. Modelling customer-related data using this type of customer networks has the advantage of revealing implicit relationships among customers based on their purchasing behaviour over a given time period. Furthermore, since purchasing events are annotated with timestamps, it is possible to extract the network state at different moments in time, thus enabling the study of customer network evolution. Additionally, if we analyse these dynamics at the community-level, we may be able to identify evolutionary profiles of groups of customers that show similar purchasing behaviour.

Albeit the origins of network studies go back a few centuries ago, in recent years we witnessed an impressive advance in network-related fields, especially in computer science and computational physics. Until recently the analysis of such networks was mainly a static investigation of the aggregated graph of the network across multiple snapshots (Takaffoli et al, 2011). Nonetheless, one of the key features of many networks is that their structure evolves over time, so approaches focusing on the analysis of a fixed snapshot of the network may fail to capture the dynamics of the evolving network.

Since the formation and changes undergone by communities reflect the dynamics at the whole network, methodologies to model and track the life-cycle of communities within dynamic social networks have been developed (Falkowski et al, 2006; Palla et al, 2007; Lin et al, 2009; Asur et al, 2009; Greene et al, 2010; Takaffoli et al, 2011; Bródka et al, 2013). Common approaches involve detecting communities at different stages of the evolving network, by applying a suitable community detection algorithm to each snapshot of the network that has been accumulated over the time span. Hence, these approaches usually consider the whole historical data at hand when performing the analysis, and typically assign the same weight to nodes and links, even if they were only active at a remote time point. Thus, methodologies relying on accumulated windows (aka *landmark windows*) can only discover small changes in communities in consecutive timeframes and any drastic change in short time may potentially remain undetected. Unlike landmark windows, sliding windows focus on the most recent state of the dynamic network when analysing their evolution, thus being able to capture more up-to-date events, especially when dealing with volatile networks.

Despite the significant body of literature addressing the problem of dynamic network analysis (see Berger-Wolf et al (2010), for an overview of the topic), to the best of our knowledge few work to date has explicitly explored the effect of selecting different time window models (also referred to as *time-frame types* in the literature) on both the stability of dynamic communities and the type of information provided by each approach. The most related work to ours is the one by Saganowski et al (2012). In this work, the authors carry out an empirical study of the influence of several time window models (e.g. sliding window with no overlap, overlapping sliding window, landmark window) on the number of detected community events (e.g. forming, shrinking, merging, splitting). Based on their experiments, they conclude that the choice of the granularity at which the time varying network is snapshotted impacts the results of the method used to extract community dynamics. While landmark windows prove to be useful to detect stable communities, the sliding window model with overlap is more suitable for extracting community evolution in rapidly changing social networks. On the other hand, extracting network snapshots for disjoint timeframes precludes the creation of complete evolutionary profile of communities, since the detected changes are too fast (e.g. formations followed by dissolutions). In our case study, we draw similar conclusions. Another relevant research is the one by Falkowski et al (2006), who developed a two-pronged methodology to analyse the evolution of two types of dynamic communities in social networks: communities with rather stable membership structure and communities with high fluctuation of members. For both scenarios, they use an overlapping sliding window approach to obtain the snapshots of the underlying network. Asur et al (2009) make use of mutually exclusive temporal snapshots to examine static versions of an evolving interaction graph at different time points. On the other hand, the work by Greene et al (2010) on the same topic suggests that the size of the timestep window can influence the obtained results, especially if the network structure is unstable. Kawadia and Sreenivasan (2012) also stress out the importance of determining the granularity of the temporal snapshots for the purpose of detecting temporal communities and argue that there is a natural multiscale of interest, driven by the application, for generating these snapshots. However, none of these works compared distinct time window models, from the point of view of dynamic community mining, within the scope of a real-world marketing application.

To partially fill this gap, this paper proposes an application-driven methodology to study the community structure of dynamic social networks based on two different window models. Our contributions are threefold: (a) application of dynamic community mining in a real-world customer network for the purpose of identifying different evolutionary profiles of customers; (b) use of a sliding window model in the study of community evolution as a complement to the conventional landmark window model and, (c) extension of a previously proposed event-based framework for monitoring clusters dynamics, dubbed MEC (Oliveira and Gama, 2012), to tackle the problem of community evolution.

The paper proceeds as follows. Section 2 provides the necessary background on social network analysis, community detection, dynamic community mining and window models. In Section 3, we outline the proposed methodology and provide a detailed description of the extended MEC, termed MECnet. Our case study on a real-world evolving customer network is presented in Section 4. Section 5 concludes the paper.

2 Background

2.1 Social Network Analysis

SNA is a quantitative methodology whose development significantly benefited from the collaborative efforts of researchers from different scientific areas (e.g. sociology, physics, computer science). SNA offers a powerful means to model, describe and analyse network structures, groups of nodes (i.e. communities) and single nodes by focusing explicitly on the relationships established between them (Wasserman and Faust, 1994). The focus on the relationships rather on the entities themselves is a fundamental axiom in SNA. This axiom stresses the notion that nodes are not independent but rather influence each other. Typical tasks in SNA involve the identification of the most prominent nodes, the estimation of their roles within the overall network structure, community detection, link prediction and the discovery of persistent patterns of relationships and emergent properties that help explain network formation and growth. Several SNA metrics were proposed to assess the overall structure of social networks and to measure the centrality of single nodes. The former encompasses metrics such as density, clustering coefficient, diameter, average geodesic distance and average degree. The latter includes metrics such as degree, betweenness, closeness, eigenvector centrality and local clustering.

We will make use of these metrics to obtain a description of the network at different stages of its evolution.

2.2 Community Detection

One of the unique features of real social networks is that they tend to show *community structure* (Newman, 2003). These mesoscopic structures usually arise as a consequence of both global and local heterogeneity of links' distribution in a graph. Thus, we often find in networks tightly connected groups of nodes, termed *communities*, which are sparsely connected to other densely connected groups. The task of *community detection*, which aims at finding meaningful group structures in networks, is itself an important strand of research on the field of SNA and a significant number of methods and algorithms have been proposed for this purpose (for a thorough review, please refer to Fortunato (2010)). In this paper, we resort to the Louvain method (Blondel et al, 2008) to detect communities, although we also perform experiments with the Label Propagation algorithm (LP) (Raghavan et al, 2007).

The Louvain method is a greedy optimization method that performs a hierarchical *modularity* (Newman and Girvan, 2004) optimization. This method comprises two phases. The first phase optimizes modularity in a local way by looking for positive gains in modularity when moving a node to a neighboring community. The second phase is similar to the first one, with the difference that now we deal with a modified network, where each vertex (or node) is a supervertex, which represents the previously found communities. Considering this higher-level setting, the steps of the first phase are repeated iteratively until a maximum of modularity is attained and new hierarchical levels and supergraphs are yielded. The algorithm stops when modularity converges to a value where no more gains are possible. This method produces good quality partitions in a very fast way.

The Label Propagation algorithm is another commonly used method for extracting communities from networks that relies on the network structure alone to find densely connected groups of nodes. The basic idea behind LP is to explore the information diffusion enabled by the network structure to identify consensual groups of nodes' labels. It starts by labeling every node with unique labels. Then, through an iterative process, the labels are updated by majority voting on the neighbourhood of the node. When this process stops, the communities will correspond to sets of nodes sharing the same label. LP has similarities with the Louvain method in the sense that it is computationally efficient and does not require any *a priori* information about the communities (e.g. number or size of communities, central nodes) to operate. However, while the Louvain method is modularity-based, relying on a two-stage hierarchical modularity optimization to detect communities, the LP algorithm does not require the optimization of a pre-defined objective function to identify network partitions.

2.3 Dynamic Community Mining

The structure of most networks (e.g. co-authorship and friendship networks) are dynamic in nature as they tend to evolve gradually, through the addition and deletion of links and nodes. As a consequence, substructures inside the network, such as communities, also change over time. Communities are unstable patterns that can evolve in both membership and content. In dynamic scenarios, communities may undergo a series of evolutionary events, such as growth, split and disappearance, which characterize their *life-cycle*. For instance, a community at time point t_i may separate into several communities in time point $t_{(i+1)}$, if the former community splits into two or more communities.

Past research on community mining discarded the temporal information by modelling the dynamic network as a static graph. This whole graph would depict all the nodes and links observed in a given time span. The most widespread approach to incorporate the dynamics into the study of communities is to convert the evolving network into an ordered sequence of static snapshots, each representing the state of the network at a given point in time. Recent work proposes to characterize the evolution of a given community by describing its life-cycle, i.e. a series of critical events undergone by communities over time (Palla et al, 2007; Lin et al, 2009; Asur et al, 2009; Greene et al, 2010; Takaffoli et al, 2011; Bródka et al, 2013). Typically, these approaches rely on a landmark window model for the process of extracting the snapshots of the evolving network, since they take into consideration all the historical network data collected up to the current time point. Alternative models, such as the sliding window, are embedded with forgetting mechanisms that drop older data from the analysis thus allowing to uncover the most recent changes occurring in the network. Hitherto, few work explored these alternative models, although they may prove more useful than accumulated windows in detecting changes in rapidly evolving networks.

2.4 Window Models

2.4.1 Landmark Windows

Landmark windows (Gehrke et al, 2001) encompass all the data from a specific point in time up to the current moment. This model is initialized by first selecting a fixed time point (the so-called *landmark*), which marks the beginning of the time window, and then it grows the window by considering all the data seen so far after the landmark. In the dynamic social networks setting, a landmark window will aggregate the network data (e.g. nodes and links) observed over the entire period of observation. By keeping track of all the connections and nodes in the network, this approach does not entail loss of information. However, since it relies on the accumulation of data over time, it is not very well suited to find current trends. Thus, recent interesting phenomena may go unnoticed due to the smoothing effect on data changes occurring over time.

2.4.2 Sliding Windows

Unlike landmark windows, the sliding window model (Datar et al, 2002) incorporates a time-based forgetting mechanism, keeping only the latest information inside the window and disregarding all the data falling outside the window. The simplest approach are sliding windows of fixed length. The window length is a user-defined parameter which influences the amount of data taken into consideration in the model. The *time-based length* sets the window length as a fixed time span. By deeming only the most recent past, this model proves useful in finding current trends. As a drawback, it might be hard to determine the right parameter settings. It is important to note that there is a trade-off between the window length and the ability to capture changes. Small windows will capture rapid changes but lose information (memory) about network stability. Within the scope of our work, this kind of window configuration is reflected in a set of network snapshots representing the state of the network for a sequence of fixed, typically short, timeframes. Due to its forgetting mechanism, this approach provides a more up-to-date representation of the network, thus allowing to capture the most current events, which would otherwise be smoothed out by the whole historical data accommodated in a landmark window.

3 Methodology

In this section, we detail our methodology for analysing the structure of evolving customer networks, in terms of community evolution. The motivation for developing this methodology was driven by the need of one of largest Portuguese companies in the electric field to tap the potential of the customer-related data they have been accumulating over time, using SNA techniques. The idea is to carry out an exploratory analysis of the implicit customer network, in order to identify profiles of customers (given in the form of *communities*) and their evolution over a given time period. This information can be further explored for the design of marketing promotions tailored to each profile.

The proposed methodology is comprised of three main sequential steps that are independently applied to two different window models. The main steps are:

1. Analysis and description of the dynamic network, at both the network-level and node-level, using well-known SNA metrics;
2. Application of our extended community evolution framework, dubbed MEC-net, to each window setting;
3. Interpretation of the dynamics of customers' communities (or profiles)

Finally, the results of these three main steps for each window model are compared.

3.1 Network Analysis

Network analysis is performed by computing and interpreting popular SNA metrics. The interpretation of these metrics provide us insight about the structure of the network and the role of each node in the network, without the need to look at its graphical representation. They are usually divided according to the level of analysis one wants to perform: at the level of the basic entities (nodes) or at the level of the whole network. The former measures how a single node is embedded in a network from that single node's perspective. The latter computes how the overall network links are organized from the perspective of an observer that has a bird's eye view of the network. In this methodology we resort to both levels of analysis to get a description of the network. At the node-level, we analyse the following metrics for undirected networks: *eigenvector centrality* and *betweenness*. At the network-level, we focus on *density* and *modularity* (Newman and Girvan, 2004).

3.2 MECnet for Tracking Community Evolution

MECnet is the term we use to refer to the extension of MEC to the community evolution setting. MEC can be easily extended to deal with communities, which is the equivalent of clusters for networks, due to its relative independence from the algorithm used to extract clusters/communities. We say *relative independence*, in the sense that MEC is not restricted to a single clustering algorithm, although it requires that the adopted algorithm partitions the data into disjoint groups (i.e. each object/node is assigned to a single cluster/community). MEC is a framework proposed by Oliveira and Gama (2012) to monitor the evolution of clusters. MEC traces evolution through the detection and categorization of clusters transitions, such as births, splits and merges. The clusters are extensionally defined, i.e. each cluster is defined by the objects that were assigned to it by a given clustering algorithm. It takes as input a set of clusterings, each one generated at a different time point. It performs pairwise mappings, between clusters obtained at time point t_i ($i = 1, ..., T$, with T denoting the last analysed time point) and at a later time point $t_{i+\Delta t}$. The *mapping* process explores the concept of conditional probability and is restricted by a user-defined threshold - the *survival threshold* τ, where $\tau \in [0.5, 1]$. This threshold indicates the proportion of mutual objects two cluster instances have to share in order for them to be considered instances of the same cluster. If $\tau < 1$, then it is assumed that a cluster can survive even without keeping all of its objects.

Similarly to the frameworks developed by Palla et al (2007), Asur et al (2009), Greene et al (2010), Takaffoli et al (2011) and Bródka et al (2013), MECnet is an event-based framework that relies on a two-stage approach. The first stage consists in independently discovering communities at each snapshot of the network. The framework is not restricted to a specific community detection algorithm, as long as the chosen algorithm partitions the network into disjoint communities. In the second stage, for each pair of successive snapshots, MEC compares the communities extracted at distinct time points based on the proportion of mutual nodes shared by them. This proportion can be obtained by computing the following conditional probability:

$$weight(C_{t_i}^m, C_{t_{i+\Delta t}}^u) = P(X \in C_{t_{i+\Delta t}}^u | X \in C_{t_i}^m) = \frac{\sum P(x \in C_{t_i}^m \cap C_{t_{i+\Delta t}}^u)}{\sum P(x \in C_{t_i}^m)} \tag{1}$$

where X is the set of entities assigned to community $C_{t_i}^m$ ($m = 1, ..., k_{t_i}$, with k_{t_i} being the number of communities returned by a given community detection algorithm at time point t_i) and $P(X \in C_{t_{i+\Delta t}}^u | X \in C_{t_i}^m)$ represents the probability of X belonging to community C^u from $t_{i+\Delta t}$ ($u = 1, ..., k_{t_{i+\Delta t}}$ with $k_{t_{i+\Delta t}}$ denoting the number of communities extracted at $t_{i+\Delta t}$) knowing that X belongs to community C_m obtained at a previous time stamp t_i.

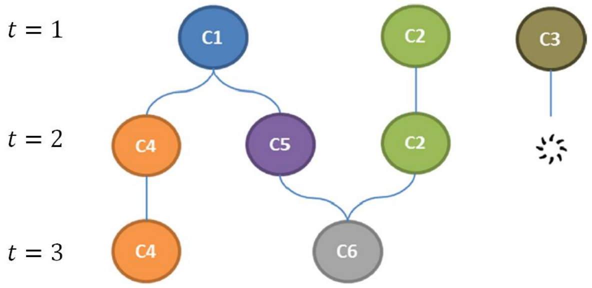
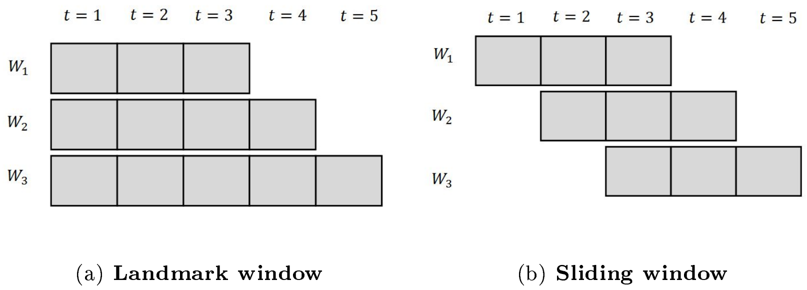


Fig. 1 Illustration of an evolution graph, for the time span $[t_1, t_3]$, depicting several types of events: split, merge, survival and death.

Based on this information, MECnet then models the communities and their transitions as nodes, respectively weighted edges, in an *evolution graph* (i.e. consecutive sets of weighted bipartite graphs). Figure 1 illustrates this model.

3.3 Window Models

Our methodology makes use of two well-known window models to analyse the dynamics of the network at the community-level: the landmark window and the overlapping sliding window. Although we have also carried out experiments with non-overlapping sliding windows of size 90 days, we considered that the obtained results were not interesting for our application due to the loss of information regarding the temporal continuity of customers' communities and consequent difficulty in finding persistent customer profiles. In other words, there were too few customers buying in consecutive timeframes, which reflected in a very irregular behaviour of the network. The specific reasons for discarding the non-overlapping sliding window from our application-driven methodology are



(a) **Landmark window** (b) **Sliding window**

Fig. 2 Illustration of (a) a landmark window, and (b) a sliding window with length three time points and a step width of one time point.

related to (a) the nature of the business of the company under analysis, which operates in the Business-to-Business (B2B) market, with the great majority of customers being other companies; and (b) the nature of the products sold by the company, which have long life-cycles and a low purchase frequency (see Section 4.1 for a more detailed description of the company's products). In this scenario, the non-overlapping sliding window model proved of limited usefulness, since it only confirmed what was already known, thus not being able to provide actionable insights into the evolution of customers. Note, however, that these conclusions only hold for this specific customer-related data. Therefore, if this methodology was applied to other types of customer-related data, the non-overlapping sliding window model could provide relevant information.

In this section, we explain how we formulate this problem for each type of window model considered in this study, on the context of MECnet.

Several taxonomies for categorizing the *transitions*, or *critical events*, that a cluster/community, may experience during its life-cycle (Falkowski et al, 2006; Palla et al, 2007; Asur et al, 2009; Greene et al, 2010; Bródka et al, 2013) were proposed in the literature. In this work we will use the one proposed in MEC. Thus, we consider that communities can undergo five different types of events: *birth*, *merge*, *split*, *survival* and *death*.

3.3.1 Landmark Window

The dynamic social network is modelled as an ordered sequence of T graphs $\{G_1, G_2, ..., G_T\}$, where $G_i = (V_i, E_i)$ represents a static cumulative snapshot of the network at a given discrete time point t_i $(i = 1, ..., T)$, depicting all the nodes and links observed up to the current time point (e.g. G_3 comprises all the nodes and edges observed in t_1, t_2 and t_3). The landmark window approach successively aggregates these static snapshots into a unique graph, as illustrated in Figure 2 - (a). MECnet is then applied to the accumulated network observed at each time point by:

1. First detecting the communities using a static community detection algorithm (e.g. Louvain method, Label Propagation algorithm) and then,
2. Modelling the evolution of these communities, for a sequence of T time windows, through an evolution graph.

The k_{t_i} communities found at time point t_i are denoted by $C_{t_i}^m$ $(m = 1, ..., k_{t_i})$. MECnet allows the characterization of the life-cycle of each dynamic community. The communities found at each time point are referred to as *instances* of a dynamic community or, alternatively, as *step communities* (Falkowski et al, 2006; Greene et al, 2010). A dynamic community is described as a sequence of step communities, whereas the life-cycle is defined as a sequence of events.

3.3.2 Sliding Window

The dynamic social network is also modelled as an ordered sequence of T graphs $\{G_1, G_2, ..., G_T\}$, where $G_i = (V_i, E_i)$ represents a static snapshot of the network at a given discrete time point t_i $(i = 1, ..., T)$. In contrast with the landmark window, where all static graphs are accumulated over time, in the sliding window approach only a pre-defined number of static graphs are considered for the temporal analysis. We propose the use of an *overlapping* sliding window in order to guarantee that there is always a mutual time point between consecutive instances of the window. This condition prevents highly disruptive transitions. The overlapping sliding window approach first partitions the time axis into time slots of fixed length w and then it employs a forgetting mechanism by considering only the static graphs falling within each one of these slots. Thus, whenever a graph G_i is observed and inserted in the window, another graph G_{i-w} $(i > w$ and $w < T)$ is forgotten. Such catastrophic forgetting allows us to focus only on current events, by considering in the analysis only the most recent nodes and links of the dynamic network. Community evolution is then studied by applying MECnet to the set of time windows (e.g. time windows G_{1-3}, G_{2-4}, G_{3-5} for a window length $w = 3$). In Figure 2-(b) we illustrate three timesteps $(W_k$, $k = 1, ..., 3)$ of an overlapping sliding window of length three time points $(w = 3)$ and step width of one time point.

4 Case Study

In this section, we proceed to validate the feasibility of our methodology using a real-world customer network, extracted from one of the largest Portuguese Groups operating in the electric field. The goal of the company was to use, for the first time, SNA techniques to perform an exploratory analysis of their customers purchasing behaviour, so as to identify differentiable customers' profiles (or customers' communities) and their evolution over a given year. Since it is known that some customers are frequent buyers, whereas others engage in more sporadic purchases, we considered relevant the analysis of the community dynamics using two distinct time window models, in an attempt to capture the behaviour of both types.

4.1 Network Data

The network data is imported from the company's CRM (*Customer Relationship Management*) application and corresponds to a time span of twelve months (year 2011). We model the network based on the similarity of the purchasing behaviour between customers of the company. Thus, there is a link between a pair of customers in t_i $(i = 1, ..., 12)$ if they both purchased the same product during t_i. This link is weighted by the number of co-purchased products. The resulting network is undirected and weighted. For the chosen time span, the company's product portfolio was comprised of nearly 200 different products, from which 152 products were actually bought by the set of customers under analysis. The company's main products are related to the electric field and, thus, some degree of technological evolution can be observed. These products are typically supported by other products sold by the company in the form of service/maintenance contracts. The company also sells additional products related to engineering and high-tech projects. Two distinctive characteristics of these products are their long lifespan and low purchase frequency. The products' nature has impact in the dynamics of customers' profiles and, consequently, in the number and type of detected events. The total number of nodes (active customers) and links in the whole network G_{1-12} is 1014 and 12259, respectively. The manipulation, visualization and analysis of the network is performed on Gephi (Bastian et al, 2009) by making use of its dynamic network analysis features.

4.2 Experimental Setting

We apply our methodology to the customer network by sequentially following the steps outlined in Section 3.

Instead of analysing a single snapshot of the entire available network G_{1-12}, we explore the dynamics of the network at a coarse-grained level (i.e. community-level), by making use of two window approaches. For the landmark model, we start with a window of 3 months and then we cumulatively grow the window by adding one month at each step. For the overlapping sliding model, we set the window length to 3 months $(w = 3)$ and its step width to one month. In both cases, the total number of timesteps is 10. The length and step width were set by the company's Business Intelligence analyst. Each timestep, denoted as W_k $(k = 1, ..., 10)$, is a time interval starting at t_i and ending at t_{i+w}. The timesteps for the landmark window are: $W_1 = [t_1, t_3]$, $W_2 = [t_1, t_4]$, $W_3 = [t_1, t_5]$, $W_4 = [t_1, t_6]$, $W_5 = [t_1, t_7]$, $W_6 = [t_1, t_8]$, $W_7 = [t_1, t_9]$, $W_8 = [t_1, t_{10}]$, $W_9 = [t_1, t_{11}]$ and $W_{10} = [t_1, t_{12}]$. The timesteps for the sliding window are: $W_1 = [t_1, t_3]$, $W_2 = [t_2, t_4]$, $W_3 = [t_3, t_5]$, $W_4 = [t_4, t_6]$, $W_5 = [t_5, t_7]$,

$W_6 = [t_6, t_8]$, $W_7 = [t_7, t_9]$, $W_8 = [t_8, t_{10}]$, $W_9 = [t_9, t_{11}]$ and $W_{10} = [t_{10}, t_{12}]$. We detect the communities at each W_k using the Louvain method (Blondel et al, 2008) since it produces disjoint partitions of good quality, in a very fast way. Due to space constraints, we do not present here the characterization of customers' profiles. Next, we apply MECnet to identify the critical events undergone by the found communities. We set the *survival threshold* of MECnet to $\tau = 0.5$ and the events are detected for successive snapshots of the network (i.e. timestep intervals $[W_k, W_{k+1}]$). The reason for choosing a low value for τ is related to the low purchasing frequency of the company's products and the consequent need to ensure a reasonable persistance of customers' communities. This choice of the survival threshold value was also supported by the results of the sensitivity analysis presented in Section 4.4.1. Finally, we evaluate both approaches based on a double perspective. First, we compute a quantitative measure, namely the *Survival Ratio* proposed by Spiliopoulou et al (2006), to measure network volatility and the frequency of community transitions in both scenarios. This ratio is given in Equation 2 and it basically computes the portion of communities found at timestep W_k $(k = 1, ..., 10)$ that survived in $W_{t_{k+\Delta t}}$.

$$SurvivalRatio(W_k) = \frac{\#SurvivedCommunities(W_{k+\Delta t})}{\#Communities(W_k)} \tag{2}$$

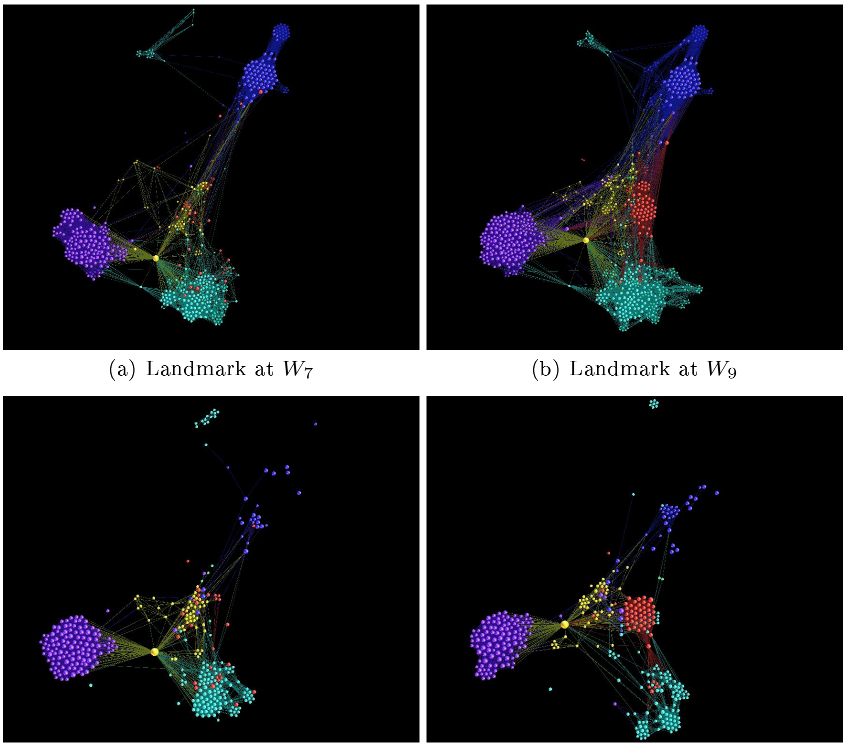
Then, we qualitatively assess the actionable insights derived from the analysis of each window model, from the business viewpoint.

4.3 Results

Following the experimental procedure described before, we obtain two dynamic networks. In Figure 3[1] we show two snapshots of each one of these dynamic networks, for timesteps W_7 and W_9. For the first scenario (Figure 3-(a)/(b)), the number of nodes varies between 336 (first timestep) and 1014 (last timestep), whereas the number of links ranges from 3509 to 12259 (see Figure 4). In contrast, in the second scenario (Figure 3-(c)/(d)), the number of nodes and links is more unstable over time, ranging from 227 to 336 customers and from 1519 to 3509 links, as can be ascertained from Figure 4. This is explained by the forgetting mechanism employed by the sliding window model.

We compute the following SNA metrics, which were considered to be of higher relevance within the scope of this case study: eigenvector centrality, betweenness, density and modularity. Since the former two are node-level metrics and, thus, need to be computed for each node, for simplicity we only present their average. The meaning of these metrics within the scope of our application is as follows:

[1] A video version of this figure is available online at http://www.youtube.com/watch?v=eEQpjspkj_8 (landmark window) and http://www.youtube.com/watch?v=X4_jI8Q4cWQ (sliding window).



(a) Landmark at W_7

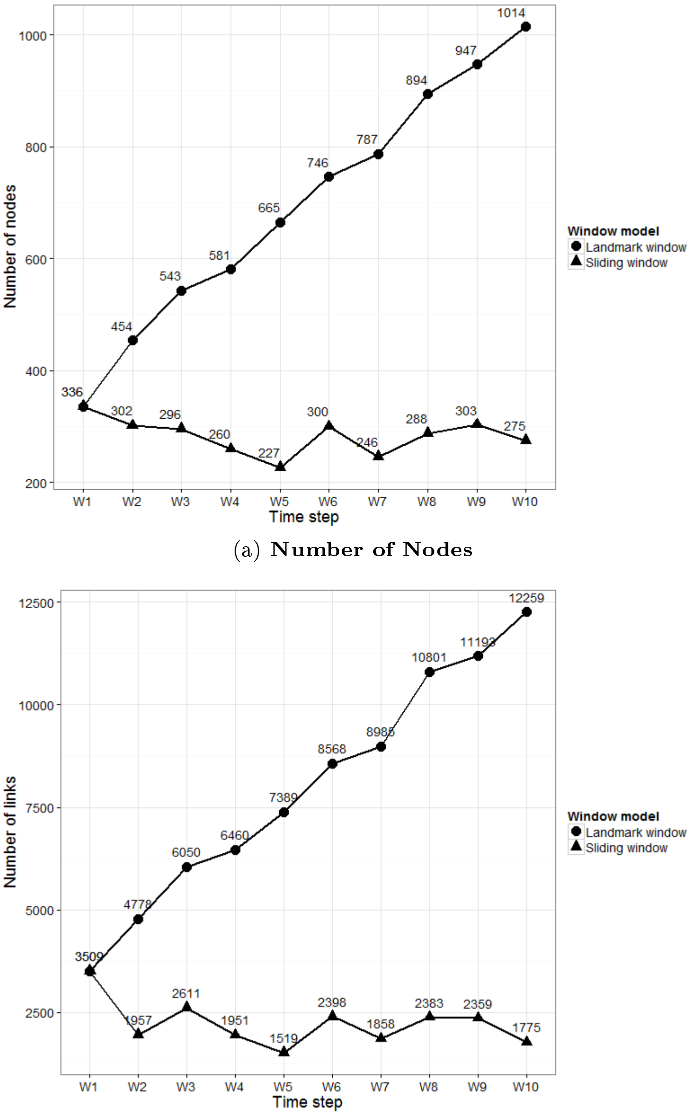
(b) Landmark at W_9

(c) Sliding at W_7

(d) Sliding at W_9

Fig. 3 Two snapshots of a dynamic customer network obtained using a landmark (top figures) and a sliding (bottom figures) window model, for two distinct timesteps.

- *Eigenvector Centrality:* when computed at the node-level, high values of this measure are associated with the so-called *storefront customers*. Due to their high visibility, *storefront customers* can be regarded as a proxy for the product's perceived quality, thus influencing the purchasing behaviour of other customers in the network.
- *Average Betweenness:* measure of the number of customers occupying gatekeeper positions in the network. Its temporal analysis can help identify trends in the customer network (e.g. diversification, change of technology) which, in turn, can help unveil overall trends in the market itself.
- *Density:* measure of the network connectedness level. When taking a dynamic view of the network, a high density reveals a certain "maturation" of the network, both in terms of customers and their purchasing behaviour. On the other hand, a sparser network indicates a less mature network, since customers exhibit different product preferences.
- *Modularity Q:* quality function that attempts to measure the merit of a given partition of the network into communities. Measures the difference between the number of within-community edges in a given set of communi-



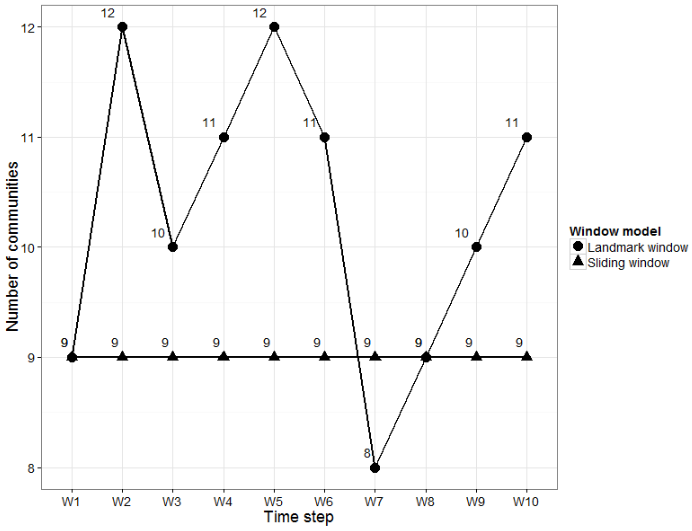
(a) **Number of Nodes**

(b) **Number of Links**

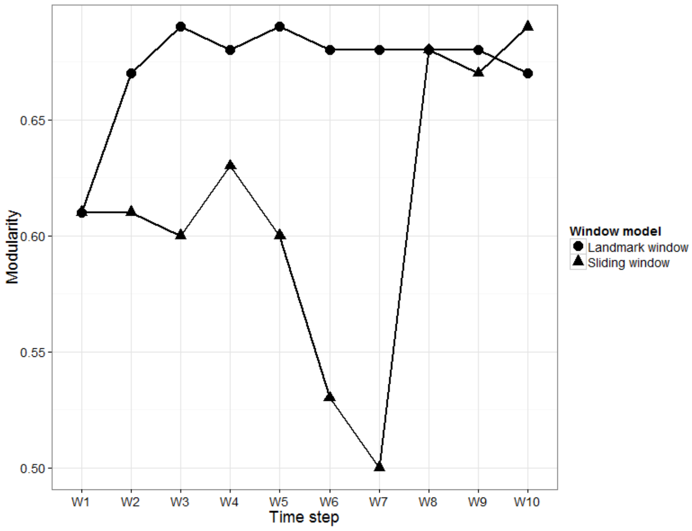
Fig. 4 Order and size of each network snapshot, for the landmark and the sliding window models.

ties and the expected number of within-community edges in a random network with the same degree distribution. Large modularity values ($Q > 0.3$) indicate the existence of meaningful community structures.

The obtained values are presented in Figures 5 and 6. Regarding the landmark window, the eigenvector centrality averaged over all customers in the

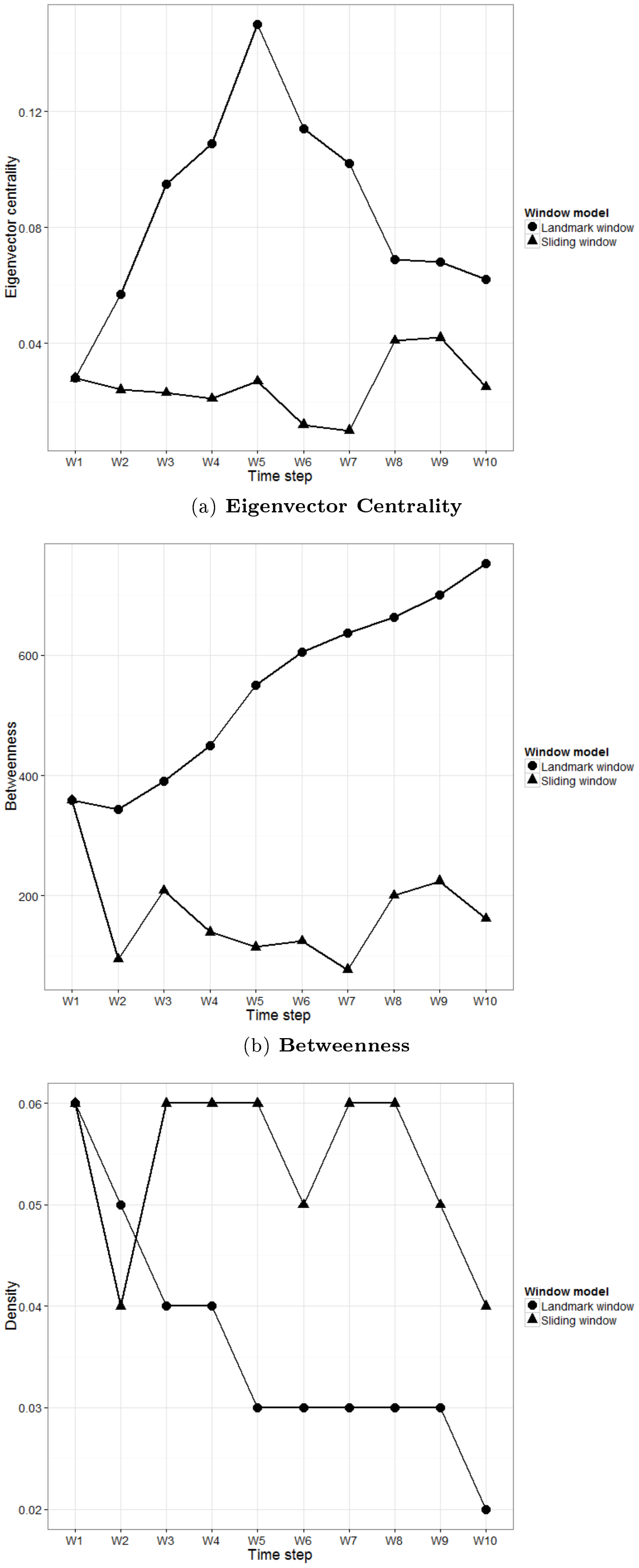


(a) **Number of Communities**



(b) **Modularity**

Fig. 5 Number of communities returned by the Louvain method at each timestep W_k $(k = 1, ..., 10)$ of the landmark and the sliding window models, and the corresponding modularity.



(a) **Eigenvector Centrality**

(b) **Betweenness**

(c) **Density**

Fig. 6 Node- and Network-level metrics obtained at each timestep W_k $(k = 1, ..., 10)$ for the landmark and the sliding window models. The node-level metrics (eigenvector centrality and betweenness) were averaged over all nodes.

network increases until $W_5 = [t_1, t_7]$ and then it starts exhibiting a downward trend, which persists until the end of the year. This behaviour makes sense because, as the network grows, there is a heightened probability that customers decrease their relative importance and, thus, their network centrality. Conversely, the average value of betweenness increases as the window covers more time intervals, due to the existence of a few temporally consistent hubs linking distinct communities. Concerning density, the low values indicate a sparse dynamic network, with different sets of customers exhibiting different buying behaviour. We also observe a decrease in density over time, which might be explained by a non-linear increase in both customers and links. These values reveal that the customer network under analysis might not yet be mature and, thus, has still space for growth.

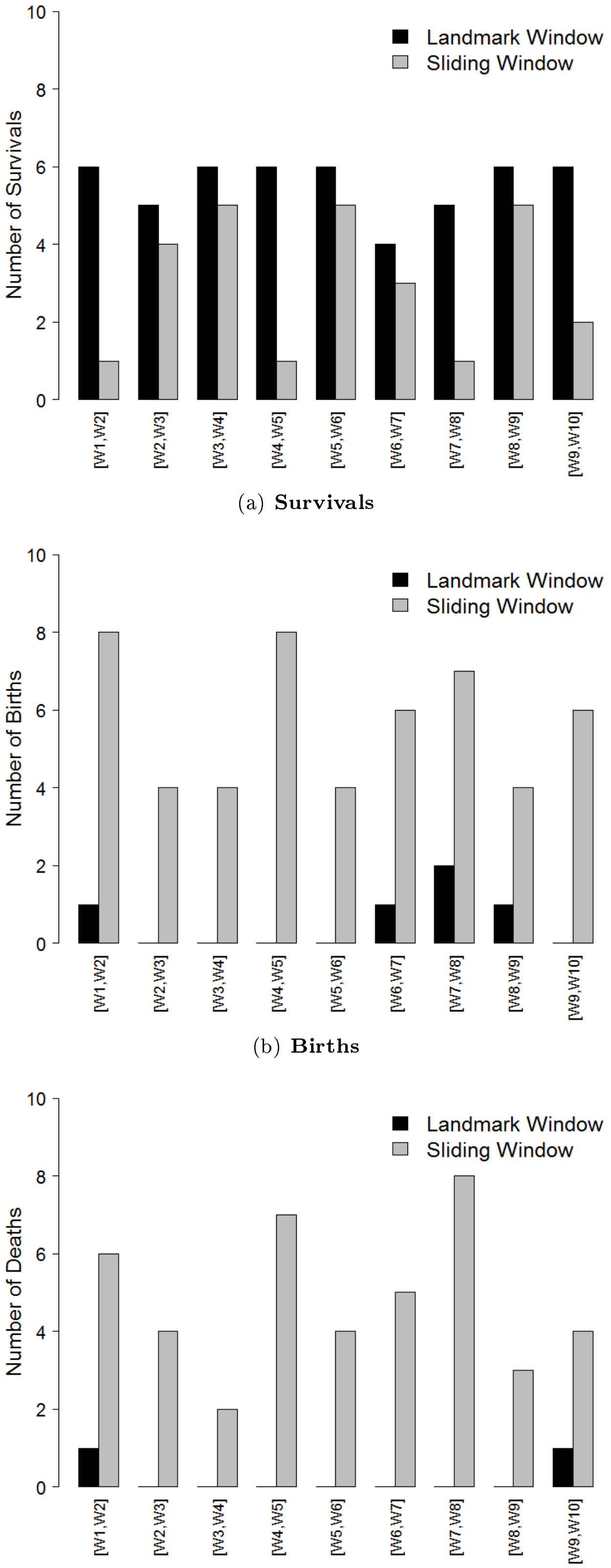
Concerning the sliding window, we observe an overall instability in the metrics values. This was expected due to the time-based forgetting mechanism incorporated in the model. The fluctuation of the eigenvector centrality suggests a lack of customers' purchasing activity. This is related to the company's business nature, which relies on long-term projects and high added value products which, in turn, are typically associated with low-frequency purchases. Once again, the density of the network is close to zero, due to the sparsity of links between customers.

4.3.1 Communities Dynamics

MECnet detects several types of events which mirror the dynamics of communities. In this paper, we focus only on *survivals*, *births* and *deaths* since they were found to be more representative of the dynamics occurring at the community-level. Given the high number of communities detected by the *Louvain* algorithm at each timestep (the initial number of communities ranged from 22 to 25, for different timesteps and for both window models), we restricted our analysis to the number of communities representing, at least, 75% of the total number of customers in each network snapshot. The motivation for imposing a threshold on the representativeness of communities was to ensure that communities formed by isolated nodes, or by a small number of nodes, were discarded from the analysis. From a business viewpoint, this choice warrants the identification of a more manageable number of customer profiles while helping the company focus on the most representative purchasing patterns of its customers' universe. This way, the company is able to focus its analysis on those customers who buy their core products, while helping the marketing team wisely manage the resources (time and money) needed to improve the customer relationship. The minimum threshold was set by the Business Intelligence analyst and its choice was guided by his business knowledge. In our experiments, the representativeness of communities ranges from 76% to 85%. The effect of using this threshold is a drop on the number of communities. In our experiments, this results in a minimum of 8 and a maximum of 12 communities, for the landmark window model, and in a consistent number of 9 communities for the sliding window model (see Figure 5-(a)). For both cases, the average modularity Q was 0.61 for the sliding window, and 0.67 for the landmark window, which suggests the existence of well-defined and meaningful communities of customers (see Figure 5-(b)).

Due to its greedy nature, the Louvain method is not completely deterministic. If we change the order of the nodes, this method might return different results. This variability in the outcome is partly explained by the sequential nature of the analysis of modularity gains, which makes the method highly dependent on the starting node. Bearing this in mind, we evaluated the stability of the Louvain method results, in terms of *modularity*, *number of communities* and *normalized mutual information* (NMI) (Danon et al, 2005), by first shuffling the nodes IDs and then running the algorithm on the resulting network. Note that the NMI is a well-known measure of similarity borrowed from information theory, which has proved to be reliable in comparing data partitions. The closer NMI is to 1, the more similar the two data partitions are. We resort to the NMI to compare a baseline community structure, i.e. the partition returned by the Louvain method without shuffling the nodes IDs, with the community structure returned by each run of the Louvain method. We perform our evaluation using the network corresponding to time window G_{1-3}, which is exactly the same for both window approaches. For 50 runs of the algorithm (i.e. 50 shuffles of nodes IDs), the average modularity was 0.67 (std. deviation = 0.002), the average number of communities, without considering the 75% threshold, was 23 (std. deviation = 1) and the average NMI was 0.93 (std. deviation = 0.03). These values suggest a high stability of the Louvain's outcomes for the analysed network, since we obtain similar community structures, as revealed by the large NMI value and by the identical number of communities, without compromising their quality, as indicated by the large modularity value.

The total number of survival, birth and death events detected by MECnet (with a survival threshold of $\tau = 0.5$) for both window approaches, at each timestep interval, is provided in Figure 7. From the analysis of these figures, we can deduce that the overlapping sliding window model is able to capture the unstable community structure of the customer network, by focusing only on the most recent past. This volatility is reflected on the high number of births and deaths and relatively low number of survivals and is captured by the values of the survival ratio (see Figure 7). These differences on the number of events show the potential of the sliding window in capturing temporary acute changes occurring in the network, reflecting more closely the changes in customers' preferences for products. At the community-level, these changes manifest themselves through sequences of death and birth events. Please note that nodes that are forgotten in the sliding window approach might re-appear at later temporal snapshots. In these cases, they are assumed as newborn nodes. Regarding the landmark window, the large number of survivals, as captured by the survival ratio, suggests little volatility between timesteps (Figure 8). The few birth/death detected events are related to more stable and long-lasting changes at the customer network.



(a) **Survivals**

(b) **Births**

(c) **Deaths**

Fig. 7 Events detected by MECnet (survival threshold $\tau = 0.5$) for the landmark and sliding window approaches.

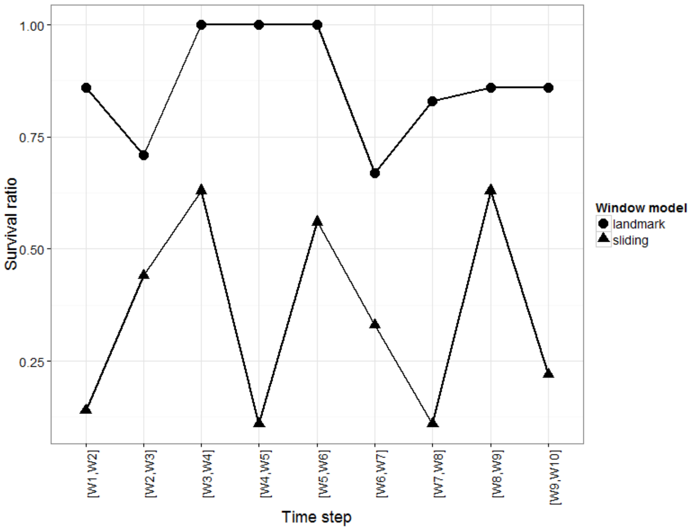


Fig. 8 Survival ratio for the landmark and sliding window approaches.

4.3.2 Qualitative Evaluation and Discussion

Cliques: our definition of *customer network* implies that customers who buy the same product in a given time frame are all linked to each other. Thus, it is likely that cliques are formed between customers who buy the same products. A closer look at the customer networks provides empirical evidence that supports this hypothesis. In fact, most members of a given community bought the same product or, alternatively, bought products of the same type. This finding suits our initial purpose of identifying customers with similar profiles in terms of products' preferences. On the other hand, when taking into account the temporal dimension of these networks, it can be hypothesized that these customers' cliques shift over time from the usage of a given product to the usage of another product. Due to the nature of this company's business, its products' characteristics and the availability of data for a single year, it is not possible to properly test this hypothesis. However, this shifting behaviour can be observed at the micro-level for several customers, signaling the possibility of observing this behaviour for whole communities in the case we had access to a longer timeframe of data.

Effect of the network model on the results: we model the implicit relationships among customers based on the similarity of their buying behaviour. In this kind of network model the type of products bought by these customers plays an important role on both the network structure and the nature of the detected events. The diversity of the company's product portfolio, in terms of products' categories, products' function and price ranges, is able to meet the needs of different types of customers, thus reflecting in a more idiosyncratic purchasing

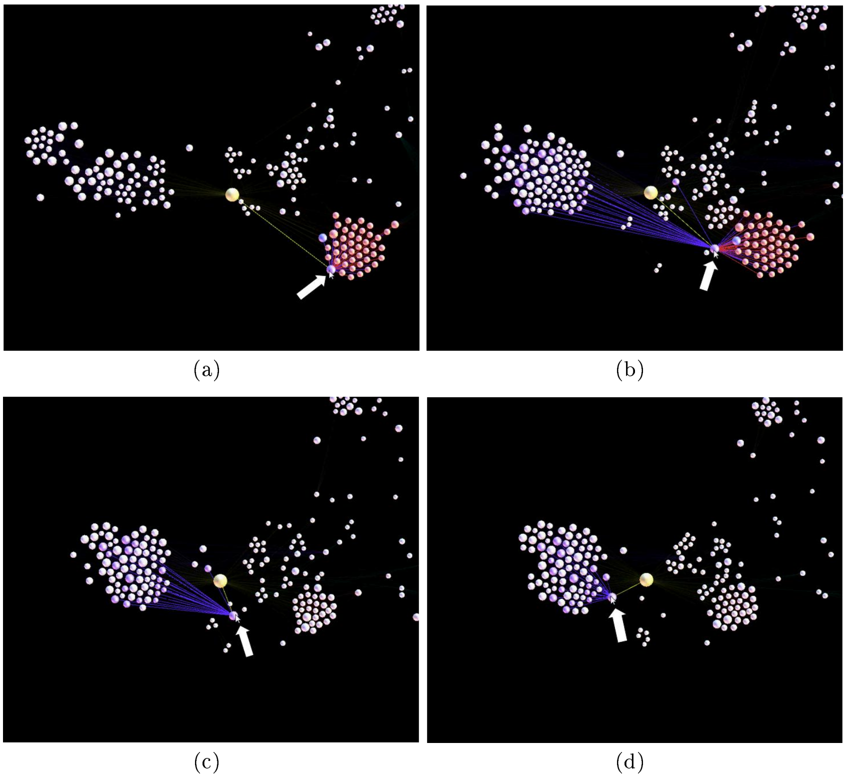


Fig. 9 Illustration of a customer's migration from one community to another. The customer's movement is marked by an arrow.

behaviour. This distinctive behaviour, shared by different sets of customers, partly explains the existence of well-defined communities revealed by the large modularity values. Furthermore, the long life-cycle of products (typically, more than 10 years) reflects in low replacement rates and low purchase frequencies. According to our data, 50% of customers made a single purchase, whereas 17% made only two purchases during 2011. This explains the high number of births and deaths of communities when using the overlapping sliding window model. Exceptions to this rule are customers that are large companies. These large corporations usually engage in repeated purchase, since they buy a product several times, as well as its associated products (e.g. service contracts). On the other hand, the merges, the splits and the individual customer's migration from one community to another, can be partly explained by the associated products the company sells (e.g. the purchase of a typical electric product is usually followed by the acquisition of a maintenance/service contract). Based on this qualitative analysis, it became clear that the choice of the network model influences the obtained results, so a careful study of the most appropriate model should precede the application of the proposed methodology.

Comparison between the landmark and the sliding window models: the analysis of the dynamics of the customer network using the sliding window approach, as a complement of the traditional landmark window, allows us to identify the migration of customers between communities. This kind of business events remains potentially undetected when relying only on a landmark window approach since, in such a case, the customer would appear as belonging to both communities. We exemplify this type of event (i.e. customer's migration) in Figure 9[2]. In this figure we take a closer look of the dynamic network (sliding window approach) by focusing on the movement of a specific customer. The identified customer is initially a member of a community characterized by customers who buy specific products. As time unfolds, this customer starts buying other products which are associated to a different community. As a consequence, the customer begins to gradually approach the second community and, as the old connection is forgotten, the customer moves to the only community it is now connected to. Due to the forgetting mechanism of the sliding window, the most up-to-date membership is highlighted thus 1bf enabling the detection of the customer's transition from one community to the other. Having the knowledge of the underlying data and of the company's business model, we are able to identify that the purchasing of the second product is a natural action after the purchasing of the first one. However, there are other companies which also supply the latter and, if this transition could be predicted, the company could have sold both products as a package in the first instance. This type of proactive commercial actions would reduce the company's commercial risk and increase its sales volume.

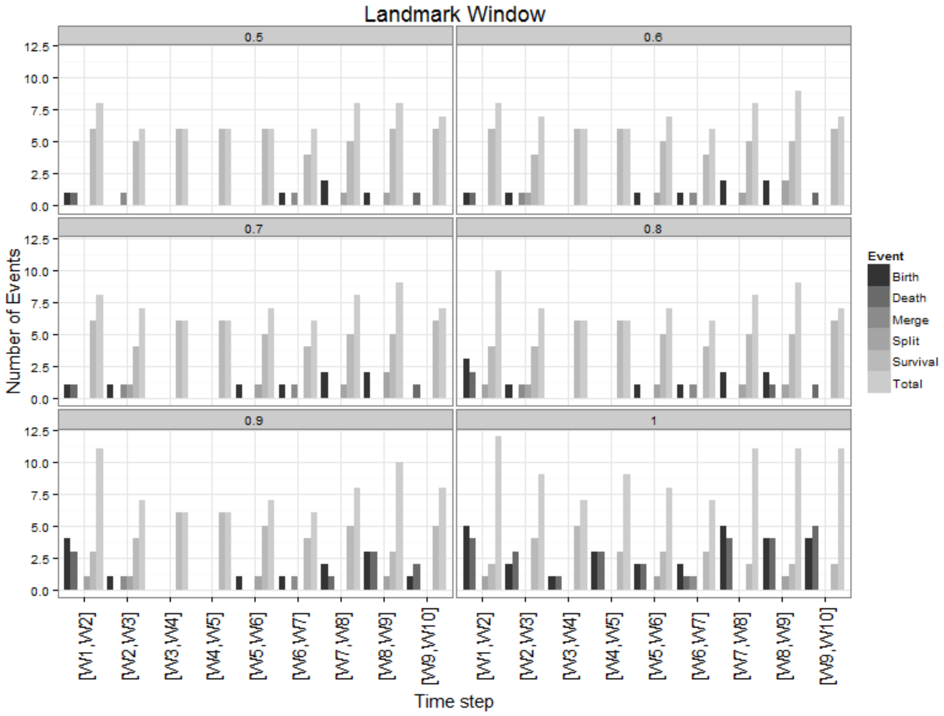
4.4 Additional Experiments

4.4.1 Sensitivity Analysis of MECnet's Survival Threshold

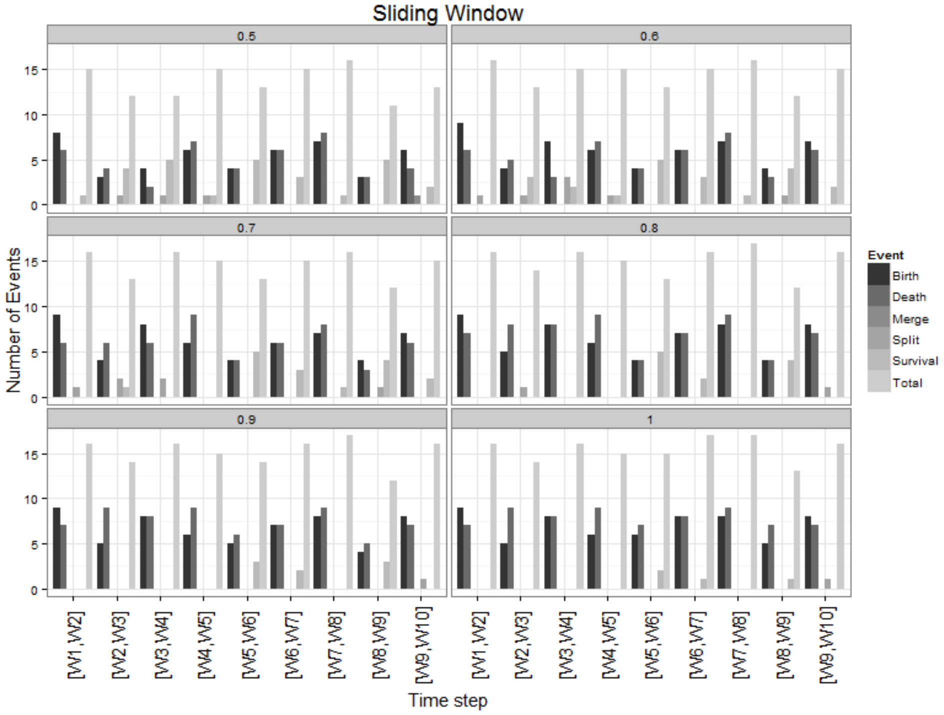
As mentioned in Section 3.2, the mapping process underlying MECnet relies on a user-defined threshold - the *survival threshold* τ-, which takes values from the interval $[0.5, 1]$. Here, we perform a sensitivity analysis of τ in order to assess the influence of choosing different values for the survival threshold in both the number and type of detected events. We run experiments with different values of τ, for both window models. The results are reported in Figure 10.

The analysis of results for both window models suggests that higher values of τ increases the number of births and deaths and decreases the number of survivals and merges. We also observe a rise in the total number of events. This finding agrees with the intuition behind the concept of survival threshold. When τ increases, we are imposing a stricter condition on MECnet for detecting communities' matches. If we set $\tau = 1$, MECnet will only consider a match between two step communities $C_{t_i}^m$ and $C_{t_i+\Delta t}^u$ if all nodes pertaining to community $C_{t_i}^m$ migrate to community $C_{t_i+\Delta t}^u$. This implies that, what was

[2] A video version of this figure is available online at `http://www.youtube.com/watch?v=SyR5jmU60Uk`.



(a) **Landmark window**



(b) **Sliding window**

Fig. 10 Influence of MECnet's survival threshold (τ) on the number of detected events (births, deaths, splits, merges and survivals).

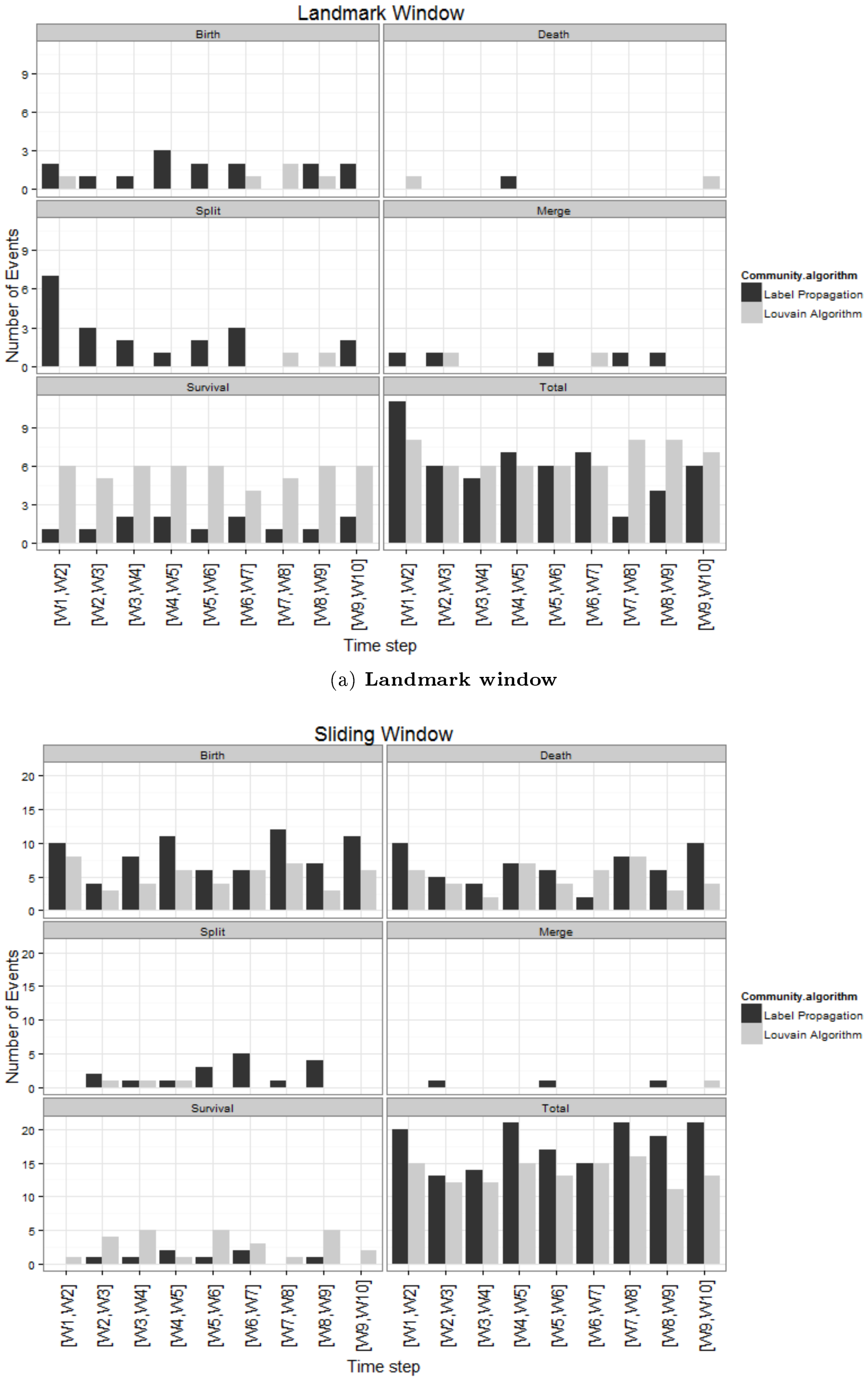
once categorized as a single *survival* event, will now be replaced by two events: a *death* and a *birth*. As a consequence, the total number of events increases. An analogous rationale applies to the merge events, since our definition requires that there are at least two community's matches for a merge to be detected. So, if the match condition (τ) becomes more demanding, the detection of merges becomes less likely.

Based on this analysis, we conclude that the choice of the threshold τ influences both the number and the type of events detected by MECnet. Lower values of τ are more flexible, allowing the detection of more survivals, which reflects in longer communities' life-cycles. In turn, setting more demanding values for τ shortens the communities' life-cycles due to the detection of disruptive events, such as births and deaths. The survivals detected for strict values of τ are, however, indicative of highly stable communities, whose detection might be useful for certain applications.

4.4.2 MECnet with Label Propagation Algorithm

In order to assess the influence of the choice of the community detection method on the MECnet results, we conducted experiments using the LP algorithm. We applied the LP algorithm to each network snapshot of each window model. In order to ensure a fair comparison with the Louvain method, we selected the communities containing at least 75% of nodes and we discarded the remaining ones. For the landmark windows, LP detected an average of 5 communities (std. deviation = 2), a maximum of 10 and a minimum of 3 communities, depending on the considered timestep. The average size of communities was 109 nodes, with the smallest group having 10 nodes, at timestep W_1, and the largest group containing 381 nodes, at timestep W_8. Regardless of the variability observed in both the number and size of communities, all community structures were meaningful, as indicated by the average modularity of 0.66 (std. deviation = 0.02). We obtain slightly different results for the sliding window. Using LP, we extracted an average of 10 communities (std. deviation = 2), a maximum of 13 and a minimum of 6 communities. From the whole set of communities, the smallest one was found at timestep W_5 and was comprised of only 6 nodes, whereas the largest community was detected at timestep W_6 and contained 106 nodes. The average modularity was 0.58 (std. deviation = 0.09). Although the modularity's average value decreased, when compared to the landmark window scenario, it is still high and indicative of the existence of meaningful community structures. Comparing these results with the ones obtained by the Louvain method (Figure 5), we conclude that, although we do not observe a stark contrast in the values of these indicators for each algorithm, it is apparent a higher variability in the number of communities over timesteps of the LP algorithm. Besides, according to the modularity criterion, the quality of the network partitions is on average higher for the Louvain method.

Regarding community dynamics, we followed the same procedure used for the Louvain method. The number of events returned by MECnet for each com-



(a) **Landmark window**

(b) **Sliding window**

Fig. 11 Influence of the community detection method (Label Propagation algorithm and Louvain method) on the number of events identified by MECnet.

munity detection algorithm, and for each window model, are shown in Figure 11. Starting with the landmark window model (Figure 11-(a)), we observe that when using the community structures discovered by the LP algorithm the number of splits and births is much higher than the ones detected using the Louvain method. This increase in splits and births is compensated by a reduction in the number of survivals. The apparent inverse relationship between the splits/births and survivals might be a consequence of the higher variability on the number of communities detected by the LP algorithm. In fact, if the number of groups is small in a given time interval and large in the next time interval, the communities either splitted into several groups or new communities were born. Concerning the sliding window (Figure 11-(b)), the total number of events detected by MECnet for the LP algorithm is consistently larger than the ones discovered for the Louvain method. By taking a closer look at the graphs, we can deduce that this large number of events is a reflection of the higher number of detected births, deaths and splits. Despite this high dynamicity on the evolution of communities, which can be explained by the forgetting mechanism employed by the sliding window model and is captured by both community detection algorithms, LP appears to extract more unstable community structures than the Louvain method. Once again, a possible justification for this difference resides on the variability of both the number and the composition of LP's communities.

Based on the performed comparison, we conclude that the choice of the community detection method influences the perceived dynamicity of communities and the number and type of events captured by MECnet.

4.4.3 Comparison of MECnet with the GED method

In this section, we compare the results returned by MECnet with the ones obtained using the GED method (Bródka et al, 2013). Similarly to MECnet, GED is an event-based framework for group evolution discovery in social networks that relies on a two-stage approach. During the first stage, communities (or groups) are discovered at each snapshot of a given temporal social network by resorting to an arbitrary community detection algorithm. Then, the relevance of nodes pertaining to each community is computed using any node importance measure (e.g. centrality degree, betweenness degree, page rank, social position). In the second stage, the *inclusion* measure is calculated for each pair of step communities found at consecutive timepoints. *Inclusion* takes into account not only the proportion of nodes shared by both step communities, but also the quality of nodes migrating from one step community to another. Based on the values of the inclusion measure and the values of three user-defined parameters (α, β and forming/dissolving threshold, also referred to as fd), the method looks for the presence of seven types of events: continuing (equivalent to MECnet's survival), splitting (equivalent to MECnet's split), merging (equivalent to MECnet's merge), forming (equivalent to MECnet's birth), dissolving (equivalent to MECnet's death), growing and shrinking. The two additional events considered in the GED method (growing and shrinking)

Table 1 Number and type of events detected by two dynamic community mining methods: MECnet and GED method, for all timestep intervals of the dynamic customer network. The proportion of *external events* (i.e. growing and shrinking events were excluded from the computation) with relation to the total number of external events, for both methods, is shown inside brackets.

Landmark Window								
Method/ Event Type	**Birth/ Forming**	**Death/ Dissolving**	**Split/ Splitting**	**Merge/ Merging**	**Survival/ Continuing**	**Growing**	**Shrinking**	**Total**
MECnet	72 (17.6%)	46 (11.2%)	24 (5.9%)	11 (2.7%)	256 (62.6%)	-	-	409
GED method	2088 (36%)	2376 (40.9%)	1344 (23.1%)	0	0	36	90	5934

Sliding Window								
Method/ Event Type	**Birth/ Forming**	**Death/ Dissolving**	**Split/ Splitting**	**Merge/ Merging**	**Survival/ Continuing**	**Growing**	**Shrinking**	**Total**
MECnet	338 (42.6%)	347 (43.8%)	20 (2.5%)	1 (0.1%)	87 (11%)	-	-	793
GED method	2736 (39.5%)	2880 (41.6%)	1241 (17.9%)	72 (1%)	0	93	78	7100

are internal events, in the sense that they capture changes concerning the contents of each community. In MECnet we focus only on external events, which relate to changes occurring in the whole community structure. Nevertheless, MECnet can be easily extended to accommodate these kind of internal events.

In order to ensure a fair comparison between MECnet and GED, we considered the same set of communities returned by the Louvain method, which were used as input for MECnet. Regarding the node importance measure required by GED, we selected the page rank (Brin and Page, 1998) instead of other centrality measures, such as degree or betweenness, due to its ability to take into account both the quantity and quality of the nodes' connections within the network. Page rank was computed for the whole set of nodes that were active at each timestep. We run the original implementation of GED method available in Piotr Bródka's web page [3]. For our experiments, we used the default values of GED's parameters: $\alpha = 50\%$, $\beta = 50\%$ and $fd = 10\%$. Since the parameter α is the GED's equivalent to MECnet's τ, the chosen α value (i.e. $\alpha = 50\%$) is consistent with the value chosen for τ in our case study experiments (i.e. $\tau = 0.5$). Such consistency guarantees a similar level of flexibility in both methods when looking for communities' matches.

We run the GED method for each window model. A comparison of the number and type of events detected by MECnet and the GED method is presented in Table 1. For the landmark windows, the GED method detected a total of 5934 events for all timestep intervals ($[W_1, W_2], ..., [W_9, W_{10}]$). MEcnet extracted a total of 409 events for the same data. The stark contrast between the total number of events of MECnet and the ones returned by GED are explained by GED's assumption that a single step community found at timepoint t_i might be involved in several events with other step communities at a later timepoint $t_{i+\Delta t}$. For instance, while MECnet categorizes a split of

[3] `http://www.ii.pwr.wroc.pl/~brodka/ged.php`

one community into three communities as a single event (i.e. one split), the GED method considers this occurrence as three splitting events. From the 5934 events discovered by GED, 2376 were dissolving, 2088 were forming, 1344 were splitting, 36 were growing and 90 were shrinking. Contrary to MECnet, no continuing events were detected by the GED method. This might be influenced by the fact that, while MECnet only takes into account the quantity of nodes shared by two step communities, the mapping process of the GED method also considers the position and importance of the nodes belonging to the step communities, as measured by page rank. A possible explanation to this is the fact that GED method considers as dissolving events communities that remain inactive over several timeframes, which is likely to happen in our data due to the company's business nature. When these communities become active again, the GED method categorizes their evolution as a forming event. Such cases are considered as *survivals* when using MECnet. Regarding the overlapping sliding windows, the application of the GED method for all timestep intervals returns a total of 7100 events. This raise in the number of total events, with respect to the landmark windows, somehow reflects the higher dynamicity of the networks obtained by the sliding window approach. This increase in the overall number of extracted events for the sliding window is consistent in both MECnet and GED. These 7100 events discovered by GED are broken down into 2880 dissolving, 2736 forming, 1241 splitting, 72 merging, 93 growing and 78 shrinking events. As expected, the number of forming/dissolving events is much larger for the sliding window than for the landmark window. This finding agrees with the results of MECnet for the sliding window, as suggested by the identical proportions of births/deaths (forming/dissolving in the terminology of GED) obtained by both methods (see Table 1).

Computation times of both methods for such small networks are not relevant, given that the experiments took only a few seconds.

5 Conclusions and Future Work

We introduce a methodology to analyse community structure dynamics in evolving customer networks, based on two window models: a landmark and a sliding window. This methodology was devised to tackle a real-world problem of one of the largest Portuguese companies on the field of electricity. The goal of the company was to explore social network research techniques in an attempt to tap the potential of their time-stamped customer base data, for marketing purposes. In this paper, we present the first results of our exploratory analysis on the company's customer network. This network uncovers the similarities of purchasing behaviour among the company's customers. The application of dynamic community mining using two different time window models allowed us to identify the evolutionary profile of groups of customers and grasp insights into the customer base. The results suggest that both window models provide complementary information regarding the dynamics of the underlying network. While the landmark window considers all the historical data, the sliding window employs a catastrophic forgetting of older data, focusing only on the most recent past. Given these distinct perspectives, the sliding window approach proves to be more suitable for ,e.g., detecting changes in customers' purchasing behaviour, whereas landmark windows are more appropriate to identify persistent customers' profiles. As future work, we intend to perform experiments using alternative time window approaches (e.g. accumulated time windows with fading links) and a few larger datasets, with longer timeframes, in order to assess the feasibility, performance and suitability of MECnet for dealing with large networks.

Acknowledgments

Authors acknowledge the support of the European Commission through the project MAESTRA (Grant Number ICT-750 2013-612944) and projects "NORTE-07-0124-FEDER-000056/59", financed by the North Portugal Regional Operational Programme (ON.2 - O Novo Norte), under the National Strategic Reference Framework (NSRF), through the Development Fund (ERDF), and by national funds, through the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT). Márcia Oliveira was also funded by FCT, under the PhD grant SFRH/BD/81339/2011. We also thank the anonymous reviewers for their valuable comments and suggestions on earlier versions of this manuscript. A special thanks to Rui Sarmento for helping us run the GED method script in Microsoft SQL Server Express 2012.

References

Asur S, Parthasarathy S, Ucar D (2009) An event-based framework for characterizing the evolutionary behavior of interaction graphs. ACM Transactions on Knowledge Discovery from Data 3(4):16:1–16:36

Bastian M, Heymann S, Jacomy M (2009) Gephi: An open source software for exploring and manipulating networks

Berger-Wolf T, Tantipathananandh C, Kempe D (2010) Dynamic Community Identification, Springer, New York, pp 307–336

Blondel V, Guillaume JL, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment 2008(10):P10,008

Bródka P, Saganowski S, Kazienko P (2013) Ged: the method for group evolution discovery in social networks. Social Network Analysis and Mining 3(1):1–14

Brin S, Page L (1998) The anatomy of a large-scale hypertextual web search engine. Computer Networks and ISDN Systems 30(1–7):107–117

Danon L, Diaz-Guilera A, Duch J, Arenas A (2005) Comparing community structure identification. Journal of Statistical Mechanics: Theory and Experiment 2005(09):P09,008

Datar M, Gionis A, Indyk P, Motwani R (2002) Maintaining stream statistics over sliding windows. SIAM Journal on Computing 31(6):1794–1813

Falkowski T, Bartelheimer J, Spiliopoulou M (2006) Mining and visualizing the evolution of subgroups in social networks. In: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, IEEE Computer Society, Washington, DC, USA, WI '06, pp 52–58

Fortunato S (2010) Community detection in graphs. Physics Report 486(3-5):75–174

Gehrke J, Korn F, Srivastava D (2001) On computing correlated aggregates over continual data streams. ACM SIGMOD Record 30(2):13–24

Greene D, Doyle D, Cunningham P (2010) Tracking the evolution of communities in dynamic social networks. In: Proceedings of the 2010 International Conference on Advances in Social Networks Analysis and Mining, IEEE Computer Society, ASONAM'2010, pp 176–183

Kawadia V, Sreenivasan S (2012) Online detection of temporal communities in evolving networks by estrangement confinement. ArXiv e-prints `1203.5126`

Lin YR, Chi Y, Zhu S, Sundaram H, Tseng BL (2009) Analyzing communities and their evolutions in dynamic social networks. ACM Transactions on Knowledge Discovery from Data 3:8:1–8:31

Newman MEJ (2003) The structure and function of complex networks. SIAM Review 45(23):167–228

Newman MEJ, Girvan M (2004) Finding and evaluating community structure in networks. Physical Review E 69(2):026,113

Oliveira M, Gama J (2012) A framework to monitor clusters' evolution applied to economy and finance problems. Intelligent Data Analysis 16:93–111

Palla G, Barabasi AL, Vicsek T (2007) Quantifying social group evolution. Nature 446:664–667

Raghavan UN, Albert R, Kumara S (2007) Near linear time algorithm to detect community structures in large-scale networks. Phys Rev E 76:036,106

Saganowski S, Bródka P, Kazienko P (2012) Influence of the dynamic social network timeframe type and size on the group evolution discovery. In: IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, IEEE Computer Society, Washington, DC, USA, pp 678–682

Spiliopoulou M, Ntoutsi I, Theodoridis Y, Schult R (2006) Monic: modeling and monitoring cluster transitions. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, USA, KDD '06, pp 706–711

Takaffoli M, Sangi F, Fagnan J, Zaiane OR (2011) Community evolution mining in dynamic social networks. Procedia - Social and Behavioral Sciences 22:49–58

Wasserman S, Faust K (1994) Social Network Analysis: Methods and Applications. Cambridge University Press, Cambridge