Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 35 (2014) 730 – 738

Procedia
Computer Science

18th International Conference on Knowledge-Based and Intelligent
Information & Engineering Systems - KES2014

# Generating groups of products using graph mining techniques

Sebastián A. Ríos[a,b,*], Ivan F. Videla–Cavieres[b]

[a]*Department of Industrial Engineering, Universidad de Chile, Av. República 701, Santiago, Chile*
[b]*Business Intelligence Research Center (CEINE), University of Chile, Av. Domeyko 2369, Santiago, Chile*

**Abstract**

Retail industry has evolved. Nowadays, companies around the world need a better and deeper understanding of their customers. In order to enhance store layout, generate customers groups, offers and personalized recommendations, among others. To accomplish these objectives, it is very important to know which products are related to each other.

Classical approaches for clustering products, such as K-means or SOFM, do not work when exist scattered and large amounts of data. Even association rules give results that are difficult to interpret. These facts motivate us to use a novel approach that generates communities of products. One of the main advantages of these communities is that are meaningful and easily interpretable by retail analysts. This approach allows the processing of billions of transaction records within a reasonable time, according to the needs of companies.

*Keywords:* Market Basket Analysis; Overlap Community Detection; Big Data; Graph Mining; Transactional Data;

## 1. Introduction

Large amounts of data are stored by companies for various reasons, such as increasing retention rates thanks to Customer Relationship Management (CRM)[14], [19]; for recommender systems[8]; for understanding of consumer behavior and generating customer profiles[2] from transactional and factual information. In fact,[4], explains that profiling customers usually generates a description of the behavior and often, will lead us to a good explanation for it.

The retail industry generates huge amounts of transactional information. Indeed, this work is focused generating valuable customer information to complement existing information.

We will generate groups of products based on the novel extension of market basket analysis proposed by Videla–Cavieres and Ríos[18] and then, we studied the stability of the communities found over different time windows. Videla–Cavieres and Ríos[18] introduced the concept of overlapped communities of products, allowing a product to belong to more than one community. This approach has proved to be useful to retail analysts, generating groups of products purchased together that are easily interpreted.

[*] Corresponding author. Tel.: +56 2 2978 0545
*E-mail address:* srios@dii.uchile.cl ; ividela@dcc.uchile.cl

doi:10.1016/j.procs.2014.08.155

## 2. Definitions and related work

One of the uses classical of the graph mining techniques is in Social Network Analysis (SNA). We can find some examples in[21,23,22] where the main purpose it is to understand the underlying structure and content inside it..

Videla–Cavieres and Ríos[18] work is focused on generating frequent item sets of products based on transactional data generated by a retail chain. Their main idea is to obtain sets of meaningful products that are interpretable by analysts. Now, we will generate groups of products following their approach and then we will study the stability of the communities over time.

In the following sections we will explain the datasets over which we apply our method; the approach proposed by Videla–Cavieres and Ríos[18] in detail and how we finally studied the stability of communities of products.

### 2.1. Data

We have transactional records from a retail chain in Chile. In terms of volume, we have around half billion records gathered within a period of twenty months, approximately 2, 200, 000 customers and over 42, 000 SKUs[1] globally.

### 2.2. Transactional data

We have a set of products and transactions. Products are defined formally as $P = \{p_1, p_2, \ldots, p_n\}$ where each $p_i$ represents an available specific SKU available. Indeed $|P| =$ *number of distinct SKUs*. A transaction $T$ is defined according to Agrawal and Srikant[3] as a set of items (products in this case) purchased in the same buying opportunity, such that $T \subseteq P$.

In our datasets, products are organized in three hierarchical level structures. Each level belongs to its predecessor based on an ad–hoc developed taxonomy by the retailer. Figure 1 shows a subset of one of our taxonomy and table 1 shows an example of product information with its hierarchy. In total we have 73 product families, 487 lines of products and 1, 447 sublines of products.

Table 1. Products characterization available

| SKU | Product name | Product Family | Product Line | Product Sub-line |
|---|---|---|---|---|
| 13231 | Milk ”The Happy Cow” | Dairy | Yogurt & Milk | Milk |
| 13201 | Yogurt ”Fancy Yogurt” | Dairy | Yogurt & Milk | Classic Yogurt |
| 13245 | Yogurt ”Smoothiest” | Dairy | Yogurt & Milk | Smoothie Yogurt |

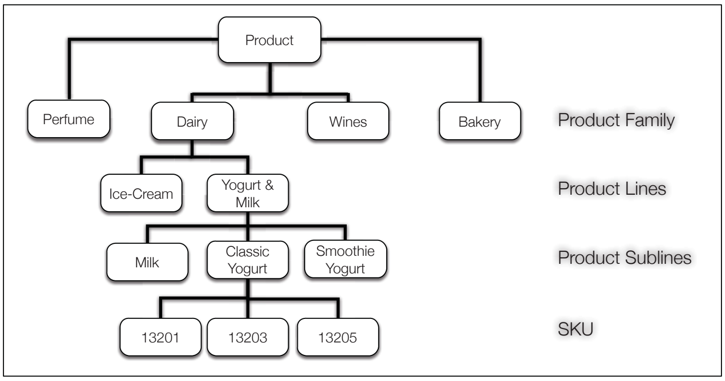

Fig. 1. Hierarchy of products.

[1] SKU : Stock Keeping Unit

Each transaction is identified by a unique number. An example of a transaction set is shown in table 2 where we see that 925 is a transaction composed of three products: P1, P2 and P4. These products were bought by customer 10021 on the date May 7th, 2009. Suppose SKU of P1 is $13,231$. On table 1, that would mean that the product is a Milk named "The Happy Cow" which belongs to Dairy Family, to Yogurt & Milk Line and to Liquid Milk Sub-line. On the other hand, transaction 926 has a *customer ID* equal to $-1$, which means that the retailer does not have that customer registered or that the customer does not want to give their identifier.

Table 2. Example of a transaction set.

| Transaction ID | Date | SKU | Customer ID | Quantity | Price | Total Price |
|---|---|---|---|---|---|---|
| 925 | 05-07-2009 | P1 | 10021 | 1 | 350 | 350 |
| 925 | 05-07-2009 | P2 | 10021 | 3 | 500 | 1500 |
| 925 | 05-07-2009 | P4 | 10021 | 2 | 500 | 1000 |
| 926 | 05-07-2009 | P3 | -1 | 4 | 600 | 2400 |
| 926 | 05-07-2009 | P4 | -1 | 9 | 500 | 4500 |
| 927 | 05-07-2009 | P1 | 1308 | 4 | 350 | 1400 |
| 927 | 05-07-2009 | P3 | 1308 | 7 | 600 | 4200 |

Table 2 presents the set of data available and how that information is stored. Another way to store that information is by the one expressed in table 3 which is a matrix whose rows are vectors of purchases. Each vector is composed of transactions and the set of products available. The first column store the transactional ID and in the following columns stored a number 1 or 0 which represents whether the product was purchased or not in that particular transaction.

Table 3. Example of a transaction set as a vector of purchase.

| Transaction ID | P1 | P2 | P3 | P4 |
|---|---|---|---|---|
| 925 | 1 | 1 | 0 | 1 |
| 926 | 1 | 0 | 1 | 1 |
| 927 | 1 | 0 | 1 | 0 |

An approach to the profiling problem where user profiles are learned from transactional histories using data mining techniques is presented by Adomavicius and Tuzhilin[1]. They also explain that profiles are constructed from two models. The first one is a *Data Model*, where we can have different types of data about customers, but this data can be classified in two basic types: -*demographic* and *transactional*, where demographic describes who the customer *is* and transactional describes what the customer *does*. Examples of these kinds of data are: name, gender, salary, etc. in the case of demographic. Table 2 is an example of transactional. The second model, *Profile Model*, is a collection of information that describes a customer. It can be classified in two groups – *factual* and *behavioral*–, factual contains specific facts based on information that can be derived from transactional data. Behavioral profiles model the behavior of a user, using conjunctive rules[3]. We built our profile based on a mix of Data and Profile models.

## 3. Product network and graph construction

We will expose the work developed by Videla–Cavieres and Ríos[18], because it is necessary to explain our characterization approach. They generated overlapped communities of products. This means, that a product can belong to more than one community at the same time. This is an important fact, because some products can not be confined to only one community. For example, a *carbonated beverage* can belong to a community of *soft drinks* and *alcoholic beverages*. The first case represents soft drinks for kids and the second, part of the community of products used to produce drinks for adults only. If we just allow this carbonated beverage to belong to one community we would be missing important information about that product. In fact, we will be losing half of the information available.

To study the stability of communities of products we have to generate the network of products following the Videla–Cavieres and Ríos[18] approach.

A *product network* is defined as a network (represented by a graph) where nodes represent products and edges represent relationships between a pair of them. In this case, an edge between two products represents that both products are present in the same ticket from the same buyer opportunity.

### 3.0.1. Network configuration

Literature presents two approaches for network generation, one is introduced by Sarwar et al. and Kim et al.[17,15,9,16], where a bipartite customer product network is built. This network links transactions with products, as depicted in figure 2(a). The second approach is introduced by Raeder and Chawla[12], based only on transactions where each product is linked to others because they appear in the same ticket from the same buyer opportunity. This kind of network is named *co–purchased product network* and is depicted in figure 2 (b). In this work the *co–purchased product network* will be used.

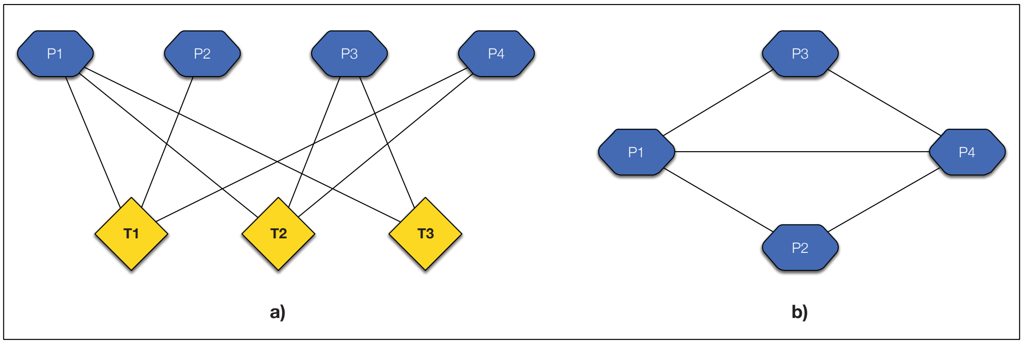

Fig. 2. Bipartite transaction products network (a) and co–purchased product network (b).

### 3.0.2. Network construction

The network will be built considering all the transactions $T$ that occurred during a certain period of time $k$. In this case, $k$ can be a particular day, week, month, quarter, semester and year. We will review each transaction $t \in T$, for which the transaction will be composed by a subset of products $P_{subset}$. For each pair of products $(p_i, p_j) \in P_{subset}$ an arc $a_{ij}$ will be created. After this process is done, the graph $G(N, E)$ is obtained called a *Temporally Transactional Weighted Product Network*. The process recently described generated over forty thousand Temporally Transactional Weighted Product Networks in our case. Then, each product network is filtered according the process described in section 3.0.3.

### 3.0.3. Network filtering

The idea is to generate and apply a filter over the product networks produced in 3.0.2. The filter is a threshold $\theta$ where each *edge* have to be at least, bigger or equal to this threshold in order to remain in the product network and not be removed. The idea behind this process is to remove edges that represent spurious and non–frequent relationships; relationships that are not lasting over time.

The process recently described requires iterating over the complete set of edges, looking for those that do not meet the threshold $\theta$. Previous approaches[12,9] to filter a network are not standard and although it can be replicated we believe that are subjective methods. For that reason, threshold setup is generated following the exact approach proposed by Videla–Cavieres and Ríos[18], which has proven to be an objective method, and it is depicted next.

*Threshold setup methodology*

Threshold $\theta$ is generated based on a process denominated by *top three heavy edges threshold* (tthet). This approach consists in ranking the edges $E = \{E_1, E_2, ..., E_m\}$ based on the weight of these in a descending order. Then *tthet* is equal to the average of the top three edges.

$$tthet = \frac{E_{max} + E_{2nd\ max} + E_{3rd\ max}}{3} \quad (1)$$

In equation 1, $E_{max}$ makes reference to the heaviest edge, $E_{2nd\ max}$ and $E_{3rd\ max}$ to the second and third heaviest edges respectively.

If we apply the obtained *tthet* to its corresponding network, only one or two elements would satisfy the minimum edge weight imposed by the threshold. This is explained because *tthet* keep the most relevant part of the *Temporally Transactional Weighted Product Network*, making useless the analysis.

This fact prompted them to generate a set of filters using the *tthet*, that allow gradually incorporating relevant edges and nodes into the analysis. These filters are a proportion of the *top three heavy edges threshold* (a proportion is a percentage of the threshold). The percentages are: $Percentage = \{5\%, 10\%, \ldots, 95\%, 100\%\}$; these percentages give 20 filters (or new thresholds), as a result of a dot product between *percentage* and *tthet* resulting in:

$$filters = percentage \times tthet \tag{2}$$

equal to:

$$filters = \{0.05 * tthet, 0.1 * tthet, \ldots, 0.95 * tthet, tthet\} \tag{3}$$

This process generates over fifty thousand new networks called, *Filtered Temporally Transactional Weighted Product Network*. If the filtered network is plotted, several zones appeared as we can see in figure 3. This figure was obtained after we applied a filter equal to the 10% of the *top three heavy edges threshold*.

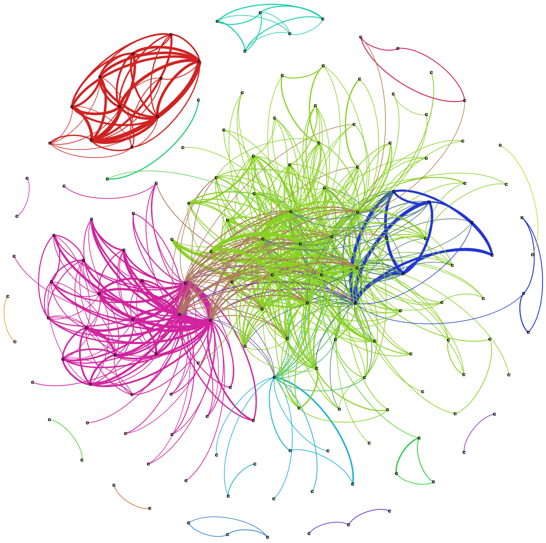

Fig. 3. Product–to–Product Network with a 10% filter.

Once the networks are filtered it is necessary to apply the overlapped community discovery algorithms to obtain and analyze the resulting communities. In the following section this process will be depicted.

### *3.1. Overlapping community discovery*

We applied the same algorithms as Videla–Cavieres and Ríos[18] to discover overlapped communities. These are COPRA[6] and SLPA (GANXiS nowadays)[20]. Both are based on the label propagation algorithm[13], in which nodes with the same label form a community. COPRA updates its belonging coefficients by averaging the coefficients from all its neighbors. Otherwise SLPA is a general speaker-listener algorithm based in the process of information propagation. SLPA spreads labels between nodes according to pairwise interaction rules. SLPA provides each node with a memory to store received information in difference to COPRA where a node forgets knowledge gained in the previous iterations.

The results obtained for a particular *Temporally Transactional Weighted Product Network* in October, 2010, after a 5% tthet was applied are depicted in table 4. The description was given by retail analysts after analyzing the obtained results.

Table 4. 10 largest communities discovered (order by number of products inside) which account for 85% of the products in the network.

| Community | # of products | Description |
|---|---|---|
| 1 | 172 | Groceries |
| 2 | 25 | Soft Drinks & Beers |
| 3 | 15 | Convenience Food |
| 4 | 7 | Juice Powder Brand A |
| 5 | 6 | Juice Powder Brand B |
| 6 | 5 | Liquid Juice Brand C |
| 7 | 4 | Yoghurt Brand D |
| 8 | 4 | Yoghurt Brand E |
| 9 | 3 | Liquid Juice Brand F |
| 10 | 3 | Cookies Brand G |

In section 3.0.2 we showed that we generated product networks over different time windows. Now, the question is: which is a *good* time window?. This question will be answered in the following section 4.

## 4. Communities stability

With the description and interpretation of the communities found, the recreation of the work made by Videla–Cavieres and Ríos[18] is concluded. We have been able to recreate their work, with excellent results. We had faced the same problems as they and this methodology allows us to generate communities of products that can be managed, analyzed and interpreted. Now, we explain how we carry out the stability analysis of communities.

One important aspect when communities are studied is their stability over time. We studied the evolution of communities, in terms of the underlying similarity between communities from different periods. This information will help to determine which is a representative period of time, containing communities that do not vary over time.

The process –to analyze the stability– consist in compare a particular *Temporally Transactional Weighted Product Network* from a specific time window ($TW_1$), for example a *day*, with another *Temporally Transactional Weighted Product Network* from a time window, for instance a *week*, ($TW_2$), and then iterate over different communities from $TW_1$ and $TW_2$ searching for the most similar community. This is defined as the community $c_i$ that contains the larger number of products both in $c_1$ and $c_2$. This process is repeated for all the time windows considered, for example day, week, month, quarter, semester and year.

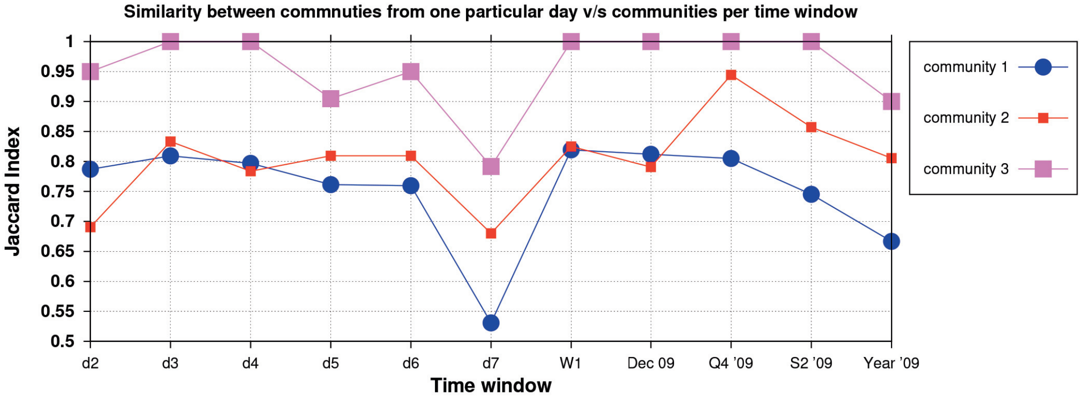

Fig. 4. Similarity between three communities from day $d1$ compared to different time window.

Mathematically, we have a set $TW = \{tw_1, tw_2, \ldots, tw_n\}$ of *Temporally Transactional Weighted Product Network* indexed by a particular time window. Each $tw_i$ contains a set $C_i$ of communities and each community $C_i^j$ contains a set $P_j$ of associated products.

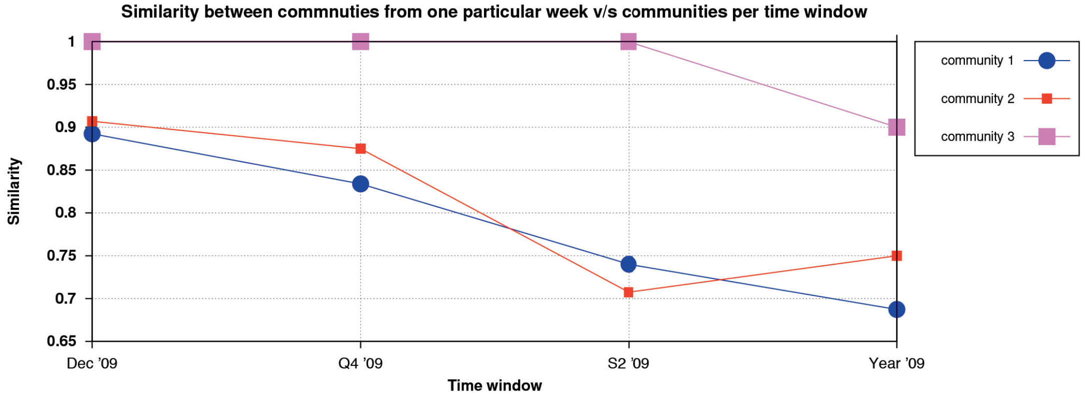

Fig. 5. Similarity between three communities from week $w1$ compared to different time window.

We calculated the Jaccard Index[7] between the common products from two communities $C_i^j$ and $C_r^q$. We then defined the similarity between those two communities $C_i^{j'}$ and $C_r^{q'}$ as the maximum Jaccard Index. It is important to note that community $C_r^{q'}$ is the most similar to community $C_i^{j'}$.

To obtain the most representative community over different time windows, it is necessary to calculate the similarity between communities. Figure 4 depicts the similarity of three communities from a particular day $d1$ in comparison with the rest of the periods.

We then repeated the process for the each time window, comparing the three most important communities with the three most relevant communities of the following periods. Figure 5 shows the similarity between a particular week (December 7th, 2009 to December 13th, 2009) and the rest of the time window.

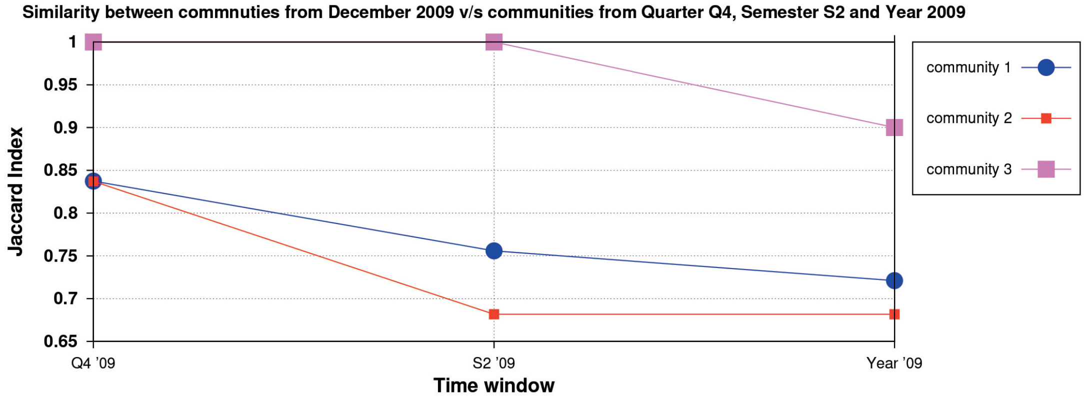

Fig. 6. Similarity between three communities from December 2009 compared to different time window.

In figure 6, the information is depicted from December 2009, in figure 7 the information of a quarter (October 2009 to December 2009). Finally, in figure 8 a semester (July 2009 to December 2009). Year 2009 (January 2009 to December 2009) is used as a pivot to make comparisons respectively.

We presented this information to analysts, who studied the results and concluded that the best time window found is a *month*, because the percentage is relatively similar over time. We analyzed the results obtained and what is represented by the products contained in each community, discovering that month is the best time window. For instance, communities from day time window, had mixed products that make impossible the analysis and the interpretation by the retail's analyst. Modularity[11] found, in communities from month time window, has values higher than 0.7 which is a value above those found in social networks[5]. It is also a very manageable time window and aligned with the time window used by the retail. Every model used by the retail is re–calibrated each *month*.

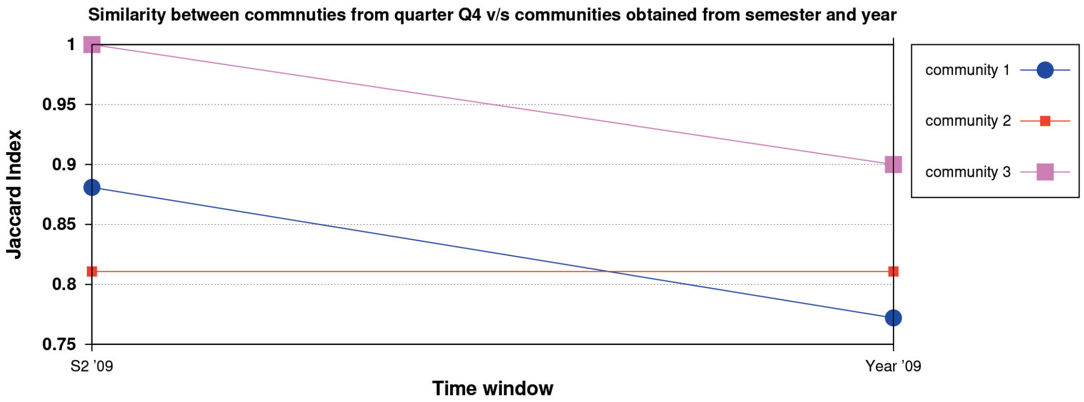

Fig. 7. Similarity between three communities from Quarter 4 (October to December 2009) compared to different time window.

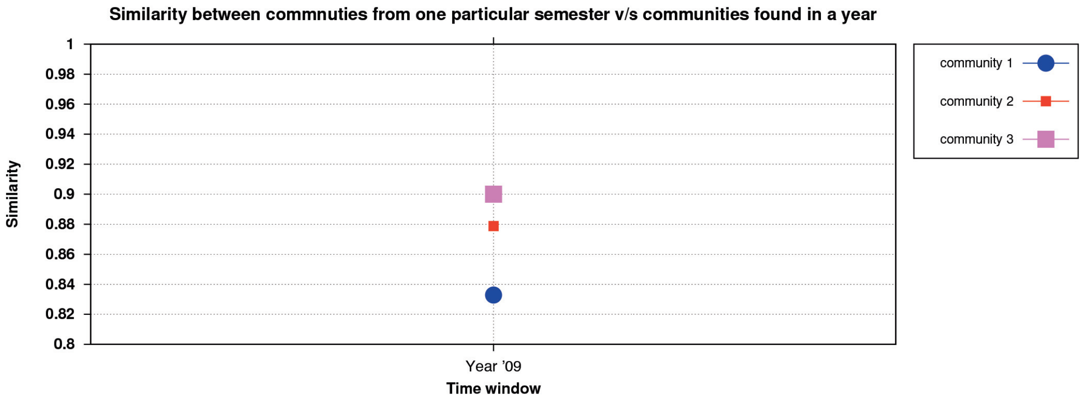

Fig. 8. Similarity between three communities from Semester 2 (July to December 2009) compared to different time window.

## 5. Conclusions and future work

We have demonstrated and validated the method proposed by Videla–Cavieres and Ríos [18], in order that their novel approach can be replicated, generating results of good quality, according to retail analysts. The method can be applied over large volumes of data, producing results aligned to the business needs in time and form. The methodology was successfully tried over real transactional data from two retail chains.

Our method it is based in an objective method that depends only on the previous purchases of a customer and communities are generated in a factual way. These methods do not need human intervention during the processing. Analysts take part only for studying the results obtained. We have shown an objective method for obtaining a representative time window, according to the needs of the retailer.

In future work, a personalized recommender system that uses this customer characterization and overlapped communities of products can be developed. A customer characterization based on the degree of membership to each community of products can be generated from previous purchases of a customer.

## Acknowledgments

The authors would like to acknowledge the support of the Chilean Millennium Institute of Complex Engineering Systems (ICM: P05-004-F FIN. ICM-FIC). We also would like to acknowledge to the Business Intelligence Research Center (CEINE) for their continuous support.

## References

1. Adomavicius, Gediminas, Alexander Tuzhilin. 1999. User profiling in personalization applications through rule discovery and validation. *KDD* 377–381.
2. Adomavicius, Gediminas, Alexander Tuzhilin. 2001. Using Data Mining Methods to Build Customer Profiles. *Research Feature* .
3. Agrawal, Rakesh, R Srikant. 1994. Fast algorithms for mining association rules. *Proc. 20th Int. Conf. Very Large Data Bases, VLDB* 1–32.
4. Berry, MJ, G Linoff. 1997. *Data Mining Techniques for Marketing Sales and Customer Support*.
5. Santo Fortunato. Community detection in graphs. *Physics Reports*, 2010.
6. Gregory, Steve. 2010. Finding overlapping communities in networks by label propagation. *New Journal of Physics* **12**(10) 103018. doi: 10.1088/1367-2630/12/10/103018.
7. Jaccard, Paul. 1901. *Distribution de la Flore Alpine: dans le Bassin des dranses et dans quelques régions voisines*. Rouge.
8. Jannach, D, M Zanker, A Felfernig, G Friedrich. 2010. *Recommender systems: an introduction*.
9. Kim, Hyea Kyeong, Jae Kyeong Kim, Qiu Yi Chen. 2012. A product network analysis for extending the market basket analysis. *Expert Systems with Applications* **39**(8) 7403–7410. doi:10.1016/j.eswa.2012.01.066.
10. Newman, M., M. Girvan. 2004. Finding and evaluating community structure in networks. *Physical Review E* **69**(2) 026113. doi: 10.1103/PhysRevE.69.026113.
11. V Nicosia and G Mangioni. Extending the definition of modularity to directed graphs with overlapping communities. *Journal of Statistical Mechanics: Theory and Experiment (JSTAT)*, March 2009, 2009.
12. Raeder, Troy, Nitesh V. Chawla. 2009. Modeling a Store's Product Space as a Social Network. *2009 International Conference on Advances in Social Network Analysis and Mining*. IEEE, 164–169. doi:10.1109/ASONAM.2009.53.
13. Raghavan, UN, R Albert, Soundar Kumara. 2007. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E* 1–12.
14. Reichheld, FF, RG Markey Jr, C Hopton. 2000. The loyalty effect - the relationship between loyalty and profits. *European Business Journal* .
15. Sarwar, B, George Karypis, J Konstan, J Riedl. 2000. Application of dimensionality reduction in recommender system-a case study .
16. Sarwar, Badrul, George Karypis, Joseph Konstan, John Riedl. 2000. Analysis of recommendation algorithms for e-commerce. *EC '00 Proceedings of the 2nd ACM conference on Electronic commerce* 158–167.
17. Sarwar, Badrul, George Karypis, Joseph Konstan, John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. *Proceedings of the 10th international conference on World Wide Web* .
18. Videla-Cavieres, Ivan F., Sebastián A. Ríos. 2014. Extending market basket analysis with graph mining techniques: A real case. *Expert Systems with Applications* **41**(4) 1928–1936. doi:10.1016/j.eswa.2013.08.088.
19. Winer, RS. 2001. Customer Relationship Management: A Framework, Research Directions, and the Future. *Haas School of Business* **April**.
20. Xie, Jierui, BK Szymanski, Xiaoming Liu. 2011. SLPA: Uncovering Overlapping Communities in Social Networks via A Speaker-listener Interaction Dynamic Process. *ICDMW 2011, 11th IEEE International Conference on Data Mining* .
21. Alvarez, Héctor and Ríos, Sebastián A and Aguilera, Felipe and Merlo, Eduardo and Guerrero, Luis. 2010. Enhancing social network analysis with a concept-based text mining approach to discover key members on a virtual community of practice. *Knowledge-Based and Intelligent Information and Engineering Systems* **2010**(4) 591–600.
22. Ríos, Sebastián A and Silva, Roberto A. 2013. A new dissimilarity measure for online social networks moderation. *Web Intelligence and Agent Systems* **11**(4) 351–364.
23. Ríos, Sebastián A and Muñoz, Ricardo. 2014. Content Patterns in Topic-Based Overlapping Communities. *The Scientific World Journal* **2014**(2014).