Big Data
Volume 9, Number 2, 2021

DOI: 10.1089/big.2020.0044

ORIGINAL ARTICLE

Measuring Customer Similarity and Identifying Cross-Selling Products by Community Detection

Lili Zhang,[1,*] Jennifer Priestley,[1] Joseph DeMaio,[2] Sherry Ni,[2] and Xiaoguang Tian[3]

Abstract

Product affinity segmentation discovers groups of customers with similar purchase preferences for cross-selling opportunities to increase sales and customer loyalty. However, this concept can be challenging to implement efficiently and effectively for actionable strategies. First, the nature of skewed and sparse product-level data in the clustering process results in less meaningful solutions. Second, customer segmentation becomes challenging on massive data sets due to the computational complexity of traditional clustering methods. Third, market basket analysis may suffer from association rules too general to be relevant for important segments. In this article, we propose to partition customers into groups with their product purchase similarity maximized by detecting communities in the customer–product bipartite graph using the Louvain algorithm. Through a case study using data from a large U.S. retailer, we demonstrate that the proposed method generates interpretable clustering results with distinct product purchase patterns. Comprehensive characteristics of customers and products in each cluster can be inferred with statistical significance since they are essentially driven by products purchased by customers. Compared with the conventional RFM (recency, frequency, monetary) model, the proposed approach leads to higher response rates in the recommendation of products to customers in the same cluster. Our analysis provides greater insights into customer purchase behaviors, improves product recommendation effectiveness, and addresses computational complexity in the context of skewed and sparse big data.

Keywords: cross-selling; product affinity segmentation; market basket analysis; community detection; bipartite graph

Introduction

Cross-selling is the practice of discovering existing customers' purchase preferences and engaging them with additional products or services,[1,2] which is an extremely valuable customer relationship management (CRM) strategy for increasing sales and customer loyalty. To save costs for the organization and to maintain the customers' interests, it is important to make the most relevant recommendations to target customers.

One commonly used analytical tool for cross-selling is collaborative filtering, which finds items based on their association patterns with purchased items across customers.[3] This is conventionally implemented with two approaches. One approach is from the perspective of products, called "market basket analysis", which identifies item sets that are frequently purchased together from all customers' transactions, providing insights on what items can be promoted together.[4,5] However, in a market basket analysis, interests of certain customer groups may be ignored, resulting in lost opportunities,[6] because the analysis is based on overall transaction data. For example, the association strength of Product A and Product B is not strong overall but can possibly be very strong for a strategically relevant customer group. The value of tailoring affinity products by customer groups is demonstrated by Mekonnen et al.[7]

Comparatively, the second approach is from the perspective of customers, called "customer segmentation". It partitions customers into groups that maximize

[1]*Analytics and Data Science Institute, Kennesaw State University, Kennesaw, Georgia, USA.*
[2]*Department of Statistics and Analytical Sciences, Kennesaw State University, Kennesaw, Georgia, USA.*
[3]*Department of Management and Marketing, Purdue University Fort Wayne, Fort Wayne, Indiana, USA.*

*Address correspondence to: *Lili Zhang, Analytics and Data Science Institute, Kennesaw State University, 3391 Town Point Drive NW, Room 2400, MD 9104, Kennesaw, GA 30144, USA, E-mail:* lzhang18@students.kennesaw.edu

customers' similarities within a group and their dissimilarities among groups based on customers' attributes used in the segmentation process. Products purchased by other customers in the same group can be used for cross-selling. It is ideal to incorporate the product-level data (i.e., each individual customer's product purchase data) in the modeling, so customers are divided into different groups based on preferences (i.e., product affinity segmentation). However, this process usually generates less meaningful solutions (e.g., one large segment, many small segments),[8] because the product-level data matrix is high dimensional, severely skewed, and contains a high frequency of 0s, due to the fact that many retailers have thousands of products and millions of customers, but most individual customers only buy a few product items. Moreover, the high computational complexity of the conventional clustering algorithms (e.g., K-means) makes it too time-expensive to be efficiently executed on millions of customers' profiles across an enterprise database. Alternatively, to avoid the product-level data issue, other relevant customer characteristics (e.g., demographics, lifetime value [LTV]) are used in the clustering model, and then, the products purchased by customers in the resulting clusters are profiled to gain additional insights.[8] Customers' similarities with respect to RFM (i.e., recency, frequency, monetary) can be discovered depending on attributes fitted into the model. However, these modeling results are not guaranteed to reflect customers' product affinity patterns.

To overcome limitations of conventional approaches and meet the needs of big data,[9] we propose to construct a customer–product bipartite graph based on customers' product-level data matrix and mine communities in the graph using the Louvain algorithm. The customers' product-level data matrix is intrinsically an adjacency matrix of a graph representation. In the graph, vertices are products and customers, and edges exist between customers and their purchased products. The Louvain algorithm is applied to detect communities, which contain both customers and products by maximizing within cluster product purchase similarity. The Louvain algorithm, as a graph clustering method, uses the modularity as the similarity measurement and forms clusters with the modularity maximized,[10] such that members in the same cluster are as similar as possible, while members in different clusters are as dissimilar as possible. Compared with other community detection algorithms, the Louvain algorithm is extremely computationally efficient, returning results in minutes or less for large graphs even comprising millions of customers and products.[10] It has been successfully applied in large graph contexts such as Twitter with 21 million vertices and 38 million edges,[11] a mobile phone network with 4 million vertices and 100 million edges,[12] and a citation network with 6 million vertices.[13]

Moreover, comprehensive characteristics of customers (e.g., RFM) and products (e.g., product class) in each cluster can be inferred based on the clustering results because all the other attributes are naturally driven by product purchases. These clusters can provide a reference for making customer-related decisions and planning better strategies on the customized product recommendation to improve profitability. For example, products can be recommended to customers in the same cluster. Compared with the RFM model, the proposed approach leads to higher response rates and is more cost-effective. The present work is a novel approach to using the Louvain algorithm in the context of customer segmentation and segment-specific product affinity recommendation.

This article is structured as follows. Relevant literature is first reviewed. Our proposed product affinity segmentation approach is then described in detail through a case study and compared with RFM model based on K-means. The findings are summarized and discussed.

Related Work

From the perspective of customer data, customer segmentation is the process of clustering customers into groups by maximizing similarities within clusters and dissimilarities between clusters. The similarity/dissimilarity between each pair of customer observations is measured by a distance function (e.g., Euclidean distance, Manhattan distance, Gower distance, Cosine) based on their attribute values.[14] Consider the K-means clustering procedure, which uses the Euclidean distance defined in Equation (1) as the similarity measurement, where p and q are vectors representing a customer's attributes and m is the number of attributes. The algorithm generates clusters by minimizing the sum of the squared distance of all observations to their closest cluster centers, defined in Equation (2), where K is the number of clusters and c_i is the centroid of the cluster C_i.[15] Its computational complexity is $O(n^2)$.[16] To improve K-means clustering solution quality, researchers have proposed new similarity functions, different from Equation (1), based on

concrete problem characteristics.[17,18] However, none of the alternative similarity functions reduces computational complexity.

$$\text{dist}(\boldsymbol{p},\ \boldsymbol{q}) = \sqrt{(p_1 - q_1)^2 + \ldots + (p_m - q_m)^2}, \quad (1)$$

$$\text{SSE} = \sum_{i=1}^{K} \sum_{x \in C_i} \text{dist}(\boldsymbol{c}_i,\ x)^2. \quad (2)$$

Because of its computational cost, K-means clustering becomes impractical on big data (e.g., customers' product-level data).[19] Table 1 shows a simple illustrative example with six customers and three products. Each customer's purchase frequency on each product is provided. For example, the customer c1 purchased the product p1 one time. In practice, this matrix is high dimensional, considering there are usually millions of customers and thousands of products in an organization's enterprise database. Moreover, this matrix is skewed and sparse because most customers only purchase a few products, generating less meaningful results.[8]

To overcome these limitations, one common strategy is to reduce the number of attributes used in clustering (e.g., only considering RFM attributes).[20,21] Marcus showed the collinearity among RFM attributes and proposed to focus on two key variables (i.e., number of purchases, average purchase amount).[22] Some researchers use RFM model outputs in their next steps to improve performance. Jonker et al. used the Markov decision process to determine the optimal marketing policy.[23] Cheng and Chen adopted the rough set theory to further mine classification rules.[24] Besides RFM, customer value attributes were also often used. Hwang et al. proposed an LTV model to include three types of customer values (i.e., current value, potential value, and customer loyalty).[25] Namvar et al. pointed out that most customer segmentation models considered the customer data only from a specific dimension, such as RFM and LTV, and then proposed a two-phase clustering method based on RFM, demographic, and LTV.[26] To further improve the computational efficiency, the two-step clustering algorithm was proposed with a preclustering step to generate a large number of small primary clusters.[27,28]

Table 1. An example of customer–product data

	p1	*p2*	*p3*
c1	1	1	0
c2	0	2	0
c3	0	1	1
c4	1	0	0
c5	0	0	3
c6	1	0	0

From the perspective of product data, market basket analysis finds the itemsets frequently purchased together from transaction data. One commonly used algorithm is the Apriori algorithm.[29] To improve the performance in finding large itemsets, Brin et al. further proposed the dynamic itemset counting algorithm.[30] Brijs et al. integrated the association rules with important microeconomic parameters and demonstrated its effectiveness on product-specific profitability.[31] Most of these algorithms generate association rules based on concepts of support, confidence, and lift.[32] The support for the rule is defined in Equation (4), where X and Y are two different itemsets, $\sigma(X \cup Y)$ is the number of transactions containing both and defined in Equation (3), and N is the total number of transactions.[3] The confidence and lift for the rule are expressed as Equations (5) and (6), respectively.[15] Alternatively, extracted association rules can be used in the performance analysis of different customer segmentation models.[33]

$$\sigma(X) = |t_i | X \subseteq t_i,\ t_i \in T|, \quad (3)$$

$$\text{Support}(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{N}, \quad (4)$$

$$\text{Confident}(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}, \quad (5)$$

$$\text{Lift}(X \rightarrow Y) = \frac{\text{Confidence}(X \rightarrow Y)}{\text{Support}(Y)}. \quad (6)$$

Regardless of the algorithms and attributes used for customer segmentation and market basket analysis, each procedure requires efforts of tuning and diagnoses for dedicated results. In our proposed approach, the Louvain algorithm directly utilizes the purchase linking between customers and products and partition customers into groups with the most similar product purchase. This is achieved by detecting communities in a customer–product bipartite graph.

A bipartite graph is a special graph structure with two disjoint sets of vertices, denoted as U and V, where edges only exist between U and V and no edge exists within either U or V.[34] In a customer–product bipartite graph, customers form one set of vertices, products form the other set of vertices, and edges go between customers and products. There exists an

Table 2. An example of customer–product bipartite graph data

Customer ID	*Product ID*	*Frequency*
c1	p1	1
c1	p2	1
c2	p2	2
c3	p3	1
c3	p2	1
c4	p1	1
c5	p3	3
c6	p1	1

edge between a customer and a product if the customer purchases the product. The edge weight is the customers purchase frequency on the product. For example, the product-level data in Table 1 can be converted to the data in Table 2 with all zero elements eliminated, which is represented by the customer–product bipartite graph in Figure 1. By only including nonzero entries, the graph structure mitigates the data sparsity issue.[35]

The Louvain algorithm, segmenting customers in the customer–product bipartite graph, has shown its success on the analysis of many complex graphs in both computational time and solution quality.[36] The algorithm was initially developed for undirected and unweighted graphs, but extended to directed and weighted graphs. It maximizes the modularity for each community, where the modularity measures the density of links within communities compared with links between communities,[10] as defined in Equation (7),

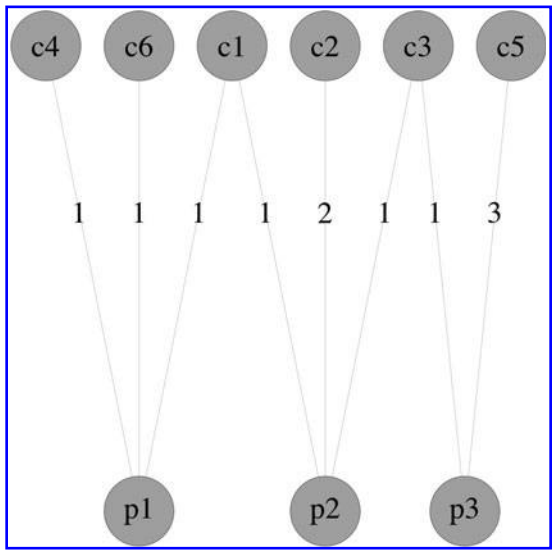

FIG. 1. An example of customer–product bipartite graph.

where A_{ij} is the weight of the edge between the vertex i and the vertex j, k_i is the sum of weights of edges connected to the vertex i, c_i is the community to which the vertex i is assigned, $\delta(c_i,\ c_j)$ is 1 if $c_i = c_j$ and 0 otherwise, and m is the sum of edge weights in the graph. It was adapted to bipartite graphs with modularity defined in Equation (8).[37]

$$Q = \frac{1}{2m}\sum_{i,j}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\delta(c_i,\ c_j), \tag{7}$$

$$Q = \sum_{i,\ j}\left[A_{ij} - \frac{k_i k_j}{2m}\right]. \tag{8}$$

The Louvain algorithm is accomplished in two phases.

- Phase I: First, treat each vertex as a community that only contains itself. Second, for each vertex, remove it from its current community and place it in its neighbor communities sequentially; after moving it to a neighbor community C, compute the modularity gain, as defined in Equation (9), where $\sum_{in}$ is the sum of edge weights inside the community C, $\sum_{tot}$ is the sum of edge weights incident to vertices in the community C, k_i is the sum of edge weights incident to the vertex i, and $k_{i,\ in}$ is the sum of edge weights from the vertex i to vertices in the community C. If the gain is positive, the vertex is moved to the neighbor community C; otherwise, it stays in its current community. Repeat the second step until no positive gain is produced. In this step, it is important to set a random seed for replication purposes.
- Phase II: First, treat the communities generated by Phase I as vertices and the sum of weights between communities as edge weights. Second, use those new vertices and edge weights to construct a new graph. Third, reapply Phase I on this new graph.

$$\begin{aligned}\Delta Q = &\left[\frac{\sum_{in} + k_{i,\ in}}{2m} - \left(\frac{\sum_{tot} + k_i}{2m}\right)^2\right] \\ &- \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right]\end{aligned} \tag{9}$$

In summary, the Louvain algorithm generates groups to achieve the same goal as K-means to make members in the same group as similar as possible

FIG. 2. Modeling process.

and members in different groups as dissimilar as possible. They both maximize similarity measurements and equivalently minimize dissimilarity measurements, although the data structure representations are different; K-means minimizes the within-cluster sum of squared errors on the matrix data structure, whereas the Louvain algorithm maximizes the modularity on the graph data structure.

Institutional review board approval has been waived because this research is not human subjects research, specifically the data used in this study are not identifiable private information.

Modeling

This study aims to discover product affinity segmentation for a large U.S. retailer with their de-identified data of sales transactions, customers, and products from January 1, 2018, to August 5, 2018, including:

- 773,999 sale transactions with 21 attributes (customer ID, product ID, order date, quantity, unit price, etc.);
- 260,386 customers with 60 attributes (customer ID, age, income, city, state, zip code, LTVs, etc.);
- 2112 products with 64 attributes (product ID, retail price, size bucket, class, style, color, active flag, etc.).

Although the data are unique for this U.S. retailer, the proposed modeling process can be generally applied by organizations with the data from sales transactions, customers, and products.

There are five phases in the proposed product affinity segmentation process: (1) Graph Data Preparation, (2) Graph Construction, (3) Community Detection and Profiling, (4) Benefit–Cost Analysis, and (5) Association Rules, as shown in Figure 2. Each phase will be described in detail.

Graph data preparation

To construct the customer–product bipartite graph, the unique customer–product pairs and their corresponding frequency values are extracted and calculated from the sales transaction data. An example of the graph data can be found in Table 2.

Graph construction

In the customer–product bipartite graph, the vertices represent either customer ID or product ID. The edges connect customers and the products they purchased. The frequency values of the products purchased by customers are stored as edge weights. In the resulted complete graph, there are 260,386 customer vertices, 2112 product vertices, and 500,729 edges. An example of the customer–product bipartite graph can be found in Figure 1, given the data in Table 2.

Community detection and profiling

To perform community detection on the constructed customer–product bipartite graph, the Louvain algorithm

Table 3. Statistics of clusters by customer–product bipartite graph clustering

Cluster	*Customer count*	*Product count*	*Cost*	*Benefit*	*Ratio*
12	4631	13	60,203	5317	0.0883
17	6329	15	94,935	7133	0.0751
13	6976	19	132,544	8198	0.0619
9	2945	25	73,625	4103	0.0557
18	4978	27	134,406	7163	0.0533
11	4573	32	146,336	7751	0.0530
16	6718	23	154,514	7752	0.0502
14	8922	29	258,738	10,353	0.0400
23	11,737	33	387,321	14,150	0.0365
10	9469	47	445,043	15,018	0.0337
24	2864	41	117,424	3897	0.0332
22	3352	50	167,600	4613	0.0275
8	10,301	49	504,749	13,525	0.0268
19	9932	59	585,988	15,389	0.0263
21	4277	61	260,897	6286	0.0241
0	7106	62	440,572	10,429	0.0237
3	5311	60	318,660	6569	0.0206
15	13,381	79	1,057,099	18,292	0.0173
20	13,735	76	1,043,860	17,495	0.0168
1	10,151	101	1,025,251	14,801	0.0144
25	8223	116	953,868	12,476	0.0131
2	17,381	155	2,694,055	27,463	0.0102
7	12,505	158	1,975,790	19,544	0.0099
4	29,176	173	5,047,448	44,397	0.0088
5	18,493	180	3,328,740	28,436	0.0085
6	26,902	420	11,298,840	44,667	0.0040
Total	260,368	2103	32,708,506	375,217	0.0115

is applied using the Python package community.[37] The execution is finished in less than 4 minutes on a Mac with 1.1G processor and 8G memory, demonstrating its computational efficiency. The total 262,498 vertices are partitioned into 35 clusters. The number of customers and products in 26 clusters is listed in Table 3. The remaining clusters are not presented because they retain fewer than five customers—small clusters are not typically kept in practice unless there is a compelling reason.

To closely examine how the resulting clusters are different from each other, Clusters 15 and 20 are used as the examples because they have very similar number of customers and products. Cluster 15 has 13,381 customers and 79 products, whereas Cluster 20 has 13,735 customers and 76 products. Most purchased products in Cluster 15 are leggings and bottoms, whereas most purchased products in Cluster 20 are sheers and socks, as shown in their subgraphs in Figures 3 and 4 with the edges colored by the product class.

Driven by different product purchases in each cluster, customers present distinct patterns regarding their RFM values, as shown in Figure 5. Compared with customers in Cluster 20, customers in Cluster 15 purchase with less recency and less frequency, but spend more money on each transaction. To check the statistical difference of RFM values in all clusters overall, statistical independence test is conducted and demonstrates their significance with p-values less than 0.01.

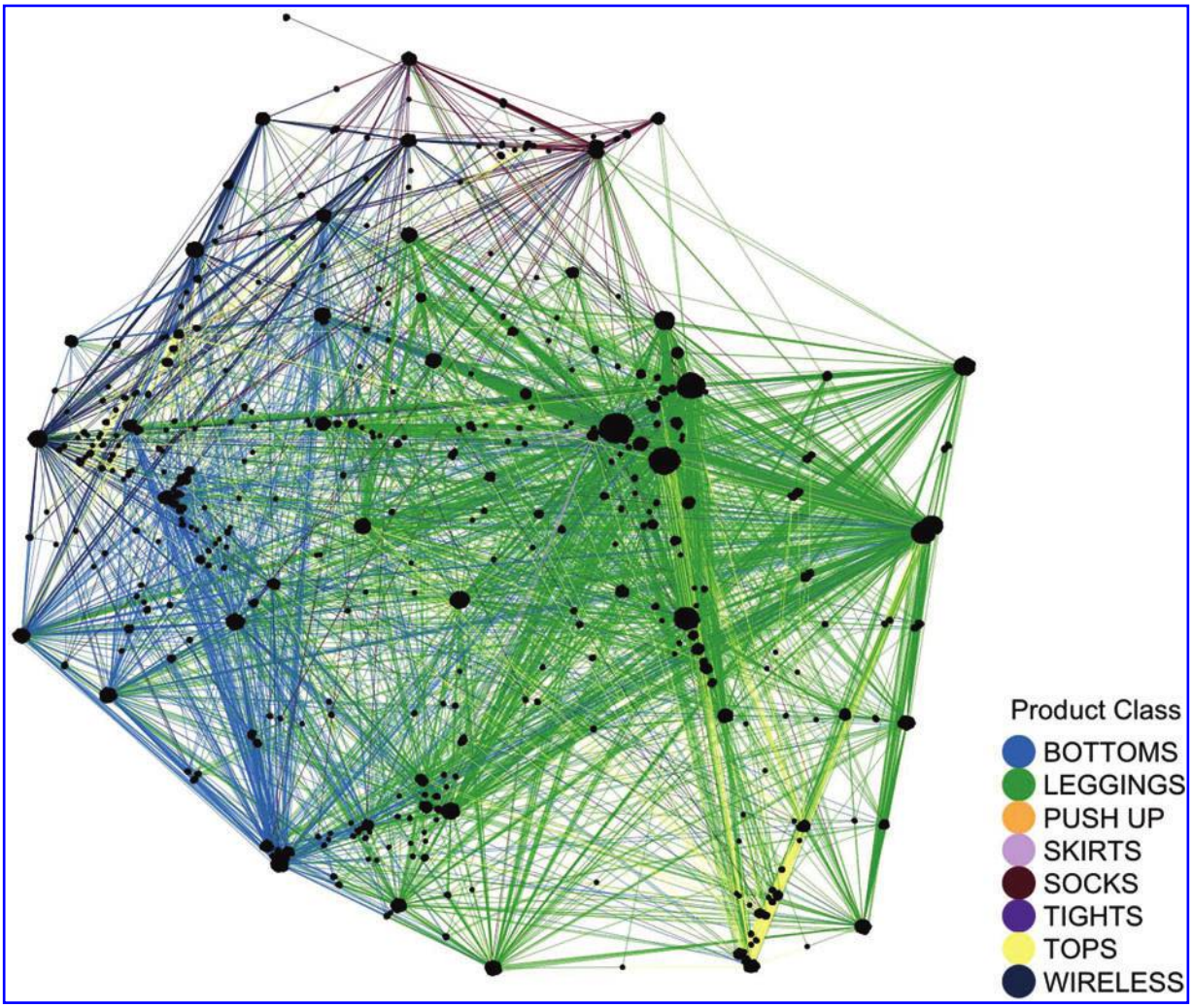

FIG. 3. Subgraph of Cluster 15. Color images are available online.

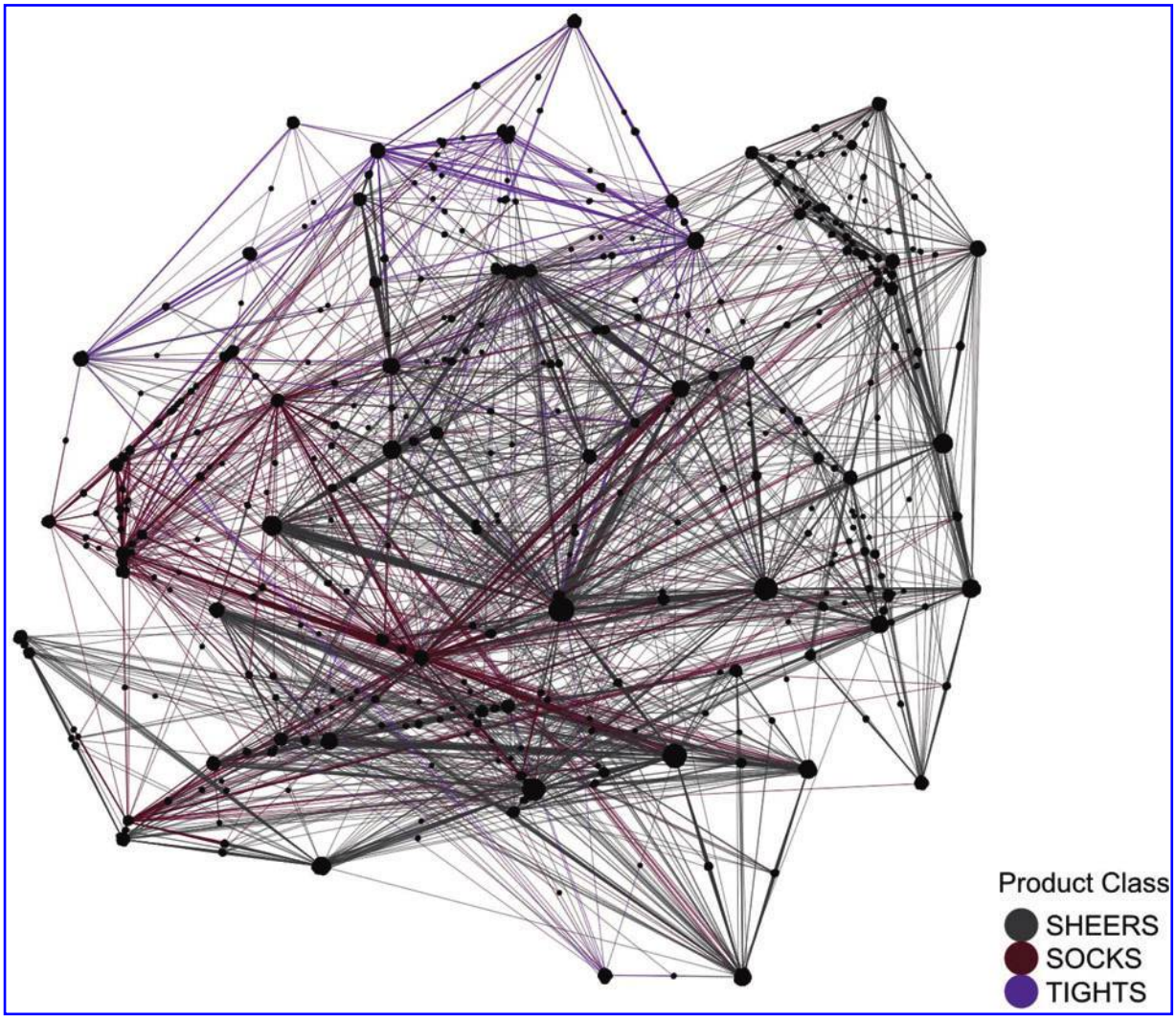

FIG. 4. Subgraph of Cluster 20. Color images are available online.

Benefit–cost analysis

Based on the customers and products in each cluster, one application is to recommend all products to all customers belonging to the same cluster. The benefit–cost analysis is performed for this application. Consider Cluster 12 with 4631 customers and 13 products. The recommendation cost would be the number of products multiplied by the number of customers, which is 60,203. The benefit for each product would be the number of responses or purchases, as shown in Table 4. For the product p100935, the benefit would be 388 because 388 customers purchased it. The benefit overall in this cluster would be the sum of benefits of all products, which is 5317. The benefit–cost ratio is the benefit divided by the cost, which can be interpreted in this context as the response rate. The statistics of each cluster can be found in Table 3. For Cluster 12, it is 0.0883, indicating that the customer response rate is 8.83%. The average customer response rate is 1.15%. This will be compared with the clusters generated by the RFM model based on K-means in the next section.

Association rules

Market basket analysis is conducted to further show different purchase behaviors and provide association rules that narrow down the recommendation cost in

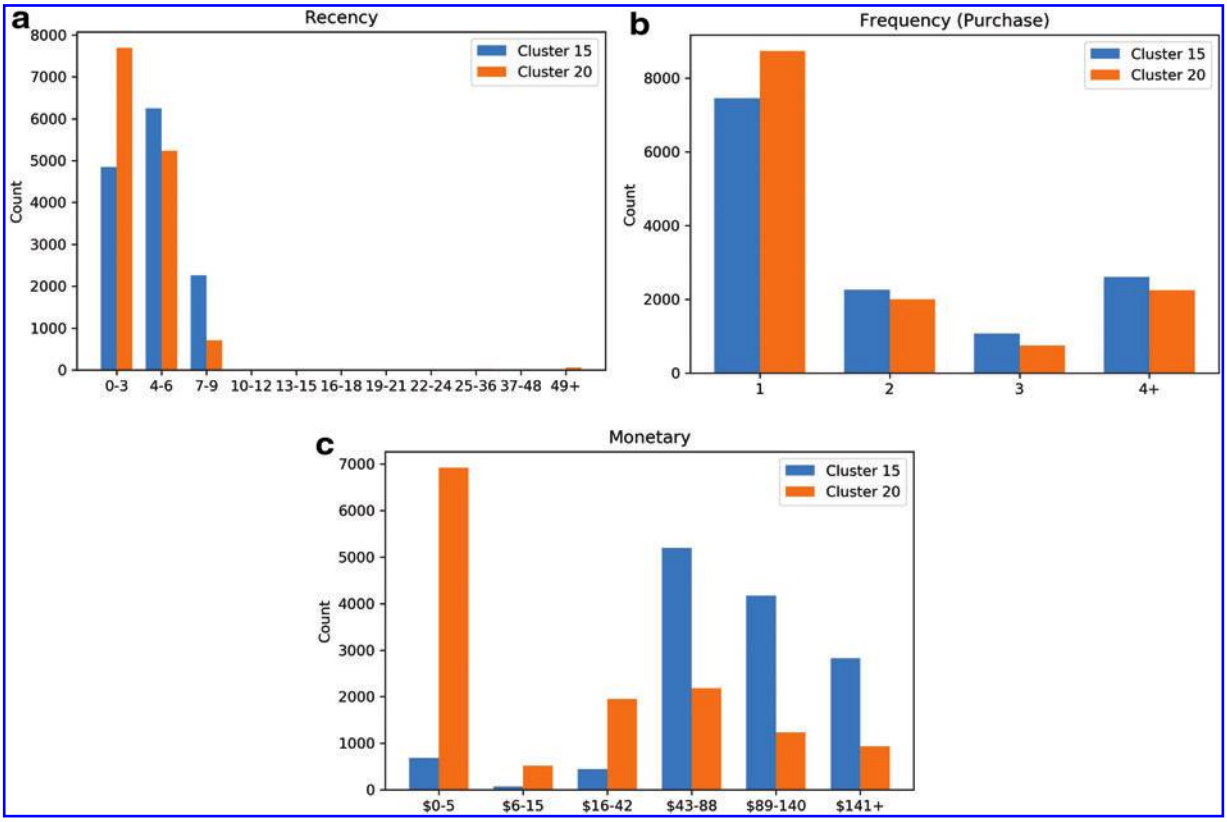

FIG. 5. Distributions of RFM. **(a)** Recency. **(b)** Frequency. **(c)** Monetary. Color images are available online.

each cluster. Take Cluster 15 and Cluster 20 as the example again. Top association rules generated from sales transactions of Cluster 15, Cluster 20, and overall can be found in Table 5, where Product IDs are presented in Rules. There is a strong association between the product p16312 and the product p16326 overall, but that is not true for Cluster 15 and Cluster 20. For Cluster 15, there is a higher possibility to achieve more sales by promoting the products p132391, p132400, and p132418 together, whereas for Cluster 20, a better opportunity is promoting the products p1236390, p29410, and p1236365 together.

Table 4. Benefit of Cluster 12 by product ID

Product ID	*Customer count (benefit)*
p100935	388
p108208	301
p108217	153
p112843	179
p112852	111
p112861	195
p122654	319
p122663	279
p122834	646
p122843	420
p125759	1117
p125768	564
p94955	645

Table 5. Top product association rules

Group	*Rules*	*Support*	*Confidence*	*Lift*
Overall	p16312 → p16326	0.0012	0.7709	185.7749
Overall	p119323 → p119127	0.0011	0.7151	150.1896
Overall	p16313 → p16327	0.0026	0.7391	84.9137
Overall	p16314 → p16328	0.0029	0.7360	86.9686
Cluster 15	{p132391, p132400} → p132418	0.0042	0.7368	25.3489
Cluster 15	{p1233139, p132078} → p1233137	0.0018	0.8571	37.6353
Cluster 15	{p132391, p132409, p132418} → p132400	0.0021	0.7778	22.3745
Cluster 15	{p119647, p81288} → p57633	0.0015	1.0000	18.4365
Cluster 20	p1258 → p1246	0.0138	0.7358	17.9739
Cluster 20	{p1236390, p29410} → p1236365	0.0011	1.0000	31.2111
Cluster 20	p127255 → p127241	0.0292	0.7130	11.9930
Cluster 20	{p1236401, p1236416} → p1236381	0.0018	1.0000	90.6129

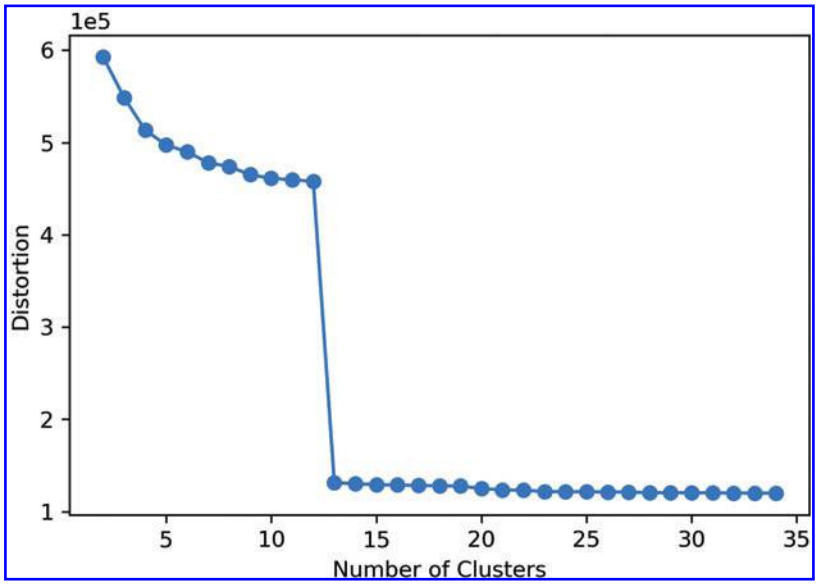

FIG. 6. Choice of number of clusters in RFM model. Color images are available online.

Comparison with RFM Model Based on K-Means

For comparison purposes, the RFM model based on K-means is built, with RFM values included in the clustering process.

- Recency (R): the number of days since the last purchase date;
- Frequency (F): the number of purchases;
- Monetary (M): the number of total price spent on all purchases.

The number of clusters is chosen to be 13, balancing the clustering quality and complexity. As shown in Figure 6, the distortion measurement, defined in Equation (2), does not meaningfully change after the number of clusters reaches 13.

The total 260,331 customers are analyzed and are placed in 13 clusters based on their RFM values, whereas 55 customers are not included in this analysis because they have zero aggregated total order quantity. The number of customers in 12 clusters is listed in Table 6. The remaining one cluster is not presented because it retains fewer than five customers.

To obtain the number of products (i.e., Product Count) associated with each cluster produced by K-means, the number of distinct products purchased by all customers in a cluster is calculated. In contrast, the Louvain algorithm assigns both products and customers to a certain cluster. Therefore, for products purchased by multiple customers who belong to different clusters assigned through K-means, these products are counted one time for each cluster, resulting in a higher Product Count value under the RFM modeling approach versus the Louvain algorithm. The cost, benefit, and ratio for each cluster are computed in the same manner as in the previous section. The highest customer response rate is 0.44% in Cluster 0, and the average customer response rate is 0.12%.

By comparing the distributions of the benefit–cost ratio (i.e., customer response rate) from Tables 3 and 6, as shown in Figure 7, we find that the clustering results of the Louvain algorithm generate higher values,

Table 6. Statistics of clusters by recency, frequency, monetary model based on K-means

Cluster	*Customer count*	*Product count*	*Cost*	*Benefit*	*Ratio*
0	2506	1837	4,603,522	20,098	0.0044
1	5081	1577	8,012,737	16,613	0.0021
8	8500	1808	15,368,000	29,378	0.0019
6	12,628	1952	24,649,856	45,588	0.0018
9	11,617	1868	21,700,556	38,883	0.0018
10	20,814	1548	32,220,072	32,748	0.001
7	27,478	1644	45,173,832	43,474	0.001
3	29,506	1769	52,196,114	47,528	0.0009
11	28,338	1778	50,384,964	45,231	0.0009
4	34,217	1810	61,932,770	54,992	0.0009
2	32,075	1865	59,819,875	50,630	0.0008
5	47,567	1819	86,524,373	72,083	0.0008
Total	260,327	21,275	462,586,671	497,246	0.0012

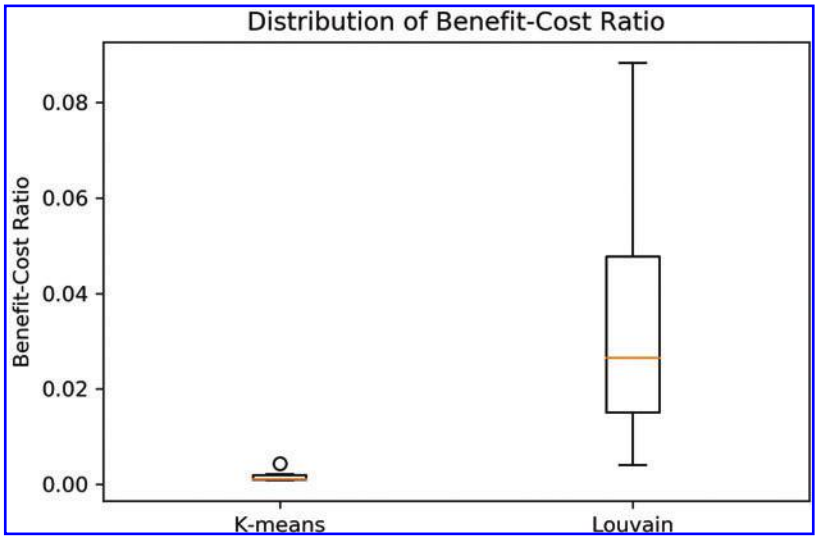

FIG. 7. Distribution of benefit–cost ratio. Color images are available online.

substantively improving response rates in the application of recommending products to customers in the same cluster. The highest customer response rate generated by Louvain (i.e., 8.83%) is 20 times of K-means (i.e., 0.44%). And, the average customer response rate increases 10 times from 0.12% to 1.15% by using Louvain. Moreover, customers' RFM values can be inferred from the clustering results of the Louvain algorithm because these values are driven by product purchase, whereas customers' product purchase cannot be distinctly reflected by their RFM values based on the K-means clustering results.

Conclusions

Product affinity segmentation can be effectively and efficiently performed by detecting communities in the customer–product bipartite graph using the Louvain algorithm in an interpretable manner. The Louvain algorithm partitions customers and products in the graph into clusters through maximizing the modularity measured by product purchase frequency similarity. The resulting clusters present distinct product purchase patterns regarding which customers purchase which products. Comprehensive attributes of customers (RFM, etc.) and products (product class, etc.) can also be inferred for each cluster since they are essentially driven by products purchased by customers. Relative to the RFM model, the proposed approach leads to higher response rates in the recommendation of products to customers in the same cluster, based on the benefit–cost analysis. Through the analysis of customers' characteristics and products associated with each cluster, decision makers can obtain greater insights into customer purchase behaviors, make better strategies for personalization strategic planning (e.g., customized product recommendation), and have higher probabilities to achieve more sales and improve profitability.

The study extends the Louvain algorithm to the business domain and includes benefit–cost analysis and association rules for improving model interpretability and application for business decision-making. The Louvain algorithm is explicitly interpreted in this study context satisfying the demand for algorithm transparency since it produces clusters in a way similar to the traditional K-means clustering method by optimizing the chosen similarity/dissimilarity measurement.

Moreover, the results of the proposed approach can effectively be applied in business practice for customer segmentation. In the decision-making process of customer segmentation decisions, the proposed approach can provide companies for designing and developing customer acquisition, retention, and loyalty programs. It may support companies to build competitive CRM strategies.

Limitations and Future Work

The proposed approach provides the cross-selling opportunities by recommending products purchased by

other customers in the same community to a customer. It differentiates a customer's preference on the products between communities, but it does not rank the products based on a specific customer's interest within a community. In practice, it is ideal to target a particular customer by promoting only some specific products or a subset of the products within a community. To achieve that, a classification model can be further used to score a customer's propensity of buying a specific product within a community. In the future, we would like to add the classification as the second phase to help diagnose and target particular customers based on their individual interests.

In the experiment of the present work, only the results between the proposed approach and K-means are compared. It is also ideal to show the performance of other community detection algorithms such as Clauset–Newman–Moore algorithm (CNM).[38] However, CNM does not converge and generates the results on our data after 24 hours, whereas the Louvain algorithm only takes around 3 minutes. The convergence issue of other community detection algorithms on large data sets has been discussed before.[10] Future work includes testing these algorithms on more powerful machines with larger data sets to obtain results and compare performance on metrics related to evaluating cross-selling opportunities (e.g., sensitivity and specificity of the classification model).

Author Disclosure Statement

No competing financial interests exist.

Funding Information

No funding was received for this article.

References

1. Li S, Sun B, Montgomery AL. Cross-selling the right product to the right customer at the right time. J Market Res. 2011;48:683–700.
2. Thuring F, Nielsen JP, Guillen M, et al. Selecting prospects for cross-selling financial products using multivariate credibility. Expert Syst Appl. 2012; 39:8809–8816.
3. Kamakura WA. Cross-selling: Offering the right product to the right customer at the right time. J Relationsh Market. 2008;6: 41–58.
4. Mostafa MM. Knowledge discovery of hidden consumer purchase behaviour: A market basket analysis. Int J Data Anal Techn Strateg. 2015;7:384–405.
5. Kaur M, Kang S. Market basket analysis: Identify the changing trends of market data using association rule mining. Proc Comput Sci. 2016;85: 78–85.
6. Brito PQ, Soares C, Almeida S, et al. Customer segmentation in a large database of an online customized fashion business. Robot Comput Integr Manuf. 2015;36:93–100.
7. Mekonnen A, Harris F, Laing A. Linking products to a cause or affinity group. Eur J Market, 2008;42:135–153.
8. Baer D, Chakraborty G. Product affinity segmentation using the doughnut clustering approach. In: Proceedings of the SAS Global Forum 2013 Conference, Cary, NC, 2013.
9. Provost F, Fawcett T. Data science and its relationship to big data and data-driven decision making. Big Data. 2013;1:51–59.
10. Blondel VD, Guillaume JL, Lambiotte R, et al. Fast unfolding of communities in large networks. J Statist Mech Theory Exp. 2008;10:P10008.
11. Pujol JM, Erramilli V, Rodriguez P. Divide and conquer: Partitioning online social networks. arXiv Preprint 2009;arXiv:0905.4918.
12. Roma G, Herrera P. Community structure in audio clip sharing. In: 2010 International Conference on Intelligent Networking and Collaborative Systems. New York, NY: IEEE, 2010, pp. 200–205.
13. Zhang L, Liu X, Janssens F, et al. Subject clustering analysis based on ISI category classification. J Informetr. 2010;4:185–193.
14. Tsiptsis KK, Chorianopoulos A. Data mining techniques in CRM: Inside customer segmentation. West Sussex, UK: John Wiley & Sons, 2011.
15. Tan PN, Steinbach M, Kumar V. Introduction to data mining. Essex, UK: Pearson Education Limited, 2016.
16. Pakhira MK. A linear time-complexity k-means algorithm using cluster shifting. In: 2014 International Conference on Computational Intelligence and Communication Networks. Washington, DC: IEEE Computer Society, 2014, pp. 1047–1051.
17. Fu X, Chen X, Shi YT, et al. User segmentation for retention management in online social games. Decis Support Syst. 2017;101:51–68.
18. Han S, Ye Y, Fu X, et al. Category role aided market segmentation approach to convenience store chain category management. Decis Support Syst. 2014;57:296–308.
19. Khalemsky A, Gelbard R. A dynamic classification unit for online segmentation of big data via small data buffers. Decis Support Syst. 2020;128:113157.
20. Wu HH, Chang EC, Lo CF. Applying RFM model and k-means method in customer value analysis of an outfitter. In: Chou S, Trappey A, Pokojski J (Eds). Global Perspective for Competitive Enterprise, Economy and Ecology. London: Springer, 2009, pp. 665–672.
21. Chen D, Sain SL, Guo K. Data mining for the online retail industry: A case study of RFM model-based customer segmentation using data mining. J Database Market Customer Strategy Manag. 2012;19: 197–208.
22. Marcus C. A practical yet meaningful approach to customer segmentation. J Consum Market. 1998;15:494–504.
23. Jonker JJ, Piersma N, Van den Poel D. Joint optimization of customer segmentation and marketing policy to maximize long-term profitability. Expert Syst Appl. 2004;27:159–168.
24. Cheng CH, Chen YS. Classifying the segmentation of customer value via RFM model and RS theory. Expert Syst Appl. 2009;36:4176–4184.
25. Hwang H, Jung T, Suh E. An LTV model and customer segmentation based on customer value: A case study on the wireless telecommunication industry. Expert Syst Appl. 2004;26:181–188.
26. Namvar M, Gholamian MR, KhakAbi S. A two phase clustering method for intelligent customer segmentation. In: 2010 International Conference on Intelligent Systems, Modelling and Simulation. Washington, DC: IEEE Computer Society, 2010, pp. 215–219.
27. Bacher J, Wenzig K, Vogler M. Spss twostep cluster-a first evaluation. Universität Erlangen-Nürnberg, Wirtschafts- und Sozialwissenschaftliche Fakultät, Sozialwissenschaftliches Institut Lehrstuhl für Soziologie, 2004, pp. 23.
28. Schiopu D. Applying twostep cluster analysis for identifying bank customers' profile. Buletinul. 2010;62:66–75.
29. Agrawal R, Srikant R. Fast algorithms for mining association rules. In: Proceedings of the 20th International Conference on Very Large Data Bases (VLDB). Santiago, Chile. 1994, pp. 487–499.
30. Brin S, Motwani R, Ullman JD, et al. Dynamic itemset counting and implication rules for market basket data. In: Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data, Tucson. 1997, pp. 255–264.
31. Brijs T, Swinnen G, Vanhoof K, et al. Using association rules for product assortment decisions: A case study. In: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, California, USA. 1999, pp. 254–260.

32. Blattberg RC, Kim BD, Neslin SA. Market basket analysis. In: Blattberg R, Kim B, Neslin S (Eds). Database Marketing. New York, NY: Springer, 2008, pp. 339–351.
33. Sarvari PA, Ustundag A, Takci H. Performance evaluation of different customer segmentation approaches based on RFM and demographics analysis. Kybernetes. 2016;45:29.
34. Asratian AS, Denley TM, Haggkvist R. Bipartite graphs and their applications, vol. 131. Cambridge, UK: Cambridge University Press, 1998.
35. Shams B, Haratizadeh S. Graph-based collaborative ranking. Expert Syst Appl. 2017;67:59–70.
36. Blonden VD, Guillaume JL, Lambiotte R, et al. The louvain method for community detection in large networks. J Statist Mechan Theory Exp. 2011;10:P10008.
37. Aynaud T. Community API documentation. Available online at https://python-louvain.readthedocs.io/en/latest (last accessed December 14, 2020).
38. Woma J, Ngo K. Comparisons of Community Detection Algorithms in the YouTube Network, 2019. Available online at http://web.stanford.edu/class/cs224w/project/26425107.pdf (last accessed December 14, 2020).

Cite this article as: Zhang L, Priestley J, DeMaio J, Ni S, Tian X (2021) Measuring customer similarity and identifying cross-selling products by community detection. *Big Data* 9:2, 132–143, DOI: 10.1089/big.2020.0044.

Abbreviations Used

CNM = Clauset–Newman–Moore algorithm
CRM = customer relationship management
RFM = recency, frequency, monetary
LTV = lifetime value