Calculating Product and Customer Sophistication on a Large Transactional Dataset

Diego Pennacchioli[1,3], Michele Coscia[2], Fosca Giannotti[3], Dino Pedreschi[4]

[1] IMT - Lucca, P.za San Ponziano, 6, Lucca, Italy, diego.pennacchioli@imtlucca.it
[2] CID - Harvard University, 79 JFK Street, Cambridge, MA, US, michele_coscia@hks.harvard.edu
[3] KDDLab ISTI-CNR, Via G. Moruzzi, 1, Pisa, Italy, name.surname@isti.cnr.it
[4] KDDLab University of Pisa, Largo B. Pontecorvo, 3, Pisa, Italy, pedre@di.unipi.it

Abstract— **The market basket transactions observed at micro-scale (each individual product bought by each individual customer at each store visit) over a large population for a long time, offer a detailed picture of customers' shopping activity. Given the high cardinality of such a detailed dataset, data mining techniques have been developed to let the hidden knowledge emerge from it. In this technical report, we propose to use the system of all customer-product connections as a whole. We create a framework able to exploit the characteristics of the customer-product matrix and we test it on a unique transaction database, recording the micro-purchases of a million customers observed for several years at the stores of the top national supermarket retailer. We propose it as a novel analytic paradigm for market basket analysis, a paradigm that is challenging both conceptually, given the high complexity of the structures we build, and computationally, given the scale of the data it needs to analyze.**

I. Introduction

Every day, millions of people use supermarkets and shops to fulfill their needs. They buy water, food, products for housework, electronic equipment, cars. Many of these purchases are electronically registered. In such a data rich environment, data mining arose as an useful tool to extract knowledge about customer behavior. Usually, the amount of data is huge. Very large databases and data warehouses are needed to handle this kind of data. In this setting, to apply standard statistical and analytic techniques is hopeless. The need for efficiently extract knowledge from these databases has favored the development of many techniques such as association rule mining [1], fast data clustering, OLAP techniques supporting business intelligence tasks and many more.

Many instances of useful knowledge have been extracted with these data mining techniques. However, so far the knowledge extracted was very specific and particular, and it lacked a general big picture about the data. In this technical report, we propose a methodology aiming at the description of the sophistication of products and customers' needs. We want to use the entire set of all customer-product connections to better understand the hidden knowledge governing the interplay between our desires and needs on one side, and the offered goods and products on the other side.

To do so, we implement a data analysis framework which mainly operates on the characteristics of the customer-product bipartite structure. This framework takes as input the customer-product adjacency matrix and it operates on this matrix with the aim of extracting its general defining pattern. The framework returns a ranking of both customers and products, which describes how much basic or sophisticated is a product, or the needs of a customer. We apply our framework on a unique transaction database. This database has been constructed by a retail supermarket chain in Italy, recording the micro-purchases of a million customers (shopping with their fidelity card that is tracking and making each customer recognizable). The data has been recorded over the 2007-2011 period.

Our results shows that the framework is able to exploit (and quantify) the defining characteristic of the customer-product matrix.

To sum up, our contribution is the developing of an analytic framework able to analyze a transactional dataset as a whole, providing the general picture that association rule mining cannot define. In Section II we present the collection of works related to our project. We describe how to calculate product and customer sophistication in Section IV. Section V provides our experimental part. Section VI concludes the technical report.

II. Related Work

Our approach is a combination of the application and the evolution of some tools present in literature. First, for some specific tasks our framework makes use of the lift measure. The lift (as the conviction, collective strength and many more) is one criterion used in association rule mining to evaluate the interestingness of a rule [2]. Second, we make use of concepts related to ecology literature [3] and macro economy [4], [5]. While using similar techniques (as the eigenvector factorization of the customer-product matrix to calculate the sophistication levels of both customers and products), our work differs from the ones presented on two axis: the first is the quality of the data to which we apply our framework (micro purchases against macro world trade or ecosystem presence/absence of animal species); the second is the quantity of data, as we work with matrices with a number of cells $\sim 10^9$ while related works do not scale beyond $\sim 10^5$ and therefore cannot be used with big data.

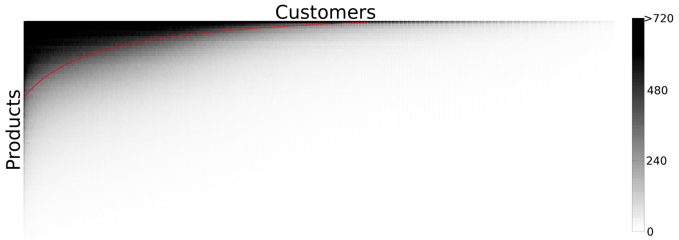

Fig. 1: The M_{cp} purchase matrix. For layout purposes, the matrix has been transposed, thus we have customers as columns and products as rows. The red line is the isocline of the matrix, highlighting the triangular structure of the matrix by dividing most of the ones (top left) from most of the zeroes (bottom right). Color image.

III. Data Preparation

To analyze the bipartite customer-product structure, we decide to deal with the adjacency matrix representing its connections. Since we want to analyze the aggregate behavior of customers and verify whether some patterns emerge on the relation between customers and products, we need to arrange the rows and the columns of the adjacency matrix in a logical way. Hence we sorted the matrix with the following criterion: fixing the top-left corner of the matrix M as the origin, we sorted the customers on the basis of the sum of the items purchased in descending order (the top buying customer at the first row and so on), and the products with the same criteria from left to right (the top sold product at the first column and so on). In this way, at the cell $(0, 0)$ we can find the quantity of top sold product purchased by the top buying customer. Using this criterion, we exploit the log-normal degree distributions of the bipartite structure, showing that the best sold products are bought by all kinds of customers, while products with a low market share are bought exclusively by customers who buy everything. This consideration is at the basis of the customer and product sophistication calculation in Section IV.

The final step of data preparation is to binarize the matrix, by identifying what purchases are significant and what are not. We cannot simply binarize the matrix considering the purchase presence/absence of a customer for a product. A matrix with a 1 if the customer c_j purchased the product p_i and 0 otherwise will result in a certain amount of noise: it takes only a single purchase to connect a customer to a product, even if generally the customer buys large amounts of everything else and the product is generally purchased in larger amount by every other customer.

We need a mechanism to evaluate how meaningful is a purchase quantity for each product p_i for each customer c_j. This evaluation is done using the concept of lift [1], that is related to association rule mining. Given a couple of itemsets (X, Y), the lift of the couple is defined as follows:

$$\text{lift}(X, Y) = \frac{\text{supp}(X, Y)}{\text{supp}(Y) \times \text{supp}(X)},$$

where $\text{supp}(I)$ is the relative support of the itemset I. The relative support of itemset I is the number of times all $i \in I$ are purchased together over all the transactions present in the dataset.

In our case, we force a particular condition: the itemset X always contains one item (the customer c_j); the itemset Y always contains one element (the product p) and the support of (c_j, p_i) is given by the corresponding entry in the matrix. In other words, $\text{supp}(c_j, p_i)$ is the relative amount of product p_i bought by customer c_j, $\text{supp}(p_i)$ is the relative amount sold of product p_i to all customers and $\text{supp}(c_j)$ is the relative amount of products bought by customer c_j.

Lift takes values from 0 (when $\text{supp}(c_j, p_i) = 0$, i.e. customer c_j never bought a single instance of product p_i) to $+\infty$. When $\text{lift}(c_j, p_i) = 1$, it means that $\text{supp}(c_j, p_i)$ is exactly the expected value, i.e. the connection between customer c_j and product p_i has the expected weight. If $\text{lift}(c_j, p_i) < 1$ it means that the customer c_j purchased the product p_i less than expected, and viceversa. Therefore, the value of 1 for the lift indicator is a reasonable threshold to discern the meaningfulness of the quantity purchased: if it is strictly higher, then the purchases are meaningful and the corresponding cell in the binary matrix is 1; otherwise the purchases are not meaningful, even if some purchases are actually made, and the corresponding cell in the binary matrix is 0. The M_{cp} matrix is built accordingly to this rule:

$$M_{cp} = \begin{cases} 1 & \text{if } \text{lift}(c_j, p_i) > 1; \\ 0 & \text{otherwise.} \end{cases}$$

This is the final output of the preprocess phase, hence from now on it will be referred as the purchase matrix and $M_{cp}(c_j, p_i)$ is the entry of M_{cp} of row j and column i. We provide an example of an M_{cp} matrix in Figure 1, that is the M_{cp} matrix extracted in the Livorno2007-2009 dataset (see Section V-A). In Figure 1, the columns of the matrix are the $317,269$ customers and the rows are the $4,817$ products. We depicted a compressed view of the matrix, where each data dot represent a 50×50 square of the original matrix and the gray gradient represents how many $1s$ are present in that section of the matrix, for space constraints.

We can observe in Figure 1 the phenomenon we described at the beginning of the section: only a small amount of popular products are bought by everyone, but a smaller and smaller set of customer purchases the rest of the products (going from the right to the left columns) and it is always composed by the same set of big buyers.

IV. Product and Customer Sophistication

In the main core of the framework, we want to quantify the sophistication level of the products sold and of the customers buying products. The basic intuition is that more sophisticated products are by definition less needed, as they are expression of a more complex need. One may be tempted to answer to this question by trivially returning the products in descending order of their popularity or price: the more a product is sold, the more basic it is. However, this is not considering an important aspect of the problem: to be sold to a large set of costumers is a necessary condition to be considered “basic”, but it is not sufficient. Another necessary condition is that the set of customers buying the product should include the set of costumers with the lowest level of sophistication of their needs. The conjunction of the two properties is now sufficient to define a product as “basic”.

This conjunction is not trivial and it is made possible by the triangular structure of the adjacency matrix. Consider Figure 1: the columns in the right part of the matrix are those customers buying only few products. Those products are more or less bought by everyone. We need to evaluate at the same time the level of sophistication of a product and of the needs of a customer using the data in the purchase matrix, and recursively correct the one with the other. We adapt the procedure of [5], adjusting it for our big data.

We calculate the sums of the purchase matrix for each customer ($k_{c,0} = \sum_p M_{cp}(c,p)$) and product ($k_{0,p} = \sum_c M_{cp}(c,p)$). To generate a more accurate measure of the sophistication of a product we need to correct the sums recursively: this requires us to calculate the average level of sophistication of the customers' needs by looking at the average sophistication of the products that they buy, and then use it to update the average sophistication of these products, and so forth. This can be expressed as follows: $k_{N,p} = \frac{1}{k_{0,p}} \sum_c M_{cp} k_{c,N-1}$. We then insert $k_{c,N-1}$ into $k_{N,p}$ obtaining:

$$k_{N,p} = \frac{1}{k_{0,p}} \sum_c M_{cp} \frac{1}{k_{c,0}} \sum_{p'} M_{cp'} k_{N-2,p'}$$

$$k_{N,p} = \sum_{p'} k_{N-2,p'} \sum_c \frac{M_{cp} M_{cp'}}{k_{0,p} k_{c,0}}$$

and rewrite this as:

$$k_{N,p} = \sum_{p'} \widetilde{M}_{pp'} k_{N-2,p'},$$

where:

$$\widetilde{M}_{pp'} = \sum_c \frac{M_{cp} M_{cp'}}{k_{0,p} k_{c,0}}.$$

We note in the last formulation $k_{N,p}$ is satisfied when $k_{N,p} = k_{N-2,p}$ and this is equal to a certain constant a. This is the eigenvector of which is associated with the largest eigenvalue (that is equal to one). Since this eigenvector is a vector composed by the same constant, it is not informative. We look, instead, for the eigenvector associated with the second largest eigenvalue. This is the eigenvector associated with the variance in the system and thus it is the correct estimate of product sophistication.

However, this formulation is very sensitive to noise, i.e. products that are bought only by a very narrow set of customers. To calculate the eigenvector on the entire set of products generates a small amount of products whose sophistication level is seven orders of magnitude larger than the rest of the products. This variance provokes the other sophistication estimates to be flattened down to the same values and therefore not meaningful. However, we do not want to simply cut the least sold products, as we aim to create a full product hierarchy, including (especially) also the least sold products. To normalize this, we employ a three step strategy. First, we calculate the eigenvector on a restricted number of more popular products (purchased by at least a given threshold δ of customers). Then we use the estimate of the sophistication of these products to estimate the sophistication of the entire set of customers (that is, as defined before, the average sophistication of the restricted set of products they buy). Finally, we use the estimated sophistication of the customers to have the final sophistication of the entire set of products, again by averaging the sophistication of the customers buying them. Hence, we define the product sophistication index (PS) as:

$$PS = -\frac{\vec{K} - \mu(\vec{K})}{\sigma(\vec{K})},$$

where $\vec{K}$ is the eigenvector of $\widetilde{M}_{pp'}$ associated to the second largest eigenvalue, normalized as described above; $\mu(\vec{K})$ is its average and $\sigma(\vec{K})$ its standard deviation. The Customer Sophistication CS is calculated using the very same procedure, by estimating $k_{c,N}$ instead of $k_{N,p}$.

V. Experiments

All the analysis presented in this section are performed with regular user-end computers. No mainframes or parallel computing techniques have been used. The eigenvector computing has been performed in less than one hour on a Dual Core Intel i7 64 bits @ 2.8 GHz laptop, equipped with 8 GB of RAM and with a kernel Linux 3.0.0-12-generic (Ubuntu 11.10), using a combination of Octave, Numpy and Scipy Python libraries.

A. The Data

Our analysis is based on real world data about customer behavior. For this reason, we use a real world dataset large enough (in terms of customers, variety of products and time window) to be considered a good representation of the reality, but fine-grained (in terms of information granularity), giving us the possibility to choose both the best level of aggregation and the right functions to do that.

The dataset we used is the retail market data of one of the largest Italian retail distribution company. The conceptual data model of the data warehouse is depicted in Figure 2. The whole dataset contains retail market data in a time window that goes from January 1st, 2007 to December, 31st 2011. The active and recognizable customers are $1,066,020$. A customer is active if he/she has purchased something during the data time window, while he/she is recognizable if the purchase has been made using a membership card. The 138 stores of the company cover the whole west coast of Italy, selling $345,208$ different items.

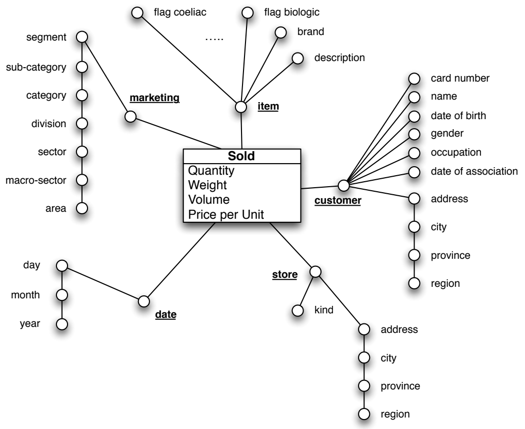

Fig. 2: The data model of the Data Warehouse

An important dimension of the data warehouse is Marketing, representing the classification of products: it is organized as a tree and it represents a hierarchy built on the product typologies, designed by marketing experts of the company. The top level of this hierarchy is called “Area” that split the products into two fundamental categories: “Food” and “No Food”. The bottom level of the hierarchy, the one that contains the leaves of the tree, is called “Segment” and it contains $7,003$ different values (see Figure 2). Hence, for each item contained in the dataset, there is an entry assigning it to the right path of the hierarchy tree.

Considering that the dataset contains more than one million customers and almost 350k items, to build a matrix “customers × items” would generate $\sim$ 370 billions of cells, that is redundant for our purposes; hence we need a sort of reduction on both the dimensions (customers and items). There are two main criteria to select the customers: on the basis of their purchase behavior (e.g. excluding from the analysis all the people that did not purchased at least a total number x items) or geographically (e.g. considering just the customers of an area). We decided to apply the latter filter, since we do not want to exclude any customer behavior. We select a subset of shops in the dataset belonging to the same areas of Italy. The number of customers per area is presented in Figure 3. We generated different views of the dataset for different purposes. Our main dataset is Livorno2007-2009, that is including all the purchases of the customers located in the city of Livorno during the period from 2007 to 2009. We use only this view for the applications of the framework’s output. We also generated the dataset Lazio2007-2009 (same period, different geographical location, the sum of the cities of Rome, Viterbo, Latina, Rieti and Frosinone) and Livorno2010-2011 (different period, same geographical location). These two views are generated to prove that the fundamental properties of the adjacency matrix needed for our framework are not bounded to a particular place or time. The following steps of data preparation are applied equally to the different datasets extracted.

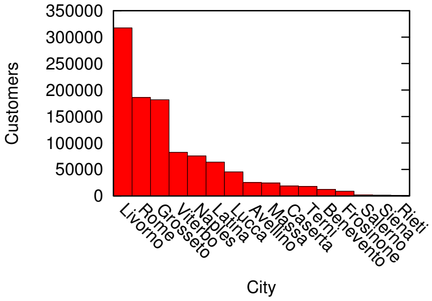

Fig. 3: Customer distribution per city in our dataset.

The second issue, as introduced above, regards the cardinality of products. There is a conceptual problem in using the level of detail of “item”: the granularity is too fine, making the analysis impractical as it would consider a very low detail level. The distinction between different packages of the same product, e.g. different sizes of bottles containing the same liquid, is not interesting here. A natural way to solve this problem is to use the marketing hierarchy on the products, substituting the item with the value of the marketing Segment. In this way, we reduce the cardinality of the dimension of the product by 98% (from $345,208$ to $7,004$), aggregating at the same time products that are equivalents.

The last step in data selection is to exclude from the analysis all the products (segments) that are either too frequent (e.g. the shopper) or meaningless for the purchasing analysis (e.g. discount vouchers, errors, segments never sold, etc.). After this last filter (and consequently the discharge of the customers that bought exclusively products classified under the removed segments), we got our adjacency matrix, ready to be provided as input of our framework. Livorno2007-2009 matrix has $317,269$ customers and $4,817$ segments, with $182,821,943$ purchases; Livorno2010-2011 has $326,010$ customers and $4,807$ segments, with $183,679,550$ purchases; and Lazio2007-2011 has $278,154$ customers and $4,641$ segments, with $135,517,300$ purchases.

B. *Framework Application*

In this section, we apply our framework on the three views of the dataset extracted. We report the application of each step.

Calculation of the M_{cp} matrices from the adjacency matrices. The results is three M_{cp} matrices for Livorno2007-2009, Livorno2010-2011 and Lazio2007-2009. The number of rows and columns of the matrices are not changed, and the total number of ones (i.e. significant purchases according to the lift) are $37,338,591$ for Livorno2007-2009, $43,982,774$ for Livorno2010-2011 and $45,410,992$ for Lazio2007-2009. Livorno2007-2009 matrix is depicted in Figure 1, Livorno2010-2011 and Lazio2007-2009 matrices are depicted in Figure 4, left and right respectively (the legend for both figures is the same as Figure 1 legend).

Calculation of the Product and Customer Sophistication. We present the most and least sophisticated products only for the Livorno2007-2009 dataset, for the same reason of the previous

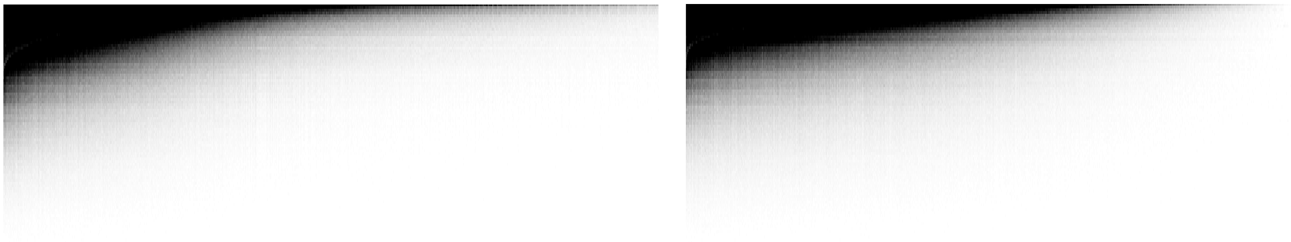

Fig. 4: The M_{cp} matrices for Livorno2010-2011 (left) and Lazio2007-2009 (right).

p_i	PS
Regular Bread	-4.41
Natural Still Water	-4.19
Yellow Nectarines (Peaches)	-3.84
Semi-Skimmed Fresh Milk	-3.81
Bananas	-3.53

TABLE I: A selection of the more basic products according to their PS values.

p_i	PS
LCD 28"/30" Televisions	2.91
DVD Music Compilations	2.86
Sauna clothing	2.66
Jewelry Bracelets	2.53
RAMs for PC	2.33

TABLE II: A selection of the more sophisticated products according to their PS values.

point. Also we do not report the Customer Sophistication for privacy concerns. In Table I we report a selection of the least sophisticated products, i.e. to ones with the lowest PS values, in the purchase matrix. The less sophisticated products should be intuitively the ones covering the most basic human needs, and this intuition is confirmed by the reported products: bread, water, fruits and milk. On the other hand, Table II reports the most sophisticated products, i.e. the ones with the largest PS values, that intuitively should be products satisfying high-level non-necessary, probably luxury, needs. In fact, what we find in Table II are hi-tech products (LCD televisions, DVD compilations, computer accessories), jewelry and very specific clothing.

VI. Conclusion

In this paper we analyzed large quantities of data extracted from the retail activity of the customer subset of an Italian supermarket chain. Our aim was to build a framework able to take advantages of some properties of the data, providing an alternative and complementary methodology to mine purchase data. The main used property is the triangular structure of the customer-product adjacency matrix. We found that customers usually start buying the same set of basic products and the more sophisticated products are only bought by customers buying everything, providing a triangular adjacency matrix for the bipartite structure. Our framework is able to analyze this structure as a whole, instead of looking at the local patterns like classical rule mining. From this consideration, we were able to define a function that can rank the sophistication level of both products and customer needs.

Our work opens the way to several different future developments. The first one concerns the validation of our observation of the product/customer sophistication indexes, as it is based on a narrow geographical set of people and on a non-standard product category classification. Also, with more data we can create an empirical observation of Maslow's pyramid of needs. Another interesting track of research may be to investigate what is the minimum time window needed to observe the triangularity of the matrix, maybe linked with the cyclic behavior of customers [6] and/or with the stability of customer and product ranking order in the matrix [7]. Another application scenario may be to fully exploit the purchase matrix as a complex system: to analyze products not only based on their product sophistication index, but by looking at the product-product relationship level; or to try to find the way of controlling the complex system [8].

Acknowledgments. We gratefully thank Muhammed Yildirim, César Hidalgo, Salvatore Rinzivillo and Sebastian Bustos for their support. We also thank the supermarket company that shares the data with us and allow us to analyze and to publish the results.

References

[1] R. Agrawal, T. Imielinski, and A. N. Swami, "Mining association rules between sets of items in large databases," in *SIGMOD International Conference*, Washington, D.C., 1993, pp. 207–216.

[2] L. Geng and H. J. Hamilton, "Interestingness measures for data mining: A survey," *ACM Comput. Surv.*, vol. 38, no. 3, pp. 9+, 2006.

[3] J. Bascompte, P. Jordano, C. J. Melián, and J. M. Olesen, "The nested assembly of plantanimal mutualistic networks," *PNAS*, vol. 100, no. 16, pp. 9383–9387, Aug. 2003.

[4] C. A. Hidalgo, B. Klinger, A. L. Barabási, and R. Hausmann, "The product space conditions the development of nations," *Science*, vol. 317, no. 5837, pp. 482–487, July 2007. [Online]. Available: http://dx.doi.org/10.1126/science.1144581

[5] C. A. Hidalgo and R. Hausmann, "The building blocks of economic complexity," *Proceedings of the National Academy of Sciences*, vol. 106, no. 26, pp. 10 570–10 575, June 2009.

[6] Z.-J. M. Shen and X. Su, "Customer behavior modeling in revenue management and auctions: A review and new research opportunities," *Production and Operations Management*, vol. 16, no. 6, pp. 713–728, 2007. [Online]. Available: http://dx.doi.org/10.1111/j.1937-5956.2007.tb00291.x

[7] M. Schich, S. Lehmann, and J. Park, "Dissecting the canon: Visual subject co-popularity networks in art research," in *ECCS2008*, September 2008.

[8] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, "Controllability of complex networks," *Nature*, vol. 473, no. 7346, pp. 167–173, May 2011. [Online]. Available: http://dx.doi.org/10.1038/nature10011