Enabling Decision Support Through Ranking and Summarization of Association Rules for TOTAL Customers

Idir Benouaret, Sihem Amer-Yahia, Senjuti Basu Roy, Christiane Kamdem-Kengne, Jalil Chagraoui

Enabling Decision Support Through Ranking and Summarization of Association Rules for TOTAL Customers

Idir Benouaret[1], Sihem Amer-Yahia[1], Senjuti Basu Roy[2], Christiane Kamdem-Kengne[3], and Jalil Chagraoui[3]

[1] CNRS, Univ. Grenoble Alpes, Grenoble, France
fname.lname@univ- grenoble-alpes.fr*
[2] New Jersey Institute of Technology Neward, NJ, USA
senjutib@njit.edu
[3] TOTAL
fname.lname@total.com

Abstract. Our focus in this experimental analysis paper is to investigate existing measures that are available to rank association rules and understand how they can be augmented further to enable real-world decision support as well as providing customers with personalized recommendations. For example, by analyzing receipts of TOTAL customers, one can find that, customers who buy windshield wash, also buy engine oil and energy drinks or middle-aged customers from the South of France subscribe to a car wash program. Such actionable insights can immediately guide business decision making, e.g., for product promotion, product recommendation or targeted advertising. We present an analysis of 30 million unique sales receipts, spanning 35 million records, by almost 1 million customers, generated at 3,463 gas stations, over three years. Our finding is that the 35 commonly used measures to rank association rules, such as Confidence and Piatetsky-Shapiro, can be summarized into 5 synthesized clusters based on similarity in their rankings. We then use one representative measure in each cluster to run a user study with a data scientist and a product manager at TOTAL. Our analysis draws actionable insights to enable decision support for TOTAL decision makers: rules that favor Confidence are best to determine which products to recommend and rules that favor Recall are well-suited to find customer segments to target. Finally, we present how association rules using the representative measures can be used to provide customers with personalized product recommendations.

Keywords: Data Mining · Association Rules · Recommendation.

1 Introduction

Association rule mining [1] is one of the most frequently used techniques to analyze customers' shopping behavior and derive actionable insights to enable

* Our work is funuded by a grant from TOTAL

decision support. Like many others in the retail industry, marketers and product managers at TOTAL conduct regular studies of customer preferences and purchasing habits. The goal of those studies is to determine two main decisions: *which products to bundle together in a promotional offer and which customers to target.* Those studies usually focus on unveiling the interest of customers for *specific products or categories* (e.g., *tire service, gas, food*) or the behavior of *pre-defined customer segments*. However, when the underlying dataset is extremely large, such as the one we use for our analysis from TOTAL (30 million receipts spanning 35 million records), it can create an explosion of association rules; therefore one has to make use of existing ranking measures of association rules, such as, *Support*, *Confidence*, *Piatetsky-Shapiro*, *Lift*, etc to rank the rules. Even after that, as there exists many ranking measures (as many as 35) [4,12], there may not be enough guideline to understand which ranking measure is to be leveraged for what types of decision making task, unless these ranking measures are further summarized.

To address that, we leverage the power of association rule mining and ranking measures for marketers to extract actionable insights from large volumes of consumer data. To make the outcome tightly aligned with the practitioners need, our workflow consists of the following 4 steps: **Step 1:** We empower non-scientist domain experts with the ability to express and analyze association rules of interest. **Step 2:** we summarize the ranking measures into a set of synthesized clusters or groups. The outcome of this process is a 5 synthesized clusters (or groups) that summarize the ranking measures effectively. **Step 3:** We allow the domain expert non-scientists to provide feedback on the synthesized clusters. **Step 4:** We show how this process can *provide actionable insights and enable decision support for virtually any customer segment* and *any product*. Our analysis shows: rules that favor Confidence are best to determine which products to promote and rules that favor Recall are well-suited to find customer segments to target.

To the best of our knowledge, this work is the first to run a large-scale empirical evaluation of insights on customer purchasing habits in the *oil* and *gas* retail domain. We summarize our contributions as follows : (i) a reproducible methodology for experimenting with different association rule ranking methods; (ii) several insights on real large-scale datasets; (iii) how to use association rules and interestingness measures in computing recommendations.

1.1 Empowering domain experts

When analysts seek to determine which products to run a promotion for or which customers to target, they conduct small to medium-scale market analysis studies. Such studies are expensive, time-consuming and hardly reproducible. We use association rule mining to unveil valuable information about any customer segment and any product. Our collaboration with analysts at TOTAL resulted in the formalization of two kinds of purchasing patterns: those representing associations between a set of products and a single product (*customers who wash their*

cars and purchase wipes also purchase a windshield washer), and those associating customer segments to a product category (*young customers in the south of France who frequently wash their cars*).

Our dataset contains 30 million unique receipts, spanning 35 million records, generated at by 1 million customers at 3,463 gas stations , over three years (from January 2017 to December 2019). The ratio 30/35 is due to the fact that, unlike in regular retail such as shopping grocery stores [12], most customers at gas stations purchase *gas* only, and a few purchase additional products such as *car wash*, *drinks* and *food* items.

Based on our initial discussion with `TOTAL` analysts, we propose two mining scenarios to capture desired purchasing patterns. The goal is to help analysts who are not necessarily tech-savvy, express their needs. In the first scenario, `prod_assoc`, the analyst specifies a target product and expects rules of the form *set of products* $\rightarrow$ *target product*. In the second scenario, `demo_assoc`, the analyst specifies a target product category and expects rules of the form *customer segment* $\rightarrow$ *category*, i.e. customers who purchase products in that category. Each scenario requires to ingest and prepare data as a set of transactions. The transactions are fed to jLCM [16], our open-source parallel and distributed pattern mining algorithm that runs on MapReduce [13], to compute association rules. To cope with the skewed distribution of our transactions, jLCM is parameterizable and is used to mine per-item top-k itemsets.

1.2 Ranking and summarization of association rules

Regardless of the mining scenario, the number of resulting rules can quickly become overwhelming. As an example, for a single target product: *TOTAL wash* and with a 1,000 minimum support, jLCM mines 4,243 frequent rules of the form *set of products* $\rightarrow$ *TOTAL wash*. Out of these, 805 have a *Confidence* of 50% or higher. Table 1 shows a ranking of the top-5 rules for the product category *Lubricants* and the top-5 rules for the product *Coca Cola* , sorted using 2 different interestingness measures proposed in the literature [4]. Given the rule $\mathcal{A} \rightarrow \mathcal{B}$, *Confidence* is akin to precision and is defined as the probability to observe $\mathcal{B}$ given that we observed $\mathcal{A}$, i.e., $P(\mathcal{B}|\mathcal{A})$. *Piatetsky-Shapiro* [22] combines how $\mathcal{A}$ and $\mathcal{B}$ occur together with how they would if they were independent, i.e., $P(\mathcal{AB}) - P(\mathcal{A})P(\mathcal{B})$. *Recall* is defined as the probability to observe $\mathcal{A}$ given that we observed $\mathcal{B}$. Clearly, different measures yield different rule rankings for both `prod_assoc` and `demo_assoc`.

To ease the burden on analysts, we propose to examine the rankings induced by existing measures (exactly 35 measures [4]) and attempt to reduce them based on similarities in rankings. We run our measures to rank association rules for 228 representative products in `prod_assoc` and for 16 representative product categories in `demo_assoc`. In each case, we use hierarchical clustering to summarize or group the rule rankings based on their similarities (we use multiple list similarities to compare rankings). Our finding is that existing measures can be clustered into 5 similar synthesized groups regardless of the mining scenario. The clusters we obtained are summarized in Table 3. They differ in their emphasis on

Table 1. Top-5 demographics association rules, Top-5 products association rules, according to different interestingness measures. For `demo_assoc`, rules are denoted {age, gender, region} $\rightarrow$ *target category*. Product category was translated to English, French regions were left unchanged. For `prod_assoc`, rules are denoted {set of products} $\rightarrow$ *target product*. Products were translated to English.

by confidence	**by Piatetsky-Shapiro [22]**
{*50-65*, *M*,*Ile-de-France*} $\rightarrow$ *Lubricants*	{*50-65*, *M*, $*$} $\rightarrow$ *Lubricants*
{*50-65*, $*$, *Ile-de-France*} $\rightarrow$ *Lubricants*	{$*$, *M*,*Ile-de-France*} $\rightarrow$ *Lubricants*
{$>$ 65, *M*,*Ile-de-France* } $\rightarrow$ *Lubricants*	{$*$, $*$,*Ile-de-France* } $\rightarrow$ *Lubricants*
{$>$ 65, *M*, $*$} $\rightarrow$ *Lubricants*	{$*$, *F*, $*$} $\rightarrow$ *Lubricants*
{*50-65* , *M*,*Hauts-de-France* } $\rightarrow$ *Lubricants*	{$*$, *M*, $*$} $\rightarrow$ *Lubricants*
{ *Bbq chips, Ham sandwish* } $\rightarrow$ *Coca Cola*	{*Coffee* } $\rightarrow$ *Coca Cola*
{*Cheese sandwish, Bbq chips* } $\rightarrow$ *Coca Cola*	{*Fuze Peche* } $\rightarrow$ *Coca Cola*
{*Bbq chips, Salted chips* } $\rightarrow$ *Coca Cola*	{*Insulated bottle*} $\rightarrow$ *Coca Cola*
{*Chicken sandwish, Salted chips* } $\rightarrow$ *Coca Cola*	{*Mars legend* } $\rightarrow$ *Coca Cola*
{*Chicken sandwish, Bbq chips* } $\rightarrow$ *Coca Cola*	{ *Snickers* } $\rightarrow$ *Coca Cola*

Confidence and *Recall*. In the case of `prod_assoc`, we observe high *Confidence* and low *Recall* for G_1 which contains 18 measures among which *Lift*, and for G_2 which contains 3 measures among which *Accuracy*. G_3 which contains 7 measures among which *J-measure*, achieves a good tradeoff between *Confidence* and *Recall*. G_4 contains 5 measures among which *Piatetsky-Shapiro* and achieves average *Confidence* and high *Recall*. G_5 contains 2 measures among which *Recall* and is characterized by the lowest *Confidence* and highest *Recall* among all groups. In the case of `demo_assoc`, we observe the same groups G_1 and G_2 with also a high *Confidence* and low *Recall*. G'_3 which contains 7 measures among which *Klosgen*, achieves a good tradeoff between *Confidence* and *Recall*. G'_4 contains 3 measures among which *Two-way support variation* and achieves low *Confidence* and high recall. Finally, G'_5 contains 4 measures among which *Recall* and is characterized by the lowest *Confidence* and the highest *Recall* overall.

1.3 Gathering feedback

The reduction of the number of interestingness measures to rank rules enabled us to conduct a user study with 2 analysts, one data scientist and one product manager (co-authors of this paper), at `TOTAL` to address the following question: **out of the 5 groups of similar interestingness measures, which ones return actionable rules?** Actionable rules are ones that can be used by analysts either to promote products or to find customer segments to target. Our study lets analysts compare 2 (hidden) ranking measures at a time for a given scenario and a given target product or category. Our first deployment was deemed "reassuring" and "unsurprising". A joint examination of the results identified two issues: (1) rules contained many "expected associations", i.e., those resulting from promotional offers that already occurred; (2) many rules were featuring "familiar" items, i.e., frequently purchased ones. After filtering unwanted items such as

gas, plastic bags, etc and offers, we ran a second deployment with our analysts. Their interactions with returned association rules (of the form $\mathcal{A} \rightarrow \mathcal{B}$, where $\mathcal{B}$ is a product or a category) were observed and their feedback recorded. This deployment yielded two insights: rankings that favor *Confidence*, i.e., $P(\mathcal{B}|\mathcal{A})$, are best to determine which products to promote while rankings that favor *Recall*, i.e., $P(\mathcal{A}|\mathcal{B})$, are well-suited to find which customer segments to target. Confidence represents how often the consequent is present when the antecedent is, that is, $P(\mathcal{B}|\mathcal{A})$, and confidence-based ranking can be used to determine which A products to bundle with a target product $\mathcal{B}$ to promote $\mathcal{B}$. Recall represents the proportion of target items that can be retrieved by a rule, that is, $P(\mathcal{A}|\mathcal{B})$, and recall-based ranking can be used to determine which customer segments $\mathcal{A}$ to target with $\mathcal{B}$.

1.4 Product recommendation

Finally, we show how association rules can effectively be used to perform product recommendation using different interestingness measures. Clustering the overwhelming number of interestingness measures into 5 synthesized clusters enabled us to conduct an offline experiment to test the effectiveness of each cluster of measures to generate accurate product recommendations. We split our data using the available timestamps into a training set (transactions from January 2017 to December 2018) and a test set (transactions from January 2019 to December 2019), i.e., we extract association rules based on past purchases to predict future purchases. The obtained accuracy results are consistent with our clustering as well as the preference of our analysts for measures that favor *Confidence* for product recommendation.

In summary, this paper presents a joint effort between researchers in academia and analysts at `TOTAL`. We leverage the power of association rule mining and augment them with the power of rule ranking and summarization to guide decision support as well as the ability of performing product recommendation. The rest of the paper could be summarized as follows: The background and the goal of the work are provided in Section 2. Our underlying process using `TOTAL` datasets is described in Section 3. In Section 4, we describe how we summarize (cluster) interestingness measures based on similarities in rule rankings. These clusters are then evaluated by analysts in Section 5 leading to insightful findings. We discuss how to turn use our findings into product recommendation through association rule ranking in Section 6. The related work is summarized in Section 7. We conclude in Section 8.

2 Background and Overall Goal

We describe the `TOTAL` dataset, the mining scenarios, and interestingness measures used to rank association rules, and finally we state our goal.

2.1 Dataset

Our dataset represents customers purchasing products at different gas stations that are geographically distributed in France, for a period of two years (from January 2017 to December 2018). The dataset $\mathcal{D}$ is a set of records of the form $\langle t, c, p \rangle$, where t is a unique receipt identifier, c is a customer, and p is a product purchased by c. The set of all receipt identifiers is denoted T. Each receipt identifier is associated with a unique customer, and multiple receipt identifiers can be associated with the same customer according to his/her visits to different gas stations. When a customer purchases multiple products in the same visit to a gas station, several records with the same receipt identifier t are generated.

The complete dataset contains over 30 million unique receipts, spanning 35 million records, generated at $3,463$ gas stations, over three years. The ratio 30/35 in our dataset is due to the fact that, unlike regular retail such as as shopping grocery stores [12], most customers at gas stations purchase *gas* only, and a few of them purchase additional products such as car services (*oil change*, *car wash*), *drinks* and *food* items.

The set of customers, $\mathcal{C}$, contains over 1 million unique records. Each customer has demographic attributes. In this study, we focus on 3 attributes: *age, gender* and *location*. The attribute *age* takes values in $\{< 35, 35-49, 50-65, > 65\}$ and the attribute *location* admits French regions as values. We use $demographics(c)$ to refer to the set of attribute values a customer c belongs to. For example, $\{ < 35$,*F,Ile-de-France*$\}$ represents a 28 years old *female* from the *Ile-de-France* region, whom we will refer to as *Mary*. The attributes are used to form customer segments. Each segment is described by a set of user attribute values that are interpreted in the usual conjunctive manner. For example, the segment $\{< 35, *,$ *Ile-de-France*$\}$ refers to young customers from the *Ile-de-France* region and the segment $\{> 65, M,$ *Normandie*$\}$ refers to Senior Male customers from the *Normandie* region.

The set of products $\mathcal{P}$ contains over $37,556$ entries, out of which 976 have been sold more than a thousand times. Each product p is associated with a product category. our dataset contains 54 different categories including *gas, lubricants, car wash, hot drinks,* and *sweets*. We use $cat(p)$ to denote the category of a product p.

2.2 Mining Customer Receipts

We describe our data preparation process - that is how to translate the sale receipts to a transactional dataset that could be further injected to the mining process. We then describe the mining scenarios and present interestingness measures to rank association rules.

Dataset Preparation Figures 1, 2 and 3 report statistics on one month in the dataset which contains $407,212$ sales records generated by $257,102$ customers for $5,479$ products at $3,079$ gas stations. For confidentiality reasons, we do not report the statistics of the full dataset. We can however state that other periods

Table 2. Our mining scenarios and example association rules.

Target Associations	**Associations and $\mathcal{T}$**
`demo_assoc`:	$\{demo(c) \cup cat(p) \mid \langle t, c, p\rangle \in \mathcal{D}\}$
segment $\rightarrow$ *category*	min support is 1,000
`prod_assoc`:	$\{\cup_{\langle t_j, c, p_i\rangle \in \mathcal{D}} p_i \mid c \in \mathcal{C}\}$
product(s) $\rightarrow$ *product*	min support is 1,000
Target Associations	**Desired Association Rules**
`demo_assoc`:	A segment of customers who are likely to purchase products in a given category
segment $\rightarrow$ *category*	$\{< 35, F, *\} \rightarrow$ *car wash*
`prod_assoc`:	Customers who purchase a set of products and are likely to purchase the target product
product(s) $\rightarrow$ *product*	{*Bbq Chips, Snickers Bar*}$\rightarrow$ *Coca Cola*

in the dataset exhibit similar distributions. The statistics clearly show that the most purchased items are *gas* and that most transactions are short.

To gain an understanding of customers' buying habits and provide them with relevant offers, analysts from `TOTAL` are interested in studying two kinds of purchasing patterns: those associating a set of products to a single product (*customers who wash their cars and purchase wipes also purchase a windshield washer*) and those representing associations between customer segments and a product category (*young customers in the south of France who frequently wash their cars*). In all cases the analyst specifies a rule target $\mathcal{B}$ which corresponds to a product or a product category, and expects rules of the form $\mathcal{A} \rightarrow \mathcal{B}$.

In the first scenario, that we denote `prod_assoc`, the analyst specifies a target product and is shown rules of the form *set of products* $\rightarrow$ *target product*, i.e. customers who purchase the set of products are likely to purchase the target product. In the second scenario, that we denote `demo_assoc`, the analyst specifies a target category, and is shown rules of the form *customer segment* $\rightarrow$ *target category*, i.e. customers who belong to some segment are likely to purchase products in the target category.

In both scenarios, the original dataset $\mathcal{D}$ is mapped into a collection of transactions $\mathcal{T}$ that is given as input to the mining process, as summarized in Table 2. The set $\mathcal{T}$ is built differently according to each scenario.

In the first scenario `prod_assoc`, we generate the set of transactions $\mathcal{T}$ by grouping records in $\mathcal{D}$ by customer identifiers. For each customer c, we generate a single transaction containing the set of all products ever purchased by c $\{p \mid \langle t, c, p\rangle \in \mathcal{D}\}$. We obtain $|\mathcal{C}|$ transactions, each of which is a subset of $\mathcal{P}$. This enables the discovery of customer patterns occurring over several visits to a station.The number of transactions in `prod_assoc` is $1,083,901$, where each transaction contains 7 products on average.

In the second scenario `demo_assoc`, a transaction is a tuple built for each record $\langle t, c, p\rangle$ by associating the customer segment $demographics(c)$ with the corresponding category of the product $cat(p)$. For example, an entry in the raw data consisting of the record $\langle 4523768, Mary, tea\rangle$ is mapped to the transaction

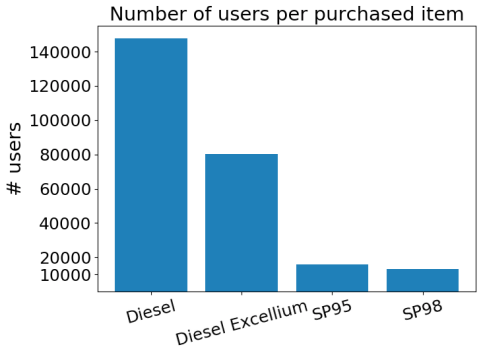


Fig. 1. Most purchased items

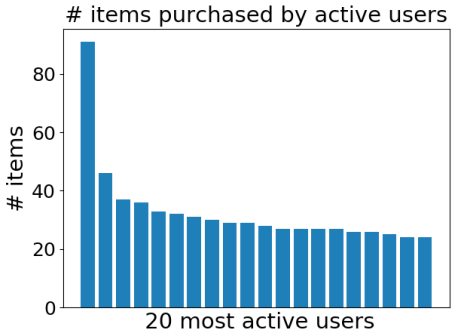


Fig. 2. Most active users

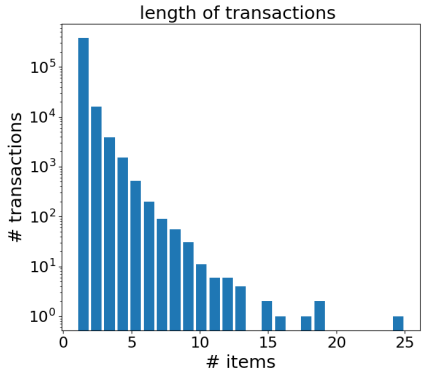


Fig. 3. Length of transactions

$\langle < 35, F,$ *Ile-de-France*, *hot drinks* $\rangle$. We obtain $|\mathcal{D}|$ transactions, and each transaction contains the segment a customer belongs to, and the category of the purchased product.

Mining Scenarios Searching for regularities in a dataset plays an essential role in data mining tasks that retrieve interesting patterns. Frequent itemset mining is the task of identifying sets of items which often occur together in the dataset. Given a frequency threshold $\varepsilon \in [1, n]$, an itemset P is said to be *frequent* in a transactions set $\mathcal{T}$ iff $support_{\mathcal{T}}(P) \geq \varepsilon$ where $support_{\mathcal{T}}(P)$ is the number of transactions in $\mathcal{T}$ that contain simultaneously all items in P. As indicated in Table 2, we set the frequency threshold to $1,000$ in both scenarios. Because marketing actions are decided and applied nation-wide, they are expected to concern at least 1,000 customers.

An itemset P is *closed* if and only if there exists no itemset $P' \supset P$ such that $support_{\mathcal{T}}(P) = support_{\mathcal{T}}(P')$ [20]. The number of closed itemsets can be orders of magnitude less important than the number of itemsets, while providing

Table 3. Interestingness measures of a rule $A \rightarrow B$. $\Diamond$, †, $\ominus$, $\otimes$ indicate measures that always produce the same rule ranking. $|\mathcal{T}|$ is the number of transactions. $P(A) = support(A)/|\mathcal{T}|$.

Measure	**Formula**
One-Way Support	$P(B\|A) \times log_2 \frac{P(AB)}{P(A)P(B)}$
Relative Risk	$P(B\|A)/P(B\|\neg A)$
Odd Multiplier $\triangleright$	$(P(AB)P(\neg B))/(P(B)P(A\neg B))$
Zhang $\triangleright$	$\frac{P(AB)-P(A)P(B)}{max(P(AB)P(\neg B),P(B)P(A\neg B))}$
Yule's Q $\Diamond$	$\frac{P(AB)P(\neg A\neg B)-P(A\neg B)P(B\neg A)}{P(AB)P(\neg A\neg B)+P(A\neg B)P(B\neg A)}$
Yule's Y $\Diamond$	$\frac{\sqrt{P(AB)P(\neg A\neg B)}-\sqrt{P(A\neg B)P(B\neg A)}}{\sqrt{P(AB)P(\neg A\neg B)}+\sqrt{P(A\neg B)P(B\neg A)}}$
Odds Ratio $\Diamond$	$(P(AB)P(\neg A\neg B))/(P(A\neg B)P(B\neg A))$
Information Gain $\ominus$	$log(P(AB)/(P(A)P(B)))$
Lift $\ominus$	$P(AB)/(P(A)P(B))$
Added Value $*$	$P(B\|A) - P(B)$
Certainty Factor $*$	$(P(B\|A) - P(B))/(1 - P(B))$
Confidence $*$	$P(B\|A)$
Laplace Correction$*$	$(support(AB) + 1)/(support(A) + 2)$
Loevinger †	$1 - P(A\neg B)/P(A)P(\neg B)$
Conviction †	$P(A)P(\neg B)/P(A\neg B)$
Example and counter-example rate$\otimes$	$1 - P(A\neg B)/P(AB)$
Sebag-Schoenauer $\otimes$	$P(AB)/P(A\neg B)$
Leverage	$P(B\|A) - P(A)P(B)$
Least Contradiction	$(P(AB) - P(A\neg B))/P(B)$
Accuracy	$P(AB) + P(\neg A\neg B)$
Gini Index	$P(A) \times (P(B\|A)^2 + P(\neg B\|A)^2) + P(\neg A) \times (P(B\|\neg A)^2 +$ $P(\neg B\|\neg A)^2) - P(B)^2 - P(\neg B)^2$
Pearson's χ^2	$\|\mathcal{T}\| \times \left(\frac{(P(AB)-P(A)P(B))^2}{P(A)P(B)} + \frac{(P(\neg AB)-P(\neg A)P(B))^2}{P(\neg A)P(B)} \right)$
	$+\|\mathcal{T}\| \times \left(\frac{(P(A\neg B)-P(A)P(\neg B))^2}{P(A)P(B)} + \frac{(P(\neg A\neg B)-P(\neg A)P(\neg B))^2}{P(\neg A)P(\neg B)} \right)$
J-measure	$P(AB)log(\frac{P(B\|A)}{P(B)}) + P(A\neg B)log(\frac{P(\neg B\|A)}{P(\neg B)})$
Φ Linear Correlation Coefficient	$(P(AB) - P(A)P(B))/\sqrt{P(A)P(B)P(\neg A)P(\neg B)}$
Two-Way Support Variation	$P(AB) \times log_2 \frac{P(AB)}{P(A)P(B)} + P(A\neg B) \times log_2 \frac{P(A\neg B)}{P(A)P(\neg B)} +$
	$P(\neg AB) \times log_2 \frac{P(\neg AB)}{P(\neg A)P(B)} + P(\neg A\neg B) \times log_2 \frac{P(\neg A\neg B)}{P(\neg A)P(\neg B)}$
Implication Index	$\sqrt{\mathcal{T}} \times \frac{P(A\neg B)-P(A)P(\neg B)}{\sqrt{P(A)P(\neg B)}}$
Klosgen	$\sqrt{P(AB)}max(P(B\|A) - P(B), P(A\|B) - P(A))$
Cosine	$P(AB)/\sqrt{P(A)P(B)}$
Jaccard	$P(AB)/(P(A) + P(B) - P(AB))$
Kappa	$\frac{P(B\|A)P(A)+P(\neg B\|\neg A)-P(A)P(B)-P(\neg A)P(\neg B)}{1-P(A)P(B)-P(\neg A)P(\neg B)}$
Piatetsky-Shapiro	$P(AB) - P(A)P(B)$
Two-Way Support	$P(AB) \times log_2 \frac{P(AB)}{P(A)P(B)}$
Specificity	$P(\neg B\|\neg A)$
Recall	$P(A\|B)$
Collective Strength	$\frac{P(AB)+P(\neg B\|\neg A)}{P(A)P(B)+P(\neg A)P(\neg B)} \times \frac{1-P(A)P(B)-P(\neg A)P(\neg B)}{1-P(AB)-P(\neg B\|\neg A)}$

Table 4. Group and description of Interestingness measures.

Measure	**Group and description**	
One-Way Support		
Relative Risk		
Odd Multiplier ▷		
Zhang ▷		
Yule's Q ◊		
Yule's Y ◊		
Odds Ratio ◊		
Information Gain ⊖	G_1	Highest confidence
Lift ⊖		Very low recall
Added Value ∗		
Certainty Factor ∗		
Confidence ∗		
Laplace Correction∗		
Loevinger †		
Conviction †		
Example and counter-example rate⊗		
Sebag-Schoenauer ⊗		
Leverage		
Least Contradiction	G_2	Very high confidence
Accuracy		Very low recall
Gini Index		
Pearson's χ^2	G_3	Average confidence
J-measure		Average recall
Φ Linear Correlation Coefficient		
Two-Way Support Variation		
Implication Index		
Klosgen		
Cosine		
Jaccard	G_4	Average confidence
Kappa		High recall
Piatetsky-Shapiro		
Two-Way Support		
Specificity		
Recall	G_5	Lowest confidence
Collective Strength		Highest recall

the same amount of information on $\mathcal{T}$. Several algorithms, including ours, focus on extracting frequent closed itemsets, increasing performance and avoiding redundancy in results [21,30].

We consider our 2 mining scenarios described in Section 2.2. Each scenario leads to the construction of a different collection of transactions $\mathcal{T}$, where a transaction is a set of items. Given $\mathcal{T}$, a frequency threshold ε, we retrieve

all closed frequent itemsets, and use them to derive association rules [28]. Each itemset P implies an association rule of the form $\mathcal{A} \rightarrow \mathcal{B}$ where $\mathcal{A}, \mathcal{B}$ is a partition of P. $\mathcal{A}$ is the antecedent of the rule, and $\mathcal{B}$ its consequent. In `prod_assoc`, $\mathcal{A}$ is a set of products ($\mathcal{A} \subseteq \mathcal{P}$) and $\mathcal{B}$ is a product. In `demo_assoc`, $\mathcal{A}$ is a customer segment and $\mathcal{B}$ is a product category. Analysts generally focus on particular products or product categories. That is why they specify the targets that they are interested in each scenario. Table 2 contains example association rules extracted from our dataset.

Interestingness Measures The ability to identify valuable rules is of utmost importance to avoid drowning analysts in useless information. Association rules $\mathcal{A} \rightarrow \mathcal{B}$ were originally selected using thresholds for *Support* ($support_{\mathcal{T}}(\mathcal{A} \cup \mathcal{B})$) and *Confidence* ($\frac{support_{\mathcal{T}}(\mathcal{A} \cup \mathcal{B})}{support_{\mathcal{T}}(\mathcal{A})}$) [1]. However, using two separate values, and guessing the right threshold is not natural. Furthermore, support and confidence do not always coincide with the interest of analysts. Hence, a number of interestingness measures that serve different analyses were proposed in the literature [4,19]. Table 3 summarizes the measures we use in this work. The first column contains the name of the measure, the second its expression. Table 4 describes the group and description of each measure and will refer to it later in the paper.

2.3 Goal

Our goal is to help analysts test and compare the rankings produced by different interestingness measures on rules extracted from $\mathcal{D}$. An analyst can specify one of 2 mining scenarios, `prod_assoc` and `demo_assoc`, and one or several targets (products in the case of `prod_assoc`, categories in the case of `demo_assoc`), and the system generates as many rule rankings as the number of interestingness measures.

3 Data Acquisition, Curation and Mining

3.1 Acquisition and storage

Each of the $3,463$ gas stations maintains a log of all customer transactions completed during one day. Whenever a customer authenticates her purchases using her loyalty card, a receipt containing the list of purchased products, their price, their category, as well as potential promotional offers, is generated. For each purchased product a record containing the receipt id, product id and customer id, is generated. These receipts are logged as $\langle r, c, p \rangle$ triples and stored in write-ahead log. Once a day, at closing time of each gas station, this log is transferred to the main data store. We have access to an SQL database containing the *sales* table where sales records are stored. Each customer is an entry in the *customers* table, which records the information she provided in her loyalty card (age, gender, region). Note that we do not have access to confidential information such name and phone number.

3.2 Data curation and preparation

We first query the *sales* table to retrieve the full raw sales records. We also query the *customers* table to retrieve for each customer the corresponding segment attributes. At the end of this step, we generate two text files. Each line in the *sales* file is a triple $\langle r, c, p\rangle$, and each line in the *customers* file is a quadruple $\langle c, age, gender, region\rangle$.

As described in Section 2.2, mining customer receipts starts with the construction of a transactions dataset $\mathcal{T}$ according to the mining scenario specified by the analyst. We rely on Apache Spark and MapReduce operations to build the dataset $\mathcal{T}$ for each mining scenario. The *sales* file is loaded as a resilient distributed dataset. We maintain a HashMap that associates to each customer her segment, and another HashMap that associates to each product its corresponding category. In the case of `prod_assoc`, the products bought by a given customer are grouped by customer identifier using a groupByKey operation. In the case of `demo_assoc`, a single *map* operation is sufficient. For each row $\langle r, c, p\rangle$ in the dataset, the map operation constructs a transaction $\langle age, gender, region, cat(p)\rangle$.

In both cases, a dataset $\mathcal{T}$ is created as a text file, with one line per transaction. In `prod_assoc`, an example of a line is *gas, car wash, cafe, sandwich* that represents all products ever purchased by a single customer. In `demo_assoc`, an example of a line is $\langle > 65, M,$ *Ile-de-France*, *Soft drinks* $\rangle$. Given a dataset $\mathcal{T}$, we can now perform the mining process.

3.3 Mining

Extracting itemsets using jLCM Generating association rules, presented in Section 2.2, requires to first extract frequent itemsets from $\mathcal{T}$. We use jLCM [16], our open-source parallel and distributed pattern mining algorithm that runs on MapReduce [13]. Mining frequent itemsets is done in two steps. We scan the input dataset $\mathcal{T}$ once and build a filtered dataset limited to transactions containing the target $\mathcal{B}$ specified by the analyst: $\mathcal{T}_{\mathcal{B}} = \{\mathcal{E} \in \mathcal{T}, \mathcal{B} \in \mathcal{E}\}$. Then, we execute jLCM on the filtered dataset. jLCM is a recursive algorithm that retrieves frequent itemsets and computes their frequency. Closed itemsets are returned along with their corresponding support, except for singletons that cannot be used to produce association rules. This extraction allows us to quickly obtain itemsets that satisfy our constraint., i.e, all extracted itemsets contain the specified target $\mathcal{B}$.

Mining rules Our analysts aim at uncovering interesting association rules expressed as $\mathcal{A} \rightarrow \mathcal{B}$ where $\mathcal{B}$ is the specified target. Evaluating the interestingness of an association rule requires computing the support of itemsets $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{A} \cup \mathcal{B}$ in $\mathcal{T}$. The standard method for mining association rules consists in finding all frequent itemsets in the dataset, and then generating the rules. Given that our analyst specifies a single target $\mathcal{B}$ at a time, this approach would be wasteful. This motivates using jLCM on the filtered dataset limited to transactions containing the target $\mathcal{B}$. The result of the itemsets extraction using jLCM contains the support of $\mathcal{B}$ and $\mathcal{A} \cup \mathcal{B}$ for all association rules we are interested in. At this

point, we need to calculate the support of each antecedent itemset $\mathcal{A}$. Thus, in a post-processing step, we scan the dataset $\mathcal{T}$ once and compute the support of all antecedents $\mathcal{A}$. This two-step approach avoids the computation of many itemsets that will never appear as a rule antecedent.

Evaluating relevant rules To evaluate the interestingness of an association rule $\mathcal{A} \rightarrow \mathcal{B}$, we only need to compute $P(\mathcal{A})$, $P(\mathcal{B})$ and $P(\mathcal{A} \cup \mathcal{B})$ because given the number of all transactions $|\mathcal{T}|$, other probabilities such as $P(\mathcal{B}|\mathcal{A})$ and $P(\mathcal{A}\neg\mathcal{B})$ can be derived. Therefore, we denormalize the results of the mining phase to store those three probabilities with each $\mathcal{A}$ and $\mathcal{B}$. The support of all rules' antecedents (used to compute $P(\mathcal{A})$) are added to the results of the mining phase (used to compute $P(\mathcal{B})$ and $P(\mathcal{A} \cup \mathcal{B})$). We create a dataframe where each row represents an association rule and has enough information to compute its interestingness. For instance, in the case of `prod_assoc`, the system computes three values for each rule. As an example, for a rule like *Coffee* $\rightarrow$ *Water*, it computes 3 values: *Support* (number of customers who purchased both *Coffee* and *Water*), *Confidence* (fraction of *Coffee* buyers who also bought *Water*) and *Recall* (fraction of *Water* buyers who also bought *Coffee*). This dataframe is augmented with 35 columns, one for each implemented measure listed in Table 3.

4 Ranking and summarization

Our goal, stated in Section 2.3, is to assist analysts in selecting the most actionable rules, those that can be used to promote products or target specific customers. In this section, we present an empirical evaluation of the 35 measures for association rules introduced in Section 2.2. The main goal of our evaluation is to compare the rankings of association rules produced by those measures on our dataset, and study their similarities. This lets us summarize ranking measures into similar clusters. We explain obtained clusters in Sections 4.2 and 4.3 and discuss their differences. This empirical evaluation automatically reduces the number of candidate measures to present to analysts in the user study.

4.1 Ranking similarity measures

We rely on the methods used in [12] to compare ranked lists of rules produced by different interestingness measures. The first three methods are taken from the literature. The last one *NDCC* is a parameter-free measure defined in [12] to emphasize differences at the top of the rankings.

We are given a set of association rules $\mathcal{R}$ to rank. Each measure, m, is seen as a function that receives a rule and generates a score, $m : \mathcal{R} \rightarrow \mathbb{R}$. We use $L^m_{\mathcal{R}}$ to denote an ordered list composed of rules in $\mathcal{R}$, sorted by decreasing score. Thus, $L^m_{\mathcal{R}} =< r_1, r_2, \ldots >$ s.t. $\forall i > i'\ m(r_i) < m(r_{i'})$. We generate multiple lists, one for each measure m, from the same set $\mathcal{R}$. $L^m_{\mathcal{R}}$ denotes a ranked list of association rules according to measure m where the rank of rule r is given as $rank(r, L^m_{\mathcal{R}}) = |\{r'|r' \in \mathcal{R},\ m(r') \geq m(r)\}|$. To assess dissimilarity between two

measures, m and m', we compute dissimilarity between their ranked lists, $L^m_{\mathcal{R}}$ and $L^{m'}_{\mathcal{R}}$. We use r^m as a shorthand notation for $rank(r, L^m_{\mathcal{R}})$.

Spearman's rank correlation coefficient Given two ranked lists $L^m_{\mathcal{R}}$ and $L^{m'}_{\mathcal{R}}$, *Spearman's rank correlation* [3] computes a linear correlation coefficient that varies between 1 (identical lists) and -1 (opposite rankings) as shown below.

$$Spearman(L^m_{\mathcal{R}}, L^{m'}_{\mathcal{R}}) = 1 - \frac{6 \sum\limits_{r \in \mathcal{R}} (r^m - r^{m'})^2}{|\mathcal{R}|(|\mathcal{R}|^2 - 1)}$$

This coefficient depends only on the difference in ranks of the element (rule) in the two lists, and not on the ranks themselves. Hence, the penalization is the same for differences occurring at the beginning or at the end of the lists.

Kendall's τ rank correlation coefficient *Kendall's τ rank correlation coefficient* [10] is based on the idea of agreement among element (rule) pairs. A rule pair is said to be *concordant* if their order is the same in $L^m_{\mathcal{R}}$ and $L^{m'}_{\mathcal{R}}$, and *discordant* otherwise. τ computes the difference between the number of concordant and discordant pairs and divides by the total number of pairs as shown below.

$$\tau(L^m_{\mathcal{R}}, L^{m'}_{\mathcal{R}}) = \frac{|C| - |D|}{\frac{1}{2}|\mathcal{R}|(|\mathcal{R}| - 1)}$$

$$C = \{(r_i, r_j) | r_i, r_j \in \mathcal{R} \wedge i < j \wedge \\ \text{sgn}(r^m_i - r^m_j) = \text{sgn}(r^{m'}_i - r^{m'}_j)\}$$

$$D = \{(r_i, r_j) | r_i, r_j \in \mathcal{R} \wedge i < j \wedge \\ \text{sgn}(r^m_i - r^m_j) \neq \text{sgn}(r^{m'}_i - r^{m'}_j)\}$$

Similar to *Spearman's*, τ varies between 1 and -1, and penalizes uniformly across all positions.

Overlap@k Overlap@k is another method for ranked lists comparison widely used in Information Retrieval. It is based on the premise that in long ranked lists, the analyst is only expected to look at the top few results that are highly ranked. While *Spearman* and τ account for all elements uniformly, Overlap@k compares two rankings by computing the overlap between their top-k elements only.

$$Overlap@k(L^m_{\mathcal{R}}, L^{m'}_{\mathcal{R}}) = \frac{|\{r \in \mathcal{R} \mid r^m \leq k \wedge r^{m'} \leq k\}|}{k}$$

Table 5. Example rankings and correlations

Ranking	Content
L^1	r_1, r_2, r_3, r_4
L^2	r_2, r_1, r_3, r_4
L^3	r_1, r_2, r_4, r_3
L^4	r_2, r_3, r_1, r_4

	Spearman	τ	*Overlap*@2	*NDCC*
L^2	0.80	0.67	1	0.20
L^3	0.80	0.67	1	0.97
L^4	0.40	0.33	0.5	−0.18

Normalized Discounted Correlation Coefficient Overlap@k, *Spearman*'s and τ sit at two different extremes. The former is conservative in that it takes into consideration only the top k elements of the list and the latter two take too liberal an approach by penalizing all parts of the lists uniformly. In practice, we aim for a good tradeoff between these extremes.

To bridge this gap, we use *NDCC* (*Normalized Discounted Correlation Coefficient*), a ranking correlation measure proposed in [12]. *NDCC* draws inspiration from *NDCG, Normalized Discounted Cumulative Gain* [9], a ranking measure commonly used in Information Retrieval. The core idea in *NDCG* is to reward a ranked list $L^m_{\mathcal{R}}$ for placing an element r of relevance rel_r by $\frac{rel_r}{\log r^m}$.

The logarithmic part acts as a smoothing discount rate representing the fact that as the rank increases, the analyst is less likely to observe r. In our setting, there is no ground truth to properly assess rel_r. Instead, we use the ranking assigned by m' as a relevance measure for r, with an identical logarithmic discount. When summing over all of $\mathcal{R}$, we obtain *DCC*, which presents the advantage of being a symmetric correlation measure between two rankings $L^m_{\mathcal{R}}$ and $L^{m'}_{\mathcal{R}}$.

$$DCC(L^m_{\mathcal{R}}, L^{m'}_{\mathcal{R}}) = \sum_{r \in \mathcal{R}} \frac{1}{\log\left(1 + r^{m'}\right) \log\left(1 + r^{m}\right)}$$

We compute *NDCC* by normalizing *DCC* between 1 (identical rankings) and −1 (reversed rankings).

$$NDCC(L^m_{\mathcal{R}}, L^{m'}_{\mathcal{R}}) = \frac{dcc - avg}{max - avg}$$

$$\begin{aligned} \text{where } dcc &= DCC(L^m_{\mathcal{R}}, L^{m'}_{\mathcal{R}}),\ max = DCC(L^{m'}_{\mathcal{R}}, L^{m'}_{\mathcal{R}}) \\ min &= DCC(L*, L^{m'}_{\mathcal{R}}),\ L* = rev(L^{m'}_{\mathcal{R}}) \\ avg &= (max + min)/2 \end{aligned}$$

Rankings comparison by example We illustrate similarities between all ranking correlation measures with an example in Table 5. This shows correlation

of a ranking L^1 with 3 others, according to each measure. *NDCC* does indeed penalize differences at higher ranks, and is more tolerant at lower ranks.

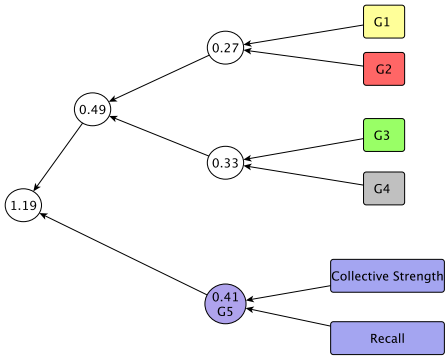
We perform a comparative analysis of the 35 interestingness measures applied to our two mining scenarios summarized in Table 2. We report the results of this comparison for `prod_assoc` in Section 4.2 and for `demo_assoc` in Section 4.3. Overall we identify 5 clusters of similar interestingness measures with some differences between the two scenarios. This confirms the need for a data-driven clustering of interestingness measures in each scenario.

4.2 Rankings comparison for prod_assoc

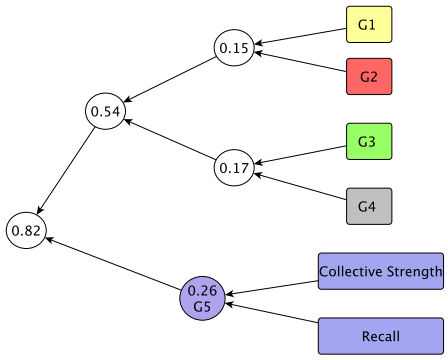
For `prod_assoc`, we generate a set of association rules $\mathcal{A} \rightarrow \mathcal{B}$, where $\mathcal{B}$ is a single product among a set of 228 representative products that were selected by our analysts. For each product $\mathcal{B}$, analysts seek to make one of two decisions: *which products $\mathcal{A}$ to bundle $\mathcal{B}$ with in an offer, and who to target for product $\mathcal{B}$ (customers who purchase products in $\mathcal{A}$).* Overall we obtain $253,334$ association rules. We compute one rule ranking per target product and per interestingness measure.

While all measures are computed differently, we notice that some of them always produce the same ranking of association rules. We identify them in Table 3 using special symbols. For example, it is easy to see that *Information gain* $= log_2(Lift)$. *Information gain* is a monotonically increasing transformation of *Lift*, so they are returning exactly the same rankings. It is also easy to see that $Loevinger = 1 - \frac{1}{Conviction}$. Thus the higher the rank of any association rule r according to *Conviction*, the higher its rank according to *Loevinger*, which leads to the exactly same rule rankings for these two measures. In addition, some of the measures that always return the same rule rankings can be easily explained analytically. Since our analyst specifies a single target product at a time, for a given ranking $P(\mathcal{B})$ is constant, which eliminates some of the differences between the considered interestingness measures. We provide on Section 4.5 a discussion about the existing relationships between all the studied measures.

Comparative analysis We now evaluate the correlation between interestingness measures that do not return the same rankings. We compute a correlation matrix of all rankings according to each correlation measure described in Section 4.1, and average them over the 228 target products that were chosen by analysts. This gives us a ranking correlation between all pairs of measures. The correlation matrix is then transformed to a distance matrix $\mathcal{M}$, i.e., the higher the correlation, the smaller the distance. Given the distance matrix $\mathcal{M}$, we can proceed to cluster interestingness measures. We choose to use hierarchical agglomerative clustering with average linkage [27]. Indeed, one of the advantages of hierarchical clustering is that it produces a complete sequence of nested clusterings, by starting with each measure in its own cluster and successively merging the two closest clusters into a new cluster until a single cluster containing all of the measures is obtained. For our hierarchical clustering implementation, we rely



(a) Kendall τ



(b) *NDCC*

Fig. 4. Summarization of interestingness measures through hierarchical clustering for `prod_assoc` (clusters are described in Table 3)

on the *cluster.hierarchy* function available from the *scipy* statistics package of Python. We obtain a dendrogram of interestingness measures and analyze their similarities. The dendrograms for $NDCC$ and τ are presented in Figure 4.

Figure 5 shows the complete dendrogram for all interestingness measures using hierarchical clustering. To describe the results more easily, we partition the interestingness measures into 5 clusters, as indicated in the third column in Table 3. G_1 is by far the largest cluster and contains 18 measures (among which *Lift, Confidence, Added value*) that produce very similar rankings, among them 6 clusters of measures always generate the same rankings. A second cluster G_2 comprising 3 measures (*Accuracy, Gini index, Least contradiction*) is similar to G_1 according to τ. But this similarity between G_1 and G_2 is higher according to $NDCC$, which shows that it is mostly caused by high ranks. A third cluster G_3 containing 7 measures (among which *J-measure*) emerges, as well as a fourth cluster G_4 containing 5 measures (among which *Piatetsky-Shapiro*), which is

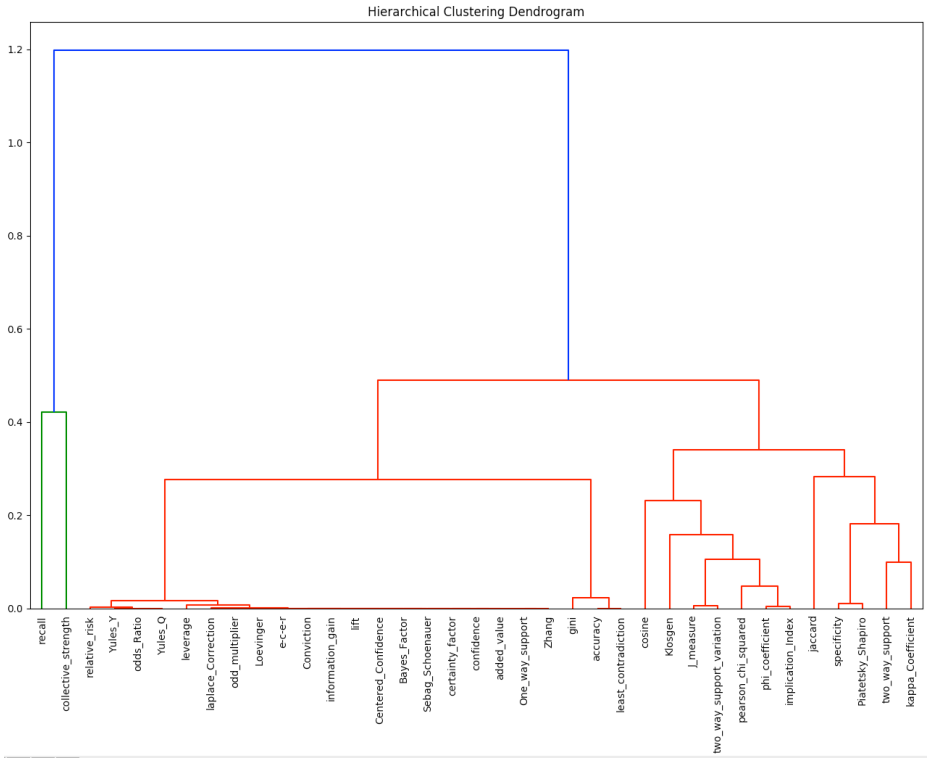


Fig. 5. Complete dendrogram for our hierarchical clustering

very similar to G_3 according to $NDCC$ as their average distance is 0.17. Finally, we have a fifth cluster G_5 containing only two measures: *Recall* and *Collective strength*.

Interestingly, we observe from the dendrograms in Figure 4 that according to $NDCC$, G_1 and G_2 are very similar. The same is true for G_3 and G_4. This difference between ranking measures illustrates the importance of accounting for rank positions. When the top of the ranked association rules is considered more important, some similarities between clusters emerge. We illustrate this behavior in Figure 6 by displaying the average rank difference between $Confidence(G_1)$ and both $Accuracy(G_2)$ and $Gini(G_2)$. This experiment clearly shows that when focusing on the top-20 ($Overlap@20$) rules the average rank difference between *Confidence* and both $Accuracy(G_2)$ and $Gini(G_2)$ is small. The same situation occurs between rankings obtained by G_3 and $G4$. This explains the differences that emerge in clustering interestingness measures when using $NDCC/Overlap$ and $\tau/Spearman$.

Explaining clusters While using hierarchical clustering on interestingness measures allows the discovery of clusters of similar measures, it does not fully ex-

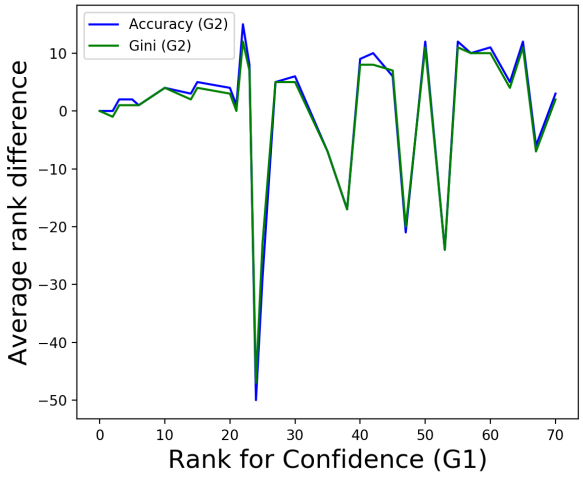


Fig. 6. Rank correlations

plain which types of results are favored by each of them. We propose to compare the output clusters according to the two most basic and intuitive interestingness measures used in data mining: *Confidence* and *Recall*. *Confidence* represents how often the consequent is present when the antecedent is, that is, $P(\mathcal{B}|\mathcal{A})$. Its counterpart, *Recall* represents the proportion of target items that can be retrieved by a rule, that is, $P(\mathcal{A}|\mathcal{B})$.

We present in Figure 7, the average *Confidence* and *Recall* values obtained on the top-20 rules ranked according to each interestingness measure. The cluster G_1 containing *Confidence* scores the highest on this dimension, but achieves a really low *Recall*. G_2 is extremely close to G_1, but achieves a slightly lower *Confidence* and *Recall*. After that, we have in order of increasing *Recall* and decreasing *Confidence* G_3 and G_4. Finally, G_5 which contains *Recall* achieves the highest value on this dimension while having the smallest *Confidence*. Figure 7 also shows that executing a Euclidean distance-based clustering, such as $k - means$, with the $Recall/Confidence$ coordinates leads to similar results as with hierarchical clustering. These results are summarized in Table 4.

4.3 Rankings comparison for demo_assoc

For `demo_assoc`, we adopt exactly the same protocol as for `prod_assoc`. We generate a set of association rules $\mathcal{A} \rightarrow \mathcal{B}$, where $\mathcal{B}$ is a product category among a set of 16 representative categories that were selected by our analysts. For each product category $\mathcal{B}$, analysts seek to answer the following question: *which customer segments $\mathcal{A}$ to target with products in category $\mathcal{B}$*. Overall we obtain

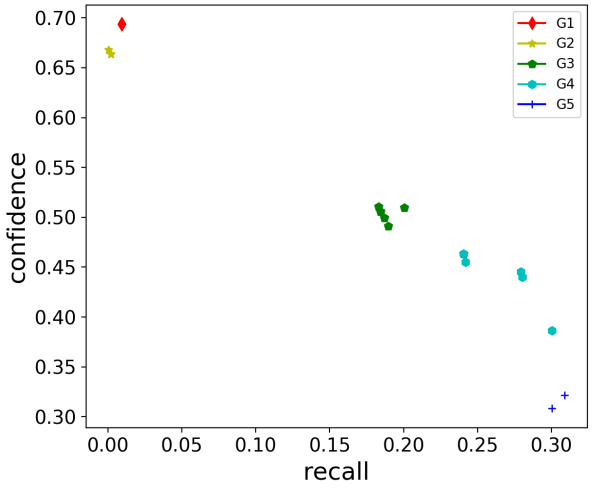


Fig. 7. Average recall/confidence of the top-20 results of interestingness measures

7,616 association rules. We compute one rule ranking per combination of target category and interestingness measure.

Similarly to `prod_assoc`, our summarization results in 5 clusters (we omit the figure due to space limitations). The first two clusters G_1 and G_2 remain unchanged. A third cluster G'_3 contains 7 measures (including *Klosgen* and *Implication index*) is very similar to G_1 according to $NDCC$ (due to accounting for high ranks). We obtain a fourth cluster G'_4 containing 3 measures (*Pearson's* χ^2, *J-measure* and *Two-way support variation*) and a fifth cluster G'_5 containing 4 measures (*Recall, Collective strength, Cosine, Jaccard*). Our hypothesis is that the observed difference between clusterings obtained for `demo_assoc` and `prod_assoc` is mainly due to high values of $P(\mathcal{A})$ in `demo_assoc` unlike the `prod_assoc` scenario.

4.4 Running time and memory consumption

Our development environment is comprised of Python 3.7.0 that invokes *j*LCM (implemented in JDK 7), for each target product or category on a 2.7 GHz Intel Core i7 machine with a 16 GB main memory, running OS X 10.13.6. Table 6) presents the average running time as well as the memory consumption of `prod_assoc` over 228 target products and that of `demo_assoc` over 16 categories. We note that `demo_assoc` runs slower than `prod_assoc`. This is mainly due to the difference in cardinalities of the constructed transactional datasets: 35,377,345 transactions in `demo_assoc` and 1,083,901 transactions in `prod_assoc`. We notice a similar trend regarding memory consumption.

Table 6. Average run time and memory consumption for mining association rules for a target product/category

Mining scenario	Average run time	Average memory usage
`prod_assoc`	12,72 seconds	111,17 Mo
`demo_assoc`	40,94 seconds	306.83 Mo

4.5 Rules that produce the same rankings

We report in this section measures that produce exactly the same rankings. Recall that we are given a set of association rules $\mathcal{R}= \{r_1, r_2, ..., r_n\}$ to rank. Given two measures m_1 and m_2 and their corresponding ranked lists of association rules $L_{\mathcal{R}}^{m_1}$ and $L_{\mathcal{R}}^{m_2}$, m_1 and m_2 produce exactly the same ranking iff $L_{\mathcal{R}}^{m_1} = L_{\mathcal{R}}^{m_2}$. More formally, for any two rules r_i and $r_j \in \mathcal{R}$, if m_1 ranks r_i before r_j then m_2 also ranks r_i before r_j, i.e., in order to prove that two measures m_1 and m_2 always produce the exact same ranking, one have to prove that:

$$\forall r_i, r_j \in \mathcal{R} : r_i^{m_1} \leq r_j^{m_1} \Longleftrightarrow r_i^{m_2} \leq r_j^{m_2}$$

where r^m is the rank of the rule r according to measure m. These theoretical dependencies between interestingness measures are studied in both *C. Tew et al.* [29] and *Dhouha* [5]. Here, we summarize the group of measures that theoretically produce indistinguishable rankings and give the existing relationship between measures. We do not provide the details of the proofs and kindly refer the reader to *C. Tew et al.* [29] and *Dhouha* [5] for the detailed proofs.

- **{Yule's Q , Yule's Y and Odds Ratio}**:

$$Yule's\, Q = \frac{Odds\ Ratio - 1}{odds\ Ratio + 1}$$

and

$$Yule's\, Q = \frac{\sqrt{Odds\ Ratio} - 1}{\sqrt{odds\ Ratio} + 1}$$

- **{ Lift, Information gain }**:
  It is easy to see that:

$$Information\ gain = log_2(lift)$$

  Information Gain is a monotonically increasing transformation of Lift, so they are returning exactly the same rankings.
- **{ Conviction, Loevinger }**:

$$Loevinger = 1 - \frac{1}{Conviction}$$

– **{ Example and Counter-example Rate, Sebag-Schoenauer }**:

$$ECR = 1 - \frac{1}{Sebag\ Schoenauer}$$

– **{Odd Multiplier, Zhang }**:

$$Zhang = \frac{Odd\ Multiplier - 1}{max(1, Odd\ Multiplier)}$$

In addition to these relationships, some others can be found in the special case when the target B is fixed. Since our analyst specifies a single target product at a time, for all ranking measures we have $P(\mathcal{B})$ constant. This eliminates some of the differences between the considered interestingness measures. Here, we highlight the measures that give exact rankings when the target is specified.

– **{Information gain, Lift, Added Value, Certainty factor, Confidence, Laplace correction }**

When, the target B is fix some dependencies can easily be proven analytically. For example, we can easily notice that:

$$Lift(A \to B) = Confidence(A \to B)/P(B)$$

Given that $P(B)$ is constant, we have the result that Lift and Confidence give exact association rule ranking. Similar observations appear between other measures. For instance,:

$$Added\ value(A \to B) = Confidence(A \to B) - P(B)$$

$$Certainty\ factor(A \to B) = \frac{Added\ value(A \to B)}{1 - P(B)}$$

5 User study

We now report the results of a user study with domain experts at `TOTAL`. The goal of this study is to assess the ability of interestingness measures to rank association rules according to the needs of an analyst. More specifically, we would like to identify which of the interestingness measures are most preferred by our analysts.

As explained in Section 4, we identified 5 clusters of similar measures, and selected a representative measure in each cluster for the user study (their names are in bold in Table 3). Representative measures are selected as the ones that most represents each clusters of measures (i.e., with the highest average similarity).

We rely on the expertise of our industrial partner to determine, for each analysis scenario, which family produces the most actionable results. Actionable

is interpreted as the most likely to lead to relevant recommendations. This experiment involved 2 experienced analysts: one data scientist and one product manager (co-authors of this paper).

For each mining scenario, `prod_assoc` and `demo_assoc`, we sampled target products and target categories respectively. Each analyst picks a mining scenario among `prod_assoc` and `demo_assoc` for which a target product or a target category must be chosen, respectively. The analyst receives a ranked list of rules. Neither the name of the measure nor its computed values for association rules are revealed because we wanted analysts to evaluate rankings without knowing how they were produced.

For a given scenario and a target product or category, our analysts completed 20 comparative evaluations showing two rankings to be compared with the top-10 rules per ranking. In each case, analysts were asked a global question on which ranking they preferred, and also to mark actionable rules in each ranking. We also collected feedback in a free-text form.

5.1 Initial study

In our initial deployment, only a few rules were marked as actionable and most rules were deemed unsurprising regardless of their ranking. After a careful examination of the rankings and of the free-text comments, we found that most rules contained products that had been bundled together as promotional offers, and that most rule antecedents in `prod_assoc` were "polluted" with frequently purchased items.

For instance, *gas* and *plastic bags* are present in many rules and only confirm what analysts already know: that most customers purchase *gas* and *plastic bags* for their groceries. Similarly, in summer, `TOTAL` regularly runs offers for multi-purpose *wipes* and for *car washing* services. Other offers are most subtle and formulated as "2 products among": *Evian, Coca-Cola, Red Bull, Lay's Chips, Haribo, Mars, Snickers, Twix, Bounty and Granola*. It is hence unsurprising to find rules associating any two of those items.

As a result, we decided to filter out *gas* and *plastic bags* from the dataset and to remove from transactions items purchased shortly after a promotional offer (identified by their reduced total price).

5.2 Feedback on ranking measures

Our second deployment was more conclusive. In summary, we observed that rankings that favor *Confidence* are best to determine which products to promote together, and rankings that favor *Recall* are well-suited for the case where a product is given and the goal is to find who to target. These conclusions resulted from deploying comparative evaluations for 5 products for `prod_assoc` and 5 categories for `demo_assoc`.

In the case of `prod_assoc`, the most preferred cluster was G_1, and an overwhelming proportion of rules in that cluster were marked as actionable. The next

most preferred in this same scenario is G_2. Both G_1 and G_2 favor *Confidence*, i.e., $P(\mathcal{B}|\mathcal{A})$, and reflect the case where a product is given and the goal is to find which other products $\mathcal{A}$ to bundle it with in a promotion.

We summarize the feedback we received.

1. **Associations between *Coffee/Coke* and other products:** *Coffee* has a high confidence with *Chocolate bars* and other drinks (*Water*, *Energy drinks* and *Soda*). This association was deemed immediately actionable. A similar observation can be made with the association between *Coca Cola* and *Sandwiches, Drinks, Potato chips* and *Desserts*.
2. **Association between car-related products:** The product *Engine Oil* has a high confidence with the car wash service *TOTAL Wash*, *Windshield wash* and a product for engine maintenance. This association was deemed immediately actionable. A similar observation was made for *Tire Spray* and *TOTAL Wash*, *Windshield wash* and different *car wipes* products.
3. **Associations between a product in different categories:** The product *Bounty* chocolate bar has a high confidence with products in the same category (other *Chocolate bars*), but also with different *Biscuits*, *Coffee* and drinks (*Water* and *Soda*). This association was deemed large scope and immediately actionable.
4. **Associations between products in the same category:** It was observed that the product *Petit Ecolier*, a chocolate biscuit, had a high confidence with other biscuits. According to our analysts, running offers on competing products is risky from a marketing point of view.

These examples illustrate the overwhelming preference for measures favoring *Confidence* for `prod_assoc` rules, and the need for domain experts in the loop to assess the actionability of rules, beyond automatic measures.

In the case of `demo_assoc`, the most preferred cluster was G'_5, and an overwhelming proportion of rules in that cluster were marked as actionable. The next most preferred in this same scenario is G'_4. Both G'_4 and G'_5 favor *Recall*, i.e., $P(\mathcal{A}|\mathcal{B})$, and reflect the case where a product category is given and the goal is to find who to target. In the case of `demo_assoc`, who to target is directly interpreted as which customer segments to target with products in that category. We summarize the feedback we received.

1. *Ice cream* products are mostly consumed in the region around Paris, in the South of France and in stations on the highway from North to South. That is the case for all consumer segments across all ages and genders. This rule led our analysts to look more carefully into the kind of station at which *Ice cream* products are consumed (e.g., on highways or not).
2. *Hot drinks* are less attractive in the South of France.
3. *Car lubricants* are mostly purchased by seniors (regardless of gender and location).

The above examples illustrate the overwhelming preference for measures offering a high *Recall* as a ranking measures for `demo_assoc` rules, and the interest of domain experts in finding which are the best customer segments to target with products in a specific category.

6 Product Recommendation

Recommendation systems are designed to guide users in a personalized way in finding useful items among a large number of possible options. Nowadays, recommendations are deployed in a wide variety of applications, such as e-commerce, online music, movies, etc. Like many retailers, `TOTAL` expressed a need for an automatic recommendation system to increase customer satisfaction and keep them away from competitor retailers. The deployment chosen by our business partners is to first design and evaluate recommendations using the synthesized interestingness measures, and then choose the right one for an actual deployment campaign in gas stations and for running personalized promotional offers.

6.1 Recommendation through Association Rules

Recommendation systems can benefit from association rules extraction [11,26]. As shown in the experimental study by Pradel *et al.* [24], association rules have demonstrated good performance in recommendations using real-world e-commerce datasets, where explicit feedback such as ratings on the products is not available. Thus, it appears necessary to evaluate the performance of recommendations based on association rules mining on our dataset.

Association rules were first used to develop top-N recommendations by *Sarwar et al* [26]. They use *support* and *confidence* to measure the strength of a rule. First, for each customer, they build a single transaction containing all products that were ever purchased by that customer. Then, they use association rule mining to retrieve all the rules satisfying a given minimum *support* and minimum *confidence* constraints. To perform top-N recommendations for a customer u, they find all the rules that are supported by the customer purchase history (i.e., the customer has purchased all the products that are antecedent of the rule). Then, they sort products that the customer has not purchased yet based on the maximum *confidence* of the association rules that were used to predict them. The N highest ranked products are kept as the recommended list. Authors in [11] use a very similar approach but they also consider additional association rules between higher-lever categories where it is assumed that products are organized into a hierarchical structure.

These works present two main drawbacks. First, they require specifying thresholds on *support* and *confidence* which might be hard to adapt for different customers, and which results in the inability to recommend products that are not very frequent. Second, searching for rules where the whole purchase history of a customer is included in the antecedent, might lead to a very low or insufficient number of associations. Thus, for every customer who purchased a single product that fails the minimum *support* constraint the approach cannot compute recommendations. To overcome these drawbacks, we adapt the approach of Pradel *et al.* [24] using bi-gram association rules which consists in computing the *relevance* of the association rules $(l \rightarrow k)$ for every pair of products l and k. In computing relevance of a rule, we do not restrict ourselves to *confidence* and leverage the results we obtained on the synthesized measures to compare how different

interestingness measures behave in practice (i.e., in providing accurate recommendations). In fact, as we show in Table 7, for the same anonymized customer, different interestingness measures (in this case: Confidence,Least-Contradiction and Piatetsky-Shapiro) provide different top-5 product recommendations.

Table 7. Purchase history of an anonymized customer and top-5 product recommendation according to different interestingness measures. Product descriptions are kept in French.

	top-5 recommendations using different interestingness measures		
Purchase history	Confidence	Least-Contradiction	Piatetsky-Shapiro
Coca Cola 50Cl Coca Cola 1.5L Chupa chups Coca Cola 33CL Cristaline 1,5L PIM'S Framboise Lu Sandwish.XXL Jamb Kinder Bueno 43G Total Wash 15 Snickers 50g	Total Wash 25 Café 10 Coca Cola Pet 1L Fuze Peche Pet 40Cl Chips Lays 45G	Total Wash 25 Café 10 Chips Lays 45G Fuze Peche 40Cl Evian 1,5L	Café 10 Fuze Peche 40Cl Cristaline 50CL Mars Legend 51G Evian 1L

More formally, let $\mathcal{U} = \{u_1, u_2, ..., u_m\}$ be the set of all customers and $\mathcal{I} = \{i_1, i_2, ..., i_n\}$ be the set of all products. For a given customer u, $H_u \subseteq \mathcal{I}$ denotes the purchase history of u , the set of all products ever purchased by u. The training stage of the algorithms we evaluate takes as input a *purchase matrix*, where each column corresponds to a product and the customers that have purchased it, and each row represents a customer and the products she purchased.

We denote $\boldsymbol{P}$ the *purchase matrix* of the m customers in $\mathcal{U}$ over the n products in $\mathcal{I}$. An entry $p_{u,i}$ in the matrix contains a boolean value (0 or 1), where $p_{u,i} = 1$ means that product i was bought by customer u at least once (0 means the opposite).

We leverage the purchase history of our customers to extract association rules of the form $i \Rightarrow j$, which means that whenever a customer purchases the product i (antecedent), she is likely to purchase the product j (consequent). Therefore, We use *bigram* rules to compute an association matrix $\boldsymbol{A}$ between each pair of products i, j. The matrix $\boldsymbol{A}$ is computed from the purchase matrix $\boldsymbol{P}$, where each entry $a_{j,l}$ corresponds to the interestingness of the association rule $j \Rightarrow l$.

$$a_{j,l} = interestingness(j \Rightarrow l) \tag{1}$$

The training phase of the approach consists of the computation of all available bi-gram association rules, and stores the corresponding values of the strength on association rules in the association matrix $\boldsymbol{A}$ of size $n \times n$. Once, the association matrix is computed, to generate top-N recommendations for customer u, we first

identify a set of association rules that are supported by the purchase history of u. i.e., rules of the form $k \Rightarrow l$, where k is purchased by u. Then, non purchased products are ranked either by their maximum value [26], [24], or the sum of values [11] of all association rules. In our case, the max aggregation was found to give slightly better results. This could be explained by the fact that given a target product purchased by the test customer in the test set (e.g, *cleaning wipes*), if the customer purchased in the past food and drinks products frequently and car wash products (e,g. *windshield washer*) less frequently. Using the sum aggregation will result to a poor prediction for the target *cleaning wipes* even if it is highly associated with the *windshield washer* because of the poor values of associations with other food and drink products that the customer purchased.

Thus, we compute the score of a product j for a customer u as follows:

$$score(u, j) = Max_{i \in H_u} interestingness(i \Rightarrow j) \tag{2}$$

where, H_u is the purchase history of customer u, and i is a candidate product for recommendation. Products are then sorted according to their respective scores and the top-N products are recommended to u.

6.2 Experiments

Protocol In this section, we present our experimental protocol and the evaluation measures we use in our experiments. The widely used strategy for evaluating recommendation accuracy in offline settings is to split the dataset into training and test sets. The test set is used to simulate future transactions (ratings, clicks, purchases, etc) and it usually contains a fraction of transactions. The remaining interactions are kept in the training set and are fed to the recommendation algorithm to output a list of top-N product recommendations for each user. The accuracy of recommendations is then evaluated on the test set. However, this setting does not reflect well the reality in the retail context as it is time agnostic.

The availability of timestamps in the purchase records enables us to attempt a more realistic experiment. We hence train our algorithm on past purchases and test the results on future purchases. We split the dataset according to a given point in time which acts as our "present" (the time we apply our algorithm). Purchases that happened before the split point are used for training, whereas future purchases after the split point are used for testing. Customers whose purchase histories are timestamped only after the split point are discarded. For our dataset, we choose 1st January 2019 to be the split date. More specifically, we use purchase records from January 2017 to December 2018 for training and records from January 2019 to December 2019 for testing.

As it is often practiced in the recommendation literature [8,25], for our experiments we discard customers who purchased fewer than 5 products in the training set. An important aspect of our dataset and of all datasets in the retail domain is the tendency to repetitively purchase the same products at different times. It is however much more valuable for the customer and even for the retailer to recommend products that the customer has not purchased recently, or

is not aware of. In addition, we noticed that if we simply randomly select N products from the purchase history of each customer as the top-N recommendations, we can reach reasonable accuracy. Thus, after several exchanges with the marketing department at TOTAL, for each test customer we decided to remove the "easy" predictions from the test set corresponding to the products that have been purchased by that customer during the training period. We also select only customers who had more than 10 purchases after removing already purchased products in the test set. This setting makes the task of predicting the correct products harder but potentially more impactful in a real-world scenario.

Evaluation Measures A recommendation algorithm outputs a sorted list of top-N product recommendations given the purchase history of a target customer. Top-N recommendations are typically evaluated in terms of their precision, recall and F1-score [6,7]. For each customer u, precision measures the percentage of recommended products that are relevant, recall measures the percentage of relevant products that are recommended, whereas, $F1$-score is defined as the harmonic mean of precision and recall. In our setting, a product i is relevant to a customer u if u has effectively purchased i in the test set.

In our approach, we have a set of test customers with a corresponding target set of products (recall that the target set contains the customer purchases in the test data). For a given customer u, the precision, recall, $F1$-score and of the top-N recommendations are respectively defined as follows:

$$Precision_u@N = \frac{|R_u@N \cap T_u|}{N} \tag{3}$$

$$Recall_u@N = \frac{|R_u@N \cap T_u|}{|T_u|} \tag{4}$$

where given a customer u, T_u is the target test set and $R_u@N$ is the set of top-N recommendations.

$$F1_u@N = \frac{2.Precision_u@N.Recall_u@N}{Precision_u@N + Recall_u@N} \tag{5}$$

To compute the final performance values, we average all metrics over all test customers.

$$Precision@N = \frac{\sum_{u:test\ customer} Precision_u@N}{Number\ of\ test\ customers} \tag{6}$$

Similar formulas are used to obtain $Recall@N$ and $F1@N$ for all customers:

$$Recall@N = \frac{\sum_{u:test\ customer} Recall_u@N}{Number\ of\ test\ customers} \tag{7}$$

$$F1@N = \frac{\sum_{u:test\ customer} F1_u@N}{Number\ of\ test\ customers} \tag{8}$$

Results In our experiments, each row of our training purchase matrix contains all known purchases of training customers before the split date at which training and test sets are separated: January 1^{st}, 2019. All algorithms using the different selected interestingness measures are evaluated using exactly the same test customers and the corresponding target sets. The reported performance results are computed following the experimental protocol described in Section 6.2 and using the evaluation measures reported in Section 6.2.

The values of recommendation accuracy: *Precision*@10, *Recall*@10 and $F1$@10 for each interestingness measure are reported in Table 8. First, we can notice that G_1 achieves the best recommendation performance and performs slightly better than G_2. The performance results confirm our findings in the user study where our domain experts preferred group G_1 and measures that favor confidence for the `prod_assoc` scenario. Second, we notice that the achieved recommendation accuracy for groups G_1 and G_2 are very close (12.56% and 12.08% for *Precision*@10, respectively). The same occurs with very similar performances for groups G_3 and G_4 (10.95% and 10.56% for *Precision*@10, respectively). These results are consistent with the clustering that we performed using $NDCC$ (Figure 4b). Since we compute top-10 lists per customer, $NDCC$ gives more importance to associations rules in the top of the lists. This explains the similarities of recommendation performances for G_1 and G_2 as well as groups G_3 and G_4, as the average distance between G_1 and G_2 in the dendrogram in Figure 4b is 0.15 and the average distance between G_3 and G_4 is 0.17. Then, we notice a really poor performance for measures in group G_5 that are not usable in practice. This is mainly due to the fact that measures having a very low confidence favor rare targets over frequent ones for ranking association rules, which results in recommending mostly irrelevant products. We also noticed that some measures that are in the same cluster may not produce similar recommendation performance. In fact, we also produced recommendations using measures within the same cluster and found that the recommendations were different for some cases. We conjecture that this is due to the fact that the obtained rules for computing recommendations focused on different subsets of purchased products according to different users and exhibit the same phenomenon as the Simpson Paradox. For instance, using Accuracy (G_2)as a ranking measure leads to a poor performance values (0.91% for precision@10), while using *Least Contradiction* (G_2) gives much better results (12.08% for precision@10), even if both measures are in the same cluster.

Finally, we implemented a `MostPop` baseline which is the method that were used so far by our analysts and which consists of a non personalized method that recommends to each customer the set of most popular products that the customer did not purchase yet. We can see from our results that except for measures in group G_5, all others groups of measures perform better than the non personalized baseline. In particular G_1 show an improvement of 53.54% in relative performance.

Table 8. Recommendation accuracy for each representative measure

Measure	Prec@10	Recall@10	F1@10
Lift (G_1)	**12.56%**	**7.03%**	**8.60%**
Least-Contradiction (G_2)	12.08%	6.69%	8.20%
Cosine (G_3)	10.95%	6.19%	7.55%
Piatetsky-Shapiro (G_4)	10.56%	6.02%	7.32%
Collective strength (G_5)	0.45%	0.28%	0.26%
`MostPop` (baseline)	8.18%	4.76%	5.76%

7 Related work

To the best of our knowledge, this paper is the first to bring a framework for association rule mining to the marketing department of an *oil* and *gas* company, and empower domain experts with the ability to conduct large-scale studies of customer purchasing habits.

The definition of quality of association rules is a well-studied topic in statistics and data mining. In their survey [4], Geng *et al.* review 38 measures for association and classification rules. They also discuss 4 sets of properties like symmetry or monotony, and how each of them highlights different meanings of "rule quality", such as novelty and generality. However, we observe no correlation between these properties and the groups of measures discovered using our framework.

These 38 measures are compared in [14]. Authors consider the case of extracting and ranking temporal rules (*event A*→*event B*) from the execution traces of programs. Each measure is evaluated in its ability to rank highly rules known from a ground truth (library specification). We observe that the measures scoring the highest are all from the groups identified in this work as G_1 and G_2, which are also favored by our analysts. There are however some counterexamples, with measures from G_1 scoring poorly. The main difference between our work and [14] is the absence of a ground truth of interesting rules for our dataset.

A close work to ours is Herbs [15]. Herbs relies on a different and smaller set of measures to cluster rule rankings. Authors perform an analysis of the properties of measures, in addition to an experimental study. The datasets used are from the health and astronomy domains. Each of them contains at most 1,728 transactions and leads to the extraction of 49 to 6,312 rules. Rankings are then compared between all pairs of measures using Kendall's τ correlation measure averaged over all datasets. The largest group of measures identified, which includes *Confidence*, is similar to G_1.

Our use of the *p-value* (via *Pearson's* χ^2 *test*) in the evaluation of rule interestingness is borrowed from [17]. In that work, the authors propose an exploration framework where rules are grouped by consequent and traversed by progressively adding items to the antecedent. The framework provides hints incrementally to help guess how each additional item would make a difference. Such a framework

is suitable to some of the scenarios we consider and could be integrated in a future version of our work.

Other significant works on clustering interestingness measures include [5,2,29]. In these studies, 61 measures are analyzed from both a theoretical and an empirical aspect to provide insights about the properties and behavior of the measures according to association rule ranking. The number of measures studied in these works is greater than ours. However, our work goes a step further as (1) we provide a user study performed with domain experts from TOTAL marketing department, (2) we show how association rules can be used to perform top-N recommendations, and (3) we show a comparative evaluation of the synthesized interestingness measures according to accuracy measures.

An interesting research area is OLAP pattern mining, which integrates online analytical processing (OLAP) with data mining so that the mining can be performed in different portions of the database [18,23]. However, the focus of our work is not on expressivity nor is it on performance computation. An interesting research direction would indeed be to extend our framework to using the full power of OLAP.

8 Conclusion

We present our framework to enable decision support through mining, ranking, and summarization of association rules. We use large longitudinal TOTAL datasets that comprises of 30 million unique sales receipts, spanning 35 million records. In conjunction with domain expert non-scientists, we studied two scenarios: associations between a set of products and a target product, and between customer segments and product categories. Both of these scenarios led to actionable insights leading to effective decision support for the TOTAL marketers. We empirically studied 35 interestingness measures for ranking association rules and further summarize them in 5 synthesized clusters or groups. Resulting groups were then evaluated in a user study involving a data scientist and a domain expert at TOTAL. We concluded that ranking measures ensuring high confidence, best fit the needs of analysts in the case of prod assoc, and measures that ensure high recall are better in the case of demo assoc. Finally, we discussed how our findings can be used to perform product recommendation using different interestingness measures for ranking association rules.

References

1. Agrawal, R., Imieliński, T., Swami, A.: Mining Association Rules between Sets of Items in Large Databases. In: Proc. SIGMOD. pp. 207–216 (1993)
2. Belohlavek, R., Grissa, D., Guillaume, S., Nguifo, E.M., Outrata, J.: Boolean factors as a means of clustering of interestingness measures of association rules. Annals of Mathematics and Artificial Intelligence **70**(1-2), 151–184 (2014)
3. Daniel, W.: Applied Nonparametric Statistics. Houghton Mifflin (1978)
4. Geng, L., Hamilton, H.J.: Interestingness Measures for Data Mining: A Survey. ACM Comput. Surv. **38**(3) (2006)

5. Grissa, D.: Etude comportementale des mesures d'intérêt d'extraction de connaissances. Ph.D. thesis (2013)
6. Gunawardana, A., Shani, G.: Evaluating recommender systems. In: Recommender systems handbook, pp. 265–308. Springer (2015)
7. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS) **22**(1), 5–53 (2004)
8. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: 2008 Eighth IEEE International Conference on Data Mining. pp. 263–272. Ieee (2008)
9. Järvelin, K., Kekäläinen, J.: Cumulated Gain-based Evaluation of IR Techniques. ACM Trans. Inf. Syst. **20**(4), 422–446 (2002)
10. Kendall, M.G.: A New Measure of Rank Correlation. Biometrika **30**(1/2), 81–93 (1938)
11. Kim, C., Kim, J.: A recommendation algorithm using multi-level association rules. In: Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003). pp. 524–527. IEEE (2003)
12. Kirchgessner, M., Leroy, V., Amer-Yahia, S., Mishra, S.: Testing interestingness measures in practice: A large-scale analysis of buying patterns. In: 2016 IEEE International Conference on Data Science and Advanced Analytics, DSAA 2016, Montreal, QC, Canada, October 17-19, 2016. pp. 547–556. IEEE (2016). https://doi.org/10.1109/DSAA.2016.53, `https://doi.org/10.1109/DSAA.2016.53`
13. Kirchgessner, M., Leroy, V., Termier, A., Amer-Yahia, S., Rousset, M.C.: jLCM. `https://github.com/slide-lig/jlcm`, [Online; accessed 27-May-2016]
14. Le, T.D., Lo, D.: Beyond Support and Confidence: Exploring Interestingness Measures for Rule-Based Specification Mining. In: Proc. SANER. pp. 331–340 (2015)
15. Lenca, P., Vaillant, B., Meyer, P., Lallich, S.: Association Rule Interestingness Measures: Experimental and Theoretical Studies. In: Quality Measures in Data Mining, pp. 51–76. Springer (2007)
16. Leroy, V., Kirchgessner, M., Termier, A., Amer-Yahia, S.: Toppi: An efficient algorithm for item-centric mining. Inf. Syst. **64**, 104–118 (2017). https://doi.org/10.1016/j.is.2016.09.001, `https://doi.org/10.1016/j.is.2016.09.001`
17. Liu, G., Feng, M., Wang, Y., Wong, L., Ng, S.K., Mah, T.L., Lee, E.J.D.: Towards Exploratory Hypothesis Testing and Analysis. In: Proc. ICDE. pp. 745–756 (2011)
18. Messaoud, R.B., Rabaséda, S.L., Boussaid, O., Missaoui, R.: Enhanced mining of association rules from data cubes. In: DOLAP 2006, ACM 9th International Workshop on Data Warehousing and OLAP, Arlington, Virginia, USA, November 10, 2006, Proceedings. pp. 11–18 (2006). https://doi.org/10.1145/1183512.1183517, `https://doi.org/10.1145/1183512.1183517`
19. ichi Minato, S., Uno, T., Tsuda, K., Terada, A., Sese, J.: A Fast Method of Statistical Assessment for Combinatorial Hypotheses based on Frequent Itemset Enumeration. Lect. Notes Artif. Int. **8725**, 422–436 (2014)
20. Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Discovering Frequent Closed Itemsets for Association Rules. In: Proc. ICDT. pp. 398–416 (1999)
21. Pei, J., Han, J., Mao, R.: Closet: An Efficient Algorithm for Mining Frequent Closed Itemsets. In: Proc. SIGMOD. pp. 21–30 (2000)
22. Piatetsky-Shapiro, G.: Knowledge Discovery in Databases. Menlo Park, CA: AAI/MIT (1991)

23. Plantevit, M., Laurent, A., Teisseire, M.: Olap-sequential mining: Summarizing trends from historical multidimensional data using closed multidimensional sequential patterns. New Trends in Data Warehousing and Data Analysis **3**, 275 (2008)
24. Pradel, B., Sean, S., Delporte, J., Guérif, S., Rouveirol, C., Usunier, N., Fogelman-Soulié, F., Dufau-Joel, F.: A case study in a recommender system based on purchase data. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 377–385. ACM (2011)
25. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. In: Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence. pp. 452–461. AUAI Press (2009)
26. Sarwar, B., Karypis, G., Konstan, J., Riedl, J., et al.: Analysis of recommendation algorithms for e-commerce. In: EC. pp. 158–167 (2000)
27. Sokal, R.R., Michener, C.D.: A Statistical Method for Evaluating Systematic Relationships. Univ. Kans. Sci. Bull. **38**, 1409–1438 (1958)
28. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining, (First Edition). W. W. Norton & Company (2007)
29. Tew, C., Giraud-Carrier, C., Tanner, K., Burton, S.: Behavior-based clustering and analysis of interestingness measures for association rule mining. Data Mining and Knowledge Discovery **28**(4), 1004–1045 (2014)
30. Uno, T., Kiyomi, M., Arimura, H.: LCM ver. 2: Efficient Mining Algorithms for Frequent/Closed/Maximal Itemsets. In: Proc. ICDM Workshop FIMI (2004)