Recommendations Using Information from Multiple Association Rules: A Probabilistic Approach

Abhijeet Ghoshal

University of Louisville, College of Business, Harry Frazier Hall, 110 W. Brandeis Ave.

Louisville, KY 40292Syam Menon, Sumit Sarkar

Naveen Jindal School of Management, University of Texas at Dallas, 800 W. Campbell Road, Richardson TX, 75080

Abstract

Business analytics has evolved from being a novelty used by a select few, to an accepted facet of conducting business. Recommender systems form a critical component of the business analytics toolkit and, by enabling firms to effectively target customers with products and services, are helping alter the e-commerce landscape. A variety of methods exist for providing recommendations, with collaborative filtering, matrix factorization, and association rule based methods being the most common. In this paper, we propose a method to improve the quality of recommendations made using association rules. This is accomplished by combining rules when possible, and stands apart from existing rule-combination methods in that it is strongly grounded in probability theory. Combining rules requires the identification of the best combination of rules from the many combinations that might exist, and we use a maximum-likelihood framework to compare alternative combinations. As it is impractical to apply the maximum likelihood framework directly in real time, we show that this problem can equivalently be represented as a set partitioning problem by translating it into an information theoretic context – the best solution corresponds to the set of rules that leads to the highest sum of mutual information associated with the rules. Through a variety of experiments that evaluate the quality of recommendations made using the proposed approach, we show that (i) a greedy heuristic used to solve the maximum likelihood estimation problem is very effective, providing results comparable to those from using the optimal set partitioning solution, (ii) the recommendations made by our approach are more accurate than those made by a variety of state-of-the-art benchmarks, including collaborative filtering and matrix factorization, and (iii) the recommendations can be made in a fraction of a second on a desktop computer, making it practical to use in real-world applications.

Keywords: Personalization, Bayesian Estimation, Maximum Likelihood, Information Theory

1. Introduction

A recent International Data Corporation (IDC) report estimates that the business analytics market will grow at a compounded rate of 9.8%, and reach 50.7 billion by 2016. This is partly fueled by the growth in the amount of customer data readily available to firms, and the potential for businesses to leverage their data through the novel use of software-based analytic techniques. Recommender systems form an integral part of the business analytics toolkit, and several studies have shown that personalized recommendations can enable firms to effectively target customers with products and services (Häubl and Trifts 2000, Tam and Ho 2003). For example, Pathak et al. (2010) find that the strength of a recommender system has a positive effect on sales and on prices.

Recommendations are made on a continuous basis, and can have a substantial impact on the bottom line. According to Hosanagar et al. (2014), 60% of Netflix rentals stem from recommendations, while 35% of Amazon's sales originate from their recommendation system. It is easy to see that even a small improvement in the quality of recommendations would be worth millions of dollars every year to a retailer.

A variety of methods exist for providing recommendations, with collaborative filtering, matrix factorization, and association rule based methods being the most common.[1] In this paper, we focus on improving the quality of rule based recommendations by combining information from multiple association rules. Rule-based approaches comprise one prominent class of techniques used to provide personalized recommendations to customers. Firms such as *BroadVision* provide rule-based tools to firms that wish to implement recommendation systems on their web sites (Hanson 2000). Rules are easy to understand, which appeals to marketers interested in cross-selling or product placement. Often, such rule based systems use association rules (Hastie et al. 2009).

Association rules are implications of the form {bread, milk}$\rightarrow${yogurt}, where {bread, milk} is called the antecedent of the rule and {yogurt} is called its consequent. While millions of such implications are possible in a typical dataset, not all of them are useful for providing recommendations. Agrawal et al. (1993) provided a method to identify those rules where the items in the rules appear in a reasonably large numbers of transactions (termed the *support* of the rule), and where a consequent has a high probability of being chosen when the items in the antecedent have already been chosen (termed the *confidence* of the rule). Every mined rule must meet minimum thresholds for both support and confidence.

Association rules compactly express how products group together (Berry and Linoff 2004), and have been successfully used for market basket analysis (Gordon 2008, Lewin 2009). Recommendation systems based on association rules leverage available rules and a customer's basket, to recommend items as the customer is shopping. Many firms implement association rule based recommender systems as they can be used unobtrusively in automated systems to provide recommendations to customers in real time. For instance, Forsblom et al. (2009) develop a mobile application for a Nokia smartphone that uses association rules to recommend retail products to customers. Prominent companies like IBM promote association rule mining capabilities in their business analytics software (IBM 2009a, 2009b, 2010). Moreover, because an association rules based system compares alternative items to recommend based on their probabilities of

[1] Some researchers consider matrix factorization and rule based systems to be types of collaborative filtering techniques. For expositional convenience we refer to them as distinct from collaborative filtering.

purchase, the system can be easily adapted to make recommendations based on expected payoffs associated with the items[2].

While there has been considerable work done on mining rules more efficiently (e.g., Ng. et al. 1998, Bayardo 1998, Bayardo and Agrawal 1999, Zaki 2000, Webb 2000, Webb and Zhang 2005, Webb 2008, Webb 2010, Calders et al. 2013, Zhou et al. 2013), research into the use of rules to make effective recommendations is scarce. Zaïane (2002) proposed a method that finds all *eligible* rules (rules whose antecedents are subsets of the basket and whose consequents are not), and recommends the consequent of the eligible rule with the highest confidence. Baralis and Garza (2002) and Baralis et al. (2004) propose related approaches (referred to as L3 and L3G, respectively) for classification based on the selective pruning and elimination of "harmful" rules; these can be adapted for recommending items as well. Wang and Shao (2004) suggest considering only maximal rules, i.e., eligible rules whose antecedents are maximal-matching[3] subsets of the basket. All these approaches focus on identifying a single rule to make the recommendation. Often however, the antecedent of the selected rule will not contain all the items in the basket. Consequently, the recommendation is made on the basis of partial information – items not present in the antecedent of the rule being used for recommendation are effectively ignored. It is not difficult to see that the item being recommended could be different had the recommendation system been able to use information from *all* the items in the basket. The set of eligible rules often contains multiple rules with the same consequent, and the quality of recommendations could potentially be improved by combining such rules effectively.

The notion of combining rules has been explored in a few studies in the past. Given a customer's basket, Lin et al. (2002) calculate the score for each item as the sum of the products of the supports and confidences of all eligible rules with this item as the consequent. The item with highest score is recommended to the customer. Wickramaratna et al. (2009) present an approach to identify rules that predict the presence and absence of an item, and propose a Dempster-Shaffer based approach for combining rules when some rules predict that a customer will purchase an item, while other rules predict the contrary. However they note that their approach is not scalable for real-time applications.

There has also been some work that attempts to combine classification rules. Li et al. (2001) suggest classifying customers using Classification based on Multiple Association Rules (CMAR). They group eligible rules with the same consequent (class), and evaluate the sum of weighted chi-squares of the rules

[2] The system can obtain the expected payoff associated with an item by computing the product of the conditional probability of the item being selected by the customer (i.e., the confidence) and the item's profit margin. The firm can then recommend the product with the highest expected margin.

[3] An antecedent is *maximal matching* if no supersets of it, present as antecedents of other rules, are subsets of the basket.

in each group. The customer is assigned to the consequent class corresponding to the group with the highest sum. Liu et al. (2003) classify customers using a score calculated based on the combination of all the eligible rules (determined based on the attributes of the customer). Their scoring formula requires the identification of rules with negations and cannot be adapted for selecting items to recommend in a traditional association rule mining context. Thabtah (2007) provides a detailed survey on various classification approaches that use single and multiple association rules for classification.

Two other techniques that have been successfully employed in recommendation systems are collaborative filtering and matrix factorization. Collaborative filtering based methods are perhaps the best known, at least since Amazon.com decided to deploy it as part of their recommender system (Linden et al. 2003). These methods use the known preferences of a group of users to make recommendations or predictions of the unknown preferences for other users. Matrix factorization methods gained recognition partly as a result of successes in the Netflix Prize competition. These methods represent users and items through factors identified from the data, and an item is recommended to a user when the item and user are similar vis-à-vis these factors (Koren et al. 2009). Su and Khoshgoftaar (2009) provide a detailed survey of several collaborative filtering and matrix factorization based approaches.

It is evident from the above discussion that there exists a considerable amount of literature on recommendation techniques. However, the literature lacks a principled approach to combine information from multiple rules. This paper makes multiple contributions in this regard.

(i) A common characteristic of the existing works that attempt to combine rules is that all the methods proposed are ad hoc in nature. In contrast, we propose a method formally grounded in probability theory to combine multiple rules and make recommendations based on as many items in the basket as possible. As with other approaches that use association rules, we assume that the rules have already been mined (using any of the methods that have been proposed for mining rules) and available for use. Given a customer's basket, we estimate the probability of each item (that can be recommended) being selected by the customer.

(ii) We view the collection of rules being combined as a probability model. When multiple potential rule combinations exist for a particular target item, the best combination of rules (i.e., the best probability model) needs to be identified. We develop a maximum likelihood approach to determine the best model; the problem is framed as one of maximizing the likelihood that the observed data (the training data used to mine the rules) is generated from the competing probability models represented by the feasible rule combinations. The problem of maximizing likelihood requires us to estimate the likelihoods from the training data at the time recommendations are made. However, it is not feasible to do this in real time as many probability parameters need to be estimated for each probability model, and the training data sets will be large in most practical cases.

We show that this problem can be transformed into an equivalent problem of maximizing the mutual information (MI) associated with each model, where the mutual information of the model is the sum of the mutual information values associated with the rules included in the model. The mutual information values associated with the mined rules can be pre-computed. These values enable the efficient comparison of alternative probability models in real time.

(iii) When the number of items in a customer's basket increases, the number of feasible rule combinations can grow rapidly. Therefore, the number of probability models to compare could be large for some problem instances. We develop a greedy heuristic that determines good solutions in real time regardless of the size of the basket. Experiments comparing the performance of an optimal approach with that of the heuristic are conducted on three real datasets. The performance of the heuristic is virtually identical to that of the optimal for experiments conducted on one dataset, and only marginally worse for experiments conducted on the other two (the differences are not statistically significant).

(iv) The effectiveness of our methodology – termed *Maximum Likelihood Recommendations* (MLR) – is demonstrated through a variety of additional computational experiments that compare it to many key benchmarks. We compare the accuracy of recommendations made by MLR to those made by (a) the single rule approaches of Zaïane (2002), Wang and Shao (2004) and Baralis et al. 2004 (L3G), (b) the rule combination approaches of Li et al. (2001) and Lin et al. (2002), (c) item-based collaborative filtering, and (d) matrix factorization (FunkSVD). MLR is shown to outperform all the benchmarks, and the performance improvements are observed to be robust at various support and confidence thresholds used for mining rules in all three datasets. When it is feasible to combine multiple rules so that a larger proportion of the basket is covered by the rule antecedents than would be possible otherwise, MLR performs particularly well.

We describe the problem in detail in Section 2, while the methodology is discussed in Section 3. Section 4 presents results of the experiments conducted to validate our approach for rule-based recommendation environments. Section 5 compares MLR with the collaborative filtering and matrix factorization approaches. Section 6 concludes the paper.

2. Problem Description

The problem being considered in this paper is best illustrated through an example. Consider a customer who has three items i_1, i_2, and i_3 in her basket $\boldsymbol{B}$, i.e., $\boldsymbol{B} = \{i_1, i_2, i_3\}$. The eligible rules for this basket – i.e., all available rules whose antecedents are subsets of the basket – are listed in Table 1. Rules R_1 – R_4 have

item x_1 as their consequent, while item x_2 is the consequent of rules R_5 – R_8. Our task is to select one of x_1 or x_2 and recommend it to the customer.

Traditional rule based approaches identify the best rule from the eligible set, and recommends the associated consequent. So, for example, Zaïane's (2002) approach would rank the eligible rules based on their confidences, and select the consequent of the rule with the highest confidence as the item to recommend. The rule with the highest confidence in our example is R_8, and consequently, x_2 would be recommended to the customer. However, recommending x_2 based on R_8 ignores some items in the basket (i_1 and i_3), despite the fact that another rule – R_5 – exists with these items in the antecedent. This is true in general – making a recommendation based on a single rule often disregards items in the basket that are not in the antecedent of the rule being used.

Rule	Antecedent	Consequent	Confidence
R_1	i_1,i_2	x_1	60%
R_2	i_2,i_3	x_1	40%
R_3	i_1	x_1	53%
R_4	i_3	x_1	43%
R_5	i_1,i_3	x_2	42%
R_6	i_2,i_3	x_2	50%
R_7	i_1	x_2	58%
R_8	i_2	x_2	62%

Table 1: Eligible rules for basket $\boldsymbol{B} = \{i_1,i_2,i_3\}$

If we could effectively combine rules and cover as many items of the basket as possible, our recommendation would be more informed. The question then becomes one of determining the best way to combine multiple rules. This is the primary objective of this paper – to provide a theoretical basis for combining rules. Given the items in a customer's basket, we combine rules when necessary to estimate the probabilities of each relevant consequent being selected by the customer, and recommend the item with the highest probability. Note that this is not unlike what the single rule approach would do, had there been a rule whose antecedent covered the entire basket – i.e., combinations of rules can be interpreted in much the same way as any single rule would be.

Before we calculate the probabilities associated with each potential recommendation however, we need to identify the rules to combine. It is quite possible that there will be multiple potential combinations to choose from. For example, we have already seen that rules R_5 and R_8 could be combined to estimate the probability for x_2. From Table 1, we can also see that rules R_6 and R_7 could be used to estimate the same probability as well. Different combinations of rules can yield different probability estimates, and we show how to choose the best combination from the different alternatives.

3. MLR: Maximum Likelihood Recommendations

MLR can be viewed as a three step process. The first step identifies all the eligible rules and from them, the feasible consequents. For each of these feasible consequents, the best probability estimate conditioned on the basket is identified in the second step. The third step selects the consequent with the highest probability.

3.1 Identifying Eligible Rules and Consequents

We first find all the eligible rules by ensuring that all items in the antecedent of a selected rule appear in the basket while its consequent does not. The consequents of the eligible rules are added to a consequent list $\mathcal{M}$. In our example, $\mathcal{M}$ contains two consequents $\{x_1, x_2\}$.

3.2 Computing the Probability of a Consequent

Given a customer with a basket **B**, we are interested in estimating $P(x \mid \mathbf{B})$, the probability that she will choose item x from $\mathcal{M}$. While we would ideally like to use a rule that has x as the consequent and an antecedent identical to **B**, such a rule may not exist. It is more likely that we will find several rules with x as the consequent, whose antecedents are subsets of **B.** These rules can be used, with appropriate conditional independence assumptions, to arrive at an estimate for $P(x \mid \mathbf{B})$. Such assumptions have been extensively used for estimation when the available data are not sufficient to estimate the full distributions, and have been found to be robust in practice (Domingos and Pazzani 1997). [4] For example, Naïve Bayes classifiers are known to perform very well in many applications (Han et al. 2012, page 350). According to Shmueli et al. (2010, p. 153), techniques using such assumptions often rely on the orderings of the probability estimates which are usually close to accurate even if many of these assumptions are violated.

Specifically, if we have multiple eligible rules with disjoint antecedents and a common consequent x, we can estimate $P(x \mid \mathbf{B})$ by combining the information in these rules under the assumption that the antecedents are conditionally independent given the common consequent x. Note that the antecedents of the rules being combined have to be disjoint to avoid double counting the impact of the common items in the rules[5].

---

[4] To examine the implications of making conditional independence assumptions, we conducted experiments where we estimated directly from the data the probabilities associated with all feasible consequents given the entire basket (so that conditional independence assumptions are not needed). The recommendations from using such estimates were much worse compared to when rules are combined with appropriate conditional independence assumptions. This is because the number of transactions that support the entire basket reduces drastically as the size of the basket grows, and the estimates when considering the full baskets become less reliable.

[5] The following example clarifies why only rules with disjoint antecedents should be combined. Suppose that for a basket **B** = {A1, A2}, there are three eligible rules that could be used for computing the probability of the user selecting

Suppose there are r eligible rules with disjoint antecedents that have x as the consequent. Let the antecedent of rule R_j be A_j and let **A** represent $\bigcup_{j=1}^{r} A_j$. By assuming that the antecedents A_j are conditionally independent given x, we can approximate $P(x \mid \mathbf{B})$ as $P(x \mid \mathbf{A})$ below.

$$P(x \mid \mathbf{A}) = \frac{P(\mathbf{A} \mid x)*P(x)}{P(\mathbf{A})} = \frac{P(\mathbf{A} \mid x)*P(x)}{P(x,\ \mathbf{A})+P(\bar{x},\mathbf{A})} = \frac{(\prod_{j=1}^{r} P(A_j \mid x))*P(x)}{(\prod_{j=1}^{r} P(A_j \mid x))*P(x)+(\prod_{j=1}^{r} P(A_j \mid \bar{x}))*P(\bar{x})} \quad (1)$$

In order to evaluate $P(x \mid \mathbf{A})$ using (1), we need to know $P(x)$ and $P(\bar{x})$, along with $P(A_j \mid x)$ and $P(A_j \mid \bar{x})$ for each of the r rules R_j. $P(x)$ is simply the support of the consequent x, while $P(\bar{x})$ is (1 - $P(x)$). Each of the parameters $P(A_j \mid x)$ and $P(A_j \mid \bar{x})$ can be obtained from the confidences of the rules involved (i.e., $P(x \mid A_j)$), and the supports of x and A_j (i.e., $P(x)$ and $P(A_j)$). All these parameters can be pre-computed from the data at the time the rules are mined.

Consider computing $P(x_1 \mid \mathbf{B})$ for consequent x_1 using rules R_1 and R_4 from the example in Table 1. Assume that $P(x_1) = 0.2$, $P(A_1) = 0.2$, and $P(A_4) = 0.21$, for our illustrative example. Using these probabilities and the confidences of the two rules, the additional parameters required in (1) can be calculated as:

$$P(\bar{x}_1) = 1 - P(x_1) = 0.8,$$

$$P(A_1 \mid x_1) = \frac{P(x_1 \mid A_1)*P(A_1)}{P(x_1)} = 0.6, \qquad P(A_1 \mid \bar{x}_1) = \frac{P(A_1)-P(x_1 \mid A_1)*P(A_1)}{(1-P(x_1))} = 0.1,$$

$$P(A_4 \mid x_1) = \frac{P(x_1 \mid A_4)*P(A_4)}{P(x_1)} = 0.45, \quad P(A_4 \mid \bar{x}_1) = \frac{P(A_4)-P(x_1 \mid A_4)*P(A_4)}{(1-P(x_1))} = 0.15.$$

Substituting these values into (1), we get

$$P(x_1 \mid \mathbf{B}) = \frac{0.6*0.45*0.2}{0.6*0.45*0.2+0.1*0.15*0.8} = 0.82.$$

This example illustrates how the information from the two rules R_1 and R_4 can be combined. The rule with the highest confidence for consequent x_1 was R_1, with a confidence of 0.6. We see that the estimated value of $P(x_1 \mid \mathbf{B})$ is much higher than 0.6. This suggests that the estimate of the probability of the customer choosing a particular consequent can be quite different when multiple rules are considered, relative to that when single rules are used.

---

consequent X – **R1**:{A1}→{X}, **R2**:{A2}→{X} and **R3**:{A1, A2}→{X}. If we allow rules with non-disjoint antecedents to be combined, we could combine all three rules. However, combining all three rules would result in an incorrect estimate, as clearly, **R3** provides the correct probability estimate for selecting X when the customer has the basket **B**.

3.3 Multiple Ways of Computing the Probability of a Consequent

While the illustration above combined rules R_1 and R_4 to estimate the probability that x_1 will be chosen given the basket **B**, this probability can also be estimated using the rules R_2 and R_3. In order to do so, we need the estimates of $P(x_1 \mid A_2)$ and $P(x_1 \mid A_3)$ from Table 1, along with $P(A_2)$ and $P(A_3)$. Suppose $P(A_2) = 0.2$ and $P(A_3) = 0.3$. Then $P(x_1 \mid \mathbf{B})$ can be estimated as 0.75 using equation (1). This estimate is different from that obtained when R_1 and R_4 were combined.

As this example illustrates, there could be many groups of rules that could be combined to estimate the probability of a feasible consequent. We call each such group an *admissible group*. Formally, an admissible group is defined as a set of eligible rules with disjoint antecedents that have the same consequent.

An admissible group to which no other eligible rule can be added while maintaining admissibility is called a *maximal admissible group*. When the union of the antecedents of the rules in the admissible group is equal to the basket **B**, we say that the group fully covers the basket; it partially covers the basket otherwise. The collection of all the eligible rules for a given consequent x is called a *consequent set*, and is denoted $\mathcal{G}(x)$. In our example, the consequent set $\mathcal{G}(x_1) = \{R_1, R_2, R_3, R_4\}$ and the two maximal admissible groups corresponding to x_1 are $\mathcal{S}_1 = \{R_1, R_4\}$ and $\mathcal{S}_2 = \{R_2, R_3\}$.

3.4 Comparing Maximal Admissible Groups

As we saw in Section 3.3, it may be possible to compute the confidence of x using one of several admissible groups. A natural question is, which admissible group should be used to estimate $P(x \mid \mathbf{B})$? In this section, we first discuss how to compare maximal admissible groups that fully cover the basket; we then extend our findings to maximal admissible groups that partially cover the basket.

Ideally, we should use that admissible group which can best approximate the true joint distribution across the items in the basket **B** and x, i.e., $P(\mathbf{B}, x)$. Therefore, we compare the admissible groups using the likelihood of each group generating the true underlying distribution $P(\mathbf{B}, x)$. The likelihoods of interest in our case are those associated with the probability models implied by the collection of rules for each admissible group. Specifically, each admissible group corresponds to a probability model with some associated conditional independence assumptions. For example, the admissible group $\mathcal{S}_1 = \{R_1, R_4\}$ assumes that the set $\{i_1, i_2\}$ is conditionally independent of the set $\{i_3\}$ given x_1, while $\mathcal{S}_2 = \{R_2, R_3\}$ assumes that the set $\{i_2, i_3\}$ is conditionally independent of the set $\{i_1\}$ given x_1. Therefore, by comparing the admissible groups using their likelihoods, we are essentially checking which conditional independence assumption is most likely to hold, given the data. In essence, this problem can be viewed as one of maximizing the

likelihood that the observed data is generated from the competing probability models represented by the admissible groups. [6]

The maximum-likelihood framework requires the estimation of the likelihoods from training data. This is not feasible in real time as training data sets can be large, and many parameters need to be estimated for each probability model. Consequently, for this to be a useful approach, we have to transform this into a problem that can be solved in real time. We show that the log-likelihood can be conveniently represented as a function of the mutual information[7] associated with the rules in an admissible group, and the entropies of the items in the basket. The mutual information terms can be pre-computed for every rule and kept available for use at run-time, which eliminates the need to estimate parameters from the data during the recommendation process.

We consider the mutual information associated with a rule to be the mutual information across all the individual attributes in the rule (including the items in both the antecedent and the consequent of the rule). Therefore, the mutual information (MI) across attributes $i_1, \ldots, i_n$ is (Kullback 1959):

$$MI(i_1, \ldots, i_n) = \sum_{i_1,\ldots,i_n} P(i_1, \ldots, i_n) \log \frac{P(i_1, \ldots, i_n)}{P(i_1) * \ldots * P(i_n)}.$$

Thus, the mutual information associated with a rule R_j having antecedent $A_j = \{i_{j1}, \ldots, i_{jk}\}$ and consequent x_m is

$$\mathrm{MI}(R_j) = \mathrm{MI}(A_j, x_m) = \mathrm{MI}(i_{j1}, \ldots, i_{jk}, x_m) = \sum_{i_{j1},\ldots,i_{jk},x_m} P(i_{j1}, \ldots, i_{jk}, x_m) \log \frac{P(i_{j1},\ldots,i_{jk},x_m)}{P(i_{j1})*\ldots*P(i_{jk})*P(x_m)}.$$

The entropy of an item i is $\mathrm{H}(i) = -\sum_{\mathrm{i}} \mathrm{P}(i) \log \mathrm{P}(i)$.

The mutual information associated with a rule captures the mutual dependency across all the items in the antecedent and the consequent of the rule. The entropy of an attribute is a measure of how much uncertainty is represented in the probability distribution of the attribute.

Proposition 1 shows that the log-likelihood associated with a maximal admissible group can be represented as the sum of the mutual information terms associated with each participating rule, less the sum of the entropies associated with every item in the basket **B** and the entropy associated with the consequent. Proposition 1 helps represent the problem using the mutual information terms associated with rules. This has intuitive appeal, as the mutual information term for a rule is higher if the items in the antecedent and

---

[6] Note that the problem of finding the best admissible group can also be viewed as one of minimizing the Kullback-Leibler distance between the true underlying distribution $P(\mathbf{B}, x)$ and the distribution implied by the rules in the selected admissible group. Minimizing the Kullback-Leibler distance is equivalent to maximizing the log-likelihood of an admissible group generating the true underlying distribution $P(\mathbf{B}, x)$ (Aalto 2014). Therefore, all our results would still hold.

[7] Mutual information is also called "total correlation" by Watanabe (1960).

the consequent are more dependent on each other. The best admissible group therefore, is the one where the rules collectively convey as much information about the consequent as possible.

Proposition 1: *Given a consequent and all corresponding admissible groups that fully cover the basket, the admissible group that maximizes the likelihood also has the highest sum of the mutual information terms associated with the participating rules.*

Proof: We first show that the log-likelihood associated with a maximal admissible group can be represented as the sum of the mutual information terms associated with each participating rule, less the sum of the entropies associated with the consequent and all the items in the basket.

Given a basket $\mathbf{B} = \{i_1, \ldots, i_n\}$, let the consequent of interest be i_{n+1}. To estimate the likelihood of a probability model associated with an admissible group, we consider the distribution associated with these items, based on the absence or presence of each item in every transaction of the data set. We denote the binary attributes corresponding to the set of items as $\mathbf{I} = \{\boldsymbol{i}_1, \ldots, \boldsymbol{i}_n, \boldsymbol{i}_{n+1}\}$. Let the dataset $\mathbf{T}$ consist of s transactions, i.e., $\mathbf{T} = \{\boldsymbol{t}^1, \boldsymbol{t}^2, \ldots, \boldsymbol{t}^s\}$, where $\boldsymbol{t}^j$ is a vector of ones and zeroes corresponding to the presence and absence of the items $\boldsymbol{i}_1, \ldots, \boldsymbol{i}_{n+1}$ in the j^{th} transaction.

Let there be r rules in an admissible group under consideration, and let $\mathbf{A}_l$ denote the attributes corresponding to the antecedent A_l in the l^{th} rule of the admissible group. The probability distribution corresponding to the rules in the admissible group can be written as $P(\mathbf{I}) = \prod_{l=1}^{r} P(\mathbf{A}_{\boldsymbol{l}} \mid \boldsymbol{i}_{\boldsymbol{n+1}}) * P(\boldsymbol{i}_{n+1})$. The probability associated with the items appearing in the j^{th} transaction, $P(\boldsymbol{t}^j)$, is represented as $\prod_{l=1}^{r} P\left(\mathbf{A}_l^j | \boldsymbol{i}_{n+1}^j\right) * P\left(\boldsymbol{i}_{n+1}^j\right)$, and the likelihood for the admissible group is

$$L = \prod_{j=1}^{s} P\left(\boldsymbol{t}^j\right) = \prod_{j=1}^{s} \left(\prod_{l=1}^{r} \left(P(\mathbf{A}_l^j | \boldsymbol{i}_{n+1}^j)\right) * P(\boldsymbol{i}_{n+1}^j)\right).$$

The log-likelihood, L', is $\log(L) = L' = \sum_{j=1}^{s} \sum_{l=1}^{r} \log P(\boldsymbol{A}_l^j | \boldsymbol{i}_{n+1}^j) + \sum_{j=1}^{s} \log P(\boldsymbol{i}_{n+1}^j)$

$$= \sum_{l=1}^{r} \sum_{j=1}^{s} \log P(\mathbf{A}_l^j | \boldsymbol{i}_{n+1}^j) + \sum_{j=1}^{s} \log P(\boldsymbol{i}_{n+1}^j). \qquad (2)$$

Each instance $(\mathbf{A}_l^j, \boldsymbol{i}_{n+1}^j)$ corresponds to one of the $2^{(n+1)}$ realizations of the attributes (i.e., the set of 0-1 values the attributes can assume) comprising the antecedent and the consequent of the l^{th} rule. Let the probability for the k^{th} realization of $(\mathbf{A}_l, \boldsymbol{i}_{n+1})$ be $P^k(\mathbf{A}_l, \boldsymbol{i}_{n+1})$. [8] Further, let the frequency of occurrences for the k^{th} realization of $(\mathbf{A}_l, \boldsymbol{i}_{n+1})$ be $f^k(\mathbf{A}_l, \boldsymbol{i}_{\boldsymbol{n+1}})$, and the frequency of occurrences for the k^{th} realization of $\boldsymbol{i}_{n+1}$ be $f^k(\boldsymbol{i}_{n+1})$. Therefore,

[8] Depending on the number of items being considered in each distribution, the number of possible realizations will vary. Hence, the range for the index k will also vary. For notational simplicity, we do not spell out the range in the following expressions, implicitly assuming that k will vary over the appropriate range in each expression.

$\sum_{j=1}^{s} \log P\left(\mathbf{A}_l^j \middle| \boldsymbol{i}_{n+1}^j\right) = \sum_k f^k(\mathbf{A}_l, \boldsymbol{i}_{n+1}) \log P^k(\mathbf{A}_l | \boldsymbol{i}_{n+1})$, and $\sum_{j=1}^{s} \log P\left(\boldsymbol{i}_{n+1}^j\right) = \sum_k f^k(\boldsymbol{i}_{n+1}) \log P^k(\boldsymbol{i}_{n+1})$.

Since $P^k(\mathbf{A}_l, \boldsymbol{i}_{n+1}) = \frac{f^k(\mathbf{A}_l, \boldsymbol{i}_{n+1})}{s}$ and $P^k(\boldsymbol{i}_{n+1}) = \frac{f^k(\boldsymbol{i}_{n+1})}{s}$, we have

$$\sum_{j=1}^{s} \log P\left(\mathbf{A}_l^j \middle| \boldsymbol{i}_{n+1}^j\right) = s \sum_k \frac{f^k(\mathbf{A}_l, \boldsymbol{i}_{n+1})}{s} \log P^k(\mathbf{A}_l | \boldsymbol{i}_{n+1}) = s \sum_k P^k(\mathbf{A}_l, \boldsymbol{i}_{n+1}) \log P^k(\mathbf{A}_l | \boldsymbol{i}_{n+1}).$$

Similarly, $\sum_{j=1}^{s} \log P\left(\boldsymbol{i}_{n+1}^j\right) = s \sum_k P^k(\boldsymbol{i}_{n+1}) \log P^k(\boldsymbol{i}_{n+1})$.

Substituting for $\sum_{j=1}^{s} \log P\left(\mathbf{A}_l^j \middle| \boldsymbol{i}_{n+1}^j\right)$ and $\sum_{j=1}^{s} \log P\left(\boldsymbol{i}_{n+1}^j\right)$ in (2) we have,

$$L' = \sum_{l=1}^{r} s \sum_k P^k(\mathbf{A}_l, \boldsymbol{i}_{n+1}) \log P^k(\mathbf{A}_l | \boldsymbol{i}_{n+1}) + s \sum_k P^k(\boldsymbol{i}_{n+1}) \log P^k(\boldsymbol{i}_{n+1})$$

$$= s \sum_{l=1}^{r} \sum_k P^k(\mathbf{A}_l, \boldsymbol{i}_{n+1}) \log P^k(\mathbf{A}_l | \boldsymbol{i}_{n+1}) + s \sum_k P^k(\boldsymbol{i}_{n+1}) \log P^k(\boldsymbol{i}_{n+1}) \quad (3)$$

Consider the term $\sum_k P^k(\mathbf{A}_l, \boldsymbol{i}_{n+1}) \log P^k(\mathbf{A}_l | \boldsymbol{i}_{n+1})$ in the first sum. Let the antecedent $\mathbf{A}_l$ comprise of the m items $\{i_1, \ldots, i_m\}$.

$$\sum_k P^k(\mathbf{A}_l, \boldsymbol{i}_{n+1}) \log P^k(\mathbf{A}_l | \boldsymbol{i}_{n+1}) = \sum_k P^k(\boldsymbol{i}_1, \boldsymbol{i}_2, \ldots, \boldsymbol{i}_m, \boldsymbol{i}_{n+1}) \log \left(\frac{P^k(\boldsymbol{i}_1, \boldsymbol{i}_2, \ldots, \boldsymbol{i}_m, \boldsymbol{i}_{n+1})}{P^k(\boldsymbol{i}_{n+1})} \right)$$

$$= \sum_k P^k(\boldsymbol{i}_1, \boldsymbol{i}_2, \ldots, \boldsymbol{i}_m, \boldsymbol{i}_{n+1}) \log \left(\frac{P^k(\boldsymbol{i}_1, \boldsymbol{i}_2, \ldots, \boldsymbol{i}_m, \boldsymbol{i}_{n+1}) * P^k(\boldsymbol{i}_1) * \ldots * P^k(\boldsymbol{i}_m)}{P^k(\boldsymbol{i}_1) * \ldots * P^k(\boldsymbol{i}_m) * P^k(\boldsymbol{i}_{n+1})} \right)$$

$$= \sum_k P^k(\boldsymbol{i}_1, \boldsymbol{i}_2, \ldots, \boldsymbol{i}_m, \boldsymbol{i}_{n+1}) \log \left(\frac{P^k(\boldsymbol{i}_1, \boldsymbol{i}_2, \ldots, \boldsymbol{i}_m, \boldsymbol{i}_{n+1})}{P^k(\boldsymbol{i}_1) * \ldots * P^k(\boldsymbol{i}_m) * P^k(\boldsymbol{i}_{n+1})} \right)$$

$$+ \sum_k P^k(\boldsymbol{i}_1, \boldsymbol{i}_2, \ldots, \boldsymbol{i}_m, \boldsymbol{i}_{n+1}) \log (P^k(\boldsymbol{i}_1) * \ldots * P^k(\boldsymbol{i}_m))$$

$$= \sum_k P^k(\boldsymbol{i}_1, \boldsymbol{i}_2, \ldots, \boldsymbol{i}_m, \boldsymbol{i}_{n+1}) \log \left(\frac{P^k(\boldsymbol{i}_1, \boldsymbol{i}_2, \ldots, \boldsymbol{i}_m, \boldsymbol{i}_{n+1})}{P^k(\boldsymbol{i}_1) * \ldots * P^k(\boldsymbol{i}_m) * P^k(\boldsymbol{i}_{n+1})} \right) +$$

$$+ \sum_k P^k(\boldsymbol{i}_1, \boldsymbol{i}_2, \ldots, \boldsymbol{i}_m, \boldsymbol{i}_{n+1}) \log P^k(\boldsymbol{i}_1) + \cdots + \sum_k P^k(\boldsymbol{i}_1, \boldsymbol{i}_2, \ldots, \boldsymbol{i}_m, \boldsymbol{i}_{n+1}) \log P^k(\boldsymbol{i}_m)$$

Over all possible realizations of $\{\boldsymbol{i}_1, \boldsymbol{i}_2, \ldots, \boldsymbol{i}_m, \boldsymbol{i}_{n+1}\}$, $\sum_k P^k(\boldsymbol{i}_1, \boldsymbol{i}_2, \ldots, \boldsymbol{i}_m, \boldsymbol{i}_{n+1}) \log P^k(\boldsymbol{i}_j)$ simplifies to $\sum_k P^k(\boldsymbol{i}_j) \log P^k(\boldsymbol{i}_j)$, with k now indexing all possible realizations of $\{\boldsymbol{i}_j\}$, i.e., 0 and 1. Therefore,

$$\sum_k P^k(\mathbf{A}_l, \boldsymbol{i}_{n+1}) \log P^k(\mathbf{A}_l | \boldsymbol{i}_{n+1}) = \sum_k P^k(\boldsymbol{i}_1, \boldsymbol{i}_2, \ldots, \boldsymbol{i}_m, \boldsymbol{i}_{n+1}) \log \left(\frac{P^k(\boldsymbol{i}_1, \boldsymbol{i}_2, \ldots, \boldsymbol{i}_m, \boldsymbol{i}_{n+1})}{P^k(\boldsymbol{i}_1) * \ldots * P^k(\boldsymbol{i}_m) * P^k(\boldsymbol{i}_{n+1})} \right)$$

$$+ \sum_k P^k(\boldsymbol{i}_1) \log P^k(\boldsymbol{i}_1) + \cdots + \sum_k P^k(\boldsymbol{i}_m) \log P^k(\boldsymbol{i}_m)$$

$$= \sum_k P^k(\boldsymbol{i}_1, \boldsymbol{i}_2, \ldots, \boldsymbol{i}_m, \boldsymbol{i}_{n+1}) \log \left(\frac{P^k(\boldsymbol{i}_1, \boldsymbol{i}_2, \ldots, \boldsymbol{i}_m, \boldsymbol{i}_{n+1})}{P^k(\boldsymbol{i}_1) * \ldots * P^k(\boldsymbol{i}_m) * P^k(\boldsymbol{i}_{n+1})} \right) + \sum_{q \in \mathbf{A}_l} \sum_k P^k(\boldsymbol{i}_q) \log P^k(\boldsymbol{i}_q)$$

$$= MI(\boldsymbol{i}_1, \boldsymbol{i}_2, \ldots, \boldsymbol{i}_m, \boldsymbol{i}_{n+1}) - \sum_{q \in \mathbf{A}_l} (H_q).$$

Therefore, by substituting for $\sum_k P^k(\mathbf{A}_l, \boldsymbol{i}_{n+1}) \log P^k(\mathbf{A}_l | \boldsymbol{i}_{n+1})$ in (3) we get

$$L' = s \sum_{l=1}^{r} MI_l - s \sum_{l=1}^{r} \sum_{q \in \mathbf{A}_l} (H_q) - sH_{n+1} = s\left(\sum_{l=1}^{r} MI_l - \sum_{q=1}^{n+1} (H_q)\right).$$

The entropy terms in the above expression are the same for every admissible group under consideration. Therefore, the admissible group that maximizes the likelihood has the highest sum of mutual information terms associated with the participating rules. ■

The mutual information terms can be pre-computed for every rule and kept available for use at run-time. Comparing admissible groups using mutual information is straightforward. For example, suppose the mutual information values of the rules in $\mathcal{S}_1 = \{R_1, R_4\}$ and $\mathcal{S}_2 = \{R_2, R_3\}$ are as in Table 2. Since the sum of the mutual information values for the rules in $\mathcal{S}_1$ (0.156 + 0.045 = 0.201) is less than the corresponding value for the rules in $\mathcal{S}_2$ (0.164 + 0.098 = 0.262), $\mathcal{S}_2$ will be preferred over $\mathcal{S}_1$.

Rules	Items in the rules	MI
R_1	i_1, i_2, x_1	0.156
R_2	i_2, i_3, x_1	0.164
R_3	i_1, x_1	0.098
R_4	i_3, x_1	0.045

Table 2: Mutual Information of rules in $\mathcal{S}_1$ and $\mathcal{S}_2$

Given a consequent x and associated consequent set $\mathcal{G}(x)$, the problem of finding the best admissible group – i.e., the admissible group that maximizes the sum of mutual information values – can be formulated as the integer program (AG) below

$$\begin{aligned} &\text{Max } \textstyle\sum_{i \in G(x)} M_i y_i, \\ &\text{s.t.} \quad \textstyle\sum_{i \in G(x)} a_{ij} y_i = 1 \qquad \forall\, j \in \mathbf{B}, \qquad \text{(AG)} \\ &\qquad y_i \in \{0, 1\} \qquad \forall\, i \in \mathcal{G}(x), \end{aligned}$$

where M_i is the mutual information corresponding to $R_i \in \mathcal{G}(x)$, a_{ij} is 1 if the j^{th} item of the basket is present in rule $R_i \in \mathcal{G}(x)$ and 0 otherwise, and y_i is a binary decision variable that is set to 1 if rule $R_i \in \mathcal{G}(x)$ is included in the solution and to 0 otherwise. The constraint ensures that an item in the basket can only be present in the antecedent of exactly one rule selected for inclusion in the admissible group. We note that AG is a set partitioning problem (Balas and Padberg 1976), and therefore NP-Hard. The reason for this intractability is the combinatorial number of ways in which rules may be combined, where each combination (admissible group) is associated with a unique set of conditional independence assumptions.

So far we have considered maximal admissible groups that fully cover the basket. However, there could exist maximal admissible groups that cover only a subset of the basket; indeed, it is possible that none of the maximal admissible groups cover the entire basket. In such cases, when considering an admissible group, we assume that the items that are not covered and the consequent are independent of each other, and that the corresponding mutual information terms are zero. While this may not be strictly true, the fact that such rules were not retained after mining suggests that the dependence is weak. This can be viewed as

ensuring that rules of the form $\{i\}\rightarrow\{x\}$ exist for every item i in the basket by adding dummy rules with mutual information values of zero wherever necessary.

3.5 Finding a Good Admissible Group

As noted earlier, the problem of finding the admissible group that maximizes the sum of the mutual information values for the rules is NP-Hard. When the number of items in a customer's basket is small, the number of possible admissible groups is likely to be small and the problem can be solved easily. However, when the basket is large, it may be difficult to determine the best admissible group quickly. We propose a greedy heuristic to solve large instances of this problem, as such approaches have been shown to work well on set partitioning problems (e.g., Ergun et al. 2007). It is easy to implement, and exploits the properties of the optimal solution presented in Proposition 2 and Corollary 1.

Proposition 2: *The mutual information corresponding to a rule* $\mathbf{A}\rightarrow\{x\}$ *is always greater than or equal to the sum of the mutual information values corresponding to rules* $\{A_1\}\rightarrow\{x\}$, $\{A_2\}\rightarrow\{x\},\ldots,$ $\{A_n\}\rightarrow\{x\}$ *if the antecedents* $A_1, A_2,\ldots,A_n$ *are mutually disjoint and* $\bigcup_{j=1}^{n} A_j = \mathbf{A}$.

Proof: The mutual information corresponding to the rule $\{A\}\rightarrow\{x\}$ is

$$= \sum_k P^k(\mathbf{A}, \boldsymbol{x}) \log\left(\frac{P^k(\mathbf{A}, x)}{\prod_{i_m \in A} P^k(\boldsymbol{i}_m) * P^k(\boldsymbol{x})}\right)$$

$$= \sum_k P^k(\boldsymbol{A}, \boldsymbol{x}) \log\left(\frac{P^k(\mathbf{A}|\, x)}{\prod_{i_m \in A} P^k(\boldsymbol{i}_m)}\right) \qquad (4)$$

The sum of mutual information values of the rules $\{A_1\}\rightarrow\{x\}$, $\{A_2\}\rightarrow\{x\},\ldots,$ $\{A_n\}\rightarrow\{x\}$ is

$$\sum_{j=1}^{n} \sum_k P^k(\boldsymbol{A}_l, \boldsymbol{x}) \log\left(\frac{P^k(\mathbf{A}_j, x)}{(\prod_{i_m \in A_j} P^k(\boldsymbol{i}_m)) * P^k(x)}\right)$$

$$= \sum_{j=1}^{n} \sum_k P^k(\boldsymbol{A}_j, \boldsymbol{x}) \log\left(\frac{P^k(\mathbf{A}_j|\boldsymbol{x}_i)}{\prod_{i_m \in A_j} P^k(\boldsymbol{i}_m)}\right)$$

$$= \sum_{j=1}^{n} \sum_k P^k(\boldsymbol{A}, \boldsymbol{x}) \log\left(\frac{P^k(\mathbf{A}_j|\boldsymbol{x})}{\prod_{i_m \in A_j} P^k(\boldsymbol{i}_m)}\right)$$

$$= \sum_k P^k(\boldsymbol{A}, \boldsymbol{x}) \log\left(\frac{\prod_{j=1}^{n} P^k(\mathbf{A}_j|\, x)}{\prod_{j=1}^{n} \prod_{i_m \in A_j} P^k(\boldsymbol{i}_m)}\right)$$

$$= \sum_k P^k(\boldsymbol{A}, \boldsymbol{x}) \log\left(\frac{\prod_{j=1}^{n} P^k(\mathbf{A}_j|\, x)}{\prod_{i_m \in A} P^k(\boldsymbol{i}_m)}\right) \qquad (5)$$

since $\bigcup_{j=1}^{n} A_j = \mathbf{A}$.

The denominators in the logarithm expressions are identical in equations (4) and (5), as are the coefficients for the logarithm terms. The numerator from equation (4) is

$$\sum_k P^k(\mathbf{A}, \boldsymbol{x}) \log P^k(\mathbf{A}|x) = \sum_k P^k(\mathbf{A}|\boldsymbol{x}) P^k(\boldsymbol{x}) \log P^k(\mathbf{A}|x).$$

Because the distribution $P^k(\mathbf{A}|\boldsymbol{x})$ is fixed, the expression $\sum_k P^k(\mathbf{A}|\boldsymbol{x}) \log P^k(\mathbf{A}|x)$ is the maximum possible for each value of $\boldsymbol{x}$. Therefore, the numerator from equation (4) is always greater than or equal to that from equation (5). It is equal if and only if all the conditional independence assumptions implied by the rules $\{A_1\} \rightarrow \{x\}$, $\{A_2\} \rightarrow \{x\}, \ldots, \{A_n\} \rightarrow \{x\}$ hold. ■

When a consequent set includes rules of the form described in Proposition 2, we say that rule $\{A\} \rightarrow \{x\}$ subsumes the rules $\{A_1\} \rightarrow \{x\}$, $\{A_2\} \rightarrow \{x\}, \ldots, \{A_n\} \rightarrow \{x\}$.

Corollary 1: *An admissible group $\mathcal{S}$ is at least as good as another admissible group $\mathcal{T}$ if the antecedent of every rule in $\mathcal{T}$ is a subset of the antecedent of some rule in $\mathcal{S}$. $\mathcal{S}$ is strictly better if any of the conditional independence assumptions implied by $\mathcal{T}$ but not by $\mathcal{S}$ do not hold.*

Proof: The mutual information values corresponding to the identical rules in $\mathcal{S}$ and $\mathcal{T}$ are the same. If there exists at least one rule R in $\mathcal{S}$ such that the antecedents of n rules $R_1, \ldots, R_n$ in $\mathcal{T}$ are proper subsets of the antecedent of R , then according to Proposition 2, the mutual information corresponding to R is greater than or equal to the sum of the mutual information values corresponding to the rules $R_1, \ldots, R_n$. Hence the sum of the mutual information values corresponding to the rules in $\mathcal{S}$ is greater than or equal to the sum of the mutual information values corresponding to the rules in $\mathcal{T}$. As shown in Proposition 2, the sum of the mutual information values corresponding to the rules $R_1, \ldots, R_n$ are strictly less if any of the conditional independence assumptions implied by $\mathcal{S}$ but not $\mathcal{T}$ do not strictly hold. ■

Input: (i) Basket $\mathbf{B} = \{i_1, \ldots, i_n\}$.

(ii) Consequent Set $\mathcal{G}(x) = \{R_1, \ldots, R_m\}$.

Output: An admissible group for x

Steps:

1. Set item list $Z = \mathbf{B}$. Initialize admissible group $Y = \text{ø}$.
2. Sort the rules in $\mathcal{G}(x)$ in decreasing order of mutual information.
3. Repeat steps 3a and 3b till no more rules can be added to Y.
   - 3a. Add the next rule from $\mathcal{G}(x)$ to Y if all items in its antecedent are in Z. This rule has the highest mutual information among all rules whose antecedents have items in Z.
   - 3b. Remove the items from Z that are present in the antecedent of the added rule.

Figure 1: Heuristic for finding a good admissible group

The heuristic to find an admissible group for a consequent x is shown in Figure 1. The intuition is to keep adding rules with high mutual information into the admissible group without violating admissibility until no more rules can be added. Hence, the rules are arranged in decreasing order of mutual information and are added to the admissible group starting from the rule with highest mutual information until all the items in the basket are covered or all the rules have been considered. By selecting rules with higher mutual information, the heuristic ensures that the solution does not include two or more rules from the consequent set that are subsumed by a single rule from that set.

3.6 Computational Complexity of MLR

As mentioned earlier, the first step when recommending an item using MLR is to identify eligible rules. This requires the items in the basket to be sorted in some pre-determined order (e.g., lexicographic), and given n items in the dataset, can be done in $O(n\log(n))$ in the worst case. The eligible rules are then identified by verifying whether the items in the antecedents and consequents of the rules are present in the basket. This can be done via binary search, in $O(\log(n))$. Therefore, the presence of all items in the antecedent can be checked in $O(n\log(n))$. If there are a total of m rules, then the complexity of checking their eligibilities therefore, is $mn\log(n)$.

Once the set of eligible rules is created, potential items for recommendation are identified from the consequents of these rules, along with the consequent sets of each. This is done by scanning the eligible rules and adding the consequent x of each rule into $\mathcal{M}$ if it is not present already, and by adding the rule to the appropriate consequent list $\mathcal{G}(x)$. The presence of an item in $\mathcal{M}$ can be checked using binary search, while binary insertion can be used to add a new consequent into $\mathcal{M}$. The complexity of checking for the presence of an item or adding an item in $\mathcal{M}$ is $O(\log(n))$. Thus, given n items in the data set, the complexity of identifying potential recommendations and creating the consequent sets is $O(n\log(n))$. When all the rules in the set of eligible rules have been considered, $\mathcal{M}$ contains all items x that can potentially be recommended and the corresponding lists $\mathcal{G}(x)$ are their consequent sets.

The next step is to apply the heuristic proposed in Figure 1 to each consequent in $\mathcal{M}$. Creating the admissible group for a consequent requires checking whether the items in the antecedents of rules in the associated consequent set are present in the basket; this is of complexity $O(n\log(n))$ for each rule. In the worst case, items in the antecedents of all m rules may have to be checked for their presence in the basket; this is of complexity $O(mn\log(n))$. This procedure has to be repeated for every possible consequent, and therefore the worst case complexity of the heuristic is $O(mn^2\log(n))$.

The probabilities can be estimated for each item in $\mathcal{M}$ using the rules in the admissible groups in O(m). Across all items therefore, the complexity is O(mn). The item with highest estimated probability can be identified in O(n), through a single scan of $\mathcal{M}$.

Thus, the overall complexity of MLR is O($mn^2\log(n)$), which is linear in the number of rules mined, and has a low-order polynomial complexity in the number of items in the dataset. Since the size of a typical basket is much smaller than n, the average complexity should be much better. Note that the complexity does not depend on the size of the dataset – once the rules are mined, MLR does not use the dataset to find items to recommend.

4. Experiments

We conduct a large number of systematic experiments on several real data sets to investigate the quality of recommendations made by MLR. First, we conduct experiments comparing recommendations made using the optimal approach to identify admissible groups (i.e., formulation AG) with those made by the greedy heuristic presented in Figure 1. Then, we conduct experiments comparing recommendations made using MLR with various benchmarks including the single rule approaches of Zaïane (2002), Wang and Shao (2004) and Baralis et al. (2004) (called L3G), and the rule combination methods of Li et al. (2001) and Lin et al. (2002) (comparison with non-rule based approaches are presented in Section 5). All the experiments are performed using code written in Java, on a Pentium Dual Core machine (2.6 GHz) with 32 GB of RAM.

4.1 Data

We use three real datasets in our experiments. Datasets *Retail* and *BMS-POS* are obtained from the FIMI repository (http://fimi.cs.helsinki.fi/data/), while the third dataset *comScore2013* is obtained from Wharton Research Data Services (WRDS). *Retail* is a market basket data set collected from a Belgian retail store (Brijs et al. 1999), *BMS-POS* is a point-of-sales dataset collected from a large electronics retailer (Zheng et al. 2001), and *comScore2013* is a transactional dataset consisting of items purchased by customers from various online e-retailers in the year 2013. The basic characteristics of the datasets are shown in Table 3.

Characteristics	***Retail***	***BMS-POS***	***comScore2013***
Number of items	16,470	1,657	60
Number of transaction_id-item pairs	908,069	3,351,381	84,963
Number of transactions	88,162	515,597	22,963
Average transaction length	10.3	6.5	3.7

Table 3: Dataset characteristics

One point of clarification is needed here with regard to dataset size. There are two conventions used to represent market basket datasets. In one, a basket is a record of items that are purchased together,

and would essentially comprise a list of items along with the id of the transaction – this is the convention we have followed in our paper. However, there is another commonly used convention where datasets (including the *comScore2013* dataset used in our experiments) represent a basket as transaction id-item pairs, breaking up a single transaction across many rows of data. For example, a transaction *i* involving the purchase of items *A*, *B* and *C* would be represented as a single row (record) {*i*, *A*, *B*, *C*} if the first convention was followed, while it would be represented as three separate records {*i*, *A*}, {*i*, *B*}, and {*i*, *C*} in the alternate representation. The former representation will contain as many records as transactions, while the latter will have as many records as transaction-item pairs. When comparing dataset size in terms of records, we need to make sure that the same convention is used. The row labeled "Number of transaction_id-item pairs" provides the dataset size using the latter approach (after eliminating duplicate/redundant records), while the row titled "Number of transactions" provides the number of transactions based on the former representation. The *BMS-POS* dataset is by far the largest, having 515,597 transactions, and over 3 million transaction-item pairs. The *comScore2013* dataset is the most current, and involves 22,963 transactions and about 85,000 non-redundant transaction-item pairs. The retail dataset is in between in size using either metric, with 88,162 transactions and approximately 900,000 pairs.

As can be seen from Table 3, *Retail* is the least dense of the three datasets we have used – customers purchase an average of 10.3 items from a maximum possible 16,470, resulting in a dataset density of (10.3/16,470) = 0.0625%. *comScore2013*, with a density of (3.7/60), or 6.17% is the densest, and the density of BMS-POS falls in between (0.39%).

All datasets have the items in the transactions ordered lexicographically based on their labels. We randomize the ordering of items in each transaction in order to avoid any bias that might result from this pre-ordering of the items. Eighty percent of the transactions from each dataset are used for training (e.g., generating rules or learning models) with the rest used for testing purposes.

4.2 MLR vs Rule-based Systems: Experimental Setup

The experiments involve performing five-fold cross validation tests using these datasets. In the experiments, baskets are provided to the recommender systems (MLR and the relevant benchmark) and the number of successful recommendations made by each approach is tracked. The experiments are designed to mimic the interactions of a customer at a web site to the extent possible. The recommender system can recommend items every time a customer adds an item to the basket. In order to replicate this process, each transaction in the test dataset is used to create multiple test baskets iteratively. The first basket created from a transaction contains the first item in the transaction. Each recommender system recommends an item for potential addition into the basket. If the recommended item is present in the remainder of the transaction,

the recommendation is considered successful, and the recommended item is then added to the basket to create the next basket. If both systems recommend different items successfully, both the items are added to the basket to avoid any potential bias from the addition of just one of them.[9] If neither recommendation is successful, a randomly selected item from the remainder of the transaction is added to the basket. This process is repeated for the transactions until at least half of the items in the transactions are included in the basket.[10] This is repeated for every transaction in the test dataset. The approaches are compared based on average accuracy of recommendations. The results reported are for baskets for which both approaches (MLR and the benchmark) provide recommendations.[11]

4.3 Finding the Best Admissible Group: Optimal vs Heuristic Approaches

As part of the MLR process, we need to identify the best admissible group from the many combinations that might exist. We had shown in Section 3.4 that this problem is NP-Hard, and can be represented as the set-partitioning problem AG. Further, we proposed a heuristic to achieve the same end (Figure 1). In our first set of experiments, we compare these two versions of MLR – one using the optimal solution from the set partitioning problem AG, and the other from the heuristic to understand the practical impact on recommendation accuracy of using the heuristic. Table 4 shows the results of the experiments for a support threshold of 0.2%[12] and confidence thresholds of 30%, 40%, 50% and 60% (increasing the confidence threshold beyond 60% generates very few rules).

These results show that while using the optimal admissible groups sometimes does lead to more successful recommendations, the improvement in performance is very small. For the experiments conducted on the *Retail* dataset, the results are virtually identical. The differences are greater for the experiments conducted on the datasets *BMS-POS* and *comScore2013*. However, none of the differences are statistically significant. We also found that the number of instances when the heuristic and optimal approaches choose different admissible groups is also very small. At the same time, the time taken by the heuristic for making recommendations is a fraction of the time taken by the optimal approach. Incorporating an integer programming solver into a recommender system is worthwhile only if the benefits over easily implemented procedures are substantial. Given the results of these experiments, that does not seem to be the case. Consequently, all the other results reported in this paper are based on using the heuristic. Of course,

[9] Experiments conducted by randomly adding one of the items to the basket yielded similar results.

[10] The results were similar when baskets are created until all items of a transaction but one are included in the basket.

[11] This is true for the results of all the experiments reported in the paper.

[12] This was one of the support thresholds used by Zheng et al. (2001).

in situations where the optimal approach is viable, the results are likely to be better than those currently being reported.

Dataset	Confidence	# of Rules Mined	# of baskets	# of Successful Recommendations	
				Optimal	Heuristic
Retail	30%	2,136	52,246	13,036	13,036
	40%	2,022	46,720	12,837	12,837
	50%	1,954	44,767	12,793	12,793
	60%	1,512	37,144	10,970	10,969
BMS-POS	30%	69,207	301,100	97,362	96,902
	40%	56,795	275,274	93,586	93,174
	50%	45,517	231,700	86,571	86,321
	60%	32,019	179,756	76,604	76,542
comScore2013	30%	168,392	7,166	2,487	2,471
	40%	141,035	7,166	2,373	2,355
	50%	111,719	7,168	2,147	2,128
	60%	84,954	7,171	1,703	1,695

Table 4: Comparing optimal and heuristic approaches

The number of rules generated for the same set of support and confidence thresholds depends on the dataset density. Therefore, mining *Retail* results in the fewest number of rules and mining *comScore2013* results in the most. The average number of rules generated from each dataset (for each of the parameter settings used in our experiments) is also provided in Table 4.

4.4 MLR vs Single-Rule Based Approaches

Several experiments are conducted to compare MLR with the single rule based approaches of Zaïane (2002), Wang and Shao (2004) and Baralis et al. (2004). The approach proposed by Baralis et al. (2004), called L3G, is for classification. We have adapted it to the item recommendation context – in a transactional data set, the consequents of eligible rules are analogous to classes to which a customer may belong. The performances of all the single-rule based approaches are similar, and therefore, we present only the results comparing MLR with the approach of Zaïane (2002). As in the previous section, the first set of experiments are performed with rules mined from the training datasets using a support threshold of 0.2% and confidence thresholds of 30%, 40%, 50% and 60%.

4.4.1 MLR Uses Multiple Rules

When making recommendations using MLR, the admissible group corresponding to the recommended item may contain one or multiple rules. When an item is recommended using an admissible group with only one

rule, it is typically the same as that recommended by the single-rule based benchmark. However, significant improvements in performance are observed when MLR recommends items using admissible groups with multiple rules. Table 5 presents the results for those instances for which items are recommended using multiple rules. The results are averaged over five cross validation experiments. The number of rules combined usually ranged between two and three.

Dataset	Conf. thres-hold	Baskets where MLR used ≥ 2 Rules	MLR		Single-Rule Approach (Zaïane 2002)		Improve-ment (%)
			Accuracy # (%)	Time (sec)	Accuracy # (%)	Time (sec)	
Retail	30%	16,929	917 (5.42%)	0.00054	776 (4.58%)	0.00048	18.20%***
	40%	15,975	914 (5.72%)	0.00053	773 (4.84%)	0.00048	18.24%***
	50%	15,761	881 (5.59%)	0.00051	742 (4.71%)	0.00045	18.71%***
	60%	9,214	233 (2.53%)	0.00050	211 (2.29%)	0.00045	10.43%
BMS-POS	30%	103,718	9,865 (9.51%)	0.00853	8,523 (8.22%)	0.00762	15.74%***
	40%	64,689	4,838 (7.48%)	0.00798	4,277 (6.61%)	0.00754	13.14%***
	50%	32,107	1,776 (5.53%)	0.00779	1,495 (4.66%)	0.00757	18.78%***
	60%	17,398	492 (2.83%)	0.00631	428 (2.46%)	0.00620	14.90%**
comScore2013	30%	689	91 (13.27%)	0.01638	54 (7.9%)	0.01303	68.01%***
	40%	494	56 (11.38%)	0.01437	29 (5.87%)	0.01282	93.79%***
	50%	291	25 (8.72%)	0.01360	14 (4.67%)	0.01286	86.76%**
	60%	129	10 (7.45%)	0.01273	6 (4.5%)	0.01226	65.52%

***: Significant at 1% level **: Significant at 5% level

Table 5: MLR vs. Single Rules: MLR uses multiple rules

The first column of Table 5 identifies the dataset. The second column shows the confidence threshold used for mining. The third column shows the numbers of baskets where MLR recommended items using multiple rules. The fourth and fifth columns show the number (and percentage) of successful recommendations with MLR, and the average time taken; the sixth and seventh columns present similar information for the single rule approach. The last column shows the percentage improvement using MLR.

Table 5 shows that MLR performs substantially better than the single rule approach when items are recommended using multiple rules. The improvements in performances by using MLR are statistically significant at the 1% level or better in most of the experiments. While the absolute improvements may appear small, the relative improvements are substantial – given the very large number of recommendations that are typically made every day, the net impact on revenues will be substantial as well. The time taken to make recommendations are in the milliseconds for both approaches – as these times are possible even on the basic desktop machines we have used to conduct the experiments, making recommendations using either of these approaches will not have any perceptible impact on the load times

of web pages. In addition, the standard deviations are also very low –in the worst case they are 0.00229 for *Retail*, 0.00853 for *BMS-POS*, and 0.01638 for *comScore2013*. This makes MLR a very viable approach for real-time situations.

For a given dataset and support threshold, there are fewer eligible rules for each consequent when a higher confidence threshold is used for mining. Consequently MLR requires less time to recommend items as the confidence threshold increases. Similarly, the average times required by the two approaches are highest for *comScore2013* and least for *Retail*, as a result of the difference in the number of rules mined.

When MLR recommends items using multiple rules, the rules used cover a much larger proportion of items in the baskets compared to the coverages of the rules used by the single rule approach. In Table 6, we show the average percentages of items in the baskets covered by the two approaches for each dataset. Combining the results in Tables 5 and 6, it is clear that increasing the coverage of items in the baskets substantively improves the performance of the recommender system. Since all the items in these baskets rarely co-occur simultaneously in transactions, they do not appear as antecedents of any rule. The rules that exist – which get used by the benchmark – cover a relatively small percentage of items in the baskets. By combining rules, MLR is able to improve the coverage and thereby perform better than the benchmark.

Dataset	Confidence Threshold	% of Items Covered when MLR Uses Multiple Rules	
		MLR	Single-Rule Approach
Retail	30%	56.26%	26.95%
	40%	56.76%	26.83%
	50%	56.46%	26.67%
	60%	46.84%	23.42%
BMS-POS	30%	85.90%	53.22%
	40%	78.01%	48.34%
	50%	76.86%	44.84%
	60%	83.70%	45.59%
comScore2013	30%	98.09%	65.38%
	40%	97.06%	63.25%
	50%	95.53%	60.23%
	60%	93.68%	57.45%

Table 6: Fraction of basket covered when MLR uses multiple rules

The coverage of items is smallest in *Retail* and largest in *comScore2013* when either of the approaches is used. This is again a direct result of dataset density - the rules generated from *Retail* have only a few items, while those from *comScore2013* have many more relative to the number of items in the

dataset. Even when MLR combines rules, only about half the items in the baskets are covered in the case of *Retail*, whereas more than 90% of the baskets are covered in the case of *comScore2013*.

4.4.2 MLR Uses One Rule

As mentioned earlier, MLR may provide a recommendation using a probability model consisting of a single rule. For completeness, we present the results for instances when MLR uses single rules in Table 7. Given that both approaches recommend the same item often, the results are as expected – the qualities of the recommendations provided by the approaches are quite similar.

Dataset	Conf. thres-hold	Baskets where MLR used 1 Rule	MLR		Single-Rule Approach (Zaïane 2002)		Improve ment (%)
			Accuracy # (%)	Time (secs)	Accuracy # (%)	Time (secs)	
Retail	30%	35,278	12,058 (34.18%)	0.00036	12,054 (34.17%)	0.00039	0.04%
	40%	30,705	11,862 (38.63%)	0.00036	11,856 (38.61%)	0.00038	0.05%
	50%	28,966	11,852 (40.92%)	0.00033	11,845 (40.89%)	0.00035	0.06%
	60%	27,904	10,693 (38.32%)	0.00034	10,690 (38.31%)	0.00035	0.03%
BMS-POS	30%	199,542	86,977 (43.59%)	0.00563	86,968 (43.58%)	0.00509	0.01%
	40%	213,461	88,707 (41.56%)	0.00582	88,657 (41.53%)	0.00531	0.06%
	50%	201,471	84,883 (42.13%)	0.00600	84,765 (42.07%)	0.00579	0.14%
	60%	163,312	76,400 (46.78%)	0.00561	76,279 (46.71%)	0.00529	0.16%
comScore 2013	30%	6,457	2365 (36.63%)	0.00968	2362 (36.58%)	0.00944	0.14%
	40%	6,087	2287 (37.58%)	0.00977	2286 (37.56%)	0.00961	0.04%
	50%	5,473	2089 (38.17%)	0.00970	2086 (38.12%)	0.00964	0.13%
	60%	3,921	1666 (42.48%)	0.00998	1661 (42.36%)	0.00990	0.28%

Table 7: MLR vs. Single Rules: MLR also uses single rules

The performances of both approaches are much better when MLR recommends items using single rules (the fourth column of Table 7) as compared to when it recommends items using multiple rules (the fourth column of Table 5). This is because the baskets for which MLR makes recommendations using multiple rules are typically larger (than baskets for which single rules are used), and the items in the baskets for which MLR makes recommendations using multiple rules co-occur less frequently in transactions. Therefore, it is less likely that there would be reliable probability estimates for many of the potential target items given the entire basket. As a result, it is far more difficult to recommend items that are likely to be purchased in the former case than in the latter; this difficulty is reflected in the marked differences in successful recommendations. When single rules are used for recommending items for the baskets included in Table 5, recommendations are significantly worse than when MLR is used. Also, as expected, MLR requires less time to recommend items when using single rules (the fifth column of Table 7) than when

using multiple rules (the fifth column of Table 5) because the available numbers of eligible rules are fewer in the former case.

Table 8 shows the percentages of items in the baskets covered by the rules used by the two approaches.

Dataset	Confidence Threshold	% of Items Covered When MLR Uses Single Rules	
		MLR	Single-Rule Benchmark
Retail	30%	60.58%	58.86%
	40%	62.51%	60.71%
	50%	63.67%	61.88%
	60%	56.48%	54.86%
BMS-POS	30%	93.91%	93.08%
	40%	88.13%	87.18%
	50%	82.52%	81.56%
	60%	81.49%	80.76%
comScore2013	30%	99.14%	98.42%
	40%	98.34%	97.58%
	50%	97.05%	96.06%
	60%	94.59%	93.12%

Table 8: Fraction of basket covered when MLR uses single rules

It is clear that the percentages are very close, and furthermore, the majority of items in the baskets are covered. Hence, the accuracies of both the approaches are similar. The effect of density is clear here as well – the fraction of the baskets covered is lowest for *Retail* and highest for *comScore2013* irrespective of the approach used.

4.4.3 Experiments with Different Supports

We perform additional experiments on *Retail* at support thresholds 0.1% and 0.3% to analyze the robustness of the two approaches to changes in the support threshold. The results are shown in Table 9. We present results only for those instances where MLR recommends items using multiple rules, as the performances of MLR and the benchmark are again quite similar for the other instances.

MLR performs better when items are recommended using multiple rules regardless of the support and confidence thresholds used for mining. The improvements are statistically significant, with an exception only when rules mined at 60% confidence thresholds are used. The improvement in performance is smaller when the rules mined at higher support thresholds are used. For example, the improvement achieved by using MLR is 23.25% when rules mined at support and confidence thresholds of 0.1% and 30%, respectively, are used, compared to 14.83%, when rules mined at 0.3% support threshold and 30%

confidence threshold are used. One possible reason could be the more frequent use of rules with higher supports by the single rule approach when rules mined at higher support thresholds are used. Rules with higher supports are more reliable. Hence, the scope for improvement by using MLR is smaller.

Sup. threshold	Conf. threshold	Baskets where MLR used ≥ 2 Rules	MLR		Single-Rule Approach		Improvement (%)
			Accuracy # (%)	Time (sec)	Accuracy # (%)	Time (sec)	
0.1%	30%	24,616	1141 (4.63%)	0.013	926 (3.76%)	0.004	23.25%***
	40%	22,818	1106 (4.85%)	0.012	901 (3.95%)	0.004	22.80%***
	50%	22,212	1006 (4.53%)	0.012	818 (3.68%)	0.004	22.99%***
	60%	14,380	340 (2.37%)	0.010	298 (2.08%)	0.004	14.08%*
0.3%	30%	12,494	780 (6.25%)	0.002	680 (5.44%)	0.001	14.83%***
	40%	12,020	780 (6.49%)	0.002	679 (5.65%)	0.001	14.87%***
	50%	11,884	761 (6.40%)	0.002	661 (5.56%)	0.001	15.16%*
	60%	6,467	171 (2.64%)	0.002	159 (2.46%)	0.001	7.28%

***: significant at 1% level or better, *: significant at 10% level

Table 9: Multiple support levels on *Retail*: MLR uses multiple rules

As evident from Table 9, the performance of association rule based recommender systems varies with the support and confidence thresholds used for mining the rules. No established theoretical basis exists for the selection of appropriate thresholds (Goh and Ang 2007). The thresholds to use depend on the application characteristics (e.g., the data density, the number of items being sold, the size of the database, etc.). They can be empirically determined by examining the performances of rules mined from representative historical data with different sets of thresholds (Liu and Hsu 2005, Goh and Ang 2007, Witten et al. 2011). Domain experts can also help determine acceptable thresholds (Schiaffino and Amandi 2005).

4.4.4 Rules Mined Using Lift and Leverage instead of Confidence

Although confidence[13] is the most widely used metric for rule generation, other metrics like lift and leverage are also used occasionally. Given a rule $\{\mathrm{A}\}\rightarrow\{\mathrm{X}\}$, lift is defined as the ratio of the confidence of the rule to the support of its consequent, i.e., $\left(\frac{P(\mathrm{A}\ and\ \mathrm{X})}{P(\mathrm{A})P(\mathrm{X})}\right)$. Lift measures how much greater is the probability that A and X occur together relative to if they had been independent. Leverage is the difference between the actual frequency of co-occurrence of A and X and the expected frequency if A and X were independent, i.e., leverage is $P(\mathrm{A}\ and\ \mathrm{X}) - P(\mathrm{A}) * P(\mathrm{X})$. In a sales setting, this would translate to the number of extra items

[13] Support is always used to filter reliable rules along with confidence, lift or leverage.

sold than the number expected under independence. We conducted additional experiments to assess the performance of MLR when rules mined using lift and leverage are used for making recommendations.

We first compared recommendation accuracies using each of the three metrics (confidence, lift and leverage). We found confidence based rules to perform significantly better than lift based rules on all three datasets. We know from Table 5 that MLR significantly outperforms the single rules based approach when rules generated using confidence are used. Therefore MLR using rules based on confidence clearly dominate rules generated using lift. While confidence based rules significantly outperform leverage-based rules on *Retail*, the improvements are not significant on *BMS-POS* and *comScore2013*. We then compared the performance of MLR with the single rule approach when the rules available are mined using leverage. As was the case earlier (with confidence based rules), we find that using MLR on leverage based rules significantly outperforms the (leverage based) single rules approach. Therefore, MLR is preferable over all the single rules based approaches.

4.4.5 Recommending Multiple Items

We also conduct experiments where two items are recommended for each basket by both approaches. We consider a recommendation to be successful when at least one of the two recommended items is present in the remainder of the transaction. *BMS-POS* is used for the experiments and the rules are mined at a support threshold of 0.2% and confidence thresholds of 15%, 20%, 25%, 30% and 35%, respectively. The reason for using rules mined at these thresholds is that the average number of items that can be recommended per basket is four or more when rules mined at these thresholds are used.

Confidence	# of baskets	MLR Accuracy #(%)	Single Rule Accuracy #(%)	Improvement (%)
15%	43,360	4,407 (10.16%)	3,569 (8.23%)	23.49%***
20%	41,916	3,851 (9.19%)	3,236 (7.72%)	18.99%***
25%	38,338	3,137 (8.18%)	2,745 (7.16%)	14.26%***
30%	30,474	2,086 (6.84%)	1,891 (6.20%)	10.31%***
35%	19,886	1,080 (5.43%)	1,002 (5.04%)	7.70%

***: significant at 1% level or better

Table 10: MLR vs. Single Rule when two items are recommended (*BMS-POS*)

Table 10 shows the results of the experiments for those baskets for which MLR recommends both items using multiple rules; the performance improvement is significant. The differences in the performances of the two approaches are not significant when either one or both items are recommended by MLR using single rules. The times taken to make the multiple recommendations are virtually identical to that when single items are recommended (i.e., as shown in Table 5).

4.5 MLR vs Rule-Combination Approaches

Various experiments were conducted comparing MLR with the rule combination approach of Lin et al. (2002) and the CMAR approach of Li et al. (2001). The improvement from using MLR compared to the approach of Lin et al. (2002) was more than the improvement over CMAR. Therefore, we only report results comparing MLR with CMAR.

CMAR was developed for classification, and therefore we had to adapt it to work in a product recommendation context (as discussed for L3G). CMAR works as follows. When a basket is provided to CMAR, it evaluates the sums of the weighted chi-squares of the rules in the individual consequent sets, and the item corresponding to the consequent set with the highest sum is recommended. The chi-square of a rule indicates the correlation between the items in the rule. Rules in a consequent set with the highest sum of weighted chi-square have the highest correlation with each other, and the consequent corresponding to that consequent set is expected to have the highest probability of occurrence (Li et al. 2001). The sum of the weighted chi-squares of the rules in a consequent is $\sum \frac{\chi^2 \chi^2}{max\chi^2}$, where χ^2 is the chi-square statistic of a rule with antecedent P and consequent c, and maxχ^2 is evaluated as:

$$max\chi^2 = \left(\min\{\sup(P), \sup(c)\} - \frac{\sup(P)\sup(c)}{|T|} \right)^2 |T| e$$

where $e = \frac{1}{sup(P)\sup(c)} + \frac{1}{\sup(P)(|T|-\sup(c))} + \frac{1}{(|T|-\sup(P))\sup(c)} + \frac{1}{(|T|-\sup(P))(|T|-\sup(c))}$,

sup(P) = number of transactions with items in P,

sup(c) = number of transactions with consequent c, and

|T| = total number of transactions.

Readers are referred to Li et al. (2001) for additional details of their approach.

As with the previous comparison, experiments are first performed using rules mined at a support threshold of 0.2% and confidence thresholds of 30%, 40%, 50% and 60% on all the datasets. Table 11 shows the results of the experiments over all the baskets.[14] Each individual recommendation is made within a fraction of a second by both approaches. For the *Retail* and *BMS-POS* datasets, the improvements in the performances achieved by using MLR are statistically significant except when rules mined at 60% confidence threshold are used. In the case of *comScore2013*, while MLR consistently performs better than CMAR, the improvements achieved by MLR are not statistically significant.

[14] We provide results aggregated over all the baskets here because, unlike in previous experiments, both MLR and CMAR use multiple rules whenever possible.

	Confidence threshold	# of Baskets	MLR Accuracy #(%)	CMAR Accuracy #(%)	Improvement (%)
Retail	30%	51,417	12,058 (23.45%)	11,635 (22.63%)	3.64%***
	40%	45,909	11,840 (25.79%)	11,515 (25.08%)	2.82%**
	50%	43,974	11,804 (26.84%)	11,498 (26.15%)	2.66%**
	60%	36,751	10,506 (28.59%)	10,439 (28.40%)	0.65%
BMS-POS	30%	59,111	17,727 (29.99%)	15,617 (26.42%)	13.51%***
	40%	54,366	17,317 (31.85%)	16,958 (31.19%)	2.12%**
	50%	45,671	16,153 (35.37%)	15,753 (34.49%)	2.54%***
	60%	35,352	14,547 (41.15%)	14,347 (40.58%	1.40%
comScore2013	30%	6,882	2,203 (32.00%)	2,152 (31.27%)	2.36%
	40%	6,346	2,102 (33.12%)	2,055 (32.38%)	2.27%
	50%	5,576	1,911 (34.27%)	1,890 (33.89%)	1.10%
	60%	3,937	1,554 (39.46%)	1,539 (39.08%)	0.96%

***: significant at 1% level or better **: significant at 5% level

Table 11: MLR vs. CMAR

We conducted additional experiments on the *Retail* dataset using rules mined at support thresholds of 0.1% and 0.3%. The relative performances of the two approaches are very similar for those experiments as well, and are not reported here for brevity.

5. Comparisons with Collaborative Filtering and Matrix Factorization

While rule based recommender systems are commonly used in the retail domain (e.g., Forsblom et al. 2009) and form integral components of many commercial software (IBM 2009a, 2009b, 2010), no individual system has been found to be universally better than others. In this section, we provide evidence of the broad applicability of MLR by comparing it to two state-of-the-art techniques – collaborative filtering and matrix factorization. Both approaches are widely used for providing recommendations and have been shown to perform well in general (Linden 2003, Deshpande and Karypis 2004, Koren et al. 2009, Ekstrand et al. 2011). We present results from several experiments conducted on the three datasets, comparing the quality of recommendations from MLR with those generated using these approaches.

While two approaches to collaborative filtering are popular, Jannach et al. (2011) point out that the need to handle millions of users in large e-commerce systems makes user-based collaborative filtering impractical in real time environments. Item-based collaborative filtering on the other hand makes predictions based on the similarity between items. These can be computed offline, which makes item-based collaborative filtering a viable approach for making real-time recommendations. Also, item-based collaborative filtering is designed to generate recommendations using transactional data (Linden et al. 2003). Therefore, we use item-to-item collaborative filtering in our experiments.

The Netflix Prize competition revealed that matrix factorization methods can also be very effective in making recommendations. These methods represent users and items via latent factors identified from the data, with an item being recommended to a user when the item and user are similar vis-à-vis these factors (Koren et al. 2009). A well-known matrix factorization technique for recommender systems is singular value decomposition (SVD), and one version of which, called FunkSVD, has been popularized by Simon Funk (2006). We use FunkSVD (Funk 2006) in our experiments.

We use the collaborative filtering and FunkSVD implementations provided by Ekstrand et al. (2011) in an open source project named *Lenskit* (lenskit.grouplens.org). As noted by the authors, Lenskit provides carefully tuned implementations of these leading algorithms (all the implementations are in Java). We note that in their experiments on three separate datasets, Ekstrand et al. (2011) find FunkSVD to perform the best on two datasets and the item-to-item collaborative filtering approach to perform the best on the third. We provide brief descriptions of the two methods below; specific details about the implementations can be found in Ekstrand et al. (2011).[15]

The idea behind the item-based approach is to find items that are rated as similar to the items that have been liked by a target user. Given a dataset involving m items, the item-based collaborative filtering procedure implemented by Ekstrand et al. requires two parameters as inputs – a *model size* (k) and a *neighborhood size* (l). The system computes scores for the items being considered for recommendation by multiplying an $m \times m$ similarity matrix (model) with a column vector representing the current basket of the user (the vector has a 1 for all items present in the basket and a 0 for the other items). The model size is the number of similarities retained in each column of the model; other similarities are set to 0. The neighborhood size is the number of similarities used to calculate the score of an item; other similarities are ignored. The item with the highest score is recommended.

The matrix factorization technique determines latent factors, associates each user with a user-factor vector and each item with an item-factor vector, and makes predictions using the inner product of such vectors. The parameters of the model are learned with the objective of minimizing the differences between predicted and actual ratings while avoiding over-fitting (Koren et al. 2009). FunkSVD accomplishes this using a stochastic gradient descent learning algorithm.

We perform, as before, five-fold cross validation experiments on all the datasets. We use a support threshold of 0.2% and confidence thresholds of 30%, 40%, 50% and 60% for MLR. We experimented with various values of model sizes (up to 500) and neighborhood sizes (up to 150) for collaborative filtering.

[15] Interested readers are directed to Deshpande and Karypis (2004) for additional details on collaborative filtering, and to Funk (2006) and Koren et al. (2009) for details on FunkSVD.

While FunkSVD was originally designed for ratings of user-item pairs (like any matrix factorization technique), it has been observed to work well for binary data if all the zero values are replaced with a small number like 0.1 (XLVector 2012). Therefore, we modify the datasets in this manner to run FunkSVD. The resulting datasets are completely dense; in fact those corresponding to *Retail* and *BMS-POS* cannot be used in their entirety by Lenskit. Therefore, for each original training dataset from *Retail*, we randomly select 2,000 transactions for model building. We are able to use 10,000 transactions (again randomly selected) as the training datasets for *BMS-POS*, because this dataset has much fewer items than *Retail*. We experimented with other numbers of randomly selected transactions for creating ratings datasets – the results do not differ significantly.[16] We were able to use all the transactions in *comScore2013* as the training datasets are relatively small. The modified datasets for *Retail* and *BMP-POS* have more than ten million values for user-item pairs (the largest dataset used by Ekstrand et al. has ten million values), whereas the modified datasets for *comScore2013* have more than a million user-item pairs.

In these experiments, the training datasets are used to create the models for the non-rule based systems. The basket creation scheme is slightly different from the one described in Section 4. In the interest of time, the baskets are created from each transaction in the test datasets by randomly selecting half the items in the transaction – the test baskets are identical for all the approaches. This does not affect the results significantly as we still get enough instances (baskets) to draw statistically reliable conclusions. The results presented are for those transactions for which all the approaches generate recommendations and are averaged over all cross-validation experiments. The performance of the collaborative filtering system is not very sensitive to changes in the model and neighborhood sizes. We report the results for the experiments with model size 500 and neighborhood size 150 since the system's performance is a little better with these settings. For FunkSVD, we use the default settings of the implementation – other settings provided similar or inferior performances. Table 12 presents the quality of recommendations provided by each approach for each dataset and confidence threshold.

The relative improvements achieved by MLR over collaborative filtering are statistically significant for every experiment on each of the datasets. MLR consistently outperforms FunkSVD as well, although the improvements in the case of *comScore2013* are not statistically significant. These results show that MLR is not only superior to other rule-based approaches for the datasets we have examined, but outperforms other state-of-the-art approaches for our datasets as well.

[16] We also experimented with randomly assigning a percentage (5%, 10%, 15%, and 25%) of unpurchased items to have the value 0.1 and leaving the rest of the unpurchased items as unrated on all the transactions in the dataset (i.e., no sampling). The results remained very similar with no qualitative differences.

Dataset	Confidence Threshold	Accuracy (%)			Improvement over CF (%)	Improvement over FunkSVD (%)
		CF	FunkSVD	MLR		
Retail	30%	32.90%	43.91%	47.27%	43.65%***	7.64%***
	40%	33.85%	45.01%	48.41%	43.00%***	7.55%***
	50%	34.42%	45.44%	48.87%	41.98%***	7.55%***
	60%	36.10%	47.64%	52.12%	44.40%***	9.41%***
BMS-POS	30%	45.21%	47.30%	50.07%	10.76%***	5.86%***
	40%	48.16%	50.43%	53.48%	11.04%***	6.05%***
	50%	50.21%	52.74%	55.71%	10.96%***	5.64%***
	60%	52.18%	55.24%	57.93%	11.02%***	4.86%***
comScore2013	30%	33.82%	36.03%	36.39%	7.60%**	0.99%
	40%	35.69%	38.18%	38.69%	8.41%**	1.36%
	50%	37.31%	39.93%	40.55%	8.70%**	1.56%
	60%	43.33%	46.22%	47.11%	8.73%**	1.92%

***: significant at 1% level or better, **: significant at 5% level or better

Table 12: MLR vs. Collaborative filtering (CF) and FunkSVD

As an additional robustness check, we conducted experiments where we included in each basket all but one randomly selected item from a transaction, and provided such baskets to MLR and the benchmark systems. MLR again performs significantly better than both collaborative filtering and matrix factorization in these experiments on the *Retail* and *BMS-POS* datasets. While MLR performs better than collaborative filtering and FunkSVD on *comScore2013*, the improvements are not statistically significant.

6. Conclusions and Managerial Implications

Traditional approaches that use only a single rule for recommending items typically ignore items in the baskets of the customer that may be present in the antecedents of other rules. We propose an approach – *Maximum Likelihood Recommendations* (MLR) – to combine multiple rules to recommend items in order to cover as many items of the basket as possible. While a few methods have been proposed to combine rules, these are all ad hoc, without a robust theoretical basis. In contrast, MLR has a strong theoretical foundation – it recommends items using rules that maximize the likelihood of generating the true underlying distribution of the dataset used for generating the rules. This process identifies the best set of rules to combine when estimating the probability that a customer will add a recommended item to her basket. Our approach tries to preserve as many important dependencies as possible based on the available rules (this typically leads to making as few conditional independence assumptions as possible across items in a basket).

It is not practical to solve the problem of maximizing likelihood directly however, as it requires the estimation of parameters from the dataset during run-time. We show that maximizing the likelihood is equivalent to maximizing the sum of the mutual information values of the participating rules – this result makes the real-time use of this approach feasible.

We conduct extensive experiments to test the viability of the proposed approach. Comparisons are made with several traditional single-rule based approaches, two methods that have been proposed to combine rules, and two other state-of-the-art recommendation approaches – collaborative filtering and matrix factorization. The experiments show that MLR consistently outperforms all the other approaches, particularly when rules are available for combination. We also find that the performance improvements are robust across datasets at various support and confidence thresholds.

While the absolute improvements may seem small at first glance, it is important to remember that recommendations are made on a continuous basis. For instance, during the holiday season in 2012, Amazon.com sold 306 items per second (Clark 2012). The 2012 annual report for Amazon.com mentions that their net sales for the fourth quarter of 2012 amounted to $21.27 billion. If 35% of these revenues are generated from recommendations (as mentioned in Hosanagar et al. 2014), even a 1% increase in success rate would amount to an increase in revenues of approximately $70 million every quarter. Such improvements can lead to increasing revenues by millions of dollars every year for smaller firms as well.

Several characteristics of MLR make it suitable for firms that provide personalized recommendations to their online customers. The use of probability calculus provides semantic clarity in an environment that is naturally fraught with uncertainty – at the same time, the approach is able to deliver recommendations effectively. The robust theoretical basis of MLR makes it very versatile. Because it compares alternative items to recommend based on their probabilities of purchase, it can be easily adapted to make recommendations based on expected payoffs associated with the items. This is not possible with any of the existing approaches, i.e., rule combination, collaborative filtering, or matrix factorization. The use of probability theory also allows MLR to be easily adapted to use lift-based approaches to make recommendations if needed – again this is not possible with any of the other approaches. While we use association rules mined from historical data in our experiments, the approach can easily accommodate rules provided by human experts. Such rules may be available from marketing experts for new items that are being offered and for which transactional data are not yet available. MLR can also use association rules with negations if such rules are found to improve the quality of recommendations. MLR is computationally quite efficient, taking only a fraction of a second per recommendation on average – further, this is accomplished using a simple desktop computing environment. As a result, using MLR instead of the single-rule based approach should not affect the quality of service during regular operation in commercial environments.

While the results of our experiments with MLR are very encouraging, we note that the performances of different approaches can be sensitive to the application domain and data characteristics. For example, our results are consistent with those of Mobasher et al. (2001) with regard to collaborative filtering – they also found an association rule based approach to outperform a collaborative filtering based one. However, Sarwar et al. (2000) found evidence to the contrary. Similarly, FunkSVD is shown to perform better than item-based collaborative filtering on two MovieLens datasets but worse on a Yahoo! Music dataset (Ekstrand et al. 2011). Given the differences in performances of alternative approaches for different application domains, firms would be prudent to evaluate the different types of available approaches in order to identify the best one for their specific context.

Our work opens up several avenues for future research. MLR can serve as a valuable new method to consider for ensemble-based approaches as its theoretical basis is quite different (and therefore independent) from those of memory-based approaches such as collaborative filtering and matrix factorization. It would be useful to develop ways to combine MLR with other extant approaches to determine which techniques best complement each other. Another interesting opportunity is to examine how MLR could be extended to incorporate probabilistic context-based approaches that account for item metadata, user demographics, etc. An important issue that has emerged in recent years is the extent to which a recommendation technique is vulnerable to manipulations that are often referred to as shilling attacks (Mobasher et al. 2007). It will be useful to examine in future research how robust MLR is to such attacks, as compared to extant techniques. Finally, it would be useful to extend recommendation query languages like REQUEST (Adomavicius et al. 2011) by incorporating a probability-based query interface that will allow end users to generate recommendations in a flexible and user-friendly manner.

Acknowledgements

The authors would like to thank Michael Ekstrand for his considerable help in using *Lenskit* for the experiments that involved collaborative filtering and matrix factorization techniques. They would also like to thank the Senior Editor Ram Gopal, the Associate Editor Gautam Pant, and the anonymous reviewers for the detailed and constructive comments and suggestions that have helped improve the paper.

References

1. Aalto. 2014. Available at https://noppa.aalto.fi/noppa/kurssi/s-114.1310/luennot/extramaterial.pdf, accessed on March 15, 2014.

2. Adomavicius, G. Tuzhilin, A., and Zheng, R. 2011. "REQUEST: A Query Language for Customizing Recommendations," *Information Systems Research***,** 22(1), pp. 99-117.

3. Agrawal, R., Imielinski, T., and Swami, A., 1993. "Mining Association Rules between Sets of Items in Large Databases," *Proc. of the ACM SIGMOD Conference on Management of Data*, Washington, D.C., pp. 207-216.

4. Balas, E. and Padberg, M, 1976. "Set Partitioning: A Survey," *SIAM Review* (18), pp. 710–760.

5. Baralis, E., Chiusano, S., and Graza, P. 2004. "On Support Thresholds in Associative Classification," *Proceedings of the 2004 ACM Symposium on Applied Computing*, pp. 554-558.

6. Baralis, E. and Garza, P. 2002. "A Lazy Approach to Pruning Classification Rules," *Proceedings of 2002 IEEE International Conference on Data Mining*, pp. 35-42.

7. Bayardo, R. J. 1998. "Efficiently mining long patterns from databases," *Proceedings of the 1998 ACM-SIGMOD International Conference on Management of Data*, pp. 85-93.

8. Bayardo, R. J. Jr., Agrawal, R., 1999. "Mining the Most Interesting Rules," *Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* pp.145-154, August 15-18, San Diego, California, United States.

9. Berry, M. and Linoff , G., 2004. "Data Mining and Techniques," 2nd Ed. *Wiley Computer Publishing*, New York.

10. Brijs, T., "Retail Market Basket Data Set," available at *http://fimi.cs.helsinki.fi/data/*, accessed on April 5, 2014.

11. Calders, T., Dexters N., Gillis, J. J.M., and Goethals, B. 2013. "Mining Frequent Itemsets in a Stream," *Information Systems*, (39), pp. 1-23.

12. Clark, K. 2012. "Amazon Has Best Holiday Season Ever, Selling 306 Items Per Second", Forbes.com, available at *http://www.forbes.com/sites/kellyclay/2012/12/27/amazon-has-best-holiday-season-ever-selling-306-items-per-second/* , accessed on April 5, 2014.

13. Deshpande, M. and Karypis, G. 2004. "Item-based Top N Recommendation Algorithms," *ACM Transactions on Information Systems*, 22(4), pp. 143-177.

14. Domingos, P. and Pazzani, M., 1997. "On the Optimality of the Simple Bayesian Classifier under Zero-One Loss," *Machine Learning* (29), pp. 103-130.

15. Drogan, M. and Hsu, J. 2003. "Enhancing the Web Customer's Experience: Techniques and Business Impacts of Web Personalization and Customization," *Proceedings of the Information Systems Education Conference*, 2003, pp. 1-16.

16. Ekstrand, M.D., Ludwig, M., Konstan, J.A. and Riedl, J.T. 2011. "Rethinking the Recommender Research Ecosystem: Reproducibility, Openness, and LensKit," *Proceedings of the fifth ACM conference on Recommender systems*, 2011, pp. 133-140.

17. Ergun O., Kuyzu, G. and Savelsbergh, M., 2007. "Reducing Truckload Transportation Costs Through Collaboration," *Transportation Science*, 41(2), pp. 206-221.

18. Forsblom, A., Nurmi, P., Floreen, P., Peltonen, P., and Saarikko, P. 2009. "Massive- An Intelligent Shopping Assistant," *Proceedings of the Workshop on Personalization in Mobile and Pervasive Computing*, Trento, Italy, 2009.

19. Funk, S. 2006. "Netflix Update: Try This at Home," available at *http://sifter.org/~simon/journal/20061211.html*, accessed on June 20, 2012.

20. Goh, D.H. and Ang, R.P. 2007. "An Introduction to Association Rule Mining: An Application in Counseling and Help-seeking Behavior of Adolescents," *Behavior Research Methods*, 39(2), pp. 259-266.

21. Gordon, L. 2008. "Leading Practices in Market Basket Analysis. How Top Retailers are Using Market Basket Analysis to Win Margin and Market Share," *Factpoint Group*. 2008.

22. Hanson, W., 2000. "Principles of Internet Marketing," *South-Western College Publishing*, Cincinnati, Ohio.

23. Hastie, T., Tibshirani, R., and Friedman, J., 2009. "The Elements of Statistical Learning," *Data Mining, Inference and Prediction*, 2nd Ed. Springer, New York.

24. Han, J., Kamber, M., and Pei, J. 2012. "Data Mining: Concepts and Techniques," 3rd Ed. *Morgan Kaufmann Publishers*.

25. Häubl, G. and Trifts, V., 2000. "Consumer Decision Making in Online Shopping Environments: The Effects of Interactive Decision Aids," *Marketing Science*, 19(1), pp. 4-21.

26. Hosanagar, K., Fleder, D.M., Lee, D., and Buja, A., 2014. "Will the Global Village Fracture Into Tribes: Recommender Systems and their Effects on Consumers," *Management Science*, 60(4), pp. 805-823..

27. IBM. 2009a. "IBM SPSS Retail Market Basket Analysis," available at *ftp://service.boulder.ibm.com/ software/uk/ data/ibm-spss-retail-datasheet.pdf*, accessed on April 11, 2012.

28. IBM. 2009b. "Retail Market Basket Analysis," available at *https://www-304.ibm.com/easyaccess/fileserve/ ?contentid=193973*, accessed on April 11, 2012.

29. IBM. 2010. "Predictive Analytics for Retail Market Basket Analysis", available at *ftp://public.dhe.ibm.com/ common/ssi/ecm/en/yts03013gben/YTS03013GBEN.PDF*, accessed on April 11, 2012.

30. Jannach, D., Zanker, M., Felfernig, A., and Friedrich, G. 2011. "Recommender Systems: An Introduction," *Cambridge University Press*, pp. 31-35.

31. Koren, Y., Bell, R., and Volinsky, C. 2009. "Matrix Factorization Techniques for Recommender Systems," *IEEE Computer*, 42(8), pp. 30-37.

32. Kullback, S., 1959. "Information Theory and Statistics," *Wiley*, New York.

33. Lewin, B.A. 2009. "Beyond The Grocery and Retail Store: Applying Market Basket Analysis to the Service Industry," *A 1010Data White Paper*. 2009.

34. Li, W., Han, J., and, Pei, J., 2001. "CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules," *Proceedings IEEE International Conference on Data Mining*, 2001, pp. 369-376.

35. Lin, W., Alvarez, S.A., and Ruiz, C. 2002. "Efficient Adaptive-Support Association Rule Mining for Recommender Systems," *Data Mining and Knowledge Discovery*, 6(1), pp. 83-105.

36. Liu, Y. and Hsu, P. 2005. "A New Approach to Generate Frequent Patterns from Enterprise Databases", *Third International Conference on Advances in Pattern Recognition*, ICAPR 2005.

37. Linden, G., Smith, B., and York, J. 2003. "Amazon.com Recommendations Item-to-Item Collaborative Filtering," *IEEE Internet Computing,* 7(1), pp. 76-80.

38. Liu, B., Ma, Y., and Wong, C.K., 2003. "Scoring the Data Using Association Rules," *Applied Intelligence*, 18, pp. 119–135.

39. Mobasher, B., Dai, H., Luo, T., and Nakagawa, M. 2001. "Effective Personalization Based on Association Rule Discovery from Web Usage Data," *Proceedings of the 3rd International Workshop on Web Information and Data Management, WIDM,* 2001, pp. 9-15.

40. Mobasher, B., Burke, R., Bhaumik, R., and Williams, C. 2007. "Towards Trustworthy Recommender Systems: An Analysis of Attack Models and Algorithm Robustness," *ACM Transactions on Internet Technology*. **7**(4), pp. 23-38.

41. Ng, R.T., Lakshmanan, L.V.S., Han, J., and Pang, A. 1998. "Exploratory Mining and Pruning Optimizations of Constrained Associations Rules," *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, pp.13-24, Seattle, Washington, United States.

42. Pathak, B., Garfinkel, R., Gopal, R., Venkatesan, R., and Yin, F. 2010. "Empirical Analysis of the Impact of Recommender Systems on Sales," *Journal of Management Information Systems*, 27(2), pp. 159-188.

43. Rosenberg M. 2001. "The Personalization Story," *IT World.com*, November 5, 2001.

44. Sarwar, B., Karypis, G., Konstan, J., and Riedl. ‚J., 2000. "Analysis of Recommendation Algorithms for E-Commerce," *EC'00*, Minneapolis, Minnesota.

45. Schiaffino, S and Amandi, A. 2006. "Personalizing User-Agent Interaction", *Knowledge-Based Systems*, 19, pp. 43-49.

46. Shmueli, G., Patel, N.R., and Bruce, P.C. 2010. "Data Mining for Business Intelligence," 2nd Ed., *John Wiley and Sons, Inc.*, Hoboken, NJ.

47. Su, X. and Khoshgoftaar, T. M. 2009. "A Survey of Collaborative Filtering Techniques," *Advances in Artificial Intelligence*, 2009, pp. 1-19.

48. Thabtah, F. A. 2007. "A Review of Associative Classification Mining," *Knowledge Engineering Review*, 22 (1), pp. 37-65.

49. Tam, K.Y. and Ho, S.Y., 2003. "Web Personalization: Is it Effective?," *IT Professional*, 5(5), pp. 53-57.

50. Wang, J. 2008. "Data Warehousing and Mining: Concepts, Methodologies, Tools and Application," *Information Science Reference*, 1st Edition, 2008, Volume 5, page 2605.

51. Wang, F.H. and Shao, H.M., 2004. “Effective Personalized Recommendation Based on Time-Framed Navigation Clustering and Association Mining,” *Expert Systems with Applications*, 27, pp. 365–377.

52. Watanabe, S. 1960. “Information Theoretical Analysis of Multivariate Correlation,” *IBM Journal of Research and Development*, 4, pp. 66–82.

53. Webb, G.I. 2000. “Efficient Search for Association Rules,” In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2000),* pp. 99-107.

54. Webb, G.I. 2008. “Layered Critical Values: A Powerful Direct-Adjustment Approach to Discovering Significant Patterns,” *Machine Learning*, 71, pp. 307–323.

55. Webb, G.I. 2010. “Self-Sufficient Itemsets: An Approach to Screening Potentially Interesting Associations between Items,” *Transactions on Knowledge Discovery from Data*, 4, pp. 3-20.

56. Webb, G.I. and Zhang, S. 2005. “k-Optimal Rule Discovery,” *Data Mining and Knowledge Discovery*, 10(1), pp. 39-79.

57. Wickramaratna, K., Kubat, M., and Premaratne, K. 2009. “Predicting Missing Items in Shopping Carts,” *IEEE Transactions on Knowledge and Data Engineering*, 21, pp. 985-998.

58. Witten, I. H., Frank, E. and Hall, M. A. 2011. “Data Mining – Practical Machine Learning Tools and Techniques”, *Third Edition*, *Morgan Kaufmann*, page 123.

59. XLVector. 2012. “XLVector – Recommender System,” available at *http://xlvector.net/blog/?p=465*, accessed on July 7 2012.

60. Zaïane, O.R., 2002. “Building a Recommender Agent for e-Learning Systems,” *Proceedings of the International Conference on Computers in Education*.

61. Zaki, M.J., 2000. “Generating Non-Redundant Association Rules,” *KDD '00: Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 34-43.

62. Zheng, Z., Kohavi, R., and Mason, L., 2001. “Real World Performance of Association Rule Algorithms,” *Proc. 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM Press, New York, pp. 401–406.

63. Zhou, C., Cule, B., and Goethals B. 2013. “Itemset Based Sequence Classification,” *In Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Data (ECML PKDD 2013)*, pp. 353-368.