Mining Vague Association Rules

An Lu, Yiping Ke, James Cheng, and Wilfred Ng

Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
Hong Kong, China
{anlu,keyiping,csjames,wilfred}@cse.ust.hk

Abstract. In many online shopping applications, traditional Association Rule (AR) mining has limitations as it only deals with the items that are sold but ignores the items that are *almost sold*. For example, those items that are put into the basket but not checked out. We say that those *almost sold* items carry *hesitation information* since customers are hesitating to buy them. The hesitation information of items is valuable knowledge for the design of good selling strategies. We apply vague set theory in the context of AR mining as to incorporate the hesitation information into the ARs. We define the concepts of attractiveness and hesitation of an item, which represent the overall information of a customer's intent on an item. Based on these two concepts, we propose the notion of Vague Association Rules (VARs) and devise an efficient algorithm to mine the VARs. Our experiments show that our algorithm is efficient and the VARs capture more specific and richer information than traditional ARs.

1 Introduction

Association Rule (*AR*) mining [1] is one of the most important data mining tasks. Traditional AR mining has been extensively studied for over a decade; however, in recent years, the emergence of many new application domains, such as the Web, has led to many possibilities and challenges of studying new forms of ARs.

Consider the classical market basket case, in which AR mining is conducted on transactions that consist of items bought by customers. However, there are also many items that are not bought but customers may have considered buying them. We call such information on a customer's consideration to buy an item the *hesitation* information of the item, since the customer is hesitating whether to buy it. The hesitation information of an item is useful knowledge for boosting the sales of the item. However, such information is not considered in traditional AR mining due to the difficulty to collect hesitation information in the past. Nevertheless, with the advance in Web technology, it is now much easier to obtain the hesitation information of the items. Consider an online shopping scenario, such as "Amazon.com", it is possible to collect huge amount of data from the Web log that can be considered as hesitation information. For example, in the online shopping scenario: (1) the items that customers put into their online shopping carts but were not checked out eventually; (2) the items that are in customers' favorite lists to buy next time; (3) the items that are in customers' wishing lists but not yet available in the store; and so on. The hesitation information can then be used to design and implement selling strategies that can potentially turn those "under consideration" items into "well sold" items.

We apply the *vague set theory* [2] as a basis to model the hesitation information of the items. Vague set theory addresses the drawback of a single membership value in *fuzzy set theory* [3] by using interval-based membership that captures three types of evidence with respect to an object in a universe of discourse: *support*, *against* and *hesitation*. Thus, we can naturally model the hesitation information of an item in the mining context as the evidence of hesitation with respect to the item. The information of the "sold" items and the "not sold" items (without any hesitation information) in the traditional setting of AR mining correspond to the evidence of support and against with respect to the item.

To study the relationship between the support evidence and the hesitation evidence with respect to an item, we propose the concepts of *attractiveness* and *hesitation* of an item, which are based on the *median membership* and the *imprecision membership* [4, 5] that are derived from the vague membership in vague sets. An item with high attractiveness means that the item is well sold and has a high possibility to be sold again next time. An item with high hesitation means that customers are always hesitating to buy the item due to some reason (e.g., the customer is waiting for price reduction) but has a high possibility to buy it next time, if the reason of giving up the item is identified and resolved (e.g., some promotion on the item is provided).

Using the notions of attractiveness and hesitation of items, we model a database with hesitation information as an AH-pair database that consists of AH-pair transactions, where A stands for attractiveness and H stands for hesitation. Based on the AH-pair database, we then propose the notion of *Vague Association Rules* (*VARs*), which capture four types of relationships between two sets of items: the implication of the attractiveness/hesitation of one set of items on the attractiveness/hesitation of the other set of items. To evaluate the quality of the different types of VARs, four types of support and confidence are defined. We also investigate the properties of the support and confidence of VARs, which can be used to speed up the mining process. Based on these properties, an efficient algorithm is then designed to mine the VARs.

Our experiments on both real and synthetic datasets verify that our algorithm to mine the VARs is efficient. Compared with the traditional ARs mined from transactional databases, the VARs mined from the AH-pair databases, which are modelled from transactional databases by taking into account the hesitation information of items, are more specific and are able to capture richer information. More importantly, we find that, by aggregating more transactions into an AH-pair transaction, our algorithm is significantly more efficient while still obtaining almost the same set of VARs.

Organization. This paper is organized as follows. Section 2 presents the VARs and defines related concepts. Section 3 discusses the algorithm to mine the VARs. Section 4 reports the experimental results and Section 5 offers the conclusions.

2 Vague Association Rules

In this section, we define the notion of *Vague Association Rules* (*VARs*) and four types of support and confidence used to evaluate the quality of the VARs. We then present some properties of VARs that can be used to speed up the process of mining VARs.

Given the transactions of the customers, we then aggregate the transactions to obtain the *intent* of each item. Based on the intent of an item, we next define the *attractiveness* and *hesitation* of it.

Definition 1. (Intent, Attractiveness and Hesitation, *AH*-Pair Transactions) *The intent of an item x, denoted as intent(x), is a vague value $[\alpha(x), 1-\beta(x)]$. The attractiveness of x, denoted as $M_A(x)$, is defined as the median membership of x, i.e., $M_A(x) = (\alpha(x) + (1-\beta(x)))/2$. The hesitation of x, denoted as $M_H(x)$, is defined as the imprecision membership of x, i.e., $M_H(x) = ((1-\beta(x)) - \alpha(x))$. The pair $\langle M_A(x), M_H(x)\rangle$ is called the AH-pair of x. An* AH-pair transaction T *is a tuple $<v_1, v_2, \ldots, v_m>$ on an itemset $I_T = \{x_1, x_2, \ldots, x_m\}$, where $I_T \subseteq I$ and $v_j = \langle M_A(x_j), M_H(x_j)\rangle$ is an AH-pair of the item x_j, for $1 \leq j \leq m$. An* AH-pair database *is a sequence of AH-pair transactions.*

We now present the notion of VARs and define the support and confidence of a VAR.

Definition 2. (Vague Association Rule) *A* Vague Association Rule (VAR), *$r = (X \Rightarrow Y)$, is an association rule obtained from an AH-pair database.*

Based on the attractiveness and hesitation of an item, we define four different types of support and confidence of a VAR depending on what kind of knowledge we want to acquire. For clarity, we use A to denote *Attractiveness* and H to denote *Hesitation*.

Definition 3. (Support) *Given an AH-pair database, D, we define four types of support for an itemset Z or a VAR $X \Rightarrow Y$, where $X \cup Y = Z$, as follows.*

1. *The A-support of Z, denoted as Asupp(Z), is defined as $\sum_{T \in D} \prod_{z \in Z} M_A(z)/|D|$.*
2. *The H-support of Z, denoted as Hsupp(Z), is defined as $\sum_{T \in D} \prod_{z \in Z} M_H(z)/|D|$.*
3. *The AH-support of Z, denoted as AHsupp(Z), is defined as $\sum_{T \in D} \prod_{x \in X, y \in Y} M_A(x) M_H(y)/|D|$.*
4. *The HA-support of Z, denoted as HAsupp(Z), is defined as $\sum_{T \in D} \prod_{x \in X, y \in Y} M_H(x) M_A(y)/|D|$.*

Z is an A (or H or AH or HA) FI if the A- (or H- or AH- or HA-) support of Z is no less than the (respective A or H or AH or HA) minimum support threshold σ.

Definition 4. (Confidence) *Given an AH-pair database, D, we define the confidence of a VAR, $r = (X \Rightarrow Y)$, where $X \cup Y = Z$, as follows.*

1. *If both X and Y are A FIs, then the confidence of r, called the A-confidence of r and denoted as $Aconf(r)$, is defined as $\frac{Asupp(Z)}{Asupp(X)}$.*
2. *If both X and Y are H FIs, then the confidence of r, called the H-confidence of r and denoted as $Hconf(r)$, is defined as $\frac{Hsupp(Z)}{Hsupp(X)}$.*
3. *If X is an A FI and Y is an H FI, then the confidence of r, called the AH-confidence of r and denoted as AHconf(r), is defined as $\frac{AHsupp(Z)}{Asupp(X)}$.*
4. *If X is an H FI and Y is an A FI, then the confidence of r, called the HA-confidence of r and denoted as HAconf(r), is defined as $\frac{HAsupp(Z)}{Hsupp(X)}$.*

Problem Description. Given an *AH*-pair database D, σ and c, the problem of VAR mining is to find all VARs r such that *supp*$(r) \geq \sigma$ and *conf*$(r) \geq c$, where *supp* and *conf* are one of the A-, H-, *AH*-, and *HA*- support and confidence.

Note that the thresholds σ and c can be different for different types of VARs. Hereafter, we just set them to be the same for different types of VARs, and this can be easily generalized to the case of different thresholds.

We give some properties of VARs which can be used to design an efficient algorithm for mining VARs. The following proposition states that the support defined for an itemset in an *AH*-pair database has the anti-monotone property.

Proposition 1. *The following statements are true.*

1. *If* $X \subseteq X'$*, then* $Asupp(X') \leq Asupp(X)$ *and* $Hsupp(X') \leq Hsupp(X)$.
2. *Given an item* x, $\frac{M_H(x)}{2} \leq M_A(x) \leq 1 - \frac{M_H(x)}{2}$.
3. *Given a VAR,* $r = (X \Rightarrow Y)$*, where* $|X| = m$ *and* $|Y| = n$*, we have* $(\frac{1}{2})^m$ *Hsupp*(r) $\leq$*AHsupp*$(r) \leq 2^n Asupp(r)$; $(\frac{1}{2})^n$ *Hsupp*$(r) \leq HAsupp(r) \leq 2^m Asupp(r)$; *AHconf*$(r) \leq 2^n Aconf(r)$; $(\frac{1}{2})^n Hconf(r) \leq HAconf(r)$.

3 Mining Vague Association Rules

In this section, we present an algorithm to mine the VARs. We mine the set of all A, H, *AH* and *HA* FIs from the input *AH*-pair database, and then generate the VARs from FIs.

Let A_i and H_i be the set of A FIs and H FIs containing i items, respectively. Let A_iH_j be the set of *AH* FIs containing i items with A values and j items with H values. Note that A_iH_j is equivalent to H_jA_i. Let C_S be the set of *candidate FIs*, from which the set of FIs S is to be generated, where S is A_i, H_i, or A_iH_j.

Algorithm 1 **MineVFI**(D, σ)

1. Mine A_1 and H_1 from D;
2. Generate C_{A_2} from A_1, $C_{A_1H_1}$ from A_1 and H_1, and C_{H_2} from H_1;
3. Verify the candidate FIs in C_{A_2}, $C_{A_1H_1}$ and C_{H_2} to give A_2, A_1H_1 and H_2, respectively;
4. **for each** $k = 3, 4, \ldots$, where $k = i + j$, **do**
5. Generate C_{A_k} from A_{i-1} and C_{H_k} from H_{i-1}, for $i = k$;
6. Generate $C_{A_iH_j}$ from $A_{i-1}H_j$, for $2 \leq i < k$, and from A_1H_{j-1}, for $i = 1$;
7. Verify the candidate FIs in C_{A_k}, C_{H_k}, and $C_{A_iH_j}$ to give A_k, H_k, and A_iH_j;
8. **return** all A_i, H_j, and A_iH_j mined;

The algorithm to compute the FIs is shown in Algorithm 1. We first mine the set of frequent items A_1 and H_1 from the input *AH*-pair database D. Next, we generate the candidate FIs that consists of two items (Line 2) and compute the FIs from the candidate FIs (Line 3). Then, we use the FIs containing $(k-1)$ items to generate the candidate FIs containing k items, for $k \geq 3$, which is described as follows.

For each pair of FIs, $x_1 \cdots x_{k-2}y$ and $x_1 \cdots x_{k-2}z$ in A_{k-1} or H_{k-1}, we generate the itemset $x_1 \cdots x_{k-2}yz$ into C_{A_k} or C_{H_k}. For each pair of FIs, $x_1 \cdots x_{i-2}uy_1 \cdots y_j$ and $x_1 \cdots x_{i-2}vy_1 \cdots y_j$ in $A_{i-1}H_j$, or $x_1y_1 \cdots y_{j-2}u$ and $x_1y_1 \cdots y_{j-2}v$ in A_1H_{j-1}, we generate the itemset $x_1 \cdots x_{i-2}uvy_1 \cdots y_j$ or $x_1y_1 \cdots y_{j-2}uv$ into $C_{A_iH_j}$.

After generating the candidate FIs, we obtain the FIs as follows. For each $Z \in C_{A_k}$ (or $Z \in C_{H_k}$), if $\exists X \subset Z$, where X contains $(k-1)$ items, $X \notin A_{k-1}$ (or $X \notin H_{k-1}$), then we remove Z from C_{A_k} (or C_{H_k}). For each $Z = x_1 \cdots x_iy_1 \cdots y_j \in C_{A_iH_j}$, if $\exists i'$, where $1 \leq i' \leq i$, $(Z - \{x_{i'}\}) \notin A_{i-1}H_j$; or $\exists j'$, where $1 \leq j' \leq j$, $(Z - \{y_{j'}\}) \notin A_iH_{j-1}$, then we remove Z from $C_{A_iH_j}$. Here, the *anti-monotone property* [1] of support is applied to prune Z if any of Z's subsets is not an FI. After that, the support of the candidate FIs is computed and only those with support at least σ are retained as FIs. Finally, the algorithm terminates when no candidate FIs are generated and returns all FIs.

After mining the set of all FIs, we generate the VARs from the FIs. There are four types of VARs. First, for each A or H FI Z, we can generate the VARs $X \Rightarrow Y, \forall X, Y$

where $X \cup Y = Z$, using the classical AR generation algorithm [1]. Then, for each *AH* (or *HA*) FI $Z = (X \cup Y)$, where X is an A FI and Y is an H FI, we generate two VARs $X \Rightarrow Y$ and $Y \Rightarrow X$. The confidence of the VARs can be computed by Definition 4.

4 Experiments

In this section, we use both real and synthetic datasets to evaluate the efficiency of the VAR mining algorithm and the usefulness of the VARs. All experiments are conducted on a Linux machine with an Intel Pentium IV 3.2GHz CPU and 1GB RAM.

4.1 Experiments on Real Datasets

For the first set of experiments, we use the Web log data from IRCache [6], which is the NLANR Web Caching project. Then we can classify Web pages into three categories: *target*, *non-target*, and *transition* according to the time spent on the Web page, the position of the Web page in the browsing trail and the number of visits to the Web page. The three categories correspond to the three status of items, i.e., 1, 0 and h.

Since the Web log data contain a huge number of different Web sites, we only report the result on the Web log of a single Web site (www.google.com) from all nine IRCache servers on a single day (Aug. 29, 2006).When σ= 0.001 and c=0.9, we obtain one VAR: *http://gmail.google.com/, http://gmail.google.com/mail/* $\Rightarrow$ *http://mail.google.com/mail/*, with HA-support of 0.003 and HA-confidence of 0.99. This VAR shows that *http://gmail.google.com/* and *http://gmail.google.com/mail/* always play the role of transition pages to the target page *http://mail.google.com/mail/*. As a possible application, we can add a direct link from the transition pages (*http://gmail.google.com/* or *http://gmail.google.com/mail/*) to the target page (*http://mail.google.com/mail/*) to facilitate the user traversal of the Web site. Actually, by typing either the URL of the two transition pages in a Web browser, it is redirected to the URL of the target page, where the redirect mechanism serves as a special kind of direct link.

In order to compare with the traditional ARs, we also test on the database that contains all the trails without distinguishing the Web pages. At σ= 0.0008 and c=1, 70 ARs are returned. Among them, 59 ARs (84%) contain the entrance page (www.google.com), which is not that interesting. Among the remaining ARs, the following rule is found: *http://mail.google.com/, http://gmail.google.com/, http://gmail.google.com/mail/* $\Rightarrow$ *http://mail.google.com/mail/* with support 0.001 and confidence 1, which is similar to the VAR we find. This result shows the effectiveness of mining VARs, since the traditional AR mining approach returns many ARs but it is difficult for the user to tell which ARs are more important for practical uses, while mining VARs can find more specific rules directly.

4.2 Experiments on Synthetic Datasets

We test on the synthetic datasets to evaluate the efficiency and the scalability of our algorithm. We modify the IBM synthetic data generator [7] by adding "hesitation" items. The ID and the number of "hesitation" items in each transaction are generated according to the same distributions as those for the original items. We generate a dataset with 100000 transactions and 100 items. We use a parameter *Step* to represent the number of transactions which are aggregated to give an AH-pair transaction.

We first test the algorithm under different values of σ. Fig. 1 and Fig. 2 report the running time and the number of FIs. From Fig. 1, the running time increases with the

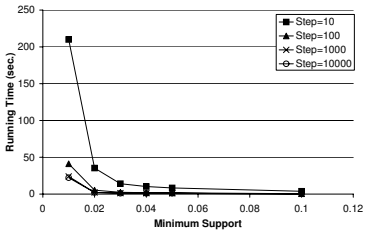

Fig. 1. Running Time

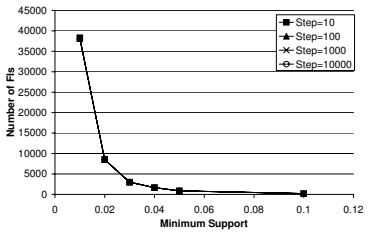

Fig. 2. Number of FIs

decrease in the value of σ due to the larger number of FIs generated. We also find that, for the same value of σ, the running time decreases significantly with the increase in the value of *Step*. This is because we aggregate more transactions to a single AH-pair transaction and hence the number of AH-pair transactions is smaller in the database. However, Fig. 2 shows that the number of FIs for the different *Step* values varies only slightly (note that all the four lines are coincided into one line in Fig. 2). We further check the FIs obtained for the different *Step* values and find that they are indeed similar. This result shows that we can actually aggregate more transactions to give the AH-pair transactions so that we can improve the efficiency of the mining operation but still obtain the same set of FIs and hence the VARs.

5 Conclusions

We apply the vague set theory to address a limitation in traditional AR mining problem, that is, the hesitation information of items is not considered. We propose the notion of VARs that incorporates the hesitation information of items into ARs. We also define different types of support and confidence for VARs in order to evaluate the quality of the VARs for different purposes. An efficient algorithm is proposed to mine the VARs, while the effectiveness of VARs is also confirmed by the experiments on real datasets.

References

1. Agrawal, R., Imielinski, T., Swami, A.N.: Mining association rules between sets of items in large databases. In Buneman, P., Jajodia, S., eds.: SIGMOD Conference, ACM Press (1993) 207–216
2. Gau, W.L., Buehrer, D.J.: Vague sets. IEEE Transactions on Systems, Man, and Cybernetics **23** (1993) 610–614
3. Zadeh, L.A.: Fuzzy sets. Information and Control **8** (1965) 338–353
4. Lu, A., Ng, W.: Managing merged data by vague functional dependencies. In Atzeni, P., Chu, W.W., Lu, H., Zhou, S., Ling, T.W., eds.: ER. Volume 3288 of Lecture Notes in Computer Science., Springer (2004) 259–272
5. Lu, A., Ng, W.: Vague sets or intuitionistic fuzzy sets for handling vague data: Which one is better? In Delcambre, L.M.L., Kop, C., Mayr, H.C., Mylopoulos, J., Pastor, O., eds.: ER. Volume 3716 of Lecture Notes in Computer Science., Springer (2005) 401–416
6. NLANR: (http://www.ircache.net/)
7. IBM Quest Data Mining Project. The Quest retail transaction data generator. http://www.almaden.ibm.com/software/quest/ (1996)