Mining Hesitation Information by Vague Association Rules

An Lu and Wilfred Ng

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

Abstract. In many online shopping applications, such as Amazon and eBay, 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. However, there is no conceptual model that is able to capture different statuses of hesitation information. Herein, we apply and extend vague set theory in the context of AR mining. 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 the two concepts, we propose the notion of Vague Association Rules (VARs). We 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 do the traditional ARs.

1 Introduction

Association Rule (*AR*) mining [1] is one of the most important data mining tasks. Consider the classical market basket case, in which AR mining is conducted on transactions that consist of items bought by customers. There are many items that are not bought but customers may have considered to buy them. We call such information on a customer's consideration to buy an item the *hesitation* information of the item, since the customer hesitates to buy it. The hesitation information of an item is useful knowledge for boosting the sales of the item. However, such information has not been considered in traditional AR mining due to the difficulty to collect the relevant data in the past. Nevertheless, with the advance in technology of data dissemination, it is now much easier for such data collection. A typical example is an online shopping scenario, such as "Amazon.com", which we are able to collect huge amount of data from the Web log that can be modelled as hesitation information. From Web logs, we can infer a customer's browsing pattern in a trail, say how many times and how much time s/he spends on a Web page, at which steps s/he quits the browsing, what and how many items are put in the basket when a trail ends, and so on. Therefore, we can further identify and categorize different browsing patterns into different hesitation information with respect to different applications.

There are many statuses of a piece of hesitation information (called *hesitation status (HS)*). Let us consider a motivating example of an online shopping scenario that

involves various statuses: (s_1) HS of the items that the customer browsed only once and left; (s_2) HS of the items that are browsed in detail (e.g., the figures and all specifications) but not put into their online shopping carts; (s_3) HS of the items that customers put into carts and were checked out eventually. All of the above-mentioned HSs are the hesitation information of those items. Some of the HSs are comparable based on some criterion, which means we can define an order on these HSs. For example, given a criterion as the possibility that the customer buys an item, we have $s_1 \leq s_2 \leq s_3$. The hesitation information can then be used to design and implement selling strategies that can potentially turn those "interesting" items into "under consideration" items and "under consideration" items into "sold" items.

Our modelling technique of HSs of an item rests on a solid foundation of *vague set theory* [2–4]. The main benefit of this approach is that the theory addresses the drawback of a single membership value in *fuzzy set theory* [5] 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 naturally model the hesitation information of an item in the mining context as the evidence of hesitation with respect to an 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. For example, if a customer bought an item 5 times, hesitated to buy (when different HSs are not distinguished) it 2 times, and did not browse it 3 times (in 10 visits), then we can obtain a vague membership value, [0.5, 0.7] (where $0.7 = 1 - 3/10$), for the item. When we distinguish different HSs, say the customer hesitated to buy the item 2 times in HSs s_1 once and s_2 once, where $s_1 \leq s_2 \leq s_3$. Then the vague membership value for s_1 is [0.5, 0.6] and that for s_2 is [0.6, 0.7]. As for s_3, since there is no hesitation evidence for it, and $s_2 \leq s_3$, its vague membership value is a single point, [0.7, 0.7].

To study the relationship between the support evidence and the hesitation evidence with respect to an item, we propose *attractiveness* and *hesitation* of an item, which 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 the customer is 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 is identified and resolved (e.g., some promotion on the item is provided). For example, given the vague membership value, [0.5, 0.7], of an item, the attractiveness is 0.6 (the median of 0.5 and 0.7) and the hesitation is 0.2 (the difference between 0.7 and 0.5), which implies that the customer may buy the item next time with a possibility of 60% and hesitate to buy the item with a possibility of 20%.

Using the 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. For example, if we find an AH-rule like "People always buy quilts and pillows(A)

but quit the process of buying beds at the step of choosing delivery method(H)". Thus, there might be something wrong with the delivery method for beds (for example, no home delivery service provided) which causes people hesitate to buy beds. 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.

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 are more specific and are able to capture richer information. Most 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. The concept of VARs is not limited to the online shopping scenario. In our experiments, we demonstrate that VARs are applied to mine Web log data.

Organization. This paper is organized as follows. Section 2 gives some preliminaries on vague sets and ARs. Section 3 introduces VARs and presents the related concepts. Section 4 discusses the algorithm that mines VARs. Section 5 reports the experimental results. Section 6 discusses the related work and Section 7 concludes the paper.

2 Preliminaries

2.1 Vague Sets

Let I be a classical set of objects, called the universe of discourse, where an element of I is denoted by x.

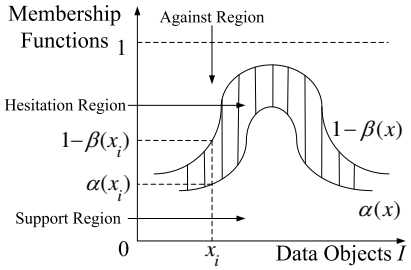

Fig. 1. The True (α) and False (β) Membership Functions of a Vague Set

Definition 1. (Vague Set) *A vague set V in a universe of discourse I is characterized by a true membership function, α_V, and a false membership function, β_V, as follows:* $\alpha_V : I \rightarrow [0,1]$, $\beta_V : I \rightarrow [0,1]$, *where* $\alpha_V(x) + \beta_V(x) \leq 1$, $\alpha_V(x)$ *is a lower bound on the grade of membership of x derived from the evidence* for x, *and* $\beta_V(x)$ *is a lower bound on the grade of membership of the* $negation$ *of x derived from the evidence* against x. *Suppose* $I = \{x_1, x_2, \ldots, x_n\}$. *A vague set V of the universe of discourse I is represented by* $V = \sum_{i=1}^{n} [\alpha(x_i), 1-\beta(x_i)]/x_i$, *where* $0 \leq \alpha(x_i) \leq (1-\beta(x_i)) \leq 1$.□

The grade of membership of x is bounded to $[\alpha_V(x), 1-\beta_V(x)]$, which is a subinterval of $[0,1]$ as depicted in Fig. 1. For brevity, we omit the subscript V from α_V and β_V.

We say that $[\alpha(x), 1-\beta(x)]/x$ is a *vague element* and the interval $[\alpha(x), 1-\beta(x)]$ is the *vague value* of the object x. For example, $[\alpha(x), 1-\beta(x)] = [0.5, 0.7]$ is interpreted

as "the degree that the object x belongs to the vague set V is 0.5 (i.e. $\alpha(x)$) and the degree that x does not belong to V is 0.3 (i.e. $\beta(x)$)." For instance, in a voting process, the vague value [0.5,0.7] can be interpreted as "50% of the votes support the motion, 30% are against, while 20% are neutral (abstentions)."

2.2 Median Memberships and Imprecision Memberships

In order to compare vague values, we introduce two derived memberships: median membership and imprecision membership [4]. Note that given a vague value $[\alpha(x), 1-\beta(x)]$, we have unique median membership $M_m(x)$ and imprecision membership $M_i(x)$, and vice versa.

Median membership is defined as $M_m = \frac{1}{2}(\alpha + (1-\beta))$, which represents the overall evidence contained in a vague value. It can be checked that $0 \leq M_m \leq 1$. Obviously, the vague value [1,1] has the highest M_m, which means the corresponding object definitely belongs to the vague set (i.e., a crisp value). While the vague value [0,0] has the lowest M_m, which means that the corresponding object definitely does not belong to the vague set.

Imprecision membership is defined as $M_i = ((1-\beta) - \alpha)$, which represents the overall imprecision of a vague value.It can be checked that $0 \leq M_i \leq 1$. The vague value $[a, a](a \in [0, 1])$ has the lowest M_i which means that the membership of the corresponding object is exact (i.e., a fuzzy value). While the vague value [0,1] has the highest M_i which means that we do not have any information about the membership of the corresponding object.

The median membership and the imprecision membership are employed in this paper to measure the attractiveness and the hesitation of an item with respect to a customer.

2.3 Association Rules

Let $I = \{x_1, x_2, \ldots, x_n\}$ be a set of items [1]. An *itemset* is a subset of I. A *transaction* is an itemset. We say that a transaction Y *supports* an itemset X if $Y \supseteq X$. For brevity, we write an itemset $X = \{x_{k_1}, x_{k_2}, \ldots, x_{k_m}\}$ as $x_{k_1} x_{k_2} \ldots x_{k_m}$.

Let D be a database of transactions. The *frequency* of an itemset X, denoted as $freq(X)$, is the number of transactions in D that support X. The *support* of X, denoted as $supp(X)$, is defined as $\frac{freq(X)}{|D|}$, where $|D|$ is the number of transactions in D. X is a *Frequent Itemset* (*FI*) if $supp(X) \geq \sigma$, where σ $(0 \leq \sigma \leq 1)$ is a user-specified *minimum support threshold*.

Given the set of all FIs, the set of ARs is obtained as follows: for each FI Y and for each non-empty subset X of Y, we generate an AR, r, of the form $X \Rightarrow Y - X$. The *support* of r, denoted as $supp(r)$, is defined as $supp(Y)$ and the *confidence* of r, denoted as $conf(r)$, is defined as $\frac{supp(Y)}{supp(X)}$. We say that r is a *valid AR* if $conf(r) \geq c$, where c $(0 \leq c \leq 1)$ is a user-specified *minimum confidence threshold*.

3 Vague Association Rules

In this section, we first propose the concept of Hesitation Statuses (*HSs*) of an item and discuss how to model HSs. Then we introduce the notion of *Vague Association Rules* (*VARs*) and four types of support and confidence used in order to fully evaluate their quality. Some properties of VARs that are useful to improve the efficiency of mining VARs are presented.

[1] We refer to the terms *item* and *object* interchangeably in this paper.

3.1 Hesitation Information Modeling

A *Hesitation Status (HS)* is a specific state between two certain situations of "buying" and "not buying" in the process of a purchase transaction.

Here we use a more detailed example of placing an order with "Amazon.com" [6] to illustrate the idea of HS. There are following nine steps, which forms a queue, to place an order: (s_1) Find the items you want; (s_2) Add the items to your shopping cart; (s_3) Proceed to checkout; (s_4) Sign in; (s_5) Enter a shipping address; (s_6) Choose a shipping method; (s_7) Provide a password and payment information; (s_8) Review and submit your order; (s_9) Check your order status.

A customer may quit the ordering process at any step for some reason, for example, forgetting the sign name or password. Therefore, the HSs with respect to different quitting steps are different, since the more steps a customer goes, the higher possibility the customer buys the item.

However, some HSs are incomparable. For example, a customer may put an item into the wishing list if the item is out of stock. The HS in this case is incomparable to the HS of the item with respect to quitting order at step 6, since we lack evidence to support any ordered relationship between these two HSs.

We now formally model the hesitation information of an item as follows.

Definition 2. (Hesitation and Overall Hesitation) *Given an item $x \in I$ and a set of HSs $S = \{s_1, s_2, \ldots, s_w\}$ with a partial order $\leq$. The* hesitation *of x with respect to an HS $s_i \in S$ is a function $h_i(x) : I \rightarrow [0, 1]$, such that $\alpha(x) + \beta(x) + \sum_{i=1}^{w} h_i(x) = 1$, where $h_i(x)$ represents the evidence for the HS s_i of x. The* overall hesitation *of x with respect to S is given by $H(x) = \sum_{i=1}^{w} h_i(x)$.* □

It can be easily checked from the above definition that $H(x) = 1 - \alpha(x) - \beta(x)$. S can also be represented by a Hasse Diagram whose vertices are elements in S and the edges correspond to $\leq$. All components in S can be partitioned into two groups of HSs: a Chain Group (CG) consists of connected components that are chains (including the case of a singleton HS node), and a Non-Chain Group (NCG) consists of components that are non-chains (not chains).

In order to capture the hesitation evidence and the hesitation order $\leq$, a subinterval of $[\alpha(x), 1-\beta(x)]$ is used to represent the customer's *intent* of each item with respect to different HSs. To obtain the intent value, the idea of linear extensions of a partial order is used. However, computing the number of extensions is a #P-complete problem [7]. An algorithm that generates all of the linear extensions of a partially ordered set in constant amortized time is given in [8]. In real applications, say the online-shopping scenario, we can simplify $\leq$ in order to reduce the computation complexity. From now on, we assume that a component G in the NCG is a chain of *Incomparable Chain Sets* (ICSs), $\{ICS_1 \leq ICS_2 \leq \cdots \leq ICS_l\}$, where $ICS_i \in G$ is a set of chains satisfying the following condition: the parent (child) HSs of the top (bottom) elements in all chains, if any, are in the same ICS.

Note that this condition implies that the parent (child) HS of a chain in the top (bottom) ICS is an empty set.

We now present an algorithm that partitions a component G in NCG into different ICSs in Algorithm 1.

Example 1. Let $S = \{s_1, \ldots, s_{17}\}$, and its Hasse diagram consists of four components as shown in Fig. 2. We see that the component g_2 is in CG, since it is a chain, and the

Algorithm 1 **PartitionNCG**(G)

1. $i := 1$;
2. **while** $G \neq \emptyset$
3. Let ICS_i be the set of all minimal HS $s \in G$;
4. **forall** $s \in ICS_i$ **do**
5. Search the longest chain segment of s such that each HS (excluding s itself) in it has a unique child, and the child has a unique parent;
6. Put all HSs of the chain segment in ICS_i if any;
7. $G := G - ICS_i$; $i := i + 1$;
8. **return** the result $\{ICS_1 \leq ICS_2 \leq \cdots \leq ICS_i\}$.

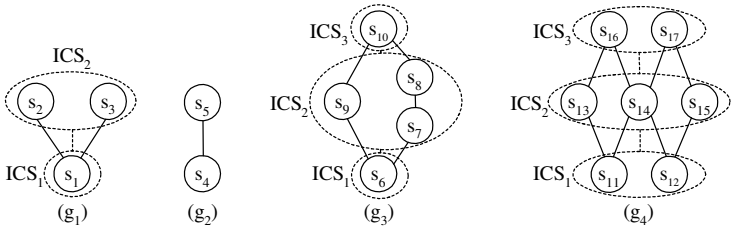

Fig. 2. The Hasse Diagram of the Ordered Set of HSs

components g_1, g_3, and g_4 are in NCG, where different ICSs are represented by the dashed ellipses. Consider the component g_1, s_1 is the only element in ICS_1, since it has more than one parent (i.e. s_2 and s_3) and the longest chain segment of s_1 (according to Line 5 of Algorithm 1) contains itself only, while s_2 and s_3 are in ICS_2, since they are the top HSs and have no parents. Thus, we partition g_1 into the chain of ICSs as $ICS_1 \leq ICS_2$. Consider the component g_3, s_6 is the only element in ICS_1. s_9 and s_7 are the minimal HSs in ICS_2. s_8 is also in ICS_2, since it has a unique child s_7 and s_7 has a unique parent s_8. s_{10} is not in ICS_2, since it has two children (i.e. s_8 and s_9). Consider the component g_4, s_{11} and s_{12} are in ICS_1, since they both have more than one parents. s_{13}, s_{14} and s_{15} are in ICS_2 and finally, the top HSs s_{16} and s_{17} form ICS_3. □

Given a set of purchase transactions, we can aggregate the transactions to obtain the *intent* of each item with respect to different HSs. Although aggregating the transactions may lose some exact information of the items for customers, in many cases the overall knowledge is more useful than the details of every transaction. For example, in our online shopping scenario, by aggregating the data we alleviate the problem of high cost of mining on huge log data sets.

Definition 3. (Intent and Overall Intent) *Given a set of HSs, $(S, \leq)$, the* intent *of an item x with respect to an HS $s_i \in S$, denoted as $int(x, s_i)$, is a vague value $[\alpha_i(x), 1 - \beta_i(x)]$ which is a subinterval of $[\alpha(x), 1 - \beta(x)]$ and satisfies the following conditions:*

1. $(1 - \beta_i(x)) = \alpha_i(x) + h_i(x)$.
2. *For each ICS_j in the chain $ICS_1 \leq ICS_2 \leq \cdots \leq ICS_j \leq \cdots \leq ICS_l$, of an NCG component G, we assume a linear extension of G ($s_1 \leq s_2 \leq \cdots \leq$*

s_m) *such that there exists a segment* ($s_{p+1} \leq \cdots \leq s_{p+q}$) *consistent with all the chains in* ICS_j, *where* $1 \leq p+1$ *and* $p+q \leq m$. *The intent for* ICS_j, *denoted as* $[\alpha_{ICS_j}(x), 1-\beta_{ICS_j}(x)]$, *is given by* $\alpha_{ICS_j}(x) = \frac{\alpha(x)+(1-\beta(x))}{2} - \frac{1}{2}\sum_{k=1}^{m} h_k(x) + \sum_{k=1}^{p} h_k(x)$, $1-\beta_{ICS_j}(x) = \alpha_{ICS_j}(x) + \sum_{k=p+1}^{p+q} h_k(x)$.

3. – *If* s_i *is in a chain of the CG,* $s_1 \leq s_2 \leq \cdots \leq s_i \leq \cdots \leq s_n$, *then for* $1 \leq i \leq n$, *we define*
$\alpha_i(x) = \frac{\alpha(x)+(1-\beta(x))}{2} - \frac{1}{2}\sum_{k=1}^{n} h_k(x) + \sum_{k=1}^{i-1} h_k(x)$.
– *If* s_i *is in a chain of* ICS_j, $s_g \leq s_{g+1} \leq \cdots \leq s_i \leq \cdots \leq s_v$, *where* $((p+1) \leq g \leq v \leq (p+q))$, *then for* $g \leq i \leq v$, *we define*
$\alpha_i(x) = \frac{\alpha_{ICS_j}(x)+(1-\beta_{ICS_j}(x))}{2} - \frac{1}{2}\sum_{k=g}^{v} h_k(x) + \sum_{k=g}^{i-1} h_k(x)$.

The overall intent *of* x, *denoted as* $INT(x)$, *is the interval* $[\alpha(x), 1-\beta(x)]$. □

Condition 1 shows the relationship among $(1-\beta_i(x))$, $\alpha_i(x)$ and $h_i(x)$. Together with condition 3, we can determine the intent $[\alpha_i(x), 1-\beta_i(x)]$, since $h_i(x)$, $\alpha(x)$ and $(1-\beta(x))$ are given parameters.

The formulas in condition condition 3 are similar, which are defined to ensure that the numerical order of median membership of the HSs is consistent with the order of HSs. This also fits for the cases in most real life applications.

Example 2. Table 1 shows the transactions of a single customer derived from an online shopping system, where we use 1 and 0 to represent that an item is bought and not bought (without any hesitation information), as in the traditional AR mining setting. The set of HSs is given by $S = \{s_1 \leq s_2, s_1 \leq s_3, s_4 \leq s_5\}$, that is, the graphs g_1 and g_2 in Fig. 2.

In Table 1, given 10 transactions, we have 7 *buy* and 1 *not buy* and 2 HSs (s_1 and s_3) for an item A. Consider $g_1 = \{s_1 \leq s_2, s_1 \leq s_3\}$, we have one of its linear extension $s_1 \leq s_2 \leq s_3$. Since $ICS_1 = \{s_1\}$ and $ICS_2 = \{s_2, s_3\}$, we have $int(A, s_1) = [0.6, 0.8]$, $int(A, s_2) = [0.85, 0.85]$ and $int(A, s_3) = [0.8, 0.9]$, according to Definition 3. As s_4 and s_5 is a chain in CG, we then obtain $int(A, s_4) = int(A, s_5) = [0.75, 0.75]$. Thus, we obtain all the intent of A for the HSs in S as shown in the first column of Table 2 and in Fig. 3. It can be checked that s_2, s_4 and s_5 are single points, since the hesitation evidence is zero for them. The intent database of all items (A, B, C, D) for different HSs (s_1, . . ., s_5) can be similarly determined, which is shown in Table 2 and also illustrated by Fig. 3. Note that the values in the last row of the table are $[\alpha(x), 1-\beta(x)]$, indicating the overall hesitation $H(x)$. □

Given the intent of an item for an HS, we further define the attractiveness of the item which represents the overall evidence for it with respect to an HS.

Definition 4. (Attractiveness and Overall Attractiveness) *The* attractiveness *of* x *with respect to an HS* s_i, *denoted as* $att(x, s_i)$, *is defined as the median membership of* x *with respect to* s_i, *that is,* $\frac{1}{2}(\alpha_i(x) + (1-\beta_i(x)))$. *The* overall attractiveness *of* x *is a function* $ATT(x) : I \rightarrow [0, 1]$, *such that* $ATT(x) = \frac{1}{2}(\alpha(x) + (1-\beta(x)))$. □

Given the intent $[\alpha_i(x), 1-\beta_i(x)]$ of an item x for an HS s_i, we have a one-one corresponding pair of the attractiveness and hesitation of x, called the *AH*-pair, denoted as $\langle att(x, s_i), h_i(x)\rangle$. Attractiveness and hesitation are two important concepts, since people may have special interest in finding ARs with items of high attractiveness (sold well) or high hesitation (almost sold).

We now define an *AH*-pair transaction and an *AH*-pair database.

Table 1. Ten Transactions of a Customer

TID	A	B	C	D
1	1	s_4	s_4	s_1
2	1	0	s_1	0
3	1	1	s_1	s_3
4	0	s_5	s_3	s_3
5	s_1	1	s_2	s_2
6	1	0	s_5	s_3
7	s_1	s_5	s_3	s_3
8	1	0	s_4	s_5
9	s_3	0	0	0
10	1	s_5	1	s_5

Table 2. An Intent Database for Different HSs

H	A	B	C	D
$h_1(x)$	[0.6,0.8]	[0.4,0.4]	[0.25,0.45]	[0.1,0.2]
$h_2(x)$	[0.85,0.85]	[0.4,0.4]	[0.55,0.65]	[0.4,0.5]
$h_3(x)$	[0.8,0.9]	[0.4,0.4]	[0.5,0.7]	[0.25,0.65]
$h_4(x)$	[0.75,0.75]	[0.2,0.3]	[0.35,0.55]	[0.3,0.3]
$h_5(x)$	[0.75,0.75]	[0.3,0.6]	[0.55,0.65]	[0.3,0.5]
$H(x)$	[0.6,0.9]	[0.2,0.6]	[0.1,0.9]	[0,0.8]

Table 3. An *AH*-pair Database for Different HSs

H	A	B	C	D
$h_1(x)$	<0.7,0.2>	<0.4,0>	<0.35,0.2>	<0.15,0.1>
$h_2(x)$	<0.85,0>	<0.4,0>	<0.6,0.1>	<0.45,0.1>
$h_3(x)$	<0.85,0.1>	<0.4,0>	<0.6,0.2>	<0.45,0.4>
$h_4(x)$	<0.75,0>	<0.25,0.1>	<0.45,0.2>	<0.3,0>
$h_5(x)$	<0.75,0>	<0.45,0.3>	<0.6,0.1>	<0.4,0.2>
$H(x)$	<0.75,0.3>	<0.4,0.4>	<0.5,0.8>	<0.4,0.8>

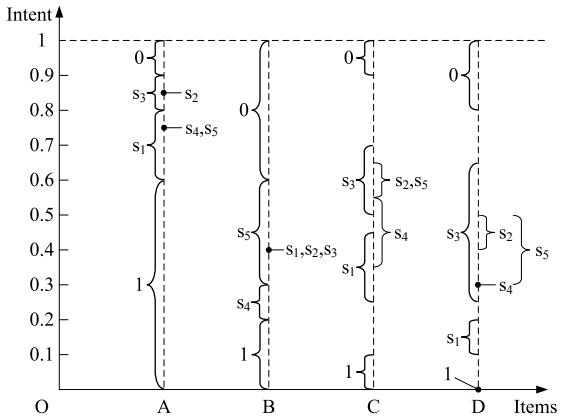

Fig. 3. Intent for Different HSs of Items

Definition 5. **(*AH*-Pair Transaction and Database)** *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 *with respect to a given HS or the overall hesitation, for* $1 \leq j \leq m$*. An* AH-pair database *is a sequence of AH-pair transactions.* □

We can transform the intent database shown in Table 2 to its equivalent *AH*-pair database shown in Table 3 without losing any information and present the attractiveness and hesitation of the items directly.

We can further calculate AH-pairs directly without calculating the intent first, and the calculation process can be simplified. Since $att(x, s_i) = \alpha_i(x) + \frac{1}{2}h_i(x)$, and

$h_i(x)$ is known, the three conditions in Definition 3 can be replaced by three equivalent conditions. Formally, we give the method of calculating AH-pairs as follows.

The AH-pair of an item x with respect to an HS s_i, $\langle att(x, s_i), h_i(x)\rangle$, satisfies the following conditions:

1. Assume the same setting as stated as condition 2 of Definition 3.
   The attractiveness for ICS_j, denoted as $att(x, ICS_j)$, is given as follows.
   $att(x, ICS_j) = ATT(x) - \frac{1}{2}\sum_{k=1}^{m} h_k(x) + \sum_{k=1}^{p} h_k(x) + \frac{1}{2}\sum_{k=p+1}^{p+q} h_k(x)$,
2. If s_i is in a chain of the CG, $s_1 \leq s_2 \leq \cdots \leq s_i \leq \cdots \leq s_n$, for $1 \leq i \leq n$, we define $att(x, s_i) = ATT(x) - \frac{1}{2}\sum_{k=1}^{n} h_k(x) + \sum_{k=1}^{i-1} h_k(x) + \frac{1}{2}h_i(x)$.
3. If s_i is in a chain of ICS_j, $s_g \leq s_{g+1} \leq \cdots \leq s_i \leq \cdots \leq s_v$, where $((p+1) \leq g \leq v \leq (p+q))$, for $g \leq i \leq v$, we define
   $att(x, s_i) = att(x, ICS_j) - \frac{1}{2}\sum_{k=g}^{v} h_k(x) + \sum_{k=g}^{i-1} h_k(x) + \frac{1}{2}h_i(x)$. □

The following proposition states the relationship between the hesitation order of two HSs and the attractiveness for them of an item.

Proposition 1. *If $s_i \leq s_j$, then $att(x, s_i) \leq att(x, s_j)$.*

Proof. It follows directly from the method of calculating AH-pairs above. □

The following proposition states the relationship between the overall attractiveness and hesitation, and the attractiveness and hesitation for different HSs.

Proposition 2. *Given $S = \{s_1, s_2, \ldots, s_w\}$, $ATT(x)H(x) = \sum_{i=1}^{w} att(x, s_i)h_i(x)$.*

Proof Sketch. We first prove the case that S contains CG components only, where condition 2 in the method of calculating AH-pairs above applies to all chains in S. Then we extend the proof to the case that S includes both CG and NCG components, where condition 3 applies for the chains of ICSs in NCG. When S contains CG components only, for any chain, $(s_1 \leq \cdots \leq s_i \leq \cdots \leq s_n)$, in S, we have $ATT(x) - att(x, s_i) = \frac{1}{2}\sum_{k=1}^{n} h_k(x) - \sum_{k=1}^{i-1} h_k(x) - \frac{1}{2}h_i(x) = \frac{1}{2}(h_{i+1}(x) + h_{i+2}(x) \cdots + h_n(x) - h_1(x) - h_2(x) - \cdots - h_{i-1}(x))$. It can be checked that $\sum_{i=1}^{n}(ATT(x) - att(x, s_i))h_i(x) = 0$. Then we extend this conclusion to the whole S, since any chains in S satisfies this conclusion, that is to say, we have $\sum_{i=1}^{w}(ATT(x) - att(x, s_i))h_i(x) = 0$. Thus, $ATT(x)H(x) = \sum_{i=1}^{w} att(x, s_i)h_i(x)$. When S contains both CG and NCG components, since each ICS_j of NCG in condition 3 can be regarded as the case of CG in condition 2, in a similar way, we can check that the conclusion also holds for the case including NCG components. □

Proposition 2 indicates that the sum of the product of attractiveness and hesitation with respect to all HSs preserves the product of overall attractiveness and hesitation.

3.2 Vague Association Rules and their Support and Confidence

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

Definition 6. (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 with respect to an HS, we can define different types of support and confidence of a VAR. For example, if we have special interest in the association between well-sold items (high attractiveness) and almost-sold items (high hesitation). Then, with some analysis between the former and the latter, we may make some improvements to boost the sales of the latter. For this purpose, we define *Attractiveness-Hesitation* (*AH*) support and *AH* confidence of a VAR to evaluate the VAR. Similarly, we can obtain the association between an itemset with high hesitation and another itemset with high attractiveness, between two itemsets with high attractiveness, and between two itemsets with high hesitation for different purposes. Accordingly, we define four types of support and confidence to evaluate the VARs as follows.

Note that here A (or H) can refer to either the overall attractiveness (or hesitation), or the attractiveness (or hesitation) of a given HS.

Definition 7. (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* $\frac{\sum_{T \in D} \prod_{z \in Z} M_A(z)}{|D|}$.
2. *The H-support of Z, denoted as Hsupp(Z), is defined as* $\frac{\sum_{T \in D} \prod_{z \in Z} M_H(z)}{|D|}$.
3. *The AH-support of Z, denoted as AHsupp(Z), is defined as* $\frac{\sum_{T \in D} \prod_{x \in X, y \in Y} M_A(x) M_H(y)}{|D|}$.
4. *The HA-support of Z, denoted as HAsupp(Z), is defined as* $\frac{\sum_{T \in D} \prod_{x \in X, y \in Y} M_H(x) M_A(y)}{|D|}$.

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 8. (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* $\frac{HAsupp(Z)}{Hsupp(X)}$. □

Example 3. Given the *AH*-pair database in Table 4 with respect to a given HS s_1 for customers with different CID , let $\sigma = 0.5$ and $c = 0.5$. Note that the first line in Table 4 is from the first line in Table 3, which represents the AH-pairs of different items for HS s_1 with respect to the customer with CID 1. Then, $Asupp(A) = (0.7 + 0.9 + 0.7 + 0.8 + 1)/5 = 0.82$, *AHsupp*$(A \Rightarrow D) = (0.7 \times 0.1 + 0.9 \times 0.8 + 0.7 \times 0.9 + 0.8 \times 0.7 + 1 \times 0.8)/5 = 0.556$, *AHconf*$(A \Rightarrow D) = \frac{0.556}{0.82} = 0.678$. Similarly, we calculate $AHsupp(A \Rightarrow B) = 0.112 \leq \sigma$, $AHconf(A \Rightarrow B) = 0.137 \leq c$. Thus, $A \Rightarrow D$ is a valid VAR with respect to *AH*-support and *AH*-confidence, but $A \Rightarrow B$ is not. □

Table 4. An AH-pair database on items with respect to s_1

CID	A	B	C	D
1	<0.7,0.2>	<0.4,0>	<0.35,0.2>	<0.15,0.1>
2	<0.9,0.2>	<0.7,0.2>	<0.6,0.8>	<0.5,0.8>
3	<0.7,0.1>	<0.8,0.4>	<0.4,0.7>	<0.5,0.9>
4	<0.8,0>	<0.9,0>	<0.5,0.9>	<0.4,0.7>
5	<1,0>	<0.9,0.1>	<0.4,0.8>	<0.6,0.8>

Table 5. The Four Types of Support and Confidence of $A \Rightarrow B$

	A	H	AH	HA
supp	0.618	0.016	0.112	0.06
conf	0.754	0.16	0.137	0.6

We also compute all the four types of support and confidence of $A \Rightarrow B$ as shown in Table 5. It shows that $A \Rightarrow B$ is a valid VAR with respect to A-support and A-confidence, but not a valid VAR with respect to other types of support and confidence.

Problem Description. Given an *AH*-pair database D with respect to an HS s_i or the overall hesitation, 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 a certain itemset with respect to HSs has the anti-monotone property.

Proposition 3. *Given two different HSs s_i and s_j, let $supp_i$ ($conf_i$) and $supp_j$ ($conf_j$) be the corresponding support (confidence) with respect to different HSs. The following statements are true.*

1. *If $s_i \leq s_j$, then $Asupp_i(Z) \leq Asupp_j(Z)$.*
2. *If $s_i \leq s_j$ and $\forall y \in Y$, $h_i(y) \leq h_j(y)$, then $AHsupp_i(Z) \leq AHsupp_j(Z)$.*
3. *If $\forall x \in X$, $h_i(x) \leq h_j(x)$ and $s_i \leq s_j$, then $HAsupp_i(Z) \leq HAsupp_j(Z)$.*
4. *If $\forall z \in Z$, $h_i(z) \leq h_j(z)$, then $Hsupp_i(Z) \leq Hsupp_j(Z)$.*

Proof. It follows from Definition 7 and Proposition 1. □

According to Proposition 3, when we find the support of an itemset with respect to an HS to be less than σ, we can prune the same itemset in the mining search space. The pruning applies to all the HSs less than or equal to, or in the same ICS with the original HS.

The following proposition states that the support defined for an itemset in an *AH*-pair database with respect to a certain HS has the anti-monotone property.

Proposition 4. *If $X \subseteq X'$, then $Asupp(X') \leq Asupp(X)$ and $Hsupp(X') \leq Hsupp(X)$.*

Proof. Since $X \subseteq X'$ and $0 \leq M_A(x) \leq 1$ ($x \in X'$), we have $\prod_{x \in X'} M_A(x) \leq \prod_{x \in X} M_A(x)$. Thus $Asupp(X') = \frac{\sum_{T \in D} \prod_{x \in X'} M_A(x)}{|D|} \leq \frac{\sum_{T \in D} \prod_{x \in X} M_A(x)}{|D|} = Asupp(X)$.

And we also have $AHsupp(X') \leq AHsupp(X)$, since $AHsupp(X') = Asupp(X')$ and $AHsupp(X) = Asupp(X)$. Similarly, we can prove the cases of $Hsupp$ and $HAsupp$. □

According to Proposition 4, when we find the support of an itemset to be less than σ, we can prune all its supersets in the mining search space. We can obtain greater pruning by the following two propositions.

Proposition 5. *Given an item x,* $\frac{M_H(x)}{2} \leq M_A(x) \leq 1 - \frac{M_H(x)}{2}$.

Proof. Since $\alpha(x) \geq 0$, $\frac{M_H(x)}{2} = \frac{(1-\beta(x))-\alpha(x)}{2} \leq \frac{(1-\beta(x))+\alpha(x)}{2} = M_A(x)$. Since $\beta(x) \geq 0$, $M_A(x) = \frac{\alpha(x)+(1-\beta(x))}{2} \leq \frac{\alpha(x)+(1+\beta(x))}{2} = 1 - \frac{(1-\beta(x))-\alpha(x)}{2} = 1 - \frac{M_H(x)}{2}$.

Proposition 6. *Given a VAR,* $r = (X \Rightarrow Y)$*, where* $|X| = m$ *and* $|Y| = n$*, we have*

1. $(\frac{1}{2})^m$ *Hsupp*$(r) \leq$ *AHsupp*$(r) \leq 2^n Asupp(r)$;
2. $(\frac{1}{2})^n$ *Hsupp*$(r) \leq HAsupp(r) \leq 2^m Asupp(r)$;
3. *AHconf*$(r) \leq 2^n Aconf(r)$;
4. $(\frac{1}{2})^n Hconf(r) \leq HAconf(r)$.

Proof Sketch. The proof follows from Proposition 5. □

By Proposition 6, we can prune VARs according to the relationship among different support and confidence. For example, if we have $2^n Asupp(r) < \sigma$, then *AHsupp*$(r) \leq 2^n Asupp(r) < \sigma$; thus, we can prune r directly without computing *AHsupp*(r).

4 Mining Vague Association Rules

In this section, we present an algorithm to mine the VARs. We first mine the set of all A, H, *AH* and *HA* FIs from the input *AH*-pair database with respect to a certain HS or the overall hesitation. Then, we generate the VARs from the set of 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_W be the set of *candidate FIs*, from which the set of FIs W is to be generated, where W is A_i, H_i, or A_iH_j.

Algorithm 2 **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 2. 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 $Z = x_1 \cdots x_i y_1 \cdots y_j \in C_{A_iH_j}$, 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 we mine 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 8.

After we generate all the VARs with respect to the given HS or overall hesitation, we can repeat our algorithm on the mi-pair database of different HS. Properties in Proposition 3 can be used to prune itemsets if the current HS has the relationships indicated in Proposition 3 with the original HS.

5 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. Due to space limitation, the experimental results are related to the overall hesitation.

5.1 Experiments on Real Datasets

For the first set of experiments, we use the Web log data from IRCache [9], which is the NLANR Web Caching project.

We first preprocess the Web log and identify the browsing trails of each user. Then, we define the weight, W_{wp}, of a Web page, wp, in a trail as the product of the time spent on wp and the position of wp in the trail. If wp appears more than once in the trail, we sum up its weights. Finally, we normalize the weights of the Web pages within a trail. Thus, W_{wp} measures the degree that wp satisfies the user. Given two thresholds H_L and H_U $(0 \leq H_L \leq H_U \leq 1)$, we can classify Web pages into three categories: *target* (if $W_{wp} \geq H_U$), *non-target* (if $W_{wp} \leq H_L$), and *transition* (if $H_L < W_{wp} < H_U$). The three categories correspond to the three statuses of items, i.e., 1, 0 and h (overall hesitation), respectively.

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). We identify 6066 trails and aggregate them by the user ID (the remote host). The corresponding AH-pair database consists of 263 AH-pair transactions and 260 items (i.e., Web pages). Here we set H_L to be 0.01 and H_U to be 0.7.

When σ= 0.001 and c=0.9, we obtain only 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.

If we set c to be 0.7, we obtain more VARs as follows:

1. H_1A_1: *http://google.com/* $\Rightarrow$ *http://www.google.com/* (0.001, 0.77)
2. H_1A_1: *http://gmail.google.com/* $\Rightarrow$ *http://mail.google.com/mail/* (0.004, 0.86)
3. A_2H_1: *http://mail.google.com/mail/, http://gmail.google.com/mail/* $\Rightarrow$ *http://gmail.google.com/* (0.001, 0.77)
4. A_2H_1: *http://mail.google.com/mail/, http://gmail.google.com/* $\Rightarrow$ *http://gmail.google.com/mail/* (0.001, 0.84)
5. H_1H_1: *http://gmail.google.com/* $\Rightarrow$ *http://gmail.google.com/mail/* (0.003, 0.75)

In each VAR, the number in the bracket shows the support and confidence of the VAR. We find that, in the first two H_1A_1 rules, the transition page is redirected to the target page. The next two A_2H_1 rules show that *http://gmail.google.com/mail/* and *http://gmail.google.com/* can both play the role of transition or target pages, while *http://mail.google.com/mail/* is always the target page with high confidence (above 0.7). The last H_1H_1 rule shows that both of the two pages are transition pages. We may combine them together or delete one of them to make the website more concise.

In order to compare with the traditional ARs, we also test on the database that contains all the trails without distinguishing the Web pages by their weights and aggregating the pages by user. 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 one of the VARs we find.

The above results show the effectiveness of mining VARs, since the traditional AR mining approach returns many ARs but it is hard for the user to tell which ARs are more important for practical uses, while mining VARs can find more specific rules directly.

5.2 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 [10] 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.

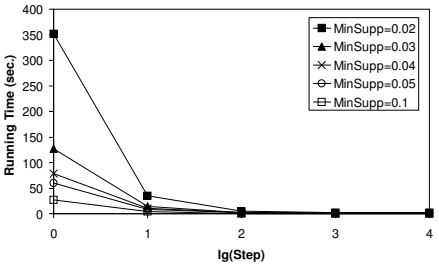

Fig. 4. Running Time

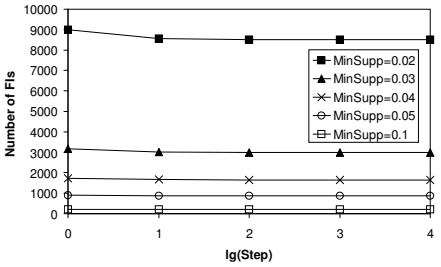

Fig. 5. Number of FIs

We first test the algorithm under different values of *Step*. Fig. 4 and Fig. 5 report the running time and the number of FIs. From Fig. 4, the running time increases with the 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. 5 shows that the number of FIs for the different *Step* values varies only slightly (note that all the five lines are nearly horizontal in Fig. 5). 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.

6 Related Work

We are aware of a number of studies that extend the traditional AR mining for uncertain data in different applications, such as mining fuzzy ARs. However, there is no modelling of hesitation information in an application [11–13]. Fuzzy ARs are proposed to handle quantitative items in the form "X is A" $\Rightarrow$ "Y is B", where X, Y are the set of items and A, B are fuzzy concepts represented by fuzzy sets. For example, "position is senior" $\Rightarrow$ "salary is high".

Although the formulas of different kinds of support and confidence in VARs seem to relate to their counterparts in fuzzy ARs, VARs and fuzzy ARs are fundamentally different. VARs focus on the associations between crisp itemsets based on the attractiveness and hesitation of items, while fuzzy ARs do not consider hesitation information and focus on the associations between fuzzy concepts.

In our previous works, we extend the concepts of Functional Dependency (FD), Chase procedure [14], SQL and AR in standard relational databases by applying vague set theory in order to handle the widely existent vague information, and propose VFD [3], VChase [15], VSQL [4] and VAR [16], respectively. In [16], a basic approach to incorporate the hesitation information into the ARs is given. However, the modelling of hesitation information with respect to different HSs is newly developed in this paper.

7 Conclusions

We model hesitation information by vague set theory in order to address a limitation in traditional AR mining problem, which ignores the hesitation information of items in transactions. We propose the notion of VARs that incorporates the hesitation information of items into ARs. We define two important concepts, attractiveness and hesitation, of an item with respect to different HSs, which reflect the overall information of a customer's intent on the item. 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 revealed by experiments on real datasets. As for future work, mining VARs in different applications is an interesting topic that deserves further study. For example, different ranking scores together with clickthrough data of a search result can be modelled as an object having different HSs. In this case VARs can be minded to reflect different users' preferences.

Acknowledgements. We would like to thank Yiping Ke and James Cheng for their valuable comments on this topic.

References

1. Agrawal, R., Imielinski, T., Swami, A.N.: Mining association rules between sets of items in large databases. In: SIGMOD Conference. (1993) 207–216
2. Gau, W.L., Buehrer, D.J.: Vague sets. IEEE Transactions on Systems, Man, and Cybernetics **23** (1993) 610–614
3. Lu, A., Ng, W.: Managing merged data by vague functional dependencies. In: ER. (2004) 259–272
4. Lu, A., Ng, W.: Vague sets or intuitionistic fuzzy sets for handling vague data: Which one is better? In: ER. (2005) 401–416
5. Zadeh, L.A.: Fuzzy sets. Information and Control **8** (1965) 338–353
6. Amazon.com Help. (http://www.amazon.com/gp/help/customer/display.html?nodeId=524700)
7. Brightwell, G., Winkler, P.: Counting linear extensions. Order **8** (1991) 225–242
8. Pruesse, G., Ruskey, F.: Generating linear extensions fast. SIAM J. Comput. **23** (1994) 373–386
9. NLANR: (http://www.ircache.net/)
10. IBM Quest Data Mining Project. The Quest retail transaction data generator. http://www.almaden.ibm.com/software/quest/ (1996)
11. Kuok, C.M., Fu, A.W.C., Wong, M.H.: Mining fuzzy association rules in databases. SIGMOD Record **27** (1998) 41–46
12. Au, W.H., Chan, K.C.C.: Mining fuzzy association rules in a bank-account database. IEEE Trans. Fuzzy Systems **11** (2003) 238–248
13. Chen, G., Wei, Q.: Fuzzy association rules and the extended mining algorithms. Inf. Sci. **147** (2002) 201–228
14. Levene, M., Loizou, G.: A Guided Tour of Relational Databases and Beyond. Springer-Verlag (1999)
15. Lu, A., Ng, W.: Handling inconsistency of vague relations with functional dependencies. In: ER. (2007)
16. Lu, A., Ke, Y., Cheng, J., Ng, W.: Mining vague association rules. In: DASFAA. (2007) 891–897