Mining Allocating Patterns in One-sum Weighted Items

Yanbo J. Wang [1], Xinwei Zheng [2], Frans Coenen [1], and Cindy Y. Li [3]
[1] *Department of Computer Science, University of Liverpool, UK*
{Y.J.Wang, Coenen}@liverpool.ac.uk
[2] *School of Accounting Economics and Finance, Deakin University, Australia*
xinwei.zheng@deakin.edu.au
[3] *Histocompatibility and Immunogenetics Laboratory,*
National Blood Service Bristol Centre, UK
Ying.Li@nbs.nhs.uk

Abstract

An Association Rule (AR) is a common knowledge model in data mining that describes an implicative co-occurring relationship between two disjoint sets of binary-valued transaction database attributes (items), expressed in the form of an "antecedent ⇒ consequent" rule. A variant of the AR is the Weighted Association Rule (WAR). With regard to a marketing context, this paper introduces a new knowledge model in data mining — ALlocating Pattern (ALP). An ALP is a special form of WAR, where each rule item is associated with a weighting score between 0 and 1, and the sum of all rule item scores is 1. It can not only indicate the implicative co-occurring relationship between two (disjoint) sets of items in a weighted setting, but also inform the "allocating" relationship among rule items. ALPs can be demonstrated to be applicable in marketing and possibly a surprising variety of other areas. We further propose an Apriori based algorithm to extract hidden and interesting ALPs from a "one-sum" weighted transaction database. The experimental results show the effectiveness of the proposed algorithm.

1. Introduction

Data mining is an area of current research and development in computer science, which is attracting increasing attention from a wide range of different groups of people. It aims to extract various types (models) of hidden, interesting, previously unknown and potentially useful knowledge (i.e. rules, patterns, regularities, customs, trends, etc.) from databases, where the volume of a collected database can be measured in gigabytes. In data mining, common models of mined knowledge include: association rules [1], classification rules [10], prediction rules [8], clustering patterns [9], emerging patterns [6], sequential patterns [13], etc.

Association Rule Mining (ARM) [1] is a well-established data mining technique for the extraction of hidden and interesting patterns called Association Rules (ARs) from a given transaction (basket) database. It deals with binary-valued data attributes (items) only, where all attributes in a transaction database are valued in a *Boolean* manner. An AR describes an implicative co-occurring relationship between two disjoint sets of items, expressed in the form of an "antecedent ⇒ consequent" rule. In a marketing context, a typical AR can be exemplified as "⟨ bread egg milk ⟩ ⇒ ⟨ butter ham ⟩", which can be interpreted as: when people purchase bread, egg and milk together, it is likely that both butter and ham are also purchased.

The original ARM problem treats the importance of all items in a uniform manner. Based on a "real-life" marketing experience, Cai *et al.* [3] indicate that not all goods (items) share the same importance in a market, and introduce the concept of weighted items to improve the applicability of ARs. With regard to a retailing business, mining from weighted items/goods enables the generation of such ARs with more emphasis on some particular goods (e.g. goods that are under promotion, goods that always make significant profits) and less emphasis on other goods. The idea of mining ARs in a special transaction database, where each item is assigned a weighting score, directly depicts the problem of mining Weighted Association Rules (WARs). As a consequence, a number of alternative Weighted Association Rule Mining (WARM) approaches have been developed over the past decade, such as [11, 12].

A special case of WAR can be introduced as the "one-sum" WAR, where each rule item is associated with a weighting score between 0 and 1, and the sum of all rule item scores is 1. A one-sum WAR can not only indicate the implicative co-occurring relationship between two disjoint sets of items in a weighted setting, but also inform the "allocating" relationship among rule items. In a marketing context, an archetypal one-sum WAR can be exemplified as "〈bread[0.15] egg[0.20] milk[0.10]〉 ⇒ 〈butter[0.20] ham [0.35]〉", which can be interpreted as: when people spend 15%, 20% and 10% of their money to purchase bread, egg and milk together, it is likely that people will also spend 20% and 35% of their money to purchase butter and ham. In this paper, we introduce the concept of one-sum WARs, as a new knowledge model in data mining, and name such WARs as ALlocating Patterns (ALPs). We further propose an algorithm, based on the well-established *Apriori* algorithm [2], which effectively extracts hidden and interesting ALPs from a one-sum weighted transaction database. We believe that ALPs can be shown to be useful in a surprising variety of applications other than just marketing.

The rest of this paper is organized as follows. In section 2, we describe some related work relevant to this study, where ARM is reviewed and three of the existing approaches in WARM are outlined. In section 3, the concept of ALP is introduced, based on describing the one-sum weighted: transaction databases, itemsets and WARs. An algorithm for ALlocating Pattern Mining (ALPM) is proposed in section 4. Experimental results are presented in section 5 that demonstrate the effectiveness of the proposed algorithm. Finally our conclusions and open issues for further research are given in section 6.

2. Related Work

2.1. Association Rule Mining

Association Rule Mining (ARM), first introduced in [1], aims to extract a set of ARs from a given transaction database D_T. It is a well-established research field in data mining. Cornelis *et al.* [5] suggest that the concept of mining ARs can be dated back to the work of Hájek *et al.* in 1966 [7]. Let $I = \{a_1, a_2, \ldots, a_{n-1}, a_n\}$ be a set of items (binary-valued database attributes), and $\mathcal{T} = \{T_1, T_2, \ldots, T_{m-1}, T_m\}$ be a set of transactions (database records), D_T is described by $\mathcal{T}$, where each $T_j \in \mathcal{T}$ comprises a set of items $I' \subseteq I$. An AR can be given as "antecedent (X) ⇒ consequent (Y)", where $X, Y \subset I$ and $X \cap Y = \emptyset$. In ARM, two threshold values are usually used to determine the significance of an AR:

1. **Support:** A set of items S is called an itemset. The support of S is the proportion of transactions T in $\mathcal{T}$ for which $S \subseteq T$. If the support of S exceeds a user-supplied support threshold σ, S is defined to be a Frequent Itemset (FI).
2. **Confidence:** Represents how "strongly" an itemset (rule antecedent) X implies another itemset (rule consequent) Y. A confidence threshold α, supplied by the user, is used to distinguish high confidence ARs from low confidence ARs.

An AR "$X \Rightarrow Y$" is said to be *valid* when the support for the co-occurrence of X and Y exceeds σ, and the confidence of this AR exceeds α. The computation of support is:

$$support(X \cup Y) = count(X \cup Y) \;/\; |\mathcal{T}| \,,$$

where $count(X \cup Y)$ is the number of transactions containing the set $X \cup Y$ in $\mathcal{T}$, and $|\mathcal{T}|$ is the size function of the set $\mathcal{T}$. The computation of confidence is:

$$confidence(X \Rightarrow Y) = support(X \cup Y) \;/\; support(X) \,.$$

Algorithm 1: The Apriori Algorithm

```
Input:   (a) A transaction database $D_T$;
         (b) A support threshold $\sigma$;
Output: A set of frequent itemsets $S_{FI}$;
Begin Algorithm:
  $k \leftarrow 1$;
  $S_{FI} \leftarrow$ an empty set for holding the identified frequent
     itemsets;
  generate all candidate 1-itemsets from $D_T$;
  while (candidate $k$-itemsets exist) do
      determine support for candidate $k$-itemsets from $D_T$;
      add frequent $k$-itemsets into $S_{FI}$;
      remove all candidate $k$-itemsets that are not
         sufficiently supported to give frequent $k$-itemsets;
      generate candidate ($k$+1)-itemsets from
         frequent $k$-itemsets using "closure property";
      $k \leftarrow k + 1$;
  end while
  return ($S_{FI}$);
End Algorithm
```

The most well-known ARM algorithm is the Apriori algorithm, developed by Agrawal and Srikant [2], which has been the basis of many subsequent ARM and/or ARM-related algorithms. In [2], it was observed that ARs can be straightforwardly generated from a set of FIs. Thus, efficiently and effectively mining FIs from data is the key to ARM. The Apriori algorithm iteratively identifies FIs in data by employing the "closure property" of itemsets in the generation of candidate itemsets, where a candidate (possibly frequent) itemset is confirmed as frequent only when all its subsets are identified as frequent in the previous pass. The "closure property" of itemsets can be described as follows: if an itemset is frequent then all its subsets will also be frequent; conversely if an itemset is infrequent then all its supersets will also be infrequent. The Apriori algorithm is outlined in Algorithm 1.

2.2. Weighted Association Rule Mining

Weighted Association Rule Mining (WARM), first introduced in [3], aims to apply the concept of weighting into ARM and consequently extract WARs from a weighted transaction database. In the past decade, a number of alternative WARM approaches have been introduced. Three major studies can be described as follows.

2.2.1. The Traditional Approach. Cai *et al.* [3] introduce the concept of weighted items and the weighted transaction database $D^W{}_T$. Let $I^W = \{a^W{}_1, a^W{}_2, \ldots, a^W{}_{n-1}, a^W{}_n\}$ be a set of weighted items, where each $a^W{}_i \in I^W$ is an item $a_i \in I$ (see section 2.1) labeling with a user-defined weighting score w_i ($0 \leq w_i \leq 1$). Let Ŧ $= \{T_1, T_2, \ldots, T_{m-1}, T_m\}$ be a set of transactions, $D^W{}_T$ is described by Ŧ, where each $T_j \in$ Ŧ comprises a set of weighted items $I^{W}{}' \subseteq I^W$. To measure the significance of a WAR, the "weighted-support — weighted-confidence" approach, an extension of the "support — confidence" framework (as described in section 2.1), was introduced in [3]. A weighted support threshold σ^W is supplied by the user that distinguishes frequent weighted itemsets from the infrequent ones. A weighted itemset $X^W \cup Y^W$ is considered to be frequent if $(\sum_{\{a^W{}_i \in (X^W \cup Y^W)\}} w_i) \times support(X^W \cup Y^W) \geq \sigma^W$, where $X^W, Y^W \subset I^W$ and $X^W \cap Y^W = \varnothing$. Having a set of frequent weighted itemsets generated from $D^W{}_T$, a set of WARs can be further obtained. A WAR "$X^W \Rightarrow Y^W$" is said to be valid when $X^W \cup Y^W$ is frequent, and $((\sum_{\{a^W{}_i \in (X^W \cup Y^W)\}} w_i) \times support(X^W \cup Y^W)) / ((\sum_{\{a^W{}_i \in X^W)\}} w_i) \times support(X^W)) \geq \alpha^W$, where α^W is a user-defined weighted confidence threshold.

2.2.2. The Variant Approach. Wang *et al.* [12] propose an alternative approach of mining WARs by introducing a variant weighted transaction database $D^W{}_T{}^*$. With regard to real-life marketing, the newly mined WARs "*can not only improve the confidence in the rules, but also provide a mechanism to do more effective target marketing by identifying or segmenting customers based on their potential degree of loyalty or volume of purchases*" [12]. In Table 1 several points, in terms of item weighting score properties, that differentiate $D^W{}_T{}^*$ from $D^W{}_T$ are listed.

Table 1. The difference between $D^W{}_T$ and $D^W{}_T{}^*$

Properties of Item Weighting Scores	$\boldsymbol{D^W{}_T}$	$\boldsymbol{D^W{}_T{}^*}$
Single-value like **vs.** **Interval-value like**	The weighting score of an item in $D^W{}_T$ is given as a single value v. The weighting score is defined as *single-value like*.	The weighting score of an item in $D^W{}_T{}^*$ is given as an interval of two values $[v_1, v_2]$, where $v_1 < v_2$. The weighting score is defined as *interval-value like*.
Percentage like **vs.** **Positive-integer like**	The value of the weighting score for an item in $D^W{}_T$ is given as $0 \leq v \leq 1$. The weighting score is defined as *percentage like*.	Both lower and upper values of the weighting score interval for an item in $D^W{}_T{}^*$ are given as $v_1, v_2 \geq 1$ and $v_1, v_2 \in \mathrm{Z}$ (both v_1, v_2 are positive integers). The weighting score is defined as *positive-integer like*.
Static like **vs.** **Dynamic like**	The weighting score of an item in $D^W{}_T$ is given as a fixed value in all transactions. The weighting score is defined as *static like.*	The weighting score of an item in $D^W{}_T{}^*$ can be valued differently in different transactions. The weighting score is defined as *dynamic like.*

In a marketing context, a typical WAR mined from $D^W{}_T{}^*$ can be exemplified as "⟨ bread[9, 14] ⟩ ⇒ ⟨ ham[12, 20] ⟩", which can be interpreted as: when bread is purchased in the quantity between 9 and 14, it is likely that ham in the quantity between 12 and 20 is also purchased. In [12] the proposed WAR generation approach comprises two phases: (1) generating a set of frequent itemsets from $D^W{}_T{}^*$ regardless the weighting issue; and (2) extracting hidden and interesting WARs based on (1). In (2) a set of candidate rules can be enumerated from the result of (1), where the consequent of each candidate rule "*only contains one weighted item for the sake of simplicity*" [12]. A number of "qualified" WARs can be further identified in the set of candidate rules with respect to the user-specified threshold values of support, confidence and density. Since this study is direct at producing maximum rules only, a set of maximum WARs — "*a qualified WAR X ⇒ Y is a maximum WAR if for any generalization X' of X and Y' of Y where X' ≠ X and Y' ≠ Y, neither of X'⇒ Y, X ⇒ Y', nor X'⇒ Y' is a qualified WAR*" [12] — is finally obtained. In [11] Tao *et al.* classify the process of mining WARs from $D^W{}_T{}^*$, proposed in [12], as a technique of post-processing or maintaining ARs.

2.2.3. The Improved Approach. Tao *et al.* [11] identify the main challenge of mining WARs: the closure property of itemsets (see section 2.1) is invalid in the generation of significant/frequent weighted itemsets. To solve this problem, an improved approach of mining WARs was proposed in [11], which takes an alternative weighted transaction database $D^{W}{}_{T}{}^{+}$ as the input. The only difference between $D^{W}{}_{T}{}^{+}$ and $D^{W}{}_{T}$ is that the item weighting scores in $D^{W}{}_{T}{}^{+}$ can be valued as any positive real number, whereas the item weighting scores in $D^{W}{}_{T}$ are valued between 0 and 1, i.e. "percentage like". This improved approach automatically assigns a weighting score w_t_j to each transaction T_j in $D^{W}{}_{T}{}^{+}$, where the computation of w_t_j is: $(\sum_{\{a^{W}{}_{i} \in T_j\}} w_i) / |T_j|$. Based on the assigned transaction scores, a set of frequent weighted itemsets $S_{FI^{W}}$ can be generated. A weighted itemset $X^W \cup Y^W$ is considered to be frequent if $(\sum_{\{j = 1\ldots|\text{Ŧ}|\ \&\ (X^W \cup Y^W) \subseteq T_j\}} w\_t_j) / (\sum_{\{j = 1\ldots|\text{Ŧ}|\}} w\_t_j) \geq \sigma^W$, where X^W, $Y^W \subset I^W$, $X^W \cap Y^W = \emptyset$, and σ^W is a user-supplied weighted support threshold. In the generation of frequent weighted itemsets, the closure property can be proven work properly. With respect to the idea presented in [2], all WARs can be further mined from $S_{FI^{W}}$.

3. Allocating Patterns

A new type of WAR, namely ALlocating Pattern (ALP), is designed in this section. As mentioned in section 1, an ALP can not only indicate the implicative co-occurring relationship between two (disjoint) sets of items in a weighted setting, but can also inform the allocating relationship among AR items. In a marketing application, ALPs can be used to show individual customer habits of allocating an amount of money to a variety of goods. This can be further used in sales and goods promotion, customer segmentation, transaction classification, etc. We would like to expect that ALPs may be proven to be applicable in a wide range of fields other than marketing related situations. The approach of mining ALPs requires a special weighted transaction database $D^{W}{}_{T\text{-}OS}$ as the input.

3.1. One-sum Weighted Transaction Database

In Table 1 three sets of item score properties were defined to analyze different weighted transaction databases. These properties are "single-value like vs. interval-value like", "percentage like vs. positive-integer like", and "static like vs. dynamic like". In $D^{W}{}_{T\text{-}OS}$ item weighing scores show an additional property ("one-sum" like) that distinguishes $D^{W}{}_{T\text{-}OS}$ from other weighted transaction databases — the sum of all item scores in each transaction is 1. Hence $D^{W}{}_{T\text{-}OS}$ can be referred to as a "one-sum" weighted transaction database.

Let $I^{OSW} = \{a^{OSW}{}_{1}, a^{OSW}{}_{2}, \ldots, a^{OSW}{}_{n-1}, a^{OSW}{}_{n}\}$ be a set of one-sum weighted items, and $\text{Ŧ} = \{T_1, T_2, \ldots, T_{m-1}, T_m\}$ be a set of transactions. Each $a^{OSW}{}_{i} \in I^{OSW}$ represents an item $a_i \in I$ (see section 2.1) that is assigned a set of weighting scores $\theta_i = \{w_{i_1}, w_{i_2}, \ldots, w_{i_{m-1}}, w_{i_m}\}$, where $0 \leq w_{i_j} \leq 1$ and $|\theta_i| = |\text{Ŧ}|$ which means: for different transactions $T_j \in \text{Ŧ}$, different scores $w_{i_j} \in \theta_i$ can be assigned to a particular item $a^{OSW}{}_{i} \in I^{OSW}$. A one-sum weighted transaction database $D^{W}{}_{T\text{-}OS}$ is described by Ŧ, where each $T_j \in \text{Ŧ}$ comprises a set of one-sum weighted items $I^{OSW\prime} \subseteq I^{OSW}$, and $\sum_{\{i = 1\ldots|I^{OSW\prime}| \text{ or } |T_j|\}} w_{j_i} = 1$. An overall comparison, in terms of item weighting score properties, of four different weighted transaction databases is provided in Table 2.

Table 2. The comparison of $D^{W}{}_{T}$, $D^{W}{}_{T}{}^{*}$, $D^{W}{}_{T}{}^{+}$ and $D^{W}{}_{T\text{-}OS}$

Properties of Item Weighting Scores	$D^{W}{}_{T}$	$D^{W}{}_{T}{}^{*}$	$D^{W}{}_{T}{}^{+}$	$D^{W}{}_{T\text{-}OS}$
Single-value like vs. Interval-value like	*Single-value like*	*Interval-value like*	*Single-value like*	*Single-value like*
Percentage like vs. Positive-integer / Positive-real like	*Percentage like*	*Positive-integer like*	*Positive-real like*	*Percentage like*
Static like vs. Dynamic like	*Static like*	*Dynamic like*	*Static like*	*Dynamic like*
One-sum like	*No*	*No*	*No*	*Yes*

3.2. One-sum Weighted Itemsets

An itemset can be recognized in a transaction database D_T if this particular set of items appears as a subset of at least one transaction T_j in D_T. A one-sum weighted itemset can be treated as an itemset that is presented in a particular weighting frame, where the item scores are assigned in a one-sum percentage manner. For example, $\{I_1[0.1], I_2[0.3], I_3[0.3], I_5[0.3]\}$ and $\{I_1[0.1], I_2[0.3], I_3[0.5], I_5[0.1]\}$ are two different weighting frames for the itemset $\{I_1, I_2, I_3, I_5\}$. An itemset can produce as many as infinity possible weighting frames. If an itemset weighting frame *IWF* appears as a subset of at least one transaction T_j in a one-sum weighted transaction database $D^{W}{}_{T\text{-}OS}$, this *IWF* can be identified as a one-sum weighted itemset in $D^{W}{}_{T\text{-}OS}$.

3.2.1. The Score Transformation Procedure. To determine whether an *IWF* is a subset of a particular T_j in $D^{W}{}_{T\text{-}OS}$ or not, the actual weighting score w_{j_i} that is assigned to each item $a^{OSW}{}_i \in T_j$ where $a^{OSW}{}_i \in IWF$ needs to be transformed as: $(w_{j_i}) / (\sum_{\{q = 1\ldots|T_j|\ \&\ (a^{OSW}{}_q \in IWF)\}} w_{j_q} \in T_j)$. The transformed scores clarify the actual allocating relationship among these *IWF*-related items in T_j. An *IWF* is defined as a subset of T_j if the score of each item involved in *IWF* matches the relative item score transformed in T_j. For example, an *IWF* can be given as $\{I_1[0.4], I_2[0.2], I_3[0.4]\}$ while a transaction T_j may be $\{I_1[0.2], I_2[0.1], I_3[0.2], I_4[0.25], I_5[0.25]\}$; the weighing scores for items I_1, I_2 and I_3 are grouped since the item intersection $IWF \cap T_j = \{I_1, I_2, I_3\}$; although the actual scores of I_1, I_2 and I_3 are presented differently in *IWF* (as "0.4", "0.2" and "0.4") and T_j (as "0.2", "0.1" and "0.2"), *IWF* is still a subset of T_j because the transformed scores of I_1, I_2 and $I_3 \in T_j$ are computed as "0.2 / (0.2 + 0.1 + 0.2) = 0.4", "0.1 / (0.2 + 0.1 + 0.2) = 0.2" and "0.2 / (0.2 + 0.1 + 0.2) = 0.4", and these match the scores given in *IWF*. The transformation of transaction item scores enables the one-sum weighted property to be translated from transactions to the extracted weighted itemsets.

3.2.2. Frequent One-sum Weighted Itemsets. A one-sum weighted itemset is considered to be frequent if it can be found as a subset of more than $(\sigma^W{}_{OS} \times |Ŧ|)$-many transactions in $D^W{}_{T\text{-}OS}$, where $\sigma^W{}_{OS}$ is a user-supplied one-sum weighted support threshold. The closure property of itemsets can also be observed in one-sum weighted itemsets, so that: if a one-sum weighted itemset is frequent then all its subsets will also be frequent; conversely if a one-sum weighted itemset is infrequent then all its supersets will also be infrequent.

3.3. One-sum Weighted Association Rules

A frequent one-sum weighted itemset is presented as $X^{OSW} \cup Y^{OSW}$, where X^{OSW}, $Y^{OSW} \subset I^{OSW}$ and $X^{OSW} \cap Y^{OSW} = \varnothing$. A one-sum WAR in the form of "$X^{OSW} \Rightarrow Y^{OSW}$" can be subsequently produced by a rule formalization procedure, namely Rule-Formalization (see Algorithm 2). In Rule-Formalization, $w(a^{OSW}{}_i) \in (X^{OSW} \cup Y^{OSW})$ represents the corresponding (actual) weighting score for the item $a^{OSW}{}_i$ in $X^{OSW} \cup Y^{OSW}$.

A one-sum WAR "$X^{OSW} \Rightarrow Y^{OSW}$" is said to be valid when $count((X^{OSW} \cup Y^{OSW}) \subseteq (T_j \in Ŧ)) / count(X^{OSW} \subseteq (T_j \in Ŧ)) \geq \alpha^W{}_{OS}$, where $\alpha^W{}_{OS}$ is a user-supplied one-sum weighted confidence threshold, *count*(*J*) is the count function that returns the number of occurrences of an object *J*, and the previously described score transformation procedure is employed to verify the "$\subseteq$" relationship.

Algorithm 2: The Rule-Formalization Procedure

```
Input:  A frequent one-sum weighted itemset in terms of
            ($X^{OSW}$, $Y^{OSW}$);
Output: A formalized one-sum weighted association rule $p$
            (as "$X^{OSW} \Rightarrow Y^{OSW}$");
Begin Algorithm:
    prepare $p$ to be a formalized one-sum weighted
        association rule;
    formalize "$\langle$" as the first part of $p$;
    for each $a^{OSW}{}_i \in X^{OSW}$ do
        update $p$ iteratively by formalizing " $a^{OSW}{}_i$ '[' $w(a^{OSW}{}_i)$
            $\in (X^{OSW} \cup Y^{OSW})$ ']' " as its second part;
    end for
    update $p$ by formalizing "$\rangle \Rightarrow \langle$" as its third part;
    for each $a^{OSW}{}_i \in Y^{OSW}$ do
        update $p$ iteratively by formalizing " $a^{OSW}{}_i$ '[' $w(a^{OSW}{}_i)$
            $\in (X^{OSW} \cup Y^{OSW})$ ']' " as its fourth part;
    end for
    update $p$ by formalizing "$\rangle$" as its last part;
    return ($p$);
End Algorithm
```

4. Allocating Pattern Mining

In this section, an ALlocating Pattern Mining (ALPM) approach is proposed to extract all hidden and interesting ALPs from a one-sum weighted transaction database $D^W{}_{T\text{-}OS}$. With respect to the traditional ARM approach presented in [2], the proposed ALPM method consists of two phases: (1) generating a set of frequent one-sum weighted itemsets from $D^W{}_{T\text{-}OS}$; and (2) mining one-sum WARs (noted as ALPs) based on (1).

4.1. Generating Frequent One-sum Weighted Itemsets

An algorithm, namely Apriori-ALP, is proposed to generate a set of frequent one-sum weighted itemsets from $D^W{}_{T\text{-}OS}$, which takes the Apriori algorithm (see Algorithm 1) as its basis. A one-sum weighted support threshold $\sigma^W{}_{OS}$, as a parameter of Apriori-ALP, is taken from the user. The Apriori-ALP algorithm is presented (see Algorithm 3).

4.2. Generating One-sum WARs (ALPs)

Given a set of frequent one-sum weighted itemsets $S_{FI^W{}_{OS}}$ that is generated by Apriori-ALP, an algorithm, namely ALP-Generation, is further proposed to extract ALPs from $S_{FI^W{}_{OS}}$. A one-sum weighted confidence threshold $\alpha^W{}_{OS}$, as a parameter of ALP-Generation, is taken from the user. According to the closure property of one-sum weighted itemsets, all subsets of a frequent one-sum weighted itemset f_i are included in $S_{FI^W{}_{OS}}$, where $|f_i| \geq 2$. Hence the process of ALP-Generation can be designed based on the closure property (see Algorithm 4).

Algorithm 3: The Apriori-ALP Algorithm

```
Input:  (a) A one-sum weighted transaction database $D^W{}_{T\text{-}OS}$;
        (b) A one-sum weighted support threshold $\sigma^W{}_{OS}$;
Output: A set of frequent one-sum weighted itemsets $S_{FI^W{}_{OS}}$;
Begin Algorithm:
  $k \leftarrow 1$;
  $S_{FI^W{}_{OS}} \leftarrow$ an empty set for holding the identified frequent
      one-sum weighted itemsets;
  $C_k \leftarrow$ generate the set of candidate $k$-itemsets from $D^W{}_{T\text{-}OS}$;
  while ($C_k \neq \emptyset$) do
      for each element $e_i \in C_k$ do
          generate all itemset weighting frames (IWFs) for $e_i$
              through scanning all transactions in $D^W{}_{T\text{-}OS}$;
          initialize a Boolean variable frequentFlag as false;
          for each IWF $f_j \in e_i$ do
              support $\leftarrow$ count($f_j \subseteq$ transactions in $D^W{}_{T\text{-}OS}$);
                  // the score transformation procedure (see section
                     3.2.1) is employed to verify the "$\subseteq$" relationship
              if ((support / $|D^W{}_{T\text{-}OS}|$) $\geq \sigma^W{}_{OS}$) then
                  add $f_j$ into $S_{FI^W{}_{OS}}$;
                     // $f_j$ is stored with its actual support value
                  set frequentFlag to be true;
          end for
          if ($\neg$frequentFlag) then
              remove $e_i$ from $C_k$;
      end for
      $k \leftarrow k + 1$;
      $C_k \leftarrow$ generate the set of candidate $k$-itemsets from
          frequent ($k$–1)-itemsets using "closure property";
  end while
  return ($S_{FI^W{}_{OS}}$);
End Algorithm
```

Algorithm 4: The ALP-Generation Algorithm

```
Input:  (a) A set of frequent one-sum weighted itemsets
            $S_{FI^W{}_{OS}}$;
        (b) A one-sum weighted confidence threshold $\alpha^W{}_{OS}$;
Output: A set of allocating patterns $S_{ALP}$;
Begin Algorithm:
  $S_{ALP} \leftarrow$ an empty set for holding the identified allocating
      patterns;
  for each frequent one-sum weighted itemset $f_i \in S_{FI^W{}_{OS}}$ do
      for each frequent one-sum weighted itemset $f_j \in S_{FI^W{}_{OS}}$ do
          if ($f_j \subset f_i$) then // the score transformation procedure
             (see section 3.2.1) is employed to verify the "$\subset$"
             relationship
           confidence $\leftarrow f_i$.support / $f_j$.support;
           if (confidence $\geq \alpha^W{}_{OS}$) then
               allocating pattern $p$
                  $\leftarrow$ Rule-Formalization($f_j$, $f_i - f_j$);
               add $p$ into $S_{ALP}$;
      end for
  end for
  return ($S_{ALP}$);
End Algorithm
```

5. Results

In this section, we aim to show the effectiveness of the proposed ALPM approach. First of all, a one-sum weighted "shopping-basket" (transaction) database was simulated in a two-stage process. In the first stage, a traditional transaction database D_T was generated using the QUEST generator described in [2]. This defines four parameters:

- N — the number of attributes (items) in D_T;
- D — the number of records (transactions) in D_T;
- T — the average number of items in a transaction; and
- I — the largest number of items expected to be found in a frequent itemset.

In a marketing context, it can be assumed that a small-sized supermarket (or convenience store) contains about 100 distinct categories of goods (i.e. N = 100); and that there are 300 ~ 350 customers (transactions) per day, so that in 1-month period there are around 10,000 transactions (i.e. D = 10,000); in average each transaction involves 10 goods (i.e. T = 10); and we expect that I = 5. Note that T = 10 and I = 5 were also used in [4] to simulate a set of "shopping-basket" data. As a result of this stage, a transaction database T10.I5.N100.D10000 was produced.

Table 3. List the top 10 and the bottom 10 mined ALPs (based on confidence)

No.	ALPs mined from T10.I5.N100.D10000.W3	Conf.
1	⟨ 13[0.25] 72[0.25] ⟩ ⇒ ⟨ 22[0.5] ⟩	0.322493
2	⟨ 9[0.2] 56[0.4] ⟩ ⇒ ⟨ 74[0.4] ⟩	0.314868
3	⟨ 74[0.25] 94[0.5] ⟩ ⇒ ⟨ 22[0.25] ⟩	0.313351
4	⟨ 9[0.4] 70[0.4] ⟩ ⇒ ⟨ 74[0.2] ⟩	0.310769
5	⟨ 22[0.25] 70[0.5] ⟩ ⇒ ⟨ 9[0.25] ⟩	0.310240
6	⟨ 13[0.249999] 74[0.249999] ⟩ ⇒ ⟨ 22[0.500001] ⟩	0.306701
7	⟨ 39[0.4] 74[0.199998] ⟩ ⇒ ⟨ 46[0.4] ⟩	0.305389
8	⟨ 9[0.5] 13[0.25] ⟩ ⇒ ⟨ 22[0.25] ⟩	0.304216
9	⟨ 26[0.500002] 74[0.249998] ⟩ ⇒ ⟨ 22[0.249998] ⟩	0.301724
10	⟨ 39[0.4] 46[0.4] ⟩ ⇒ ⟨ 74[0.199998] ⟩	0.3
…	…	…
69	⟨ 22[0.249998] 46[0.249998] ⟩ ⇒ ⟨ 9[0.500002] ⟩	0.229729
70	⟨ 46[0.4] 74[0.199998] ⟩ ⇒ ⟨ 9[0.4] ⟩	0.228310
71	⟨ 22[0.249998] 74[0.249998] ⟩ ⇒ ⟨ 71[0.500002] ⟩	0.226611
72	⟨ 22[0.199998] 46[0.4] ⟩ ⇒ ⟨ 13[0.4] ⟩	0.226215
73	⟨ 22[0.4] 74[0.199998] ⟩ ⇒ ⟨ 9[0.4] ⟩	0.221757
74	⟨ 22[0.249998] 74[0.249998] ⟩ ⇒ ⟨ 26[0.500002] ⟩	0.218295
75	⟨ 22[0.400001] 74[0.400001] ⟩ ⇒ ⟨ 98[0.199997] ⟩	0.207900
76	⟨ 22[0.4] 74[0.4] ⟩ ⇒ ⟨ 71[0.199998] ⟩	0.207900
77	⟨ 90[0.333331] ⟩ ⇒ ⟨ 74[0.666668] ⟩	0.207897
78	⟨ 90[0.5] ⟩ ⇒ ⟨ 22[0.5] ⟩	0.200929

In the second stage of the database simulation, the one-sum weighting score was assigned to each transaction item, which simulates the customer habits of allocating their money to different goods. Firstly, an integer ω_i was given to each item a_i in a transaction T_j (in T10.I5.N100.D10000), where ω_i is randomly chosen from {1, 2, 3}. Secondly, the one-sum weighting score w_i for a_i was then calculated as: ω_i / ($\sum_{\{k = 1 \ldots |T_j|\}} \omega_k$). As a consequence, the simulated one-sum weighted "shopping-basket" database, namely

T10.I5.N100.D10000.W3, was generated, where W denotes the size of the random integer set in item (one-sum) weighting.

A set of ALPs were then mined from T10.I5.N100.D10000.W3, using the proposed ALPM method that has been implemented as a standard Java program. The experiments were run on a 1.87 GHz Intel(R) Core(TM)2 CPU with 2.00 GB of RAM running under Unix operating system. With regard to a one-sum weighted support threshold value of 1% and a one-sum weighted confidence threshold value of 20%, 78 ALPs were extracted. We ordered these ALPs based on their confidence value (in a descending manner); the top 10 and the bottom 10 ALPs are presented in Table 3. Note that in Table 3 the integers shown before the square brackets are the item ID-numbers, and the real (decimal) numbers shown in the square brackets represent the item one-sum weights.

6. Conclusions

This paper is concerned with the design of a new knowledge model in data mining — ALlocating Pattern (ALP). The concept of ALPs can be seen as an extension of the well-established Association Rules in a special weighted setting. In this paper, the applicability of mining ALPs in marketing related situations has been stated. We expect that ALPs may be further proven applicable in a surprising variety of areas/fields.

An overview of the traditional Association Rule Mining approach and three major Weighted Association Rule Mining studies was provided in section 2. The newly designed ALP concept was presented in section 3. In section 4 an Apriori based method was proposed to identify hidden and interesting ALPs in data. From the experimental results, the effectiveness of the proposed ALlocating Pattern Mining (ALPM) method was demonstrated with respect to a simulated one-sum weighted "shopping-basket" database.

Further research is suggested to develop improved ALPM approaches with respect to the efficiency. Another direction of the future work is to explore the wide applicability of this new knowledge model.

7. Acknowledgement

The authors would like to thank Professor Paul Leng, Dr. Robert Sanderson, Kamal Ali Albashiri, Chuntao Jiang and Maya Wardeh of the Department of Computer Science at the University of Liverpool for their support with respect to the work described here.

8. References

[1] R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules between Sets of Items in Large Databases", In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, ACM Press, Washington, DC, USA, May 1993, pp. 207-216.

[2] R. Agrawal, and R. Srikant, "Fast Algorithms for Mining Association Rules", In *Proceedings of the 20th International Conference on Very Large Data Bases*, Morgan Kaufmann Publishers, Santiago de Chile, Chile, September 1994, pp. 487-499.

[3] C.H. Cai, A.W.C. Fu, C.H. Chen, and W.W. Kwong, "Mining Association Rules with Weighted Items", In *Proceedings of the 1998 International Database Engineering and Application Symposium*, IEEE Computer Society, Cardiff, Wales, UK, July 1998, pp. 68-77.

[4] F. Coenen, and P. Leng, "Finding Association Rules with Some Very Frequent Attributes", In *Proceedings of the 6th European Conference on Principles and Practice of Knowledge Discovery in Databases*, Springer-Verlag, Helsinki, Finland, August 2002, pp. 99-111.

[5] C. Cornelis, P. Yan, X. Zhang, and G. Chen, "Mining Positive and Negative Association Rules from Large Databases", In *Proceedings of the 2006 IEEE International Conference on Cybernetics and Intelligent Systems*, IEEE Computer Society, Bangkok, Thailand, June 2006, pp. 613-618.

[6] G. Dong, and J. Li, "Efficient Mining of Emerging Patterns: Discovering Trends and Differences", In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM Press, San Diago, CA, USA, August 1999, pp. 43-52.

[7] P. Hájek, I. Havel, and M. Chytil, "The GUHA Method of Automatic Hypotheses Determination", *Computing* 1, 1966, pp. 293-308.

[8] Han, J., and M. Kamber, *Data Mining: Concepts and Techniques (Second Edition)*, Morgan Kaufmann Publishers, San Francisco, CA, USA, 2006.

[9] Mirkin, B., and B.G. Mirkin, *Clustering for Data Mining: A Data Recovery Approach*, Chapman & Hall / CRC, Virginia Beach, VA, USA, 2005.

[10] Quinlan, J.R., *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Francisco, CA, USA, 1993.

[11] F. Tao, F. Murtagh, and M. Farid, "Weighted Association Rule Mining using Weighted Support and Significance Framework", In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ACM Press, Washington, DC, USA, August 2003, pp. 661-666.

[12] W. Wang, J. Yang, and P. Yu, "Efficient Mining of Weighted Association Rules (WAR)", In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ACM Press, Boston, MA, USA, August 2000, pp. 270-274.

[13] Wang, W. and J. Yang, *Mining Sequential Patterns from Large Data Sets*, Springer-Verlag, 2005.