IEEE ICDM-FDM-08 Workshop, Pisa, Italy

Mining Allocating Patterns in One-sum Weighted Items

Yanbo J. Wang [1], Xinwei Zheng [2], Frans Coenen [3], and Cindy Y. Li [4]

[1] China Minsheng Banking Corp., Ltd., China
[2] Deakin University, Australia
[3] The University of Liverpool, UK
[4] National Blood Service, Bristol Centre, UK

December 2008

Presentation Structure

- Motivation.

- Previous work:
  - Overview
  - Utility ARM [A]
  - (Dynamic) Weighted ARM (WARM) [B].
  - Downward Closure Property WARM [C].

- New Approach (Allocating Pattern Mining):
  - Overview
  - Weighting Frames
  - Algorithm

- Summary and Conclusions

Motivation

- We are interested in mining patterns from data which demonstrate how some resource is allocated across different items.

- A trivial example of such pattern might be:

  { bread[0.15], egg[0.20], milk[0.10] }
  $\Rightarrow$ **{ 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.

- This pattern can be recognised as a *quasi* (weighted) association rule with a special weighted setting.

Weighted Association Rule Mining

- The original ARM problem (Cai et al 1998) treats the importance of all items in a uniform manner. Based on "real-life" marketing experience, not all goods (items) share the same importance in a market.
- Weighted Association Rules (WARs), as a variant of ARs, was introduced to improve the applicability of the AR.
- Weighted Association Rule Mining (WARM) aims to extract WARs from weighted transaction database.
- There are a number of different approaches to WARM reported in the literature.

Weighted Association Rule Mining [A_1]

- The "Utility" Approach (static weighting)
  - $I^W = \{a^W{}_1, a^W{}_2, \ldots, a^W{}_{n-1}, a^W{}_n\}$ be a set of weighted items 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 in a weighted transaction database $D^W{}_T$ where each $T_j \in Ŧ$ comprises a set of weighted items $I^{W\prime} \subseteq I^W$.
  - To measure the significance of a WAR some "weighted-support-confidence" framework is introduced:

    (1) A weighted-support threshold σ^W to distinguishes frequent weighted itemsets from the infrequent ones.

    (2) A weighted-confidence threshold α^W to distinguishes high confidence WARs from low confidence ones.

Weighted Association Rule Mining [A_2]

- A WAR "$X^W \Rightarrow Y^W$" (where $X^W, Y^W \subset I^W$ and $X^W \cap Y^W = \emptyset$) is said to be *valid* when the weighted-support of $X^W \cup Y^W$ exceeds σ^W, and the weighted-confidence of this WAR exceeds α^W.

- The computation of weighted-support is:

  weighted-support($X^W \cup Y^W$) =

  $\left(\sum_{\{a^W_i \in (X^W \cup Y^W)\}} w_i\right) \times$ count ($X^W \cup Y^W$) .

- The computation of weighted-confidence is:

  weighted-confidence($X^W \Rightarrow Y^W$) =

  weighted-support($X^W \cup Y^W$) / weighted-support(X^W))

  .

- Mining from weighted items/goods (in D^W_T) enables the generation of rules (i.e. WARs) with have more emphasis on some particular items and less emphasis on other items.

Weighted Association Rule Mining [B_1]

- The "Variant" Approach (dynamic weighting)
  - The Variant Approach (Wang et al. 2000) to mining WARs is directed at dynamically weighted transaction database $D^W{}_T$*.
  - In a marketing context, an archetypal 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. Dynamic weightings do not have to be ranges.

Weighted Association Rule Mining $[C_1]$

- Improved Approach (with downward closure property)
  - The main challenge of mining WARs is that the "downward closure property" of itemsets no longer holds.
  - To solve this problem, an improved approach of mining WARs was introduced, which takes an alternative weighted transaction database $D^{W}{}_{T}{}^{+}$ as the input.
  - The weighting scores in $D^{W}{}_{T}{}^{+}$ can be any positive real number.

Weighted Association Rule Mining [C_2]

- The Improved WARM assigns a weighting score w_t_j to each transaction T_j in $D^W{}_T{}^+$, where the computation of w_t_j is:

$$w\_t_j = (\sum_{\{a^w{}_i \in T_j\}} w_i) / |T_j| .$$

- A WAR $X^W \Rightarrow Y^W$ (where X^W, $Y^W \subset I^W$ and $X^W \cap Y^W = \emptyset$) is said to be *valid* when the weighted-support of $X^W \cup Y^W$ exceeds the weighted-support threshold σ^W, and the weighted-confidence exceeds the weighted-confidence threshold α^W.

Weighted Association Rule Mining [C_3]

- The computation of weighted-support herein is:

$$\text{weighted-support}^{+}(X^{w} \cup Y^{w}) = (\Sigma\{ j = 1\ldots|Ŧ| \ \&\ (X^{w} \cup Y^{w}) \subseteq T_j\} \ w\_t_j) \ / \ (\Sigma\{ j = 1\ldots|Ŧ|\} \ w\_t_j)$$

- The computation of weighted-confidence herein is:

$$\text{weighted-confidence}^{+}(X^{w} \Rightarrow Y^{w}) = \text{weighted-support}^{+}(X^{w} \cup Y^{w}) \ / \ \text{weighted-support}^{+}(X^{w})$$

- For the generation of frequent weighted itemsets, the "downward closure property" holds.

Allocating Pattern Mining 1

- In our study, we present a different type 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.
- This patterns produced indicate both the implicative co-occurring relationship between two (disjoint) sets of items in a weighted setting, and the "allocating" relationship among rule items (i.e. how a resource is allocated across items).
- We name this new pattern to be an ALlocating Pattern (or ALP).
- The approach of mining ALPs requires a special weighted transaction database, "*One-sum*" Weighted Transaction Database ($D^{W}_{T\text{-}OS}$), as the input.

Allocating Pattern Mining 2

- 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 $Ŧ = \{T_1, T_2, \ldots, T_{m-1}, T_m\}$ be a set of transactions.

- Each $a^{OSW}_i \in I^{OSW}$ represents an item $a_i \in I$ that is assigned a set of weighting scores $\theta_i = \{w_i1, w_i2, \ldots, w_im\text{-}1, w_im\}$, where $0 \le w_ij \le 1$ and $|\theta_i| = |Ŧ|$ which means: for each transactions $T_j \in Ŧ$, different scores $w_ij \in \theta_i$ can be assigned to a particular item $a^{OSW}_i \in I^{OSW}$.

- The resulting one-sum weighted transaction database $D^W_{T\text{-}OS}$ is described by Ŧ, where each $T_j \in Ŧ$ comprises a set of one-sum weighted items $I^{OSW\prime} \subseteq I^{OSW}$, and $\sum_{\{i = 1\ldots|T_j|\}} w_ij = 1$ (the sum of all item scores in each transaction is 1).

- The "one-sum" property serves to distinguishes $D^W_{T\text{-}OS}$ from other

Allocating Pattern Mining (Weighting Frames) $_3$

- Mining Frequent One-sum Weighted Itemsets
  - A one-sum weighted itemset can be treated as an itemset that is presented in a particular <u>weighting frame</u>, where the item scores are assigned in a one-sum "percentage" manner.
  - For example, $\{I_1[0.1], I_2[0.3], I_3[0.3], I_4[0.3]\}$ and $\{I_1[0.1], I_2[0.3], I_3[0.5], I_4[0.1]\}$ are two different weighting frames for the itemset $\{I_1, I_2, I_3, I_4\}$.
  - If an <u>Itemset Weighting Frame</u> (IWF) appears as a subset of more than $(\sigma^{W}_{OS} \times |Ŧ|)$ transactions in $D^{W}_{T\text{-}OS}$, where σ^{W}_{OS} is a user-supplied one-sum weighted-support threshold, this IWF can be identified as a frequent one-sum weighted itemset.

Allocating Pattern Mining 4

- To determine whether an Item Weighting Frame (IWF) is a subset of a particular T_j in $D^W_{T\text{-}OS}$ or not, a **Score Transformation Procedure** is applied to transfer the actual weighting score $w_i j$ for each item $a^{OSW}_i \in T_j$ where $a^{OSW}_i \in$ IWF to a new score. The computation of new weighting score is:

$$\mathbf{new\_score_i j = (w_i j) / (\Sigma\{ q = 1 \ldots |T_j| \ \& \ a^{OSW}_q \in IWF\} \ w_q j \in T_j)}\ .$$

- An IWF is defined as a subset of T_j if the score of each item involved in IWF matches the relative item (new) score transformed in T_j.
- Best illustrated using an example.

Example:

- $IWF = \{I_1[0.4], I_2[0.2], I_3[0.4]\}$
- $T_j = \{I_1[0.2], I_2[0.1], I_3[0.2], I_4[0.25], I_5[0.25]\}$
- The weighting scores for items I_1, I_2 and I_3 are grouped since the item intersection $IWF \cap T_j = \{I_1, I_2, I_3\}$.
- 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 (new) 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.
- (Same distribution).

Allocating Pattern Mining 5

The algorithm to generate one-Sum Frequent weighted Itemsets (SFIs)

Input: (a) A one-sum weighted transaction database $D^{W}_{T\text{-}OS}$;
(b) A one-sum weighted-support threshold σ^{W}_{OS};
Output: A set of frequent one-sum weighted itemsets SFI^{W}_{OS};

k ← 1;
SFI^{W}_{OS} ← an empty set for holding the identified frequent one-sum weighted itemsets;
C_k ← **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 ← **count**($f_j \subseteq$ transactions in $D^{W}_{T\text{-}OS}$); // the **Score Transformation Procedure** is employed to verify the "⊆" relationship
      if ((support / $|D^{W}_{T\text{-}OS}|$) ≥ σ^{W}_{OS}) **then**
        add f_j into SFI^{W}_{OS}; // f_j is stored with its actual support value
        set frequentFlag to be true;
    end for
    if (¬frequentFlag) **then**
      remove e_i from C_k;
  end for
  k ← k + 1;
  C_k ← **generate** the set of candidate k-itemsets from frequent (k–1)-itemsets using "closure property";

Allocating Pattern Mining 6

Mining Allocating Patterns

- An allocating pattern (ALP) "$X^{OSW} \Rightarrow Y^{OSW}$" (where X^{OSW}, $Y^{OSW} \subset I^{OSW}$ and $X^{OSW} \cap Y^{OSW} = \emptyset$) is said to be *valid* when $X^{OSW} \cup Y^{OSW}$ is found in SFI^{W}_{OS}, and the one-sum weighted-confidence of this ALP exceeds a user-defined one-sum weighted-confidence threshold α^{W}_{OS}.

- The computation of one-sum weighted-confidence is:

  weighted-confidence$^{\text{one-sum}}(X^{OSW} \Rightarrow Y^{OSW})$ = count($(X^{OSW} \cup Y^{OSW}) \subseteq (T_j \in Ŧ)$) / count($X^{OSW} \subseteq (T_j \in Ŧ)$) ,

  where count( ) is the count function that returns the number of occurrences of an object. The **Score Transformation Procedure** is employed to verify the "$\subseteq$" relationship herein.

Allocating Pattern Mining 7

The algorithm to generate allocating patterns

Input: (a) A set of frequent one-sum weighted itemsets SFI^{W}_{OS};
(b) A one-sum weighted-confidence threshold α^{W}_{OS};
Output: A set of allocating patterns SALP;

SALP ← an empty set for holding the identified allocating patterns;
for each frequent one-sum weighted itemset $f_i \in SFI^{W}_{OS}$ **do**
for each frequent one-sum weighted itemset $f_j \in SFI^{W}_{OS}$ **do**
if ($f_j \subset f_i$) **then** // the **Score Transformation Procedure** is employed to verify the "⊂" relationship
confidence ← f_i.support / f_j.support;
if (confidence ≥ α^{W}_{OS}) **then**
allocating pattern p ← "{ f_j [with score in f_i] } ⇒ { (f_i – f_j) [with score in f_i] }";
add p into SALP;
end for
end for
return (SALP);

Evaluation (Experimental Setup)1

- A one-sum weighted "shopping-basket" (transaction) database is simulated in a two–stage process.

- In **Stage 1**, a traditional transaction database D_T is generated using the QUEST generator. 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. As a result of this stage, a transaction database T10.I5.N100.D10000 is produced.

Evaluation 2

- In Stage 2, the one-sum weighting score is 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: $\omega_i / (\sum_{\{k = 1 \ldots |T_j|\}} \omega_k)$. As a consequence, the simulated one-sum weighted "shopping-basket" database, namely T10.I5.N100.D10000.W3, is generated, where W denotes the size of the random integer set in item (one-sum) weighting.

- A set of ALPs was mined from T10.I5.N100. D10000.W3, using our proposed allocating pattern mining method (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.

Evaluation 3

- With regard to a one-sum weighted-support threshold value of 1% and a one-sum weighted-confidence threshold value of 20%, 78 ALPs are extracted. We order these ALPs based on their confidence value (in a descending manner), and present the top 10 and the bottom 10 ALPs.

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

(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.)

Conclusions

- In this study, we introduce the concept of ALlocating Patterns (ALPs). This is seen as an extension of the well-established Association Rule (AR) in a special (one-sum) weighted setting.
- 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.
- 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 pattern.

The End