WeightTransmitter: Weighted Association Rule Mining Using Landmark Weights

Yun Sing Koh[1], Russel Pears[2], and Gillian Dobbie[1]

[1] Department of Computer Science, University of Auckland, New Zealand
{ykoh,gill}@cs.auckland.ac.nz
[2] School of Computing and Mathematical Sciences, AUT University, New Zealand
rpears@aut.ac.nz

Abstract. Weighted Association Rule Mining (WARM) is a technique that is commonly used to overcome the well-known limitations of the classical Association Rule Mining approach. The assignment of high weights to important items enables rules that express relationships between high weight items to be ranked ahead of rules that only feature less important items. Most previous research to weight assignment has used subjective measures to assign weights and are reliant on domain specific information. Whilst there have been a few approaches that automatically deduce weights from patterns of interaction between items, none of them take advantage of the situation where weights of only a subset of items are known in advance. We propose a model, WeightTransmitter, that interpolates the unknown weights from a known subset of weights.

Keywords: Weight Estimation, Landmark Weights, Association Rule Mining.

1 Introduction

Weighted Association Rule Mining has been proposed as a method of generating a compact rule base whose rules contain items that are of most interest to the user [2,8,10,11]. Items are typically weighted based on background domain knowledge. For example, items in a market basket dataset may be weighted based on the profit they generate. However, in many applications pre-assignment of weights is not practical. In high dimensional datasets containing thousands of different items it may not be feasible to gather domain specific information on every single item, especially in a dynamically changing environment. In such situations it is more practical to exploit domain information to set weights for only a small subset of items (which we refer to as landmark items) and to then estimate the weights of the rest through the use of a suitable interpolation mechanism. This research addresses the issue of constructing a suitable model which will facilitate the estimation of unknown weights in terms of a given small subset of items with known weights.

Another key issue that needs to be addressed is the validity of assigning item weights based on domain specific input alone. Typically, items are supplied weights based on their perceived importance, for example the profit that they generate. However, such weight assessments are made in isolation to other items and thus do not account for the indirect profit that an item generates by promoting the sale of other items which may

P.-N. Tan et al. (Eds.): PAKDD 2012, Part II, LNAI 7302, pp. 37–48, 2012.

be of high profit. For example, retailers often reduce their profit margin on items that already have relatively low profit and market them as a package deal involving high profit items. A concrete example is a discount on a mobile handset that is conditional on the customer signing a long term contract with the phone company involved. In such situations, the "low profit" item (mobile handset) is used as an incentive to entice customers into buying the high profit items (calling plan contract). Clearly, in such contexts the actual profit margin of the low profit item does not accurately reflect its importance. Thus one of the premises of this research is that domain input on item weighting even when available may not be sufficient by itself in characterizing the importance of an item. Transactional linkages between items add value to domain specific information and when these two inputs are combined in a suitable manner a more accurate assessment can be made on an item's importance.

Two major contributions of this research are the development of a model that expresses the unknown weights of items in terms of known weights (landmark weights) and an interpolation mechanism that estimates weights by taking into account linkages between items that occur together. The rest of the paper is organized as follows. In the next section, we examine previous work in the area of weighted association rule mining. In Section 3 we give a formal definition of the weighted estimation problem. Section 4 presents our model for weight estimation. Our experimental results are presented in Section 5. Finally we summarize our research contributions in Section 6.

2 Related Work

In the context of weighted association rule mining a number of different schemes have been proposed for item weighting. Most of the schemes propose that domain information be utilized for setting weights for items. Tao et al. [9], Cai et al. [2] and Sanjay et al. [6] propose that item profit be utilized for assigning weights in retail environments for items while Yan et al. [11] use page dwelling time to assign weights in a web click stream data environment. More recent work reported in [8,3,4] took a different approach to the weight assignment problem. Sun and Bai introduced the concept of w-support which assigns weights to items based on the properties of transactions in a given dataset thus removing the need for domain specific input. The dataset was first converted into a bipartite graph, with one set of nodes representing the items and the other set of nodes representing the transactions. The w-support was then calculated by counting the links between items and the transactions that the items appeared in. Koh et al. [3] proposed a Valency model where the weight of an item was defined as a linear combination of its *purity* and its *connectivity*. Purity takes into account the number of items that a given item interacts with, while Connectivity accounted for the degree of interaction between an item and its neighboring items. Pears et al. [4] used a weight inference mechanism based on Eigenvectors derived from a transactional dataset. Given a dataset D, the covariance of every pair of items that occur in transactions across D was expressed in terms of its covariance matrix M. The first Eigenvector derived from the covariance matrix M was used to assign weights to items.

None of the work done so far in weight inference directly addresses the issue of weight estimation from a set of known landmark weights. A simple extension such as

restricting the set of items input to only include the unknown items will not suffice as the weights will be computed only on the basis of the interactions between the set of unknown items and the interactions with the landmark items will be neglected. As such, none of the above work can directly be utilized in their entirety.

3 Problem Definition

The weight fitting problem that we frame is to estimate the *overall* weight of items and not simply the domain specific weights for items that are unspecified (*i.e.*, item not in the landmark set, L). As stated in the introduction, certain items that are perceived to be of low importance on the basis of domain knowledge may actually assume a higher importance than their perceived rating due to strong interactions with items that are of high importance. In our problem setting we associate with each item a domain weight and an interaction weight. Domain weights dw are only available for the set of landmark items, L, whereas interaction weights iw are available for *all* items as these can be deduced from the co-occurrences of items given a transaction database.

Given a set of items I, a subset L of landmark items where $L \subset I$, and a transaction database D, the acquired weight w_i of a given item i is determined by:

$$w_i = \frac{\sum_{l \in L} iw(i,l).(w_l + dw_l) + \sum_{m \in M} iw(i,m).(w_m)}{\sum_{k \in N} iw(i,k)} \quad (1)$$

where N represents the set of neighbors of item i, $dw_i \geq 0$ when $i \in L$ and $dw_i = 0, otherwise$. Thus an item i acquires a weight from its interactions with its neighbors who transmit their own weights in a quantity proportional to the degree of interaction, iw. Neighbors that are landmarks transmit their domain weights as well as their acquired weights while neighbors in the set M of items that are not landmarks only transmit their acquired weights. In the context of this research a neighbor of a given item i is taken to be any item j that co-occurs with item i when taken across the database D. In effect, an item that is a landmark item contributes both its own domain weight and the weight acquired from its neighbors, while non landmark items simply transmit their acquired weight which in turn was obtained from their own interactions with neighboring items, which could include landmark items. Henceforth we shall abbreviate the term acquired weight simply by the term weight, except when it is necessary to emphasize the composite nature of the weight assignment.

The accuracy of the weight estimation mechanism expressed by Equation 1 above is dependent on how the interaction component is modeled. This specification is a modeling issue which does not impact on the general definition of the problem and so further discussion of this component is deferred to Section 4 which deals with the model developed to solve the problem.

For a given set of landmark items L the problem can now be stated formally as follows: return all items $i \in H$ where

$$H = \{i | i \in I \text{ is in the top } p\% \text{ of items when ranked on acquired weight from Eq (1)}\} \quad (2)$$

where p is a user-supplied threshold that determines the minimum overall weight to be returned to the user for use in a subsequent weighted association rule mining phase.

4 Weight Transmitter Model

In this section we present a model that we use as the basis of the solution to the weight estimation problem. We use a graph structure (N, E) where nodes are represented by items and edges by interactions between pairs of items. Each node i is associated with the weight w_i of the item, while an edge between items i and j is represented by $G(i, j)$ where G is the Gini information index [5]. The Gini information index $G(i, j)$ measures the degree of dependence of item j on item i. A high value of $G(i, j)$ indicates that item j occurs with a high degree of probability whenever item i occurs, likewise, the non occurrence of item i leads to the non occurrence of item j with a high degree of probability. Thus the G value captures the degree of dependence between pairs of items. The higher the dependence of item j on item i, the higher the proportion of weight transmission from item i to item j, and as such the Gini index can be used to express the interaction weight component of the model.

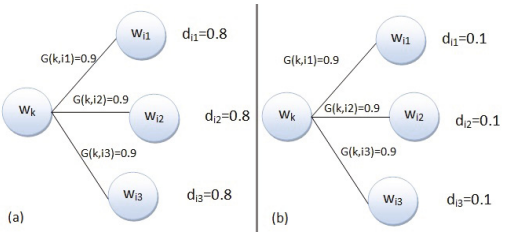

Fig. 1. Influence of Neighborhood in Weight Estimation

As an illustrative example, consider two different scenarios with four items whereby we have item k with unknown domain weight and three other items $i1$, $i2$, and $i3$ with known domain weights. In the first case, (Figure 1 (a)) each of $i1$, $i2$ and $i3$ have domain specified weights of 0.8 and each interacts with item k with a G value of 0.9. The WeightTransmitter model that we propose returns a weight value of 2.4 for each of the items, which when normalized yields a value of 0.89. With the same set of items but with domain specific weights set to 0.1 (Figure 1(b)) all weights for the four items end up with the same value of 0.3, which when normalized yields a value of 0.11. This example illustrates the importance of neighborhood in the weight estimation process; an item which is strongly connected through high G values to high weight items will acquire a high weight, whereas the same item when connected to low weight items will acquire a low weight, regardless of the strength of the connections.

We now present the WeightTransmitter model by expressing the weight of a given item k in terms of the weights of its neighbors as:

$$w_k = \frac{\sum_{i \in S_1} G(i,k).(w_i + dw_i) + \sum_{j \in S_2} G(j,k).(w_j)}{\sum_{i \in S_1} G(i,k) + \sum_{i \in S_2} G(i,k)} \tag{3}$$

where S_1 represents the set of neighbors of item i whose domain supplied weight dw_i components are known in advance and S_2 is the set of neighbors of item i whose domain

weights are unknown. Now $\sum_{i \in S_1} G(k,i) + \sum_{i \in S_2} G(k,i)$ represents a known quantity c_{1k}, since all G index values can be calculated from the transaction database. The dw_i terms in set S_1 also represent known quantities. We denote $\sum_{j \in S_1} G(k,i).dw_i$ by c_{2k}. Substituting the known constants c_{1k}, c_{2k} in the above equation and re-arranging the terms gives:

$$c_{1k}.w_k - \sum_{i \in S} G(i,k).w_i = c_{2k} \tag{4}$$

where $S = S_1 \cup S_2$ represents the complete neighborhood of item k. The above now represents a system of k linear simultaneous equations in k unknowns which has an exact solution with the Gaussian elimination method which we employ. The algorithm below illustrates how the WeightTransmitter model fits in with the traditional weighted association rule mining algorithm.

Algorithm: WeightTransmitter Model
Input: Transaction database T, known landmark weights dw, universe of items I
Output: Item Weights W

Step 1: Build a one level graph of the neighborhood of item i
$N(i) \leftarrow \{k | k \in t, t \in T, i \in t\}$
Step 2: Calculate G values for interactions between item i and neighbors $N(i)$
Step 3: Compute $C1 = \{|c_{1k}k \in I\}$ and $C2 = \{|c_{2k}k \in I\}$
Step 4: Solve for weight vector W
$W \leftarrow \{w(i) | \text{GaussianElimination}(I, C1, C2), i \in I\}$

5 Experimental Results

Our evaluation is divided into three sections: weight estimation evaluation, rule evaluation, and runtime evaluation. In the next section we describe the datasets that were used in these evaluations.

5.1 Datasets

Our experiments were conducted on five real-world datasets which are described below.

- **Retail dataset.** We used a retail market basket dataset supplied by a supermarket store that contained the unit profit values for each item which were supplied in a separate file [1].
- **Nasa weblog datasets.** We also used two different web log files from the NASA Kennedy Space Center in Florida collected over the months of July and August 1995. In these datasets pages represented items, and transactions consisted of a sequence of clicks on a set of web pages that took place across a session, which we set to have a maximum time of 15 minutes. The average dwelling time on a web page (taken across all transactions) was taken as a proxy for item weight.
- **Computer Science Lab datasets.** Finally, we used two datasets containing web log requests from a computer science lab at the University of Auckland between the months of December 2007 - February 2008, and February 2008 - December

2008. We preprocessed the dataset using the same technique as the Nasa datasets and used the same proxy for item weight. Overall there were 1764 items and 5415 instances for the first of these datasets, while the second had 2315 items and 5591 instances.

5.2 Weight Estimation Evaluation

This evaluation was designed with three key objectives in mind. Firstly, to establish the degree of sampling required in order to achieve convergence between estimated and actual weight on the composite weight measure. Ideally, convergence should be achieved at a low level of sampling for the weight estimation process to be effective. Secondly, to identify items that were flagged as being low weight according to domain information but were assigned high weight values by the WeightTransmitter model. These items are potentially interesting as they highlight situations where domain knowledge is inadequate to characterize the true importance of such items. Thirdly, we wanted to assess the level of accuracy achieved by the weight estimation process at the point of convergence. Since we had access to the domain weights for the complete set of items we were able to establish the ground truth in terms of the composite weights by simply running WeightTransmitter with a landmark sampling level at 100%.

To evaluate the accuracy of the weights produced by the WeightTransmitter model we varied the percentage of landmark weights in the range 10% to 90% and tracked the overall accuracy and precision across the high weight items. We start with the accuracy evaluation. At each of the sampling levels 30 different runs were used that chose different sets of landmark items at random. The accuracy measures presented represent an average of the measure taken across the 30 different trials.

Weight Convergence and Accuracy Analysis. Accuracy was tracked using three different measures: Correlation, Precision on high weight items, and Target Lift [7]. Target Lift is a measure commonly used to measure the lift in response rate that occurs when marketing offers are sent to a small set of customers who are likely to respond (identified through some prediction method) rather than a mass marketing campaign that targets the entire population. In the context of weight estimation the set of items returned by WeightTransmitter which it regards as high weight corresponds to the set of probable customers and the universe of items represents the entire customer population.

In the first analysis, we ran WeightTransmitter with the set of landmark weights as input and collected the results into the set S_l. We then re-ran WeightTransmitter with the complete set of known weights as input and collected the results into the set S_c. We then plotted the Pearson correlation between the two result sets against the sampling percentages that we used for the landmark weights. Figure 2 shows that there is a stabilization of correlation around the 30% mark; the average correlation value is 89%, with a standard deviation of 0.07. As expected, as the percentage of landmark items increases the greater is the degree of convergence between estimated weight and actual values on the composite weight value. Figure 2 shows that reasonable convergence of weights is achieved around the 30% mark.

For the second analysis each of the sets S_l and S_c were divided into two parts (bins): *low* and *high*. For each of the two sets, the top 10 percentile of items in terms of weight

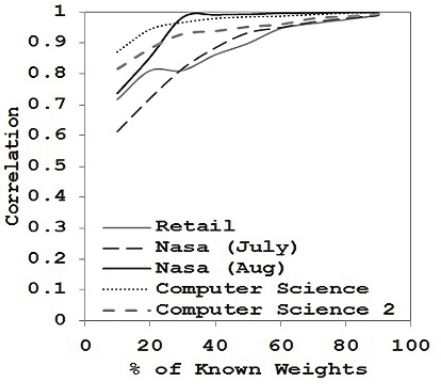

Fig. 2. Correlation Analysis

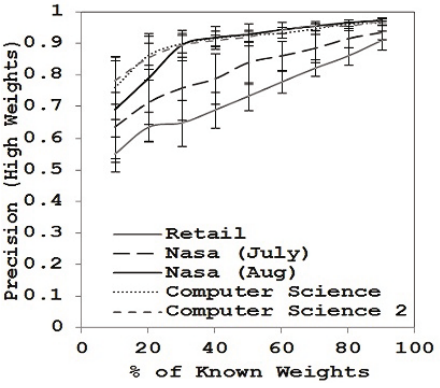

Fig. 3. Precision Analysis (High Weights)

were allocated to *high* bin, and all other items to the *low* bin. Using the bins based on the set S_c(*i.e.*, by running WeightTransmitter at 100% level of sampling) to establish a benchmark we were able to compute precision on the high weight category (bin). Figure 3 shows the precision in the high weight weight category as a function of the sampling percentage. At 30% the average precision value for the high weight items is 80%, with a standard deviation of 0.06.

In the third analysis we calculated the target lift. Table 1 shows that the lift in the true positive rate at a 30% sampling rate is much greater than 1 across all datasets, thus demonstrating the effectiveness of WeightTransmitter over a random weight assignment scheme in identifying high weight items.

Table 1. Target Lift Value at 30%

Dataset	Retail	Nasa (July)	Nasa (Aug)	Computer Science	Computer Science 2
Target Lift	5.04	6.99	7.75	8.53	3.41

Profit Analysis. Our weight accuracy analysis in the previous section establishes the effectiveness of our model in accurately estimating composite weight. However, we were also interested in tracking our other research premise which was the effect of the weighting scheme on items that interacted strongly with items that were known to have high weight. In particular, we were interested in tracking the set of items (H') where $H' = \{i | i \in I$ where i is in the top p percentile on the basis of composite weight but not on the basis of domain weight$\}$. We were able to compute the set H' as we had access to the weights of all items. For all items belonging to H' we defined a profit measure (P) that took into account the amount of indirect profit that such items generated. The profit measure for a given item $i \in H'$ was computed by taking the total profit (P_1) over all transactions (T_1) in which item i occurs and then subtracting from this value the total profit (P_2) over all transactions (T_2) in which item i does not occur.

In order to isolate the confounding effect of transactions in T_2 having more items than T_1 we restricted each of the transactions involved in T_2 to only have the neighbors of the item i under consideration. Furthermore, we also compensated for the differences in the sizes of T_1 and T_2 by scaling P_1 with the factor $\frac{|T_2|}{|T_1|}$.

$$P(i) = \frac{|T_2|}{|T_1|}. \sum_{k \in t_1} \sum_{t_1 \in T_1} w(k) - \sum_{k \in t_2} \sum_{t_2 \in T_2} w(k), \forall t_1, t_2 \in T \tag{5}$$

where $w(k)$ represents the weight of a high weight item k that is connected to item i and T is the set of all transactions in the transaction database. Equation 5 as defined above captures the indirect profit due to item i without the effects of the confounding factors just mentioned. However, the profit measure P by itself has little meaning unless it is compared with the profit generated by the set of items NH that remain low in weight without making the transition to the high weight category. For our premise that domain input on item weighting may not be sufficient by itself in characterizing the importance of an item the P values of items in the set H' needs to be substantially higher than the profit values in the set NH. Table 2 shows that this is indeed the case as the values in the H' column contains much higher values than the NH column for all of the datasets that we tracked. Table 2 contains the following columns; the percentage of items which have transited to the high weight category when transactional linkages are accounted for, average profit of items rated high by WeightTransmitter but not by domain weighting (*i.e.*, the set H'), average profit of items rated high by domain weighting (*i.e.*, the set H''), and average profit of items that were not rated high by WeightTransmitter (*i.e.*, the set NH).

Table 2. Weight Evaluation Based on Profit

Dataset	% Change	H' Items	H'' Items	NH Items
Retail	10	4647.53	3371.64	2418.20
Nasa (July)	11	5448.06	4375.46	3027.62
Nasa (Aug)	11	5101.86	4387.05	3424.50
Computer Science	11	99006.29	57231.93	49504.58
Computer Science 2	11	46219.19	32158.67	40224.14

Sensitivity Analysis. Given that the WeightTransmitter model achieved a high level of precision at the relatively small sampling level of 30% we were interested in investigating how robust the scheme was to changes in the data distribution. In particular, we were interested in tracking the sensitivity of Precision to the degree of variance in the data. Due to the fact that WeightTransmitter uses a sample defined over the set of landmark items, the question that arises is whether the error caused by the sampling remains stable or changes substantially when the underlying data distribution changes. To investigate this issue we used the Retail, Nasa (June), and Computer Science 1 datasets. Each weight value w in each of the selected datasets was perturbed by adding white Gaussian noise to it. Each weight value w for a given set was transformed into a weight value, $w_p = w + N(0, w/d)$, where d is a parameter that controls the level of variance injected into the dataset. We experimented with different values of d so as to obtain 3 levels of drift in variance from the baseline. The drift levels we used

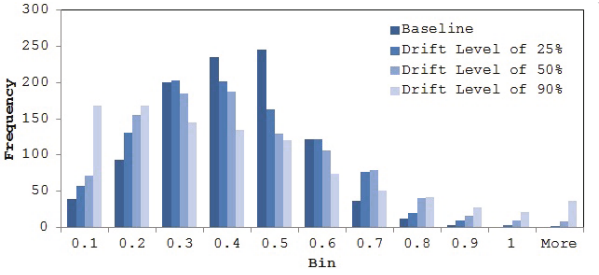

Fig. 4. Histogram of Support Distribution (Computer Science Dataset)

were 25%, 50% and 90% where drift level for a transformed dataset D' is defined by: $drift(D') = \frac{(stdev(D') - stdev(D))}{stdev(D)}$, where $stdev(D), stdev(D')$ represents the standard deviations across the baseline and perturbed datasets respectively. Figure 4 is the histogram of support distribution for the Computer Science dataset. The other datasets follow a similar distribution. The baseline represents the situation when the complete ground truth is known, *i.e.*, the domain weights for all items are known, thus enabling the composite weight to be calculated exactly with no error involved. As mentioned before we had access to the complete set of domain weights for each of the datasets that we experimented with, thus enabling us to measure the true deviation in precision with the degree of drift.

Table 3. Precision results deviation from the baseline

Dataset	Percentages of Known Items									Average
	10%	20%	30%	40%	50%	60%	70%	80%	90%	
Retail 25%	3.74	6.34	4.62	1.21	6.94	1.57	0.16	0.28	0.00	2.76
Retail 50%	3.38	5.43	3.03	2.82	7.02	1.15	2.02	3.31	0.23	3.16
Retail 90%	2.92	7.34	7.97	5.82	9.91	9.56	10.53	10.59	1.01	7.29
Nasa 25%	1.79	3.42	2.39	0.64	1.30	0.25	0.24	0.00	0.12	1.13
Nasa 50%	2.00	1.95	1.68	2.04	0.50	0.19	0.00	0.24	0.24	0.98
Nasa 90%	1.65	0.27	0.71	1.21	1.73	0.50	1.04	0.24	0.12	0.83
CS 25%	2.85	0.00	0.32	1.78	0.31	0.31	0.00	0.00	0.10	0.63
CS 50%	2.09	0.70	0.32	0.21	0.00	0.31	0.93	0.41	0.51	0.61
CS 90%	6.20	1.19	0.00	0.52	0.94	0.00	0.21	0.21	0.21	1.05

Table 3 shows that for the Retail dataset the deviation in Precision from the baseline ranged from 0 to 10.59%. In general as the level of sampling increased the error decreased. The deviation showed some sensitivity to the degree of variance in the data; as the drift level increased the deviation tended to increase. However, even at the extreme drift level of 90%, the deviation was no more than 10%. A similar pattern was observed for the Nasa and Computer Science datasets although the extent of the decrease in precision at the higher degrees of drift was on a smaller scale than with the Retail dataset. These results demonstrate that WeightTransmitter was not overly dependent on which items were chosen as landmarks, even with data that had a very high degree of variability. This is a very desirable feature of a weight estimation mechanism in general and

in terms of WeightTransmitter it inspires more confidence that the good performance at the 30% sampling level will generalize to a wide variety of different datasets.

5.3 Rule Evaluation

One of the major premises behind this research was that the true weight of an item is dependent not just on its individual importance, but also by its interaction with other items that it co-occurs with. For our premise to be true the rule base should contain rules of the form $X \rightarrow Y$ where X is a low weight item based on domain knowledge whereas Y is a highly rated item on the basis of domain knowledge. If such patterns occur then they signify that the set X of items appearing in rule antecedents should be weighted much more heavily than what is suggested on the basis of domain knowledge alone as such items co-occur strongly with highly weighted items.

The rule base was generated by inputing the top $p\%$ of items produced by WeightTransmitter to a standard rule generator. The rules generated for each dataset were subjected to a minimum support threshold of 0.03, confidence threshold of 0.75 and a lift threshold of 1.0. We computed rule interest measures such as Coherence and All Confidence and ranked the rule bases by the Coherence measure. We then analyzed the rule base to look for patterns of the form $X \rightarrow Y$ as described above that either support or refute this premise. The top p parameter was set at 20% for the Retail dataset and at 40% for the rest of the datasets. Figure 5 shows a small sample of 4 rules produced on the Retail dataset that exhibit this pattern.

541145000000 (Low) → 250000 (High)
210000 (Low) 541145000000 (Low) → 250000 (High)
5400136 (Low) → 541015000000 (High)
540014000000 (Low) → 250000 (High)

Fig. 5. Sample of rules

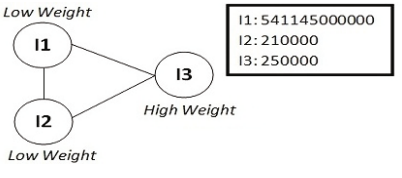

Fig. 6. Sample of WeightTransmitter Model

The presence of such rules validates one of the major premises behind this research. The rule bases produced from the other 3 datasets also exhibited such patterns but could not be presented due to the limitations of space. In terms of the Retail environment the practical value of such rules is that although items such as 541145000000 and 210000 are low profit items they are nevertheless important as the purchase of these items leads to the purchase of the high profit item, 250000. It is also important to note that the rules above would not have been generated if the items were weighted merely on the basis of their domain weights (i.e profit margins) alone as they would have not met the top $p\%$ threshold and would thus not have participated in the rule generation phase. As such, this represents one of the key contributions of this research.

Rules 1 and 2 in Figure 5 reveal the existence of a clique of 3 items: 541145000000, 210000, and 250000 that interact with each other strongly as shown in Figure 6. In the WeightTransmitter model item 250000 transmits its high domain weight to both

items 541145000000 and 210000 in proportions $G(3,1)$ and $G(3,2)$ respectively, thus increasing the domain weights of item 1 (541145000000) and item 2 (210000). This results in transforming these two items into high weight items. At the same time each of items 1 and 2 transmit their respective domain weights to item 3 in proportion to $G(1,3)$ and $G(2,3)$ thus increasing the weight of item 3. This transmission of weights, although increasing the weight of item 3 does not have a significant effect as item 3 is already of high weight.

5.4 Runtime Evaluation

As shown in the previous section the WeightTransmitter model leads to the discovery of valuable knowledge in the form of patterns that can be exploited in a useful manner. However, the model does introduce run time overheads in solving a system of linear equations. As such, our final experiment was to quantify what these overheads were and to ascertain whether the rule generation run time remained within reasonable bounds. Table 4 shows the runtime (measured in seconds) for our experiments with 30%, 60%, and 90% of items used as landmarks, along with the time taken to generate a rule base on the basis of domain knowledge alone, without the use of the WeightTransmitter model. In generating the latter rule base we used exactly the same constraints on minimum support, Confidence and Lift (with the same top p value) in order to keep the comparison fair. Table 4 reveals that the run time overhead introduced by WeightTransmitter does remain within reasonable bounds and that such overhead tends to decrease as a higher rate of landmark sampling is used. The decrease in run time at higher sampling levels is caused by the reduced number of operations required to transform the initial matrix into row echelon form due to the presence of more known values in the form of domain weights. The only result that goes against the above trend was with the Computer Science 2 dataset where the run time actually increased for the generation of the rule base built with the use of domain knowledge only. This was due to the larger number of items being returned in the top p list when compared to the list generated by WeightTransmitter. This resulted in a larger number of itemsets being generated which in turn resulted in a larger rule base, thus contributing to the increase in run time.

Table 4. Execution Time

Dataset	30% Known Weights	60% Known Weights	90% Known Weights	Original Weights
Retail	476	355	234	102
Nasa	56	45	40	14
Nasa Aug	58	48	45	14
Computer Science	48	43	50	45
Computer Science 2	111	108	107	211

6 Conclusions

This research has revealed that weight estimation based on a small set of landmark weights can be performed accurately through the use of the novel WeightTransmitter model that we introduced. Furthermore, we showed through a Profit Analysis conducted

on ground truth data that a substantial percentage of items switched status from the low or moderate weight categories to the high weight category, thus supporting our premise that weight assessments on an item should not be made in isolation to other items.

The use of other methods other than simple random sampling to identify landmark items will be explored in future work. As alternatives to simple random sampling, we plan to investigate the use of stratified random sampling as well as entropy based methods to identify influential items that will act as landmarks.

References

1. Brijs, T., Swinnen, G., Vanhoof, K., Wets, G.: Using association rules for product assortment decisions: A case study. In: Knowledge Discovery and Data Mining, pp. 254–260 (1999)
2. Cai, C.H., Fu, A.W.C., Cheng, C.H., Kwong, W.W.: Mining association rules with weighted items. In: IDEAS 1998: Proceedings of the 1998 International Symposium on Database Engineering & Applications, pp. 68–77. IEEE Computer Society, Washington, DC (1998)
3. Koh, Y.S., Pears, R., Yeap, W.: Valency Based Weighted Association Rule Mining. In: Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V. (eds.) PAKDD 2010. LNCS, vol. 6118, pp. 274–285. Springer, Heidelberg (2010)
4. Pears, R., Sing Koh, Y., Dobbie, G.: EWGen: Automatic Generation of Item Weights for Weighted Association Rule Mining. In: Cao, L., Feng, Y., Zhong, J. (eds.) ADMA 2010, Part I. LNCS, vol. 6440, pp. 36–47. Springer, Heidelberg (2010)
5. Raileanu, L.E., Stoffel, K.: Theoretical comparison between the gini index and information gain criteria. Annals of Mathematics and Artificial Intelligence 41(1), 77–93 (2004)
6. Ramkumar, G.D., Sanjay, R., Tsur, S.: Weighted association rules: Model and algorithm. In: Proc. Fourth ACM Int'l Conf. Knowledge Discovery and Data Mining (1998)
7. Roiger, R.J., Geatz, M.W.: Data Mining: A Tutorial Based Primer. Addison Edu. Inc. (2003)
8. Sun, K., Bai, F.: Mining weighted association rules without preassigned weights. IEEE Trans. on Knowl. and Data Eng. 20(4), 489–495 (2008)
9. Tao, F., Murtagh, F., Farid, M.: Weighted association rule mining using weighted support and significance framework. In: KDD 2003: Proceedings of the Ninth International Conference on Knowledge Discovery and Data Mining, pp. 661–666. ACM, New York (2003)
10. Wang, W., Yang, J., Yu, P.S.: Efficient mining of weighted association rules (WAR). In: KDD 2000: Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 270–274. ACM, New York (2000)
11. Yan, L., Li, C.: Incorporating Pageview Weight into an Association-Rule-Based Web Recommendation System. In: Sattar, A., Kang, B.-h. (eds.) AI 2006. LNCS (LNAI), vol. 4304, pp. 577–586. Springer, Heidelberg (2006)