

# SemGrAM - Integrating Semantic Graphs into Association Rule Mining

John F. Roddick<sup>1</sup> and Peter Fule<sup>1,2</sup>

<sup>1</sup> School of Informatics and Engineering,  
Flinders University,  
PO Box 2100, Adelaide,  
South Australia 5001

Email: roddick@infoeng.flinders.edu.au

<sup>2</sup> Defence Science and Technology Organisation  
PO Box 1500, Edinburgh, South Australia 5111  
Email: Peter.Fule@dsto.defence.gov.au

## Abstract

To date, most association rule mining algorithms have assumed that the domains of items are either discrete or, in a limited number of cases, hierarchical, categorical or linear. This constrains the search for interesting rules to those that satisfy the specified quality metrics as independent values or as higher level concepts of those values. However, in many cases the determination of a single hierarchy is not practicable and, for many datasets, an item's value may be taken from a domain that is more conveniently structured as a graph with weights indicating semantic (or conceptual) distance. Research in the development of algorithms that generate disjunctive association rules has allowed the production of rules such as *Radios*  $\vee$  *TVs*  $\rightarrow$  *Cables*. In many cases there is little semantic relationship between the disjunctive terms and arguably less readable rules such as *Radios*  $\vee$  *Tuesday*  $\rightarrow$  *Cables* can result. This paper describes two association rule mining algorithms, SemGrAM<sub>G</sub> and SemGrAM<sub>P</sub>, that accommodate conceptual distance information contained in a semantic graph. The SemGrAM algorithms permit the discovery of rules that include an association between sets of cognate groups of item values. The paper discusses the algorithms, the design decisions made during their development and some experimental results.

*Keywords:* Association Mining, SemGrAM, SemGrAM<sub>G</sub>, SemGrAM<sub>P</sub>, Disjunctive Rules, Semantic Graphs.

## 1 Introduction

Current association rule mining algorithms make a number of assumptions about the domains over which items are defined. In early work, the domains were assumed to be binary – the existence (or not) of an item in a transaction (Agrawal et al. 1993). This was extended to handle discrete domains (often by simply qualifying the item with the attribute name) and hierarchical domains (Han & Fu 1999, Lu 1997, Shen & Shen 1998, Suk & Park 1999). Categorical and linear domains have also been accommodated

(Lent et al. 1997, Gray & Orłowska 1998) as have fuzzy data (Kuok et al. 1998), spatial data (Koperski & Han 1995) and temporal data (Roddick & Spiliopoulou 2002). Ignoring such domain structure constrains the search and may result in missed rules – assuming discrete values, for example, means that item values must satisfy the quality metrics as independent values. To our knowledge, a single algorithm capable of handling more than one type of domain structure has not been developed.

The accommodation of hierarchies allows higher level concepts of those values through predefined or dynamically generated concept trees. For example, rules such as

*Sunday, Coffee*  $\rightarrow$  *Croissant*

may be found where

*Coffee*  $\supseteq$  {*Cappuccino, Latte, Macchiato*}

Unfortunately, this may convey the impression that *Latte* contributes to the rule when it may not. Moreover, in many cases the determination of a single hierarchy is not possible. Indeed, for many datasets, hierarchies may be imposed when it would be more appropriate to define the domain over a graph with weights indicating semantic (or conceptual) distance (see Figure 1). Apart from those domains that lend themselves to graph representation, directed graphs have the advantage that they subsume other domain structures.

Disjunctive association rule generation algorithms (Nanavati et al. 2001) aim to create rules that include disjunctive combinations of terms such as:

*Sunday*  $\vee$  *Tuesday, Macchiato*  $\rightarrow$  *Croissant*

Disjunctive rules are flexible in that no domain knowledge is required and perform well in many domains, particularly where Zipf's Law is evident such as in market basket data. However, often the rules produced contain disjunctions between unrelated terms, for example:

*Sunday*  $\vee$  *Water, Macchiato*  $\rightarrow$  *Croissant*

This mixing of concepts can reduce the readability of the results. Moreover, the disjunctive rule generation techniques outlined to date might combine two items, such as *Sunday* and *Tuesday* and omit *Monday* which might, for this dataset, just fall short of the metrics specified – that is, the semantic proximity of items is not taken into account when the disjunctive sets are formed.

One associated issue for data mining is the problem of scaling effects which occur particular for spatial and temporal data but can occur more pervasively. Essentially, many analyses are sensitive to the length, interval, area, volume or metric over which a variable is distributed. Often termed the *Modifiable Areal Unit Problem* (MAUP), or the *Ecological Fallacy*, it is an important characteristic in many problems (Openshaw 1983). Put simply, the granularity chosen for data collection determines which spatial or other phenomena can be identified. If spatial data are aggregated then the larger the unit of aggregation the more likely attributes will be correlated. Moreover, by aggregating into different groups you can get different results. Importantly for data mining, attributes which exist in the data at one level of support can vanish at coarser or finer scales, or other orientations. Thus the development of an algorithm capable of aggregating data *at the level of significance* is important.

In this paper we propose two algorithms<sup>1</sup>,  $\text{SemGrAM}_G$  and  $\text{SemGrAM}_P$ , that are able to use conceptual distance information, contained in one or more semantic graphs, within an association rule mining system to produce association rules with a new type of item grouping. The algorithms dynamically join elementary items into composite *itemgroups* within the itemsets. The itemgroup thus formed represents a disjunctive aggregation of a number of items that are similar, as determined by the semantic graph. The increased support of *itemgroups*, and that of the resulting itemset, can be calculated to find association rules from the itemset. In effect, *itemgroups* allow for specialised disjunctions of similar items in a single association – a particular form of disjunctive rule (cf. (Nanavati et al. 2001)). For example:

$[Tuesday, Wednesday]_{Cappuccino} \rightarrow Croissant$

In  $\text{SemGrAM}$ , more than one semantic graph can be used. In order to disambiguate the reason for an itemgroup’s construction, where there is need, the graph is noted. For example,

$[Adelaide, California]_{Weather} \rightarrow Wine$

or

$[Montana, Idaho]_{Proximity} \rightarrow BlackBear$

The advantages of such an approach include the following:

- A finer granularity of the items able to be included in the source data set without loss of succinctness in the resulting rule. Unlike pre-supplied hierarchies, the items included in an itemgroup are determined dynamically and thus can be constructed to include only items that contribute to the rule in a meaningful way,
- The ability to adopt different sets of conceptual distances for different tasks over the same dataset,
- The capture of itemsets that are otherwise below threshold but nevertheless contain useful information when items are joined,
- The ability to incorporate domains that are more complex than those already accommodated. That is, to capture semantics of trees, circular lists, and so on as a result of the subsuming semantics of graphs. Moreover, there is the ability to accommodate multiple domain structures such as lists of trees,

- Its use in text mining to consolidate terms with similar meaning but differing representation (qv. (Mooney et al. 2006)), and
- The clustering of rules with spatial attributes within them (accommodating some of the advantages of the work of Lent et al. (1997)).

Figure 1 shows some examples of semantic graphs that present domain knowledge. Using the *Stock Description* graph (Figure 1(a)), consider a set of results containing one itemset that includes *ottoman* and a second including *sofa* with the remainder of the itemset in common. If both itemsets fall below the support threshold we can use the information in the semantic graph to determine that an *ottoman* is similar to a *sofa* and create a new itemgroup –  $[sofa, ottoman]$  – as an abstraction of the two similar items. The new itemset incorporating the itemgroup will have a higher support, possibly meeting the support threshold. That itemset, and any resulting rules, effectively captures information about a concept made from the itemgroup formed by a joining the two items.

## 2 Related Work

This approach differs from previous research. For example, the work of Srikant & Agrawal (1997) uses an unweighted *is-a* (directed acyclic graph) hierarchy. In this work we allow the use of weighted graphs and remove the acyclic condition. The difference is significant and results not only in a different algorithm being needed but also in rules possessing a different semantic structure.

Multi-level association rule algorithm research (Han & Fu 1999, Shen & Shen 1998, Ong et al. 2001) differs from that outlined in this paper. Han and Fu, for example, develop a top-down approach using *a priori* supplied hierarchies. Mining with level wise abstraction uses a hierarchy containing the items as leaf nodes and adjusts the support threshold for the level of abstraction of the component items. The research presented here uses a semantic graph to combine items dynamically, so that they rise to meet an unchanging support threshold. Hierarchies are, of course, special cases of semantic graphs and some of the ideas presented by Han, Fu and others are applicable here also.

Informally, the process we adopt is the aggregation of items into *itemgroups* in cases where two or more *k*-itemsets do not possess the required support by themselves but where all the items in the itemsets are either identical or pairwise conceptually similar. This is discussed in detail in Section 3.3. While our approach requires the semantic graph to be supplied, the itemgroups discussed here are developed dynamically and, significantly, for  $\text{SemGrAM}_G$  may differ from rule to rule.

The idea of grouping association rules has been explored previously. For example, Lent *et al.* cluster association rules to find more general associations (Lent et al. 1997). Their system provides for association rules that contain quantitative attributes which are clustered, as a individual points, on a two dimensional plane. The plane is constructed from the quantitative linear attributes found in the rules with the clusters representing the groups of rules. The research presented in this paper clusters both categorical and non-categorical attributes within the itemsets and thus the work of Lent *et al.* is complementary to this work.

An algorithm for creating disjunctive association rules has been presented by Nanavati et al. (2001). Their work inspects itemsets creating generalised disjunctions without using semantic graphs. As such,

<sup>1</sup>Where there is no necessity to distinguish between  $\text{SemGrAM}_G$  and  $\text{SemGrAM}_P$  we simply refer to them as  $\text{SemGrAM}$ .

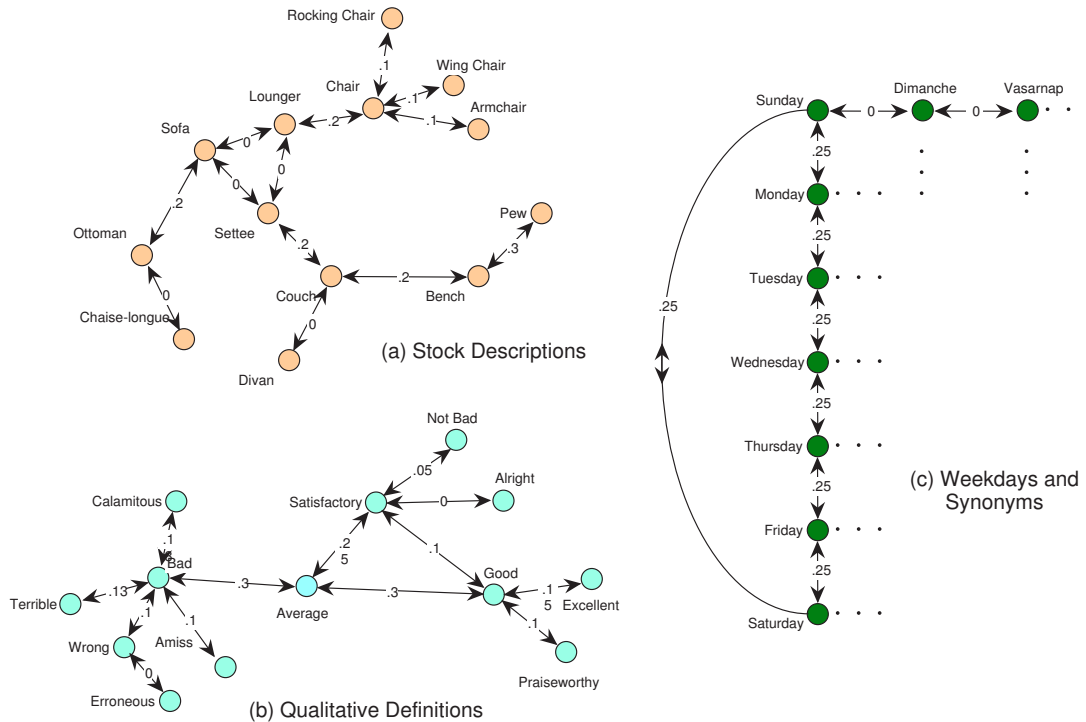


Figure 1: Situations in which graphs are used can be common.

their work is on the one hand more flexible (as no graph is needed) but may also be too general as it produces rules without reference to the conceptual distance between items, and thus may group dissimilar items.

In other work, Han et al. (1997) present a system that constructs a hypergraph based on the confidence of association rules, then clusters the items by partitioning the graph. Further research by Guha et al. (1999) indicates that the algorithm breaks down in certain cases. Guha *et al.* present the ROCK algorithm for clustering transactions. The links counting algorithm they present could have been used to generate the distance matrix presentation for the SemGrAM algorithm discussed later.

In terms of other *higher order* algorithms, Kusters et al. (1999) describe a system that clusters transactions based on the association rules generated from them. It takes the rules with the highest support, using the antecedents as the clusters for the transactions, to create a hierarchical clustering scheme for the data set. Similarly, Ertöz et al. (2002, 2003) present a nearest shared neighbour approach to clustering documents which is based on an algorithm by Jarvis & Patrick (1973).

Finally, Mazlack & Coppock (2002) present research into data preparation techniques involving the partitioning of the values of the input data set to help produce better results. The ideas presented focus on attributes with qualitative values and the best methods of partitioning those values globally. The work presented here partitions categorical attributes into new items that best suit the generation of new results for each subset of itemsets, although it should be noted that the granularity of partitioning of non-categorical attributes influences the results of the algorithm presented here.

### 3 Algorithm Design and Description

#### 3.1 Semantic Graphs

The advantage of graphs is that they subsume all other structures including lists and trees with

weighted uni-directional graphs being the most general. Importantly, although they are not used widely, semantic graphs are not uncommon – WordNet (Fellbaum 1998, Budanitsky & Hirst 2000), Roget’s Thesaurus (Jarmasz 2003), colour chart comparisons (CMYK / RGB / websafe / proprietary descriptions ...), geographic features, and so on provide a substantial resource and are readily available. Importantly, many such resources are not readily accommodated in a hierarchy and thus multi-level association rule mining solutions cannot be employed.

In addition, the MAUP (discussed earlier in Section 1) is accommodated by allowing graphs of different scales to be used with a fixed support threshold. That is, regardless of the granularity, rules with the predefined interest level will be reported.

For the purposes of this work graphs are assigned to a *family* with those in the same family able to be combined when creating an itemgroup. For example, consider the three semantic graphs – html colours, colour descriptions and geographic markers. A node in the html colour graph, say xFFA500, is comparable to the description orange and thus distances in the two graphs are aggregative. Such graphs are considered to be in the same *family*. Values in the third, geographic markers, are not comparable and would thus be placed in a different family.

In SemGrAM we allow graphs to be combined within each family as long as the edge weights can be normalised. In practice, we assume all graphs to be in different families unless two points of contact are specified between two graphs. Using these two points, the relativities between the weights used in each graph can be checked against the value of the *traversal threshold*  $\tau$  (discussed in the next section).

#### 3.2 Terminology

The input dataset consists of a finite number of transactions each containing a subset of the finite number of items  $S$ ,

$$S = \{i_1, i_2, \dots, i_n\}$$

Each item  $i_j$  is provided either as a simple value (eg. *Blue*) or as an attribute.value pair (eg. *Colour.Blue*). If the former is used, the prefix can be used to determine which families of graphs are appropriate for that item. In SemGrAM, we correlate graph families to attribute names through a simple list.

The input data is provided as transactions, with each transaction  $T_i$  containing a variable number of non-repeated items from  $S$ ,

$$T_x = \{i_{x_1}, i_{x_2}, \dots, i_{x_q}\} : \forall i_j \in S, q \geq 1$$

Each transaction generally contains significantly fewer items than are present in  $S$ , i.e.  $q \ll n$ .

An *itemgroup*  $G_i$  is a set of elementary items  $[i_1 \dots i_n]_\Gamma$  grouped for the purposes of itemset formation by virtue of their proximity in semantic graph  $\Gamma$ . The *itemgroup*  $G_i$  is then considered an atomic item within any *itemset*  $I_j$ <sup>2</sup>. Itemsets can consist of either or both of elementary items or itemgroups, (i.e.  $I_j = \{i_1 \dots i_m, G_1 \dots G_n\}$ ) however, once grouped, an itemgroup is treated as an atomic item for all subsequent purposes with respect to that itemset. Thus for the example above, the itemgroup [sofa, ottoman]<sub>Stock</sub> might be contained within the itemset {Green, [sofa, ottoman]<sub>Stock</sub>}. Note that the grouping of items  $[i_1 \dots i_n]$  as an itemgroup in  $I_j$  does not imply that they will be grouped in the same way in some different itemset  $I_k$ .

A semantic graph  $\Gamma_i$  is defined as a weighted, directional graph. Formally, each graph is a 4-tuple

$$\Gamma = \{S, E, \tau, \Phi\}$$

where a subset of the items in  $S$  are represented by nodes in the graph. The set of edges

$$E = \{e_1, e_2, \dots, e_k\}$$

with each edge

$$e = \{i_x, i_y, d\} : \forall i_j \in S, 0 \leq d \leq \tau$$

between the nodes represents a semantic distance between the items in the context of that graph. Each edge has a distance  $d$  representing the strength of the relationship between the two items it connects, higher values indicating a more distant or weaker relationship and zero indicating a synonym<sup>3</sup>. Any edge with a traversal distance greater than the maximum defined *traversal threshold*  $\tau$  is excluded from  $\Gamma_i$ . Each graph is assigned to a family  $\Phi_i$ .

Items omitted from the graph (or included without a connecting edge) are assumed to be dissimilar (i.e. to have infinite distance between them). The *traversal threshold*  $\tau$  is used to normalise distances across multiple graphs, making the scale used in the construction of the graph unimportant.

Semantic graphs are created either from expert knowledge of the context from which input dataset  $S$  is taken or are extracted from generally available knowledge. In SemGrAM, the graphs are stored as a dataset of triples  $\langle i_x, i_y, d \rangle$ , from which transitive distances are obtained recursively.

### 3.3 The SemGrAM Algorithms

While the ideas behind SemGrAM are common, this section describes two distinct algorithms, SemGrAM<sub>G</sub>

and SemGrAM<sub>P</sub> and discusses some of the design decisions. SemGrAM<sub>G</sub> is a flexible, but greedy algorithm while SemGrAM<sub>P</sub> is more efficient but imposes some constraints on the ruleset discovered. Specifically,

**SemGrAM<sub>G</sub>** operates in a greedy manner by aggregating appropriate itemsets that have a support that falls just under the minimum support threshold. As a result, SemGrAM<sub>G</sub> is able to use the semantic information to combine itemsets in different ways for different sets of rules. It is also independent of the underlying itemset generation algorithm.

**SemGrAM<sub>P</sub>** operates parsimoniously by amending FP-Trees and is thus more efficient but results in a ruleset where the same merger of items into an itemgroup may appear in multiple rules. SemGrAM<sub>P</sub> is at present based on the manipulation of FP-Trees and thus tied to FPGrowth (Han et al. 2000).

As for all association rule mining routines, the SemGrAM algorithms mine transactions to find common and significant co-occurrences of items<sup>4</sup>. Association rule mining routines typically utilise, *inter alia*, a support metric  $\sigma$ , which indicates the frequency of the co-occurrence of the items contained within each itemset,

$$I_x = \{i_{x_1}, i_{x_2}, \dots, i_{x_m} \mid \forall i \in S, m \geq 1\}, \sigma$$

where  $m$  indicates the cardinality of the itemset. The itemset can be viewed as an intersection of the items it contains where the support indicates the strength of the intersection.

SemGrAM uses three user defined support thresholds.

1. The traditional support threshold ( $\sigma$ ) that applies to all itemsets. If the support of any itemset is less than this threshold then the itemset is not used for rule production and thus not reported in the final set of results.
2. A *near support threshold* ( $\beta$ ) to partition itemsets of low cardinality, with itemsets that have support between  $\sigma$  and  $\beta$  termed *near support itemsets* or *nsi's*. The range of support values between the normal and near support values is termed the *near support range*<sup>5</sup>.
3. An *itemgroup cohesion threshold* ( $\gamma$ ). When an itemgroup is created the cohesion of the group is assessed, and if below  $\gamma$  is removed from consideration. Finding itemgroups is an optimisation problem that balances the potential gain in support through grouping the items with the loss of semantic precision (the *cohesion*) as items are added. For example if an item was defined over a graph of colour hues, red would be similar to crimson and vermilion, and may be grouped if the circumstances suggested it. If the itemset containing this itemgroup was still unable to reach the regular support, it may need to widen the semantics of the itemgroup by using other higher support items that were conceptually more distant. If pink, for example, had a high support it may be beneficial to include it in the itemgroup to help the itemset reach the normal support threshold, but it would start to stretch the semantic cohesion of the itemgroup;  $\gamma$  controls this semantic spread.

<sup>2</sup>For clarity we use square brackets for itemgroups and curly brackets for itemsets. Where obvious, the suffix indicating the graph is omitted.

<sup>3</sup>The semantic graph traversal concepts are explained in more detail elsewhere (Roddick et al. 2003).

<sup>4</sup>For a full survey of association mining algorithms see the recent survey by Ceglar & Roddick (2006).

<sup>5</sup>The concept of *nsi's* has already been investigated for other purposes in research into incremental association rule mining (Cheung et al. 1996, Rainsford et al. 1997, Kouris et al. 2003, Lee et al. 2005).

The thresholds have been adopted because the cognitive and computational complexity of merging low cardinality itemsets can be high.  $\beta$  and  $\gamma$  thus enable the user to manage the scope of the additional itemsets<sup>6</sup>.

### 3.3.1 SemGrAM<sub>G</sub>

The mining algorithm used as a base for SemGrAM<sub>G</sub> could have been chosen from any of the existing algorithms including, for example, Apriori (Agrawal et al. 1993), FPGrowth (Han et al. 2000) or Eclat (Zaki et al. 1997), as long as the algorithm is capable of supporting the multiple thresholds. In our implementation (see Section 4) we use the FPGrowth algorithm.

---

#### Algorithm 3.1 SemGrAM<sub>G</sub> $\beta$ -graph construction

---

```

1: Generate FP-tree
2: for each itemset  $I_j : \beta \leq support(I_j) < \sigma$  do
3:   add  $I_j$  node to  $\beta$ -graph;
4:   for each item  $I_k \in \beta$ -graph do
5:     if  $|I_j| = |I_k|$  and  $diff(I_j, I_k) = 1$  then
6:        $x, y =$  differing values
7:       for each graph family  $\Phi_i$  do
8:         if graph  $\Phi_i$  is applicable to both  $x$  and  $y$  then
9:            $weight = \infty$ 
10:          for each graph  $\Gamma_j \in \Phi_i$  do
11:             $weight = \min(weight, \frac{dist(x,y,\Gamma_j)}{\tau_{\Gamma_j}})$ 
12:          end for
13:          if  $weight \leq \gamma$  then
14:            create edge  $e_{j,k}$  between  $I_j$  between  $I_k$  in  $\beta$ -graph labelled with  $weight$ 
15:          end if
16:        end if
17:      end for
18:    end if
19:  end for
20: end for

```

---

Broadly, SemGrAM<sub>G</sub> re-examines the *nsi*'s in conjunction with information in the semantic graph with a view to forming new itemsets that will meet the normal support threshold. This is accomplished by constructing a  $\beta$ -graph for all *nsi*'s in which the itemsets are nodes and the edges indicate their similarity according to a family of graphs as shown in Figure 2 and as outlined in Algorithm 3.1. Note that there may be more than one edge between a pair of nodes if more than one family of graphs is applicable.

---

#### Algorithm 3.2 SemGrAM<sub>G</sub> $\beta$ -graph traversal

---

```

1: while edges left in  $\beta$ -graph do
2:   for each edge  $e_{j,k}$  between  $I_j$  between  $I_k$  in ascending order of weight do
3:     combine node  $I_{new}$  creating a new itemgroup containing  $x$  and  $y$ 
4:     for each all other edges to  $I_{other}$  from  $I_j$  or  $I_k$  do
5:        $weight =$  average of weights from  $I_{other}$  to  $I_j$  and  $I_{other}$  to  $I_k$ 
6:       if  $weight \leq \gamma$  then
7:         create new edge between  $I_{new}$  and  $I_{other}$  in  $\beta$ -graph labelled with  $weight$ 
8:       end if
9:       delete edge between  $I_{other}$  and  $I_j$  and  $I_{other}$  and  $I_k$ 
10:     end for
11:      $supp(I_{new}) = supp(I_j) + supp(I_k) - supp(I_j \cap I_k)$ 
12:     if  $support(I_{new}) \geq \sigma$  then
13:       Remove all edges connected to  $I_{new}$ 
14:     end if
15:   end for
16: end while

```

---

The function *diff* in Algorithm 3.1 operates in the same way as the *confusion matrix* of Oommen & Loke

<sup>6</sup>It is possible that the  $\beta$  and  $\gamma$  thresholds could be merged (for example, the support for an itemset might be deprecated as the cohesion decreases) but how this is achieved is large application domain specific and we have chosen to retain the two independent thresholds.

(1995), Oommen & Zhang (1996)<sup>7</sup> to examine two same-length itemsets returning the number of differences between them.  $dist(x, y, \Gamma_i)$  returns the semantic distance calculated (perhaps transitively) between the two nodes  $x$  and  $y$  in  $\Gamma_i$ . Following the construction of the  $\beta$ -graph, SemGrAM recursively searches the graph and combines the closest nodes.

In the current algorithm, once an itemset's support reaches  $\sigma$  it is removed from further merges (see Algorithm 3.2#13-15). This keeps the cohesion of the itemgroups as tight as possible. If these lines are omitted, the algorithm will produce more rules may have higher support at the expense of items with less semantic precision. Finally, rule production is relatively easy as given in Algorithm 3.3.

---

#### Algorithm 3.3 Rule Production

---

```

1: extract rules from FP-tree as per (Han et al. 2000)
2: for each nodes  $I_j$  in  $\beta$ -graph do
3:   if  $support(I_j) \geq \sigma$  then
4:     extract rule
5:   end if
6: end for

```

---

### 3.4 SemGrAM<sub>P</sub>

This version of SemGrAM was written to investigate the utility of providing a more efficient algorithm at the expense of some semantic flexibility. SemGrAM<sub>P</sub> inspects the FP-Tree and merges branches that are between  $\sigma$  and  $\beta$  (ie those that would result in *nsi*'s). The effect of merging them in the FP-tree means that the SemGrAM<sub>P</sub> runs more efficiently (in terms of both time and space) than SemGrAM<sub>G</sub> as can be seen from the experimental results. The algorithm is in two parts - first the FP-tree is modified as outlined in Algorithm 3.3. Second, rules are produced as per (Han et al. 2000). The first part looks for sections of the FP-tree such that can be merged.

---

#### Algorithm 3.4 SemGrAM<sub>P</sub> FP-tree traversal

---

```

1: recursively search the FP-Tree
2: if
  (a) The weight of a node (in the context of what comes above it) is greater than  $\sigma$ 
  (b) There are at least two children ( $n_1 \dots n_i$ ) of that node such that
    i. they are within the threshold semantic distance  $\gamma$  (ie as per Algorithm 3.1#7-17)
    AND
    ii. they have weights between  $\sigma$  and  $\beta$  then
3:   Create a new item representing the itemgroup  $[n_1 \dots n_i]$ 
4:   Merge subtrees of those children using the new item as root
5: end if

```

---

SemGrAM<sub>P</sub> is considerably simpler, both to code and execute but it should be noted that the same merging of subtrees may result in the same itemgroup being used in a number of rules.

## 4 Evaluation of Proof-of-Concept System

To demonstrate the concept, we implemented SemGrAM<sub>G</sub> and SemGrAM<sub>P</sub> (and for comparison, FPGrowth) in Java and ran experiments on a 1.5GHz Mac PowerPC G4. This implementation has shown it to be tractable and to reveal interesting rules that would otherwise not be reported. Note that the ability to dynamically create itemgroups means that items can be specified at a lower granularity (although not without affecting performance).

<sup>7</sup>Oommen uses a confusion matrix to determine the probability of striking a wrong key on a keyboard, which can then be incorporated into an edit distance function.

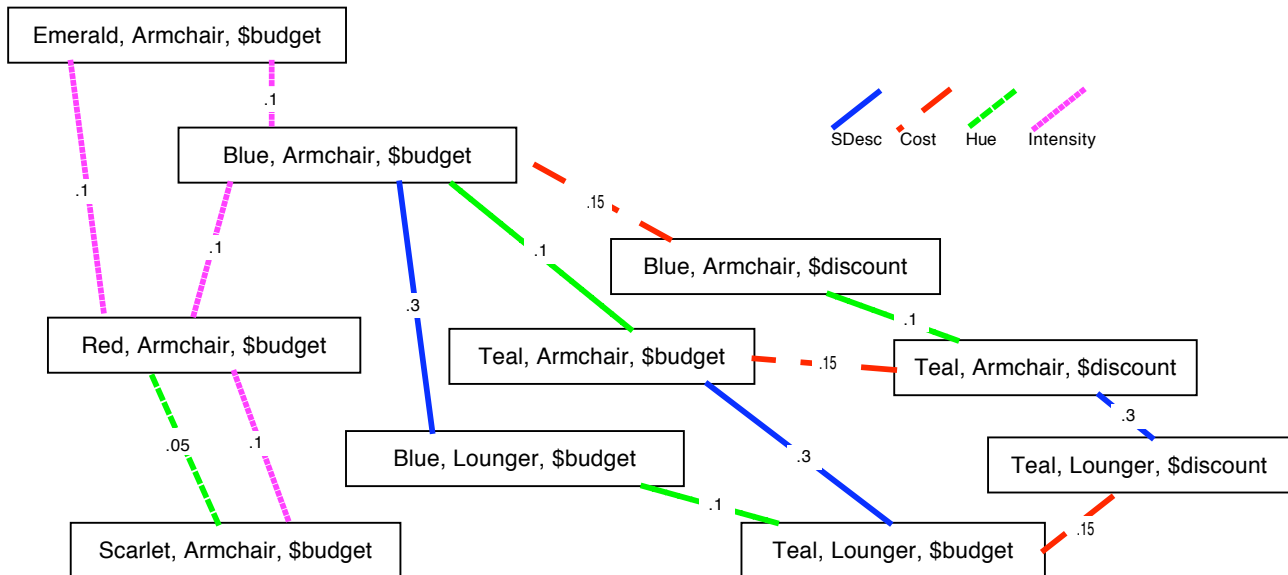


Figure 2: Example  $\beta$ -graph. Note that the different graphs for Hue and Intensity means that  $\langle \text{Red, Armchair, \$budget} \rangle$  cannot be put in the same itemgroup as  $\langle \text{Teal, Armchair, \$budget} \rangle$ .

A 487Kb, 10,000 transaction synthetic dataset was constructed together with semantic graphs that covered 25%, 50%, 75% or 100% of the items listed. Results are shown in Figure 3. Two important points to note are that the premium for handling semantic graphs is currently up to 52% (although for a lower coverage and a larger dataset the premium is more reasonable at below 10%). This makes the concept tractable although further efficiencies may be useful. Secondly, the SemGrAM algorithms are linear in the time taken to process each itemset regardless of input file size. On all datasets tested, both SemGrAM<sub>G</sub> and SemGrAM<sub>P</sub> have shown that they scale satisfactorily. Moreover, as the dataset size increases, the cost per itemset reduces. Finally, the effects of changes to the *itemgroup cohesion threshold* ( $\gamma$ ) can be large. As  $\gamma$  increases the interconnection between items accelerates showing the *double jump* phenomenon reported elsewhere (Spencer 2001).

Currently we limit SemGrAM<sub>G</sub> to looking for itemsets that vary by a single item. It can easily be seen that there would be cases where itemsets are similar but vary by more than a single item. In principle, the algorithm could be modified to assess the distances between a number of items in a set of itemsets and find the combination of items that creates the minimum distance. For example, the itemsets  $\langle \text{Blue, Armchair, \$Budget} \rangle$  and  $\langle \text{Teal, Lounger, \$Budget} \rangle$  in Figure 2 might be considered mergable. The issues arise in recording which items made the itemsets similar and making judgments as to whether the itemgroups created make sense semantically. There is, for example, a chance that inappropriate inferences may result.

## 5 Conclusions and Further Research

This paper has outlined two new algorithms for accommodating semantic graphs within association rule mining. In so doing it not only accommodates graph structured domains but also those for which a weighted, directed graph can be used to simulate other domain structures (such as lists and hierarchies). In some respects this work can thus be considered to subsume some earlier work in these areas.

The focus of the proof-of-concept implementation was not on performance but on proving that the

design decision were sound. Nevertheless, the implementation shows that even in this implementation, the premium paid for the extra processing is not excessively high, even in the case of SemGrAM<sub>G</sub>. In practice, one of the major advantages will be that items can be specified at a lower granularity with the algorithm selecting the most appropriate aggregations.

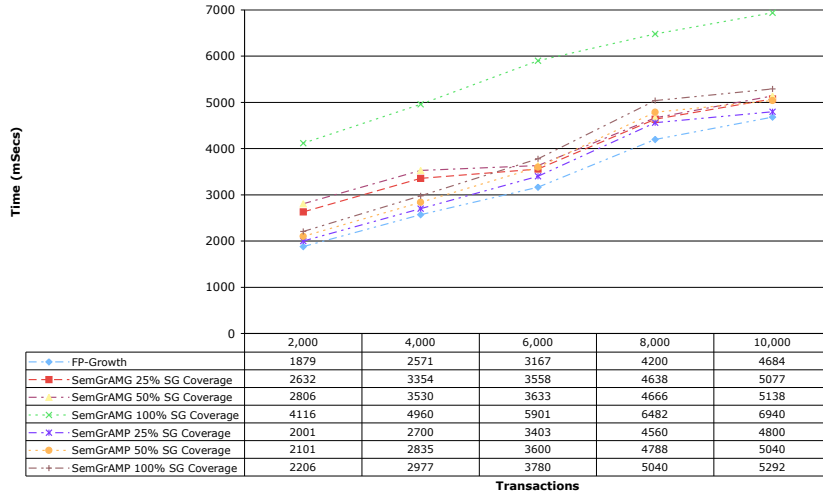
Further work can be envisaged for the algorithm, some of which is discussed in Section 4. In particular, the work of Nanavati et al. (2001) is complimentary to our work and it is possible that both ideas could be combined in a single algorithm.

The algorithms are currently dependent on the prior definition of the semantic graph. We have not yet investigated automatic generation of the graph or the use of functional (as opposed to enumerated) descriptions of graphs. However, there is pre-existing work in this area which could be utilised.

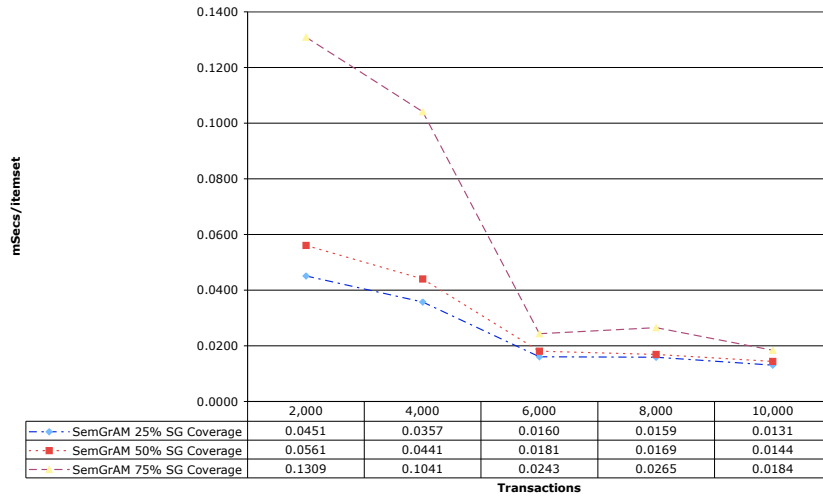
Since the overhead involved in the user's comprehension of the rules produced can significantly outweigh variations in processing time, it would potentially be useful to provide a good user-interface to the system allowing the full exploration of results through the semantic graph structure. For instance, if a result includes an itemgroup it could allow a one click look-up of the items that are a part of that itemgroup and show their contributions to the support of that itemset. It should be able to display the semantic graph clustering with varying levels of cohesion, possibly with varying clustering algorithms.

The system may also be applicable to the problem of clustering of association rules (cf. Lent et al. (1997)). Specifically, if one of the items is a spatial attribute, such as zip code, then we could potentially generate clusters of rules.

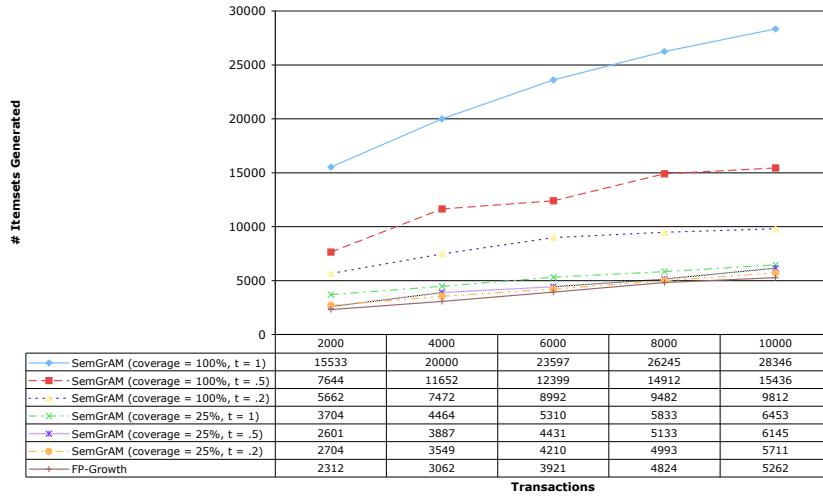
On a broader level, semantic graphs have not been accommodated in many areas of data mining to date and a wider program of research could be considered for which this research would be an example. In particular, sequential pattern mining (Agrawal & Srikant 1995, Srikant & Agrawal 1996) offers an opportunity for enhancement.



(a) Effect of varying coverage of semantic graph



(b) Scalability of SemGrAM



(c) Effects of varying  $\tau$

Transaction file size	Distinct Itemsets	Av. time (mSecs)	$\mu$ s/trans	$\mu$ s/itemset
2,000	15,533	4,116	2.058	0.265
4,000	20,000	4,960	1.240	0.248
6,000	23,597	5,901	0.983	0.250
8,000	26,245	6,482	0.810	0.247
10,000	28,346	7,170	0.717	0.253

(d) Experimental Timing (SemGrAM<sub>G</sub>)

Figure 3: Experimental Results

## References

- Agrawal, R., Imielinski, T. & Swami, A. (1993), Mining association rules between sets of items in large databases, in P. Buneman & S. Jaajodia, eds, 'ACM SIGMOD International Conference on the Management of Data', ACM Press, Washington DC, USA, pp. 207–216.
- Agrawal, R. & Srikant, R. (1995), Mining sequential patterns, in P. Yu & A. Chen, eds, '11th International Conference on Data Engineering (ICDE'95)', IEEE Computer Society Press, Taipei, Taiwan, pp. 3–14.
- Budanitsky, A. & Hirst, G. (2000), Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures, in 'Workshop on WordNet and Other Lexical Resources, in the North American Chapter of the Association for Computational Linguistics (NAACL-2000)', Pittsburgh, PA, USA.
- Ceglar, A. & Roddick, J. F. (2006), 'Association mining', *ACM Computing Surveys* **38**(2).
- Cheung, D., Han, J., Ng, V. & Wong, C. (1996), Maintenance of discovered association rules in large databases: an incremental updating technique, in S. Su, ed., '12th International Conference on Data Engineering (ICDE'96)', IEEE Computer Society, New Orleans, Louisiana, USA, pp. 106–114.
- Ertöz, L., Steinbach, M. & Kumar, V. (2002), A new shared nearest neighbor clustering algorithm and its applications, in R. Grossman, J. Han, V. Kumar, H. Mannila & R. Motwani, eds, '2nd SIAM International Conference on Data Mining (SDM'02)', SIAM, Arlington, VA, USA.
- Ertöz, L., Steinbach, M. & Kumar, V. (2003), Finding topics in collections of documents: A shared nearest neighbor approach, in W. Wu, H. Xiong & S. Shekhar, eds, 'Clustering and Information Retrieval 2003', Kluwer, pp. 83–104.
- Fellbaum, C., ed. (1998), *WordNet: An Electronic Lexical Database*, Bradford Books.
- Gray, B. & Orłowska, M. (1998), CCAIIA: Clustering categorical attributes into interesting association rules, in X. Wu, K. Ramamohanarao & K. Korb, eds, '2nd Pacific-Asia Conference on Research and Development in Knowledge Discovery and Data Mining (PAKDD-98)', Vol. 1394 of *LNAI*, Springer, Melbourne, Australia, pp. 132–143.
- Guha, S., Rastogi, R. & Shim, K. (1999), Rock: A robust clustering algorithm for categorical attributes, in '15th International Conference on Data Engineering', IEEE Computer Society Press, Sydney, Australia, pp. 512–521.
- Han, E.-H., Karypis, G., Kumar, V. & Mobashar, B. (1997), Clustering based on association rule hypergraphs, in 'Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'97)', Tucson, Arizona.
- Han, J. & Fu, Y. (1999), 'Mining multiple-level association rules from large databases', *IEEE Transactions on Knowledge and Data Engineering* **11**(5), 798–804.
- Han, J., Pei, J. & Yin, Y. (2000), Mining frequent patterns without candidate generation, in W. Chen, J. Naughton & P. Bernstein, eds, 'ACM SIGMOD International Conference on the Management of Data (SIGMOD 2000)', ACM Press, Dallas, TX, USA, pp. 1–12.
- Jarmasz, M. (2003), Roget's Thesaurus as a Lexical Resource for Natural Language Processing, Masters, University of Ottawa.
- Jarvis, R. A. & Patrick, E. A. (1973), 'Clustering using a similarity measure based on shared nearest neighbors', *IEEE Transactions on Computers* **C-22**(11).
- Koperski, K. & Han, J. (1995), Discovery of spatial association rules in geographic information databases, in '4th International Symposium on Large Spatial Databases', Maine, pp. 47–66.
- Kosters, W. A., Marchiori, E. & Oerlemans, Ard, A. J. (1999), Mining clusters with association rules, in D. Hand, J. Kok & M. Berthold, eds, '3rd International Symposium on Advances in Intelligent Data Analysis, IDA-99', Vol. 1642 of *LNCS*, Springer, Amsterdam, p. 39.
- Kouris, I. N., Makris, C. & Tsakalidis, A. K. (2003), An improved algorithm for minign association rules using multiple support values, in I. Russell & S. Haller, eds, '16th Florida International Artificial Intelligence Research Society Conference', AAAI Press, St. Augustine, Florida, USA, pp. 309–313.
- Kuok, C., Fu, A. & Wong, M. H. (1998), 'Mining fuzzy association rules in databases', *ACM SIGMOD Record* **27**(1), 41–46.
- Lee, Y.-C., Hong, T.-P. & Lin, W.-Y. (2005), 'Mining association rules with multiple minimum supoports using maximum constraints', *International Journal of Approximate Reasoning* **40**(1-2), 44–54.
- Lent, B., Swami, A. & Widom, J. (1997), Clustering association rules, in A. Gray & P.-A. Larson, eds, '13th International Conference on Data Engineering', IEEE Computer Society Press, Birmingham, UK, pp. 220–231.
- Lu, Y. (1997), Concept Hierarchy in Data Mining: Specification, Generation and Implementation, Master of science, Simon Fraser University.
- Mazlack, L. & Coppock, S. (2002), Granulating data on non-scalar attribute values, in 'IEEE International Conference on Fuzzy Systems', Honolulu, pp. 944–949.
- Mooney, C. H., De Vries, D. & Roddick, J. F. (2006), A multi-level framework for the analysis of sequential data, in S. Simoff & G. Williams, eds, 'Data Mining: Theory, Methodology, Techniques, and Applications', Vol. 3755 of *LNAI*, Springer, Heidelberg, Germany, pp. 229–243.
- Nanavati, A., Chitrapura, K. P., Joshi, S. & Krishnapuram, R. (2001), Mining generalised disjunctive association rules, in '10th International Conference on Information and Knowledge Management (CIKM'01)', ACM Press, Atlanta, Georgia, USA, pp. 482–489.
- Ong, K. L., Ng, W. K. & Lim, E. P. (2001), Large mining multi-level rules with recurrent items using fp-tree, in '3rd IEEE Conference on Information, Communications and Signal Processing (ICICS'2001)', Springer, Singapore.
- Oommen, B. J. & Loke, R. K. S. (1995), Pattern recognition of strings with substitutions, insertions, deletions and generalized transpositions, in 'IEEE International Conference on Systems, Man and Cybernetics', Vol. 2, IEEE, pp. 1154–1159.

- Oommen, B. J. & Zhang, K. (1996), 'The normalized string editing problem revisited', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**(6), 669–672.
- Openshaw, S. (1983), *The Modifiable Areal Unit Problem (Concepts and Techniques in Modern Geography)*, Geo Books, Norwich, UK.
- Rainsford, C., Mohania, M. & Roddick, J. F. (1997), A temporal windowing approach to the incremental maintenance of association rules, in J. Fong, ed., '8th International Database Workshop, Data Mining, Data Warehousing and Client/Server Databases (IDW'97)', Springer, Hong Kong, pp. 78–94.
- Roddick, J. F., Hornsby, K. & De Vries, D. (2003), A unifying semantic distance model for determining the similarity of attribute values, in M. Oudshoorn, ed., '26th Australasian Computer Science Conference (ACSC2003)', Vol. 16 of *CRPIT*, ACS, Adelaide, Australia, pp. 111–118.
- Roddick, J. F. & Spiliopoulou, M. (2002), 'A survey of temporal knowledge discovery paradigms and methods', *IEEE Transactions on Knowledge and Data Engineering* **14**(4), 750–767.
- Shen, L. & Shen, H. (1998), Mining flexible multiple-level association rules in all concept hierarchies, in G. Quirchmayr, E. Schweighofer & T. Bench-Capon, eds, '9th International Conference on Database and Expert Systems Applications, DEXA'98', Vol. 1460 of *LNCS*, Springer, Vienna, Austria, pp. 786–795.
- Spencer, J. (2001), *The Strange Logic of Random Graphs*, Springer.
- Srikant, R. & Agrawal, R. (1996), Mining sequential patterns: generalisations and performance improvements, in P. M. G. Apers, M. Bouzeghoub & G. Gardarin, eds, 'International Conference on Extending Database Technology, EDBT'96', Vol. 1057 of *LNCS*, Springer, Avignon, France, pp. 3–17.
- Srikant, R. & Agrawal, R. (1997), 'Mining generalized association rules', *Future Generation Computer Systems* **13**(2-3), 161–180.
- Suk, C.-Y. & Park, E. (1999), An approach to intensional query answering at multiple abstraction levels using data mining approaches, in '32nd Annual Hawaii International Conference on Systems Sciences, HICSS-32', IEEE Comput. Soc, Los Alamitos, CA, USA.
- Zaki, M. J., Parthasarathy, S., Ogihara, M. & Li, W. (1997), New algorithms for fast discovery of association rules, in '3rd International Conference on Knowledge Discovery and Data Mining (KDD-97)', AAAI Press, Newport Beach, CA, USA, pp. 283–286.