The Fuzzy Frequent Pattern Tree

STERGIOS PAPADIMITRIOU[1] SEFERINA MAVROUDI[2]

1. Department of Information Management, Technological Educational Institute of Kavala,
65404 Kavala, Greece

2. Pattern Recognition Laboratory, Department of Computer Engineering and Informatics, School of Engineering, University of Patras,
Rion, Patras, 26500, Greece

Abstract: A significant data mining issue is the effective discovery of association rules. The extraction of association rules faces the problem of the combinatorial explosion of the search space, and the loss of information by the discretization of values.
The first problem is confronted effectively by the Frequent Pattern Tree approach of [10]. This approach avoids the candidate generation phase of *Apriori* like algorithms. But, the discretization of the values of the attributes (i.e. the "items") at the basic Frequent Pattern Tree approach implies a loss of information. This loss usually either deteriorates significantly the results, or constitues them completely intolerable.
This work extends appropriately the Frequent Pattern Tree approach in the fuzzy domain. The presented Fuzzy Frequent Pattern Tree retains the efficiency of the crisp Frequent Pattern Tree, while at the same time the careful updating of the fuzzy sets at all the phases of the algorithm tries to preserve most of the original information at the data set.
The paper presents an application of the Fuzzy Frequent Pattern Tree approach to the difficult problem of the discovery of fuzzy association rules between genes from massive gene expression measurements.

keywords: Association Rules, Fuzzy Association Rules, Depth-First Search, Frequent Pattern Tree, Data Mining, Gene Expression Analysis

1. Introduction

A fundamental concern in the data mining research is the effective discovery of association rules. However, the association rule extraction algorithms face two basic problems: the combinatorial explosion of the search space [6, 9] and the loss of information by the discretization of values [3,4].

The Frequent Pattern Tree approach of [10] confronts effectively the first problem by avoiding the candidate generation phase of *Apriori* like algorithms, that is amenable to combinatorial explosion of the required computational resources.

However, the discretization of the values of the attributes (i.e. the "items") at the basic Frequent Pattern Tree approach implies a loss of information. For many applications, this loss either deteriorates significantly the results, or is completely intolerable.

The present paper extends appropriately the Frequent Pattern Tree approach in the fuzzy domain. The derived Fuzzy Frequent Pattern Tree algorithm retains the efficiency of the crisp Frequent Pattern Tree, while at the same time the careful updating of the fuzzy sets at all the phases of the algorithm tries to preserve most of the original information at the data set.

We apply the Fuzzy Frequent Pattern Tree approach to the difficult problem of the discovery of fuzzy association rules between genes from massive gene expression measurements, that usually were analyzed previously mainly with clustering techniques [1,8] and with traditional Apriori like approaches [5].

The paper proceeds as follows: Section 2 presents the concept of fuzzifying the attribute values. Section 3 outlines the form of the fuzzy association rules. Section 4 describes the Fuzzy Frequent Pattern Tree (FFPT) data structure and the construction algorithms. Section 5 deals with the utilization of the FFPT data structure for frequent pattern mining. Section 6 utilizes the extracted frequent patterns for fuzzy rule extraction and finally, the paper concludes with some directions for future work.

2. Fuzzification of attribute values

A prerequisite for the application of any fuzzy association rule algorithm is the proper fuzzification of the values of each database's attribute. We describe this step for the particular example of mining gene expression data, where:

a. The *transactions* correspond to the *experimental conditions* at which the expression level of each gene is measured.

b. The *attributes* are the *genes* and their values assert the relative expression level of the corresponding gene at the analogous condition.

For this application for each gene *g* and each condition *k* we need to define at least two fuzzy sets, i.e. m_{kg}^{L}, m_{kg}^{H} , where the first one quantifies the "Low" expression content of the gene (i.e., under expression) and the "High" the degree of over expression. Clearly, these concepts can be applied to any other database.

The evaluation of fuzzy association rules at the context of the concerning application perplexes the following steps (similar steps are required for any other relevant application):

1. The transformation of the quantitative value of the expression level of each gene $g=1,\dots,N$ for every condition $k=1,\dots,C$ to the membership values at the fuzzy sets m_{kg}^{L}, m_{kg}^{H} for the Low expression and for the High one.
2. We calculate the scalar cardinality for every gene *g* of each of the regions *L* and *H* over all the conditions in the gene expression data as $\text{count}_g L=\sum_{k=1}^{C} m_{kg}^{L}$ and $\text{count}_g H=\sum_{k=1}^{C} m_{kg}^{H}$ respectively.
   These counts for every gene quantify expresses how frequent are the occurences of values *L* and *H* across all the conditions, i.e. the underexpressed and overexpressed content of the gene. Clearly, a gene that appears overexpressed in some conditions and underexpressed in others can display large values for both counts.
3. If these linguistic partitions for each gene *j*, e.g. $g_j H$, exceeds the required support threshold, the corresponding "items" are included at the frequent 1-itemsets.

We should note that these steps are also required and for the traditional Fuzzy Apriori algorithm [4]. However, the next steps of our approach (presented below) are based on the Fuzzy Frequent Pattern Tree (FFPT) data structure, that avoids the computationally demanding step of the candidate generation.

3. Fuzzy Association Rules

For a data set $D=\{t_{1,} t_{2,} \dots, t_C\}$, consisting of *C* transactions, with *N* attributes $A=\{a_1, a_2, \dots, a_N\}$ and fuzzy sets associated with each attribute, the purpose of fuzzy association is to detect interesting and potentially useful regularities. These regularities are expressed in terms of fuzzy association rules of the form:

if $P=\{a_1, a_2, \dots, a_N\}$ is $V=\{f_1, f_2, \dots, f_N\}$

then $P'=\{a'_{1,} a'_{2,} \dots, a'_N\}$ is $V'=\{f'_{1,} f'_{2,} \dots, f'_N\}$

where f_i, f'_i are fuzzy sets related to attributes a_i, a'_i respectively and P, P' are disjoint itemsets in the sense that they do not share common attributes.

The purpose is to detect the interesting rules, i.e. those that have enough support and high confidence value [3,4].

An association rule expresses general patterns of dependencies between attributes. It has the general form where Premise and Consequent are *itemsets*. The term itemset has its roots to commercial data mining applications where the concern is to detect sets of items that are correlated in transactions.

In the gene expression mining application of the example, we view the genes as the *features* and the experimental conditions (corresponding to the notion of *transactions*) of the microarray experiments as the *patterns*. Thus a "transaction" refers to the acquired expression values of the genes in a fixed experiment.

The objective is to detect all *significant* rules of the form $\text{Premise} \rightarrow \text{Consequent}$, which correlate gene patterns of expression in Premise with those of Consequent.

The significance of each rule is quantified by means of its *support* and *confidence*. The *support* of a rule is the number of instances (i.e. patterns) that contain both the itemset of its premise and the itemset of its consequence. The support is estimated by counting the number of instances N_{PC} that contain both the Premise and the Consequence of the rule. At the application of concern, a support of 8 for example, means that we require the gene expression pattern that corresponds to the association rule to hold for at least of 8 experimental conditions, in order to obtain statistical confidence.

However, in the fuzzy case since an item appears in a transaction with an associated degree, the support of a rule becomes the sum of degrees of the merged itemset corresponding to the Premise and Consequence of the rule (i.e. c*oncat*(Premise, Consequence)), summed over all the transactions, i.e.

$$N_{PC} = \text{Support}(\text{Premise} \rightarrow \text{Consequence})$$
$$\equiv \sum_{t \in \text{Transactions}} \text{Degree}_t(\text{concat}(\text{Premise}, \text{Consequence}))$$

The parameter Degree_t above is computed as the product of the degree of memberships of the individual item-sets, e.g.

$\text{Degree}(g_1 L = 0.9, g_4 H = 0.8) = 0.9 \cdot 0.8 = 0.72$.

A related important parameter is that of the *confidence*. We denote by N_p the number of instances that contain at least the itemset of the premise. Then the *confidence* is defined with the ratio $\text{confidence} = \frac{N_{PC}}{N_P}$, and it estimates the degree to which the condition at the premise "causes" the consequence. The statistical significance of the rule confidence estimation increases with increasing rule support, e.g. clearly a computed confidence $\frac{3}{4}$, computed with the outcome of 4 experiments is not as reliable an estimate as the same confidence, $\frac{3}{4} = \frac{300}{400}$, computed with 400 experiments. As previously, in the fuzzy case each transaction contains to a varying degree the itemset of the premise, thus $N_P \equiv \sum_{t \in \text{Transactions}} \text{Degree}_t(\text{Premise})$. The objective is to generate confident rules, i.e. rules that adhere to a specified *minimum confidence*.

4. Fuzzy Frequent Pattern Tree

A compact data structure is designed in accordance with [10] and based at the following observations:

- Since only the frequent items play a role in the frequent pattern mining, it is necessary to perform one scan of the data in order to identify the set of frequent items. However, the frequency of an "item" is defined according to the fuzzy membership functions as described in Section 3.
- If the information of the *fuzzy association* at the appearance of a *set* of frequent items can be captured with a compact data structure, we can avoid the repeatitive scanning of the original database.
- When multiple "transactions" share to some *degree* a set of frequent items, we merge the shared fuzzy sets by aggregating their co-occurrences with a count value.

Conditions	**Condition's (i.e. Transaction's) Items** (Notation: $g_i(m_{ki}^L, m_{ki}^H)$)	**Condition (i.e. Transaction) Frequent Items**
C_1	$[g_1(0.95, 0.05)], [g_2(0.1, 0.9)], [g_3(0.1, 0.9)], [g_4(0.4, 0.6)]$	$(g_3 H, 0.9)$, $(g_1 L, 0.95)$, $(g_2 H, 0.9)$
C_2	$[g_1(0.9, 0.1)], [g_2(0.8, 0.2)], [g_3(0.05, 0.95)], [g_4(0.2, 0.8)]$	$(g_3 H, 0.95)$, $(g_1 L, 0.9)$, $(g_2 L, 0.8)$, $(g_4 H, 0.8)$
C_3	$[g_1(0.8, 0.2)], [g_2(0.2, 0.8)], [g_3(0.1, 0.9)], [g_4(0.5, 0.5)]$	$(g_3 H, 0.9)$, $(g_1 L, 0.8)$, $(g_2 H, 0.8)$
C_4	$[g_1(0.1, 0.9)], [g_2(0.8, 0.2)], [g_3(0.4, 0.6)], [g_4(0.3, 0.7)]$	$(g_2 L, 0.8)$, $(g_4 H, 0.7)$

Table 1 Example of the Fuzzy Frequent Pattern Mining Algorithm

The *global frequency list* is obtained by processing all the transactions of the database. The first processing step is the thresholding of the fuzzy membership functions with a threshold θ. Values smaller than θ are ignored since it is implied t hat the condition possesses very marginally the gene at the corresponding state, e.g. for a value of $g_1 = 0.6$ the “item” $g_1 L$ is ignored with $\theta = 0.7$. The second step is to sum over all the transactions. Consider for example the data of Table 1. Setting a threshold of $\theta = 0.7$ we obtain the frequencies:

$$FL_0 = [(g_1 L: 2.65), (g_1 H: 0.9), (g_2 L: 1.6), (g_2 H: 1.7), (g_3 L: 0), (g_3 H: 2.75), (g_4 L: 0), (g_4 H: 1.5)]$$

Requiring a support of at least 1.5 the (ordered) global frequency list becomes:

$$Flist = [(g_3 H: 2.75), (g_1 L: 2.65), (g_2 H: 1.7), (g_2 L: 1.6), (g_4 H: 1.5)] .$$

After constructing the (global) frequent items list we construct the frequent items for every condition, i.e. those items of the condition that appear in the global frequency item list (i.e. the Condition Frequent Items column of Table 1).

Clearly, any item that is not by itself frequent, it cannot be frequently associated with any other item (i.e. the celebrated *Apriori* principle). Therefore, any algorithm should focus on these items ignoring the rest. We order these *condition frequent items* according to the global frequent items list.

After constructing the global frequency list and the condition frequent items list (e.g. Table 1) we can proceed by organizing the information at the Fuzzy Frequent Pattern Tree (FFPT).

Algorithm *Fuzzy Frequent Pattern Tree Construction*

Input: A transaction database for which with some items we can associate a degree of appearance (e.g. an overexpression of a gene in a condition with a degree 0.9). Also a minimum support threshold θ.

Output: The Fuzzy Frequent Pattern Tree (FFPT) of the database.

Method: The steps for the construction of the Fuzzy Frequency Pattern Tree are as follows:

- Computation of the set of global frequent items with an aggregation of their fuzzy counts of appearance. Let *Flist* be the list of these items (as the example previously).
- Creation of the root of the FFPT. For the data item of each transaction *Trans* we do the following. We select the frequent items in *Trans* and we sort them in accordance with *Flist*. This step creates the *condition items frequent lists* described previously.We denote the sorted condition items frequency list of *Trans* by *[head | Tail]* where *head* is the item at the head of the list and *Tail* the remaining frequent items. We denote by $\mu_{head}(\text{Trans})$ the membership degree of the item head at the transaction Trans. The insertion is performed with a function that implements the following algorithm for each transaction *Trans.*

```
FFPTreeNode insertFFPT( head, Tail, TreeNode)
  if TreeNode has a child node Child such that
Child.ItemID = head then
        Child.count = Child.count+ $\mu_{head}(\text{Trans})$
      else {
    create a new node Child;
    Child.parent = TreeNode;
    Child.count = $\mu_{head}(\text{Trans})$ ;
    connect Child to the Node-Links structure
      } // else
  return Child;
```

For every transaction (i.e. condition at the gene's example), we insert the whole conditions item list frequent item list by using the following pseudocode:

```
InsertionPoint = root;
while the transactions item list is not empty do
      get head and Tail element;
      InsertionPoint = insertFFPT(head, Tail,
InsertionPoint);
 end;
```

In order to facilitate the subsequent data mining we construct a *Header Table* data structure that has one entry for each frequent 1-itemset (i.e. the items appearing at the global *Flist*). This entry keeps the name of the item, its total frequency count and the *node links* pointers that connects with a linked list all the information concerning the item at the Fuzzy FPT. The Fuzzy FPT obtained with the data of Table 1 by the construction algorithm is illustrated with Figure 1.

5. Mining the Frequent Patterns with the Fuzzy Frequent Pattern Tree

Similarly to the crisp Frequency Pattern Tree of [10] the FFPT also exhibits the *node-link* property, i.e.:

Node-link property The detection of all the possible patterns containing a frequent pattern p_f can be accomplished by following the node links of the header table entry for p_f .

Concerning the example of Fig. 1, starting from the item $g_4 H: 1.5$ of the item header table we derive (following the node links from the header table) the corresponding paths and patterns for this item from the frequent pattern tree: the path $g_3 H, g_1 L, g_2 L, g_4 H$ for supporting condition C_2 and the path $g_2 L, g_4 H$ for the supporting condition C_4 . Therefore to study which items appear together with $g_4 H$ only the prefixes $(g_3 H, g_1 L, g_2 L: 0.8)$ and $(g_2 L: 0.8)$ should be considered. These items form the *Conditional Fuzzy Pattern Base* (CFPB) (i.e. the frequent pattern base conditioned on the $g_4 H$ existence). The frequent item list for the above CFPB is $g_2 L: 1.6$. Therefore, we derive one frequent 2-itemset, the $(g_4 H: 1.5, g_2 L: 1.6)$.

We proceed with the mining of the CFPB corresponding to element $g_2 L: 1.6$. From the node links of the Header Table for $g_2 L$ we derive the following path from the FFPT: $(g_3 H, g_1 L: 0.8)$. Trying to construct the CFPB for $g_2 L$ on the basis of the above pattern, we derive an empty frequent list (neither item fullfills the requirement of support ≥ 1.5). Thus, we do not mine any frequent itemsets for $g_2 L$.

Continuing with $g_2 H: 1.7$ the path $(g_3 H, g_1 L: 1.7)$ is derived. Since we end up with a single linear path, that adheres to the minimum support requirements, it is evident that the frequent itemsets consists of all the possible combinations, i.e. $(g_3 H, g_2 H), (g_1 L, g_2 H), (g_3 H, g_1 L, g_2 H)$

For the item $g_1 L: 2.65$ we derive the path consisting of the single element $g_3 H: 2.65$ and thus $(g_1 L, g_3 H)$ is another frequent itemset.

For the item $g_3 H: 2.75$ we cannot derive any path and thus we conclude the mining process.

We can state now the FFPT mining algorithm in general terms:

```
for all frequent patterns of the HeaderTable do
   // path detection for the current frequent item fitem
   NodeListPointer = head link pointer for the item fitem from the Header Table
   Conditional Pattern Base (CPB) = null;
   while NodeListPointer not null do
      CurrentPath = path from root to the current frequent item pointed by NodeListPointer;
         CPB = CPB + AdjustFrequencyOfItems (CurrentPath, NodeListPointer, fitem)
         NodeListPointer = NodeListPointer.next;
   end while
   // mining of CPB
   if the CPB is a single path
      ouput all the combinations as frequent items
   else
      RecursiveMine(CPB)
      Disconnect currently examined frequent item fitem from the FFPT since all frequent items concerning it were examined;
```

The function *AdjustFrequencyOfItems(Path, Node, Item)* adjusts the frequency of the item *Item* over the whole path *Path* as the frequency of the last node *Node* of the path.

6. Derivation of fuzzy association rules from the frequent item lists

After the detection of the frequent patterns from the Fuzzy FPT data structure, we can formulate the last phase of the mining approach, i.e. the extraction of fuzzy association rules. This phase examines for each frequent item set, the significance of all the possible associations.

For example we consider the frequent pair $(g_2 L, g_4 H)$ that we have derived. There are two possible associations:

1. $g_2 L \rightarrow g_4 H$ implying that a low (underexpressed) value of gene g_2 effects on high(overexpressed) value for the gene g_4 and
2. $g_4 H \rightarrow g_2 L$ implying that gene g_4 has an inhibitive effect on the expression of gene g_2.

Clearly, these two "directions" of association correspond to distinct biological rules, that however are not mutually exclusive or competitive. For the particular example both can be true, i.e. both g_2 can inhibit g_4 (case 1) and g_4 can inhibit g_2. Let now consider the study of the direction of case 1, i.e. $g_2 L \rightarrow g_4 H$.

We proceed by considering how much each condition supports this direction first separately and then we aggregate over all the conditions. Denoting the format of an association rule as $Premise \rightarrow Consequent$, we evaluate the *Fuzzy Confidence Value (FCV)* parameter [4], computed as:

$$\frac{\sum_{t \in \text{Transactions}} \text{CorrelationOfPremiseAndConsequentItemSets}(t)}{\sum_{t \in \text{Transactions}} \text{DegreeOfPremiseItemSet}(t)}$$

In order to understand this formula consider three transactions at which $(g_2 L, g_2 H)$ appear with fuzzy counts $(0.9, 0.6), (0.8, 0.7), (0.7, 0.5)$ respectively. Then

$$FCV(g_2 L \rightarrow g_2 H) = \frac{0.9 \cdot 0.6 + 0.8 \cdot 0.7 + 0.7 \cdot 0.5}{0.9 + 0.8 + 0.7} = 0.6$$

while

$$FCV(g_2 H \rightarrow g_2 L) = \frac{0.9 \cdot 0.6 + 0.8 \cdot 0.7 + 0.7 \cdot 0.5}{0.6 + 0.7 + 0.5} = 0.8$$

We have applied the algorithms for analyzing microarray expression data from the budding yeast Saccharomyces cerevisiae. These data are public available from the Stanford web site. They were generated by studying this fully sequenced organism with microarrays, containing essentially every Open Reading Frame (ORF). The samples used were collected at various time points during the diauxic shift, the mitotic cell division cycle and sporulation. The data set consists of 80-element gene expression vectors for 6,221 genes.

The format of the extracted rules displays the genes involved at the *Premise* and those at the *Consequence*. Therefore by examining them we can obtain evidence on some possible gene regulation relations at the presented application. However these rules should be further elaborated. Multiple conditions either at the Premise or the Precondition are in an implied conjunctive form.

For example, some of the extracted rules are:

YAL001C=High and YAL003W=High => YAL002W=High
YAL001C=Low ==> YAL003W=Low and YAL004W=Low
YAL003W=Low ==> YAL001C=Low and YAL004W=Low
YAL004W=Low ==> YAL001C=Low and YAL003W=Low
e.g. at the first rule the overexpression of the genes YAL001C and YAL003W is associated with an overexpression of the gene YAL002W.

7. Conclusions

The paper has extended the Frequent Pattern Tree approach of [10] at the fuzzy domain. The presented Fuzzy Frequent Pattern Tree algorithm confronts effectively the combinatorial complexity problem of the Apriori based approaches by avoiding the candidate generation phase.
Also, it outperforms the crisp Frequent Pattern Tree by avoiding the discretization of the values of the attributes (i.e. the "items") of the former approach that implies a loss of information.
For many applications, this loss either deteriorates significantly the results, or is completely intolerable. The derived Fuzzy Frequent Pattern Tree algorithm retains the efficiency of the crisp Frequent Pattern Tree, while at thesame time the careful updating of the fuzzy sets at all the phases of the algorithm tries to preserve most of the original information at the data set.
We presented an application of the Fuzzy Frequent Pattern Tree approach to the difficult problem of the discovery of fuzzy association rules between genes from massive gene expression measurements. These rules can complement approaches based on regulatory gene network construction [2, 7].
The implementation of the Fuzzy Frequent Pattern Tree algorithm is in the Java programming language and it is available upon request from the authors.

References

[1] Mavroudi Seferina, Papadimitriou Stergios, Bezerianos Anastasios, "Gene Expression Analysis with a Dynamically Extended Self-Organized Map that Exploits Class Information", *Bioinformatics*, Vol. 18, no 11, 2002, p 1446-1453

[2] Friedman, N., M. Linial, I. Nachman, and D'Peier, "Using Bayesian networks to analyze expression data", *J. Comp. Bio. 7*, 2000, 601-620,

[3] Ian H. Witten, Eibe Frank, *Data Mining*, Morgan Kaufmann Publishers, 2000

[4] Sushmita Mitra, Tinku Acharya, *"Data Mining: Multimedia, Soft Computing, and Bioinformatics*", John Wiley & Sons, 2003

[5] Chad Creighton, Samir Hanash, "Mining gene expression databases for association rules", *Bioinformatics*, Vol. 19, no. 1, 2003, pp. 79-86

[6] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases", in *Proceedings of 20th International Conferenence on Very Large Databases*, pp. 478-499, September 1994

[7] Seiya Imoto, Sunyong Kim, Takao Goto, Satoru Miyano, "Bayesian Network and Nonparametric heteroscedastic regression for nonlinear modeling of genetic network", *Journal of Bioinformatics and Computational Biology*, Vol. 1, No. 2, (2003), 231-252

[8] S. Papadimitriou, S.D. Likothanassis, “Kernel-Based Self-Organized Maps trained with Supervised Bias for Gene Expression Data Analysis”,

Journal of Bioinformatics and Computational Biology, Imperial College Press,January 2004, Vol. 1, No. 4, 647-680

[9] R. Srikant, Q. Vu, and R. Aggrawal, "Mining association rules with item constraints", In *Proc. 1997 Int. Conf. Knowledge Discovery and Data Mining (KDD'97),* p. 67-73, Newport Beach, CA, Aug. 1997

[10] Jiawei Han, Jian Pei, Yiwen Yin, and Runying Mao, "Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach", *Data Mining and Knowledge Discovery*, 8, 53-87, 2004

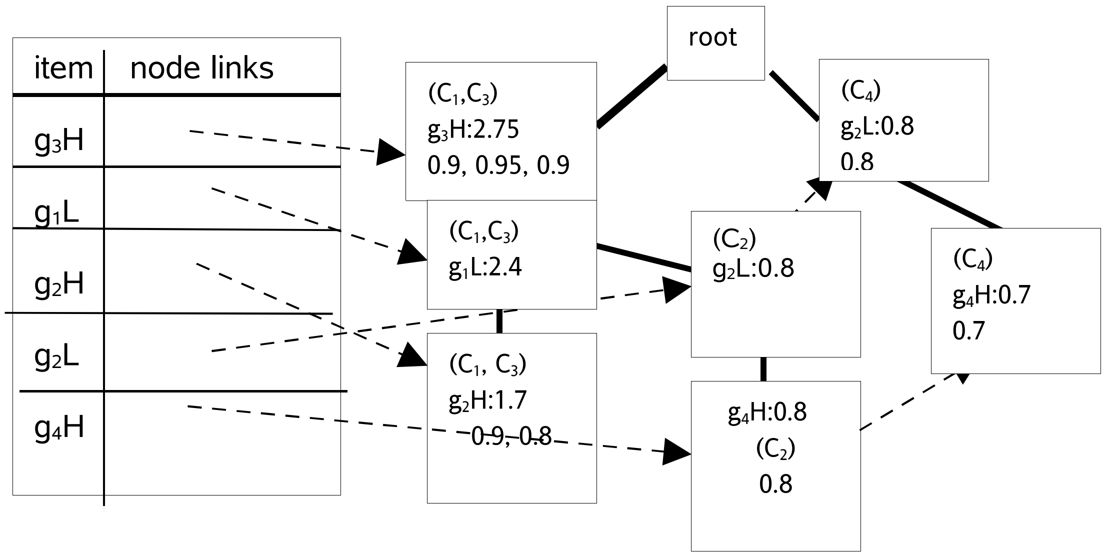

Figure 1. The Fuzzy Frequent Pattern Tree for the data of Table 1