Ontology-driven Rule Generalization and Categorization for Market Data

Dongwoo Won and Dennis McLeod
Department of Computer Science
University of Southern California
{dwon, mcleod}@usc.edu

***Abstract*—Radio Frequency Identification (RFID) is an emerging technique that can significantly enhance supply chain processes and deliver customer service improvements. RFID provides user with efficient tracking on the flow of products throughout the wholesale process. However, the large number of information that has been generated from such a process creates difficulty in extracting and analyzing useful information. In this paper, we propose a method to mine the large data sets that allows smaller and more relevant search space compared to the original data sets. Our work is constructed from the following approaches: ontology-driven rule generalization which concentrates on controlling the level of items, and rule categorization using hierarchical association rule clustering that group the generated rules from the given problem space into hierarchical search space. The detailed steps for rule generalization based on ontologies are presented, as well as the algorithms for rule categorization using hierarchical association rule clustering is developed. Our experiment proves the feasibility of our work which shows the significant reduction of the search space by decreasing the number of rules to be looked at and increasing the relevance among the rules.**

***Keywords*—Association Rules, Data Mining, Hierarchical Clustering, Ontology, RFID.**

I. Introduction

Radio Frequency Identification (RFID) is a wireless technology that nowadays getting large attention from many organizations. RFID can identify objects using radio frequency by storing customized information into RFID tags which consist of antenna that detects radio waves and responds with signals, and a chip that stores and manipulates data. Then there exists a RFID reader that recognizes the stored information. Different from bar code that requires a contact with the reader, RFID, on the other hand, do not need line of sight identification. Therefore, RFID is fast in reading, saves labor cost, and enables multiple reading. It is also possible to modify the stored information and track the location of the tags.

In [2], the author suggests RFID data set in the form of (*EPC, location, time*). We are going to add a new *price* factor to those three factors that is more suitable to our shopping cart application. Using this information, people can track the movement of the customer, duration and location of staying in particular section, and the type of product that the customer buys. As a result, we can analyze this useful information either to place the goods to prevent customer from assembling in a particular section or to place the goods that are bought together in nearby section to arouse customers' interest. However, these kinds of tracking processes produce tons of data, so we need an efficient technique to mine the data. In this regard, we use the approaches that are introduced in [1] - generalizing association rules based on domain ontology and hierarchically clustering association rules by their relevance.

The use of ontologies allows us to have pre-knowledge about the data. Ontologies also provide a way to represent information or knowledge that includes the key concepts and the inter-relationships between them. We are using this basic idea to generalize and reduce the item set which as a result produces fewer, but more closely associated rules. With these generalized association rules, we find sub-categories consisting of rules that are more relevant to the generalized association rule by hierarchically clustering association rules by their relevance. This is an approach to combine similar association rules to reduce the large number of rules and to sort the rules into more relevant search space.

Our approach of ontology-driven rule generalization and rule categorization using hierarchical association rule clustering produces as a result, smaller space of more relevant clusters of item sets that are easy to understand and easy to analyze for marketing purposes. The rest of this paper is organized as follows. The works that are related to our approach is shown in section 2. In section 3 and 4, the two approaches of using ontology-driven rule generalization and rule categorization are presented. In section 5 we discuss our evaluation and experimental results. We summarize our conclusion and discuss our future work in section 6.

II. Related Works

Mostly, the recent research on RFID challenges on three big issues: privacy and security, data standards, and large data management. Privacy is needed because RFID tags enable tracking of items or tracking of customers without any understanding or knowledge of them. This surely causes the possibility of hidden attacks. [6] talks about the necessity of privacy for every stakeholder involved in the deployment of

RFID technology. [5] discusses about a specific technology to protect privacy using trusted computing by splitting the RFID reader. A call for RFID standards to unify different tag data, application-device communication, and device-tag communication is continuously getting wide attention, because of the immaturity of the RFID market and few existences of RFID standards [12, 13, 14]. But, the generation of large set of data by the RFID reader causes the most challenge among the others. The generated data has to be filtered to remove any duplicate or redundant data and the consolidated data has to be stored in databases or data warehouses for effective use. Past work on RFID application-specific issues such as the existence of noisy and duplicate readings in large volume real-time RFID data streams is concerned in [7]. A temporal data model is proposed in [8] that supports RFID data tracking and monitoring. [2, 3, 4] has presented a model for warehousing RFID data to support high-level analysis in multidimensional space. Although our model is application-specific, we support large RFID data set, remove redundant data using ontology-based rule generalization, and categorize rules using hierarchical association rule clustering [1].

III. RULE GENERALIZATION

Domain ontologies have been a useful mechanism to represent knowledge in a specific application. The underlying domain knowledge in ontology represents the concepts and the relationships between them. Due to the hierarchical structure of ontologies, we facilitate the usage for generalizing our data set. In our case, we attempt to capture hierarchical semantics by mapping items or EPC code to predefined concepts.

A. Data Set

The data set that we use is a cleaned version of the streamed RFID data set. As we introduced earlier, we consider item, location, time, and price as the four factors. Time and price are numerical values, so analyzing the two values is a simple task. We concentrate on analyzing item and location as our main task. Following is the form of our data set.

$$\mathrm{Cart}_i = \{\mathrm{Itemlist}\}$$
$$\mathrm{Itemlist} \supset \mathrm{I}_j = \{\mathrm{location}, \mathrm{price}\}[\, t_s, t_f \,]$$

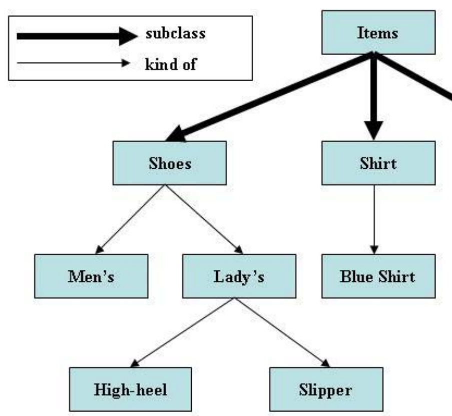
where i = number of shopping carts (i.e. customer), j = item number (EPC), t_s = starting time, t_f = ending time. When a customer enters the market, he or she takes a $Cart_i$ and buys multiple items (*Itemlist*). We consider one cart per customer at this stage. A cart includes one or many items and records the location and price for each item. In case of multiple items, it also records the first and last timestamp for purchasing that item.

B. Ontology Creation

The ontologies creation needs some effort to find the terms and the relationship between them. Domain ontologies are created manually, since the size of our shopping cart ontologies is small. Our domain ontology consists of 50 nodes including the root node. A small ontology is useful in a way to get global agreement among people [15] and currently there are no automatic ontology creation tools available to generate large ontologies. It is always possible to extend our ontologies later with merging to other domain ontologies to generate global ontologies. But, to accomplish that, a new issue of ontology integration must be dealt. This is an important, yet another huge research area. We are not going to address this problem at this point.

C. Item Generalization

When business person do marketing analysis, they do on various level of details. Using the domain ontologies we created, it is possible to analyze the data more efficiently. In Figure 1, we have shown the sample domain ontologies of shopping items. All items with general concepts are placed in the next level of the root node *Items*. Each general item has all the related detailed concepts as its child nodes. The general concepts are compared with the section name in a supermarket. The middle level of concepts is the sub-categories within that section and the lowest level of concepts is the items within that section.



Name	Cur	Par	Top
Items	0	-	-
Shoes	1	0	-
Coffee	2	0	-
Shirt	3	0	-
Men's	4	1	1
Lady's	5	1	1
High-Heel	6	5	1
Slipper	7	5	1
Blue Shirt	8	3	3

Fig. 1. Sample domain ontologies for shopping items. The table shows hierarchical relationship among the nodes. “Cur” is the number of current node, “Par” is the number of parent node, and “Top” is the number of the second level node.

D. Supermarket Analysis Types

In supermarket data analysis, association rule mining technique is used. Our goal here is to produce less but more relevant rules for efficient analysis. Below are the three terms of analysis types that we are going to use throughout this paper.

Section-To-Section Analysis is to see the relationship among the section, we do not need the full item list that belongs to that section, but only the concepts from the second level in our ontologies. For example, let us consider the relationships among Shoes, Coffee, and Shirt sections. In Figure 2, three kinds of items have converted to two types of general item name – shoes and shirt. This enables us to produce less number of association rules and allows us to make simple but more relevant analysis about the relationship among the sections.

In-Section Analysis is to see the relationship among the items within one particular section, all the lowest level of children nodes within one general concept are used for this analysis. For example, let us consider the relationship among High-heel and Slipper that is under Shoes section. This allows us to look at only the items that belong to that particular section, and therefore increases the relevance among the generated association rules.

In-Section-To-In-Section Analysis is to see the relationship among the items in different sections, we need to use the current item list. This means that the size of generated association rules is enormous. Let us consider the example in Figure 2. Slipper, Blue Shirt, and High Heel are used to produce association rules. Therefore, no rule reduction is expected for the lowest level of detail. The second type In-Section Analysis is a kind of In-Section-To-In-Section Analysis.

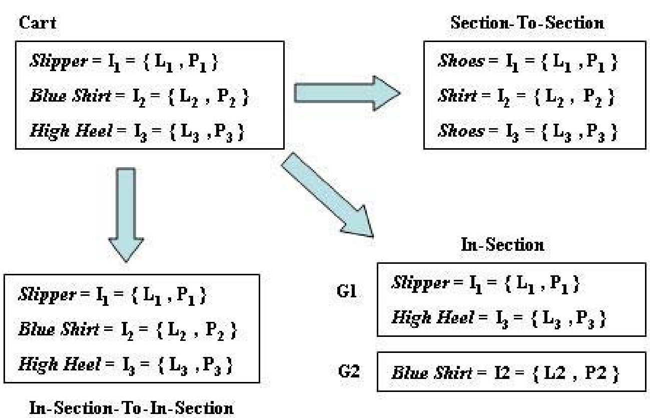


Fig. 2 Rule generalization for supermarket data. Itemlist for a customer is converted using the three supermarket data analysis types.

IV. RULE CATEGORIZATION

As we have shown in section 3, rule generalization is used to generate fewer association rules. Our second method is to use hierarchical association rule clustering to categorize the originally created rules into more relevant group of rules.

A. General Idea

We have seen three types of approach for market data analysis. We can reduce high number of rules in *Section-To-Section* analysis. But, there is not much gain for the other two supermarket analysis types. Instead, we need extra effort to group the items for the *In-Section* analysis and we hardly can get rule reduction for *In-Section-To-In-Section* analysis. So, we suggest an approach to use the generated association rules from *Section-To-Section* analysis to handle the rules in other two types of analysis.

In Figure 3, *R1* and *R2* represent the association rules generated after the rule generalization. For *Section-To-Section* analysis, we just use the generalized association rules. For the two other types of analysis, we generalize the original rules to find a match to *R1* and *R2*. When the form of the rules is like *R2*, where all generalized item names are the same, this means that all the original rules belong to the same section and we can do *In-Section* analysis. For the original rules, we cluster them by relevance from bottom to top. We explain the detailed algorithm in next section. Other than the form like R2, we can do *In-Section-To-In-Section* analysis. Same cluster algorithm is used for this analysis.

The good thing about this rule categorization is that once we cluster the original rules under the generalized rule like *R1* or *R2*, we do not need to scan all the association rules each time, when we try to analyze the data set. Instead, we select a generalized rule and work with the sub-categories and rules that belong that generalized rule. Also, since the rules are hierarchically clustered by relevance, we can choose the level of detail for working with the rules. To search only a category instead of the whole association rule is an enormous plus for efficient analysis.

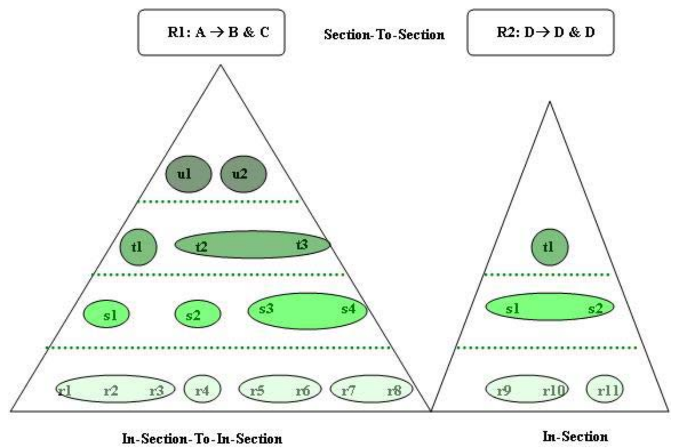


Fig. 3 Rule Categorization by the three supermarket analysis types. Highly relevant rules are grouped together to a new sub-category until no relevant rule exists.

B. HARC Algorithm

Based on the previous idea, we give the algorithm for the complete process of hierarchically clustering the generated association rules. The basic idea of the Hierarchical Association Rule Clustering (HARC) algorithm is to recursively group rules according to the relevance that satisfies the defined threshold value. We already show the sample output of our algorithm in Figure 3. R_1, R_2, …, R_n are categories for the data set. Each category contains a tree of sub-categories (clusters), where each non-leaf sub-category is the unification of all the child nodes and leaf sub-category is equivalent to the original rule. By grouping the rules hierarchically, we also can prune trivial rules that do not belong to any sub-categories per each level [16].

In Figure 4, the HARC algorithm is presented. It has three inputs and one output. The generalized association rule *R* and original association rule *r* is generated from the previous section. The threshold value ε is a user-defined value, usually set as 0.75 at the beginning. The HARC algorithm outputs all the categories and all the sub-categories generated.

Algorithm: Hierarchical Association Rule Clustering

Input: Generalized Association Rules $R_1, R_2, R_3, \ldots, R_n$
Original Association Rules $r_1, r_2, r_3, \ldots, r_m$
Threshold ε

Output: Rule Categories $C_1, C_2, C_3, \ldots, C_n$

Steps:

1. Generalize rules $r_1, r_2, r_3, \ldots, r_m$ to $G_1, G_2, G_3, \ldots, G_m$ using table from Figure 1.
2. For all rules $G_1, G_2, G_3, \ldots, G_m$, search $R_1, R_2, R_3, \ldots, R_n$ for identical rules.
   For 1 to n
     For 1 to m
       If $G_m == R_n$ **Then** $r_m \rightarrow R_n$
3. Calculate *relevance* and group rules to sub-categories *s*.
   i:= 1
   T:= number of rules in R_i
   For i to T
     For 1 to T
       If (!i)
         rel ← r_i *relevance* r_T
         If rel > ε
           s_i ← groupRules()
   i++
4. Adjust threshold and do step 3 until rel >= ε .
5. Prune trivial association rules.
6. C_i ← all sub-categories $s_1, s_2, s_3, \ldots, s_i$
7. Do step 3 through 6 for the rules $R_2, R_3, R_4, \ldots, R_n$.
8. **Return** $C_1, C_2, C_3, \ldots, C_n$

Fig. 4 Hierarchical Association Rule Clustering (HARC) Algorithm

Our algorithm runs as follows. The whole process runs iteratively until all rules are grouped and set to the generalized rules. First, using the table in Figure 1, generalize all the original association rules $r_1, r_2, \ldots, r_m$ to new rules $G_1, G_2, \ldots, G_m$. Second, we store the original association rules *r* to the generalized association rules *R*, when association rules *r* is equal to *G*. For example, we store rule r_1 to R_1 when r_1 is equal to G_1. We do this iteratively for the all rules from G_1 to G_m. This require us $O(n*m)$ steps to run this step. But, as the data size gets larger and produces larger rules, size *n* becomes so large that *m* is negligible. After, storing all rules, we calculate the *relevance* and group the rules to a new sub-category according to their *relevance* value. The equation for the *relevance* between two association rules r_1 and r_2 are given below:

$$\mathbf{relevance(r_1,r_2)} = \frac{\mathbf{2*\{lhs(r_1)\cap lhs(r_2)\}+\{lhs(r_1)\cap rhs(r_2)\}+\{rhs(r_1)\cap lhs(r_2)\}}}{\mathbf{all(r_1)\cup all(r_2)}}$$

where r_1 has the form of $lhs(r_1) \rightarrow rhs(r_2)$, r_2 has the form of $lhs(r_1) \rightarrow rhs(r_2)$, $lhs(r_1)$ means all the items on the left side in rule r_1, $rhs(r_1)$ means all the items on the right side in rule r_1, and $all(r_1)$ means all the items in rule r_1. The *relevance* is the total number of common items in the two association rules over the number of total items. Here, we give more weight to common items on the left hand side of the rules. The value of *relevance* varies from 0 to 1.5. Next to calculating the *relevance*, we use the threshold value to group rules. We combine the two values when *relevance* is greater than the default threshold value. As the level of category gets higher, threshold value is adjusted to a new value. The adjusting value is pre-defined by user and is added to the previous threshold value. In our system, it is set to 0.05. The function *groupRules()* is as follows:

$$\mathbf{groupRules(r1,r2)\{\, s := lhs(r1) \cup lhs(r2) \rightarrow rhs(r1) \cup rhs(r2)\,\}}$$

where *s* becomes the new sub-category. For example, given two association rules {ab → c, bd → c}, the HARC algorithm can generate a sub-category of {abd → c} when the *relevance* of the two rules is larger than the threshold. We continue this process until no other rule exceeds the threshold value. Also, there supposed to be overlaps among the sub-categories, the result becomes a group of hierarchical clusters. Along this step, we are able to prune rules that do not belong to any sub-categories and has low *relevance* value. Next, we do the iteration in creating the sub-categories for the whole generalized rule R_1 to R_n. Finally, the HARC algorithm returns the categories and sub-categories C_1 to C_n.

V. RESULTS

Here, we perform an experiment of our process and algorithm. All experiment is implemented using an Intel Pentium 1.4GHz System with 512MB of RAM running Windows XP. We generate association rules using a free Data Mining Java tool called Tanagra [17] which uses the apriori algorithms presented in [18].

A. Data Set

The complete set of our data set consists of 600 rows of customer and 40 columns of different items. In order to evaluate out method properly, we have performed our experiment by dividing the data set into 10 different data sets. To test the number of generated rules depending on the number of items, we substitute the number of items with 10, 20, 30, and 40 at fixed number of 10 customers. Also, to test the number of generated rules depending on the number of customers, we replace the number of customers with 10, 20, 100, 300, and 600 at fixed number of 20 items.

B. Rule Generalization (RG)

Our experiment is to see how much rule is reduced by rule generalization. The first experiment is performed by increasing the number of items while the second experiment is carried out by increasing the number of customers. To generate the association rule, apriori algorithm is selected and we are using the default support, confidence and lift values set in Tanagra. The minimum support is set to 0.33, minimum confidence to 0.75, and maximum item set size to 4. The minimum lift value is set to 1.1 for the first experiment, but it is changed to 1.005 for the second experiment, since our data set does not produce significant number of rules with the default lift value. Different from [1], our evaluation concentrates more on the number of rules produced rather than support, confidence, and lift value. Our data set is not real data, but randomly generated test data. So, the content is not meaningful enough to analyze with the above three values.

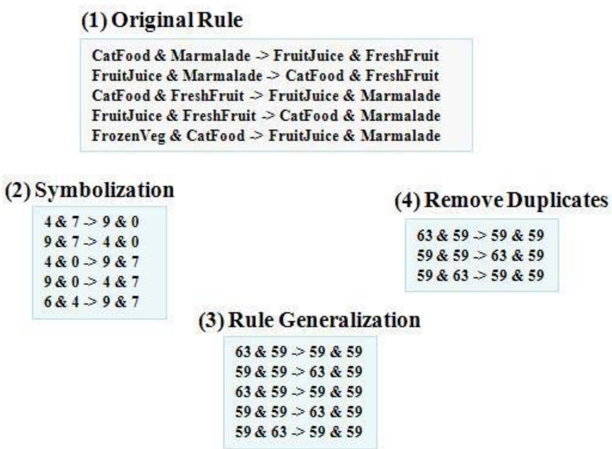


Fig. 5 The steps for Rule Generalization. An example Rule Generalization from Customer=10, item=10 data set.

1) Steps for RG

In Figure 5, it shows the top five rules produced from our first data set (customer=10, item=10) as a rule generalization example. (1) shows the rules produced from Tanagra. In step (2), to simplify our process, we symbolize each item name to a defined item code using our item table from Figure 1. Then in step (3), we convert the current item code to its top item code from the item table to generalize each rule. For example, the top item code for item code 0(Fresh fruit), 7 (marmalade), and 9(Fruit juice) is 59 (food), while 4(Cat food) is 63. We do this rule generalization for all items. Lastly in step (4), we finalize our rule generalization by removing duplicate rules from the generalized rules. By removing redundancy, we produce fewer rules as we can see from the result below.

Customer = 10

ITEM	10	20	30	40
Original #ofrules	472	2432	15056	87867
#ofrules after RG	29	124	176	648

Item = 20

CUSTOMER	10	20	100	300	600
Original #of rules	3882	5026	19214	4578	2244
#ofrules after RG	201	202	321	185	157

Fig. 6 Results after Rule Generalization (RG). First table shows the result depending on number of items, while the second table is the result depending on number of customer.

2) Results for RG

As we briefly mentioned above, we evaluate our approach depending on the number of items and customers following the four steps for RG. Figure 7 shows the difference between the number of rules produced from the original data set and from the generalized rules. As the number of item is increased, we can see the difference of generated rules is getting larger. Figure 6 presents the exact number of rules generated in detail. Due to the limitation of Tanagra, 40 is the maximum number of items we have tested.

Figure 8 shows the results of generated number of rules depending on the number of customer. The exact number of rule generated is in Figure 6. One special thing about the result is that the number of rules first increases till some point, but decrease there after. The reason for this fluctuation is due to the sparsity of our data set, as more items are located near to the first and last row.

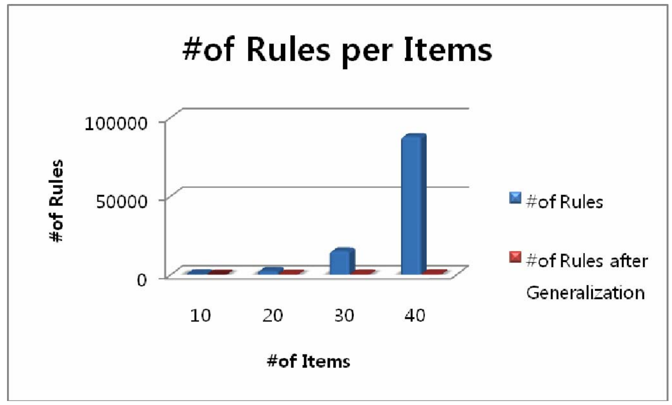


Fig. 7 Rules generated after rule generalization. The number of customer is fixed at 20, while the number of items is changed.

Fig. 8 Rules produced after rule generalization. The number of item is fixed at 20, while the number of customer is changed.

From our evaluation, it is clear that rule generalization contributes greatly in *Section-To-Section* analysis by reducing the number of rules to look at. As the number of items or the number of customers is getting larger, more rule reduction has occurred. In the following section, we use these results of generalized rules to categorize the original rules into more relevant sub-categories. We show that our approach, especially the HARC algorithm, successfully contributes to *In-Section* and *In-Section-To-In-Section* analysis as well.

C. Rule Categorization (RC)

A drawback of using apriori algorithm is that it produces large number of rules which causes difficulty of identifying important and interesting rules. Here, we prove that our approach groups relevant rules together to make *In-Section* and *In-Section-To-In-Section* analysis process simple and easier.

After rule generalization, we have all the generalized rules as well as the original rules that belong to those rules. So instead of scanning manually the whole rule set every time, we just produce a category of rules once. Then, without any knowledge behind it, user only selects an interesting rule, in our case the generalized rule, and analyzes the rules within that category only. This reduces the time as well as the number of rules to be looked at. Figure 9 shows an example for our HARC algorithm.

First, for a generalized rule, for example, 59 & 59 & 63 & → 9, we know that there exist six rules that have produced the generalized rule. Next, using the six original rules, we calculate the *relevance* among these rules. Of course, it needs {n (n-1)}/2 steps to calculate the *relevance*. In our example, 15 *relevance* values have been calculated. Then the result is compared to default threshold value 0.75. 12 rules are survived from this process, and are grouped to sub-categories by their *relevance* values.

As a result, new sub-categories that unify the original rules are created. Again, *relevance* is calculated and compared to the adjusted threshold value. The threshold value is increased per loop by 0.05. Here, all rules are accepted and grouped to a one large rule. The algorithm terminates returning the hierarchically clustered sub-categories (1) to (6) as the lowest level, (7) to (10) as the middle level, and (11) as the highest sub-category within the selected category (generalized rule).

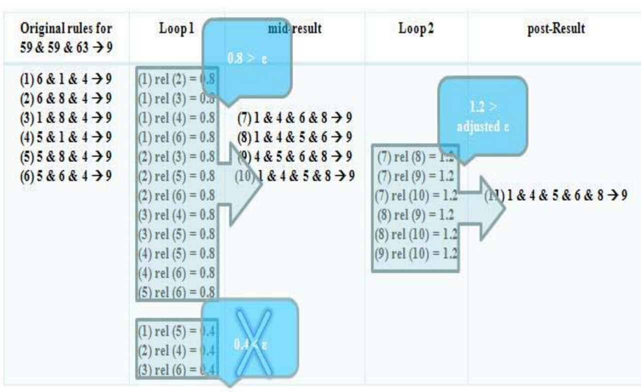


Fig. 9 Hierarchical sub-categories after Rule Categorization (RC). Relevance among rules are calculated and grouped to sub-categories according to their relevance values.

VI. Conclusion And Future Works

In analyzing marketing data, association rule mining or clustering technique has been used with limitations to provide adequate solution. In this paper, we have presented a process for mining large problem space of market data into a hierarchically structured search space that is efficient for analysis.

We have used association rule mining in three types of supermarket data analysis: *Section-To-Section*, *In-Section*, and *In-Section-To-In-Section*. Rule generalization uses domain ontologies to merge and simplify items into more general concepts. Our result shows that this step reduces the total number of rules being generated. Rule Categorization hierarchically groups the rules by relevance into new clusters, called sub-categories, in which reduces the number of rules to be looked at or searched for analysis.

For next part of our ongoing work, we plan to extend our process to support multidimensional space. Currently, we work on items only, but we want to support the relationship among location, price, and time factor as well. Also, hierarchically clustering the generalized association rules (categories) by their *relevance* can contribute to better analysis for *Section-To-Section* analysis.

Acknowledgment

This research was supported in part by the Integrated Media Systems Center, A National Science Foundation Engineering Research Center, Cooperative Agreement No. EEC-9529152, and in part by NASA's Computational Technologies Project - portions of this work were carried out by the Jet Propulsion Laboratory, California Institute of Technology under contract with NASA.

References

[1] D. Won, B. M. Song, and D. McLeod, "An Approach to Clustering Marketing Data," *2nd International Advanced Database Conference (IADC)*, San Diego, June 2006.

[2] H. Gonzalez, J. Han, X. Li, and D. Klabjan, "Warehousing and Analyzing Massive RFID Data Sets," in *Proc. 22nd International Conference on Data Engineering*, Atlanta, GA, April 2006.

[3] H. Gonzalez, J. Han, and X. Li, "FlowCube: Constructing RFID FlowCubes for Multi-Dimensional Analysis of Commodity Flows," in *Proc. 32nd International Conference on Very Large Data Bases (VLDB 2006)*, Seoul, Korea, Sept 2006.

[4] H. Gonzalez, J. Han, and X. Li, "Mining Compressed Commodity Workflows From Massive RFID Data Sets", in *Proc. 2006 Int. Conf. on Information and Knowledge Management (CIKM'06)*, Arlington, VA, Nov. 2006.

[5] D. Molnar, A. Soppera, and D. Wagner. "Privacy For RFID Through Trusted Computing", in *Proc. Workshop on Privacy in the Electronic Society WPEST '05,* Alexandria, VA, USA, Nov 2005.

[6] D. Nguyen, and A. Kobsa, "Better RFID Privacy Is Good for Consumers, and Manufacturers, and Distributors, and Retailers," in *Proc. PEP06, CHI 2006 Workshop on Privacy-Enhanced Personalization,* Montreal, Canada.

[7] Y. Bai, F. Wang, P. Liu, "Efficiently Filtering RFID Data Streams." in *Proc. 1st Int. VLDB Workshop on Clean Databases (CleanDB'06)*, Seoul, Korea, Sept. 2006.

[8] F. Wang and P. Liu, "Temporal Management of RFID Data." in *Proc. 31st International Conference on Very Large Data Bases (VLDB 2005)*, Trondheim, Norway, 2005.

[9] S. S. Chawathe, V. Krishnamurthy, S. Ramachandrany, and S. Sarma, "Managing RFID Data" in *Proc. 30th International Conference on Very Large Data Bases (VLDB 2004)*, Toronto, Canada, 2004.

[10] M.D. Mills-Harris, A. Soylemezoglu, C. Saygin, "RFID Data-Based Inventory Management of Time-Sensitive Materials." in *Proc. 31st Annual Conference of the IEEE Industrial Electronics Society (IECON'05)*, Raleigh, North Carolina, 6-10 November, 2005.

[11] S. R. Jeffery, M. Garofalakis, and M. J. Franklin, "Adaptive Cleaning for RFID Data Streams." in *Proc. 32nd International Conference on Very Large Data Bases (VLDB 2006)*, Seoul, Korea, 2006.

[12] ScanSource's RFID Edge. Legislation & Standards. Available: http://www.scansource.com/rfidedge/pages/legislation_standards.html

[13] United States Government Accountability Office. (2005, May). Radio Frequency Identification Technology in the Federal Government. GAO-05-551. Available: http://www.gao.gov/new.items/d05551.pdf

[14] M. Gerst, R. Bunduchi, and I. Graham, “Current issues in RFID standardization.” in *Proc. Workshop on Interoperability Standards – Interop-ESA conference (pp. 13)*, February 22-25th, Geneva: Hermes Science Publishing.

[15] Y. Ding and S. Foo, “Ontology research and development: Part 2-A review of ontology mapping and evolving.” *J. Information Science*, vol 28, no. 5, pp. 375-388, 2002.

[16] G. I. Webb and S. Zhang, “Removing Trivial Association Rule Discovery.” in *Proc. First International NAISO Congress on Autonomous Intelligent Systems (ICAIS 2002)*, Geelong, Australia, Canada/The Netherlands: NAISO Academic Press.

[17] TANAGRA: A Free Data Mining Software for Research and Education. 2005. Available: http://eric.univ-lyon2.fr/~ricco/tanagra/.

[18] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules.” in *Proc. 20th International Conference on Very Large Databases (VLDB-94)*, pages 487-499, Santiago, Chile, Sept. 1994.