

Rapid Development Framework of Supply Chain Knowledge Management Based on Mining Association Rules

Yudong Guo

School of Finance,

Hunan University of Technology

Zhuzhou, China

School of Economics and Management,

China University of Geosciences

Wuhan, China

Kirsty1012@163.com

Abstract—Supply chain management is the typical representative of "horizontal integration" management ideology, realize the knowledge sharing among the supply chain member enterprises is the key to victory in the fierce market competition. Associations rules is top ten algorithm and an important data mining model studied extensively by the database and data mining community. Association rules are applied to the supply chain management in this paper and acquire expected result .Finally, supply chain knowledge management system was proposed which based on the rapid implement framework.

Keywords- Rapid Development Framework; Supply Chain Knowledge Management; Association Rules; Data Mining;Horizontal Integration

I. INTRODUCTION

A supply chain is a system of organizations, people, activities, information, and resources involved in moving a product or service from supplier to customer. Supply chain activities transform natural resources, raw materials, and components into a finished product that is delivered to the end customer. Supply chain knowledge management is the management of an interconnected or interlinked between network, channel, information and node businesses involved in supply chain knowledge required by participator in a supply chain.

In data mining, association rule learning is a popular and well researched method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using different measures of interestingness. Based on the concept of strong rules, association rules were proposed by Agrawal et al in 1993. Today association rule application in many areas including business, web usage mining, intrusion detection, Continuous production and bioinformatics.

This paper build a rapid development framework and use WEKA open source library directly to implement the association rule mining algorithm that search supply chain knowledge in the supply chain knowledge database. Rapid

development framework have a great practical value that can application in various software development.

II. THEORY INTRODUCTION

A. Mining Association Rules

There are a large number mining association rules algorithms, they use different strategies and data structure and their resulting sets of rules are all the same.

Resulting sets of rules model is described as follows[1]:

$I = \{i_1, i_2, \dots, i_m\}$ is a set of items. Transaction t is a set of items, and $t \subseteq I$. Transaction Database T is a set of transactions $T = \{t_1, t_2, \dots, t_n\}$. A transaction t contains X , a set of items (itemset) in I , if $X \subseteq t$.

An association rule is an implication of the form: $X \rightarrow Y$, where $X, Y \subset I$, and $X \cap Y = \emptyset$. An itemset is a set of items and a k -itemset is an itemset with k items.

The rule holds with support s in T (the transaction data set) if $s\%$ of transactions contain $X \cup Y$. $s = \Pr(X \cup Y)$. The rule holds in T with confidence c if $c\%$ of transactions that contain X also contain Y . $\text{conf} = \Pr(Y | X)$. An association rule is a pattern that states when X occurs, Y occurs with certain probability.

The support count of an itemset X , denoted by $X.\text{count}$, in a data set T is the number of transactions in T that contain X . Assume T has n transactions. Then, for the rule $X \rightarrow Y$,

$$\text{support} = \frac{(X \cup Y).\text{count}}{n} \quad (1)$$

$$\text{confidence} = \frac{(X \cup Y).\text{count}}{X.\text{count}} \quad (2)$$

The goal of algorithm is find all rules that satisfy the user-specified minimum support (minsup) and minimum confidence (minconf).

The support of an itemset X in a transaction Database T containing n transactions is defined as:

$$support = \frac{X.count}{n} \quad (3)$$

A frequent itemset is an itemset whose support is \geq minsup.

B. The Apriori Algorithm

There are a large number of mining association rules. They use different strategies and data structures, but their resulting sets of rules are all the same that given a transaction data set T, and a minimum support and a minimum confident, the set of association rules existing in T is uniquely determined.

Any algorithm should find the same set of rules although their computational efficiencies and memory requirements may be different. This paper study the apriori algorithm for it is the best known algorithm mining association rules.

There are two steps[2-8]: first, find all itemsets that have minimum support (frequent itemsets, also called large itemsets). Second, use frequent itemsets to generate rules.

Mining all frequent itemsets' key idea is the apriori property (downward closure property) what any subsets of a frequent itemset are also frequent itemsets. Details are follows:

Step1: Iterative algo (also called level-wise search) what find all 1-item frequent itemsets; then all 2-item frequent itemsets, and so on. Its steps are:

- generate length (k+1) candidate itemsets from length k frequent itemsets, and
- test the candidates against DB

Step2: Find frequent itemsets of size 1: F_1

Step3: From $k = 2, C_k =$ candidates of size k: those itemsets of size k that could be frequent, given F_{k-1} ; $F_k =$ those itemsets that are actually frequent, $F_k \subseteq C_k$ (need to check the database once).

The candidate-gen function takes F_{k-1} and returns a superset (called the candidates) of the set of all frequent k-itemsets. It has two steps:

- join step: Generate all possible candidate itemsets C_k of length k
- prune step: Remove those candidates in C_k that cannot be frequent.

Generating rules from frequent itemsets[9-12] is Frequent itemsets \neq association rules and one more step is needed to generate association rules. For each frequent itemset X, for each proper nonempty subset A of X, let $B = X - A$, $A \rightarrow B$ is an association rule if $confidence(A \rightarrow B) \geq minconf$, $support(A \rightarrow B) = support(A \cup B) = support(X)$ and $confidence(A \rightarrow B) = support(A \cup B) / support(A)$.

If a subset X of a frequent itemset L does not generate a rule, then no need to consider generating rules from subset of X. If a rule $(L-c) \rightarrow c$ holds, then all the rules of the form $(L-x) \rightarrow c$ also holds, where x is a non-empty subset of c.

In summary, Generating rules is to recap, in order to obtain $A \rightarrow B$, we need to have $support(A \cup B)$ and $support(A)$; all the required information for confidence

computation has already been recorded in itemset generation. No need to see the data T anymore; this step is not as time-consuming as frequent itemsets generation.

III. RAPID DEVELOPMENT FRAMEWORK

Rapid development framework mainly adopts template, component technology in the web development, so as to ensure the maximum flexibility and scalability. In the process of application, achieve optimization component design in many aspects such as cache, flow control, file processing, graphics, etc. so it can be said, rapid development framework has its own unique advantages.

Compared with the existing three layer open source framework, rapid development framework's main features include:

Unified transaction management that ensure a transaction to form a single operation, if the operation is abnormal, it will roll back the transaction, only all operations success it will commit the transaction, in order to ensure the accuracy of the business logic.

Unified connection management, page level connection based on a unified management that ensure the page at the start of the operation for the connection, the connection is automatically released logout page, to avoid leakage resulting collapse pool.

Unified exception management that ensure all the error messages generated, it can be fully reflected in the final pages and ensure timely tracking and correct positioning error, ensure the stability of the program.

Component based design that can easily maintain the old version and the new version.

Based on the configuration parameters, for different supply chain knowledge module, member enterprises of different subsystems, provide flexible configuration parameters, reflect the needs of the application of different.

Flow configuration based on different members, according to enterprise users, depending on the configuration of the business process, provide demand difference based on unified version

Provides log tracking components, can be in accordance with the flexible search the specified log, convenient maintenance location problem.

Component and reusable technology is the key of rapid development framework. There are seven layers in rapid development framework, every layer has own unique function. The reusable component technology is shown as fig.1:

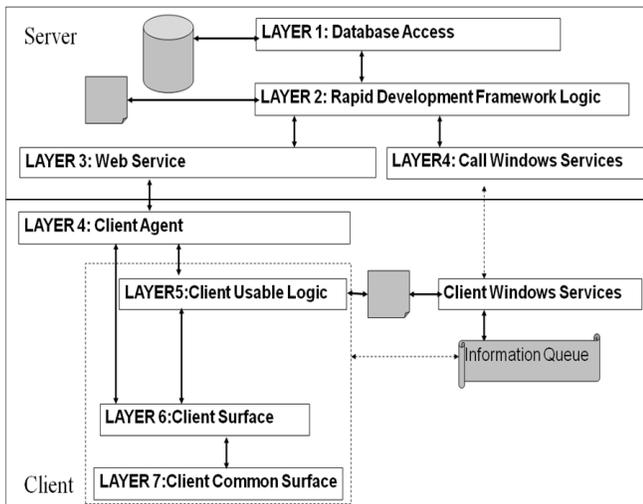


Figure 1. Multi-layer framework structure

IV. FAST IMPLEMENTATION OF SUPPLY CHAIN KNOWLEDGE MANAGEMENT SYSTEM BASED ON RAPID DEVELOPMENT FRAMEWORK

First, implement the association rule mining on Weka employing the Association Rule discovery algorithm. Specifically, a data set named supplychainknowledge.txt.

Weka's data format is special and need prepare in advance. On weka, the standard format of data set can be recognized is .csv or .arff. Therefore, in the first, supplychainknowledge.txt should be transformed to .arff.

Second, implement rapid development framework's configuration, there are four steps.

A. Rapid Development Framework Run Environment Configuration

1) Install and configure eclipse,JDK,ant,weblogic and Rapid development framework, run build in eclipse of ant to initialization.

2) Run weblogic `http://localhost:7001/console`.

3) Install and deploy web application in weblogic.

B. Rapid Development Framework Run Flow

4) Web application search deploy describe symbol(web.xml) as entrance to run program, and loading servlet,filter,monitor and session.

5) Client post visit path and send request to server, server search client visit servlet in web.xml and then find app servlet.

6) Find configuration file .application that has the same name with app servlet.

7) Loading .application's configuration such as engine configuration,visit configuration,page configuration,etc.

8) According to client post page's name to search page file in .application.

9) Through page file search java,html file.

C. Web Application Deploy Configuration

```

<filter>
<filter-name>redirect</filter-name>
<filter-class>org.apache.tapestry.RedirectFilter
</filter-class>
</filter>
<filter-mapping>
<filter-name>redirect</filter-name>
<url-pattern>/</url-pattern>
</filter-mapping>
<servlet>
<servlet-name>quickstart</servlet-name>
<servlet-class>
org.apache.tapestry.ApplicationServlet</servlet-class>
<load-on-startup>0</load-on-startup>
</servlet>
<servlet-mapping>
<servlet-name>quickstart</servlet-name>
<url-pattern>/app</url-pattern>
</servlet-mapping>
<session-config>
<session-timeout>1800</session-timeout>
</session-config>

```

D. System Whole Situation Configuration

```

<project
name="SupplyChainKnowledge"
productmode="false"
uploadpath="upload"
version="1.5.1"
model="HY"
logintype="WEB"/>
<tuxconn type="WTC"/>
<subsys default="sale">
<DynamicKnowledge
addr="http://191.118.0.1:7000"/>
<StaticKnowledgeaddr="http://191.118.0.2:7000"/>
/>
</subsys>
<database default="ehy">
<sdcrml type="SCK" SCK="RDF"/>
<sdcenl type="SCK" SCK="RDF"/>
</database>

```

Third, write source code of page template (.html), page specification (.page),page class(.java) and some component class related to the business logic in supply chain knowledge management system. The following are several instances illustrate the writing method:

```

<- system page template example ->
<html xmlns="http://www.w3.org/1999/xhtml">
<head SCKid="@ RDF:Head">
<meta http-equiv="Content-Type"
content="text/html; charset=gbk"/>
<title> SupplyChainKnowledge </title>
<link href="/component/styles/styles_all.css"
rel="stylesheet" type="text/css" media="screen"/>
</head>
<body SCKid="@Body">
<form SCKid="@Form">
...

```

```

</form>
</body>
</html>
<- system page specification example ->
<page-specification class="com.linkage.quickstart.
SupplyChainKnowledge ">
<property-specification name="condition" type="com.
framework.IData"/>
</page-specification>
<- system page class example ->
public abstract class SupplyChainKnowledge
extends AppPage {
/* define supply knowledge object in page template
*/
public abstract void setInfos(IDataset infos);
public void querySupplyChainKnowledge (IReq
cycle) throws Exception {
/* supply knowledge query */
PageData pd = getPageData();
IDataset infos = ...(supply chain knowledge);
setInfos(infos);
}
}

```

V. SUGGESTIONS

The application of knowledge management in supply chain can solve some supply chain's problems including sharing of information flow and workflow and high efficiency operation. But the supply chain knowledge management involves many different enterprises and individual, there are some difficulties in organization, management and technology. Rapid development framework lack of innovation and decision analysis models, so supply chain knowledge management system have not innovation and decision analysis, this is future effort direction.

ACKNOWLEDGMENT

Thanks are due to Dr.Wang Jin for assistance with the Oriented middleware programming and to Dr.He Ying for valuable discussion.

REFERENCES

- [1] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. SIGMOD'93.
- [2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. VLDB'94.
- [3] H. Mannila, H. Toivonen, and A. I. Verkamo. Efficient algorithms for discovering association rules. KDD'94.
- [4] A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining association rules in large databases. VLDB'95.
- [5] J. S. Park, M. S. Chen, and P. S. Yu. An effective hash-based algorithm for mining association rules. SIGMOD'95.
- [6] H. Toivonen. Sampling large databases for association rules. VLDB'96.
- [7] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket analysis. SIGMOD'97.
- [8] S. Sarawagi, S. Thomas, and R. Agrawal. Integrating association rule mining with relational database systems: Alternatives and implications. SIGMOD'98.
- [9] J. Han, J. Wang, Y. Lu, and P. Tzvetkov. Mining Top-K Frequent Closed Patterns without Minimum Support. ICDM'02.
- [10] R. Agarwal, C. Aggarwal, and V. V. V. Prasad. A tree projection algorithm for generation of frequent itemsets. J. Parallel and Distributed Computing:02.
- [11] G. Grahne and J. Zhu, Efficiently Using Prefix-Trees in Mining Frequent Itemsets, Proc. ICDM'03 Int. Workshop on Frequent Itemset Mining Implementations (FIMI'03), Melbourne, FL, Nov. 2003
- [12] H. Liu, J. Han, D. Xin, and Z. Shao, Mining Interesting Patterns from Very High Dimensional Data: A Top-Down Row Enumeration approach,SDM'06.