

# The design and implementation of the analysis system of data-providing behavior based on data mining

WangXiaoGuo

School of Electronics Engineering  
Tongji University  
Shanghai, China  
xiaoguowang@tongji.edu.cn

Sun Chuan

School of Electronics Engineering  
Tongji University  
Shanghai, China  
082349@tongji.edu.cn

Zhang DanDan

School of Electronics Engineering  
Tongji University  
Shanghai, China  
092382@tongji.edu.cn

**Abstract**—In this paper, we conduct the analysis of the behavior of data providers in a given period of time, utilizing the theory of data warehouse, data mining, version control and data comparison. We classify data providers via decision tree, then designing and implementing the whole analysis system of data-providing behavior. The system can automatically receive and integrate data from various sources according to the classification of data providers, which provides an effective way for the prediction system of the set of undergraduate professional to receive and manage incoming data better.

**Keywords**- data warehouse; version control; data comparison; data mining; behavior analysis

## I. INTRODUCTION

In the prediction system of the set of undergraduate professional, data comes from various sources. Analyzing and processing these data has been a key task for system integration. Data providers in the prediction system come from different departments and industries, with their data provided different in terms of providing frequency, data quality, and the change in data structure. The system conducts the analysis of data-providing behavior and the classification of data providers in order to guarantee the reception and processing of system data, together with future work of the project.

## II. REQUIREMENTS

This paper designs the analysis system of data-providing behavior with functions listed below:

- 1) *Data receiving and transformation*: receive various kinds of data from data providers, storing them by category and conducting version control and XML format transformation.
- 2) *Establishing data warehouse*: design data model, conduct the data extraction and transformation based on data model, and establish the data warehouse.
- 3) *Data comparison*: conduct both structural and content comparison for the transformed XML data, then calculating data similarity.
- 4) *Analysis of data-providing behavior*: analyze and visualize the data-providing behavior of data providers in a given period of time.

- 5) *Classification*: classify data providers according to the features of their data-providing using classification algorithms in data mining.

## III. SYSTEM DESIGN

### A. Design of System Framework

The system uses the framework shown in Fig. 1, separated into presentation layer, data pre-processing layer, data interchange layer, data storage layer and data warehouse.

- 1) *Presentation layer*: uses web browser to provide systematic user interface, logging in, data upload, verification and error warning for users. The components of front-end design includes: About controls and JQuery framework, which improves user experience and separates the code of front-end design from that of back-end logic.

- 2) *Data pre-processing layer*: this layer mainly provides data pre-processing for various data formats, receiving formats such as Excel files, DMP files, DBF files, CSV files, providing different pre-processing methods for different file formats and transforming them into XML documents.

- 3) *Data interchange layer*: provide universal interface to import the pre-processed data into system database, which facilitates the import of files in other formats by adding corresponding data-transformation interface and vastly improves system extensibility.

- 4) *Data storage layer and data warehouse*: store the data, and load them into data warehouse through data extraction and transformation.

### B. Work Flow

The logic flow of the system is shown in Fig. 2. Authenticated users can access the system and upload data. The system will automatically do the pre-processing work like version control and XML transformation to the data uploaded, meanwhile analyzing and visualizing the data-providing behavior.

By the time it receives data, the system will classify data providers according to providing frequency and data quality using classification algorithms like decision tree. After classification, the system will adopt different data receiving strategy on users in different categories, which is shown in the dashed line box in Fig. 2.

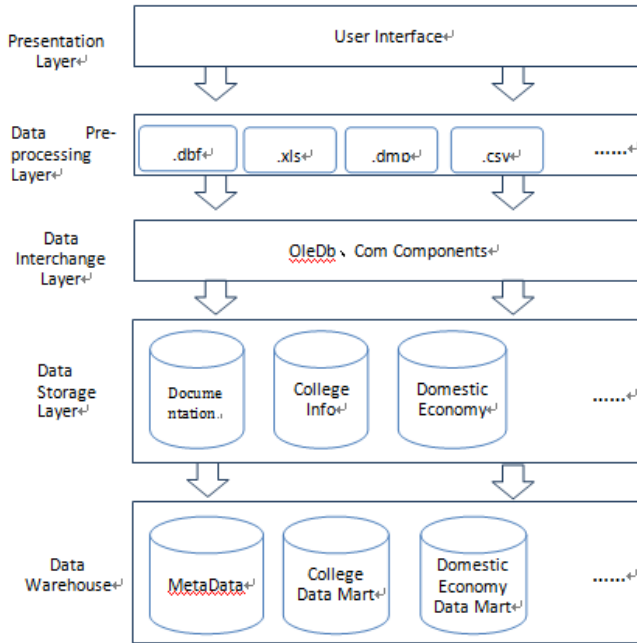


Fig. 1. Data Frame Structure of the system

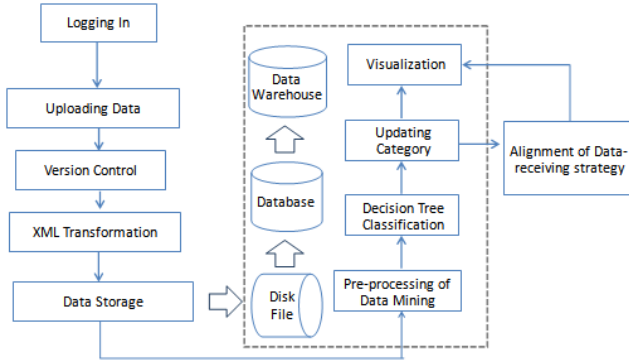


Fig. 2. System work flow

#### IV. SYSTEM IMPLEMENTATION

##### A. Data receiving and transformation

###### 1) Data receiving

First of all, the system renames the data being uploaded to the server by appending time stamp onto its filename, in order to distinguish different versions of the same file, as is shown in Fig. 3.

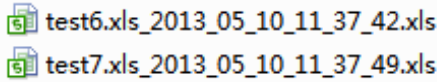


Fig. 3. Storage of files with the same name

Besides, time stamp is used to manage data versions. Version info like filename, uploading time and provider info are imported into the database, with only database query needed when managing data.

###### 2) XML transformation

After data receiving and version control, the system needs to transform data into XML format for further analysis and comparison. The system allows the upload of formats like dbf, xls, xlsx and csv files, and the data transforming module stores these data into DataSets first, then importing the

DataSets into XML documents according to construction standards of XML documentation[1]. The standard structure of XML data is shown in Fig. 4.

##### B. Establishing Data Warehouse

In the whole system project, data are separated into 3 categories: the data of college personnel training, the data of domestic economy and the data of employment statistics. The first part falls into 4 further themes: college info, professor info, student info and major info. The system designs the multi-dimensional data model according to corresponding details of each theme. The star schema is used here, exemplified by the theme of major in Fig. 5.

```
<?xml version="1.0" standalone="yes"?>
<NewDataSet>
  <INFO>
    <YEAR>1996</YEAR>
    <PRICE_OF_PRIMARY_INDUSTRY>14015.4</PRICE_OF_PRIMARY_INDUSTRY>
    <STATIC_PRICE_OF_PRIMARY_INDUSTRY_1998>14185.4526956694</STATIC_PRICE_OF_PRIMARY_INDUSTRY_1998>
    <STATIC_PRICE_OF_SECOND_INDUSTRY>33835</STATIC_PRICE_OF_SECOND_INDUSTRY>
    <STATIC_PRICE_OF_SECOND_INDUSTRY_1997>34245.529343292</STATIC_PRICE_OF_SECOND_INDUSTRY_1997>
    <PRICE_OF_THIRD_INDUSTRY>23326.2</PRICE_OF_THIRD_INDUSTRY>
    <STATIC_PRICE_OF_THIRD_INDUSTRY_1997>23609.2231880448</STATIC_PRICE_OF_THIRD_INDUSTRY_1997>
  </INFO>
  <INFO>
    <YEAR>1997</YEAR>
    <PRICE_OF_PRIMARY_INDUSTRY>14441.9</PRICE_OF_PRIMARY_INDUSTRY>
    <STATIC_PRICE_OF_PRIMARY_INDUSTRY_1998>14441.9</STATIC_PRICE_OF_PRIMARY_INDUSTRY_1998>
    <PRICE_OF_PRIMARY_INDUSTRY>37543</PRICE_OF_PRIMARY_INDUSTRY>
    <STATIC_PRICE_OF_SECOND_INDUSTRY>37543</STATIC_PRICE_OF_SECOND_INDUSTRY>
    <PRICE_OF_THIRD_INDUSTRY>26988.1</PRICE_OF_THIRD_INDUSTRY>
    <STATIC_PRICE_OF_THIRD_INDUSTRY_1997>26988.1</STATIC_PRICE_OF_THIRD_INDUSTRY_1997>
    <GROWTH_RATE_PRIMARY_INDUSTRY>1.80781896660152</GROWTH_RATE_PRIMARY_INDUSTRY>
    <GROWTH_RATE_SECOND_INDUSTRY>9.62890841503066</GROWTH_RATE_SECOND_INDUSTRY>
    <GROWTH_RATE_THIRD_INDUSTRY>14.3116814350171</GROWTH_RATE_THIRD_INDUSTRY>
  </INFO>
</NewDataSet>
```

Fig. 4. Standard structure of XML data

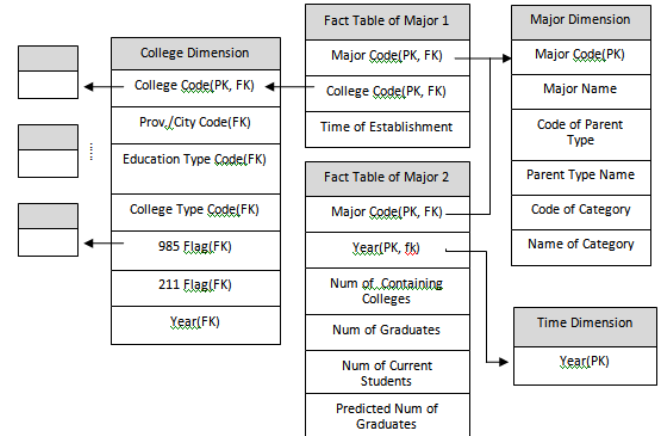


Fig. 5. Multi-dimensional data model of the major theme

##### C. Data Comparison

Data can be processed with comparison after being transformed into XML format.

###### 1) Data selection

Data of the same name are distinguished through version. In other words, same-named data are allowed in the database, but they are not duplicate as they carry different info.

When a new data is submitted, the data comparison module searches for the same-named data in database first. If found, it goes on to searches for the nearest version for comparison, otherwise it reckons the data uploaded to be a new type without any similar data files, and initializes its similarity to zero.

## 2) Structural and content comparison

The system conducts two kinds of comparison: structural and content comparison. For one single data provider, structural comparison falls into two situations: structure changed and structure not changed.

Deciding whether the structure has changed is a must for content comparison. If changed, the system regards the changed one as a new version, which is used for data comparison hereafter.

## 3) Structural comparison

In structural comparison, the system first extracts the initial symbol of each attribute in XML files, namely the Name value of the data whose NodeType attribute is Element in the XmlTextReader class, as is shown in Fig. 6. These initial symbols equal to the column names in the table, thus the structure comparison is done via the comparison of initial symbols.

```
public List<string> FilterByNodeType(XmlNodeType[] xmlNodeType)
{
    List<string> BL = new List<string>();
    this.xmlTextRd = new XmlTextReader(this.xmlPath);
    try
    {
        while (xmlTextRd.Read())
        {
            for (int i = 0; i < xmlNodeType.Length; i++)
            {
                if (xmlTextRd.NodeType == xmlNodeType[i])
                {
                    if (xmlTextRd.Name != "xml" && xmlTextRd.Name != "NewDataSet" && xmlTextRd.Name != "INFO")
                    {
                        if (!BL.Contains(xmlTextRd.Name)) BL.Add(xmlTextRd.Name);
                    }
                }
            }
        }
        BL.Sort();
        return BL;
    }
}
```

Fig. 6. Filtering of XML node attribute

## 4) Content comparison

In content comparison, it is often hard to get the row similarity of data, namely to get which row is added or deleted when the row number changes. The content comparison logic here reckons that when over half of the data in a row changes, the row is not the same row as before, and the row number changes. The logic will then count changes in row number and the value of different cells, calculating their similarity respectively, as is shown in Fig. 7.

## D. Behavior Analysis

The behavior analysis module is separated into user behavior analysis and overall analysis, providing the data foundation for subsequent data mining and classification[2].

```
//load xml to RAM, store columnName and cell value
string path = "E:\\uploads";
xmlPath = path + filename;
xRead = new Xml(xmlPath);
xRead.LoadXml();
List<Dictionary<string, string>> cell = xRead.xmlTable;
//match available only if match value > half of the num in row
if (maxsim >= cell[j].Count / 2)
{
    f[ak] = true;
    count += maxsim;
}
```

Fig. 7. Content comparison

## 1) User behavior analysis

In the sub-module of user behavior analysis, the system logs, analyzes and visualizes the behavior patterns of data

providers such as upload time, upload frequency, average similarity of provided data, etc. As is shown in Fig. 8, upload frequency is on the left with average similarity on the right.

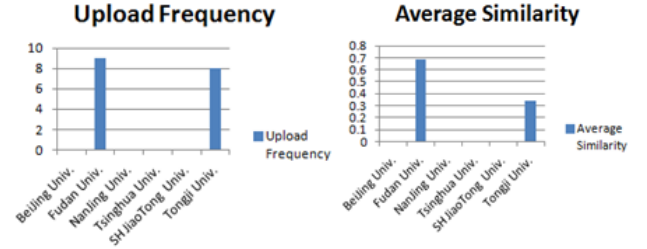


Fig. 8. Visualization of overall user comparison

## 2) Overall analysis

The sub-module of overall analysis adds the time dimension, offering the logging and visualization of data-providing behavior in different periods of time (e.g. by year, by quarter, by month, etc.), which facilitates the model training of the subsequent data mining using data under different time slices.

## E. Data mining

The system uses classification algorithms in data mining to classify data providers according to their data-providing features. Standards used include two aspects: upload frequency and data quality, which fall into several detailed features to train the model and apply the algorithms.

## 1) Pre-processing of data mining

Before classification, the system needs to extract data from data warehouse and normalize it to make its data structure fit for classification. Fig. 9 shows the feature design, which is composed of frequency and similarity, with each one separated into several columns depicting upload frequency, data quality and whether the upload pattern has changed recently. The value of each column is extracted via group selection and aggregation.

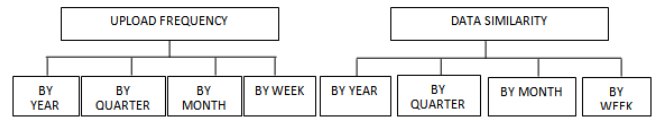


Fig. 9. Feature design

## 2) Decision tree training

The system's training set comes from history system data providers. Since continuous features are included, the system uses C4.5 algorithm to train models. Eight categories of the decision tree are defined according to frequency (high/ low), data quality (high/ low), and whether has changed recently (yes/ no), as is shown in table1.

After pre-processing, we got an analog data set that suits real data-providing patterns based on authentic user behavior. We use 2/3 of them as the training set, and ensure that the number of each category in training set are roughly equal in order to improve training accuracy and comprehensiveness.

TABLE I. CATEGORIES OF CLASSIFICATION

Upload Frequency	Data Quality	Whether Has Changed Recently	Label
HIGH	HIGH	NO	CATEGORY1
HIGH	HIGH	YES	CATEGORY2
HIGH	LOW	NO	CATEGORY3
HIGH	LOW	YES	CATEGORY4
LOW	HIGH	NO	CATEGORY5
LOW	HIGH	YES	CATEGORY6
LOW	LOW	NO	CATEGORY7
LOW	LOW	YES	CATEGORY8

### 3) Classification and subsequent work

After model training, the system classifies data providers with certain time interval, storing their category labels in the college dimension and user tables[3]. The time interval of classification depends on user's upload frequency. The system will update users' category more frequently for those who uploads more frequently, and the users with better data quality will be given higher data receiving priority.

In the experiment conducted in this paper, we use the remaining 1/3 of the analog data set as the testing set, and ensure that every category in it is close in quantity[4]. Finally the testing set quantity of the 8 category labels and its corresponding accuracy of classification is shown in Fig. 10. Besides, the system will dynamically update the two data receiving strategies mentioned above according to whether a user's data-providing behavior has changed recently.

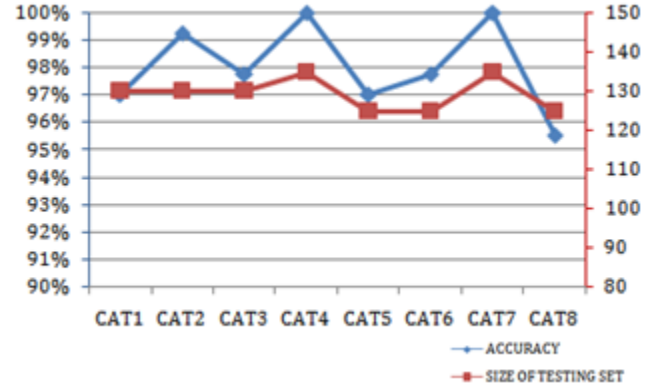


Fig. 10. Accuracy and size of testing set for each category

## V. CONCLUSIONS

This paper conducts the analysis of data-providing behavior via XML format transformation, XML storage, structural and content comparisons of received data, and then designs and implements the analysis system of data-providing behavior based on data mining. The system provides an effective solution for better storage and management of received data, setting up a solid foundation for the subsequent work of the prediction system of the set of undergraduate professional.

## REFERENCES

- [1] I. Taranov, I. Shcheklein, A. Kalinin, L. Novak, S. Kuznetsov, R. Pastukhov, et al. "Sedna: Native XML Database Management System," in SIGMOD'10, June 6-11, 2010, pp.1037-1045.
- [2] Yu Ye,Guowen Wu,Xin Luo. Research on Interest Model of User Behavior[A]. Proceedings of 2011 International Conference on Computer Science and Information Technology(ICCSIT 2011)[C]. 2011.
- [3] Y. Ma and S. Banerjee. A smart pre-classifier to reduce power consumption of tcams for multi-dimensional packet classification. In Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication, pages 335-346. ACM, 2012.
- [4] Clonal Selection Classification Algorithm for High-Dimensional Data[A]. Final Program and Book of Abstracts of the 2010 International Conference on Life System Modeling and Simulation & 2010 International Conference on Intelligent Computing for Sustainable Energy and Environment[C]. 2010.