A Novel Reconfigurable Character based Token Set Pruner (RCBTSP) for Heterogeneous Environment

Reeta Budhani[1], Dalima Parwani[2], Meenu Tahilyani[3], Subuhi Kashif Ansari[4]

Asst. Professor Computer Science, Sant Hirdaram Girls College[1,2,3,4]
pamnani.pinky@yahoo.com[1]
dalimaparwani@gmail.com[2]
meenu_mgh@yahoo.co.in[3]
subuhiwasim_786@yahoo.co.in[4]

Abstract

Extracting useful insights from large and detailed collections of data is called data mining. With the increased possibilities in modern society for companies and institutions to gather data cheaply and efficiently, this subject has become of increasing importance. This interest has inspired a rapidly maturing research field with developments both on a theoretical, as well as on a practical level with the availability of a range of commercial tools. In this paper we proposed a novel Reconfigurable Character based Token Set Pruner (RCBTSP) for heterogeneous environment. Our algorithm contains six phases 1) Authentication 2)Reading Database 3) Define the reconfigurable character 4) Define the minimum support 5) Find the token based on the minimum support 6) Prune phase. Finally our algorithm shows better performance showing the simulation result. Finally, through the simulation our proposed methods were shown to deliver excellent performance in terms of efficiency, accuracy and applicability under various system conditions.

Keywords

Data Mining, Reconfigurable pattern, RCBTSP, Token

1. Introduction

The primary ingredient of any Data Mining [1][2][3][4][5] exercise is the database. A database is an organized and typically large collection of detailed facts concerning some domain in the outside world. The aim of Data Mining is to examine this database for regularities that may lead to a better understanding of the domain described by the database. In Data Mining we generally assume that the database consists of a collection of individuals. Depending on the domain, individuals can be anything from customers of a bank to molecular compounds or books in a library.

According to CRISP-DM, a consortium that attempted to standardize data mining process, data mining methodology is described in terms of a hierarchical process that includes four levels as shown in Figure 1.1. The first level is data mining phases, or processes of how to deploy data mining to solve business problems. Each phase consists of several generic tasks or, in other words, all possible data mining situations. The next level contains specialized tasks or actions to be taken in order to carry out in certain situations. To make it unambiguous, the generic tasks of the second phase have to be enumerated in greater details. The questions of how, when, where and by whom have to be answered in order to develop a detailed execution plan. Finally, the fourth level, process instances, is a record of the actions, decisions and results of an actual data mining engagement or, in short, the final output of each phase.

The top level, data mining process, consists of six phases which are business understanding, data understanding, data preparation, modeling, evaluation and deployment.

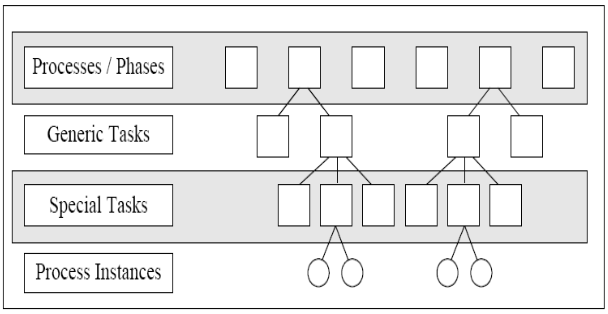


Figure 1 Four Level Breakdown of Data Mining

We provide here an overview of executing data mining services. The rest of this paper is arranged as follows: Section 2 introduces Data mining and knowledge and discovery; Section 3 describes about classification and prediction; Section 4 shows the evolution and recent scenario; Section 5 describes the proposed work. Section 6 describes Conclusion and outlook.

2. Data Mining and knowledge Discovery

Mining frequent patterns in large transactional database is a highly researched area in the field of data mining. The different existing frequent pattern discovery algorithms suffer from the same problem. That is, they are all inherently dependent on the amount of main memory available. The most important task for the management personnel is to guarantee the level of quality service the customer requires, and for that they have to master the above tasks. Terplan suggests a choice of appropriate tools (though does not discuss them in detail) for the task, tools that should be useful for monitoring components, collecting data, and correlating information from many sources to help solve problems. Terplan points out that a large amount of data poses problems, and that advanced, automated methods are needed to help the personnel analyze the problems. More specifically, he mentions methods such as expert systems, neural networks, knowledge conservation, case-based reasoning, and data mining but omits more detailed discussion of the subject.

Generally, data mining (sometimes called knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information, which can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

With the explosion of the Internet the rate of accumulation is increasing exponentially. Methods to explore such data would stimulate research in many fields. Knowledge discovery and data mining (KDD) is the area of computer science that tries to generate an integrated approach to extracting valuable information from such data by combining ideas drawn from databases, machine learning, artificial intelligence, knowledge-based systems, information retrieval, statistics, pattern recognition, visualization, and parallel and distributed computing [6][7][8][9]. It has been defined as" The nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data". The goal is to discover and present knowledge in a form, which is easily comprehensible to humans.

An important aspect of KDD is that it is an ongoing iterative process. Following are the steps in the iterative KDD process along with an example from the real world. Data Acquisition: The raw data is collected usually as a byproduct of operation of the business.

1. Choose a Goal: Choose a question to ask the database. There are data on various welfare program participation. From the database, one could be interested in the following questions: What are the common patterns of participation in these welfare programs? What are the main variations?
2. Define the Problem: Define the problem statement so that the data can give the answer. What are the underlying patterns in the sequences of sets?
3. Define the Data: Define/view the data to answer the problem statement?
4. Data Cleaning and Integration: Remove noise and errors in the data and combine multiple data sources to build the data warehouse as defined in the" Define the Data" step.
    a. Cleaning: Clean all bad data (such as those with invalid data points)
    b. Integration: Merge the appropriate database on different welfare programs by recipients.
5. Data Selection and Transformation: Select the appropriate data and transform them as necessary for analysis.
    c. Selection: Separate out adults and children for separate analysis.
    d. Transform: Define participation for each program and transform the data accordingly.
2. For example, if someone received at least one TANF welfare check then code as T for that month.
    a. Transform: Build the transformed welfare program participation data into actual sequences.
6. Data Mining: The essential step where intelligent methods are applied to extract the information.
7. Patterns and Model Evaluation: Identify interesting and useful patterns.

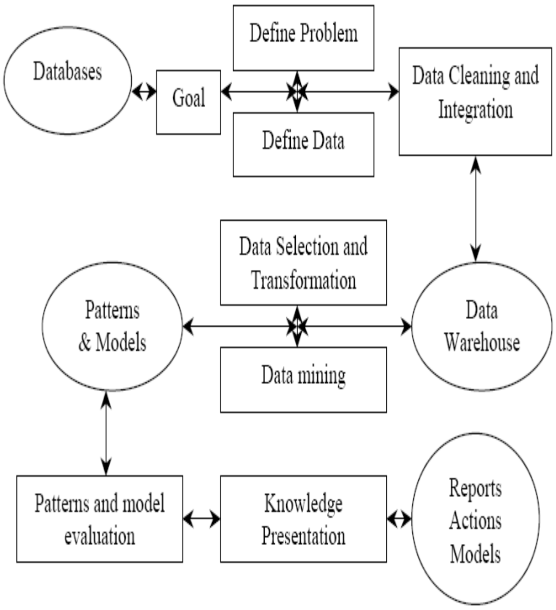


Figure 2: Complete KDD Process

3. Classification and Prediction

Classification is the process of finding models, also known as classifiers, or functions that map records into one of several discrete prescribed classes. It is mostly used for predictive purpose. Typically, the model construction begins with two types of data sets training and testing. The training data sets, with prescribed class labels, are fed into the model so that the model is able to find parameters or characters that distinguish one class from the other. This step is called learning process. Then, the testing data sets, without pre-classified labels, are fed into the model. The model will, ideally, automatically assign the precise class labels for those testing items. If the results of testing are unsatisfactory, then more training iterations are required. On the other hand, if the results are satisfactory, the model can be used to predict the classes of target items whose class labels are unknown.

4. Evolution and Recent Scenario

In 2006, C. Pautasso et al. [10] proposed about parallelism that could be effectively exploited in data mining workflows is data parallelism , where a large data set is split into smaller chunks, each chunk is processed in parallel, and the results of each processing are then combined to produce a single result.

In 2007,D Talia et al. [11] proposed about The Knowledge Grid which is another service-oriented system supporting distributed data mining workflow execution.LikeWeka4WS, it uses WSRF as enabling technology.UnlikeWeka4WS, which extends an already existing workflow system (the Weka Knowledge Flow), the Knowledge Grid defines its own workflow formalism and provides a set of services to support the workflow execution.

In 2009, Bin Cao et al. [12] proposed about Karma which is a tool that collects and manages provenance data. Karma has a modular architecture that supports multiple types of data sources for provenance data. Karma can listen to notifications on a messenger bus or receive messages synchronously and process the notifications to determine provenance information.

Workflow engines [13] are used for representing task dependencies and controlling execution. Generic Application Factory (GFac)[11] and Opal toolkit provide tools to wrap legacy scientific application codes as web services. The wrapper handles grid security and interaction with other grid services for file transfer and job submission. However the execution logic state for each application has to be managed individually and there is no easy way to abstract out, customize and reuse policies (e.g., resource selection) or code (e.g., provenance instrumentation) across implementations. This is very fruitful in terms of accuracy and efficiency in terms of traditional approaches.

In 2010, David Schumm et al.[14] proposed about process views which is technology independent and can be applied to any process language which can be represented by a process graph, such as the Business Process Modeling Notation (BPMN) and Event-driven Process Chains (EPC).

In 2010, Tobias Pontz et al. [15] proposed about an IT infrastructure based on service and grid computing technology. Additionally, a virtual value creation chain has been introduced to integrate virtual prototyping methodologies. The current contribution elaborates the importance of differentiating, defining and managing both value and knowledge flows in such a virtual value creation chain. Consequently, a service-oriented knowledge management system is envisaged by describing tasks of a knowledge manager and deducing a solution concept.

In 2010, Alexander Wöhrer et al. [16] proposed about rationale, theory, design and application of logical optimization of dataflows for data mining and integration processes. A dataflow model is defined and several optimization algorithms, namely dead elements elimination, process re-ordering, parallelization,and data by-passing are developed. The first research prototype of the framework has been implemented in the context of the ADMIRE Data Mining and Integration Process Designer for logical optimization of specifications expressed in the DISPEL language developed in the ADMIRE project.

In 2010, David Chiu et al. [17] proposed an approach to accelerate service processing in a Cloud setting. We have developed a cooperative scheme for caching data output from services for reuse. They propose an algorithm for scaling our cache system up during peak querying times, and back down to save costs. Using the Amazon EC2 public Cloud, a detailed evaluation of our system has been performed, considering speed up and elastic scalability in terms resource allocation and relaxation.

5. Proposed algorithm (RCBTSP)

In this paper we proposed a novel Reconfigurable Character based Token Set Pruner (RCBTSP) for heterogeneous environment. Our algorithm contains six phases 1) Authentication 2)Reading Database 3) Define the reconfigurable character 4) Define the minimum support 5) Find the token based on the minimum support 6) Prune phase.
Evaluating the execution times of the different steps
Assumptions-
Min-support-Minimum Support value defined by user.

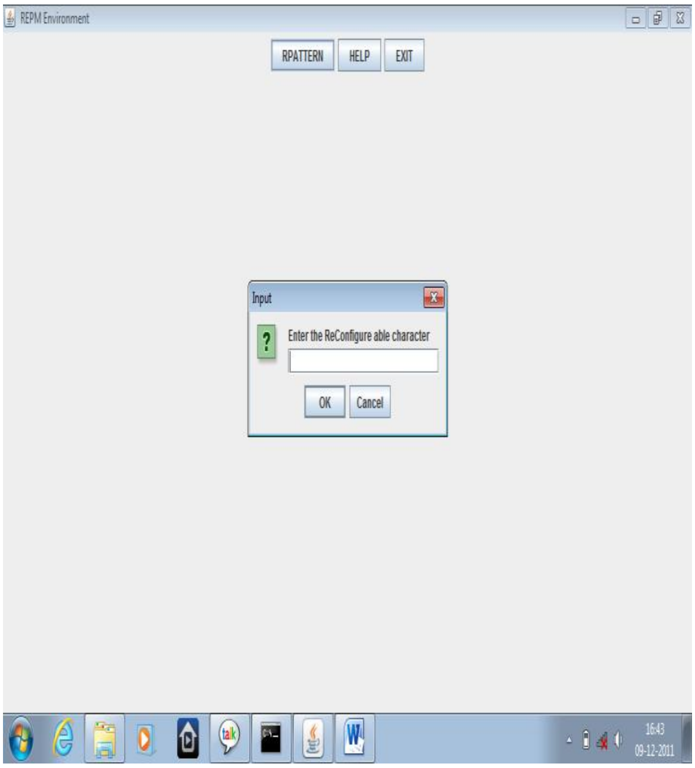
DS- Data Set
TEMP- Temporary Buff
Txt1 – Text File
MIN-SUPPORT – Minimum support value

```
Algorithm RCBTSP (DS)

Generate Mining rules.
STEP 1:  [AUTHENTICATION]
          IF (TRUE)
            PRINT ("WELCOME IN DATABASE");
          ELSE
          EXIT (0);

STEP 2: [READING DATABASE]
         [READ FROM THE FILE]
         If (object.read()!=-1)
         Txt1=(char)ob.write();

STEP 3: [ENTER THE CHARACTER BASED ON WHICH
         WE SEPARTE THE TOKENS]
         String s= ab.charAt(0).nextLine();
         TEMP(X)=Ys

STEP 3:  [DEFINE THE MINIMUM SUPPORT]
          MIN_SUPPORT = VALUE [DEFINE BY THE
         OWNER]

 TOKEN (DS, MIN_SUPPORT)
[MINIMUM SUPPORT IS INPUT IN CASE OF ADMIN
AND FIX IN CASE OF NORMAL USER]
         3.1: STORES THE VALUE IN TEMP
               FOR i = 0 to n-1
                 IF (FREQ >= MIN_SUPPORT)
                     TEMP = DS[i];
                  X →YS (MIN_SUPPORT)
                  T(X)→YS(FREQ(X))

        3.2:  [FIND THE SUPERSET]
              PEBUFFER = [ALL THE VALUE > =
         MIN_SUPPORT]
         T(X)→YS(Freq(X>=MIN_SUPPORT))

       4.3:  [DELETE THE REPEATED VALUE]
       4.4:   PRUNE (PEBUFFER)

STEP 5: PRUNE (PEBUFFER)
            FOR i = 1 to   n-1
              IF P (FREQUENT)
               PRBUF = APPEND. FREQUENT (DS);
              ELSE
            Delete repeated values
          PRINT (PRBUF);
```

Our algorithm contains six phases

1) Authentication: we check the authentication of the user only authorized user can use our data mining framework.
2) Reading Database: In second phase after authentication, we can read the data from the database.

3) Define the reconfigurable character: Then we enter the character by which we can generate tokens. This phase is shown in figure 3.



Enter the ReConfigure able character

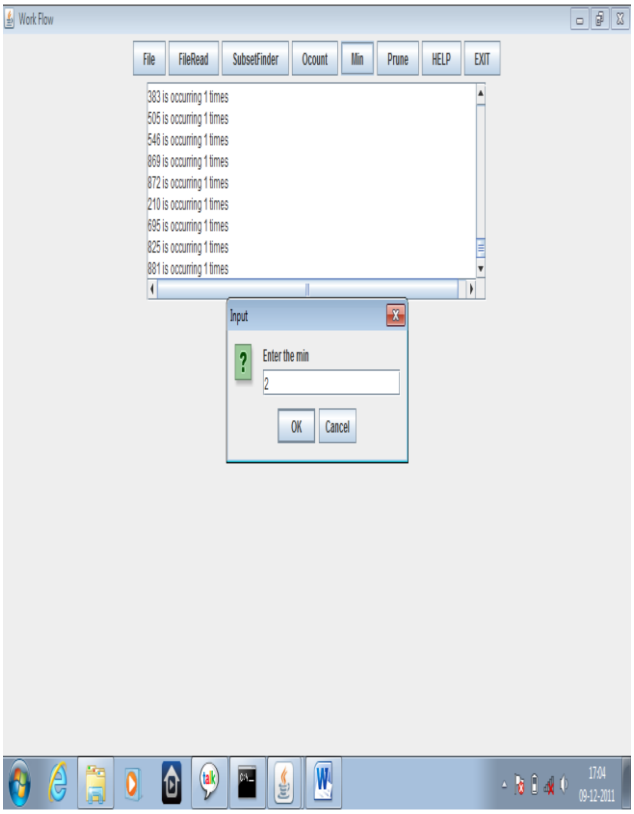
Figure 3 Output Screen for reconfigurable Character

4) Define the minimum support. Minimum support is the value, which defines those items which are frequently purchase from the shopping mall. This phase is shown in Figure 4.

5) Find the token based on the minimum support
We use a Subset Finder technique by which we can assign a minimum support value and according to that value we can compare the transactions of an employee. If count of sequences is greater than the minimum support we include those sequences in the pruning list, otherwise we skip those transactions from the list .Each such maximal Incremental Sequence represents an Incremental sequence pattern which is included for Prune.

6) Prune phase: We apply Prune on Possible Sequence with support greater than or equal to Min-sup in the database of employee transaction. First we read the data set from the database. The database is short in comparison of the original database, we only consider those item set which are frequent in the database means which having a support greater than or equal to minimum support. We apply the prune algorithm on the database until all the transactions are compared. Let the first element in the database is P. If P is not frequent in the current database, remove its subset and superset from the

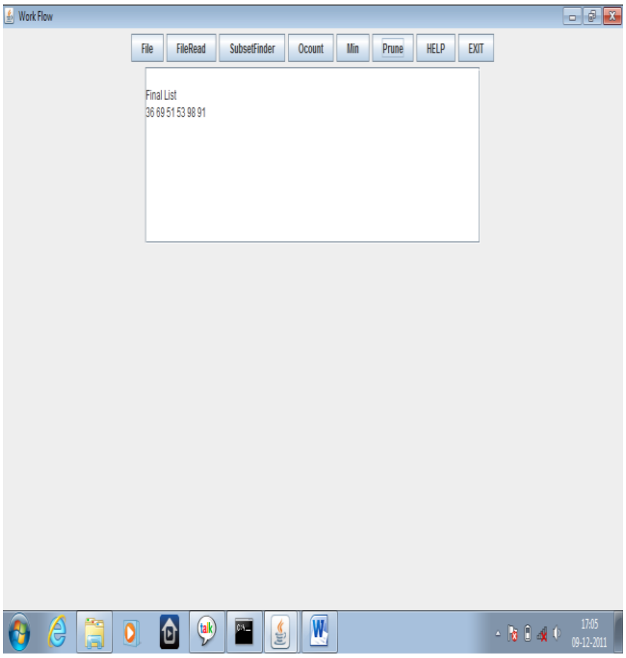
current database which is a text file. The same procedure is repeated until we finish all the values in the data set. This phase is shown in Figure5.



```
383 is occuring 1 times
505 is occuring 1 times
546 is occuring 1 times
869 is occuring 1 times
872 is occuring 1 times
210 is occuring 1 times
695 is occuring 1 times
825 is occuring 1 times
881 is occuring 1 times
```

Enter the min: 2

Figure 4 Output Screen for minimum support



```
Final List
36 69 51 53 98 91
```

Figure 5 Output Screen for final prune list

6. Conclusion and Outlook

In this paper we proposed a novel Reconfigurable Character based Token Set Pruner (RCBTSP) for heterogeneous environment. Our algorithm contains six phases 1) Authentication 2)Reading Database 3) Define the reconfigurable character 4) Define the minimum support 5) Find the token based on the minimum support 6) Prune phase. Finally our algorithm shows better performance showing the simulation result. Finally, through the simulation our proposed methods were shown to deliver excellent performance in terms of efficiency, accuracy and applicability under various system conditions. In future we show the comparison based on traditional algorithm.

7. References

[1] Dasu, T., Johnson, T. Exploratory Data Mining and Data Cleaning, Wiley, 2003.

[2] Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R., Advances in Knowledge Discovery and Data Mining, MIT Press, 1996.

[3] Hand, D., Mannila H., Smyth, P., Principles of Data Mining, MIT Press, 2001.

[4] Miyahara, T., Shoudai, T., Uchida, T., Takahashi, K. and Ueda, H. Discovery of frequent tree structured patterns in semi structured web documents. In Proceedings PAKDD2001, 47-52, 2001.

[5] Wang, K., Liu, H. Discovering structural association of semi structured data, IEEE Trans. Knowledge and Data Engineering (TKDE2000), 12(3):353-371, 2000.

[6] M. Cannataro, "Clusters and grids for distributed and parallel knowledge discovery", International conference on high performance computing and networking, vol. 2000.

[7] D. D. Roure, N. R. Jennings, and N.l R. Shadbolt, "The Semantic Grid: Past, Present and Future", The Proceeding of IEEE, 2005.

[8] Cannataro M. and D. Talia, "KNOWLEDGE GRID An Architecture for Distributed Knowledge Discovery", CACM,Vol. 46, No. 1, pp. 89-93, January 2003.

[9] W.-S. Soh and H. Kim, "QoS Provisioning in Cellular Networks Based on Mobility Prediction Techniques," IEEE Comm. Magazine, vol. 41, no. 1, pp. 86-92, Jan. 2003.

[10] C. Pautasso, G. Alonso, Parallel Computing Patterns for Grid Workflows, Workshop on Workflows in Support of Large-Scale Science, 2006.

[11] A. Congiusta, D. Talia, P. Trunfio. Distributed data mining services leveraging WSRF. Future Generation Computer Systems, 23(1):34-41, 2007.

[12] Bin Cao, Beth Plale, Girish Subramanian, Ed Robertson, Yogesh Simmhan, "Provenance Information Model of Karma Version

3,"Services, IEEE Congress on, pp. 348-351, 2009 Congress on Services - I, 2009.

[13] D. Leake and K.-M. Joseph, Towards Case-Based Support for e-Science Workflow Generation by Mining Provenance, in Proc of the 9th European conference on Advances in Case-Based Reasoning. 2008.

[14] David Schumm, Tobias Anstett, Frank Leymann, Daniel Schleicher, 14th IEEE International Enterprise Distributed Object Computing Conference Workshops,IEEE 2010.

[15] Tobias Pontz, Manfred Grauer, Daniel Metz, Sachin Karadgi, 3rd International Conference on Information Management, Innovation Management and Industrial Engineering,2010,IEEE.

[16] Alexander Wöhrer, Eduard Mehofer and Peter Brezany, 2010 Sixth IEEE International Conference on e–Science Workshops.

[17] David Chiu, Apeksha Shetty and Gagan Agrawal, SC10 November 2010, New Orleans, Louisiana,IEEE.