Minable Data Warehouse

David Morgan[1], Jai W. Kang[1], and James M. Kang[2]

[1] College of Computing and Information Sciences, Rochester Institute of Technology
Rochester, NY, USA
davidmorgan@mail.rit.edu, jai.kang@rit.edu
[2] Department of Computer Science, University of Minnesota, Minneapolis, MN, USA
jkang@cs.umn.edu

Abstract. Data warehouses have been widely used in various capacities such as large corporations or public institutions. These systems contain large and rich datasets that are often used by several data mining techniques to discover interesting patterns. However, before data mining techniques can be applied to data warehouses, arduous and convoluted preprocessing techniques must be completed. Thus, we propose a minable data warehouse that integrates the preprocessing stage in a data mining technique within the cleansing and transformation process in a data warehouse. This framework will allow data mining techniques to be computed without any additional preprocessing steps. We present our proposed framework using a synthetically generated dataset and a classical data mining technique called Apriori to discover association rules within instant messaging datasets.

Keywords: Data warehouse, Data mart, Data mining, Apriori, Association rule mining.

1 Introduction

Motivation. Many corporations all over the world maintain their valuable historical datasets using data warehouses that collect an abundant amount of information. Due to the integrated and cleansed information stored in data warehouses, these systems have been one of the major sources for data mining techniques. A popular data mining technique that often uses these types of datasets is association rules. Association rule mining (ARM) identifies sets of object types that may co-occur together more often than other item types. Thus, it is crucial that the information stored in a data mart must be readily accessible to these data mining methods to determine specific patterns quickly and efficiently.

Problem Description. Given one or more data marts in a data warehouse, the goal is to have data readily available for a data mining algorithm. The main objective is to reduce the amount of effort to prepare the dataset for data mining methods. For example, in ARM of market basket datasets, a crucial criterion of the input to ARM is to have transactional information containing a unique transaction id and the various item types for each transaction.

Challenges. Preparing the data from a data mart quickly for a data mining technique is extremely challenging for several reasons. First, information stored in a data mart for

J. Filipe and J. Cordeiro (Eds.): ICEIS 2009, LNBIP 24, pp. 125–136, 2009.

a certain business unit is often prepared based on their own needs and is not suitable for direct use of data mining techniques (e.g. association rule mining). Second, the preprocessing time for data mining methods is often convoluted and may remove pertinent information that may be crucial for the resulting pattern set. Finally, as the datasets in a data mart get updated, the preprocessing step needs to be re-computed before the data mining technique can execute.

Related Work. Several researchers and analysts have claimed that a data warehouse may be an excellent data source for data mining methods and have postulated that these two can create a "symbiotic relationship" [5,8]. They have not explored how a data warehouse may be used for data mining specifically. For example, ARM has been used in data marts to refine the set of patterns but has not addressed the preprocessing problem that occurs between each system [7]. Likewise, materialized views in data warehouses have been used for data mining methods to allow for repetition of methods [2] but have not examined the preprocessing steps to produce the end result. Modifying data warehouses to accomodate data mining methods has been explored by maintaining a single attribute using bit mapped indexes [10] but has not been shown directly to handle multiple attributes for association rule mining. Thus, to the best of the authors' knowledge, no previous works have explored the integration of data mining and data warehouse by reducing the amount of preprocessing for multi-attributes.

Contributions. In this paper, we propose a novel framework that allows data mining techniques to access a data mart without any preprocessing steps in between. In general, data that are stored in a data mart must go through an entire cleansing and transformation process to prepare the data before they are stored in the system. Likewise, data mining techniques must also perform some form of cleansing or preprocessing before the technique can be executed. Thus, we propose to re-construct the traditional cleansing and transformation process in data warehouses to align with the requirements in data mining techniques. This would allow an efficient streamline between data warehouses and data mining techniques for quick and efficient pattern sets. We evaluate this framework using a synthetic dataset of instant messaging and the classical association rule mining algorithm called Apriori. In summary, this paper makes the following contributions:

1. We propose a novel framework called a *Minable Data Mart* that will allow data mining techniques to perform on a data mart without the use of any preprocessing steps.
2. We evaluate our framework using a case study that utilizes the Apriori method to discover associations in a transactional dataset.
3. We present our system with several screen shots to illustrate the proposed framework.

Organization. The rest of the paper is organized as follows. Section 2 presents the general basic concepts and an example used throughout this paper. Three different frameworks are given, specifically *Data Mining without Data Warehouses*, *Data Mining with Data Warehouses*, and *Mineable Data Warehouse* in Sections 3, 4, and 5 respectively. A demonstration of the mineable data warehouse is given in Section 6. Finally, Section 7 concludes the paper and discusses future work.

2 Basic Concepts

This section presents the basic concepts that are used throughout this paper. First, we introduce several basic concepts in data warehouses. Second, we highlight one of the classical data mining approaches we use as part of our examples and within the demonstration. Finally, we give an example using a synthetic dataset of instant messaging with data warehouses and data mining.

2.1 Data Warehouses

In general, a data warehouse is an integrated system that is designed to facilitate user analysis. This system provides a single, clean, and consistent source of historical data for many types of decision making. Also, it provides strategic information based on an enterprise-wide scale.

Data warehouses can be developed as either a top-down [5] or a bottom-up [9] approach. In a top-down approach, a data warehouse is initially defined as the entire organization where each faction may have a different operation (i.e., Enterprise Data Warehouse). Data Marts are extracted from the top-down creation of the data warehouse. However, in a bottom-up approach, one business unit may create their own data mart (e.g. sales, marketing, etc.) first. Then, the next business unit creates their own data mart which conforms to the dimensions in the previously created data mart. This process continuous for all business units in a corporation until the final data warehouse is completed.

Data Warehousing has traditionally been used for on-line analytical processing (OL AP) systems [6,8]. Data that are stored in a data warehouse or data mart must go through an entire cleansing and transformation process to be stored and ensure that the data is clean [5,9]. Data miners must also go through a similar cleansing and preparing process to run their mining algorithms on the same source data. It has been said that a data warehouse would provide an excellent base for data miners to use as source data because of its cleaned nature [9].

A data mart is a dimensional model that represents a single business process in an organization. Its dimensional data must be atomic and should also be built using conformed dimensions in relation to other existing or future data marts. Generally speaking, a data mart is developed using a star schema with a central fact table connected to outlying dimensions [9]. These dimensions will allow for “drilling down” type queries to determine the underlying facts for particular dimensional combinations. Data marts are generally used for OLAP analysis. This analysis helps to inform decision makers about making key business decisions. Multiple conformed dimensions can also be shared with other data marts to build a data warehouse.

An essential step to create a data warehouse is the ETL (Extract, Transform, and Load) phase that has three main steps [5,9]. First, the data is extracted from operational or external data stores. Then, the data is transformed by performing cleansing, aggregation, summarization, integration, and coding transformations. Finally, the data are loaded into the data warehouse. The main goal of this entire process is to obtain clean, consistent, integrated, and possible summarized data.

2.2 Association Rule Mining

One of the classical techniques in data mining is the Apriori algorithm used for association rule learning [1]. In general, association rule learning attempts to discover rules or sets various types that tend to occur together more often than other variables. For example, one of the popular applications for association rule mining is on market basket datasets. In general, market basket datasets may contain itemsets of multiple different types (e.g., various products bought at a grocery). The goal of association rule mining is to discover rules where a certain sets of items are often bought together, i.e., when a certain item type is seen, another item will also be seen. There are an extensive number of algorithms that discover association rules (e.g., see [4,3]), but one of the classical and fundamental techniques is the Apriori algorithm. One of the common problems used for the Apriori algorithm is to determine which sets of items types occur together more often than other combinations of item types. Identifying which items types that will be examined is based on the candidate generation method. Within the Apriori technique, there are two basic interest measures that are often used called *support* and *confidence*.

Support is the general measure that determines how often an itemset occurs within the transaction dataset. For example, suppose we have three transactions having a set of items: $T1(A, B, C), T2(B, C), T3(A, B)$. Out of these three transactions, there are three different item types, A, B, and C. A support threshold may be applied after each size of an itemset is generated, where any itemset that has fewer than the support threshold is not considered the next candidate generation. Suppose the support threshold was 1. First, there are three singletons (i.e. one item type) datasets. Based on the notation of itemset:support, the singletons have the following: $A : 2$, $B : 3$, $C : 2$. All singletons satisfy the support threshold and they can all be used for candidate generation. The next set of candidate itemsets is of size two: $(A, B), (A, C), (B, C)$, where order is irrelevant. The support of each itemset is found as: $(A, B) : 2$, $(A, C) : 1$, $(B, C) : 2$. Since itemset (A, C) does not satisfy our support threshold, this itemset is removed from our answer list. Based on the remaining two itemsets, a final itemset can be created of size three: (A, B, C). This itemset has a support of only one and does not satisfy our threshold. In general, the set of answers is the itemsets that are the longest and satisfies the support threshold. Thus, the final answers in this example are: (A, B) and (B, C).

Confidence is an interest measure used in Apriori that determines whether a certain "rule" best represents the transactional dataset. For example, based on the frequent itemsets we discovered based on the support threshold, (A, B) and (B, C), we can generate a set of possible rules in the form of: $A \Rightarrow B$, $B \Rightarrow A$, $B \Rightarrow C$, and $C \Rightarrow B$. For rule $A \Rightarrow B$, the confidence can be determined by first calculating the number of times item B occurs whenever an item A also occurs by the total number of transactions item A occurs. Thus, the confidence for this rule is: $2/2 = 1$ which means that for every transaction where the item A occurs, item B also occurs. Using the notation of "rules::confidence", the confidence of each rule in our example is: $A \Rightarrow B$::2/2, $B \Rightarrow A$::2/3, $B \Rightarrow C$::2/3, and $C \Rightarrow B$::2/2. Suppose our confidence threshold is 1, then the rules that satisfy are $A \Rightarrow B$ and $C \Rightarrow B$.

2.3 Example

In this paper, we use a synthetically generated dataset that is populated in a data warehouse and then accessed directly by a data mining method. Out of simplicity, the main example used throughout this paper is an instant messaging dataset that was synthetically generated and is mined by using the Apriori approach within within a data mining toolkit called Weka [12]. Other forms of datasets may be applied, and other data mining techniques can be applied to our framework.

Instant messaging datasets contain messages between users and within each message contain a set of words. Identifying associations between words may have several interesting applications such as identifying strong relationships between users, predictive text, etc. Instant messaging datasets can be modeled by having a message as transaction, where each word is an item type. Since this paper simply demonstrates that a data mining technique can use a data warehouse directly without any preprocessing, the actual types of preprocessing such as identifying synonyms and the removal of stop words are not considered in this paper.

3 Data Mining without a Data Warehouse

Several data mining techniques can be performed without the use of a data warehouse. In general, some form of a dataset is provided to the data miner that could by either synthetic or real. One of the main steps before a data mining algorithm can be applied is the preprocessing stage. The preprocessing stage may be different depending on the type of the algorithm used and the actual format needed by the algorithm.

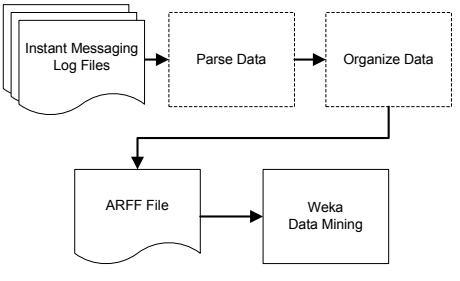

Fig. 1. Data Mining w/o Data Warehouse

Figure 1 depicts an example framework of performing a data mining algorithm without the use of a data warehouse. In this example, the input dataset contains instant messaging logs that may consist of information such as the user name, timestamp of the message, the actual message itself, etc. One of the basic steps in preprocessing is to parse the data to obtain which part of the log files is the user name, timestamp, or the message. Once the data is parsed, further preprocessing is completed to clean (i.e., remove invalid data) the data and re-format the file. In this example, the data is reformatted to an Attribute-Relation File Format (ARFF) [12] that can then be used within the Weka.

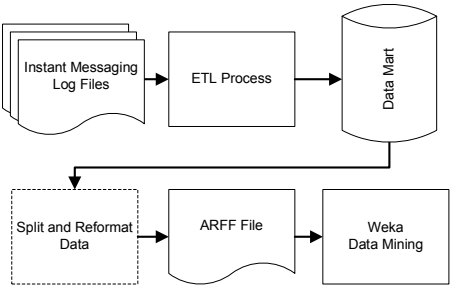

Fig. 2. Data Mining w/ Data Warehouse

4 Data Mining with a Data Warehouse

Data warehouses can be an excellent source for rich and cleaned datasets, and are commonly used for several data mining algorithms. Datasets in a data warehouse can naturally be cleaned using the ETL process. However, data from a data mart may not be able to be directly used for a data mining method. Thus, additional preprocessing may be required to conform the data to the required format for a data mining algorithm.

Figure 2 gives the general framework of performing a data mining method using a data warehouse. Unlike in Figure 1 where there is no data warehouse, Figure 2 does not need a separate step to parse the dataset. Rather, the dataset can be parsed and cleaned as one of the steps in building a data warehouse. Once the datasets are cleaned, additional preprocessing is required to conform the dataset in the required ARFF format that is later used in Weka.

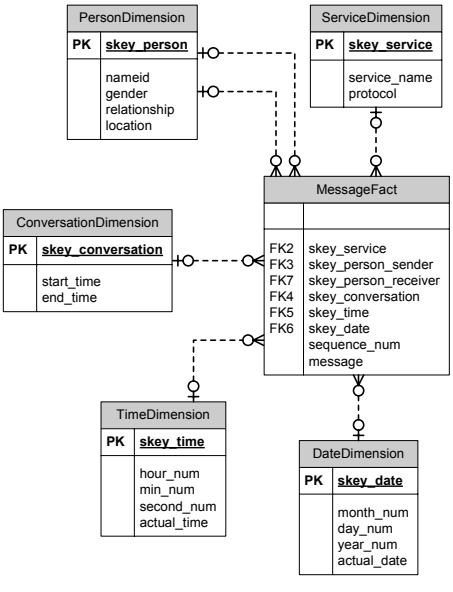

Fig. 3. Initial Mart Design

Figure 3 gives an example of the initial data mart design using the instant messaging dataset. The *grain* of the fact table is a message, and there are several dimensions including the messaging protocol (i.e., ServiceDimension), user name (i.e., PersonDimension), duration of the conversation (i.e., ConversationDimension), time of the message (i.e., TimeDimension), and the date of the message (i.e., DateDimension). It is important to note that one of the attributes within the fact table is "message" which contains the entire message for this user at this time. Further preprocessing will be required by the data miner to split each word in the message and format it for use in Weka.

5 Proposed Framework: Minable Data Warehouse

One of the main limitations in frameworks in Figures 1 and 2 is the aspect of creating additional preprocessing to manipulate the dataset. It is obvious as there are several manipulations done without a data warehouse, but even with a data warehouse, there is still additional preprocessing to parse each word in the messages. Thus, we propose a framework where all unnecessary preprocessing is removed in the entire process and that the data miner can access the data from the data warehouse directly and with ease.

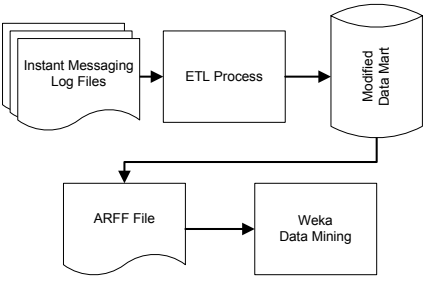

Fig. 4. Mineable Data Warehouse

Figure 4 gives our proposed framework to remove any unnecessary preprocessing between the data miner and the data warehouse. As in the second framework (Figure 2), the proposed framework uses the ETL process to ensure that the data is cleaned. However, by simply reducing the *grain* from the message to the word level will not make the data mart minable due to having a single attribute. For example, the Apriori technique needs a transaction that contains a set of words for each message. When we implement these words as a separate dimension, we have a many-to-many (M:N) relationship between the fact table and the word dimension table. A bridge table may be used to connect multiple words [9,11] to a message. The bridge table allows access to the fact table at a lower grain (word), and thus no additional pre-preocessing is required by the Apriori approach. The only preprocessing that is required is to produce the ARFF file for the Weka system.

Figure 5 gives an example of the logical design of our proposed framework. The key difference between our proposed framework and Figure 3 is the bridge table that links between the fact table and the word dimension. The word dimension contains all the words within each message. The data mart using a bridge table would allow the

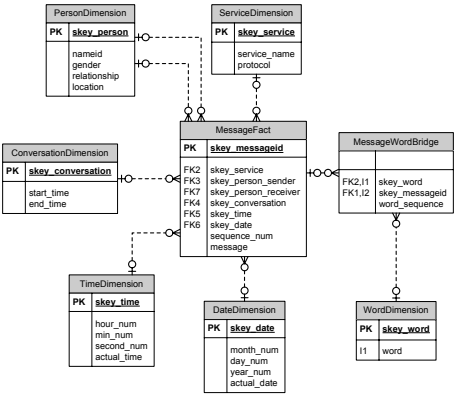

Fig. 5. Data Mart with Bridge Table

creation of an ARFF file without any additional preprocessing to parse the words from the message as in the framework in Figure 2.

5.1 Design Decision

A key design decision is proposed to eliminate all aspects of preprocessing between the data miner and the data warehouse. The general idea is to have all available information within the data warehouse itself and allow for a system such as Weka to directly access it.

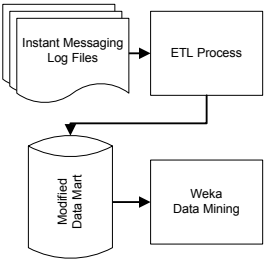

Fig. 6. Design Decision

Figure 6 gives the proposed framework using the design decision. The main difference between this framework and the one in Figure 4 is that an ARFF file does not need to be generated or preprocessed before a data mining system such as Weka can be executed. This is possible due to the structure of the data mart within the logical diagram.

Figure 7 gives the logical diagram of our proposed framework using the design decision. Essentially, the fact table is widened to contain all the words as separate columns in the message. Each word is in a bitmap format to reduce the space needed for each message and for a certain word occurring in the message. This allows a system such as

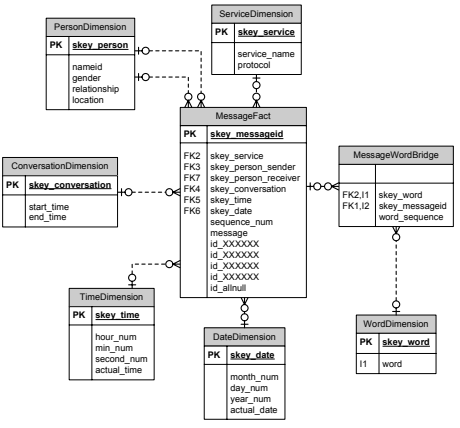

Fig. 7. Final Modified Data Mart

Weka to query the data warehouse directly and extract the data as a single transaction in its standard format (i.e., ARFF). Although this design decision will improve productivity significantly over previous frameworks, maintenance cost may be increased due to the additional columns in the fact table. Further partitioning may be required to reduce the number of words in the fact table and will be explored for future work.

6 Demonstration

In this section, we present a demonstration of our proposed framework (Figure 4) using an instant messaging dataset, Weka, and the output of the association rules. This demonstration illustrates that a data mining technique such as Apriori can directly access a data warehouse and produce association rules.

6.1 Input Files: Instant Messaging Files

Figure 8 gives an example of the synthetically generated instant message files that we used within our proposed framework. The input file contains the user name, the time stamp the message occurred and the message string. Since the proposed work focused on the general framework of using data warehouses and data mining, we simply used each word in the message.

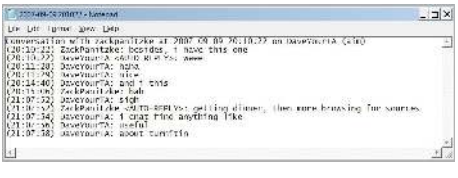

Fig. 8. A Sample Instant Messaging Log File (Best Viewed in Color)

6.2 Input to the Data Mining Tool: Weka

In our proposed framwork (Figure 4), we used the instant messaging files (Figure 8) and loaded them into the data warehouse. Under this proposed framework, we can produce an ARFF file which contains the words in the instant message file in terms of its bitmaps (Figure 9).

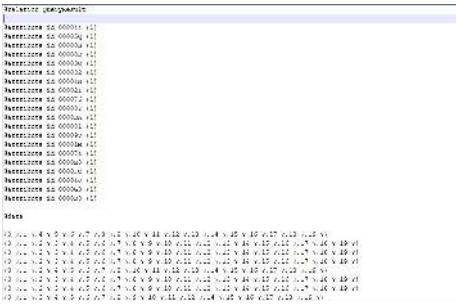

Fig. 9. ARFF File (Best Viewed in Color)

Based on our proposed approach using the design decision, another alternative approach in accessing the information from the data warehouse is by performing a query directly using the Weka system as shown in Figure 10. In this approach, we posed the following query "select * from messagefact", where "messagefact" is the fact table. This query will extract all the messages and its words into the Weka system.

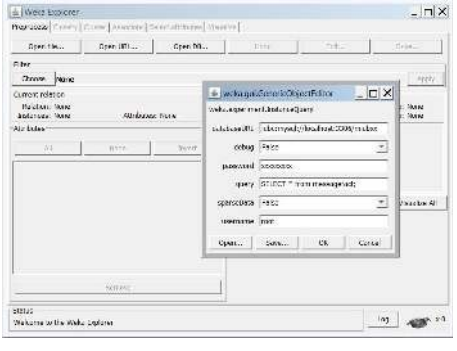

Fig. 10. Access Information from Data Warehouse Directly (Best Viewed in Color)

6.3 Files Viewed on Weka

Based on either using the ARFF file that was generated based on our proposed framework (Figure 4) or using our design decision (Figure 6), the information can then be viewed in Weka (Figure 11). The result illustrates the messages with its respective words on the left pane in the bitmap format along with the count of each word in the right pane in Figure 11.

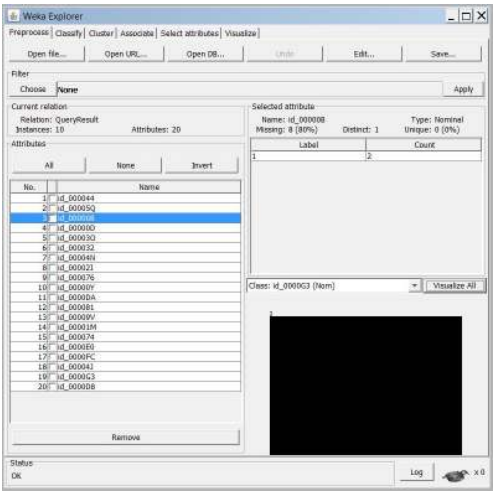

Fig. 11. Files Viewed on Weka (Best Viewed in Color)

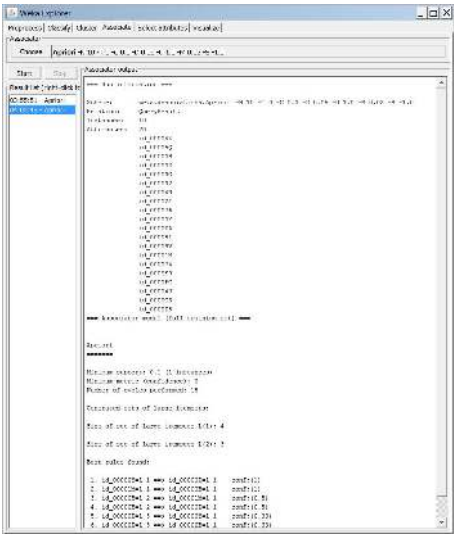

Fig. 12. Generated Association Rules (Best Viewed in Color)

6.4 Generated Association Rules

Figure 12 gives the generated set of rules produced by Weka based on the information provided by our proposed framework. The rules are located at the bottom portion of this figure. It is important to note that the illustration of these rules is simply to show that our framework has the capability to produce data mining patterns directly and is not intended to show the quality of the results. Thus, we can show that Weka can access the data warehouse directly and produce the following association rules in Figure 12.

7 Conclusions and Future Work

In this paper, we proposed a new framework that reduces the amount of preprocessing time that is required by the data miner when using a data warehouse as the main data source. We also present a design decision to remove all aspects of preprocessing to allow for an increased amount of efficiency to obtain data mining patterns. We evaluated our framework using sythetically generated instant messaging datasets and applied it using an Apriori method for association rule mining. We also presented a demonstration of our work using Weka.

We plan to explore alternative methods to improve the maintenance costs of our design decision while maintaining the efficiency to obtain the data mining patterns. Also, we plan on examining other forms of datasets that may be used for data mining techniques. Finally, we plan to generalize our framework to allow for other forms of data mining techniques to be applied by the proposed framework.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: ACM SIGMOD International Conference on Management of Data (1993)
2. Czejdo, B., Morzy, M., Wojciechowski, M., Zakrzewicz, M.: Materialized views in data mining. In: 13th International Workshop on Database and Expert Systems Applications, p. 827 (2002)
3. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: ACM SIGMOD International Conference on Management of Data (2000)
4. Hipp, J., Guntzer, U., Nakhaeizadeh, G.: Algorithms for association rule mining – a general survey and comparison. In: ACM SIGKDD Explorations Newsletter, vol. 2, pp. 58–64 (2000)
5. Inmon, W.H.: The data warehouse and data mining. Communications of the ACM 39(11), 49–50 (1996)
6. Inmon, W.H.: Building the Data Warehouse. John Wiley & Sons, Chichester (2002)
7. Jukic, N., Nestorov, S.: Comprehensive data warehouse exploration with qualified association-rule mining. Decision Support Systems 42(2), 859–878 (2006)
8. Kimball, R., Reeves, L., Ross, M., Thornthwaite, W.: The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing, and Deploying Data Warehouses. John Wiley & Sons, Chichester (1998)
9. Kimball, R., Ross, M.: The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling, 2nd edn. John Wiley & Sons, Chichester (2002)
10. Mclaren, I.: Designing the data warehouse for effective data mining (1998)
11. Song, I., Rowen, W., Medske, C., Ewen, E.: An analysis of many-to-many relationships between fact and diemension tables in demensional modeling. In: International Workshop on Design and Management of Data Warehouses (2001)
12. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, London (2005)