IMPROVING DATA INTEGRATION FOR DATA WAREHOUSE: A DATA MINING APPROACH

Kalinka Mihaylova Kaloyanova
"St. Kliment Ohridski" University of Sofia,
Faculty of Mathematics and Informatics
Sofia 1164, Bulgaria
kkaloyanova@fmi.uni-sofia.bg

ABSTRACT

Data warehousing embraces technology of integrating data from multiple distributed data sources and using that data in annotated and aggregated form to support business decision-making and enterprise management. Although many techniques have been revisited or newly developed in the context of data warehouses, such as view maintenance and OLAP, little attention has been paid to data mining techniques for supporting the most important and costly tasks of data integration for data warehouse design.

INTRODUCTION

Since the past decade data warehouses have been gaining enormous ground in the business intelligence (BI) domain. A corporate data warehouse was on every organization's priority list. Companies began to rely more and more on these BI systems. Critical business decisions were based on the current and historical data available in the data warehouse.

Data warehouse (DW) is a system that extracts, cleans, conforms, and delivers source data into a dimensional data store and then supports and implements querying and analysis for the purpose of decision making. For business leaders a corporate data warehouse and then consequently data mining seemed to the long term strategy. Sophisticated OLAP tools, which facilitate multidimensional analysis were used. Business trends are identified using data mining (DM) tools and applying complex business models.

As businesses grow from local to global, the complexities and parameters involved in decision making and analysis became more complex. [1]

The most visible part of a data warehouse project is the data access portion—usually in the form of products—and some attention is brought to the dimensional model. But by spotlighting only those portions, a gaping hole is left out of the data warehouse lifecycle. When it comes time to make the data warehouse a reality, the data access tool can be in place, and the dimensional model can be created, but then it takes many months from that point until the data

[1] This paper is partially supported by the project "Data processing and modeling in GRID" – Sofia University SRF under Contract № 63/2005

warehouse is actually usable because the ETL process (extraction, transformation, loading) still needs to be completed.

Data warehousing is the process of taking data from legacy and transaction database systems and transforming it into organized information in a user-friendly format to encourage data analysis and support fact-based business decision-making. The process that involves transforming data from its original format to a dimensional data store accounts for at least 70 percent of the time, effort, and expense of most data warehouse projects.

As it is very costly and critical part of a data warehouse implementation there is a variety of data extraction and data cleaning tools, and load and refresh utilities for DW.

Here we present another point of view to this problem – using data mining techniques to facilitate the integration of data in DW.

DATA INTEGRATION

The integration is one of the most important characteristic of the data warehouse. Data is fed from multiple disparate sources into the data warehouse. As the data is fed it is converted, reformatted, summarized, and so forth. The result is that data—once it resides in the data warehouse—has a single physical corporate image.

Many problems arise in this process. Designers of different applications made up their decisions over the years in different ways. In the past, when application designers built an application, they never considered that the data they were operating on would ever have to be integrated with other data. Such a consideration was only a wild theory. Consequently, across multiple applications there is no application consistency in encoding, naming conventions, physical attributes, measurement of attributes, and so forth. Each application designer has had free rein to make his or her own design decisions. The result is that any application is very different from any other application.

One simple example of lack of integration is data that is not encoded consistently, as shown by the encoding of gender. In one application, gender is encoded as ***m*** or ***f***. In another, it is encoded as **0** or **1**. As data passes to the data warehouse, the applications' different values must be correctly deciphered and recoded with the proper value.

This consideration of consistency applies to all application design issues, such as naming conventions, key structure, measurement of attributes, and physical characteristics of data. Some of the same data exists in various places with different names, some data is labeled the same way in different places, some data is all in the same place with the same name but reflects a different measurement, and so on.

Whatever integration architecture we choose, there are different problems that come up when trying to integrate data from various sources.

SCHEMA INTEGRATION

The most important issue in data integration is the *Schema integration*. How can equivalent real-world entities from multiple data sources be matched up? This is referred to as entity identification process. Terms may be given different interpretations at different sources. For example, how can be data analyst be sure that *customer_id* in one database and *cust_number* in another refer the same entity?

Data mining algorithms can be used to discover the implicit information about the semantics of the data structures of the information sources.

Often, the exact meaning of an attribute cannot be deduced from its name and data type. The task of reconstructing the meaning of attributes would be optimally supported by dependency modeling using data mining techniques and mapping this model against expert knowledge, e.g., business models. Association rules are suited for this purpose. Other data mining techniques, e.g., classification tree and rule induction, and statistical methods, e.g., multivariate regression, probabilistic networks, can also produce useful hypotheses in this context.

Many attribute values are (numerically) encoded. Identifying inter-field dependencies helps to build hypotheses about encoding schemes when the semantics of some fields are known. Also encoding schemes change over. Data mining algorithms are useful to identify changes in encoding schemes, the time when they took place, and the part of the code that is effected. Methods which use data sets to train a “normal” behavior can be adapted to the task. The model learned can be used to evaluate significant changes. A further approach would be to partition the data set, to build models on these partitions applying the same data mining algorithms, and to compare the differences between these models.

Data mining and statistical methods can be used to induce integrity constraint candidates from the data. These include, for example, visualization methods to identify distributions for finding domains of attributes or methods for dependency modeling. Other data mining methods can find intervals of attribute values, which are rather compact and cover a high percentage of the existing values.

Once each single data source is understood, content and structural integration follows. This step involves resolving different kinds of structural and semantic conflicts. To a certain degree, data mining methods can be used to identify and resolve these conflicts.

Data mining methods can discover functional relationships between different databases when they are not too complex. A linear regression method would discover the corresponding conversion factors. If the type of functional dependency (linear, quadratic, exponential etc.) is a priori not known, model search instead of parameter search has to be applied.

REDUNDANCY

Redundancy is another important issue. An attribute may be redundant if it can be “derived” from another table, e.g. *annual revenue*. In addition to detecting redundancies between attributes, duplication can be detected at the tuple level (e.g., where there are two or more identical tuples for a given unique data entry case).

Some redundancies can be detected by correlation analysis. For example, given two attributes, such analysis can measure how strongly one attribute implies the other, based on available data.

Inconsistencies in attribute or dimension naming can also cause redundancies in the resulting data set.

INCONSISTENCIES

Since a data warehouse is used for decision-making, it is important that the data in the warehouse are correct. However, since large volumes of data from multiple sources are involved, there is a high probability of errors and anomalies in the data. Real-world data tend to be incomplete, noisy and inconsistence.

Data cleansing is a non-trivial task in data warehouse environments. The main focus is the identification of missing or incorrect data (noise) and conflicts between data of different sources and the correction of these problems.

Data cleansing routines attempt to fill in missing values, smooth out noise while identifying outliers, and correct inconsistencies in the data. Some examples where data cleaning becomes necessary are: inconsistent field lengths, inconsistent descriptions, inconsistent value assignments, missing entries and violation of integrity constraints.

Typically, missing values are indicated by blank fields or special attribute values. A way to handle these records is to replace them by the mean of most frequent value or the value, which is most common to similar objects. Simple transformation rules can be specified; e.g., "replace the string *gender* by *sex*".

Missing values may be determined with regression, inference-based tools, using a Bayesian formalism, or decision tree induction.

Advanced data mining methods for completion could be similarity based methods or methods for dependency modeling to get hypotheses for missing values.

On the other hand, data entries might be wrong or noisy. For such kind of problems is proper to be used tree or rule induction methods. If we have a model, that describe dependencies between data, every transaction that deviate from the model is a hint for an error or noise.

There are some *Data scrubbing* tools that use domain-specific knowledge (e.g., postal addresses) to do the scrubbing of data. They often exploit parsing and fuzzy matching techniques to accomplish cleaning from multiple sources. Some tools make it possible to specify the "relative cleanliness" of sources. Another tools - *Data auditing* tools make it possible to discover rules and relationships (or to signal violation of stated rules) by scanning data. Thus, such tools may be considered as variants of data mining tools. For example, such a tool may discover a suspicious pattern (based on statistical analysis) that a certain car dealer has never received any complaints.

Different sources can contain ambiguous or inconsistent data, e.g., different values for the address of a customer or the price of the same article. With clustering methods you might find records, which describe the same real-world entity but differ in some attributes, which have to be cleaned.

Record linkage techniques are used to link together records, which relate to the same entity (e.g. patient or customer) in one or more data sets where a unique identifier is not available.

MULTIDIMENSIONAL DATA MODELING

Data sets for analysis may contain hundreds of attributes, many of which may be irrelevant or redundant. Sometimes it is not sensible to model all the fields as dimensions of the cube as some fields are functionally dependent and other fields do not strongly influence the measures.

Although it may be possible for domain expert to pick out some of the useful attributes, this can be a difficult time-consuming task, especially when the behavior of data is not well known.

Data mining methods can help to rank the variables according to their importance in the domain. Non-correlated sets of attributes can be found with correlation analysis. Furthermore, using data mining methods to drive this cube design seems promising as they can help to identify (possibly weak) functional dependencies, which indicate non-orthogonal dimension attributes.

Sparse regions should be avoided during modeling. Using special cluster methods (e.g. probability density estimation) data points can be identified as the center of dense regions.

The multidimensional paradigm demands that dimensions are of discrete data type. Therefore, attributes with a continuous domain have to be mapped to discrete values if this attribute is to be modeled as a dimension. Algorithms, that find meaningful intervals in numeric attributes help to get discrete values.

CONCLUSION

The paper presents the idea of using data mining methods for supporting the most important and costly tasks of data warehousing – data integration. Tools that use data mining techniques for this process still are rare. Building of such kind of tools is an important direction for more efficient implementation of data warehousing.

REFERENCES:

1. Trillium Software System: *Achieving Enterprise Wide Data Quality*, White Paper, 2000
2. Chaudhuri S., Dayal U.: *An Overview of Data Warehousing and OLAP Technology*, SIGMOD Record 26(1): 65-74 ,1997
3. Netz, A.; Chaudhuri, S.; Fayyad, U.; Bernhardt, J.: *Integrating data mining with SQL databases: OLE DB for data mining* , ICDE 2001, Page(s): 379 -387
4. Oracle Corporation:*Oracle9i Data Mining Concepts Release 2 (9.2*), P.No.A95961-01, 2002
5. Oracle Corporation*: Oracle9i Data Mining*, www.oracle.com/technology/products/oracle9i/pdf/o9idm_bwp.pdf, 2002
6. Paul S., MacLennan J., Tang Z., Oveson S.: *Microsoft SQL Server 2005 Data Mining Tutorial*. Microsoft Corporation, 2004