글로벌경영연구
第 22 卷 1 號
2010年 2月

데이터마이닝 관점에서 대용량 모델관리에 대한 고찰[†]

조준서[*]

The Model Management Concerns of Large Collections-Data Mining Perspectives

Abstract

데이터마이닝은 최근 기업의 성공에 중요한 요소가 되었다. 기업의 여러 프로세스로부터 많은 데이터가 모아지고 이에 따라 복잡한 모델의 데이터마이닝 모델에 대한 이해가 요구된다. 이 논문은 데이터마이닝 관점에서 어떻게 대용량 모델을 관리할 수 있는가에 대해서 논의하며, 이에 대한 중요성과 접근방법 및 가이드라인을 제시한다. 또한 기대되는 혜택에 대해서도 논의한다.

Ⅰ. Introduction

Data mining has become increasingly critical for the success of companies in this emerging era. As company management activities increasingly shift to the web, the challenge facing corporate management is maintaining competitive advantage by building strong relations with employees, customers, and suppliers, and partners. Business analysts in financial, telecommunication, and retail industries collect huge amounts of data on sales, customer behavior and partner profiles, etc. Data mining models require deep understanding of complex composites models.

There are serious challenges regarding building, updating, and sharing complex data mining models across the industries. Model building is a key objective of data mining and data analysis applications. In

[†] 본 논문은 2009년도 한국외국어대학교 교내 연구비지원에 의한 것임.

[*] 한국외국어대학교 글로벌 경영대학

the past, such applications required only a few models built by a single data analyst. As more and more data has been collected and real world problems have become more complex, it has become increasingly difficult for that data analyst to build all the required models and manage them manually.

As data analysis and data mining is increasingly widely used in practice, there is a clear need to manage the data mining process and the generated models. It is the aim of this research to provide guideline and help data analysts in the model building process by automating the process of building, managing and analyzing the models to the fullest extent possible. In this research we will discuss about Model Management concerns with Large Data Mining Models.

Ⅱ. Context of Research in Model Management

Model management has been studied in the Information Systems (IS) community in the context of managing models in decision support systems (DSS) since the mid-70's when the term "model management" was coined in (Sprague and Watson, 1975) and (Will, 1975). The early work on model management was greatly influenced by the work on database management. In particular, it was argued that, as in databases, it is important to insulate users from physical details of storing and processing models (Dolk and Konsynski, 1984). This led to the approach treating models as black boxes having only names, inputs and outputs and to the development of query languages and algebras for manipulating the models that had such operators as model solution, model composition and sensitivity analysis operators (Blanning, 1985). However, some researchers also argued that treating models as black boxes has certain limitations and that there is a need to consider the structure of a model in the context of a modeling lifecycle (Ari et al., 2008; Geoffrion, 1987). This lifecycle modeling work has primarily been focusing on the OR/Management Science (OR/MS) types of models, such as mathematical programming, production and distribution, network, transportation and other types of OR/MS models and covered all the aspects of the lifecycle modeling ranging from the problem identification to the model maintenance stages of the modeling lifecycle (Tuzhilin, 2002). Some of the more recent surveys of model management can be found in (Blanning, 1993; Krishinan and Chari, 2000) and include such topics as model formulation, selection, composition, integration, implementation, and interpretation issues.

Although this work in the IS community introduced several useful ideas, most of them were applicable primarily to OR/MS models and focused on organizational issues. There was little work on managing statistical and data mining models, on model analysis and inferencing, on the development of algorithms for managing these models, and also on building actual systems.

In the data mining community, the problem of managing very large numbers of discovered rules was studied by a number of researchers within the context of data mining query languages. One of the earliest data mining query language is the one based on templates (Klementoen et al., 1994). In this technique, the user uses a template to specify what items should be in or not in a rule, and what level of support and/or confidence are required. The system then checks each mined rule to find those matching rules. Templates can be seen as a special case of a query language on modelbases that are not limited to collections of rules but also contain various types of heterogeneous models.

(Han et al., 1996) presents a data mining query language, called DMQL. DMQL allows the user to specify from what table (and database) to mine what types of rules. Its main purpose is to select the right data to mine different types of rules (Meo et al., 1996). proposes an SQL-like operator for data mining (MINE RULE). Also, (Shen et al., 1996) reports a meta-query language for data mining. Both these approaches are similar to DMQL. They are not designed for querying the mined rules, but enabling the user to specify what data mining task to perform and what its required data are (Imielinski et al., 1999; Virmani and Imielinski, 1999). report a more powerful data mining query language, called MSQL. MSQL can be used not only for rule generation, but also for querying the discovered rules. With regard to rule querying, MSQL is similar to templates but allows more complex conditions. Like templates, MSQL's query conditions can be checked using the information contained in each rule itself, e.g., support, confidence, rule length, items on the left-hand side or the right-hand side of the rule (Morzy and Zakrzewicz, 1998). proposes an index for retrieval of association rules stored in a relational database.

(Agrawal et al., 1995) report a language for querying shapes of history. The shapes of history refer to ups and downs of supports or confidences of a rule over a number of time periods. This language enables the user to define the shapes of the rules that the user is interested in. For example, one may want to find rules whose support increases in one time period and then falls down.

The idea of managing large collections of data mining models, beyond querying large numbers of association rules, has been expressed recently in the data mining community. In particular, it provides several model building methods, including decision tree, Naïve Bayes, association rules and clustering, and enables manipulation of models using the Data Mining Extension (DMX) language that is similar to SQL. Tools are also provided for testing and comparing model accuracy.

Also, in several papers (Bernstein and Melnik, 2007; Bernstein and Rahm, 2000; Bernstein, 2003; Melnik et al., 2003; Zhu and Tang, 2009), Bernstein and his colleagues studied the model management problem in the database context. Although it uses the same name, the concepts are quite different. In their work, models mainly refer to schemas and meta-data of relational database systems. Its goal is to develop a generic infrastructure to facilitate model-driven applications, such as database tools, application design tools, message translators, and customizable commercial applications. More specifically, examples of model-driven applications are object-at-a-time programming on relational schemas, DTDs, web-site structures, ER diagrams, UML models, etc. However, some of the high level ideas advocated by Bernstein et al. may be applicable to our model management context. Note that the reason that we use the term “model management” is because it is an understood term in data analysis and business communities, which will be the main user communities of our techniques and systems.

We believe that it is time to revisit the model management problem studied in the IS community, refocus it on managing large collections of data mining models.

Ⅲ. Discussion of Model Management based on Large Collections

1. Objectives

This paper aims to discuss guidelines

of methods for building, managing and analyzing large heterogeneous modelbases consisting of large collections of different types of data mining and statistical models. In particular, we discuss to focus on the followings:

- Process of improving automated/semi-automated generation of a large collection of models.
- Analysis of the modelbase, and identification of underperforming and redundant models.
- Determination of how to improve the modelbase by modifying and removing "poor" models and adding new promising models.

Moreover, we plan to make these improvements continuous through an iterative process of identification and fixing of "weaknesses" in the modelbase. Since we deal with large collections of models, there is a need to automate model generation, management and analysis processes, and at the same time letting the data analyst provide crucial inputs into these processes. All these tasks need novel solutions and techniques. So far, little research has been done with a broad range of models. Apart from research, integrating all the tasks and new techniques in a single framework and system presents an engineering challenge as well because we no longer deal with a single type of models. New techniques need to work across multiple types of models, which have very different structures and semantics.

2. Significance

In the past, statistical and data mining applications required only a few models that are built by a data analyst. As real-world applications become more and more complex and require a larger and larger number of models, it is getting very hard for a data analyst to build all these different models and to manually manage them. Even in applications that need only a single good model, the data analyst typically has to try and build a large number of models based on available data, insight, domain knowledge and previous experiences to generate the final model. The process is labor intensive and very time consuming. Managing such large collections of models is becoming a pressing issue.

For example, customer segmentation constitutes one of the key concepts in marketing (Kotler, 2002). Traditionally, marketers divided their customer bases into a small number of segments, such as pool-and-patio (suburban well-to-do customers who would usually own a house with a pool) and empty-nesters (middle-aged customers whose children left the house for college), and manually built statistical models describing behavior of each segment. A more recent trend in marketing is to partition customer bases into smaller and smaller segments, called micro-segments (or niche-segments) (Kotler, 2002), such as the pool-and-patio customers living in a certain zip code. In applications with large customer bases, such as major credit card applications, there can be thousands of such micro-segments. If purchasing behavior of each segment is represented with several models describing different aspects of the customer behavior, then the total number of models for such applications can be measured in tens or even hundreds of thousands of models.

Consider a credit-card marketing application. The credit-card-issuing company wishes to build models describing the behavior of small segments of customers, or microsegments. Examples are middle-age customers with children in college living in zip code and graduate engineering students at university. A large credit-card company might have to deal with tens of thousands of such microsegments, each involving dozens of different models. Therefore, it may need to build and support hundreds of thousands of models. Similar problems also occur in personalized applications and e-commerce.

Similar situation also occurs in some bio-informatics applications, such as microarray applications, where dimensionality of data is very large, often measured in tens of thousands of variables (attributes). To have a good understanding of the problem, one may need to build different models on the microarray data using different subsets of variables (attributes). Because of the combinatorial explosion, this can result in a huge number of models, often measured in hundreds of thousands or even millions of models (Tuzhilin and Adomavicius, 2002).

Another example requiring management of a large collection of data mining models occurs when a data analyst generates a large number of tries before finding the right model. This situation often occurs in exploratory data analysis. It is seldom the case that a successful data mining application is carried out by simply extracting the relevant data from a database and then running a data mining algorithm. Usually, the process of data analysis constitutes a manual and very labor-intensive and time-consuming intellectual activity requiring understanding of the application problem, insightful knowledge and experiences of the data analyst and a large number of tries and tests in order to produce the final (useful) models. To add to the complexity, there are many types of models that one can try, e.g., decision trees, regressions, SVMs, rules, etc. Clearly, there is a need to help the data analyst manage this process and all the different models so that the analyst can easily study the models, ask questions about them and test them with minimal effort and also without being lost in the process.

The traditional approach is to aggregate the data into large segments, then use domain knowledge combined with "intelligent" model-building methods to produce a few good models. Intelligent means selecting the right functions and model types based on automatic algorithms and domain expert knowledge. This approach reduces the number of models. However, it does not eliminate the need for a large number of models in practice and is not ideal because some important characteristics of smaller models are lost in the aggregated models. The approach thus represents a compromise due to a human analyst's own limited intellectual capacity. With today's computing power, building a large number of models is not an issue. The bottleneck is the human analyst.

An initial approach to developing such tools (Liu et al., 2006) involved an application in Motorola, Inc. The system (called "Opportunity Map"), which has been in regular use since 2006, includes a set of customer designed rule-manipulation and visualization operations. However, the approach lacks a formal foundation, and new features are constantly being added in an ad hoc manner. Model-management research should give the approach a theoretical foundation and extend it to other types of models rather than only to rules. So, we need to develop efficient tools and methods for model management based on models including large collection of data mining models.

3. Contents and Methods

This research aims to give guideline of new methods for building, managing and analyzing large heterogeneous modelbases consisting of different types of data mining and statistical models. In particular, we will focus on the following research questions in this research:

- How to automatically generate a large collection of statistical and data mining models.
- How to analyze this modelbase and identify underperforming and redundant models that either need to be modified to improve their performances or deleted from the modelbase. We also need to study the problem of making inferences on the modelbase in order to identify new models to be added to the modelbase. The process of modifying a model can either be manually performed by a domain expert using various analysis tools or be automatically performed by a software system deploying various model inferencing techniques.
- Development of an iterative process of continuous improvement of the modelbase, i.e., a method that would analyze the modelbase using the techniques described the above paragraph and would identify underperforming and also important models missing from the modelbase. If such models are identified, the modelbase would be updated with the help of the domain expert, and the iterative process would continue.

According to accomplish our goal of this research, we need to perform the contents as follows:

- Generation of a Large Initial Set of Data Mining Models

An important research question is how to generate a large number of models using inputs from the domain expert and integrate these inputs into automated model generation. This is a non-trivial problem. We investigate it in the proposed research by considering an iterative procedure in which a domain expert iteratively specifies constraints on the types of models he or she wants to generate.

- Manual Approach to Model Analysis

The "manual" approach constitutes a collection of different techniques and tools that allow the domain expert to examine and evaluate large collections of heterogeneous models quickly and effectively. We need to study the following manual techniques and tools including query language, usage analysis and reporting methods, and examination operators.

- Automated Identification and Modification of Underperforming Models

We need to discuss how to identify underperforming models and replace them with better performing models.

- Automated Identification of Dominated Models in the Modelbase

We need to generalize these concepts of dominance to other types of models and develop efficient methods that would effectively remove dominated models from the modelbase.

- Adding New Models to the Modelbase

We need to work on methods for deriving better models from models that already exist in the model, evaluating them and considering for possible additions to the modelbase.

Furthermore, we need to consider how to implement the system which is based on our discussion with the experimental evaluation.

4. Expected Research Benefits

Model management systems should help the data analyst to build and manage very large modelbases (VLMBs) and improve their performance by identifying underperforming models and changing or even removing them from the modelbase. Moreover, model management systems should help organizations by providing commonly shared modeling resources and easy access to these resources by less sophisticated end-users via user-friendly model access capabilities. The resulting model management systems would provide the following benefits:

- They would help data analysts to explore more possibilities by automatically identifying underperforming or "useless" models and also determining new models that should be added to the modelbase.
- They would significantly expand cognitive limitations of data analysts and allow them to build and manage a much larger number of models and manage the model building process.
- Model management systems would make data mining models a commonly shared resource in an enterprise similar to the way that DBMSes make data a commonly shared resource. Consequently data mining technologies would become more accessible to larger audiences.

It is our belief that our system can make it easy for general public to benefit from data mining results. For example, rules and other models mined from a disease database can be published on the Web so that the general public can query the models using a friendly user-interface. Also, model management systems should help organizations by providing commonly shared modeling resources and easy access to these resources by less sophisticated end-users via user-friendly model access capabilities. The proposed methods and systems can be applied to statistical and data mining applications, provided to educational and research organizations for model management.

Ⅳ. Conclusion

As data analysis and data mining is increasingly widely used in practice, there is a clear need to manage the data mining process and the generated models. We discussed about the guidelines of methods for building, managing and analyzing large heterogeneous models consisting of large collections of different types of data mining and statistical models. The aim of this research is to help data analysts in the model building process by automating the process of building, managing and analyzing the models to the fullest extent possible.

Research guidelines should be performed step by step with the large collections of data mining models. One of the issues that we will need to address as a part of the implementation research is how to organize and store a model. We need to implement the proposed system in Relational Database systems (RDBMS) such as Oracle and SQL Server etc. Implementing our system as a part of an existing RDBMS has many advantages. It will not only allow querying of models to be an integrated part of the mainstream database systems, but would also enable seamless integration of data querying and model querying, which are important in practice and for our research.

It is our expectation that the results of our research should be of interest to our marketing colleagues and should have impact in the marketing profession (both in the academia and the industry). We believe that this guideline presents significant intellectual challenges for both research and system building in data mining area.

References

[1] Agrawal, R., Psaila, G., Wimmers,

E., and Zait, M., "Querying shape of histories," *VLDB-95*, 1995.

[2] Ari, I. Li, J. Jain, J. and Kozlov, A., "Management of Data Mining Model Lifecycle to Support Intelligent Business Services," *ACM Computer Human Interaction for Management of IT*, Nov, 2008.

[3] Bernstein P.A. and Melnik S, Model Management 2.0: Manipulating Richer Mappings, *SIGMOD*, 2007.

[4] Liu, B., Zhao, K., Benkler, J., and Xiao, W., "Rule interestingness analysis using OLAP operations," *In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006.

[5] Bernstein, P.A. and E. Rahm, "Data Warehouse Scenarios for Model Management," *ER2000 Conference Proceedings*, Springer-Verlag, 2000.

[6] Bernstein, P.A., "Applying Model Management to Classical Meta Data Problems," *Proc. CIDR*, 2003.

[7] Blanning, R.W., "A Relational Theory of Model Management," Working paper presented at the NATO ASI Program on Decision Support Systems, Maratea, Italy, June 1985.

[8] Blanning, R.W., Model Management Systems: An Overview, *Decision Support Systems*, Vol.9(1993).

[9] Dolk, D.K. and B.R. Konsynski, "Knowledge Representation for Model Management Systems," *IEEE Transactions on Software Engineering*, Vol.10, No.6(1984).

[10] Geoffrion, A.M., "An Introduction to Structured Modeling," *Management Science*, 1987.

[11] Han, J., Fu, Y., Wang, W., Koperski, K., and Zaiane, O., "DMQL: a data mining query language for relational databases," *SIGMOD Workshop on DMKD*, 1996.

[12] Imielinski, T., Virmani A., and Abdulghani, A., "DMajor-Appliction programming interface for database mining," *Journal of DMKD*, 1999.

[13] Klemetinen, M. Mannila, H. Ronkainen, P. Toivonen, H., and Verkamo, A.I., "Finding interesting rules from large sets of discovered association rules," *CIKM-1994*, 1994.

[14] Kotler, P., Marketing Management, Prentice Hall, 11th. ed., 2002.

[15] Krishnan, R. and Chari, K., "Model Management: Survey, Future Directions and a Bibliography," *Interactive Transactions of OR/MS*, Vol.3, No.1(2000).

[16] Melnik, S., Rahm, E., Bernstein, P.,

"Rondo: A Programming Platform for Generic Model Management," *SIG-MOD*, 2003.

[17] Meo, R. Psaila, G., and Ceri, S., "A new SQL-like operator for mining association rules," *VLDB-96*, 1996.

[18] Morzy, T. and Zakrzewicz, M., "Group bitmap index: A structure for association rules retrieval," *KDD-98*, 1998.

[19] Shen, W-M. Ong, K-L. Mitbander, B., and Zaniolo, C., "Metagueries for data mining," *In Adv. in Knowledge Disc., and Data Mining*, AAAI/MIT Press, 1996.

[20] Sprague, R.H. and H.J. Watson, "Model Management in MIS," *Proceedings of Seventeenth National AIDS*, Nov. 1975.

[21] Tuzhilin, A. and Adomavicius, G., "Handling Very Large Numbers of Association Rules in the Analysis of Microarray Data," *KDD-02*, 2002.

[22] Virmani A. and Imielinski, T., "M-SQL: A query language for database mining," *Journal of DMKD*, 1999.

[23] Will, H.J., "Model Management Systems" in *Information Systems and Organization Structure*, ed. by Edwin Grochia and Norbert Szyperski, 1975.

[24] Zhu, J. and Tang, Y., "A Dynamic Data Mining Models for Engineering Management," *ISECS International Colloquium on Computing, Communication, Control, and Management*, 2009.