Data Mining: A Clustering Application

Lin Chang  
*Shenyang Institute of Technology, lchang@sut.edu.cn*

Xue Bai  
*Shenyang Institute of Technology, xbai@sut.edu.cn*

Follow this and additional works at: [http://aisel.aisnet.org/pacis2010](http://aisel.aisnet.org/pacis2010)

Recommended Citation

[http://aisel.aisnet.org/pacis2010/184](http://aisel.aisnet.org/pacis2010/184)
DATA MINING: A CLUSTERING APPLICATION

Lin Chang, Shenyang Institute of Technology, P.R. China, lchang@sut.edu.cn
Xue Bai, Shenyang Institute of Technology, P.R. China, xbai@sut.edu.cn

Abstract

Developing a good clustering technique is a challenging task in data mining. Several models, such as artificial neural network approaches and genetic algorithms, represent a promising approach to address this challenge. This paper presents an experiment comparing a map-based training method used for clustering with a traditional clustering method. We test the effectiveness of both methods in helping decision makers in the information systems domain. Our analysis shows that the experimental group outperformed the controllable group. Statistical evidence shows that both training approaches are effective and present good results. Recommendations are made to researchers and practitioners to improve the efficacy of the map-based training approach.

Keywords: Data Mining, Information Theory, Knowledge Map.
1 INTRODUCTION

Two approaches are commonly used in data mining to generate models: directed or undirected (Woo and Hu, 2009). Directed modeling is a goal-oriented modeling. The task is to explain the value of some particular fields. The user selects the target field and directs the computer to tell how to estimate, classify, or predict it. In undirected modeling, there is no target field. The user asks the computer to identify patterns in the data that may be significant. In other words, undirected modeling is used to recognize relationships in the data, and directed modeling is used to explain those relationships once they have been found (Duchamp and Green, 2009). This paper refers to directed modeling, as it intends to focus on a specific field in the database, to perform a comprehensive analysis and to make some predictions.

In this review a distinction between data mining methods, data mining techniques and data mining tools is made. While data mining methods refer to the general conceptions of data mining, data mining techniques specify the particular procedures of “how to mine” (Fang, 2009a). Data mining tools are software or algorithms which make use of the data mining methods and techniques to effectively mine and to provide an enlightened and meaningful analysis (Porter and Green, 2009).

Traditional data mining methods were based on manual handling (Ben-Zvi, 2009). This involved a user formulating a conjecture about a possible relationship in the data and evaluating this hypothesis through a statistical test. This was a largely time-consuming top-down approach to data analysis. Today, data mining is conducted by an automated data mining algorithm rather than by the user. Data mining is a data-influenced, bottom-up approach to data analysis (Fang, 2009b). The different approaches uncover patterns and relationships through the stacks of data, and then make predictions, describe behaviors or facilitate decision support (Ben-Zvi, 2010).

Contemporary methods combine data management technologies and innovative computational methods with analytical techniques taken from the diverse fields of statistics, machine learning and artificial intelligence: regression, classification, clustering, and so on. The array of different algorithms under each of these headings can often be bewildering to both the novice and the experienced data analyst (Fayyad, Piatetsky-Shapiro and Smyth 1996b). The actual underlying model representation being used by a particular method typically comes from a composition of a small number of well-known options: polynomials, kernel and basis functions, threshold-boolean functions, and so on. Thus, algorithms tend to differ primarily in the goodness-of-fit criterion used to evaluate model fit or in the search method used to find a good fit.

When considering object-oriented and database-driven programming skills, many challenges are ahead for novice users who wish to acquire those skills. The building blocks of object-oriented programming languages like VB.NET or ASP.NET are comprised of at least five skill sets: (1) procedural diagram, (2) event-driven diagram, (3) object-oriented paradigm, (4) integrated development environment (IDE), and (5) Microsoft Development Network (MSDN). Each skill set has its own complexity. For instance, MSDN can be configured in numerous ways to provide services for different programming languages.

Unlike scripted programming languages (e.g. COBOL, C++), object-oriented programming languages rely on six basic ideas, including objects, classes, messages, encapsulation, inheritance and polymorphism. Grasping all realms of these ideas is essential to master object-oriented programming languages, and not too difficult to comprehend for a novice user. Yet, applying these ideas in coding and debugging coding errors are somewhat challenging because errors can be spread into five layers: (1) class/object, (2) structure, (3) service, (4) attribute, and (5) subject layers. As a result, helping novice students master these skill sets within a regular school semester has proven a challenging task for many instructors. The learning issue can be further exacerbated with the introduction of database access component like ADO.NET into an object-oriented programming course. In addition to those five skill sets, a novice user needs to have solid knowledge and skills in the area of database in order to successfully complete an object-oriented and database-driven software course. These challenges
call for our attention to devise efficient and effective teaching methods to help novice users succeed in this learning task.

We consider a knowledge map method as an alternative approach to the traditional regression, classification and clustering methods. We show that this method can be used in data mining applications and yield good results. The next section elaborates on this method.

2 KNOWLEDGE MAPS AS AN ALTERNATIVE

Many software applications fail to meet users’ requirements on time and on budget. Unified modeling language (UML) allows programmers to model application structure, behavior, architecture, business process, and data structure. UML is a diagramming method to improve the success rate of software design projects. The creation of UML is based on the idea of concept maps. Concept maps use a visual structure to show the location of major concepts and the relationships among them. Concept maps can enhance a meaningful and active learning process by helping users integrate the existing cognitive structures of learners with new concepts (Novak, 1985). New knowledge can be constructed from the integration process. A concept map can be customized to represent the cognitive structure (schema and mental models) of users to ease the process of constructing new concepts. A learner is more likely to have deeper learning or better learning performance when the individual learner forms his/her own cognitive structure (Bruner, 1966).

The knowledge map is an enhanced version of concept map. Knowledge maps are a tool to organize ideas, opinions and propositions, the constitution of knowledge (Anderson, 1983), in a multi-level and hierarchical structure to represent mental models of a learner. The visualized approach to represent the structure of abstract knowledge of a learner allows an instructor to assess the learning performance of a trainee (Laffey and Singer, 1997). Constructing learning via knowledge maps can help students externalize their knowledge in a visual format, which can lead to a higher cognitive ability.

Concept and knowledge maps have commonalities in using concept-like notes and labeled links between nodes (Chmeilewski et al., 1998). Nodes of knowledge maps contain more information than nodes of concept maps. Thus, the size of a node is larger in a knowledge map. Concept maps are primarily hierarchical in nature while knowledge maps are more flexible in structure (Lambiotte et al., 1989). A well-constructed knowledge map can help students recognize and recall course materials and build up new knowledge based on the understanding of the studied course materials (Chmeilewski and Dansereau, 1988). Knowledge maps are an effective alternative of hypertext-based and list interfaces in assisting the learning process of users (Dansereau, 1990). Knowledge maps are networks of concepts, consisting of nodes (concepts) and links (relationships between concepts). Links can be non-, uni- or bi-directional to represent different categories of concepts and relationships, such as causal or temporal relationships. Knowledge maps are useful tools to (1) communicate complex ideas, (2) improve learning by constructing new knowledge based on the existing knowledge, and (3) assess understanding of studied subjects (Novak, 1991).

Some essential attributes are required for the construction of knowledge maps (Table 1). Various attributes and symbols are adopted to represent different relationships among knowledge objects, descriptive relationship, dynamic relationship and instructional relationship. Mindmapper (http://www.mindmapper.com) and Semiomap (http://www.semio.com) are two popular knowledge map applications that have been widely adopted in educational and organizational contexts, respectively. These knowledge mapping tools capitalize on the strength of visual/spatial display, which is considered a better delivery method than the traditional text presentation for some users (Briscoe and LaMaster, 1991). Technological advances in the display of visual information further eases the creation of knowledge maps to improve learning effectiveness. A more sophisticated tool to create knowledge maps allows users to display spatial networks in multiple dimensions. The technological improvement has made knowledge maps a viable tool to facilitate individual and cooperative learning (Patterson et al., 1992). Novice trainees who have a lack of experience or knowledge on the studied subjects can also benefit significantly from the use of knowledge map
Knowledge maps have been widely applied to the educational and organizational contexts.

Although UML has been widely adopted in assisting the learning of object-oriented programming languages, knowledge maps have not been readily considered and applied in the same context. The main purpose of this study is to investigate if knowledge map-based teaching method outperforms F2F method in helping novice users acquire object-oriented and database-driven software skills. Another interest of this project is to assess if novice users can advance their understanding and skills in this target software after receiving knowledge map-based instruction. Two hypotheses were proposed to accomplish these objectives:

Hypothesis 1: Knowledge map-based training method is more effective than traditional F2F training method in helping novice users acquire object-oriented and database-driven software skills.

Hypothesis 2: Knowledge map-based training method is effective at improving object-oriented and database-driven software skills of novice users

3 RESEARCH METHODOLOGY

We conducted an experimental study with 55 subjects, divided into an experimental and a controllable group. The experimental group received the knowledge-based training, whereas the controllable group received the traditional face-to-face training. Differences in learning outcomes between these two groups were used to measure the influence of the knowledge map approach on learning outcomes, in relation to the other approach. ADO.NET encloses both object and database components. Subjects of this study were asked to acquire this programming language skill.

The exercise of randomization control helps us mitigate the self-selection bias of subjects when creating experimental and controllable groups. A pre-test was conducted to assess the degree of understandings of subjects about the tested topics. This treatment allowed us to screen out users who had experiences with ADO.NET language. Subjects were college students taking the same courses and have similar educational background. The demographics were relatively identical between subjects. The course content was designed accordingly, except the way to represent course materials was different. Two graduate assistants were instructors and facilitators in this experimental study. Both assistants have similar educational backgrounds. They received the same undergraduate degree from the same university.

Seventy nine juniors who were taking a webpage development course were invited for the participation of this experimental study. After excluding the subjects who participated in pilot study, there were fifty five valid samples. The researchers further divided these valid samples into two groups of subjects. Twenty four students participated in the knowledge map training, whereas thirty one students participated in the controllable group, the traditional training. All trainees had taken the prerequisite webpage development course and acquired knowledge and skills in HTML, SQL, and VB.NET. Most users had basic concepts about webpage development and know how to write programming languages.

Measurement tools to assess the learning outcomes in the pre-test sessions were multiple choice questions and open-ended questions. This arrangement allowed us to filter students who were knowledgeable or experienced in ADO.NET from novice students. A post-test was administered immediately right after the training session. The purpose of this test was to understand the influence of different training methods on learning outcomes. Ten multiple choice questions on ASP.NET and ADO.NET were asked in the post-test. Subjects also needed to complete a questionnaire. The internal validity and reliability of the questionnaire was assessed before they were distributed to subjects. Questions 1, 3, 8, and 9 were deliberately worded to allow an assessment of the direct influence of training methods on learning outcomes. The whole experiment lasted about three hours. The pretest lasted for an hour. The training session lasted another hour. Students could take a break and practiced for 25 minutes before participating in the post-test for forty minutes. Table 2 and 3 show the controls of experimental groups and operational procedures.
DATA ANALYSIS

SPSS 11.0 for MS Windows was used to analyze the collected data. Statistical analysis methods adopted in this study included Cronbach $\alpha$ and T test. Questionnaires were analyzed of its reliability with Cronbach $\alpha$. Cronbach $\alpha$ was an indicator to assess the reliability of a set of items (or variables) used to measure a single undimensional latent construct. The authors designate the cut-off coefficient of reliability (or consistency) to include or exclude items of the questionnaire at a Cronbach’s $\alpha$ of 0.66. Cronbach’s $\alpha$ values for the questionnaires used in the pre- and post-tests were 0.66 and 0.73, respectively. Their standardized item alpha values were 0.70 and 0.73, which show that average inter-correlations among the items were larger than the designated coefficients of reliability. Both Cronbach’s $\alpha$ values have medium to high reliability.

Validity analysis was also conducted to assess the content, face and construct validities. The prerequisite knowledge of the main topic ADO.NET is comprised of three categories: HTML, VB.NET and SQL. Students’ scores in these three categories of topics were summed up. The t-test was used to assess whether the mean scores of experimental and controllable groups in these prerequisite knowledge were statistically different from each other. A risk level (alpha level) was predetermined at 0.05.

The mean score was relatively the same for both groups. The p-values were 0.897 and 0.896 with (1) equal variances assumed, and (2) equal variances not assumed, respectively (Table 4). Both p-values were substantially higher than the pre-determined alpha value (Table 5). This indicates that these two groups were homogeneous and were not statistically different from each other.

Pre-test questions also include questions directly related to ADO.NET. Another t-test was conducted to assess the group differences in the ability to answer these pre-test questions (excluding those prerequisite questions). The mean score was 19.17 for the experimental group and 19.35 for the controllable group. The mean score was also relatively the same for both groups (Table 6). The p-values were 0.974 and 0.974 with (1) equal variances assumed, and (2) equal variances not assumed, respectively (Table 7). Both p-values were substantially higher than the pre-determined alpha value. This indicates that these two groups were homogeneous and were not statistically different from each other. The influence of prerequisite knowledge was negligible.

The indifference in personal background and prerequisite knowledge allowed us to have a better control of learning approach and investigate its influence on learning outcomes. Table 8 shows the mean scores of experimental (57.92) and controllable groups (50.97) after receiving their respective training. The group of subjects receiving knowledge map-based training performed better than that receiving the traditional F2F training. However, it is unclear if the marginal advantage of knowledge map-based training over F2F training is statistically significant. A t-test was conducted to assess the mean difference between these two groups. The p-values were 0.317 and 0.315 with (1) equal variances assumed, and (2) equal variances not assumed, respectively. P-values were improved (p=0.897 to p=0.317), in comparison with the pre-test (Table 9). However, p values were still not statistically significantly. This indicates that the experimental group of subjects, who receive knowledge map-based training, has similar performance to the controllable group of subjects. This leads to the rejection of Hypothesis 1.

A further investigation on the performance of subjects in pre- and post-tests showed that both groups improved their understanding and skills of ADO.NET after the training. The mean scores of the experimental group were improved from 19.17 to 57.92, whereas those of controllable group were improved from 19.35 to 50.97 (Table 10). The improved performance of both groups was statistically significant (p=0.000<0.05) (Table 11). A closer examination showed that post-test scores of the experimental group (57.92) were slightly higher than controllable group (50.97). This leads to the support of Hypotheses 2: Knowledge map-based training method is effective at improving object-oriented and database-driven software skills of novice users.
5 DISCUSSION

T-tests showed promising results for the difference in scores between pre- and post-tests (p<0.005). This clearly indicates that knowledge map-based training method can significantly help novice users acquire object-oriented and database-driven software skills. Some object-oriented ideas and tools that were hard to be delivered or expressed in the traditional face-to-face settings can potentially be overcome via the knowledge map-based training method. However, Hypothesis 1 was not supported even though the experimental group scored an average of 7.13 points higher than the controlled group after receiving respective training methods. The topics taught in this experiment were limited to ADO.NET of ASP.NET, a database access component of web programming skills. Future research projects may want to explore other object-oriented software areas (e.g. site design, componentization, web services, and e-commerce applications) to assess relative efficacy of knowledge map-based training method. To teach and learn object-oriented programming languages, instructors and students both need to clearly understand software design principals. The instructors of both the experimental and controllable groups were graduate students who had limited experience and understanding in software design principals. Instructors with more experiences in these areas may be able to deliver courses in a more effective manner, making the differences in learning outcomes more distinctly appealing.

6 CONCLUSION

This paper introduces knowledge map concepts as an alternative to the traditional instruction method to assist users in acquiring object-oriented and database-driven software skills. Literature shows that knowledge maps have flexibility in constructing knowledge objects in visual forms to help individuals recall studied subjects and build up new knowledge. UML is a modeling language to effectively help a programmer design and program object-oriented languages. The concept of UML resembles to that of knowledge maps, an enhanced version of concept maps. Thus, we assert that knowledge map-based training is a more effective training method than the traditional F2F training.

Our findings show that knowledge map-based training method indeed can help novice users acquire object-oriented and database-driven software skills. Yet, this training method does not seem to be superior to the F2F training method. If exercised correctly, both training methods have their adequacy in delivering object-oriented and database-driven programming skills. Knowledge map-based training method shows its potentials in this study. Future research can continue to explore this training method in different directions (e.g. subject areas, encapsulation, inheritance, layers of complexity, objects vs. classes, and experienced instructors) in response to the growing demand of object-oriented and database-driven software skills.

REFERENCES