Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 729–734
ISBN 978-83-60810-27-9
ISSN 1896-7094

Building Personalized Interfaces by Data Mining Integration

Mihăescu Marian Cristian
University of Craiova
Software Engineering Department,
Bvd. Decebal, Nr. 107, 200440,
Craiova, Dolj, Romania
Email: mihaescu@software.ucv.ro

***Abstract*—Building personalized high quality multimedia interfaces represents a great challenge. This paper presents a custom procedure of buiding personalized interfaces within e-Learning environments. The procedure has an interdisciplinary approach since the following domains are met: multimedia interfaces, data mining and e-Learning. A large variety of learners with possible very different background and goals may access an e-Learning system. This situation yields to the necessity that the interface to by dynamically build according with the current state of the learner. The business logic that decides which resources are available for the learner is based on Bayesian network learning.**

Introduction

THIS paper presents a custom procedure for building high quality personalized multimedia interface within an e-Learning environment.

E-Learning domain has received great amount of effort in last decade. E-learning represents a modern form of conducting education. The e-Learning domain developed greatly due to enormous development of Internet technologies. There are many areas in which e-Learning has progressed. One of the most important areas regard building storing and delivering e-Learning materials, assessment and monitoring of student progress, building recommender systems for learners. This paper is closely related with the last domain.

One of the main characteristics of traditional learning lays in the guidance offered by the professor to the learner. With time, professors gain experience and thus are able to guide learners according with their background and abilities. In education, this ability is highly appreciated and can make the difference in a context where learning resources are similar.

In same manner, e-Learning tries to emulate the experience and the ability of the real professor. Of course, human characteristics are very hard to be modeled and that is why the goal of the presented work is not an easy one yet very challenging.

The first step that needs to be accomplished represents setting up the input and the output. The input is represented by various types of data. The e-Learning context is represented by the e-Learning resources. This also regards the way e-Learning materials are structured. Another important input is represented by the actions performed by learners. All performed actions are important in the way that they will provide important information regarding the behavior of the learners. This will represent in a hard and the structured form the experience of the crowd. The core idea of the paper is represented by a custom representation of this data such that high quality personalized interface may be obtained. Thus, the output of the presented procedure has as output the obtained interface and more exactly a list of resources that need to be accessed. There will be obtained also a ranking of needed resources thus leading to a dynamic learning path that may be created for a certain learner. Under these circumstances the following issues need a great deal of attention: the employed methods, the e-Learning infrastructure, the input data and the analysis process itself.

The main analysis methods are Concept Maps [1, 2, 3] and Bayesian Network Learning [4, 5]. These methods are presented in second section. The e-Learning infrastructure is represented by Tesys e-Learning platform [6]. It is presented in third section along with the procedure of obtaining input data for the analysis process. Fourth section will present the analysis process in detail. Section five presents a sample experiment where real data are processed. Finally, in section six there will be presented conclusions and future works.

Analysis Methods

Concept Maps

Concept mapping may be used as a tool for understanding, collaborating, validating, and integrating curriculum content that is designed to develop specific competencies. Concept mapping, a tool originally developed to facilitate student learning by organizing key and supporting concepts into visual frameworks, can also facilitate communication among faculty and administrators about curricular structures, complex cognitive frameworks, and competency-based learning outcomes.

To validate the relationships among the competencies articulated by specialized accrediting agencies, certification boards, and professional associations, faculty may find the concept mapping tool beneficial in illustrating relationships among, approaches to, and compliance with competencies [7].

The usage of concept maps has a proper motivation. Using this approach, the responsibility for failure at school was

to be attributed exclusively to the innate (and, therefore, unalterable) intellectual capacities of the pupil. The learning/teaching process was, then, looked upon in a simplistic, linear way: the teacher transmits (and is the repository of) knowledge, while the learner is required to comply with the teacher and store the ideas being imparted [8].

Usage of concept maps may be very useful for students when starting to learn about a subject. The concept map may bring valuable general overlook of the subject for the whole period of study.

It may be advisable that a concept map should be presented to the students at the very first meeting. This will help them to have a good overview regarding what they will study.

Bayesian Networks

A Bayesian network [5] encodes the joint probability distribution of a set of v variables, $\{x_1, x_2, \ldots, x_v\}$, as a directed acyclic graph and a set of conditional probability tables (CPTs). In this paper we assume all variables are discrete. An instance is represented by a learner from the e-Learning environment. Each instance is described by a set of features which in this context represent the variables. Each node corresponds to a variable, and the CPT associated with it contains the probability of each state of the variable given every possible combination of states of its parents. The set of parents of x_i, denoted π_i, is the set of nodes with an arc to x_i in the graph. The structure of the network encodes the assertion that each node is conditionally independent of its non-descendants given its parents. Thus the probability of an arbitrary event $X = (x_1, x_2, \ldots, x_v)$ can be computed as

$$P(X) = \prod_{i=1}^{v} P(x_i|\pi_i)$$

In general, encoding the joint distribution of a set of v discrete variables requires space exponential in v; Bayesian networks reduce this to space exponential in $max_{i\in\{1,\ldots,v\}}|\pi_i|$.

Bayesian networks represent a generalization of naïve Bayesian classification. In [9] it was proved that naïve Bayes classification outperforms unrestricted Bayesian network classification for a large number of datasets. Their explanation was that the scoring functions used in standard Bayesian network learning attempt to optimize the likelihood of the entire data, rather than just the conditional likelihood of the class given the attributes. Such scoring results in suboptimal choices during the search process whenever the two functions favor differing changes to the network. The natural solution would then be to use conditional likelihood as the objective function.

That is why, when using Bayesian networks conditional independence of used variables needs a great attention.

E-Learning Infrastructure

So far, e-Learning platforms are mainly concerned with delivery and management of content (e.g. courses, quizzes, exams, etc.). An important feature that misses is represented by the intelligent characteristic. This may be achieved by embedding knowledge management techniques that will improve the learning process.

For running such a process the e-Learning infrastructure must have some characteristics. The process is designed to run at chapter level. This means a discipline needs to be partitioned into chapters. The chapter has to have assigned a concept map which may consist of about 20 concepts. Each concept has assigned a set of documents and a set of quiz questions. There are three tree documents that may be attached to each concept: overview, detailed description and examples. Each concept and each quiz has a weight, depending of its importance in the hierarchy.

Figure 1 presents a general e-Learning infrastructure for a discipline. Once a course manager has been assigned a discipline he has to set up its chapters by specifying their names and their associated concept maps. For each concept managers have the possibility of setting up three documents and one pool of questions.

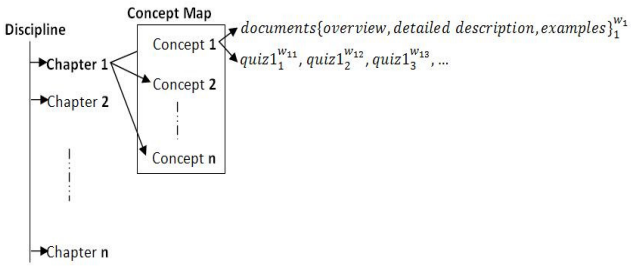

Fig. 1. General structure of a discipline

When the discipline is fully set, the learning process may start for learners. Any opening of a document and any test quiz that is taken by a learner is registered. The business logic of document retrieval tool will use this data for determining the moment when it is able to determine the document (or the documents) that are considered to need more attention from the learner. The course manager specifies the number of questions that will be randomly extracted for creating a test or an exam.

Let us suppose that for a chapter the professor created 50 test quizzes and he has set to 5 the number of quizzes that are randomly withdrawn for testing and 15 the number of quizzes that are randomly withdrawn for final exam. It means that when a student takes a test from this chapter 5 questions from the pool of test question are randomly withdrawn. When the student takes the final examination at the discipline from which the chapter is part, 15 questions are randomly withdrawn. This manner of creating tests and exams is intended to be flexible enough for the professor. This means, the professor may easily manage the test and exam questions that belong to a chapter. Also, tests and exams composition may be easily managed by professors through custom settings. The difficulty of created test and exam may be controlled with the weights that were assigned to concepts and quizzes.

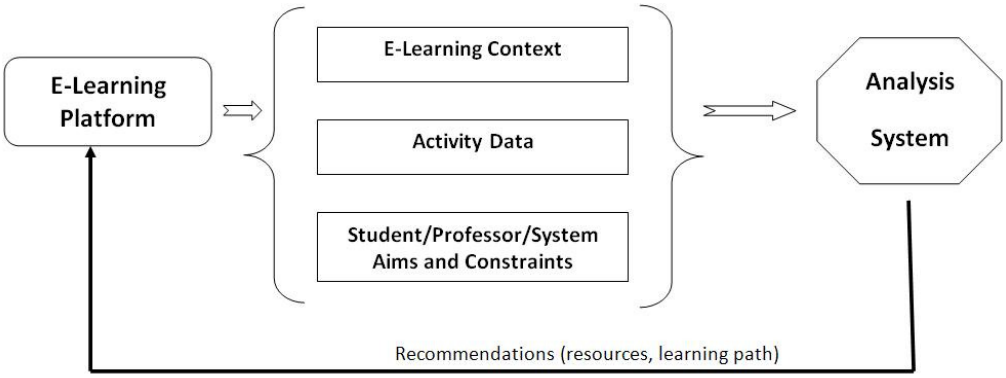

Fig. 2. General view of analysis process

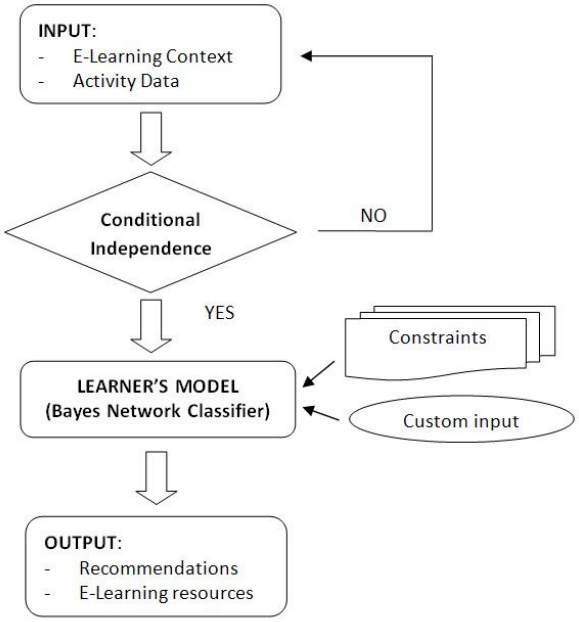

Fig. 3. Detailed view of analysis process

Analysis Process

The analysis process runs along the served e-Learning platform. The e-Learning platform is supposed to be able to provide in a standard format data regarding the context, the performed activity by learners and the aims/constraints provided by learners, professors or system administrator itself.

The e-Learning context represents the set of e-Learning resources that are available for a certain chapter of a discipline. The data that represents the context regards the concept map associated with the chapter along with resources associated to each concept or phrase from the concept map. The resources are represented by documents and quizzes as presented in section three.

The analysis system works as a service that loads the e-Learning context provided by the e-Learning platform and performs updates in a scheduled manner regarding performed activities and the constraints provided by learners, professors or administrator of the e-Learning platform. The constraints work as threshold within the analysis process.

The first step regards checking the conditional independence of attributes. If this condition does not hold than the input must be reviewed. This might mean changes regarding the attributes or even data pruning.

Once the conditional independence of attributes is met the learner's model is build. It will represent the "ground truth" against which any custom request will be evaluated. The custom input regards personal data of a certain learner. It may be regarded as the current status of the learner.

The final outcome of the analysis process is represented by the recommendations and/or a list of resources that need more attention from the learner.

The interface of the learner will be dynamically loaded with links to needed resources thus obtaining a personalized interface

Setup and Experiment

The presented experiment consists in an off-line step by step running of the analysis procedure with real data obtained from Tesys e-Learning platform.

The context has an xml representation. Below it is presented a sample of the xml file representing Computer Science program, Algorithms and Data Structures discipline, Binary Search Trees and Height Balanced Trees chapters.

```
<module>
<id>1</id>
  <name>Computer Science</name>
  <discipline>
   <id>1</id>
   <name>Algorithms and Data Structures</name>
   <chapter>
     <id>1</id>
     <name>Binary Search Trees</name>
     <concepts>
       <concept>
         <id>1</id>
         <name>BST</name>
       </concept>
       <concept>
         <id>2</id>
         <name>Node</name>
       </concept>
       ....
     </concepts>
     <quiz>
       <id>1</id>
       <text>text quiz 1</text>
       <visibleAns>abcd</visibleAns>
       <cotectAns>a</ cotectAns >
       <conceptId>1</ conceptId >
     </quiz>
     ……
   </chapter>
    <chapter>
     <id>2</id>
     <name>Height Balanced Trees</name>
   </chapter>
  ……
  </discipline>
</module>
```

It may be observed that each chapter has associated a set of concepts and each quiz has associated a certain concept.

Figure 4 presents the concept map associated with the Binary Search Tree chapter.

The data representing the activities performed by learners needs to be obtained. Firstly, the parameters that represent a learner and their possible values must be defined. For this study the parameters are: *nLogings* – the number of entries on the e-Learning platform; *nTests* – the number of tests taken by the learner; *noOfSentMessages* – the number of sent messages to professors; *chapterCoverage* – the weighted chapter coverage from the testing activities. Their computed

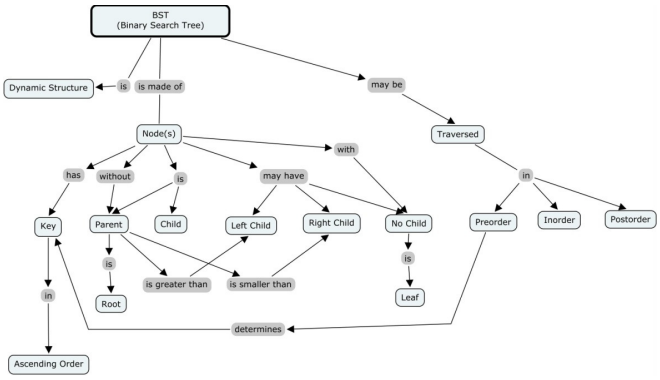

Fig. 4. Binary Search Tree Concept Map

values a scaled to one of the following possibilities: VF – very few, F – few, A – average, M – many, VM – very many. The number of attributes and their meaning has a great importance for the whole process since irrelevant attributes may degrade classification performance in sense of relevance. On the other hand, the more attributes we have the more time the algorithm will take to produce a result. Domain knowledge and of course common sense are crucial assets for obtaining relevant results.

The preparation gets data from the database and puts it into a form ready for processing of the model. Since the processing is done using custom implementation, the output of preparation step is in the form of an *arff* file. Under these circumstances, we have developed an offline Java application that queries the platform's database and crates the input data file called *activity.arff*. This process is automated and is driven by a property file in which there is specified what data/attributes will lay in *activity.arff* file.

For a student in our platform we may have a very large number of attributes. Still, in our procedure we use only four: the number of logings, the number of taken tests, the number of sent messages and the weighted chapter coverage from the testing activities. Here is how the arff file looks like:

```
@relation activity
@attribute nLogings {VF, F, A, M, VM}
@attribute nTests {VF, F, A, M, VM}
@attribute noOfSentMessages {VF, F, A, M, VM}
@attribute chapterCoverage {VF, F, A, M, VM}
@data
VF, F, A, A,
F, A, M, VM,
A, M, VM, A, V,
VM, A, VM, M,
```

As it can be seen from the definition of the attributes each of them has a set of five nominal values from which only one may be assigned. The values of the attributes are computed for each student that participates in the study and are set in the @data section of the file. For example, the first line says that the student logged in very few times, took few tests, sent an average number of messages to professors and had average chapter coverage.

In order to obtain relevant results, we pruned noisy data. We considered that students for which the number of logings, the number of taken tests or the number of sent messages is zero are not interesting for our study and degrade performance; this is the reason why all such records were deleted.

Once the dataset is obtained the conditional independence is assessed. This is necessary because the causal structure of attributes needs to be revealed. If conditional independency is identified between two variables then there will be no arrow between those two variables.

As metric regarding the conditional independence there are estimated expected utilities. This metric will specify how well a Bayesian network performs on a given dataset. Cross validation provides an out of sample evaluation method to facilitate this by repeatedly splitting the data in training and validation sets. A Bayesian network structure can be evaluated by estimating the network's parameters from the training set and the resulting Bayesian network's performance determined against the validation set. The average performance of the Bayesian network over the validation sets provides a metric for the quality of the network.

Running Bayes Net algorithm in weka [10] produced the following output:

```
=== Run information ===

Scheme:       weka.classifiers.bayes.BayesNetB -S BAYES
-A 0.5 -P 100000
Relation:     activity
Instances:    261
Attributes:   4
              nLogings
              nTests
              noOfSentMessages
              chapterCoverage
Test mode:    10-fold cross-validation
=== Classifier model (full training set) ===
Bayes Network Classifier
Using ADTree
#attributes=4 #classindex=3
Network structure (nodes followed by parents)
nLogings(5): chapterCoverage nTests
nTests(5): chapterCoverage
noOfSentMessages(5): chapterCoverage nTests
chapterCoverage(5):
LogScore Bayes: -77.14595781124575
LogScore MDL: -597.9372820270846
LogScore ENTROPY: -287.4073451362291
LogScore AIC: -511.4073451362291
-S
Time taken to build model: 0.12 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances         228               87.5    %
Incorrectly Classified Instances        33               12.5    %
Kappa statistic                          0.7881
Mean absolute error                      0.0814
Root mean squared error                  0.1909
Relative absolute error                 31.9335 %
Root relative squared error             55.2006 %
Total Number of Instances              16
=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
 0         0          0         0         0        VF
 0         0          0         0         0        F
 0.889     0.143      0.889     0.889     0.889    A
 0.75      0          1         0.75      0.857    M
 1         0.077      0.75      1         0.857    VM
=== Confusion Matrix ===
  a   b   c   d   e   <-- classified as
  0   0   0   0   0    | a = VF
  0   0   0   0   0    | b = F
  0   0   150 0   15   | c = A
  0   0   18  50  0    | d = M
  0   0   0   0   28   | e = VM
```

The Bayesian network obtained in weka has the following graph.

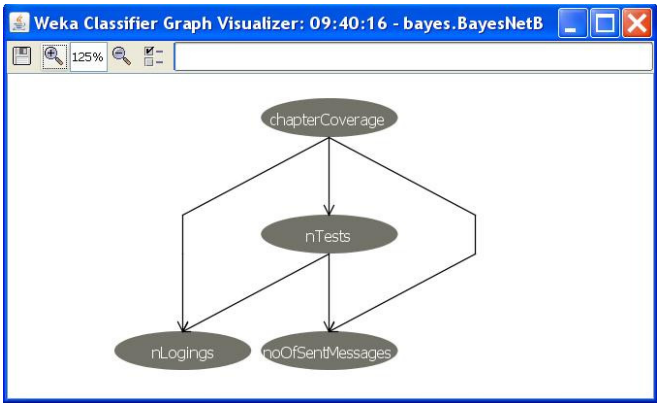

Fig. 5. Detailed view of analysis process

As it can be seen in above figure the chapter coverage is the variable with greatest conditional dependence towards all other variables. On the other hand, variables *nLogings* and *noOfSentMessages* are conditional independent which means they need to be used in further developments.

Once the Bayes Net has been obtained it may be used for obtaining the items that compose the interface for the learner. The procedure *findItems* determines the needed resources.

```
Items procedure findItems (LearnerModel LM, Constr-
taints CS, Lerner l) {
Class C = classify (l,LM);
Class D = findClass (LM, C, CS);
Items items = determineItems(C, D);
return items;
}
```

Firstly, the learner is classified against the current learner model. Thus, the actual class to which the learner belongs is determined. Secondly, the destination class D is determined taking into consideration the current learner model, the class of the learner and the constraints set up by system professor or learner himself. Finally there is determined the set of items that need to be accessed by learner by analyzing classes C and D. As general idea, there are determined the items where class D is better representation than in class C. Such a metric may also rank the resources. Firstly, there are presented the resources with smaller distance between classes. It is supposed that these resources need immediate attention from the learner.

Conclusions and Future Works

This paper presents custom data analysis process which has as main outcome obtaining a personalized interface for an e-Learning platform.

The main inputs of the process are: the context of the platform, the activity data, the constraints of the involved parties and data regarding the learner for which the personalized interface is built.

The activity data managed by the analysis process is represented by actions performed by learners within the e-Learning environment. From the great variety of performed actions there were taken into consideration only four: the number of entries on the e-Learning platform, the number of tests taken by the learner, the number of sent messages to professors and the weighted chapter coverage from the testing activities.

The business logic uses Bayes Network Classifier implemented in weka for building the learner's model against which any learner is classified. For obtaining sound classification results the conditional independence is verified.

Once the conditional independence is met there may be started the procedure for obtaining the items that will be recommended. The procedure classifies the learner, finds the destination class and determines the items. Each item represents a resource (document or quiz) that needs attention from the learner.

As future works, there are some issues that need to be addressed. One issue regards the conditional independence assessment of variables. When this condition is not met the procedure for data pruning and feature selection may need improvement.

Another issue regards the granularity with which items are obtained by *findItems* procedure. Optimization of complexity calculus for determining the destination class and especially the set of items is needed.

Acknowledgment

This work was supported by the strategic grant POSDRU/89/1.5/S/61968, Project ID61968 (2009), co-financed by the European Social Fund within the Sectorial Operational Program Human Resources Development 2007–2013.

References

[1] Novak, J. D., Learning, Creating, and Using Knowledge: Concept Maps as Facilitative Tools in Schools and Corporations. Mahwah, NJ: Lawrence Erlbaum Associates, 1998.

[2] McDaniel,E., Roth, B., and Miller, M. "Concept Mapping as a Tool for Curriculum Design", Issues in Informing Science and Information Technology.

[3] Vecchia, L., Pedroni, M., Concept Maps as a Learning Assessment Tool. Issues in Informing Science and Information Technology, Volume 4, 2007.

[4] D. Heckerman. A tutorial on learning with bayesian networks. In M. Jordan, editor, Learning in Graphical Models. MIT Press, Cambridge, MA, 1999.

[5] Pearl, J., Probabilistic reasoning in intelligent systems: Networks of plausible inference. San Francisco, CA: Morgan Kaufmann, 1988.

[6] Burdescu, D.D., Mihăescu, M.C., 2006. Tesys: e-Learning Application Built on a Web Platform. In Proceedings of International Joint Conference on e-Business and Telecommunications, pp. 315-318, Setubal, Portugal. INSTICC Press.

[7] MAC (2010), http://mac.concord.org

[8] Kolodner, J. L., Camp, P. J., Crismond, D., Fasse, B., Gray, J., Holbrook, J., Puntambekar, S., and Ryan, M. Problem-based learning meets case-based reasoning in the middle-school science classroom: Putting learning by design into practice, The Journal of the Learning Sciences, 12 (4), 2003, 495-547.

[9] Friedman, N., Geiger, D., & Goldszmidt, M., Bayesian network classiers, Machine Learning, 29, 131-163, 1997.

[10] Weka (2010), www.cs.waikato.ac.nz/ml/weka