



Recommender system in collaborative learning environment using an influence diagram



Antonio R. Anaya*, Manuel Luque, Tomás García-Saiz

Dept. of Artificial Intelligence, UNED, Juan del Rosal, 16, 28040 Madrid, Spain

ARTICLE INFO

Keywords:

Recommender systems
Probabilistic graphical models
Collaborative learning
Data mining
Machine learning
e-Learning

ABSTRACT

Giving useful recommendations to students to improve collaboration in a learning experience requires tracking and analyzing student team interactions, identifying the problems and the target student. Previously, we proposed an approach to track students and assess their collaboration, but it did not perform any decision analysis to choose a recommendation for the student. In this paper, we propose an influence diagram, which includes the observable variables relevant for assessing collaboration, and the variable representing whether the student collaborates or not. We have analyzed the influence diagram with two machine learning techniques: an attribute selector, indicating the most important attributes that the model uses to recommend, and a decision tree algorithm revealing four different scenarios of recommendation. These analyses provide two useful outputs: (a) an automatic recommender, which can warn of problematic circumstances, and (b) a pedagogical support system (decision tree) that provides a visual explanation of the recommendation suggested.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

Nowadays educational institutions support students with e-learning environments, where collaboration is possible and advisable (Swan, Shen, & Hiltz, 2006). Giving recommendations in these environments has not been studied in depth by the research community (Caballé, Daradoumis, Xhafa, & Juan, 2011), because frequent and regular analysis of student interactions is required to know whether collaboration takes place (Johnson & Johnson, 2004; Swan et al., 2006).

However, once students have been tracked and their collaboration has been assessed (Anaya & Boticario, 2011b), two questions arise: Who will be recommended? and What will be recommended? Both questions must be answered in accordance with the student circumstances, which will be known when their tracking and assessments are compared with those of all the students.

Thus, our main objective is to build a system that analyzes student tracking and collaboration assessments in the context of collaborative learning in an e-learning environment to identify the student circumstances and proposes a personal recommendation to a target student. Once the circumstances are known, the collaboration problem identified, and the decision resolved, the student circumstances support and explain the recommendation.

For the development of the system, we used the tracking data and assessment from previous research (Anaya & Boticario, 2011a). This previous research analyzed student interactions in different collaborative learning experiences in the academic years 2006–07, 2007–08, 2008–09, 2010–11, and 2011–12. We proposed two data mining (DM) approaches to assess student collaboration. In addition, an expert-based analysis classified students according to their collaboration.

Owing to the nature of collaborative learning, whose descriptive variables depend on social and psychological student behavior (Johnson & Johnson, 2004), which is out of control in a typical e-learning environment, and the nature of collaboration as a prototypical category (Rosch, 1978), the collaboration analysis must deal with uncertainty and probability. To deal with the uncertainty problems other researchers have proposed Bayesian networks (BNs). BNs (Pearl, 1988) constitute a graphical framework that can easily be understood by a human being, where independence and dependence probabilistic relations among the variables are easily represented (Lacave & Díez, 2002). Moreover, there exist software tools for building and debugging BNs and assisting the knowledge engineer and the expert during the process. These reasons, as well as the fact that BNs can combine statistical data and subjective estimated probabilities when the model is being built, make BNs an ideal framework for accomplishing our system.

However, our objective not only consists of analyzing student tracking and collaboration assessments, but also of finding a personal recommendation for a target student. That means we have a decision problem, and we therefore need a decision support

* Corresponding author. Tel.: +34 913986550.

E-mail addresses: arodriguez@dia.uned.es (A.R. Anaya), mluque@dia.uned.es (M. Luque), tomasgs@dia.uned.es (T. García-Saiz).

system. Influence diagrams (IDs) (Howard & Matheson, 1984) are usually used as decision support systems when the probabilistic relations of the variables are represented by a BN. Thus, we created an ID, whose nodes are the attributes that tracked student interactions and assessed student collaboration, and the arcs were chosen according to the cause-effect and probabilistic relations among the collaboration variable and attributes selected. The probabilities were automatically learned from the dataset recording the student tracking and assessments, and the classification labels were obtained from the expert's analysis.

When the ID was built, an evaluation algorithm was used to compute an optimal policy for the decision variable. The policy took the form of a table containing a tuple for each combination of attribute values observed and it provided a recommended action for each tuple. Thanks to this table, an automatic recommender can send a warning to a target student or teacher to indicate that a collaboration problem has been detected.

The policy table was analyzed using two different machine learning (ML) techniques. On the one hand, the ID attributes were checked to select the most informative attributes that warn of the collaborative learning problems. On the other hand, a decision tree (DT) algorithm was applied to the decision table to obtain a visual explanation of the decision. An expert system used as an intelligent tutor must be able to communicate the knowledge that it contains to the learner, how the knowledge has been applied to reach a conclusion, and what would have happened if the observed evidence had been different (what-if reasoning) (Lacave, Luque, & Díez, 2007). The DT obtained offers a representation of the policy understandable to the expert by providing an explanation of the reasoning followed by the system. The expert studied the DT recommendations, identified four clearly distinguishable recommendation scenarios and concluded that the system reasoning is consistent with his knowledge.

In this paper we propose an original recommender system approach in the context of collaborative learning. We have extended our previous research on analyzing and assessing student collaboration with an ID. The output of ID assessment is a decision table, which explains, after an analysis, the circumstances to recommend. Finally, this analysis provides the teacher and students with a user-friendly explanation that may help them to correct deficiencies in the collaboration process, thereby increasing their confidence and improving their learning.

Following we describe the preliminaries of our research. For this reason in Section 2.1 we explain other research whose aims were collaboration analysis. In Section 2.2 we give a brief overview of probabilistic graphical models (PGMs), starting with BNs and continuing with IDs. Next the construction of our system, the ID (selection of variables, construction of the network and elicitation of the probabilities and utilities), is explained in Section 3. The paper then reports the inference performed on the ID and the subsequent inductive learning conducted in Section 4. We evaluate the results obtained and discuss the applications as a pedagogical tool in Section 5. Finally, we explain the main findings and discuss future improvements.

2. Preliminaries

2.1. Collaboration Analysis Approaches

Before the recommendation is offered, the system should analyze the context and behavior of the target person to select and create a personalized recommendation. Thus in collaborative learning, a collaboration analysis should be performed (Gaudioso, Montero, Talavera, & del Olmo, 2009; Johnson & Johnson, 2004). However, there is still no consensus on how to proceed because standards

and comparative studies are insufficient (Strijbos & Fischer, 2007). In the following paragraph we summarize the different collaboration analysis approaches. We focused on data mining (DM) techniques, which have been used to discover knowledge that is hidden in student interactions (Romero & Ventura, 2010). Following the classification provided by Romero and Ventura (2010), a generic DM approach can be divided into three parts: the data acquisition method, the inferring method, and how the approach results are used.

Acquisition methods include: (1) asking students and teachers for assessment of student collaboration actions (Collazos et al., 2007; Park & Hyun, 2006), (2) storing student interactions in the e-learning platform (Duque & Bravo, 2007; Gaudioso et al., 2009; Perera, Kay, Yacef, & Koprinska, 2007), and (3) combining the former two methods (Bratitsis, Dimitracopoulou, Martínez-Mónes, Marcos-García, & Dimitriadis, 2008; Daradoumis, Martínez-Mónes, & Xhafa, 2006).

There are also differences in the inferring methods. Either students or teachers were in charge of assessing student collaboration, asking directly (Kahrimanis et al., 2009), or showing a visual tool to help them in the assessment process (Bratitsis et al., 2008; Daradoumis et al., 2006). Duque and Bravo (2007) analyzed student collaboration without any human intervention. They proposed a collaboration model intended to be the best way of collaborating, and this model was compared with student interaction models by means of fuzzy logic. Gaudioso et al. (2009) and Perera et al. (2007) proposed inferring methods that assessed collaboration with ML techniques.

There are three strategies to analyze students interactions and assessments to improve the learning (Soller, Martinez, Jermann, & Muehlenbrock, 2005): monitoring tools, metacognitive tools and guide systems. First, Bratitsis et al. (2008), Daradoumis et al. (2006) and Kahrimanis et al. (2009) offered *monitoring tools* that showed data collected from student interactions. These monitoring tools displayed information on student behavior to students and teachers, depending on the approach. Second, beyond monitoring, analyzing student interactions depends on desired outcomes. Duque and Bravo (2007), Gaudioso et al. (2009) and Perera et al. (2007) collected data from student interactions, analyzed these data and obtained student assessments. They proposed different *metacognitive tools* (Soller et al., 2005), which displayed the information collected and the assessments to students or teachers. Baghaei and Mitrovic (2007) and Caballé et al. (2011) proposed a *guide or recommendation system*. Baghaei and Mitrovic (2007) established a constraint-based collaboration model. When constraints were broken, a pre-programmed recommendation was sent to the student. Thus, student collaboration was not analyzed and a specific recommendation model had to be created again by means of a collaboration analysis process in a different environment. Caballé et al. (2011) proposed a conceptual framework that tracked the students with a quantitative method, which counted and classified the interactions in forums, and a qualitative method, which asked the students to label their messages, and the teacher to assess the student actions. Then the results were compared with a model to classify the performance and to select a recommendation. Both researchers proposed a collaboration model before students act, which makes a dynamic and personalized recommendation difficult.

2.2. Probabilistic Graphical Models

Probabilistic Graphical Models (PGMs), in particular BNs and IDs, were developed in the 1980s by Artificial Intelligence researchers with the purpose of solving problems with uncertainty.

BNs (Pearl, 1988) constitute a modeling framework where cause-effect and probabilistic dependence and independence

relations are easily represented. BNs have been widely used in medical domains for performing diagnosis tasks, but they have also been applied to computing, education, natural sciences and engineering. One important feature of BNs is that they enable easy communication with the expert when the model is being built and debugged (Lacave & Díez, 2002).

IDs (Howard & Matheson, 1984) are an extension of BNs and include decision variables and the preferences of the decision maker. IDs, prior to BNs, have benefited from the advances in research on BNs. Some of the main ID applications have been applied to medicine. IDs allow the expert not only to perform diagnosis tasks as in BNs, but they also provide very interesting possibilities (Bielza, Gómez, & Shenoy, 2011): (1) to know which is the best action in each scenario, (2) to obtain automatically explanations of the system reasoning, and (3) to formulate queries about the posterior probabilities and expected utilities of some variables of the model.

The educational context is a suitable field to use BNs to deal with the uncertainty inherent in the context problems. The most common application domain of BNs in education has been focused on student modeling (Conati, Gertner, VanLehn, & Druzdzel, 1997; Conati, Gertner, & VanLehn, 2002; Millán & Pérez-De-La-Cruz, 2002; Millán, Loboda, & de-la Cruz, 2010; Reye, 1996). Moreover, some proposals have been applied to aspects of student modeling as learning styles (García, Amandi, Schiaffino, & Campo, 2007) or knowledge level identification (Mislevy & Gitomer, 1996). Chang (2003) proposed a course diagram method, based on an ID framework, which can be used by an instructor to design a course structure. The diagram organizes the instructional material and the tests. However, IDs have not been widely used as recommender systems in an educational context.

3. Construction of the system

Before we describe the system, we must outline the collaboration learning experience, which was offered to students during five academic years (2006–07, 2007–08, 2008–09, 2010–11, 2011–12). The interaction data were obtained from this collaboration learning experience to analyze student collaboration and the system can be applied to these data. The collaborative learning experience is divided into two phases. In the first phase the students have to do an individual task, and a learning platform is offered to students so that they can work and communicate with one another. In the second phase the students that have done the task in the previous phase are grouped into three-member teams. In this phase a private space in the learning platform is offered to every team. In this private space enough services are supported to work and collaborate (forums, news, task manager, FAQs, etc.). The team members have to do five collaborative tasks and can use the forums in the private space to manage the teamwork. In previous papers the collaborative learning experience structure is described in depth (Anaya & Boticario, 2011a, 2011b). Furthermore, in these papers we proposed an approach to analyze student collaboration using indicators of student interactions in the forums of the teams' private spaces. Some of these indicators are used in this research and they are explained in the following section.

The proposed approach tackled student interactions in the collaborative learning experience to analyze collaboration frequently and regularly. However, real student collaboration is unknown. This feature is one that can be deduced by comparing the behavior of one student with another who has a prototype collaboration behavior following the theory of categorization by Rosch (1978). Our previous research compared one student's interactions with all the other student interactions to assess collaboration. However it is known that some variables, which take part in a learning process, depend on the student context (social, psychological)

(Johnson & Johnson, 2004) and all of these variables are thereby not under the control of the researchers. Thus, we have to deal with uncertainty in the collaboration process because some unknown or hidden variables modify student behavior and, hence their interactions, which can be tracked over the collaboration process.

In Section 2.2 we talked about PGMs and we mentioned some applications in education. PGMs are always applied in contexts characterized by the presence of uncertainty, that is, some variables are unknown or hidden, or their values are imprecise. We have to deal with a similar context in the collaboration problem, because we cannot know all the variables that influence student collaboration due to its nature according to Rosch (1978). However, we have tracked student interactions and analyzed their collaboration, which enables us to perform a diagnostic task on our collaboration problem. Variables influencing the diagnosis (the collaboration variable and the attributes that tracked the student interactions) are probabilistically related, and the expert knows the structure of their relations. We therefore have an ideal problem to be represented as a BN. However, our objective is not only to diagnose the collaboration variable, but also to identify the best personal recommendation for a target student. Thus there is a decision problem, and it is therefore more appropriate to represent it as an ID, which is an extension of a BN, as we noted in Section 2.2.

3.1. Identification of the variables

The variables of our system are the indicators of student interactions in forums and student collaboration assessments. Since the students worked together asynchronously in a learning management system, in our previous research we tracked their interactions in the forums. From the interactions we obtained statistical indicators that were used to infer student collaboration assessments. The statistical indicators were explained in depth in Anaya and Boticario (2011b). Table 1 shows the statistical indicators used in this research.

We proposed a set of statistical indicators to track student initiative, activity and regularity, and acknowledgment by their peers. These statistical indicators are shown in Table 1. The indicators in the first row refer to student initiative (N_{thrd}) and the regularity of the initiative (L_{thrd}), the indicators in the second row to activity (N_{msg}) and the regularity of the activity (L_{msg}), and the indicators in the third row to student acknowledgment (N_{r_thrd}, N_{r_msg}) by their peers. We calculated the indicators N_{thrd} and N_{msg} , which are connected to a period of time (one day), to obtain their variances that are related to the regularity. If all students had the same value of N_{thrd} or N_{msg} , lower variance values would indicate higher regularity. Since students generally had different values of N_{thrd} or N_{msg} , we proposed an additional indicator L_{thrd} or L_{msg} , which related student initiative and activity to student regularity. In the case of the indicators measuring the replies to student interactions, acknowledgment can be measured by the replies to student initiative N_{r_thrd} or activity N_{r_msg} . These indicators were chosen due to their understandability and the fact that they are relevant for the problem that we are considering.

In addition to the statistical indicators above, an expert did an analysis. An expert, who was the tutor of the collaborative learning experiences, read the forum messages in the private spaces of all the students who participated in the collaborative learning experiences, and labeled students according to their collaboration into three labels (high, medium, low). Note that the expert was no longer needed once the inferring methods were validated (Anaya & Boticario, 2011b).

We also proposed two DM approaches, which used the statistical indicators, to assess student collaboration. The different approaches were: the *Clustering Approach* and *Metric Approach* (Anaya & Boticario, 2011b).

Table 1Quantitative statistical indicators of the student interactions in forums. Parameter n is the number of days in the experience.

Forum conversations started	$N_{thrd} = \sum_{i=1}^n x_i$, where x_i is the number of threads started on day i	$L_{thrd} = N_{thrd} / \sqrt{\text{Variance}(N_{thrd})}$
Forum messages sent	$N_{msg} = \sum_{i=1}^n x_i$, where x_i is the number of messages sent on day i	$L_{msg} = N_{msg} / \sqrt{\text{Variance}(N_{msg})}$
Replies to student interactions	N_{r_thrd} is the number of messages in the thread started by the user	N_{r_msg} is the number of replies to messages sent by the user

The *Clustering Approach* consisted of: (1) Building datasets with the statistical indicators of student interactions in forums for every experience; (2) Running the EM clustering algorithm with every dataset to group the instances into three final clusters; (3) Comparing the clusters obtained with the expert-based analysis. Once we had done the previous steps, we noted that one cluster grouped students with low collaboration, another cluster with medium collaboration and the last cluster with high collaboration. Thus the cluster algorithms grouped students according to their collaboration.

The *Metric Approach* is basically an ML-based method that provides a set of indicators related to student collaboration. The Metric Approach includes: (1) Datasets with the student quantitative statistical indicators and the collaboration label; (2) A set of nine different DT algorithms for neglecting the individual bias, which were trained to classify students according to the student collaboration label; (3) Counting the number of algorithms that use a certain indicator in their DT for each of the datasets to select the indicators most related to student collaboration classification. We found that the most related indicators were L_{thrd} , L_{msg} and N_{r_msg} ; (4) A metric. This mathematical formula was built from the statistical indicators that were most related to student collaboration. The metric offered a numerical value related to student collaboration.

Finally, the variables corresponding to the selected indicators are: related to the student initiative (N_{thrd}); the regularity of the initiative (L_{thrd}); student activity (N_{msg}); regularity of the activity (L_{msg}); student acknowledgment (N_{r_thrd} and N_{r_msg}). Another two variables that give information about the global collaboration, which were inferred by the DM approaches, are: collaboration level (*Level*); and collaboration Metric (*Metric*). The expert-based analysis collaboration label is represented by variable *Collaboration*. This variable is a hidden attribute, unknown during the experience, and its value is only assessed by the expert after the collaborative learning experience has finished. However, during the experience the values of the aforementioned variables are known, and they are influenced by the variable *Collaboration*, which will allow the ID to infer indirectly the probabilities of the variable *Collaboration*. The decision of the problem is represented by variable *D*. It indicates if an intervention is necessary or not to solve the collaboration issues detected by the system. The preferences of the decision maker are represented by the utility variable *U*.

3.2. Domain of the variables

We had to determine the possible values for each variable. Chance variables of our problem can be interpreted as continuous quantities, but we had to discretize them because the software tool used for the construction and inference only admitted discrete variables.

The level of granularity when discretizing the chance variables was intended to find an equilibrium among (1) the precision of the model, (2) the difficulty of the parameter elicitation and the interpretation of the results, and (3) the understandability of the values by someone who was not an expert. All chance variables are ternary, with high, medium and low values, and these values were assigned by splitting the numerical scale of the possible values of the variable into three intervals.

Decision variable *D* is discrete, with two possible states (*yes* and *no*) corresponding to the actions that can be chosen by the decision maker. Value *yes* of *D* indicates that an action must be taken to solve the collaboration problem, while value *no* of *D* indicates no action is required. The utility variable *U* is continuous, and it is assumed to take values ranging from 0 to 100 (see Section 3.4.2). This range of *U* was chosen because it is easily understandable by the tutor of the collaborative learning experience and therefore it facilitates him/her to estimate the utility parameters.

3.3. Structure of the influence diagram

Each variable identified in the problem, called the indicator above, is represented with a node in the ID graph (see Fig. 1). The structure of the graph was built by following the expert-s knowledge of the cause-effect and probabilistic dependence and independence relations among the variables.

The node *Collaboration* is the target variable in the graph (Millán et al., 2010). Nodes representing statistical indicators (N_{thrd} , L_{thrd} , N_{msg} , L_{msg} , N_{r_thrd} and N_{r_msg}) and assessments indicators (*Level* and *Metric*) have a causal and probabilistic relationship with node *Collaboration*. This justifies the arcs pointing from *Collaboration*.

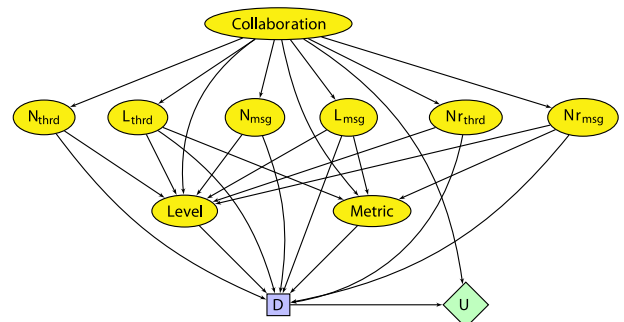
Moreover, nodes *Level* and *Metric* also depend probabilistically on other indicator nodes. For example, variable *Metric* depends on L_{thrd} , L_{msg} , and N_{r_msg} (and obviously on *Collaboration*). In this example, it is reflected in the graph by drawing the corresponding arcs to the node *Metric*.

Attribute *D* is the only decision in the diagram. Given that variable *Collaboration* is not observable, this variable is unknown when making decision *D* and there is thus no arc from the node *Collaboration* to node *D*. However, variables N_{thrd} , L_{thrd} , N_{msg} , L_{msg} , N_{r_thrd} , N_{r_msg} , *Level* and *Metric* are observable, their values are known when making decision *D*, and they can therefore be observed by the decision maker at that moment. This explains the arcs pointing from those nodes to the node *D*.

Finally, the parents of only utility node *U* in the diagram, are *Collaboration* and *D*; they correspond to the variables defining the domain of the utility function *U*.

3.4. Elicitation of numerical parameters

In order to obtain the ID model it is necessary to complete its quantitative part, which consists of a set of probability and utility

**Fig. 1.** Influence diagram for the collaboration problem.

potentials.¹ For example, for chance node *Metric* we must give a conditional probability potential $p(\text{Metric}|pa(\text{Metric}))$ for each configuration of its parents, $pa(\text{Metric})$, which are the nodes $L_{thrd}, L_{msg}, N_{r_msg}$ and *Collaboration*. This table requires 243 numbers. However, given the restriction that for any tuple of values of variables in $pa(\text{Metric})$ the sum of the probabilities of the values of *Metric* are equal to 1, only some of them are independent (162 in this example).

In our diagram, every chance variable has required the elicitation of six probability parameters, except the following variables: *Collaboration*(2), *Metric*(162), and *Level*(4374). It is remarkable that while the number of parameters elicited in the ID is high (4436), most of them correspond to the probability table of variable *Level*.

Next we explain the elicitation of numerical parameters, which was divided into two phases. First, probabilities were learned, and second, the utilities were elicited.

3.4.1. Learning probabilities

The numerical parameters of probability tables were automatically learned from a data base case file. This file is a comma separated values (CSV) file containing a row for each registry, where each column indicates a variable. The probability parameters were obtained by using the learning algorithms available in OpenMarkov (Arias & Díez, 2008), an open source software tool for PGMs. The data base case file did not contain all the necessary registries for the computation of the entire set of parameters of conditional probabilities, and so it was necessary to use Laplace's correction (Niblett, 1987) to alleviate the problem.

3.4.2. Elicitation of the utilities

The system recommendations should benefit students and their collaboration process. We have created a collaboration model that is represented with the ID network (see Fig. 1). In this model there are three clear scenarios: when a student has high, medium or low collaboration. To elicitate the utilities we took into account the three scenarios of student collaboration and the advice given by the tutor of the collaborative learning experience. The scale of utilities in this research was selected by the tutor to make the values easily and intuitively comprehensible and understandable. Thus, he chose a numeric scale from 0 to 100, and, therefore, each utility value can be interpreted as a percentage. The tutor assigned a utility value to each scenario indicating how desirable the scenario is according to the objective of collaboration. Numerical values of utilities were subjectively estimated by the tutor by using his expertise in collaborative learning experiences.

- If student collaboration is high, the recommendations should not bother the student, because s/he is doing good work, but should sometimes encourage. For this reason we proposed utility elicitations in this instance as: 90 not recommend and 10 recommend. These values means that the system should not recommend in the most cases but recommend only sometimes.
- If student collaboration is medium, the recommendation should inform the student without bothering him/her that the circumstances are not bad but they could become worse. Thus, we proposed 40 not recommend and 60 recommend. These values means that the system should recommend a bit more than the half of times.
- If student collaboration is low, the recommendation should encourage students to interact more. We proposed 5 not recommend and 100 recommend. The objective is always to recommend in order to solve the collaboration problem detected.

However, because each inference has an inherent error, the tutor proposed a value of 5 not to recommend taking into account the uncertainty.

The elicitations of the utilities can be checked by comparing the attributes of student interactions and collaboration assessments before and after the recommendation. Therefore, we can consider that one recommendation is useful when the interactions increase or these interactions are more balanced, that is, they are similar to each other.

4. Inference

4.1. Computation of the optimal policy

One objective of the construction of our ID is to compute an optimal policy for the decision variable *D*. The optimal policy in software tools for PGMs is usually represented as a table containing a tuple for each combination of the values of the variables observed in the decision. More precisely, in our model, the variables observed in *D* are $N_{thrd}, L_{thrd}, N_{msg}, L_{msg}, N_{r_thrd}, N_{r_msg}, \text{Metric}$ and *Level*. For each tuple the policy for *D* prescribes the best action (yes or no).

However, the presentation of the action prescribed by the policy table can differ slightly from one software tool to another. In our case, the ID was edited and inference was performed with OpenMarkov. OpenMarkov does not explicitly prescribe the best action; it considers instead that several actions may be optimal in the same scenario. Thus OpenMarkov rates these actions with the same value.

For example, Fig. 2 displays a part of the optimal policy for the decision of our ID. The first eight rows enumerate the possible combination of variables for the variance observed when deciding on *D*. The last two rows indicate which action is optimal when the value is 1.0 (the cell is also in red); when both actions are optimal, they are assigned to 0.5 values.

Observing the scroll bar in the dialog of Fig. 2 we can imagine how huge the policy table is. A person could try to look for any combination of values of observed variables to find out the optimal action. However, that would be unfeasible in most cases, because the table contains 19683 tuples and the process of looking for a tuple would be very tedious.

Moreover, given the huge size of the table, it would also be quite difficult for the expert, the teacher, to try to find any pattern in the table in order to understand the policy.

Accordingly, we need to synthesize the policy and present it to the expert in a more compact and graphical way, as we see in the next section.

4.2. Post-analysis of the strategy

When the ID has been built and its decision table has been calculated, we can use the student attribute values, which we call student circumstances, to identify the decision proposed by the system: value yes means the system recommends the student to collaborate, and value no means it does not recommend any change in student behavior. In order to gain acceptance by the expert, the teacher performed two tasks with the decision table. However, before doing this, we first eliminated from the decision table the cases in which OpenMarkov was indifferent as to which is the best action for *D* (indicated with 0.5 value in the policy, as seen in Fig. 2). The table obtained was 4452 tuples in size, which is considerably smaller than the initial size of 19683 tuples.

The first task consisted of the expert analyzing the decision table to discover the most relevant attributes in the decision.

¹ A potential is a real-valued function over a domain of finite variables.

Metric	Low	Low	Low	Low	Low	Low	Low	Low	Low	Low	Low	Low
Nrt...	Low	Low	Low	Low	Low	Low	Low	Low	Low	Low	Low	Low
Nthrd	Low	Low	Low	Low	Low	Low	Low	Low	Low	Low	Low	Low
Nrm...	Low	Low	Low	Low	Low	Low	Low	Low	Low	Low	Low	Low
Level	Low	Low	Low	Low	Low	Low	Low	Low	Low	Low	Low	Low
Lthrd	Low	Low	Low	Low	Low	Low	Low	Low	Low	Medium	Medium	Medium
Lmsg	Low	Low	Low	Medium	Medium	Medium	High	High	High	Low	Low	Low
Nmsg	Low	Medium	High	Low	Medium	High	Low	Medium	High	Low	Medium	High
yes	1	1	1	1	1	0	1	1	1	1	1	0
no	0	0	0	0	0	1	0	0	0	0	0	1

Fig. 2. Policy for the decision of the influence diagram.

```
Attribute selection output
```

```
=== Run information ===
```

```
Evaluator:      weka.attributeSelection.CfsSubsetEval
Search:weka.attributeSelection.GreedyStepwise -R -T -1.7976931348623157E308 -N -1
Relation:      Collaboration-Infer-Observ-Coll
Instances:     4452
Attributes:    9
               Nthrd
               Metric
               Nrmsg
               Lthrd
               Nmsg
               Level
               Nrthrd
               Lmsg
               D
Evaluation mode:10-fold cross-validation
```

```
=== Attribute selection 10 fold cross-validation (stratified), seed: 1 ===
```

average merit	average rank	attribute
0.121 +- 0.002	1 +- 0	3 Nrmsg
0.129 +- 0.001	2 +- 0	2 Metric
0.121 +- 0.001	3 +- 0	5 Nmsg
0.116 +- 0.001	4 +- 0	7 Nrthrd
0.109 +- 0.003	5.2 +- 0.4	8 Lmsg
0.106 +- 0.002	5.8 +- 0.4	4 Lthrd
0.099 +- 0.001	7 +- 0	1 Nthrd
0.093 +- 0.001	8 +- 0	6 Level

Fig. 3. Attribute evaluation of the decision table with CfsSubsetEval.

Second, a graphical representation of the policy had to be built to represent visually the student circumstances for recommendation, which would allow the expert, and also the students, to better understand the recommendation problem. The representation would provide the expert with a general vision of the recommendation scenarios and students would have more confidence in the system, which would therefore increase the probabilities of the user following the system's recommendation.

4.2.1. Analysis/ Selection of attributes

The expert needs to know which attributes are more relevant in the decision table obtained as output of the ID. For this reason, we need to analyze the importance of the attributes when deciding to recommend or not. Weka² offers a complete set of algorithms that can infer the most relevant attributes to learn one class. In our case class *D* was the policy decided for recommendation (yes or no) and the attributes were the six observable attributes and the two inferred attributes (Metric and Level). The only constraint that we introduced in this analysis was that the statistical algorithm had to support a quantitative estimator of importance of each attribute. For this reason we used only 10 attribute evaluators: CfsSubsetEval, ChiSquaredAttributeEval, ConsistencySubsetEval, GainRatioAttributeEval, InfoGainAttributeEval, OneRAttributeEval, ReliefAttributeEval,

SymmetricalUncertAttributeEval, WrapperSubsetEval with the classifier NaiveBayes, WrapperSubsetEval with the classifier J4.8. These attribute evaluators support with the quantitative estimations: *average merit* and *average rank* for each attribute. Fig. 3 shows an example of the results offered by the attribute evaluator CfsSubsetEval.

This process was repeated with the 10 attribute evaluators and we obtained 10 values of *average merit* and *average rank* for each attribute. Table 2 shows the average of these quantitative estimations.

The attributes in Table 2 are ordered according to the *Merit* column descending from the maximum relation to the class (1 value) to the minimum (0 value). The column *Rank* scale ranges from 1 (maximum relation to the class) to 8 (minimum). The first attribute, *Nrmsg*, is the attribute most related to class *D* and the last attribute, *Level* is the least related attribute to the class according to the quantitative estimator *Merit*. Three cells in the *Rank* column are highlighted in bold face. These cells show that the quantitative estimators *Rank* and *Merit* do not always agree, although the differences should be negligible.

However, Table 2 reveals that:

1. The number of replies (*N_{rmsg}*), which we have considered as an indicator of user acknowledgment by their peers, is more relevant than the activity (*N_{msg}*, *L_{msg}*) or initiative (*N_{thrd}*, *L_{thrd}*) indicators to recommend. As the proposed decision strategy was focused on sending a recommendation when collaboration

² <http://www.cs.waikato.ac.nz/ml/weka/>.

Table 2

Quantitative estimations average of each attribute in relation to class D.

Attribute	Merit [1, 0]	Rank [1, 8]
N_{r_msg}	0.95	1.48
<i>Metric</i>	0.77	2.32
N_{thrd}	0.50	4.8
N_{msg}	0.49	4.36
L_{msg}	0.48	4.7
N_{r_thrd}	0.45	5.09
L_{thrd}	0.43	5.25
<i>Level</i>	0.33	8

problems were identified, we can advise tutors that student acknowledgment is a better indicator of collaboration than activity or initiative.

- The global collaboration attribute *Metric* (see the third row of Table 2) is more related to the decision of recommending than the global collaboration attribute *Level* (see the 8th row of the Table 2). This issue compares the results or the DM approaches and hints at the quality of the Metric Approach.

4.2.2. Learning with decision trees

Given the huge size of the policy table, it is almost impossible for a human expert to understand the policy and validate it. Thus it is necessary to use some synthesis method to obtain a representation of the policy that can be understood by the expert and can allow us to discuss the policy with him. For this reason we used an ML technique, DT algorithms, to create a classification tree (Quinlan, 1986) that could predict the best action in each decision scenario of the policy table. A pruned DT identified the scenarios to recommend and the expert assessed whether these scenarios were suitable for recommending. Each path from the root to a leaf in the DT is a decision scenario, containing a tuple of values for some of the observed variables, and the leaf is labeled with the action recommended for D.

We used the DT algorithm J4.8 for learning the policy. This algorithm is an adaptation to Weka of the DT algorithm C4.5 by Quinlan (1993), which has also been widely used in education data mining (Gaudioso et al., 2009). J4.8 algorithm has been applied to the decision table by varying one parameter, the minimum number of instances per leaf (*minNumObj*), from 5 to 55 in fives. In general, we should expect that the higher the minimum number of instances per leaf, the more pruned the tree is, thus the logical tree should be easier to understand but be more imprecise.

For each DT obtained by varying *minNumObj* we recorded three parameters: the percentage of instances incorrectly classified (*Error*), the number of nodes in the pruned tree (*size*), and the number of leaves in the pruned tree. These three values for each tree are shown in Table 3.

Table 3

Parameters to select the most suitable tree for explaining the student circumstances.

minNumObj	Error	Number of leaves	Size
5	5.00	69	103
10	6.78	39	58
15	6.78	39	58
20	6.76	35	52
25	6.83	35	52
30	7.12	27	40
35	7.93	27	40
40	8.11	25	37
45	8.13	25	37
50	9.81	25	37
55	9.70	21	31

Table 3 was analyzed to select the best tree. This selection was done by trying to find a balance between the precision of the tree and its complexity. A size around 50 is too big and the tree is too complex to be analyzed by a tutor. When the incorrect instance classification error is higher than 9%, the logical tree is imprecise and the recommendation scenarios are pedagogically inconsistent. Fig. 4 shows the selected tree, which was pruned with the value of 40 in the minimal number of instances per leaf parameter. The error is below 9% and a tutor was able to identify low inaccuracy recommendation scenarios.

We note the tree in Fig. 4 is small enough to be understood and to explain the general recommended scenarios. The objective of Fig. 4 is to explain the general behavior of the policy of the ID instead of checking every individual case. We note that the number of instances is 4452, equal to the number of individual cases.

The attributes used by the DT were N_{r_msg} , *Metric*, L_{msg} , N_{msg} , L_{thrd} and N_{r_thrd} , which are some of the attributes more related to class D according to Table 3 in Section 4.2.1. We can observe that the root is the attribute N_{r_msg} , which was the best-valued attribute in the previous analysis.

4.2.3. Evaluation of the decision tree

An expert studied the DT in the Fig. 4 to assess the recommendation decision process. The DT shows clearly four scenarios of recommendation:

- When the attribute N_{r_msg} is *high*. The ID proposes not recommending for most cases. When a student received high acknowledgment by his/her peers, his/her interactions are useful for the other members of the team. This means that the student collaborates well and there is no collaboration problem. If the recommendation is proposed (no recommendation in 1610 cases and recommendation in 32 cases see the right side of Fig. 4), the aim should encourage students.
- When the N_{r_msg} value is *medium*. The ID proposes recommending mainly for cases where the attribute values are not balanced, that is, student initiative, activity and acknowledgment attributes have very different values. This could indicate that a possible collaboration problem is arising. The recommendation should persuade the student to procure a more balanced collaboration process.
- When the N_{r_msg} value is *low*. We can identify two scenarios:
 - When the values of the majority of attributes are *low*, this means that the student has few interactions and his/her peers do not interact with him/her. Thus the student does not collaborate at all. The recommendation should persuade the student to increase his/her interactions.
 - When the values of some attributes are *low* but *Metric* or L_{msg} values are *high*, this means that student behavior, from a collaboration point of view, is not balanced. The student could interact but his/her peers ignore him/her. Of course, there is a collaboration problem. The recommendation should induce the peers to interact with the student.

Fig. 4 shows a DT where different recommendation scenarios are clearly distinguishable. Every scenario can be solved by proposing a different type of recommendation from the pedagogical point of view. The ID can therefore identify a problematic collaboration process scenario and detect different collaboration problems. We claim that the ID approach, which we have explained before in this paper, is a suitable recommendation system in the context of collaborative learning.

reasons for the recommendation. An automatic recommender can use the recommendation decision table to warn students and tutors of the possible problematic circumstances. Tutors thus have the advantage of studying first the problematic circumstances to propose a recommendation that can solve the problem.

We propose an ML module that can increase confidence in the system. In this module we did two different analyses of the decision table recommendation. On the one hand, the ID attributes were analyzed to decide whether to recommend or not. The results of this analysis are explained in Section 4.2.1 and we note that the most important student feature for identifying circumstances that advise recommendation is acknowledgment by his/her team mates.

On the other hand, we have represented the recommendations of the decision table visually using a DT algorithm that analyzes the recommendations proposed and synthesizes the information contained in the decision table. Once the DT had been pruned, the recommendation scenarios were made visible and we claim that these scenarios explain student circumstances that advise recommendation. In Section 4.2.3 the results are shown. We claim that the ID can identify different collaboration scenarios to recommend from a pedagogical point of view, which is more than we expected according to the expert's knowledge.

Moreover, we can develop a pedagogical support module that will clearly display the location of the student circumstances in the logical tree and will propose a recommendation taking into account the collaboration scenarios identified for recommending.

Owing to the dynamic nature of collaboration learning an expert would have to check the system warnings and recommendations to feed back the system. The feedback process can be carried out with the help of the system administrator, who should debug the system following the expert's indications:

- In the collaboration model module: adding or removing collaboration indicators to be taken into account in the ID.
- In the ID module: changing the network structure by adding or removing attributes or reordering them, eliciting another strategy by changing the numerical values to make some scenario more or less important, and modifying the probabilities of some cases to improve smoothly the system results.

Being able to modify the ID model will allow the teacher to adapt the learning process to an environment that can change due to many factors. In general, the discrepancies between the recommendations suggested by the system and those expected by the human being according to his/her knowledge of the domain and the environment should warn of the need to reconsider changes. These changes would be in the structure and/or the quantitative part of the model to adapt the ID so that it is still a useful decision support system.

6. Conclusions and future work

We have proposed a recommendation system based on an ID in the context of collaborative learning in the e-learning environment. The ID solution has provided us with a recommendation decision table, which warns us of some problematic situations. After the analysis of the decision table by two ML processes, we conclude that the most important findings in this research are: (1) acknowledgment of the student by team mates is the most relevant indicator for our system to take a decision and the tutors to make a recommendation, and (2) we have also found that the analysis based on DT algorithms has helped the expert to identify four collaboration scenarios. For each collaboration scenario the recommendation aims to solve a collaboration problem (the individual or

peer one), to warn of a possible collaboration problem or to encourage because of the good teamwork.

Improving the proposed recommendation system is a task that the experts and developers should do interactively to tune the system. The objective is that the system can adapt to the dynamic collaboration process. However, future works need to be done. The automatic recommender together with the pedagogical support module will automatically offer students and tutors warnings and recommendations about the collaboration process. Moreover, the structure of the model was built manually with the help of the expert. We preferred this approach instead of using automatic learning of the structure from the data base file (Neapolitan, 2004) because the expert had knowledge about the cause-effect and probabilistic relations among the variables. Moreover, the expert's participation in the construction of the ID structure helped to increase his/her confidence in the system. However, another possibility would be to follow a very recent approach, *interactive learning* (Bermejo, Oliva, Díez, & Arias, 2012), which consists of a process in which the structure is learned with the help of automatic algorithms, but the expert can participate at any stage of the process by modifying the executions of the algorithms according to his/her knowledge of the domain.

References

- Anaya, A. R., & Boticario, J. G. (2011a). Content-free collaborative learning modeling using data mining. *User Modeling and User-adapted Interaction*, 21, 181–216.
- Anaya, A. R., & Boticario, J. G. (2011b). Application of machine learning techniques to analyze student interactions and improve the collaboration process. *Expert Systems with Applications*, 38, 1171–1181.
- Arias, M., & Díez, F. J. (2008). Carmen: An open source project for probabilistic graphical models. In *Proceedings of the Fourth European Workshop on Probabilistic Graphical Models (PGM'08)*, Hirtshals, Denmark (pp. 25–32).
- Baghaei, N., & Mitrovic, A. (2007). From modelling domain knowledge to metacognitive skills: Extending a constraint-based tutoring system to support collaboration. In C. Conati, K. McCoy, & G. Paliouras (Eds.), *UM 2007. LNAI* (Vol. 4511, pp. 217–227). Springer Verlag.
- Bermejo, I., Oliva, J., Díez, F. J., & Arias, M. (2012). Interactive learning of Bayesian networks with OpenMarkov. In *Proceedings of the 6th European workshop on probabilistic graphical models (PGM-2012)*, Granada, Spain (pp. 27–34).
- Bielza, C., Gómez, M., & Shenoy, P. P. (2011). A review of representation issues and modelling challenges with influence diagrams. *Omega*, 39, 227–241.
- Bratitsis, T., Dimitracopoulou, A., Martínez-Monés, A., Marcos-García, J., & Dimitriadis, Y. (2008). Supporting members of a learning community using interaction analysis tools: The example of the Kaleidoscope NoE scientific network. In *Proceedings of the IEEE international conference on advanced learning technologies, ICALT 2008* (pp. 809–813).
- Caballé, S., Daradoumis, T., Xhafa, F., & Juan, A. (2011). Providing effective feedback, monitoring and evaluation to on-line collaborative learning discussions. *Computers in Human Behavior*, 27, 1372–1381.
- Chang, F.-I. (2003). Quantitative analysis on distance learning courseware. *Multimedia Tools and Applications*, 20, 51–65.
- Collazos, C. A., Guerrero, L. A., Pino, J. A., Renzi, S., Klobas, J., Ortega, M., et al. (2007). Evaluating collaborative learning processes using system-based measurement. *Educational Technology and Society*, 10, 257–274.
- Conati, C., Gertner, A., & VanLehn, K. (2002). Using Bayesian networks to manage uncertainty in student modeling. *User Modeling and User-Adapted Idots*, 12, 371–417.
- Conati, C., Gertner, A. S., VanLehn, K., & Druzdzel, M. J. (1997). On-line student modeling for coached problem solving using Bayesian networks. In *Proceedings of the sixth international conference on user modeling (um'97)* (pp. 231–242). Springer Verlag.
- Daradoumis, T., Martínez-Mónes, A., & Xhafa, F. (2006). A layered framework for evaluating online collaborative learning interactions. *International Journal of Human-Computer Studies*, 64, 622–635.
- Duque, R., & Bravo, C. (2007). A method to classify collaboration in CSCI systems. In B. Beliczynski et al. (Eds.), *ICANNGA 2007. Part I: LNCS 4431* (pp. 649–656). Berlin Heidelberg: Springer.
- García, P., Amandi, A., Schiaffino, S., & Campo, M. (2007). Evaluating Bayesian networks' precision for detecting students' learning styles. *Computers & Education*, 49, 794–808.
- Gaudio, E., Montero, M., Talavera, L., & del Olmo, F. H. (2009). Supporting teachers in collaborative student modeling: A framework and an implementation. *Expert Systems with Applications*, 36, 2260–2265.
- Howard, R. A., & Matheson, J. E. (1984). Influence diagrams. In R. A. Howard & J. E. Matheson (Eds.), *Readings on the principles and applications of decision analysis* (pp. 719–762). Menlo Park, CA: Strategic Decisions Group.

- Johnson, D., & Johnson, R. (2004). Cooperation and the use of technology. In *Handbook of research on educational communications and technology* (pp. 401–424). Taylor and Francis Group.
- Kahrimanis, G., Meier, A., Chounta, I.-A., Voyiatzaki, E., Spada, H., Rummel, N., et al. (2009). Assessing collaboration quality in synchronous CSCL problem-solving activities: Adaptation and empirical evaluation of a rating scheme. In *Learning in the synergy of multiple disciplines. 4th European conference on technology enhanced learning, EC-TEL 2009* (pp. 267–272). Springer Verlag.
- Lacave, C., & Díez, F. J. (2002). A review of explanation methods for Bayesian networks. *Knowledge Engineering Review*, 17, 107–127.
- Lacave, C., Luque, M., & Díez, F. J. (2007). Explanation of Bayesian networks and influence diagrams in Elvira. *IEEE Transactions on Systems, Man and Cybernetics—Part B: Cybernetics*, 37, 952–965.
- Millán, E., Loboda, T., & de-la Cruz, J. P. (2010). Bayesian networks for student model engineering. *Computers & Education*, 55, 1663–1683.
- Millán, E., & Pérez-De-La-Cruz, J. (2002). A Bayesian diagnostic algorithm for student modeling and its evaluation. *User Modeling and User-Adapted Interaction*, 12, 281–330.
- Mislevy, R., & Gitomer, D. (1996). The role of probability-based inference in an intelligent tutoring system. *User Modeling and User-Adapted Interaction*, 5, 253–282.
- Neapolitan, R. E. (2004). *Learning Bayesian networks*. Upper Saddle River, NJ: Prentice-Hall.
- Niblett, T. (1987). Constructing decision trees in noisy domains. In *EWSL'87* (pp. 67–78).
- Park, C.J., & Hyun, J.S. (2006). Comparison of two learning models for collaborative e-learning. In *Edutainment 2006, LNCS* (Vol. 3942, pp. 50–59).
- Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. San Mateo, CA: Morgan Kaufmann.
- Perera, D., Kay, J., Yacef, K., & Koprincka, I. (2007). Mining learners' traces from an online collaboration tool. In *Proceedings of the 13th international conference of artificial intelligence in education, workshop educational data mining* (pp. 60–69).
- Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1, 81–106.
- Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. San Diego: Morgan Kaufmann.
- Reye, J. (1996). A belief net backbone for student modelling. In *Intelligent tutoring systems* (pp. 596–604).
- Romero, C., & Ventura, S. (2010). Educational data mining: A review of the state-of-the-art. *IEEE Transaction on Systems, Man, and Cybernetics. Part C: Applications and Reviews*, 40, 601–618.
- Rosch, E. (1978). Principles of categorization. In E. Rosch & B. Lloyd (Eds.), *Cognition and categorization* (pp. 27–48). Hillsdale, NJ: Lawrence Erlbaum.
- Soller, A., Martinez, A., Jermann, P., & Muehlenbrock, M. (2005). From mirroring to guiding: A review of state of the art technology for supporting collaborative learning. *International Journal of Artificial Intelligence in Education*, 15, 261–290.
- Strijbos, J.-W., & Fischer, F. (2007). Methodological challenges for collaborative learning research. *Learning and Instruction*, 17, 389–393.
- Swan, K., Shen, J., & Hiltz, S. (2006). Assessment and collaboration in online learning. *Journal of Asynchronous Learning Networks*, 10, 45–62.