HAL
open science

# Trace-Based Weighting Approach for Multiple Criteria Decision Making

Hoang Nam Ho, Mourad Rabah, Samuel Nowakowski, Pascal Estraillier

## To cite this version:

Hoang Nam Ho, Mourad Rabah, Samuel Nowakowski, Pascal Estraillier. Trace-Based Weighting Approach for Multiple Criteria Decision Making. Journal of Software, 2014, 9 (8), pp.2180-2187. hal-01058156

**HAL Id: hal-01058156**

**https://hal.science/hal-01058156v1**

Submitted on 26 Aug 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Trace-Based Weighting Approach for Multiple Criteria Decision Making

Hoang Nam HO[1]
Email: hoang_nam.ho@univ-lr.fr

Mourad RABAH[1], Samuel NOWAKOWSKI[2], Pascal ESTRAILLIER[1]
Email: {mourad.rabah, pascal.estraillier}@univ-lr.fr, samuel.nowakowski@univ-lorraine.fr
(1) L3i Laboratory – University of La Rochelle – La Rochelle – France
(2) University of Lorraine – LORIA, UMR 7503 – Nancy – France

***Abstract*— Decision-making has evolved as an interesting research problem for decision community. We consider a decision problem that takes into account several criteria called Multiple Criteria Decision Making (MCDM) in an interactive application for adaptive execution. In this paper, we present a method for automatically weighting criteria generation based on users' traces. In our method, we suggest a process that contains all steps describing alternately what is necessary to prepare in order to weight all criteria. We propose a modified method using Naïve Bayes network to exploit the traces (the past of users), there will be used as information for estimating the score of criteria. Experimental results are presented to illustrate a full process and an automatic generation of weighting criteria by a set of values.**

***Index Terms*— criteria weighting, interactive adaptive system, Multiple Criteria Decision Making, Naïve Bayes, traces.**

## I. Introduction

The context of our work is the decision making in Interactive Adaptive Systems (IAS). An IAS is a system that can adapt its execution according to users behaviours [1], [2]. The system checks users interactions and adapt application's scenario with respect to designer's constraints. Indeed, when users are in an interactive application, they perform actions to achieve a given goal. During the execution, it could happen that the system cannot continue according to the current scenario due to an ambiguity of user's actions during the interaction or if the current data are not sufficient to continue the logic of the execution. At this point, a decision support system is needed to resolve this situation. Decision-making must reach two conditions: one is to provide a solution that allows the user to continue the interactions and the other is to reach the objective of the application (designer's constraints). The main purpose of decision-making is to find the action that resolves the problem according to several application related criteria. These criteria may be numerous and heterogeneous. Making the decision must take into account all of them in order to choose among a set of possible solutions the best one. This is MCDM, the Multiple Criteria Decision Making [3], [4].

Moreover, since we consider an interactive application with many interactions between system's actors. During application's execution, actors actions can generate traces [5], [6]. These traces may contain valuable information and represent huge quantity of data to compute. Based on these traces, we can know users' past habits and abilities. This information helps the system to solve the adaptation problem. In this paper, we propose a method to use these traces for criteria weighting.

MCDM is the most well known branch of decision-making problem. So, there is a lot of works that focus on the strength of this approach in order to make the best decision. In [7], [8] authors propose a typical decision matrix based computation. This decision matrix is a matrix in which elements indicate the performance of the alternative when it is evaluated according to each defined criterion. Others approaches have been detailed in [9], [10] based on utility function. In these methods, the decision maker has determined the weights of the decision criteria. The decision maker will estimate the score by himself. It does not rely on any factor or principle to compute. That is only a feeling guided weighting. The perception through feeling must be completed by perception through reason. It is necessary to have a method that can automatically estimate the score of each criterion in order to classify criteria according to their respective priority.

Therefore the criteria weighting evolve as a challenge for decision community. For example, authors determined criteria weights from partial ranking of the alternatives, individual criteria, or criteria pairs [11]. [12] integrate the assessment of the scores (cardinal input) and rankings (ordinal input) of the decision-makers' preference structure. Relative criteria importance is represented by a set of cardinal weights or ranks. The analytic hierarchy process (AHP) [13] is based on decomposing a complex MCDM into a hierarchic structure. It contains three levels: the first is the goal of the problem, the second lists all necessary criteria to evaluate the goal and the final level indicates all of the existing solutions to solve the problem. A compared matrix is constructed in order to express the relative importance between two criteria. Each element of the matrix is estimated by a value indicating the relative

© 2014 ACADEMY PUBLISHER
doi:10.4304/jsw.9.8.2180-2187

importance ranging from 1 (equal importance) to 9 (extreme importance of one over the other). Therefore, from this matrix, we calculate an Eigen vector; this vector is the priority vector of criteria. Each value is the weight that corresponds to the criterion for every line of the matrix.

These methods are turning-points in decision-making community because they support a very clear model and a mathematical theory about the structure of MCDM and the criteria weighting. However the drawback is that it relies on the decision maker to determine the ranks or the relative importance between criteria. These values depend on decision maker's perception. It is not automatic and it depends extremely on the intuition of the decision maker.

In this paper, we propose an automatic weighting process of decision criteria for interactive adaptive systems. We use user traces as input data to score the criteria weights. The main idea of our proposition is the construction of a prediction model based on users' traces in order to estimate the ability of criteria achievement by the users. To build a prediction model, the data mining approach emerges as an efficient method to solve the problem. We apply Naïve Bayes network and adjust the results of the prediction step to get a score for each criterion. The obtained weights are aimed to be used in the decision-making process for system adaptation.

To illustrate our approach we have chosen an e-Learning case study. The considered e-Learning system is an IAS that has to adapt to users profile and progress. To make a decision, it uses users traces to predict future users behaviours and accordingly adapt the execution.

The paper is organized as follows: section II gives a process that indicates globally the trace-based MCDM for IAS; all of the steps of the proposed process are described in section III where we only focus on steps that concern criteria weighting; experimental results and discussion are presented in section IV, section V concludes the paper.

## II. Process for Trace-Based MCDM for IAS

Our proposed process for MCDM in interactive adaptive systems is a two phases process. The first phase describes how to estimate the score for each criterion. The second phase is devoted to decision-making. We introduce the complete process of MCDM in order to have a full overview of our methodology, but in this paper we will only consider the criteria weighting phase (step (i) to (v) below).

The full process consists in 8 steps:

- Objective identification (i): it defines the main objective the user wants to achieve.
- Criteria identification (ii): all of criteria will be determined in order to evaluate these contributions to the achievement of the objective.
- Data preparation (iii): mentioned data are users' traces based. All of traces created by the user will be collected and structured in a traces system.
- Data extraction (iv): this step removes the irrelevant information and gets the most appropriated information to criteria examination.
- Criteria weighting (v): this step provides the weight of each criterion determined in (ii). Indeed, there are more criteria; they are not classified at the same level because each of them has a different relative importance of others.
- Alternatives analysis (vi): it is also a necessary step in the decision because decision method requires a set of alternatives to give a best choice.
- Decision-making (vii): this phase is the core of the decision. During the computation of this step, it can happen that one or more criteria are completed then, it could return to the step (v) to re-estimate the score for each criterion.
- Results back up (viii) traces for next executions.

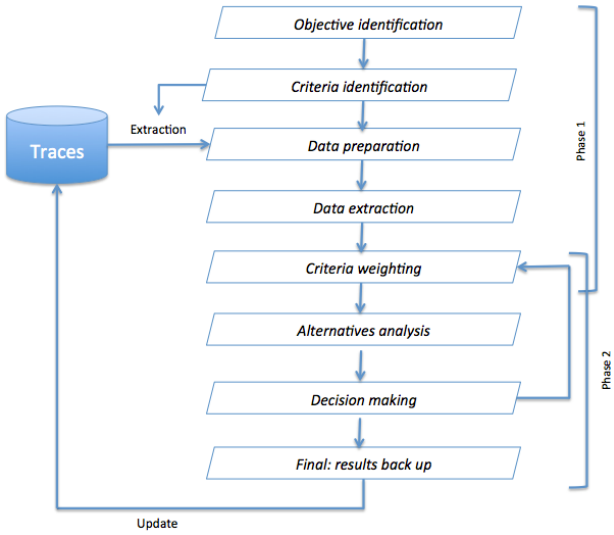


Figure 1. Proposed process for MCDM problem.

## III. Criteria Weighting Description

We will detail our criteria weighting approach. It is based on users' traces. We explain in this section how to exploit these traces in order to score decision criteria.

### A. Objective Identification

An objective is an outcome that the user wants to achieve in an application. In fact, one objective must satisfy several properties including five characteristics listed in the following acronym SMART (Specific, Measurable, Applicable, Realistic, Temporal) [14]. They allow the user to understand and facilitate the interpretation.

- **S**pecific: a purpose's statement should be stated specifically and unambiguously. It is also important to indicate the formulation of an objective without negation. This means that the user should avoid targets containing "must not". The statement must focus the user on the essential target.
- **M**easurable: an objective is measurable because it allows the achievement estimation of the target.
- **A**pplicable: it must concern the context. The objective is a part that relates to considered problem, it cannot exceed the scope of the problem.

© 2014 ACADEMY PUBLISHER

- **R**ealistic: when an objective is defined, it must be reachable. For an achievable objective, it can be determined how to make it succeed.
- **T**emporal: a good objective is a target that satisfies the time constraints.

### *B. Criteria Identification*

A criterion is a principle or an element of reference that allows to judge, to estimate and to define something. In our case, the criteria are references to evaluate the achievement of the objective. In order to evaluate an objective, it must be based on one or more criteria. So it requires to identify all criteria that relate to the objective achievement [15]. The criteria identification depends on each application and the intuition of the designer.

A criterion can be continuous or discrete. For example, the users' number in the application is calculated by a discrete value, the execution time of a task is represented by a continuous value.

In this paper, we assume that a problem, which contains n criteria, is *{c1, c2,…, cn}*.

### *C. Data Preparation and Extraction*

A context is an observable activity because we observe all the information and traces in the past of the users. We present in this section how to structure a user profile [16] combining the users' traces as shown in Fig. 2.

The information of the user is divided into four categories: Administration, Goals, Skills and Activities. We will describe in detail each category in the following:

- Administration: personal information (name, date of birth, gender), certification and diplomas, preferences and non-preferences about languages (English, French, Spanish…), technology.
- Goals: the target that the user wishes to achieve.
- Skills: user's knowledge about personal expertise.
- Activities: the information recorded by the user.

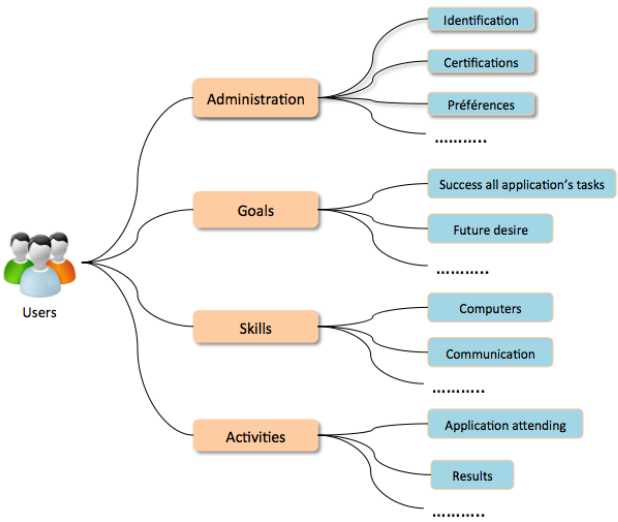


Figure 2. Modeling a user by different information tuples.

In our approach, data are a combination of basic information and the traces of the users. Knowing that, we now determine what information is usable to evaluate the users' criteria. The extracted information concerns skills, language, activities that contribute in evaluating the criteria accomplishment.

After extracting contributed information (attribute), we filter all the users who dealt with actual activity and we also observe the result of the criteria completion. This result is indicated by Success (user complete this criteria) or Failure (user cannot reach this criteria). This database will be used as a training database. The structure of the training database is described as follow:

TABLE I.
DATA AND TRACES ASSOCIATED WITH CRITERIA

| Attribute 1 | …… | Attribute k | Criterion 1 | … | Criterion 2 |
|---|---|---|---|---|---|
| Val 1 | …… | Val k | Success | … | Failure |
| Val 2 | …… | Val 1 | Failure | … | Success |

For new users, we will collect all information that concerns the attribute to determine the criteria achievement $u\ (att_1, att_2,\ldots, att_k)$. Each actual user will estimate the probability to reach the defined criteria. It is the base for criteria weighting before making decision.

### *D. Criteria Weighting*

We will detail in this section an approach whereby the criteria are automatically weighted. In fact, in order to make decision engine working, we must first solve the problem by preparing these previous steps because all decision algorithms require a volume of information to process. One part of the data is the weight of criteria or in other words having a list of priority of the criteria. Indeed, one problem obviously contains many criteria that users want to achieve; perhaps there is a criterion that is not as important as the others. It will consider what the target users would reach early. If the criteria are not prioritized, the engine will assume that they have the same of importance level. This can influence the achievement of the criteria for the final objective. So it is the reason why we must supply to decision engine a list of the criteria weight to find a satisfying alternative. From the extracted data in the previous step, we will process and compute to obtain a priority list of criteria. The weight considered in our approach is a numerical value.

We now present the way to apply data mining in our approach. The goal is to use link relevant information to criteria, as shown in Table I, and then to predict the probability of criteria accomplishment. The users want to reach the defined objective, and this is evaluated by *n* criteria. Based on users' skills and traces, we can estimate what criteria the users can reach by computing the accomplishment probability. These probabilities represent the ability of criteria accomplishment. We realize that this problem is as a binary classification problem. For each user and for each criterion, we use the extracted data to predict if he can satisfy the criterion (Success) or not (Failure). Among classification methods, we make an analysis of these methods via the characteristic of our problem in order to choose the one that we can take as prediction algorithm. We analyse the algorithms that have been often used in the literature for data classification. We evaluate these methods by four characteristics relating our classification problem. First, we consider the

© 2014 ACADEMY PUBLISHER

data, because we must analyse the values of the users' traces and skills (both of continuous and discrete type) in order to estimate. Second, we need a method that runs fast during the execution because if the algorithm runs for a long time, it will influence the others phases in the system. Then, the next factor linked to our prediction problem is that the number of records in the training database because some methods require huge samples to train. The four following methods are possible candidates to build a prediction model: Naïve Bayes [17] [18], k Nearest Neighbors (k-NN) [19], Decision tree C4.5 [20], Support Vector Machine (SVM) [21]. Among these four methods, we decide to use Naïve Bayes algorithm because it is the most suitable for our problem. In fact, the data that we have seen and used in our case are of various types. Values are numeric, string (continuous or discrete). This is the reason why we do not use the SVM (Support Vector Machine) although the prediction quality of SVM is very high. In addition, the Bayes method is simple and easy to implement. Because of the small size of the training database, we cannot apply k-NN method. Between the two remaining methods, Naïve Bayes and decision tree C4.5, we realize that the decision tree must choose an attribute to partition data. If the selection attribute is not good, the quality of the tree will be affected. To conclude, we consider the Naïve Bayes method the appropriate method for our case.

The Naïve Bayes method is based on Bayes theorem. It is a classification technique based on conditional probability. A posteriori probability *P(h/D)* of a hypothesis h given a set of data *D* is given by Bayes theorem:

$$P(h/D) = P(h).P(D/h)/P(D) \quad (1)$$

*P(D/h)* is called likelihood, and it represents that some data is produced under the assumption of this hypothesis *h*. *P(h)* is a priori probability of hypothesis *h* (probability before having data *D*). *P(D)* is a priori probability of data *D*.

If there is a user *u* that wants to start an application with *n* criteria, we will exploit his profile, his traces (*k* attributes) to predict the probability of achievement for each criterion. So a user can be represented by a record containing the values for each extracted attributes. We assume that we extract *m* attributes for one user. A criterion is labelled Success or Failure. There are two hypotheses in our case. In order to build a prediction model based on Naïve Bayes, we need to have our training base. Assume that we have a base of records as Table I; we calculate the likelihood of Success and Failure:

$$P(Success/u) = P(Success) \times \prod_{i=1}^{k} P(att_1 / Success) \Big/ P(u) \quad (2)$$

$$P(Failure/u) = P(Failure) \times \prod_{i=1}^{k} P(att_1 / Failure) \Big/ P(u) \quad (3)$$

The Table III is the result of conversion into probability by normalization given:

$$P(Success) = P(Success/u) \big/ \big(P(Success/u) + P(Failure/u)\big) \quad (4)$$

$$P(Failure) = P(Failure/u) \big/ \big(P(Success/u) + P(Failure/u)\big) \quad (5)$$

TABLE II.
PROBABILITY OF SUCCESS AND FAILURE IN PREDICTION

| | Criterion 1 | ……. | Criterion n |
|---|---|---|---|
| Success | $P_1(Success)$ | …….. | $P_n(Success)$ |
| Failure | $P_1(Failure)$ | …….. | $P_n(Failure)$ |

Then, we propose a statement that is the new principle to solve our problem. It states: if a user has a tendency to reach a criterion in a set of criteria, the disregard of prioritizing it over the others can be justifiable. We should, instead, focus on the criteria whose achievement rates are low in order to allow the user to complete all the criteria.

After the computation of the percentage of each criterion, we consider the Failure probability. We classify these values in decreasing order; we will deduce a list of priority of criteria that corresponds to the order of the probability values. We associate for each criterion a Failure probability that indicates its weight.

In addition, our presentation below corresponds to a case with one user. In the case of many users who want to execute the application, how can we react? We also propose an adjusted method that can be solved in this case. For each user, we use the predictive model to estimate the achievement for each criterion (Success or Failure). We regroup the results as shown in Table III.

TABLE III.
PROBABILITY OF SUCCESS AND FAILURE FOR EACH CRITERION

| User | Criterion 1 | ……. | Criterion n |
|---|---|---|---|
| User 1 | Success | ……. | Failure |
| User 2 | Failure | ……. | Failure |
| ……. | ……. | ……. | ……. |
| User m | Success | ……. | Success |

According to our statement, we focus on the failure probability. But in this context, we have more than one user; all of them want to finish the application by reaching all criteria. If many users can reach a criterion than the others in a set of criteria, it must not prioritize it over the others. We should focus on the criteria whose users' number of achievement is low in order to help users to finish all the criteria. So we are interested in the number of users who cannot reach the criteria, in other hands, the number of users who have a Failure probability. Then, we divide this number with the total of users who attend in the activity. Finally, we obtain a rate, which is the weight of criteria.

## IV. EXPERIMENTATION

Experimentation has been carried out to illustrate the performance of the proposed method. First, we present the problem we want to solve. Our approach will be illustrated in this section. The considered context is a learning environment, where there are one teacher and

© 2014 ACADEMY PUBLISHER

several learners. During the execution i.e., the course, teacher and learners proceed sequentially by action. If there is a moment when the teacher or learners cannot decide to carry out an action, a decision engine will help them to choose the best action on the set of the possible actions. The best action is defined as an action that satisfies both the goals of the teacher and the learners. Evidently, the success of the course is evaluated by the contribution of the teacher and the learners via the criteria. So the decision engine must take into account all of the criteria in order to obtain the best solution. But what is priority law inside this set of criteria?

In the following, we will describe all the steps of criteria weighting procedure. The result of this experimentation is a list of weights.

### A. Identification of the Objective

We will now define the final objective at the end of this course. We use SMART standard in section III.A.

- **S**pecific: the statement of this objective is understandable.
- **M**easurable: deciding a course is completed or not belongs to the teacher.
- **A**pplicable: in education, is the result of a course applicable?
- **R**ealistic: the accomplishment of the course is an objective that can be reached.
- **T**emporal: A course has to be feasible with the deadlines given by the teacher.

### B. Criteria Identification

We will now consider the criteria in which we evaluate the final objective. In fact, in the learning course, in order to determine if the course ends or not, we need to take into account learners and teacher both. If we consider only one actor, the evaluation of the course could not be exact. Thus, we propose to identify criteria for assessing the quality of a course. We identify three criteria:

- Does the course follow the sequence of actions defined by the teacher?
- Is course's program respected?
- The evaluation of teacher about the comprehension of learners.

### C. Data Preparation and Extraction

We would like to extract information in the profile of learners who enrolled in a Data Mining course. Previously obtained data are used as training base to build a prediction model of the learners' achievement criteria. Suppose we have a base of learners' profiles. For each registered learner in learning environment, we have created a profile containing data as shown in Fig. 2. We only consider courses in the field of Probability, Mathematics, Algorithm, Database, Programming and English because these courses are the prerequisite for the Data Mining course.

For all the learners, we associate the results of the three criteria: time, sequence and level of understanding.

Then, we present the way to assess these six attributes. For probability and mathematics, we use the obtained grades. For algorithm and database, we use a scale from 0 (learner do not have knowledge about the domain) to 5 (learner is expert in the domain). We use levels to evaluate programming skill: Basic, Normal, Advanced, and Expert. The English speaking ability can be obtained with these levels: weak, intermediate and advanced. Table IV summarizes the attributes.

TABLE IV.
DATA MINING COURSE ASSESSMENT

| Requirement | Assessment |
|---|---|
| Probability (P) | [0, 20] |
| Mathematic (M) | |
| Algorithm (A) | 0 (have no knowledge) to 5 (expert in the domain) |
| Database (D) | |
| Programming (Prog) | Basic or Normal or Advanced or Expert |
| English (E) | Weak or Intermediate or Advanced |

We now associate the result of each learner the criteria accomplishment in the database. It means, for each learner who have these requirements, we observe that he is on Success (S) or Failure (F) in the evaluation of three criteria: time, program and comprehension. Indeed, a learner has different skills and abilities. Between the result and skills, there is a very intimate relationship. If a learner has the best skills, it will be probably reach all the criteria of the teacher. If a skill is not good, it can be influenced by one or more criteria.

In order to have a training database for our experimentation, we will build a database of 1000 learners (1000 samples). That means there were 1000 learners have been attended in the Data mining course. When they have been finished the course, we observed the accomplishment for each determined criterion. For six attributes, we create data in respecting the discrete uniform distribution function. In fact, each attribute is obtained a value in a set of value; the probability of obtaining each value is equal. Due to the difficulty of criteria evaluation, we create a database by referencing the idea of teachers of data mining. They are many experiences in the domain and the learners' evaluation. So based on their ideas, we have the data in respecting the rule: if half of attributes are superior to the average thresholds (for two attributes about probability and mathematic: 10, for algorithm and database: 2, for programming: Normal and for English: Intermediate), the learner will reach the criterion of time. If the values of 4 or 5 attributes are superior to the average, the criterion of programming will be achieved. If six attributes are perfect, the learner will reach the criterion of comprehension. In addition, besides respecting theses rules, we also create data that symbolize the noise of the data. For example, a learner has a good evaluation in six attributes, but he does not concentrate on the course, so he cannot achieve this course. As another example, if a learner has only three attributes that satisfy with the average, but he works with a lot of effort, he can reach all the criteria of the course. We combine all possible cases in order to have a completed database. It contains normal and also noise data; the goal is to increase the robustness of the prediction model.

© 2014 ACADEMY PUBLISHER

All records in the training base above is reserved in a database and each record contains six fields corresponding to the six attributes of learners and three fields are the result of each criterion. An example illustrates in Table V:

TABLE V.
EXAMPLE OF COMPLETED DATABASE

| P | M | A | D | Prog | E | Time | Pro-gram | Compre-hension |
|---|---|---|---|---|---|---|---|---|
| 19 | 15 | 1 | 3 | basic | Advance | S | S | S |
| 11 | 10 | 1 | 0 | basic | Weak | S | F | F |

For new learners, we consider only six attributes to predict the probability of achievement for each criterion. The constructions of the prediction model in the phase of criteria weighting will estimate if the new learner can be reach the criteria of the course. The detailed description of weighting will be presented in the next section.

### D. Criteria Weighting

We develop in this section a process for ordering the priority of the criteria. In order to prioritize the criteria, we use the training database created above to construct the prediction model. We will predict results of achievement of the learners based on the skills and abilities of learners.

From 1000 samples, we test the predictive ability of the Naïve Bayes algorithm. The test protocol that we use is k-fold method. The main idea of this protocol is to divide randomly the initial base in 10 folds. Nine of the 10 folds used for the training base, the rest is used for the testing base. We repeat 10 times to have a variety of data. We calculate the prediction rate every once time. After making 10 times, we calculate the average of 10 times, this value shows the rate of correct prediction of the model or the performance of our prediction model.

We present in the following the performance of three prediction models with three criteria: the time prediction, the program prediction and the comprehension prediction. The Table VI details the performance of each model.

TABLE VI.
PERFORMANCE OF PREDICTION MODEL ON THREE CRITERIA

| Prediction | Time | Program | Comprehension |
|---|---|---|---|
| Success rate | 99.5% | 84.9% | 76.3% |
| Failure rate | 0.5% | 15.1% | 23.7% |

For the criterion of time: the performance of prediction model is extremely high. We realize that the success rate decrease in two criteria program and comprehension. Because, the comprehension of learner is very difficult to evaluate, and the three prediction models that we build is an approximate model. They do not predict exactly all cases, but there is an error tolerance for each model, there are success and failure rate.

A learner has the information about the skills, the traces in the past as follows: Probability = 12, Math = 15, Algorithm = 2, Database = 1, Programming = Normal and English = Courant. We denote *learner*$_1$ *(att*$_1$*=12, att*$_2$*=15, att*$_3$*=2, att*$_4$*=1, att*$_5$*=Normal, att*$_6$*=Courant).* Using these values for three prediction models to estimate the probability of achievement of three criteria. For example, we estimate *P(Success/learner*$_1$*)* and *P(Failure/learner*$_1$*)* on the criterion of Time based on (2) and (3). *P(Success)* et *P(Failure)* are calculated during the building of the prediction model on training base:

$$P(Success/learner_1) = P(Success) \times \prod_{i=1}^{6} P(att_i / Success) / P(learner_1)$$
$$= P(Success) \times P(12/Success) \times ... \times P(Courant/Success) / P(learner_1)$$
$$P(Failure/learner_1) = P(Failure) \times \prod_{i=1}^{6} P(att_i / Failure) / P(learner_1)$$
$$= P(Failure) \times P(12/Failure) \times ... \times P(Courant/Failure) / P(learner_1)$$

Then, we use (4) and (5) to normalize *P(Success/learner*$_1$*)* and *P(Failure/learner*$_1$*)* as Table VII.

TABLE VII.
PREDICTION OF ACHIEVEMENT PROBABILITY OF LEARNER

| | Time | Program | Comprehension |
|---|---|---|---|
| Success | 0.999 | 0.033 | 0.031 |
| Failure | 0.001 | 0.967 | 0.969 |

We apply the statement indicated in the section III.D. We arrange the priority of criteria by arranging the failure probability in the Table VII. We will get the priority and the weight of three criteria: *Comprehension* (**0.969**), *Program* (**0.967**), *Time* (**0.001**).

For this user, it requires to concentrate on the comprehension because of the program and the time criteria, he can be reached within difficulty. The probability of prediction of comprehension criterion is low, so it is very important to make comprehension in priority in order to help this user finish all of the criteria.

This is the case of having one learner. If we want to weight the criteria for a class with many learners, how to process our problem? We will illustrate an example of a class that contains 15 learners who will participate in the Data Mining course as Table VIII.

TABLE VIII.
ATTRIBUTES OF 15 LEARNERS

| Learner | Proba | Math | Algo | Data | Prog | English |
|---|---|---|---|---|---|---|
| 1 | 12 | 14 | 2 | 3 | Normal | Weak |
| 2 | 9 | 15 | 4 | 1 | Basic | Intermediate |
| 3 | 6 | 9 | 3 | 1 | Advance | Weak |
| 4 | 14 | 10 | 1 | 1 | Expert | Intermediate |
| 5 | 5 | 7 | 2 | 0 | Normal | Weak |
| 6 | 11 | 8 | 1 | 1 | Advance | Intermediate |
| 7 | 15 | 17 | 4 | 3 | Advance | Advance |
| 8 | 8 | 10 | 3 | 4 | Normal | Intermediate |
| 9 | 6 | 9 | 4 | 5 | Expert | Intermediate |
| 10 | 14 | 12 | 3 | 2 | Normal | Weak |
| 11 | 16 | 11 | 1 | 1 | Advance | Advance |
| 12 | 9 | 16 | 2 | 3 | Expert | Weak |
| 13 | 10 | 11 | 3 | 5 | Basic | Intermediate |
| 14 | 7 | 9 | 1 | 1 | Basic | Weak |
| 15 | 11 | 14 | 5 | 5 | Normal | Intemediate |

Each learner will be estimated by prediction model that is built by Naïve Bayes. The result of every time is the

© 2014 ACADEMY PUBLISHER

success or the failure of the learner for each criterion. The Table IX describes the detailed result for estimation.

TABLE IX.
RESULT OF THE CRITERIA PREDICTION OF 15 LEARNERS

| Learner | Time | | Program | | Comprehension | |
|---|---|---|---|---|---|---|
| | *Success* | *Failure* | *Success* | *Failure* | *Success* | *Failure* |
| 1 | x | | x | | x | |
| 2 | x | | | x | | x |
| 3 | x | | | x | | x |
| 4 | x | | | x | | x |
| 5 | | x | | x | | x |
| 6 | x | | x | | | x |
| 7 | x | | x | | x | |
| 8 | x | | | x | | x |
| 9 | x | | | x | | x |
| 10 | x | | x | | x | |
| 11 | x | | x | | | x |
| 12 | x | | x | | x | |
| 13 | x | | x | | x | |
| 14 | x | | | x | | x |
| 15 | x | | x | | x | |
| Total | 14 | 1 | 8 | 7 | 6 | 9 |

We consider only the total number of learners who do not reach the criteria. Based on these values, we divide by the total of learners; we obtain the weight of three criteria as follows: *Time* (**1/15 = 0.067**), *Program* (**7/15 = 0.467**) and *Comprehension* (**9/15 = 0.6**).

There are the weights of three criteria. They can be used in all of decision methods that require a list of criteria weights.

This experimentation illustrated how to estimate the criteria in MCDM. Our approach is applicable in interactive application under the hypothesis of having a traces base in order to build a prediction model for criteria accomplishment. If we consider a situation that we cannot collect or observe the traces, our approach is not applicable.

## V. CONCLUSION

In this paper, we have presented a process to estimate the weight of criteria in multi criteria decision making for an interactive application. Firstly, we have identified firstly the objectives and the criteria that contribute to reach these objectives. To weight criteria, we use both user's profile and user's traces. The collected data is put in a database. We extract from the obtained database the pertinent information that is used to configure criteria accomplishment prediction model. We have used the Naïve Bayes algorithm to construct this model. Each defined criterion is predicted in order to determine the probability of criterion achievement. We focus only the criterion failure prediction in order to help users to increase the weight of criteria which finishing probability is low. This principle will help users to take into account criteria being difficult to achieve. Experimentation has been introduced to illustrate our approach by describing all the steps of the presented process.

Our work on criteria weighting is a first contribution to our methodology to make decisions in interactive adaptive systems. We will use our results to set the weights in decision methods. These weights are estimated using a prediction model, which is more precise than the weights obtained by the intuition of the decision maker.

Our future work is devoted to the problem of increasing the performance of our criteria prediction model. We will also try to build a database with real and consequent data in order to increase the robustness of the model by considering data containing noise as in reality.

## REFERENCES


[1] P. Brun and M. Beaudouin-Lafon, "A taxonomy and evaluation of formalisms for the specification of interactive systems," in *Proc of the HCI'95 conference on People and computers*, 1995, pp. 197–212.

[2] P. Pham, M. Rabah, and P. Estraillier, "Handling the Misunderstanding in Interactions: Definition and Solution," in *Int. Conf. on Software Engineering & Applications SEA 2011*, 2011, pp. 47–52.

[3] E. Triantaphyllou, B. Shu, S. N. Sanchez, and T. Ray, "Multi-Criteria Decision Making□: An Operations Research Approach," vol. 15, pp. 175–186, 1998.

[4] M. M. Koksalan, W. Jyrki, and S. Zionts, *Multiple Criteria Decision Making: From Early History to the 21st Century*, World Scientist. 2011.

[5] J. Laflaquière, L. S. Settouti, Y. Prié, and A. Mille, "Trace-Based Framework for Experience Management and Engineering," in *10th International Conference, KES 2006*, 2006, pp. 1171–1178.

[6] D. Clauzel, K. Sehaba, and Y. Prié, "Enhancing synchronous collaboration by using interactive visualisation of modelled traces," *Simul. Model. Pract. Theory*, vol. 19, no. 1, pp. 84–97, 2011.

[7] P. Fishburn, "Additive Utilities with Incomplete Product Set: Application to Priorities and Assignments," *Oper. Res. Soc. Am.*, vol. 15, pp. 537–542, 1967.

[8] D. W. Miller and M. K. Starr, *Executive Decisions and Operations Research*, Prentice-H. 1969, p. 607.

[9] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice H. 2010.

[10] S. Angilella, S. Greco, F. Lamantia, and B. Matarazzo, "Assessing non-additive utility for multicriteria decision aid," *Eur. J. Oper. Res.*, vol. 158, no. 3, pp. 734–744, Nov. 2004.

[11] J.L. Marichal and M. Roubens, "Determination of weights of interacting criteria from a reference set," *Eur. J. Oper. Res.*, pp. 641–650, 2000.

[12] E. Hinloopen, P. Nijkamp, and P. Rietveld, "Integration of ordinal and cardinal information in multi-criteria ranking with imperfect compensation," *Eur. J. Oper. Res.*, vol. 158, pp. 317–338, 2004.

[13] T. L. Saaty, "How to make a decision: The analytic hierarchy process," *Eur. J. Oper. Res.*, vol. 48, no. 1, pp. 9–26, Sep. 1990.

[14] W. S. University, "S.M.A.R.T. Objectives." [Online]. Available: http://wayne.edu/hr/leads/phase1/smart-objectives.php. [Accessed: 02-Oct-2013].

[15] A. Haarstrick and A. N. A. Lazarevska, "Multi-criteria decision making MCDM - A conceptual approach to optimal landfill monitoring," in *Third International Workshop "Hydro – Physico - Mechanics of Landfills*," 2009.

[16] I. IMS Global Learning Consortium, "IMS ePortfolio Best Practice and Implementation Guide," Version 1.0 Final Specification, 2005, http://www.imsglobal.org/ep/epv1p0imsep_bestv1p0.html.


© 2014 ACADEMY PUBLISHER

[17] P. Domingos and M. Pazzani, "On the optimality of the simple Bayesian classifier under zero-one loss," Mach *Learn*, vol. 29, no. 2–3, pp. 103–130, 1997.
[18] D. J.Hand and K. Yu, "Idiot's Bayes - not so stupid after all?," *Int. Stat. Rev*., vol. 69, no. 3, 2001.
[19] P-N. Tan, M. Steinbach, and V. Kumar, *Introduction to data mining*. Wesley: Pearson Addison, 2006.
[20] J. R. Quinlan, "C4.5: Programs for Machine Learning," *Mach. Learn*., vol. 16, no. 3, pp. 235–240, 1993.
[21] V. Vapnik, "The Nature of Statistical Learning Theory," *Springer-Verlag*, New York, 2000.

**Hoang Nam Ho** is a PhD student at the research Laboratory L3i "Informatics, Image, Interactions", University of La Rochelle. His researches concern the adaptive execution in the context of interactive applications. He uses data analysis technologies, especially data mining and decision-making, in order to perform the adaptation during application execution.

**Mourad Rabah** obtained his PhD in computer science at LAAS-CNRS laboratory in Toulouse (France) in the dependability domain. Since 2002, he is an Associate Professor at University of La Rochelle within the L3i Laboratory. His current work deals with Interactive Adaptive Systems where the system adapts its execution to users interactions and behaviours. He explores the application of fault tolerant technics in order to improve application scenario structuring and adaptation decision-making in interactive systems.

**Pascal Estraillier** is a full professor at the Computer Sciences department and the research Laboratory L3i "Informatics, Image, Interactions" at the University of La Rochelle. He is also scientific adviser in the directorate "National Management of Research and Innovation" at the French Ministry of Research, in charge of ICT area. His researches concern the architecture of software components in distributed and cooperative systems. He applies its results on Multi-agent Paradigm and uses formal specification theories in order to validate the behaviour and the interactions between components and to manage interoperability constraints. The results are mainly applied to the games, serious game and e-learning domains in many research projects founded by the E.U, the french A.N.R, the regional council, and several companies (France Telecom, XD-Production,...).

**Samuel Nowakowski** is an associate professor at University of Lorraine as member of LORIA (Lorraine Research Laboratory in Computer Science and its Applications), a research unit (UMR 7503) common to CNRS, the University of Lorraine and INRIA. His current research activities are focused on robust Control methods for recommender systems, topology and social web modelling. He is involved in research projects dealing with ePortfolio in higher education and adult education at regional and European levels. He is involved in research activities dealing with identity through digital technologies.

© 2014 ACADEMY PUBLISHER