# Using Possibilistic Networks to Compute Learning Course Indicators

Guillaume Petiot(✉)

Catholic Institute of Toulouse, 31 rue de la Fonderie, 31000 Toulouse, France
guillaume.petiot@ict-toulouse.fr

**Abstract.** E-learning systems generate more and more data that can be used to improve pedagogy. They can also provide a better understanding of a student's learning style. As a result, it is possible to propose a differentiated pedagogy which takes into account learners' needs. The aim of this research is to build course indicators by using expert knowledge in order to provide a synthesis of information about students. As knowledge is often imprecise and uncertain, we used possibility theory to represent knowledge through a possibilistic network. Firstly, we used a message passing algorithm to compute learning course indicators, then we proposed several improvements. Indeed, the use of uncertain gates allows us to generate automatically Conditional Possibility Tables (CPT) instead of eliciting all parameters. Next, we compiled the junction tree of the possibilistic network in order to improve computation time. We compared our compiling approach with message passing inference. A decision support system is generated automatically at the end of the computations. The indicators are presented in a decision support system in which color codes illustrate certainty.

**Keywords:** Compiling knowledge · Decision Making · Education · Possibilistic networks · Possibility theory · Uncertainty

## 1 Introduction

E-learning platforms generate a huge amount of data that cannot be fully interpreted by teachers. So researchers have tried to use the AI tools as a solution to this problem [3,6,13,26]. Several applications of AI have already been proposed [1,31] in order to personalize students' learning experience, to develop adaptive learning, model learning behaviour, improve decision-making, analyze the learners' sentiments, give recommendations, perform a classroom monitoring, propose an intelligent tutoring systems, etc. The aforementioned researchers also tried to highlight the students risking dropping out or failing at the examination. They made use of Bayesian networks, neural networks, support vector machines, reinforcement learning, deep learning, and so on.

We propose in this study to compute learning course indicators by using teachers' knowledge. Our previous paper published in [24] presented an overview of our approach. We present here a more detailed description of our research and particularly of the algorithms used to compute indicators. We provide more examples and results obtained by using our solution.


A. P. Rocha et al. (Eds.): ICAART 2021, LNAI 13251, pp. 135–157, 2022.
https://doi.org/10.1007/978-3-031-10161-8_7

Defining indicators by using knowledge leads us to consider several problems. Indeed, expert knowledge is often imprecise, uncertain and sometime incomplete. Possibility theory, proposed by L. A. Zadeh [30] in 1978 after the fuzzy set theory in 1965, can be used to solve this problem. The indicators can be represented by a Directional Acyclic Graph that shows the causal link between the variables. We can use a possibilistic network [4] which is an adaptation of the Bayesian network [18,20] to possibility theory. In the possibilistic network, we need to define for each variable a CPT. After the injection of evidence, which is new information in the network, we can compute its effect on the indicators. The problem however arises when the number of parents of a variable grows because the number of parameters of the CPT grows exponentially. That is why it may be more appropriate to use uncertain logical gates [10]. Moreover, they allow us to represent unknown variables of a complex system by adding a leakage variable. The authors of [10] proposed to encode variables with several ordered states. For example the states low, medium and high of a variable can be encoded into a scale of numerical values, 0, 1 and 2.

Several algorithms of exact inference can be used in a possibilistic network. They are inspired by the algorithms that exist for Bayesian networks (e.g., the message propagation inference algorithm, loop cut-set conditioning proposed by J. Pearl [21,22], arc reversal [27,28], the variable elimination [32], Shenoy-Shafer [29], Hugin [14], etc.). Most of the inference algorithms based on a junction tree share an exponential computation time which is proportional to the largest clique in the junction tree.

In this paper, we would like to perform an experimentation of indicator calculation by using possibilistic networks and uncertain gates. To improve the running time of the inference, we propose to use a new approach based on the compiling of the junction tree. We will compare this solution to the traditional message passing algorithm.

In our experimentation, we will use an existing dataset, fully anonymized, made up of Moodle logs for a course of spreadsheet, and some external information, such as attendance and results at the examination. The knowledge of the course indicators is provided by the teachers and extracted by data mining [23].

To do this, we will first present possibility theory and uncertain gates. Then, we will describe our message passing algorithm and compile our possibilistic networks. Finally, we will discuss our results.

## 2 Possibility Theory

Possibility theory was proposed in 1978 by L.A. Zadeh [30] as an extension of the fuzzy set theory. Possibility theory, in which imprecise knowledge can be represented by a possibility distribution $\pi$, deals with the management of uncertainty and provides two dual operators, the possibility measure $\Pi$ and the necessity measure $N$ from $P(\Omega)$ in $[0,1]$, as presented by the authors [11]. $\Omega$ is the universe of the discourse and $P(\Omega)$ is the set of all subsets of $\Omega$. The possibility distribution must be normalized ($\exists x \in \Omega$ such as $\pi(x) = 1$). The possibility measure and the necessity measure are defined as follows [24]:

$$\forall A \in P(\Omega), \Pi(A) = \sup_{x \in A} \pi(x) \tag{1}$$

$$\forall A \in P(\Omega), N(A) = 1 - \Pi(\neg A) = \underset{x \notin A}{inf}\ 1 - \pi(x) \tag{2}$$

Possibility theory is not additive but maxitive. We have the following properties:

$$\forall A, B \in P(\Omega), \Pi(A \cup B) = \max(\Pi(A), \Pi(B)). \tag{3}$$

$$\forall A, B \in P(\Omega), \Pi(A \cap B) \leq \min(\Pi(A), \Pi(B)). \tag{4}$$

We also have the following properties for the dual necessity measure:

$$\forall A, B \in P(\Omega), N(A \cap B) = \min(N(A), N(B)). \tag{5}$$

$$\forall A, B \in P(\Omega), N(A \cup B) \geq \max(N(A), N(B)). \tag{6}$$

E. Hisdal [12] proposed a solution to compute the possibility of a variable $A$ given the variable $B$, generalized by D. Dubois and H. Prade [11]:

$$\Pi(A|B) = \begin{cases} \Pi(A, B) & \text{if } \Pi(A, B) < \Pi(B), \\ 1 & \text{if } \Pi(A, B) = \Pi(B). \end{cases} \tag{7}$$

Possibilistic networks [4,5] are the counterpart of Bayesian networks in possibility theory and can be defined as follows:

**Definition 1.** *A possibilistic network $(G, \Sigma)$ is defined when the following elements are given:*

- *A Directional Acyclic Graph G, $G = (V, E)$, where $V$ is the set of nodes of the graph and $E$ the edges of G;*
- *The set of all conditional possibility distributions noted $\Sigma$. All conditional possibility distributions must be normalized;*
- *The factoring property, where the possibility $\Pi(V)$ can be factorized toward the graph* G*:*

$$\Pi(V) = \bigotimes_{X \in V} \Pi(X/Pa(X)). \tag{8}$$

*The function $Pa(X)$ returns the parents of the variable* X.

There are two classes of possibilistic networks. Min-based possibilistic networks (qualitative) if $\bigotimes$ is the minimum and the Product-based possibilistic networks (quantitative) if $\bigotimes$ is the product. In min-based possibilistic network the possibility distribution is a mapping from $\Omega$ to an ordinal scale leading to consider only the ordering of the values. The product-based possibilistic network is very similar to the Bayesian network in the sense that the possibilistic scale is numerical and can be combined by using arithmetic operators. In this case the possibility degree can be interpreted in the ranking scale $[0, 1]$. In this research, we will use a min-based possibilistic network because we have chosen to compare the possibilistic values instead of using an intensity scale in $[0, 1]$.

Uncertain logical gates were proposed for the first time by the authors of [10] to compute automatically the CPTs in a possibilistic network. They are the counterpart of noisy gates in possibility theory. They use the property of the Independence of Causal Influence to provide a model that represents uncertainty between a set of causal variables $X_1, ..., X_n$ and an effect variable $Y$. This model is built by introducing an intermediate variable $Z_i$ between each causal variable and the effect variable.

This allows us to represent two possible behaviors: inhibition and substitution. The former appears when a cause is met and the effect variable $Y$ is not produced. The latter takes place when a cause is not met and the variable $Y$ is produced. In fact, these behaviours are due to inhibitor parameters $\kappa$ and substitute parameters noted $s$.

The possibilistic model with the ICI is summarized in the following Fig. 1:

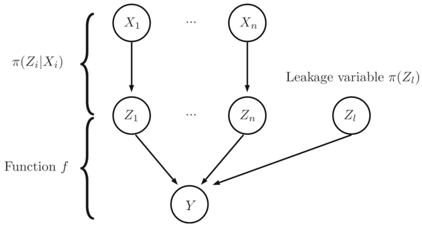



**Fig. 1.** Possibilistic model with ICI [24].

In this model, there is a deterministic function $f$ that combines the influence of the variables $Z_i s$ to compute the variable $Y$: $Y = f(Z_1, ..., Z_n)$. To represent the unknown knowledge, we can add a leakage variable $Z_l$. This new variable represents all unknown knowledge and brings forth an uncertain leaky model [10]. For all instantiations $y$ of the variable $Y$, $x_i$ of the variables $X_i$, $z_i$ of the variables $Z_i$ and $z_l$ of the variable $Z_l$, we obtain the following equation for a Min-based possibilistic network [24]:

$$\pi(y|x_1, ..., x_n) = \bigoplus_{z_1,...,z_n,z_l:y=f(z_1,...,z_n,z_l)} \bigotimes_{i=1}^{n} \pi(z_i|x_i) \otimes \pi(z_l) \tag{9}$$

The $\otimes$ is the minimum and $\oplus$ is the maximum. There are several possible functions for $f$, for example AND, OR, NOT, INV, XOR, MAX, MIN, MEAN, linear combination, etc. To compute the CPT from the equation, we must define $\pi(Z_i|X_i)$, $\pi(Z_l)$, and we must choose a function $f$. In our experimentation, all variables have three ordered states of intensity: low, medium and high. We propose to encode these states as the authors [10]: 0 for low, 1 for medium and 2 for high. Here is an example of a table for $\pi(Z_i|X_i)$ (Table 1):

**Table 1.** Possibility table for 3 ordered states [24].

| $\pi(Z_i\|X_i)$ | $x_i = 2$ | $x_i = 1$ | $x_i = 0$ |
|---|---|---|---|
| $z_i = 2$ | 1 | $s_i^{2,1}$ | $s_i^{2,0}$ |
| $z_i = 1$ | $\kappa_i^{1,2}$ | 1 | $s_i^{1,0}$ |
| $z_i = 0$ | $\kappa_i^{0,2}$ | $\kappa_i^{0,1}$ | 1 |

In the above table, $\kappa_i$ represents the possibility that an inhibitor exists and $s_i$ the possibility that a substitute exists. If a cause of weak intensity cannot produce a strong effect, then all $s_i = 0$. So in the above example, there are 6 parameters at most per variable and 2 parameters for $\pi(Z_l)$. Another constraint is that $\kappa_i^{1,2} \geq \kappa_i^{0,2}$.

The authors of [10] proposed to use as the function $f$ the function MIN and MAX leading to the connectors uncertain MIN ($\perp$) and uncertain MAX ($\top$). We will use these connectors in our experimentation. We will also use a weighted average function (WAVG) and a MYCIN Like connector ($\hbar$) [23,24]. The result of the function $f$ must be in the domain of $Y$. We can see that the connectors uncertain MIN and uncertain MAX satisfy this property. Nevertheless, this is not the case for the weighted average function and the MYCIN Like function. Our solution is to use a scaling function $f_s$, such that $f = f_s \circ g$ where $g$ is the weighted average function or the MYCIN Like function. If we consider the example of the weighted average function, then $g(z_1, ..., z_n) = \omega_1 z_1 + ... + \omega_n z_n$. The parameters $\omega_i$ are the weights of the weighted average. If all weights $\omega_i$ are equal to $\frac{1}{n}$, then we calculate the average of the intensities. If all weights $\omega_i = 1$, then we make the sum of the intensities (connector $\sum$). If $(\epsilon_0, \epsilon_1, ..., \epsilon_{m-1})$ are the $m$ ordered states of $Y$, then the function $f_s$ can be as follows:

$$f_s(x) = \begin{cases} \epsilon_0 & \text{if } x \leq \theta_0 \\ \epsilon_1 & \text{if } \theta_0 < x \leq \theta_1 \\ \vdots & \vdots \\ \epsilon_{m-1} & \text{if } \theta_{m-2} < x \end{cases} \tag{10}$$

The parameters $\theta_i$ allow us to adjust the behaviour of $f_s$. If the values of $\theta_i$ are well defined $\theta_i = i + \frac{1}{2}$, then we perform a rounding to the nearest value.

## 3 Message Passing Inference

The message passing algorithm for possibilistic networks was inspired by the algorithm proposed for Bayesian networks [18,20]. We used an algorithm of message passing in the junction tree [5,17], which contains two kinds of nodes: cliques and separators. To extract the cliques, we first compute the moral graph, then we compute the triangulated graph by using Kjaerulff's algorithm [15], and finally, we compute the maximal spanning tree by using Kruskal's algorithm [16]. The propagation of evidence is performed by using three phases. The first is the initialization of evidence in the graph, then we perform the phase of collect that consists in propagating evidence from the leaves to the root. The last phase, called distribution, is the propagation from the root to the leaves. Then, we can compute the possibility of the variables. During the initialization, all separators are initialized to 1 and all variables are affected to only one clique. The potential of the cliques can be computed as follows:

$$w_{C_i}(v) = \bigotimes_{X \in C_i, X \notin C_j, j<i} \pi(X = v_k / pa(X)). \tag{11}$$

where $v = (v_1, ..., v_{n_i})$ is an instantiation of the variables of the clique $C_i$, $n_i$ is the size of the clique $C_i$, $X = v_k$ is the instantiation of the variable X in v, and $pa(X)$ is the instantiation of the parents of the variable X. If $C_i$ is the first clique, all variables of the clique are taken into account to compute $w_{C_i}(v)$. Message passing between the cliques $C_i$ and $C_j$ requires the marginalization of the variables of the clique $C_i$ regarding the variables in the separator $S_{i,j}$ of the two cliques: $w^*_{S_{i,j}}(s) = \bigoplus_{v \in C_i \setminus S_{i,j}} w_{C_i}(s, v)$, where $s$ is an instantiation of the separator $S_{i,j}$ and $v$ is an instantiation of the variables in $C_i \setminus S_{i,j}$. Then, we update the possibility table of the clique $C_j$: $w^*_{C_j}(s, v) = w_{C_j}(s, v) \otimes w^*_{S_{i,j}}(s)$ where $s$ is an instantiation of the separator $S_{i,j}$ and $v$ is an instantiation of $C_j \setminus S_{i,j}$. To compute the possibility of a variable given evidence $\epsilon$, we compute the combination of the possibility of the variable and the possibility of evidence by using conditioning. When all evidence is injected in the network, we apply the propagation algorithm:

**Algorithm 1.** Possibilistic message passing.

**Input** : The evidence $\epsilon$ and a root cluster;
**Output:** The conditional possibility of all cluster given $\epsilon$ : $\forall_i\ \pi(C_i|\epsilon)$;

/* Initialization
**forall** $S_{i,j}$ ***and*** *$s$ instantiation of $S_{i,j}$* **do**
  $w^{[0]}_{S_{i,j}}(s) = 1$
**forall** $C_i$ ***and*** $v = (x_1, ..., x_{n_i})$ *instantiation of $C_i$* **do**
  $w^{[0]}_{C_i}(v) = \bigotimes_{X \in C_i, X \notin C_j, j<i} \pi(X = x_k/pa(X))$

/* Collect
**forall** *$C_i$ from the leaves to the root with a unique adjacent clique $C_j$ of potential $w^{[1]}_{C_j}$ not yet computed* **do**
  Marginalize $C_i$ on $C_i \setminus S_{i,j}$
  **forall** *$s$ (instantiation of $S_{i,j}$)* **do**
    Compute $w^{[1]}_{S_{i,j}}(s)$
  **forall** *$s$ instantiation of $S_{i,j}$ and $u$ instantiation of $C_j \setminus S_{i,j}$* **do**
    $w^{[1]}_{C_j}(s, u) = w^{[0]}_{C_j}(s, u) \otimes w^{[1]}_{S_{i,j}}(s)$

/* Distribution
**forall** *$C_i$ from the root to the leaves with a unique adjacent clique $C_j$ of potential $w^{[2]}_{C_j}$ not yet computed* **do**
  Marginalize $C_i$ on $C_i \setminus S_{i,j}$
  **forall** *$s$ instantiation of $S_{i,j}$* **do**
    Compute $w^{[2]}_{S_{i,j}}(s)$
  **forall** *$s$ instantiation of $S_{i,j}$ and $v$ instantiation of $C_j \setminus S_{i,j}$* **do**
    $w^{[2]}_{C_j}(s, v) = w^{[1]}_{C_j}(s, v) \otimes w^{[2]}_{S_{i,j}}(s)$

This algorithm allows us to compute the possibility measure of each variable by marginalizing the cluster which contains the variable and its parents. Then, we can deduce the dual necessity measure. This second measure represents certainty.

## 4 Compiling Possibilistic Networks

In the previous section, we considered the propagation of evidence in a possibilistic network by using a message passing algorithm. Nevertheless, we can also compute possibility and necessity measures by using a different approach. Indeed, the junction tree of a possibilistic network, can be compiled before evaluating the effects of evidence. The same reasoning as in compiling Bayesian networks [8] can be used. The compiling of possibilistic network has been introduced by the authors of [25] but the compiling of the junction tree is not presented. The authors of [19] proposed an algorithm to differentiate the arithmetic circuit of a junction tree. We propose the counterpart of this approach for possibilistic networks.

As for the multilinear function of Bayesian networks, we propose to represent the possibilistic network by using a function $f$. This function can be defined as follows [24]:

**Definition 2.** *If $P$ is a possibilistic network, $V = v$ the instantiations of the variables of the possibilistic network and $U = u$ the consistent instantiation of the parents of a variable $X$ with the instantiation $X = x$, then the function $f$ of $P$ is:*

$$f = \bigoplus_{v} \bigotimes_{xu \sim v} \lambda_x \otimes \theta_{x|u} \tag{12}$$

*In the above formula, $xu$ denotes the instantiation of the family of $X$ and its parents $U$ compatible with the instantiation $v$. $\lambda_x$ are evidence indicators and $\theta_{x|u}$ are the parameters of the CPTs. In fact, for all network CPT parameters of $\pi(X|U)$, we define a parameter $\theta_{x|u}$ where $u$ is an instantiation of $U$, the parents of the variable $X$ and $x$ an instantiation of the variable $X$.*

*The operator $\bigoplus$ can be the function maximum and $\bigotimes$ the function minimum if we consider a qualitative possibilistic network.*

We can study, as an example, the following possibilistic network (Table 2):

**Table 2.** Example of a possibilistic network $A \to B \to C$ [24].

| A | B | |
|---|---|---|
| true | true | $\theta_{b|a}$ |
| true | false | $\theta_{\bar{b}|a}$ |
| false | true | $\theta_{b|\bar{a}}$ |
| false | false | $\theta_{\bar{b}|\bar{a}}$ |

| A | |
|---|---|
| true | $\theta_a$ |
| false | $\theta_{\bar{a}}$ |

| B | C | |
|---|---|---|
| true | true | $\theta_{c|b}$ |
| true | false | $\theta_{\bar{c}|b}$ |
| false | true | $\theta_{c|\bar{b}}$ |
| false | false | $\theta_{\bar{c}|\bar{b}}$ |

In this case the function $f$ is:

$$\begin{aligned} f = \lambda_a \otimes \lambda_b \otimes \lambda_c \otimes \theta_a \otimes \theta_{b|a} \otimes \theta_{c|b} \\ \oplus \lambda_a \otimes \lambda_b \otimes \lambda_{\bar{c}} \otimes \theta_a \otimes \theta_{b|a} \otimes \theta_{\bar{c}|b} \\ \vdots \\ \oplus \lambda_{\bar{a}} \otimes \lambda_{\bar{b}} \otimes \lambda_{\bar{c}} \otimes \theta_{\bar{a}} \otimes \theta_{\bar{b}|\bar{a}} \otimes \theta_{\bar{c}|\bar{b}} \end{aligned} \tag{13}$$

The evidence corresponds to an instantiation of several variables of the possibilistic network. The value of $f(e)$ can be computed by replacing the evidence indicator consistent with the evidence $e$ by 1 or by 0. This assumption leads us to the following definition:

**Definition 3.** *If the evidence $e$ is an instantiation of several variables, then we have the property $f(e) = \pi(e)$.*

We consider the following example:

**Table 3.** Example of a possibilistic network $A \rightarrow B$ [24].

| A | B | |
|---|---|---|
| true | true | 1 |
| true | false | 0.2 |
| false | true | 0.1 |
| false | false | 1 |

| A | |
|---|---|
| true | 1 |
| false | 0.1 |

If the evidence is $\bar{a}$, then we obtain $\lambda_a = 0$, $\lambda_{\bar{a}} = 1$, $\lambda_b = 1$, $\lambda_{\bar{b}} = 1$ and the computation of $f(e)$ is: $f(\bar{a}) = f(\lambda_a = 0, \lambda_{\bar{a}} = 1, \lambda_b = 1, \lambda_{\bar{b}} = 1) = \theta_{\bar{a}} \otimes \theta_{b|\bar{a}} \oplus \theta_{\bar{a}} \otimes \theta_{\bar{b}|\bar{a}} = 0.1 \otimes 0.1 \oplus 0.1 \otimes 1.0 = 0.1$. The evaluation of $f$ leads us to compute $\pi(e)$.

We can compute the possibility of the variable $X$ given the evidence $e$ by using the conditioning of Eq. 7.

If the variable $X$ has $n$ states and $x$ is one of its states, then we must discuss two cases: if $X$ is not in the evidence $e$, then $\pi(x|e)$ can be computed by using $\pi(x, e) = f(e, 1_{\lambda_x})$ with $1_{\lambda_x} = (\lambda_{x_1} = 0, ..., \lambda_x = 1, ..., \lambda_{x_n} = 0)$. Otherwise, if $X$ is in $e$, we have to compute $\pi(x|e - X)$. This leads us to the definition of Evidence Retraction [8] in possibility theory. In fact, $e - X$ denotes the instantiation $e$ without the instantiation of the variable X.

If the operator $\bigoplus$ is the function maximum and $\bigotimes$ is the function minimum, then the function $f$ can be transformed into a MIN-MAX circuit. We propose the following definition of a MIN-MAX circuit:

**Definition 4.** *If we have a min-based possibilistic network $P = (G, \Sigma)$ and its function $f$, the MIN-MAX circuit of the function $f$ is a directed acyclic graph. The latter is built by considering a root node which is the result of $f$ and the child nodes that are the functions MIN and MAX used in $f$. The leaf nodes are evidence indicators $\lambda$ and network parameters $\theta$.*

It is possible to represent a function $f$ of exponential size by a MIN-MAX circuit of linear size [25]. This improvement results in reducing the number of operations to perform to compute $f$. Moreover, it can also reduce memory used and computation time. We present an example of a MIN-MAX circuit in the following Fig. 2:

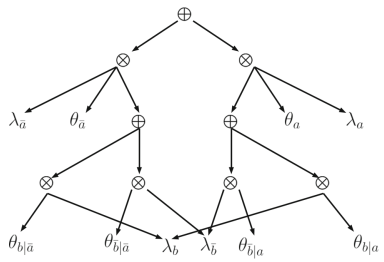



**Fig. 2.** MIN-MAX circuit of the example [24].

We have chosen to perform the factorization of the function $f$ and then to use the junction tree method. To compile a Bayesian network under evidence, we generate an arithmetic circuit and we differentiate the circuit in order to obtain all posterior probabilities $p(x|e)$. The differentiation is very easy with the arithmetic circuit in probability theory. So we propose to use this advantage to find an algorithm to compute posterior possibilities from the MIN-MAX circuit of a junction tree in possibility theory. Our approach consists of three steps: encoding the MIN-MAX circuit into an arithmetic circuit, differentiating the arithmetic circuit to find an algorithm, and finally performing the inverse encoding of the previous algorithm.

**Definition 5.** *We obtain the multi-linear function $f'$ of $f$ by applying the $\Delta$ operator which replaces the $\otimes$ by multiplications and the $\oplus$ by additions. The circuit generated by applying the same operator to the MIN-MAX circuit is the arithmetic circuit of $f'$. The operator $\Delta^{-1}$ is the inverse operator which replaces the multiplication by $\otimes$ and the addition by $\oplus$.*

The following example summarizes this process (Fig. 3):

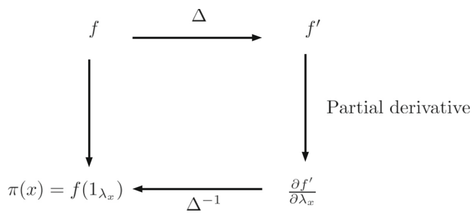



**Fig. 3.** Use of operators $\Delta$ and $\Delta^{-1}$.

We can deduce the proof easily if we consider that the partial derivative way replaces the evidence indicator of $x$ by 1 and substitutes 0 for any other state of this variable. We can verify that the result is the same when we apply $\Delta^{-1}$ to $\frac{\partial f'}{\partial \lambda_x}$ and when we compute $f(1_{\lambda_x})$. This leads us to compute $\pi(x)$.

If we consider the example of Table 3, we obtain the following function:

$$f = \lambda_a \otimes \theta_a \otimes (\lambda_b \otimes \theta_{b|a} \oplus \lambda_{\bar{b}} \otimes \theta_{\bar{b}|a}) \oplus \lambda_{\bar{a}} \otimes \theta_{\bar{a}} \otimes (\lambda_b \otimes \theta_{b|\bar{a}} \oplus \lambda_{\bar{b}} \otimes \theta_{\bar{b}|\bar{a}}) \quad (14)$$

We apply the $\Delta$ operator and after the transformation, we obtain the following polynomial:

$$f' = \lambda_a \theta_a (\lambda_b \theta_{b|a} + \lambda_{\bar{b}} \theta_{\bar{b}|a}) + \lambda_{\bar{a}} \theta_{\bar{a}} (\lambda_b \theta_{b|\bar{a}} + \lambda_{\bar{b}} \theta_{\bar{b}|\bar{a}}) \quad (15)$$

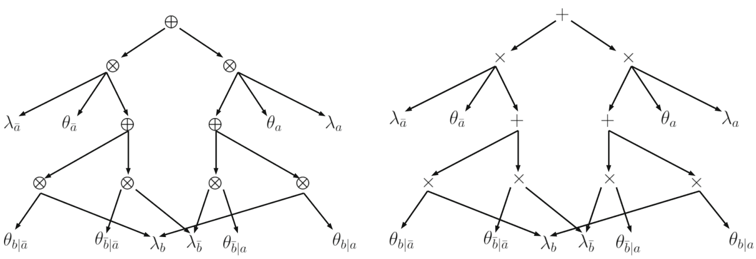
For example, if we suppose that $e = b$, then $f'(e) = f'(\lambda_b = 1; \lambda_{\bar{b}} = 0; \lambda_a = 1; \lambda_{\bar{a}} = 1)$. To compute $\pi(a, e)$ we must at first compute $\frac{\partial f'(e)}{\partial \lambda_a}$ because $a$ is not in $e$. We obtain the following result:

$$\frac{\partial f'(e)}{\partial \lambda_a} = \theta_a \theta_{b|a} \quad (16)$$

To obtain $\pi(a, e)$ we apply the inverse operation $\Delta^{-1}$ that replaces the additions by $\oplus$ and the multiplications by $\otimes$ in the above equation:

$$\pi(a, e) = \theta_a \otimes \theta_{b|a} \quad (17)$$

We can also encode the MIN-MAX circuit into an arithmetic circuit as follows (Fig. 4):



(a) MIN-MAX circuit. (b) Arithmetic circuit.

**Fig. 4.** Arithmetic circuit of a MIN-MAX circuit [24].

We propose to build the MIN-MAX circuit of a junction tree obtained from a possibilistic network. To do this we must first select a root node which is the result of $f$. The children of the output node $f$ are the $\otimes$ nodes of the root cluster. We add a $\oplus$ node for each instantiation of a separator between a current cluster and a child cluster. We add a $\otimes$ node for each instantiation of the variables of a cluster. The children of the $\oplus$ nodes are compatible nodes generated by the child clusters and the children of a $\otimes$ node are compatible nodes generated by the child separators. We have only one node $\lambda_x$ for each instantiation of a variable X and only one node $\theta_{x|u}$ for each instantiation of a node $X$ and its parents $U$. Moreover each variable, evidence indicators $\lambda$, and network parameters $\theta_{x|u}$ are affected to only one cluster.

We propose now to differentiate the arithmetic circuit of $f'$ generated from the MIN-MAX circuit of $f$ by using the operator $\Delta$. If $v$ is the current node and $P$ represents the parents of $v$, then we can compute $\frac{\partial f'}{\partial v}$ by using the chain rule [24]:

$$\frac{\partial f'}{\partial v} = \sum_{p \in P} \frac{\partial f'}{\partial p} \frac{\partial p}{\partial v} \tag{18}$$

If one of the parents noted $p$ of $P$ has $n$ other children $v_i$ different from the node $v$, there are several cases to discuss:

– If $v$ is the root node then $\frac{\partial f'}{\partial v} = 1$;
– If $p$ is an addition node then $\frac{\partial p}{\partial v} = \frac{\partial(v + \sum_{i=1}^{n} v_i)}{\partial v} = 1$;
– If $p$ is a multiplication node then $\frac{\partial p}{\partial v} = \frac{\partial(v \prod_{i=1}^{n} v_i)}{\partial v} = \prod_{i=1}^{n} v_i$.

As a result, by using the operator $\Delta^{-1}$ we obtain the following step to evaluate the MIN-MAX circuit of a junction tree [24]. We need two registers for each node to perform the computation noted $u$ and $d$.

1. Upward: compute the value of the node $v$ and store it in $u(v)$;
2. If $v$ is the root then set $d(v) = 1$ else set $d(v) = 0$;
3. Downward: for each parent $p$ of the node $v$ compute $d(v)$ as follows:
   (a) if $p$ is a node $\oplus$:
$$d(v) = d(v) \bigoplus d(p) \tag{19}$$
   (b) if $p$ is a node $\otimes$:
$$d(v) = d(v) \bigoplus \left[ d(p) \bigotimes \left[ \bigotimes_{i=1}^{n} u(v_i^p) \right] \right] \tag{20}$$
   The nodes $v_i^p$ are the other children of $p$;

As a result, we obtain the following algorithm:

**Algorithm 2.** Junction tree compiling algorithm.

**Input :** The MIN-MAX circuit $\Gamma$ and its root node $r$
**Output:** The computing of $u$ and $v$ for all nodes of $\Gamma$

```
Initialize(u)
Upward(r)
forall v ∈ Γ do
    if v == r then
        set d(v) = 1
    else
        set d(v) = 0
Downward(r)
```

The recursive function Upward is described in the following algorithm:

**Algorithm 3.** Upward.

```
Input  : node e
Output: The computing of u(e)
if NumberOfChildren(e) == 0 then
    return u(e)
else
    if Operator(e) == ⊕ then
        u(e) = 0
    else
        u(e) = 1
    forall c ∈ ChildrenOf(e) do
        if Operator(e) == ⊕ then
            u(e) = u(e) ⊕ Upward(c)
        else
            u(e) = u(e) ⊗ Upward(c)
    return u(e)
```

In this algorithm, the function NumberOfChildren$(e)$ returns the number of the children of $e$. The function ChildrenOf$(e)$ returns the set of children of the node $e$ and the function Operator$(e)$ returns the type of the node of $e$: $\otimes$ or $\oplus$. The recursive function Downward is as follows:

**Algorithm 4.** Downward.

```
Input  : node e
Output: The computing of d(c) for all children c of e
if NumberOfChildren(e)! = 0 then
    forall c ∈ ChildrenOf(e) do
        if Operator(e) == ⊕ then
            d(c) = d(e) ⊕ d(c)
        else
            t = d(e)
            forall b ∈ ChildrenOf(e) and b! = c do
                t = t ⊗ u(b)
            d(c) = t ⊕ d(c)
    forall c ∈ ChildrenOf(e) do
        Downward(c)
```

# 5 Experimentation

## 5.1 Presentation

In our experimentation, we used an existing anonymized dataset for a Spreadsheet course at bachelor level proposed in face-to-face learning enriched by an online supplement on Moodle. This dataset was compiled by gathering all data of logs in a table. Then a process of anonymization was performed.

For example, we used the data of Moodle, such as quiz results, sources consulted, wiki consulted, forum participation,... and external data such as attendance, groups, etc. The quiz questions were categorized by skills. When data are missing there are several methods to estimate the missing data in education [2,7,9]. For example one can use mean imputation, regression imputation, Maximum Likelihood Expectation-Maximization (EM) imputation, multiple imputation, hot deck imputation, zero imputation (replace missing values by 0), iterative PCA imputation, ... We chose iterative PCA imputation also called EM-PCA because this method takes into account the profile of the students to provide an estimation of the missing data.

The knowledge about the indicators was provided by the teachers and extracted from the data. We used exploratory statistics approach such as correlation graph, Principal Component Analysis, and Ascending Hierarchical Classification to extract knowledge from the data.

To represent this knowledge we have chosen to use a DAG (Fig. 5):

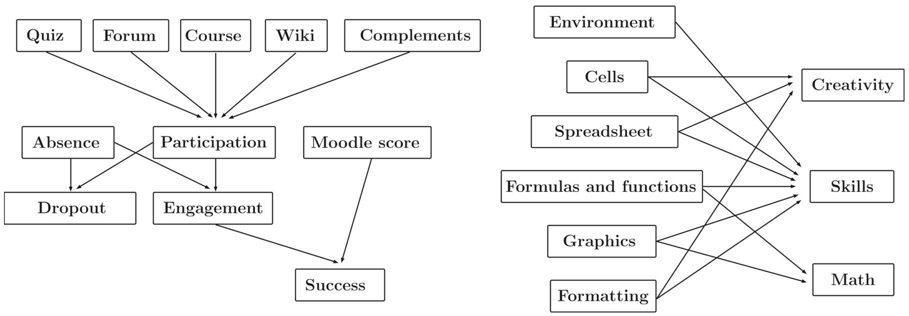
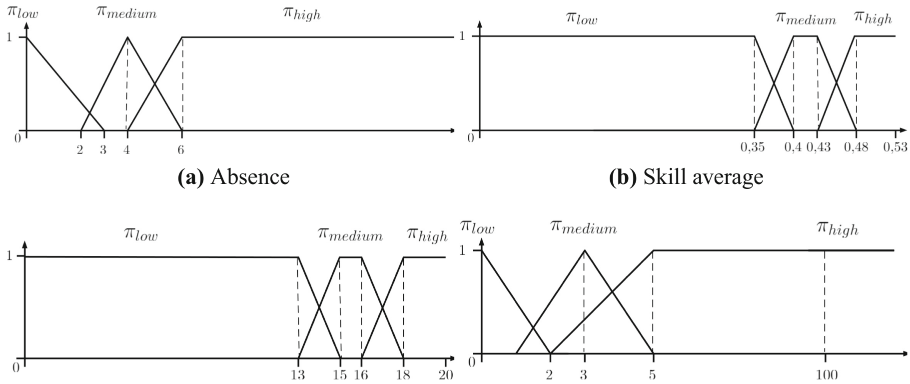



**Fig. 5.** Modeling of knowledge by a DAG [24].

The variables in the DAG have 3 ordered modalities (low, medium, high) encoded with the numerical values (0,1,2). The description of the indicators by teachers is often imprecise, so we used a possibility distribution to represent each state of a variable. In fact, the variables are linguistic variables and we used the following possibility distributions to compute the evidence from the data (Fig. 6):




**(a)** Absence **(b)** Skill average

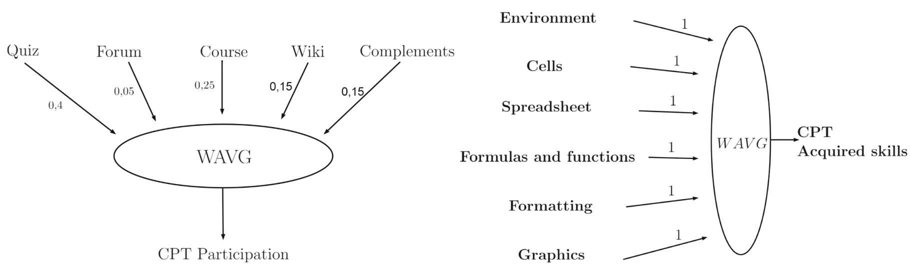
**(c)** Quizz score **(d)** Participation (number of visit)

**Fig. 6.** Possibility distributions of the variables.

The possibilistic networks require defining all CPTs but if the number of parents of a variable is high, then there are too many parameters to elicit. The use of uncertain gates provides a solution to this problem. Indeed, the CPTs are computed automatically.

We used the uncertain MIN connector ($\perp$) for conjunctive behavior and the uncertain hybrid connector ($\hbar$) for indicators which need a compromise in case of conflict and a reinforcement if the values are concordant.

We propose to use a connector WAVG to merge the information about the sources consulted in Moodle in order to build an indicator of participation which takes into account their importance. The weights were provided by the teachers. We also computed an indicator of acquired skills by using the WAVG connector with all weights equal to 1. The name of this connector is $\sum$. We summarize in the following figure the use of the WAVG connector to compute the indicators (Fig. 7):




**(a)** Indicator of participation. **(b)** Indicator of acquired skills.

**Fig. 7.** Weights of the WAVG connectors [24].

As a result, we obtain the following model (Fig. 8):

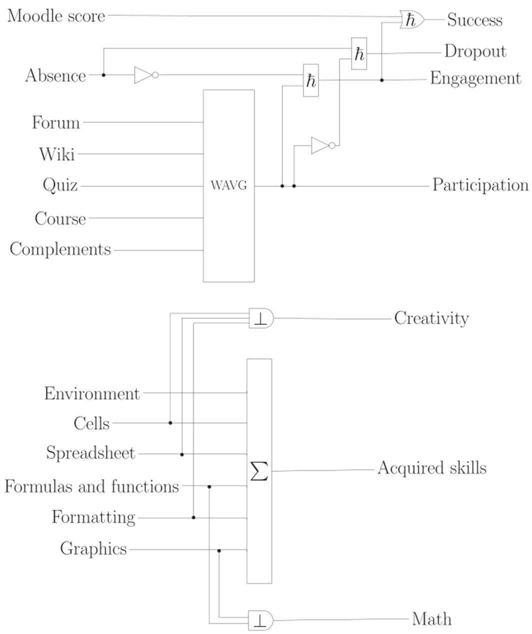
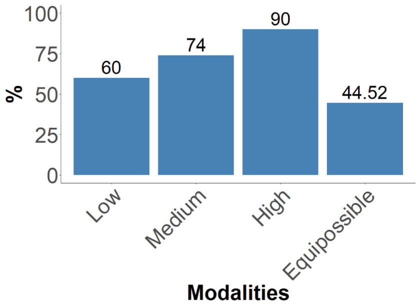



**Fig. 8.** Knowledge modeling with uncertain connectors [24].

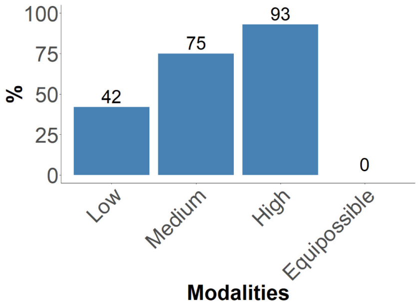
The learning course indicators of our experimentation are computed after several processing operations. Before the propagation of new information, we have to compute the CPTs of all the uncertain gates. Then we compile the junction tree of the possibilistic network and finally we perform the initialization of evidence before applying the upward pass and downward pass. As a result, we obtain for each state of the learning course indicator a possibility measure and a necessity measure. We have compared this approach with the message passing algorithm studied in our previous research [23].

### 5.2 Results

We have performed several improvements of the initial approach based on a possibilistic network. We have elicited all CPT parameters and performed the computation of the indicators by using the message passing algorithm. The first improvement proposed was to use uncertain gates to avoid the eliciting of all the CPT parameters. Then the computation time was improved by compiling the junction tree of the possibilistic networks. We compared the compilation of the possibilistic networks and the message passing algorithm. As expected, the results of the indicators in both approaches were identical. For example, the indicator of success deals with the prediction of a student's success at the examination. We have computed the percentage of success for each state of the indicator of success. When all states of a variable have a possibility equal to 1 they are equipossible. We obtain the results of Fig. 9 by using the compilation of the possibilistic networks. On the x-axis we have added the number of equipossible results after the modalities.




**(a)** Without the estimation of missing data.




**(b)** With the estimation of missing data.

**Fig. 9.** Indicator of success with and without the estimation of missing data [24].

We can see in Fig. 9(a) a lot of equipossible results (with all possibilities equal to 1) due to missing data. To reduce the equipossible variables, we have performed an imputation of missing data using an iterative PCA algorithm [2]. We present the results in Fig. 9(b). Another advantage of our approach is the use of uncertain gates in order to avoid eliciting all parameters of the CPTs. We have compared the result of the indicator of success with and without uncertain gates by compiling the possibilistic networks. The results are the following (Fig. 10):

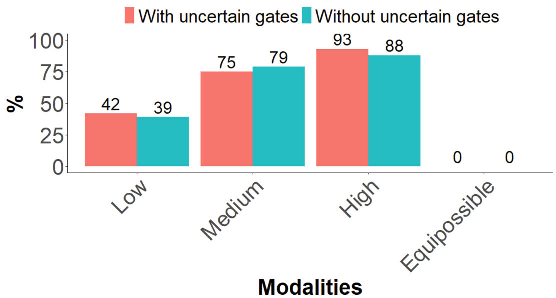



**Fig. 10.** Comparison of the indicator of success with and without uncertain gates [24].

The results are very close but uncertain gates require fewer parameters than the CPTs elicited by a human expert (Fig. 11).

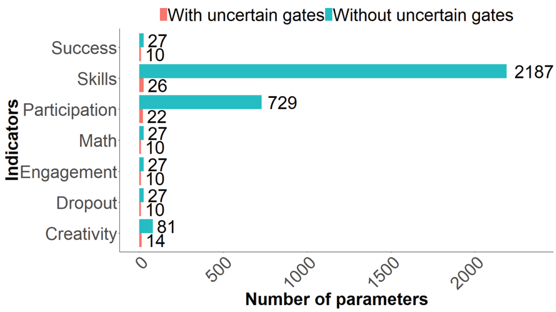



**Fig. 11.** Comparison of the results with and without uncertain gates by using the compiling of the possibilistic network [24].

The above figure shows that the number of parameters is highly decreased by using uncertain gates for all indicators. We have also compared the running time of the computation of the indicators by using the compiling of the possibilistic networks and the message passing algorithm. The results are the following (Fig. 12):

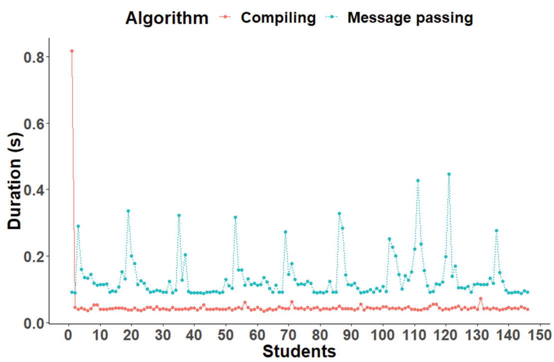



**Fig. 12.** Comparison of computation time for all students.

We can see in the above figure that the computation time for the indicators of the first student is higher because of the circuit generation. The variation of the computation time between the other students are due to the operating system. Indeed, we used a reinitialization module that allows us to reuse the circuit instead of rebuilding every circuit for each student. The calculation is then faster for the other students because we reuse the first circuit. We have computed the average of the computation time for both methods and we obtain (Fig. 13):

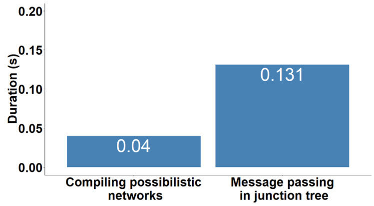



**Fig. 13.** The average computation time [24].

We can see that the computation time is improved by compiling the junction tree of the possibilistic network. The compiling approach is faster than the message passing algorithm. We have presented the results of the indicators in an Educational Decision Support System (EDSS). The architecture of the system is summarized in the following Fig. 14:

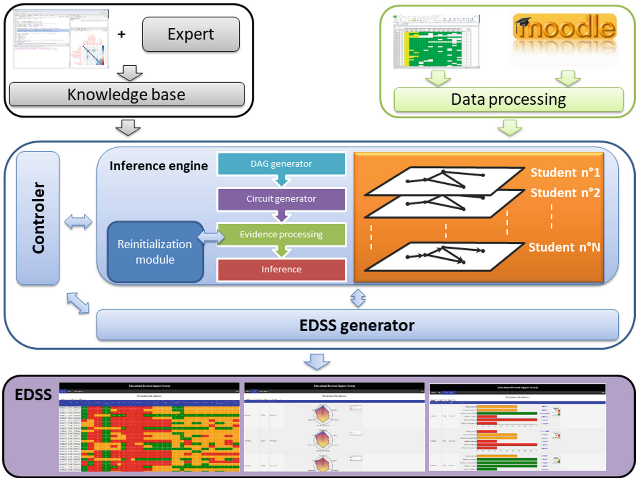
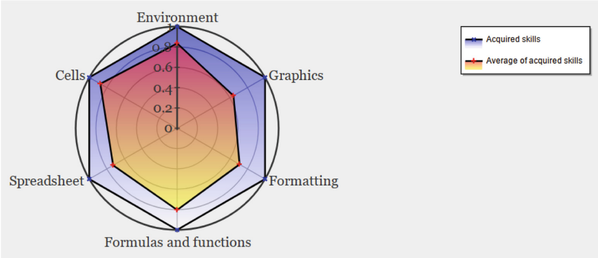
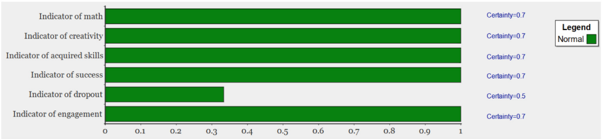



**Fig. 14.** The EDSS architecture.

This visualization of the indicator is easy to interpret. Indeed, the possibilistic results are transformed to present only the modality of the indicators with the highest necessity. We used a radar graph for the indicator of skills and a horizontal bar graph for the other indicators. We also used a color code to indicate the students with difficulties. The indicators allow us to detect the students at risk. Nevertheless, we must be careful with the interpretation of the results because the indicators are shortcuts of reasoning. Further investigations must be performed by the teachers to confirm the results before taking a decision. We present here two examples of results, the first one concerns successful students:



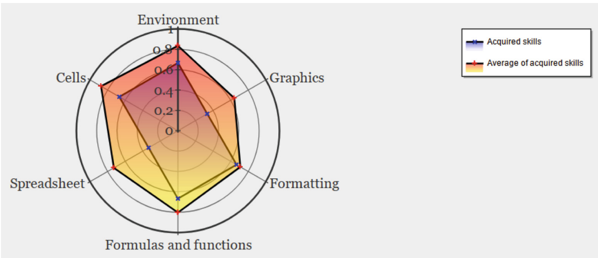

**(a)** Skills



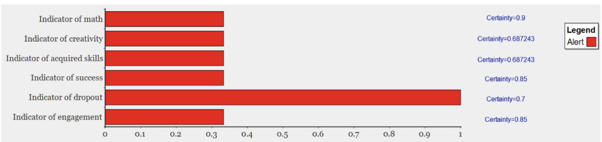

**(b)** Indicators

**Fig. 15.** Student with good results.

In Fig. 15(a) we can see that the curve in blue, representing the score for all skills, has its full value. We have also presented the average skill level in red. In Fig. 15 (b) we can see that all indicators are green and we can see the certainty of the indicators at the right of the graph. Certainty is the necessity measure. The following figures represent a student's disengagement:



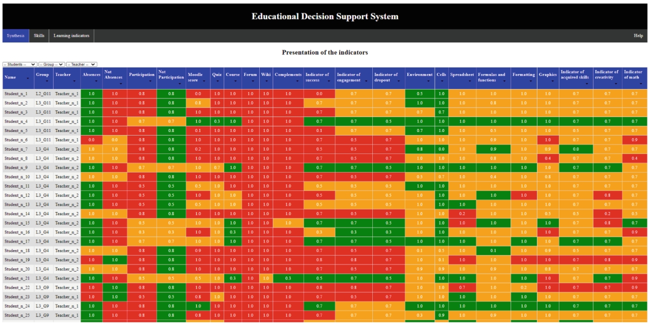

**(a)** Skills




**(b)** Indicators

**Fig. 16.** Dropout student.

In Fig. 16(b) we can see all indicators in red and a certainty of 0.85 for the indicator of success. The indicator of dropout is also at its highest with a certainty of 0.7. All of these indicators show that the student will probably fail at the examination. The EDSS is generated automatically at the end of the calculation as a PHP web site with three tabs. The EDSS is presented below (Fig. 17):




**(a)** First tab.


**(b)** Second tab.


**(c)** Third tab.

**Fig. 17.** The tabs of the EDSS.

The first tab is the synthesis of all information with its certainty and it allows us to sort data for all columns. The second tab gathers all skill information in a radar graph. We can compare the skills of a student to the average skills of the year group, the class or the teacher's groups. We can also visualize the certainty for all skills. The last tab concerns all course indicators and uses a color code to highlight the students with difficulties.

## 6 Conclusion

We proposed to compute learning course indicators for a course of Spreadsheet based on the teachers' knowledge. The indicators were presented in a decision making system for the teachers. To do this we used possibility theory to manage the uncertainties and imprecisions of the teachers' knowledge. As the latter can be represented by a DAG, we used possibilistic networks to compute the indicators by using a message passing algorithm. Then, we performed several improvements, the first of which was using uncertain gates to compute automatically the CPTs. Indeed, uncertain gates allow us to reduce the number of parameters to elicit. Then we proposed a new approach of exact inference based on the compilation of the junction tree. The first step was to generate the MIN-MAX circuit, then we applied an upward pass followed by a downward pass. We compared the performance of both algorithms and highlighted the performance of the compiling approach. The computation time is improved compared to the message passing algorithm.

In future, we would like to perform further experimentations in order to better evaluate the performance of our algorithm proposed for the compilation of the junction tree of a possibilistic network. We would like to compare our approach with approaches based on propositional logic and more especially the deterministic decomposable negation normal form (d-DNNF). We would like to conduct other experiments concerning the computation of learning indicators. Another perspective can be to build a recommendation system based on these indicators and to use a chatbot as a virtual companion to provide advice to students.

## References

1. Ahmad, K., et al.: Artificial intelligence in education: a panoramic review (06 2020)
2. Audigier, V., Husson, F., Josse, J.: A principal components method to impute missing values for mixed data. Adv. Data Anal. Classif. **10**(1), 5–26 (2015)
3. Baker, R.S.J.D., Yacef, K.: The state of educational data mining in 2009: a review and future visions. J. Educ. Data Min. **1**(1), 3–17 (2009)
4. Benferhat, S., Dubois, D., Garcia, L., Prade, H.: Possibilistic logic bases and possibilistic graphs. In: the Conference on Uncertainty in Artificial Intelligence, pp. 57–64 (1999)
5. Borgelt, C., Gebhardt, J., Kruse, R.: Possibilistic graphical models. Comput. Intell. Data Min. **26**, 51–68 (2000)
6. Bousbia, N., Labat, J.M., Balla, A., Rebaï, I.: Analyzing learning styles using behavioral indicators in web based learning environments. In: EDM 2010 International Conference on Educational Data Mining, pp. 279–280 (2010)

7. Cheema, J.R.: A review of missing data handling methods in education research. Rev. Educ. Res. **84**, 487–508 (2014)
8. Darwiche, A.: A differential approach to inference in Bayesian networks. J. ACM **50**(3), 280–305 (2003)
9. Dray, S., Josse, J.: Principal component analysis with missing values: a comparative survey of methods. Plant Ecol. **216**(5), 657–667 (2014). https://doi.org/10.1007/s11258-014-0406-z
10. Dubois, D., Fusco, G., Prade, H., Tettamanzi, A.: Uncertain logical gates in possibilistic networks. an application to human geography. In: Beierle, C., Dekhtyar, A. (eds.) SUM 2015. LNCS (LNAI), vol. 9310, pp. 249–263. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23540-0_17
11. Dubois, D., Prade, H.: Possibility Theory: An Approach to Computerized Processing of Uncertainty. Plenum Press, New York (1988)
12. Hisdal, E.: Conditional possibilities independence and noninteraction. Fuzzy Sets Syst. **1**(4), 283–297 (1978)
13. Huebner, R.A.: A survey of educational data mining research. Res. High. Educ. J. **19**, 13 (2013)
14. Jensen, F.V., Lauritzen, S., Olesen, K.: Bayesian updating in causal probabilistic networks by local computation. Comput. Stat. Q. **4**, 269–282 (1990)
15. Kjaerulff, U.: Reduction of computational complexity in Bayesian networks through removal of week dependences. In: Proceeding of the 10th Conference on Uncertainty in Artificial Intelligence, pp. 374–382. Morgan Kaufmann (1994)
16. Kruskal, J.B.: On the shortest spanning subtree of a graph and the travelling salesman problem. In: Proceedings of the American Mathematical Society, pp. 48–50 (1956)
17. Lauritzen, S., Spiegelhalter, D.: Local computation with probabilities on graphical structures and their application to expert systems. J. R. Stat. Soc. **50**(2), 157–224 (1988)
18. Neapolitan, R.E.: Probabilistic Reasoning in Expert Systems: Theory and Algorithms. Wiley, Hoboken (1990)
19. Park, J.D., Darwiche, A.: A differential semantics for jointree algorithms. In: Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS 2002, 9–14 December 2002, Vancouver, British Columbia, Canada], pp. 785–784 (2002)
20. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers Inc., San Francisco (1988)
21. Pearl, J.: A constraint - propagation approach to probabilistic reasoning. In: Kanal, L.N., Lemmer, J.F. (eds.) Uncertainty in Artificial Intelligence. Machine Intelligence and Pattern Recognition, vol. 4, pp. 357–369. North-Holland (1986)
22. Pearl, J.: Fusion, propagation, and structuring in belief networks. Artif. Intell. **29**(3), 241–288 (1986)
23. Petiot, G.: Merging information using uncertain gates: an application to educational indicators. In: Medina, J., et al. (eds.) Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Foundations - 17th International Conference, IPMU (2018) Cádiz, Spain, June 11-15, 2018, Proceedings, Part I. Communications in Computer and Information Science, vol. 853, pp. 183–194. Springer (2018)
24. Petiot, G.: Compiling possibilistic networks to compute learning indicators. In: Proceedings of the 13th International Conference on Agents and Artificial Intelligence-Volume 2: ICAART, pp. 169–176. INSTICC, SciTePress (2021)
25. Raouia, A., Amor, N.B., Benferhat, S., Rolf, H.: Compiling possibilistic networks: alternative approaches to possibilistic inference. In: Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence. pp. 40–47. UAI'10, AUAI Press, Arlington, Virginia, United States (2010)

26. Shabani, Z., Eshaghian, M.: Decision support system using for learning management systems personalization. Am. J. Syst. Softw. **2**(5), 131–138 (2014)
27. Shachter, R.: Intelligent probabilistic inference. In: Kanal, L.N., Lemmer, J.F. (eds.) Uncertainty in Artificial Intelligence, pp. 371–382. North-Holland (1986)
28. Shachter, R.: Evidence absorption and propagation through evidence reversals. Mach. Intell. Pattern Recogn. **10** (2013). https://doi.org/10.1016/B978-0-444-88738-2.50021-X
29. Shenoy, P.P., Shafer, G.: Axioms for probability and belief-function propagation. In: Shachter, R.D., Levitt, T.S., Kanal, L.N., Lemmer, J.F. (eds.) Uncertainty in Artificial Intelligence. Machine Intelligence and Pattern Recognition, vol. 9, pp. 169–198. North-Holland (1990). https://doi.org/10.1007/978-3-540-44792-4_20
30. Zadeh, L.A.: Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets Syst. **1**, 3–28 (1978)
31. Zawacki-Richter, O., Marín, V.I., Bond, M., Gouverneur, F.: Systematic review of research on artificial intelligence applications in higher education - where are the educators? Int. J. Educ. Technol. High. Educ. **16**, 1–27 (2019)
32. Zhang, N.L., Poole, D.: A simple approach to Bayesian network computations. In: Proceedings of the Tenth Canadian Conference on Artificial Intelligence, pp. 171–178 (1994)