Hybrid approach of the fuzzy C-means and the K-nearest neighbors methods during the retrieve phase of dynamic case based reasoning for personalized follow-up of learners in real time

El Ghouch Nihad[1], En-Naimi El Mokhtar[2], Zouhair Abdelhamid[3], Al Achhab Mohammed[4]
[1,2]LIST Laboratory, the Faculty of Sciences and Technologies, Abdelmalek Essaadi University, Tangier, Morocco
[3]The National School of Applied Sciences, Abdelmalek Essaadi University, Al-Hoceima, Morocco
[4]The National School of Applied Sciences, Abdelmalek Essaadi University, Tetouan, Morocco

Article Info

Article history:

Received Jan 21, 2019
Revised Mar 31, 2019
Accepted Jun 25, 2019

Keywords:

Adaptive learning systems
Cluster
Felder-silverman learning style model (FSLSM)
Fuzzy C-means method (FCM)
Incremental dynamic case based reasoning (IDCBR)
K-Nearest neighbors method (KNN)
Learning path
Learning style

ABSTRACT

The goal of adaptive learning systems is to help the learner achieve their goals and guide their learning. These systems make it possible to adapt the presentation of learning resources according to learners' needs, characteristics and learning styles, by offering them personalized courses. We propose an approach to an adaptive learning system that takes into account the initial learning profile based on Felder Silverman's learning style model in order to propose an initial learning path and the dynamic change of his behavior during the learning process using the Incremental Dynamic Case Based Reasoning approach to monitor and control its behavior in real time, based on the successful experiences of other learners, to personalize the learning. These learner experiences are grouped into homogeneous classes at the behavioral level, using the Fuzzy C-Means unsupervised machine learning method to facilitate the search for learners with similar behaviors using the supervised machine learning method K- Nearest Neighbors.

Corresponding Author:

El Ghouch Nihad,
LIST Laboratory,
Department of Computer Science,
Faculty of Sciences and Technologies,
Abdelmalek Essaadi University,
Street of charf B. P 410, Tangier, Morocco.
Email: nihad_elghouch@hotmail.fr

1. INTRODUCTION

Adaptive learning systems are intended to provide the necessary pedagogical tools and to create the appropriate atmosphere to promote learning among learners. These systems need to tailor learning according to the preferences, needs and learning styles of each learner to provide a personalized follow-up in real time. Individualized learner follow-up plays an important role in adaptive learning systems, since without the information on the progression and the results obtained by the learners, it will be very difficult to accompany and control their behaviors. To achieve real-time monitoring of learners, we will need "smart" components to define in real time the learner's profile and preferences for adapting learning to each moment of the learning process to avoid problems of failure and demotivation. This adaptation should be based on the experiences of other learners who will be grouped into classes using data mining techniques [1], which can extract and discover important data from a collection of large amounts of data to facilitate the research and classification of learners.

In this paper, we propose an approach to an adaptive learning system based on the Felder-Silverman learning style model (FSLSM) and Incremantal Dynamic Case Base Reasoning (IDCBR) to follow the learner in difficulty in real time by offering him a path adapted to his profile and the dynamic change of his behavior during the learning process. To perform this dynamic tracking, we used a cascade of the Fuzzy C-Means method (FCM) and the K-Nearest Neighbors method (KNN) in the second step of the Dynamic Case Base Reasoning to propose a path adapted to each moment of learning based on the behavior of other learners who are classified as homogeneous groups. In the second section, we present the techniques used: the incremental Dynamic Case BaseReasoning Approach, the K- Nearest Neighbors method (KNN) and the Fuzzy C-Means method (FCM). In the third section, we describe our approach by detailing the operation and the different steps of the proposed approach. The fourth section will present the results of the experiment of the combination of the FCM method and the KNN method on a sample of 22 learners using the "MERISE" method as a learning material, the results are presented by the Matlab software and we finish with a conclusion.

2. METHODOLOGIES

2.1. Case based reasoning

Case Based Reasoning (CBR) is a problem-solving approach that uses past experiences to solve new problems [2]. A very important feature of the CBR is its relationship with learning, it allows updating cases and learning new cases. Solving a problem using the Case Based Reasoning approach can be done through a typical cycle with a set of steps. These steps are more detailed in [3-5]. The traditional cycle of the CBR includes the following steps: Elaboration, Retrieve, Reuse, Revise and Retain. Whereas, in the Dynamic Case Based Reasoning (DCBR) cycle, the target case (the problem to solve) is presented by dynamic descriptors that change over time, a new star CBR cycle is proposed by [4, 5] in Figure 1, this is the Incremental Dynamic Case Based Reasoning (IDCBR). This new cycle has resulted in changes in the order and content of the traditional CBR cycle (some steps can be repeated several times).

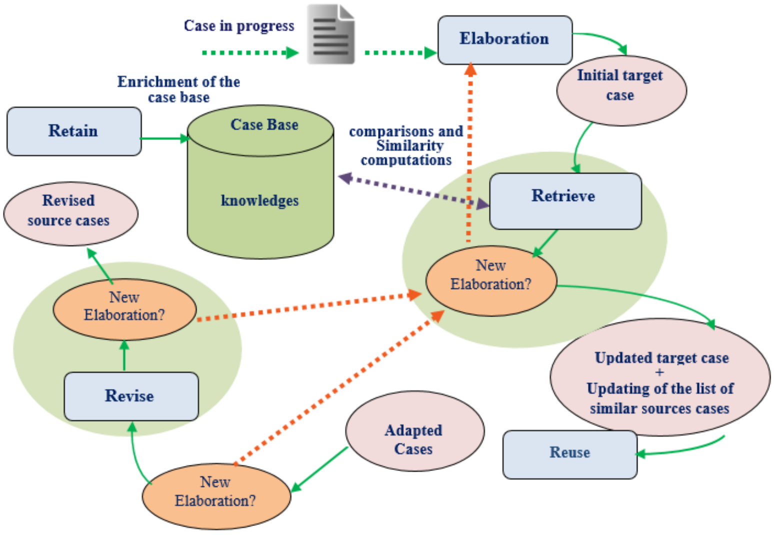


Figure 1. IDCBR cycle [4, 5]

2.2. Classification with k-nearest neighbors

The K-Nearest Neighbors (KNN) algorithm [6] is non-parametric and supervised method of classification [7] and introduced in [8], It can be used both for classification and prediction. It is based on a simple and intuitive principle of grouping data according to their neighborhood. The purpose of this algorithm is to classify new data based on attributes and data samples. Each data is assigned to the class most represented among its k nearest neighbors [9].

The K-Nearest Neighbors algorithm is essentially based on two main principles:

a. A number of nearest neighbors (K) to use [10];
b. A metric to measure nearest neighbors [11]. This measurement is necessary to determine the distances, it influences on the result of the classification and the quality of the predictions.

The K-Nearest Neighbors algorithm:

a. Let k be the number of nearest neighbors and D the set of training data (y_j).
b. For each new observation xi, calculate d(x_i, y_j) using a measure of distance for each data y_j of D.
c. Select the k nearest data (y_j) of the new observation xi.
d. Classify the new observation xi according to the majority class among its neighbors.

Although the KNN method is easy to implement, it has some limitations such as: the high computational complexity to find the k nearest neighbor samples, the computation time to compute the similarities [12]. But there are heuristics to reduce the set of data to classify, to increase the speed of classification [13, 14] and to accelerate the K-NN algorithm based on clustering and attribute filtering [15]. There is some work integrating the KNN method in the CBR cycle, we quote the work of [16], which showed that the KNN algorithm is suitable for use in the CBR approach.

2.3. Fuzzy c-means

The Fuzzy C-Means (CMF) is a fuzzy clustering technique [17] that generalizes the C-Means technique, derived from the K-Means algorithm. The Fuzzy C-Means algorithm allows elements to belong to several clusters simultaneously [18], it is based on the optimization of a quadratic classification criterion where each class is represented by its center of gravity [19, 20].

This iterative algorithm assigns a membership of an object to a cluster, based on the similarity of an object with a particular cluster to all other clusters. FCM minimizes the following objective function [21]:

$$\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{m}\left|x_i - c_j\right|^2$$

with

a. m> 1 is a parameter controlling the degree of fuzziness (usually m = 2);
b. c_j is the center of a cluster;
c. x_i, denotes the ith element of the measured data
d. u_{ij} represents the degree of membership of an element xi in the jth cluster, the larger uij is the stronger the cluster membership ($u_{ij}\epsilon[0,1]$ and $\sum_{j=1}^{n} u_{ij}$=1).

The main steps of the Fuzzy C-Means algorithm are:

a. Choose the number of clusters [22, 23];
b. Initialize the membership matrix u_{ij};
c. Calculate the centroids c_j: $c_j = \left(\sum_i (u_{ij})^m . x_i\right) / \left(\sum_i (u_{ij})^m\right)$
d. Re-adjust the membership matrix according to the position of the centroids: $u_{ij} = 1/\left(\sum_{k=1,c}(d_{ij}/d_{ik})^{(2/(m-1))}\right)$ with : $d_{ij} = \left|x_i - c_j\right|$
e. If $\left\|u_{(j+1)} - u_{(j)}\right\| < \varepsilon$ (threshold representing the convergence error), then the algorithm stops, otherwise the return to step b.

The Fuzzy C- Means method has a hybrid character (the concept of center of gravity and the Fuzzy concept), makes it simple and fast. The FCM requires input parameters, and that he partition matrix is fuzzy, which needs to be initialized in an appropriate manner.

3. DESCRIPTION OF OUR APPROACH

Our approach is to provide a personalized and individualized follow-up of the learner in real time, according to his FSLSM learning style, his observed learning traces and the past successful experiences of other learners. Since each learner's learning process evolves over time, it requires the use of intelligent techniques to automatically adapt to dynamic changes in the learner's behavior in real time during the learning process.

Our approach allows:

a. Detecting the learner profile using the Felder-Silverman learning style model, creating profile classes and providing an initial learning path for each learner to start learning.
b. Following the learner who has encountered learning problems in real time by using the IDCBR to offer a personalized path according to the dynamic change of his behavior. This approach is triggered with each change in the learning process by observing and analyzing the learning traces left in the system. The IDCBR integrates in the second step a combination of two methods of machine learning, the FCM

method to group learners into homogeneous and similar classes in order to facilitate the search of the closest learners through the KNN method.

Schematically, we can present our approach as follows refer Figure 2.

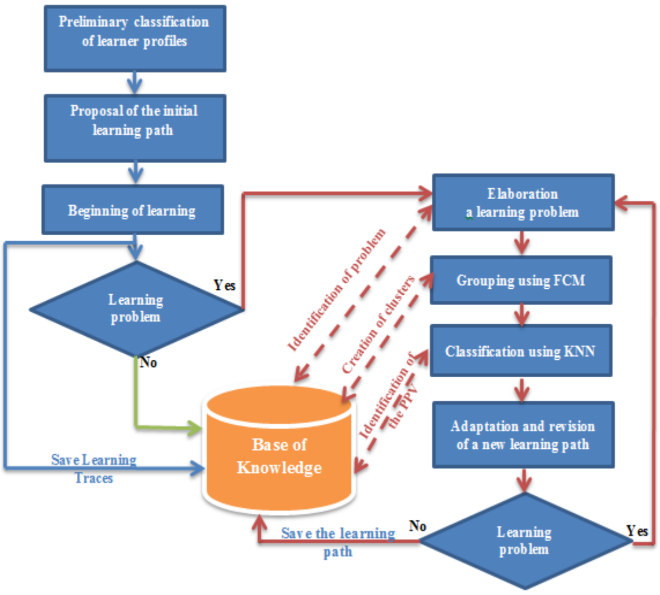

Preliminary classification of learner profiles
Proposal of the initial learning path
Beginning of learning
Learning problem
Yes
No
Save Learning Traces
Base of Knowledge
Elaboration a learning problem
Grouping using FCM
Classification using KNN
Adaptation and revision of a new learning path
Identification of problem
Creation of clusters
Identification of the PPV
Learning problem
No
Yes
Save the learning path

Figure 2. Diagram of our approach

3.1. Preliminary classification of learner profiles

In our learning system, we used the Felder-Silverman Learning Style Model (FSLSM) [24] to detect the learning style of each learner. This model classifies the learner into four dimensions: "Sensing / Intuitive" (perception), "Visual / Verbal" (input), "Active / reflective" (process) and "Sequential / Global" (comprehension), by providing a test questionnaire to him.

We are interested in two dimensions of the FSLSM Model: "Sensing / Intuitive" and "Visual / Verbal". The choice of these two dimensions is due to the fact that our system does not allow interactions between learners (Active / reflective) and considers that freedom of navigation is implicitly included in our system (Sequential / Global). Therefore, the learning style of each learner is presented by the "Sensing / Intuitive" dimension which indicates the preferred learning resources for learners and the "Visual / Verbal" dimension which indicates the formats of the preferred pedagogical objects among learners. According to the results obtained from the FSLSM test, the learner is classified according to 4 groups of profiles (Sensing / Visual, Sensing / Verbal, Intuitive / Visual and Intuitive / Verbal).

3.2. Proposal of an initial learning path

Our system proposes initial learning paths which are constituted by a series of learning objects. These paths are constructed through the correspondence between the FSLSM learning style of each learner and the metadata describing the learning objects [25]. The Table 1 shows the correspondence between the FSLSM learning style and the learning objects.

Table 1. Correspondence between FSLSM learning style and learning object metadata

Profiles	Initial leaning path
Sensing / Visual	Example, exercise and quiz in the form of a picture, a video
Sensing / Verbal	Example, exercise and quiz in the form of a text, an audio
Intuitive / Visual	Notion, definition, algorithm of a picture, a video
Intuitive / Verbal	Notion, definition, algorithm in the form of a text, an audio

3.3. Beginning of learning

After detecting the learning style of the learner that offers him a learning path according to his style. The learning process of each learner takes two scenarios:

a. Normal: the system goes directly to the last step of Dynamic Case Based Reasoning, which records learning style, learning path, and interaction traces of each learner as a new case (a new successful experience) in the base of cases (Figure 2 green line);
b. Abnormal: When the system detects an anomaly or problem during the learning process, Dynamic Case Based Reasoning is triggered to provide a real-time coaching (Figure 2 red line).

In the second situation, the DCBR adapts the learning process according to the profile of each learner through:

a. The observation of the learning process of the learner (his learning path);
b. The comparison of the behavior of the learner in difficulty at each moment ti with the behaviors of the other learners realized during the previous instants (from t_0 to t_{i-1}).

3.4. Elaboration of the learning problem

3.4.1. Generation of the learning problem

Data collected from the learning process is processed to extract relevant information that will be modeled as cases. These cases characterize learner’s behaviors (learning paths) through the traces saved in the base of knowledge (base of cases).

The observed learning path of each learner is composed of a series of learning objects numbered according to the index i (with i = {0 ... Nbr_Obj}). Each learning object consulted at each instant ti represents a trace at time ti which contains indicators that we considered relevant:

a. t_i: the episode ti corresponds to a consultation of a learning object numbered i;
b. Crs: chosen course;
c. VSCrs: learning object;
d. DtVSCrs: date of the visit of the learning object;
e. NbrVstVSCr: number of visits to the learning object;
f. DvVSCr: duration of the visit of the learning object**;**
g. RVSCrs: resource of the learning object;
h. FVSCrs: nature of the learning object.

The learning traces are presented by a vector ***Episode****(ti, j)* which describes a trace of the learner j at time t_i.

$$Episode_{(ti,j)} = \begin{pmatrix} VSCrs \\ DtVSCrs \\ NbrVstVSCr \\ DvVSCr \\ FVSCr \\ RVSCr \end{pmatrix} \tag{1}$$

In our approach, the learning path (target case / source case) is presented in an object representation format (Figure 3) that adapts to the dynamic side of adaptive learning systems. It is modeled by an object that is characterized by a "Traces" object vector and a "Validation" attributes indicating the validation of this learning path.

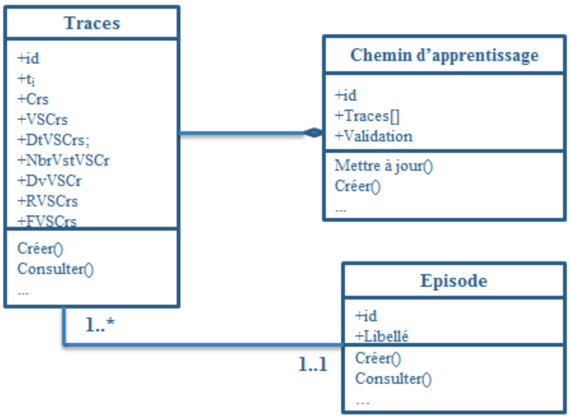


Figure 3. Object representation of a learning path

3.4.2. Description of the learning problem

The purpose of this step is the recognition of a new learning path, which is why our system must

a. Detect the instant t_p, including:
   1. The learner does not follow the proposed path;
   2. The learner is blocked at the level of a learning object;
   3. The learner scored negatively during the assessment.

b. Describe the **APC** learner (target case) by collecting the traces that are represented as a new, invalidated learning path (a new case that needs to be solved), through the creation of the "**Episode**$_{\mathbf{(ti,\ j)}}$" vectors at each instant t_i (i from 0 to p-1).

3.5. Grouping using FCM

To solve the KNN algorithm problems (in terms of classification time and sample size) and to facilitate the search for similar candidate learners, we used the FCM method to group learners by similar degree. We use this method to create clusters whose learners have similar behaviors based on the learning traces presented as **Episode**$_{\mathbf{(ti,j)}}$ at each moment t_i of the learning process.

The purpose of this step is to identify the **CLAPS** cluster closest to the learner in difficulty APC. This cluster contains all learners with similar behaviors to the learner APC (even if they have different profiles). In this step, we retrieve the IDs of the learners belonging to the **CLAPS** cluster as a **V_CLAPS** vector.

3.6. Classification using KNN

The previous step allows limiting the number of learners to make the search of the nearest learners easy and fast. In order to carry out the classification using the KNN method, we will execute the following steps:

a. Create the **Consultation_Episode**$_{\mathbf{(ti)}}$ matrix for each moment t_i (i from 0 to p-1) from the M_CLAPS matrix created in the previous step, based on equation (1):

$$Consultation\_Episode_{(ti)} = \begin{cases} \sum_{j=1}^{N\_Aps} Episode_{(ti,j)}\ , & i = 0 \\ \sum_{i=0}^{t_{p-1}} \sum_{j=1}^{N\_Aps} Episode_{(ti,j)}\ , & i \geq 1 \end{cases} \quad (2)$$

b. Calculate the similarity "d" between $\mathbf{Episode}_{(\mathbf{ti},\mathbf{Aps})}$ and each line of the **Consultation_Episode**$_{\mathbf{(ti)}}$ matrix for each moment t_i (i from 0 to p-1).

c. Save the distances d_0 ... d_{p-1} in a table D_{ti} and sort this table in increasing order for each moment t_i (i from 0 to p-1).

d. Retrieve the learners according to the distance table sorted for each moment t_i (i from 0 to pb-1).

e. Get the first k learners for each moment t_i (i from 0 to p-1).

f. Calculate $\sum_{i=0}^{p-1} d_i$

g. Find the **APS** learner who has a minimum distance min $\sum_{i=0}^{p-1} d_i$

h. Assign **APS** behavior to **APC** and update the **APC** profile.

3.7. Adaptation and revision of a new learning path

The learner in difficulty will inherit the same behavior and learning path of the nearest neighbor learner who has a minimal distance. If the system detects another learning problem in the learning process, it returns to the elaboration of learning problem step, otherwise a control of the adapted learning path will be executed through the actual observation of the learning process or the results obtained during the evaluation. If the system detects another learning problem in the learning process, it returns to the elaboration of the learning problem.

3.8. Saving a learning path

If the system detects no change in the revised learning path. The new learning path of the learner in difficulty APC and their learning traces becomes a new learning experience that is solved and maintained in the learning base to solve future learning problems.

4. RESULTS AND DISCUSSIONS

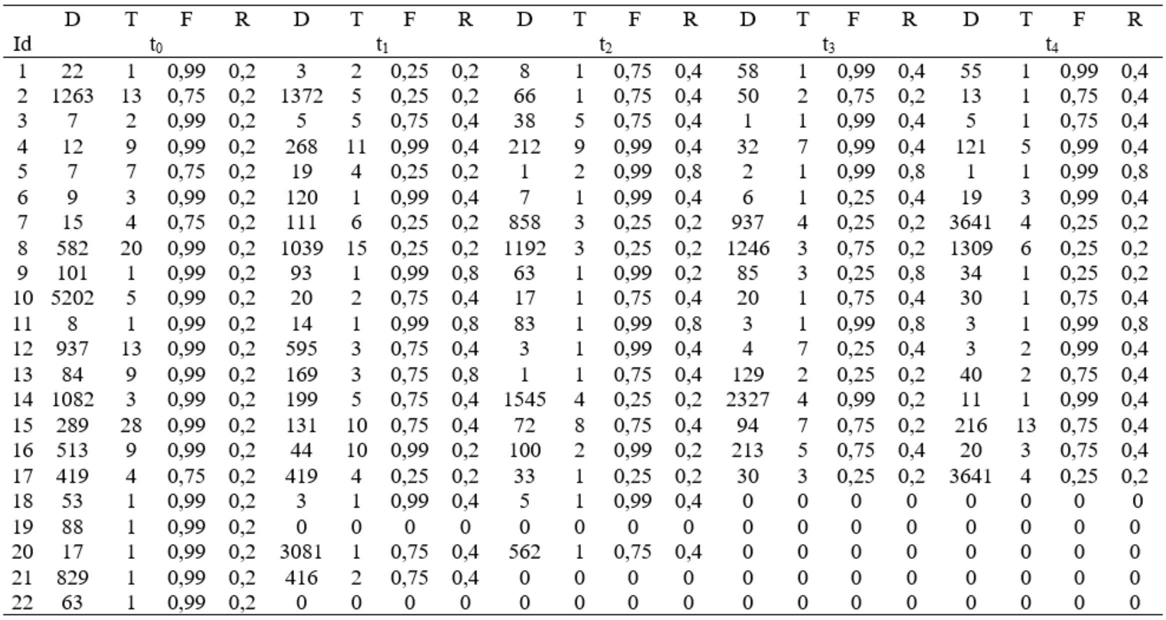
4.1. Presentation of the data

In this section, we test the effectiveness of the combination of the FCM method and the KNN method in the Retrieve step of IDCBR recall step. We first apply the KNN method and then we combine the FCM and KNN to find the nearest learners, using the Matlab software. We define the following elements:

a. The number of learners: 22 learners.
b. The Merise course contains 5 learning units, each unit of the course is presented by different versions.
c. The initials profiles of the learners: 4 profiles (Sensing / Visual, Sensing / Verbal, Intuitive / Verbal and Intuitive / Visual)
d. The number of K neighbors: in general, the good value of K is $\sqrt{n}$ where n is the population [26]. In our case, n is the number of learners who validated the course.
e. The measure of **distance**: the Euclidean distance (d);

We have grouped the learners' traces in the different instants t_i (i from 0 to 4) in the Table 2. Each row of the table presents the matrix $\mathbf{Consultation\_Episode}_{(\mathbf{ti})}$ (with i from 0 to 4 and j is the Id of the learner) and each row of this matrix contains the vector $Episode_{(ti,j)}$ which presents the trace of the learner j in moment t_i. The trace contains a set of information on the activity carried out by the learner at each moment t_i such as the duration of consultation of the learning object (D), the number of times of consultation (T), the format (F) and the resource (R) of this learning object.

Table 2. Traces of learners at each episode t_i



	D	T	F	R	D	T	F	R	D	T	F	R	D	T	F	R	D	T	F	R
Id	t_0				t_1				t_2				t_3				t_4			
1	22	1	0,99	0,2	3	2	0,25	0,2	8	1	0,75	0,4	58	1	0,99	0,4	55	1	0,99	0,4
2	1263	13	0,75	0,2	1372	5	0,25	0,2	66	1	0,75	0,4	50	2	0,75	0,2	13	1	0,75	0,4
3	7	2	0,99	0,2	5	5	0,75	0,4	38	5	0,75	0,4	1	1	0,99	0,4	5	1	0,75	0,4
4	12	9	0,99	0,2	268	11	0,99	0,4	212	9	0,99	0,4	32	7	0,99	0,4	121	5	0,99	0,4
5	7	7	0,75	0,2	19	4	0,25	0,2	1	2	0,99	0,8	2	1	0,99	0,8	1	1	0,99	0,8
6	9	3	0,99	0,2	120	1	0,99	0,4	7	1	0,99	0,4	6	1	0,25	0,4	19	3	0,99	0,4
7	15	4	0,75	0,2	111	6	0,25	0,2	858	3	0,25	0,2	937	4	0,25	0,2	3641	4	0,25	0,2
8	582	20	0,99	0,2	1039	15	0,25	0,2	1192	3	0,25	0,2	1246	3	0,75	0,2	1309	6	0,25	0,2
9	101	1	0,99	0,2	93	1	0,99	0,8	63	1	0,99	0,2	85	3	0,25	0,8	34	1	0,25	0,2
10	5202	5	0,99	0,2	20	2	0,75	0,4	17	1	0,75	0,4	20	1	0,75	0,4	30	1	0,75	0,4
11	8	1	0,99	0,2	14	1	0,99	0,8	83	1	0,99	0,8	3	1	0,99	0,8	3	1	0,99	0,8
12	937	13	0,99	0,2	595	3	0,75	0,4	3	1	0,99	0,4	4	7	0,25	0,4	3	2	0,99	0,4
13	84	9	0,99	0,2	169	3	0,75	0,8	1	1	0,75	0,4	129	2	0,25	0,2	40	2	0,75	0,4
14	1082	3	0,99	0,2	199	5	0,75	0,4	1545	4	0,25	0,2	2327	4	0,99	0,2	11	1	0,99	0,4
15	289	28	0,99	0,2	131	10	0,75	0,4	72	8	0,75	0,4	94	7	0,75	0,2	216	13	0,75	0,4
16	513	9	0,99	0,2	44	10	0,99	0,2	100	2	0,99	0,2	213	5	0,75	0,4	20	3	0,75	0,4
17	419	4	0,75	0,2	419	4	0,25	0,2	33	1	0,25	0,2	30	3	0,25	0,2	3641	4	0,25	0,2
18	53	1	0,99	0,2	3	1	0,99	0,4	5	1	0,99	0,4	0	0	0	0	0	0	0	0
19	88	1	0,99	0,2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	17	1	0,99	0,2	3081	1	0,75	0,4	562	1	0,75	0,4	0	0	0	0	0	0	0	0
21	829	1	0,99	0,2	416	2	0,75	0,4	0	0	0	0	0	0	0	0	0	0	0	0
22	63	1	0,99	0,2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The Table 3 presents the number of learners by profile. Of the 22 students, 17 validated the course (Id 1-17) and the other students have learning problems (Id 18-22). Table 4 show learners that have a learning problem. We take by test, the learner in difficulty whose ID is 18 who has a learning problem in learning unit 3 (moment t_2). The Figure 4 illustrated by the Matlab software shows the distribution of learners according to the first 3 attributes at moment t_0, before classification in clusters (the duration of consultation, the number of visit, the resource and the format of the object learning). The learner ID 18 is presented by the sign +.

Table 3. Number of learners by profile

Profiles	Number of learners
Sensing / Visual	15
Sensing / Verbal	3
Intuitive / Verbal	3
Intuitive / Visual	1

Table 4. Learners have a learning problem

Learner	Initial profile
18	Sensing / Visual
19	Sensing / Visual
20	Sensing / Visual
21	Intuitive / Visual
22	Intuitive / Verbal

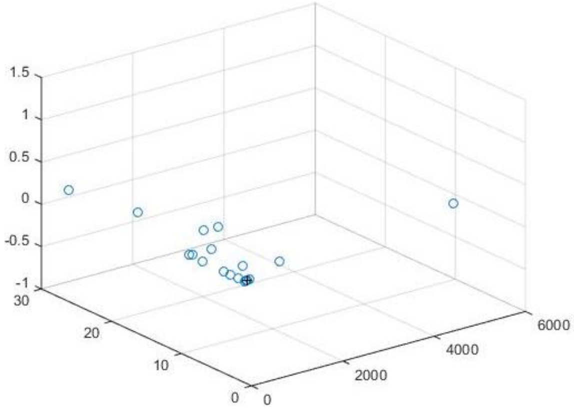


Figure 4. Distribution of learners before classification

4.2. Classification using KNN

We apply the K Nearest Neighbors algorithm by comparing the traces of the learner Id 18 with the traces of 17 learners who validated the course, we obtain the Ids of the K nearest with the Matlab software as shown in Table 5. We apply the K Nearest Neighbors algorithm at each instant t_i, we obtain the following results with the Matlab software as shown in Table 6.

Table 5. Id of learners according to KNN

Number of the K nearest learners					
2		3		4	
Id	d	Id	d	Id	d
1	31,02	1	31,02	1	31,02
3	46,22	3	46,22	3	46,22
		11	46,32	11	46, 32
				5	49,16

Table 6. ID of learners according to KNN for each moment t_i

Number of the K nearest learners K=2				Number of the K nearest learners K=3				Number of the K nearest learners K=4			
t_0		t_1		t_0		t_1		t_0		t_1	
Id	d	Id	d	Id	D	Id	d	Id	d	Id	d
1	31	1	1,26	1	31	1	1,26	1	31	1	1,26
13	32,01	3	4,47	13	32,01	3	4,47	13	32,01	3	4,47
				7	38,11	11	11	7	38,11	11	11
								4	41,77	5	16,29

4.3. Combinaison of KNN and FCM

We apply the Fuzzy C-Means algorithm; we obtain the results presented in the Table 7 with the Matlab software, which indicate the number of learners existing in the cluster of which the learner Id18 belongs, according to the number of clusters. We are running the K Nearest Neighbors algorithm in the cluster of which the Learner Id 18 belongs, we get the following results as shown in Table 8. The Figure 5 shows the grouping of learners in the 4 clusters (red, green, yellow, blue), according to the first three attributes at time t_0. We run the K-Plus Nearest Neighbors algorithm in the cluster of which the learner Id 18 belongs to each t_i as shwon in Tables 9-11.

Table 7. Number of learners by cluster

Number of Clusters C	Number of learners
2	16
3	12
4	11

Table 8. ID of K nearest learners by cluster number

Number of Clusters C	Number of the K nearest learners K					
	2		3		4	
	Id	D	Id	D	Id	d
2	1	31,02	1	31,02	1	31,02
	3	46,22	3	46,22	3	46,22
			10	46,32	10	46,32
					5	49,16
3	1	31,02	1	31,02	1	31,02
	2	46,22	2	46,22	2	46,22
			8	46,32	8	46,32
					4	49,16
4	1	31,02	1	31,02	1	31,02
	2	46,22	2	46,22	2	46,22
			8	46,32	8	46,32
					4	49,16

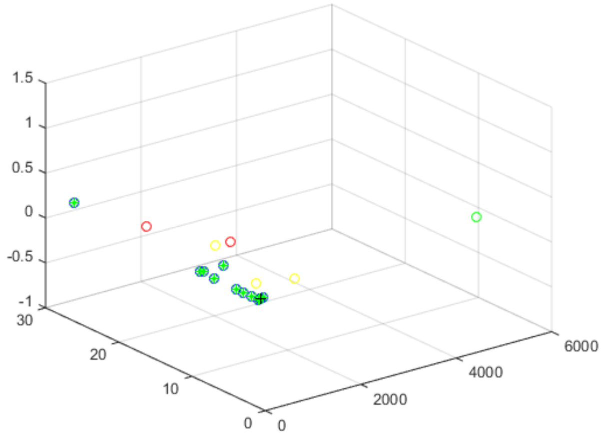


Figure 5. Repartition of learners after classification into 4 clusters

Table 9. ID of K nearest learners by cluster number (C = 2) for each moment t_i

Number of Clusters C	Number of the K nearest learners K=2				Number of the K nearest learners K=3				Number of the K nearest learners K=4			
	t_0		t_1		t_0		t_1		t_1		t_0	
	Id	d	Id	Id	D	Id	Id	d	Id	Id	d	Id
2	1	31	1	1,26	1	31	1	1,26	1	31	1	1,26
	12	32,01	3	4,47	12	32,01	3	4,47	12	32,01	3	4,47
					7	38,11	10	11	7	38,11	10	11
									4	41,77	5	16,29

Table 10. ID of K nearest learners by cluster number (C = 3) for each moment t_i

Number of Clusters C	Number of the K nearest learners K=2				Number of the K nearest learners K=3				Number of the K nearest learners K=4			
	t_0		t_1		t_0		t_1		t_1		t_0	
	Id	d	Id	d	Id	d	Id	D	Id	d	Id	d
3	1	31	1	1,26	1	31	1	1,26	1	31	1	1,26
	9	32,01	2	4,47	9	32,01	2	4,47	9	32,01	2	4,47
					6	38,11	8	11	6	38,11	8	11
									3	41,77	4	16,29

Table 11. ID of K nearest learners by cluster number (C = 4) for each moment t_i

Number of Clusters C	Number of the K nearest learners K=2				Number of the K nearest learners K=3				Number of the K nearest learners K=4			
	t_0		t_1		t_0		t_1		t_1		t_0	
	Id	d	Id	Id	d	Id	Id	d	Id	Id	d	Id
4	1	31	1	1,26	1	31	1	1,26	1	31	1	1,26
	12	32,01	3	4,47	12	32,01	3	4,47	9	32,01	2	4,47
					7	38,11	10	11	6	38,11	8	11
									3	41,77	4	16,29

4.4. Graphic representations & discussions

From Table 5 and Table 8, we generate the following graph as shown in Figure 6 to present the different learner Ids by distance using KNN and combining KNN with FCM. From Table 6 and Table 11, we generate the following graphs as shown in Figure 7 to present the different learner IDs by distance using KNN for the moment t_1 as shown in Figure 8.

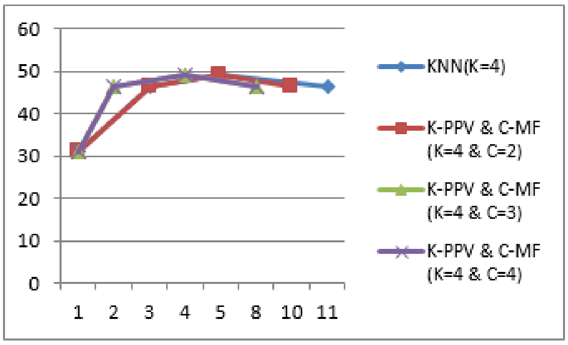


Figure 6. ID of learners according to distances according to KNN & combination FCM and KNN (with K = 4)

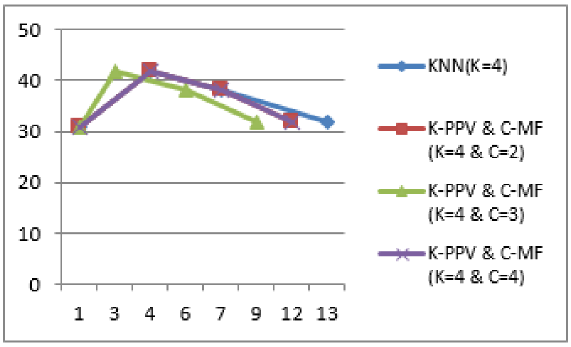


Figure 7. K (k = 4) nearest learners according to the combination of KNN and FCM (C = 2, 3, 4) for t_0

According to Figure 6, Figure 7 and Figure 8, the results obtained by KNN and the combination of KNN with FCM give similar results. But the results of the combination of KNN with FCM are more accurate. The learner Id 1 is the nearest learner to the learner Id 18 (in difficulty) at times t_0 and t_1, using KNN or the combination of KNN and FCM. The difference remains in the number of learners to compare and the time of calculation of similarity. Using the KNN method, the number of samples is 17 learners however with the combination of FCM and KNN methods, the number of sample is 11 learners which reduces the classification time.

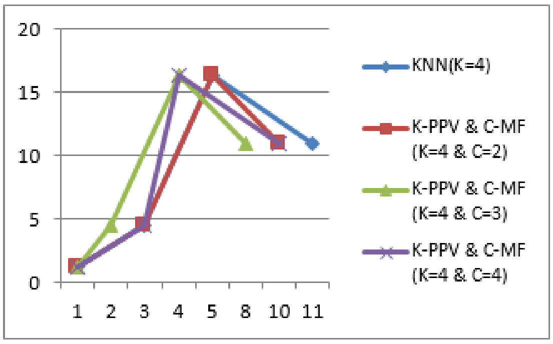


Figure 8. K (k = 4) nearest learners according to the combination of KNN and FCM (C = 2, 3, 4) for t_1

5. CONCLUSION & PERSPECTIVES

The real-time personalized follow-up of the learner in an adaptive learning system allows to offer him a learning adapted to his profile and his needs in order to avoid the problem of failure and abandonment. We have developed an approach to an adaptive learning system that supports and accompanies the learner in their learning process based on the IDCBR, which takes into account dynamic changes in learner behavior, the arrival of new data during the learning process and the successful experiences of other learners. For these reasons, we opted to use the IDCBR approach that integrate the KNN algorithm in the Retrieve step to search for learners with similar behaviors, these learners are grouped into homogeneous clusters using the FCM method.

This combination makes it possible on the one hand to offer a learning path adapted to the profile of the learner and based on the experiences of other learners with similar behavior by observing and analyzing their learning traces, and on the other hand, overcomes the limitations of the KNN algorithm in terms of computation time and memory. Our future work is to test our approach on a large number of learners, to organize the knowledge base (case base) to facilitate the search for learners with similar behaviors, to develop the different steps of the IDCBR and to associate a mobile agent at each step of the IDCBR cycle to perform specific tasks.

REFERENCES

[1] A. Fernandez, 2018. Available: https://www.piloter.org/business-intelligence/datamining.htm.

[2] Kolodner J. L., “Case-Based Reasoning,” Morgan Kaufmann, 1993.

[3] Mille A., “Traces bases reasoning (TBR) definition, illustration and echoes with storytelling,” Rapport Technique RR-LIRIS-2006-002, LIRIS UMR 5205 CNRS/INSA de Lyon/Université Claude Bernard Lyon1/Université Lumière Lyon 2/Ecole Centrale de Lyon, 2006.

[4] A. Zouhair, “Raisonnement à Partir de CasDynamique Multi-Agents: application à un système de tuteur intelligent»,” PhD in computer science, in Cotutelle between the Faculty of Sciences and Technologies of Tangier (Morocco) and the University of Le Havre (France), 2014.

[5] E. M. E. Naimi and A. Zouhair, “Intelligent dynamic case-based reasoning using multiagents system in adaptive e-service, e-commerce and e-learning systems,” *Int. J. of Knowledge and Learning (IJKL)*, vol. 11, pp. 42-57, 2016.

[6] Z. Zhang, “Introduction to machine learning: k-nearest neighbors,” *Annals of translational medicine*, vol. 4, 2016.

[7] C. Chesneau, “Éléments de classification,” Master, France, 2016.

[8] R. Duda and P. Hart, “Pattern Classification and Scene Analysis,” Wiley, New York, 1997.

[9] K. Teknomo, “K-Nearest Neighbors Tutorial.” http://people.revoledu.com/kardi/tutorial/KNN/index.html.

[10] R. Lajugie, 2017. Available: http://remi-lajugie.fr/docs/kPpv.pdf.

[11] B. Fayssal, “Conception d’un classifeur à base des règles floues,” PhD in computer science, University Aboubekr Belkaid, Tlemcen, Algeria, 2016.

[12] W. Yu and W. Zhengguo, “A Fast kNN algorithm for text categorization,” *Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong*, pp. 3436-3441, 2007.

[13] P. Indyk and R. Motwani, “Approximate nearest neighbor: towards removing the curse of dimensionality,” *Proc. 30th Annu. ACM Symp. Comput. Geometry*, pp. 604-613, 1998.

[14] A. Andoni and P. Indyk, “Ner-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions,” *Communications of the ACM*, vol. 51, pp. 117-122, 2008.

[15] W. Cherif, “Optimization of K-NN algorithm by clustering and reliability coefficients: application to breast-cancer diagnosis,” *Procedia Computer Science*, vol. 127, pp. 293-299, 2018.

[16] Y. A. Gerhana, *et al*., “The implementation of K-nearest neighbor algorithm in case-based reasoning model for forming automatic answer identity and searching answer similarity of algorithm case,” *Cyber and IT Service Management (CITSM), 2017 5th International Conference on*, pp. 1-5, 2017.

[17] M. B. Fouad, “La logique floue pour Classification Des Feuilles de vigne,” Mémoire Master Académique, Faculty of New Technologies of Information and Communication, University Kasdi Marbah, Ouargla, Algiria, 2015.

[18] J. Hathaway, *et al*., “On relational versions of c-means algorithm,” *Pattern Recognition Letters*, vol. 17, pp. 607-612, 1996.

[19] H. D. Cheng, *et al*., “Color image segmentation : advances and propects,” *Pattern Recognition*, vol. 34, pp. 2259-2281, 2001.

[20] A. Nabila and M. B. Amira, “Méthode coopérative pour la segmentation d’image : Application à l’imagerie médicale,” PhD, University Abderrahmane Mira-Bejaia, Faculty de Technologie, Algeria, 2016.

[21] S. Miyamoto and K. Umayahara, “Methods in Hard and Fuzzy Clustering,” in Z. Q. Liu and S. Miyamoto (Eds.), “Soft Computing and Human-Centered Machines,” Springer, pp. 85-129, 2000.

[22] A. Mahboub, *et al*., “Distributed energy efficient clustering algorithm based on fuzzy logic approach applied for heterogeneous WSN,” *Proceedings of the 2nd International Conference on Computing and Wireless Communication Systems*, pp. 38, 2017.

[23] A. M. Sîrbu, “Dynamic machine learning for supervised and unsupervised classification,” Doctoral dissertation, INSA de Rouen, 2016.

[24] R. M. Felder and L. K. Silverman, “Learning Styles and Teaching Styles in Engineering Education,” *Presented at the 1987 Annual Meeting of the American Institute of Chemical Engineers, New York*, 1987.

[25] N. Elghouch, *et al.*, “An application to index the didactic resources in an adaptive learning system (ALS),” *Information & Communication Technology and Accessibility, 5th International Conference on*, pp. 1-3, 2015.

[26] S. Thirumuruganathan, “A Detailed Introduction to K -Nearest Neighbor (KNN) Algorithm,” 2010.

BIOGRAPHIES OF AUTHORS

El Ghouch Nihad is a PhD student in LIST (Laboratoire d'Informatique, Systèmes et Télécommunications) Laboratory, Department of Computer Science. Faculty of Sciences and Techniques, Tangier, Morocco, atthe University of Abdelmalek Essaadi. The research topics of interest are adaptive learning, personalized learning.

Dr. El Mokhtar EN-NAIMI is a Professor in UAE, Faculty of Sciences and Technologies of Tangier, Department of Computer Science, since 2000 (Temporary: from 2000 to 2003 and Professor Permanent: since 2003/2004 until Now). He is a Head of Computer Science Department, since October 2016 until now. He was responsible for a Bachelor of Science and Technology, BST Computer Engineering (“Licence LST-GI”), from January 2012 to October 2016. He is also a founding member of the Laboratory LIST (Laboratoire d'Informatique, Systèmes et Télécommunications), the University of Abdelmalek Essaâdi, FST of Tangier, Morocco. He is also an Expert Evaluator with the ANEAQ, since the academic year 2016/2017 until now, that an Expert of the Private Establishments belonging to the territory of the UAE and also an Expert of the Initial or Fundamental Formations and Formations Continuous at the Ministry of Higher Education, Scientific Research and Executive Training and also at the UAE University and the FST Tangier since 2012/2013 until Now. He is an Author/Co-Author of several Articles, published in The International Journals in Computer Science, in particular, in MAS, Cases Based Reasoning (CBR), eLearning, MOOC, Big DATA, Data-mining, Wireless Sensor Network, VANet, MANet, Smart City, …, etc. He is also Director of several Doctoral Theses in Computer Science. In addition, he is an associate member of the ISCN - Institute of Complex Systems in Normandy, the University of the Havre, France, since 2009 until Now.

Dr. Zouhair Abdelhamid is a Professor of Computer Science at The National School of Applied Sciences, Al-Hoceima, UAE, Morocco. Since March 2012-10 May 2016: Quality Project Manager, Geographic Information System and Computer head at the Urban Agency of Tetouan, Ministry of Urban Planning and Development. August 2011- March 2012: Senior Executive at the Urban Agency of Tetouan. July 2003- September 2008: IT Manager at Tronico Atlas, Tangier, Morocco. PhD in Computer Science at the University of The Havre, laboratory LITIS, France, and at the, Faculty of Sciences and Technologies of Tangier, University of Abdelmalek Essaâdi, Tangier, Morocco (cotutelle doctoral program) in October 2014. He is an author of several articles in Computer Science.

Dr. Al Achhab Moahmmed received his PhD in December 2006 from the University of Franche-Comté, Besançon, France, in the field of formal verification of reactive systems. He received a Master degree in July 2003 from University of Franche-Comté, in the field of software engineering and artificial intelligence. He was Temporary Lecturer and Research Assistant, at the University of Franche-Comté. He was an assistant professor at Faculty of Sciences Dhar El mehraz, Fez from 2007 to 2012. Currently, he is a professor at the National School of Applied Sciences of Tetuan. He is a member of the steering committee of the department of computer sciences and member of the National School board. His research focuses on analysis and validation of business process, natural language processing, case of study Arabic and adaptive e-learning. He is the co-author of more than 20 peer-reviewed publications. To his credit also, is being Member in many scientific associations notably IEEE, innove, and mocit. He has been participating in many national and international scientific and organizing committees. He is the mentor of the IEEE UAE Student Branch since 2015. He is the conference chair of the 2014 and 2016 IEEE CiSt editions…, etc.