Computers & Education 53 (2009) 273–285

Contents lists available at ScienceDirect

Computers & Education

journal homepage: www.elsevier.com/locate/compedu

A learning style classification mechanism for e-learning

Yi-Chun Chang [a,*], Wen-Yan Kao [a], Chih-Ping Chu [a], Chiung-Hui Chiu [b]

[a] Department of Computer Science and Information Engineering, National Cheng-Kung University, Tainan, Taiwan
[b] Graduate Institute of Information and Computer Education, National Taiwan Normal University, Taipei, Taiwan

ARTICLE INFO

Article history:
Received 8 January 2007
Received in revised form 26 January 2009
Accepted 2 February 2009

Keywords:
Adaptive learning
Genetic algorithm (GA)
k-Nearest neighbor classification
Learning style
E-learning

ABSTRACT

With the growing demand in e-learning, numerous research works have been done to enhance teaching quality in e-learning environments. Among these studies, researchers have indicated that adaptive learning is a critical requirement for promoting the learning performance of students. Adaptive learning provides adaptive learning materials, learning strategies and/or courses according to a student's learning style. Hence, the first step for achieving adaptive learning environments is to identify students' learning styles. This paper proposes a learning style classification mechanism to classify and then identify students' learning styles. The proposed mechanism improves *k*-nearest neighbor (*k*-NN) classification and combines it with genetic algorithms (GA). To demonstrate the viability of the proposed mechanism, the proposed mechanism is implemented on an open-learning management system. The learning behavioral features of 117 elementary school students are collected and then classified by the proposed mechanism. The experimental results indicate that the proposed classification mechanism can effectively classify and identify students' learning styles.

1. Introduction

With the popularization of Internet, the demand of e-learning has greatly increased (Allen & Seaman, 2003; Gerald & Hussar, 2003; Waits & Lewis, 2003; Wirt, Choy, Rooney, Provasnik, & Tobin, 2004). Numerous research works regarding to e-learning have been done to enhance teaching quality in e-learning environments. Among these studies, researchers have indicated that adaptive learning is a critical requirement for promoting the learning performance of students (Brusilovsky, 1999; Brusilovsky, Eklund, & Schwarz, 1998; Brusilovsky & Maybury, 2002; Graf & Kinshuk, 2006; Sessink, Beeftink, Tramper, & Hartog, 2003). Adaptive learning provides the adaptive learning materials, learning strategies and/or courses according to a student's learning style (Brusilovsky, 1999; Brusilovsky et al., 1998; Brusilovsky & Maybury, 2002; Carver, Howard, & Lane, 1999; Shang, Shi, & Chen, 2001; Pena, Narzo, & Rosa, 2002; Trantafillou, Poportsis, & Demetriadis, 2003). Learning style is an indicator of how a student learns and likes to learn, and how an instructor teaches to successfully address the needs of the individual students (Gregorc & Ward, 1977; Keefe, 1987; Tseng, Chu, Hwang, & Tsai, 2008). Hence, the first step for achieving adaptive learning environments is to identify students' learning styles. Several learning style models have been proposed for defining and measuring learning styles (Dunn, Dunn, & Price, 1984; Felder & Silverman, 1988; Keefe, 1987; Kolb, 1984). Depending on one learning style model, numerous research works also have provided mechanisms to detect and identify learning styles for achieving an adaptive e-learning environment (Chen, Lee, & Chen, 2005; García, Amandi, Schiaffino, & Campo, 2007; Schiaffino, Garcia, & Amandi, 2008; Xenos, 2004). These mechanisms need to be based on a large number of student's samples. As a result, the collecting process of a large number of student's samples is time-consuming and the processing of these students' samples is complicated.

Hence, to solve the aforementioned weaknesses, this paper proposes a learning style classification mechanism to classify and then identify a student's learning style. This paper uses the classification technique to classify learning styles. One of most popular classification techniques is *k*-nearest neighbor (*k*-NN) classification (Chen, Yen, & Ho, 2004; Kuncheva & Jain, 1999). However, *k*-NN classification has some weaknesses. Therefore, the proposed mechanism improves *k*-NN classification (Chen et al., 2004; Kuncheva & Jain, 1999) and combines it with genetic algorithms (GA) (Holland, 1975; Kuncheva, 1995, 1997; Mitchell, 1997). Since *k*-NN classification only needs a small amount of samples, the weaknesses of aforementioned research works can be solved. Furthermore, the classification mechanism proposed in this paper is independent of learning style models. That is, the proposed mechanism possesses compatibility, which can be adopted for any learning style model according to users' demand. To demonstrate the viability of the proposed mechanism, this paper implements the

* Corresponding author.
E-mail addresses: changyj@csie.ncku.edu.tw (Y.-C. Chang), chucp@csie.ncku.edu.tw (C.-P. Chu).

0360-1315/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compedu.2009.02.008

proposed mechanism on a Learning Management System (LMS) that conforms to Shareable Content Object Reference Model (SCORM) e-learning standard (SCORM, 2004). After the learning behavioral features of 117 elementary school students are collected, the proposed mechanism is used to classify the learning styles of 117 students to verify the classification accuracy of the proposed mechanism.

The rest of this paper is organized as follows: Section 2 briefly reviews related works. Section 3 presents the learning style classification mechanism in detail. Section 4 demonstrates the experimental results. Finally, Section 5 provides a conclusion. Besides, Appendix presents an illustrative example, which demonstrates how the proposed mechanism works.

2. Related works

2.1. Learning style

As the descriptions in Section 1, learning style indicates how a student learns and likes to learn, and facilitates that an instructor can successfully teach to adapt to individual students (Gregorc & Ward, 1977; Keefe, 1987; Tseng et al., 2008). Each learning style involves differently behavioral features that can be collected and analyzed from the learning behavior of a student.

There have been several models for defining and measuring learning styles, such as (1) Kolb (1984) proposed that learners can be classified into convergent learners, divergent learners, assimilators, and accommodators (Kolb, 1984). (2) In Keefe's learning style test, the learner can be identified into Sequential Processing Skill, Discrimination Skill, Analytic Skill and Spatial Skill (Keefe, 1987). (3) Felder & Silverman's model comprises the category of intuitive/sensitive, global/sequential, visual/verbal, inductive/deductive and active/reflective, which can be used to discriminate 32 learning styles. For example, the sensitive/ sequential/ verbal/ deductive/ active is a learning style (Felder & Silverman, 1988). (4) Stangl (2002) distinguished learners into four styles, i.e., acting, hearing, reading and seeing (Stangl, 2002).

2.2. The basic concepts of genetic algorithms (GA)

Genetic algorithm (GA) is proposed by Holland in 1975 (Holland, 1975), which is widely used to solve an optimization problem by a systematic way (Hwang, Yin, Wang, Tseng, & Hwang, 2008). The standard GA consists of several executing steps, namely chromosome encoding and population initialization, fitness evaluation, selection, crossover and mutation (Holland, 1975; Mitchell, 1997; Rothlauf, 2006; Yao, 1999):

1. *Chromosome encoding and population initialization*: Chromosome encoding is for transferring a candidate solution into a chromosome according to an optimization problem. A chromosome consists of several genes. Depending on an optimization problem, a gene can be a binary bit, an integer or a real number. A population is composed of several chromosomes. An initial population is usually randomly generated. Through a number of genetic evolutional processes, the population evolves from one generation to next generation to improve the quality of chromosomes. The population size, i.e., the number of chromosomes in a population, is kept constant.
2. *Fitness evaluation*: To evaluate the fitness of each chromosome, a fitness function needs to be defined according to an optimization problem. The fitness value derived from the fitness function can be used to determine which chromosomes are better solutions for the optimization problem.
3. *Selection*: According to roulette selection (Holland, 1975), the chromosomes that have fitter fitness values possess higher probability to be selected to propagate offspring.
4. *Crossover*: It is a genetic evolutional process, in which each individual has a chance to interchange gene information from two parent chromosomes. Crossover is performed with a crossover probability. A random number can be generated between [0.0, 1.0] for each mating pair. If the random number is less than the crossover probability, the crossover is performed to propagate offspring. Otherwise, no crossover is performed.
   There are two broadly adopted crossover operators (Syswerda, 1989). One is the single point crossover, which yields offspring by interchanging all the genes after a random position from the parent chromosomes, as Fig. 1a depicts. The other is the two-point crossover, which generates two random positions and interchanges the genes between the two positions from the parent chromosomes, as Fig. 1b depicts.

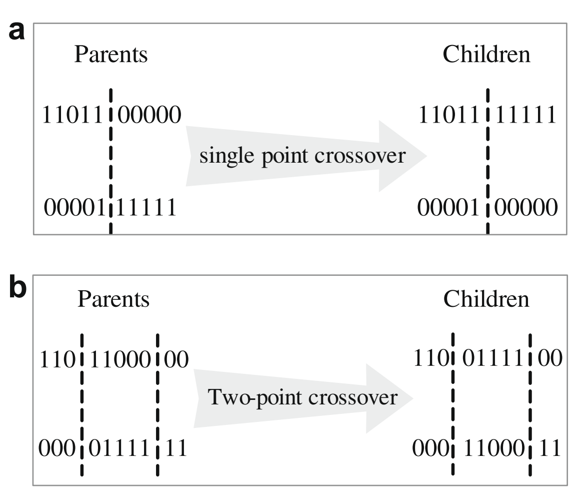

Fig. 1. An illustrative example of crossover

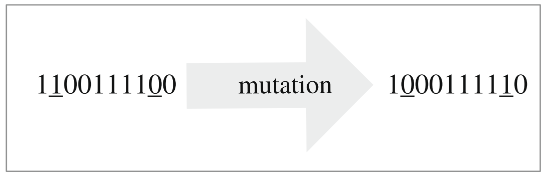

Fig. 2. An illustrative example of mutation

5. *Mutation*: It also is a genetic evolutional process, which creates a new gene that does not inherit from the parent chromosomes. Every bit in an offspring chromosome could mutate with a predefined mutation probability. In general, mutation is performed with a very low mutation probability. For performing mutation, two genes are randomly selected to exchange their position in a chromosome. Fig. 2 is an example of mutation, where the mutated gene positions are underlined.
6. *Stop criterion*: In general, the stop criterion is 100 generations. That is, the steps 2–5 iteratively are performed 100 times. After 100 generations, a near optimal solution can be derived by the fitness function.

3. The learning style classification mechanism using enhanced *k*-NN classification combined with GA

In this section, a learning style classification mechanism based on *k*-NN classification is proposed. Since we observe that some weaknesses exist in *k*-NN classification, an enhanced *k*-NN classification is proposed and GA is employed to improve *k*-NN classification. Firstly, the basic definition of *k*-NN classification is presented. Then, how the enhanced *k*-NN classification and GA solve the weaknesses of *k*-NN classification are explained. Finally, the learning style classification mechanism combined the enhance *k*-NN classification with GA is presented in details.

3.1. Basic definitions of k-NN classification

This paper uses the classification technique to classify learning styles. One of most popular classification techniques is *k*-NN classification (Chen et al., 2004; Kuncheva & Jain, 1999). For *k*-NN classification, a sample can be represented as a point in an *n*-dimensional space. If two similar samples belong to the same class, their distance is shorter than that of other samples that belong to other class. An unclassified data sample can be determined a class by computing the distance between it and the samples. Suppose that an unclassified data sample contains N features, which is denoted as $[a_1(x_u), a_2(x_u), \ldots, a_n(x_u)]$, $1 \leqslant n \leqslant N$, where $a_n(x_u)$ denotes the *n*th feature value for the unclassified data sample x_u. Similarly, a sample x_s also contains N features, which is denoted as $[a_1(x_s), a_2(x_s), \ldots, a_n(x_s)]$, $1 \leqslant n \leqslant N$, where $a_n(x_s)$ denotes the *n*th feature value for the sample x_s. The distance between an unclassified data sample x_u and a sample x_s can be derived using

$$d(x_u, x_s) = \sqrt{\sum_{n=1}^{N} (a_n(x_u) - a_n(x_s))^2} \tag{1}$$

This paper adopts the function of *k*-NN classification as (Mitchell, 1997)

$$f(x_u) \leftarrow c_t$$
$$\sum_{i=1}^{k} g(c_t, f(x_i)) = \max\left\{\sum_{i=1}^{k} g(c_t, f(x_i)) | c_t \in C = \{c_1, c_2, \ldots, c_m\},\ 2 \leqslant m \leqslant M,\ \text{if } a = b,\ \text{then } g(a,b) = 1;\ \text{if} a \neq b,\ \text{then } g(a,b) = 0\right\} \tag{2}$$

where $f(x)$ denotes the class to which x belongs. x_i, $1 \leqslant i \leqslant k$, denotes k closest samples. $g(a,b)$ is a function that determines whether or not a and b belong to the same class. If the result is positive, $g(a,b) = 1$. Otherwise, $g(a,b) = 0$. C denotes the set of classes and M is the number of classes.

After being computed, the k closest samples for the unclassified data sample x_u can be listed. A set $S = \{(c_1,R_1),(c_2,R_2),\ldots,(c_m,R_m)\}$ and $R_1 + R_2 + \cdots + R_m = k$ can be given, where (c, R) represents that the class c contains R samples. If $R_1 \geqslant R_2 \geqslant \cdots \geqslant R_m$, the class ranks for the unclassified data sample x_u are $\{c_1, c_2 \ldots, c_m\}$, it means that x_u has the higher priority belonging to the class c_1.

3.2. Enhanced k-NN classification

In this paper, since *k*-NN classification is used to classify and identify a student's learning style, the involved parameters are defined as follows:

- $C = \{c_1, c_2, \ldots, c_m\}$, $2 \leqslant m \leqslant M$: denotes the set of classes, where M is the number of leaning styles. A class represents a learning style.
- $BF = \{bf_1, bf_2, \ldots, bf_n\}$, $1 \leqslant n \leqslant N$: denotes the set of learning behavioral features, where N is the number of leaning behavioral features. bf_n denotes the *n*th learning behavioral feature.
- L_{uc}: denotes an unclassified student. Each unclassified student contains N learning behavioral features, which is denoted as $[bf_1(L_{uc}), bf_2(L_{uc}), \ldots, bf_n(L_{uc})]$, $1 \leqslant n \leqslant N$. $bf_n(L_{uc})$ denotes the value of the *n*th learning behavioral feature bf_n for the unclassified student L_{uc}.

- DB_{sample}: denotes a sample database, which is represented as

$$DB_{sample} = \{L_{s\_1}, L_{s\_2}, \ldots, L_{s\_z}\} = \begin{bmatrix} bf_1(L_{s\_1}), bf_2(L_{s\_1}), \ldots, bf_n(L_{s\_1}), c_{s\_1} \\ bf_1(L_{s\_2}), bf_2(L_{s\_2}), \ldots, bf_n(L_{s\_2}), c_{s\_2} \\ \cdots \\ bf_1(L_{s\_z}), bf_2(L_{s\_z}), \ldots, bf_n(L_{s\_z}), c_{s\_z} \end{bmatrix}, \quad C_{s\_z} \in C,\ 1 \leqslant z \leqslant Z \tag{3}$$

where Z denotes the number of samples in the sample database DB_{sample}. L_{s_z} denotes the zth sample that belongs to the class $c_{s\_z}bf_n(L_{s\_z})$ denotes the value of the nth learning behavioral feature bf_n for the sample L_{s_z}.

According to Eq. (1), the distance between the unclassified student L_{uc} and the sample L_{s_z} can be derived using

$$d(L_{uc}, L_{s\_z}) = \sqrt{\sum_{n=1}^{N} (bf_n(L_{uc}) - bf_n(L_{s\_z}))^2} \tag{4}$$

Similarly, according to Eq. (2), the function of k-NN classification in this paper is denotes as (Mitchell, 1997)

$$f(L_{uc}) \leftarrow c_t$$

$$\sum^{k} g(c_t, f(L_{s\_z})) = \max\left\{\sum^{k} g(c_t, f(L_{s\_z})) | c_t \in C = \{c_1, c_2, \ldots, c_m\},\ 2 \leqslant m \leqslant M,\ \text{if } a = b,\ \text{then } g(a,b) = 1;\ \text{if } a \in b,\ \text{then } g(a,b) = 0\right\} \tag{5}$$

After being computed, the k closest samples for the unclassified student L_{uc} can be listed. Given $S = \{(c_1,R_1),(c_2,R_2),\ldots,(c_m,R_m)\}$ and $R_1 + R_2 + \cdots + R_m = k$, where (c,R) represents that the class c contains R samples. If $R_1 \geqslant R_2 \geqslant \cdots \geqslant R_m$, the class ranks for the unclassified student L_{uc} are listed as $\{c_1,c_2,\ldots,c_m\}$, it means that the unclassified student L_{uc} has the higher priority belonging to the class c_1.

We observe that k-NN classification has three weaknesses:

(I) If a large number of learning behavioral features need to be considered, it results in a heavy computation complexity.
(II) When there is a large quantity of samples, the process of computing distance is time-consuming.
(III) After the executing process of k-NN classification, $S = \{(c_1,R_1),(c_2,R_2),\ldots,(c_m,R_m)\}$ can be derived. In S, more than one class has the same number of samples. In this situation, the priority order in the class ranks is difficult to be determined.

Hence, to solve weakness (I), this paper employs GA to extract the learning behavioral features and then to reduce computation complexity, which is presented in next sub-section. Besides, this paper proposes two algorithms, i.e., Pre-Contrast algorithm and Post-Comparison algorithm, to improve weakness (II) and (III), respectively.

(1) *Pre-Contrast algorithm*

Pre-Contrast algorithm is defined in Algorithm_1 to improve the weakness (II), which is executed before k-NN classification. As Algorithm_1 depicts, by contrasting the previously classified records, Pre-Contrast algorithm can reduce the time for computing the distance between an unclassified data and samples.

In Pre-Contrast algorithm, an error check table T_{check} is for recording that the students have been classified. The vector space of the error check table T_{check} is defined as:

$$T_{check} = \{L_{ch\_1}, L_{ch\_2}, \ldots, L_{ch\_y}\} = \begin{bmatrix} bf_1(L_{ch\_1}), bf_2(L_{ch\_1}), \ldots, bf_n(L_{ch\_1}), C_{error\_1}, C_{ch\_1} \\ bf_1(L_{ch\_2}), bf_2(L_{ch\_2}), \ldots, bf_n(L_{ch\_2}), C_{error\_1}, C_{ch\_2} \\ \cdots \\ bf_1(L_{ch\_y}), bf_2(L_{ch\_y}), \ldots, bf_n(L_{ch\_y}), C_{error\_y}, C_{ch\_y} \end{bmatrix}, \quad 1 \leqslant y \leqslant Y, \tag{6}$$

where Y denotes the number of records in the error check table T_{check}. L_{ch_y} denotes to the yth record. $bf_n(L_{ch\_y})$ denotes the value of the nth learning behavioral feature bf_n for the record L_{ch_y}. C_{error_y} denotes the incorrect class ranks for the yth record. C_{ch_y} denotes the correct class ranks for the yth record.

Algorithm_1. Pre_Contrast (L_{uc}, T_{check})

if (T_{check} is not empty) **then**
  $y \leftarrow 1$;
    do ([$bf_1(L_{uc})$, $bf_2(L_{uc})$, ..., $bf_n(L_{uc})$] is contrasted with $[bf_1(L_{ch\_y}), bf_2(L_{ch\_y}), \ldots, bf_n(L_{ch\_y})]$;)
      if $([bf_1(L_{uc}), bf_2(L_{uc}), \ldots, bf_n(L_{uc})] = [(bf_1(L_{ch\_y}), bf_2(L_{ch\_y}), \ldots, bf_n(L_{ch\_y})])$ **then**
      $L_{uc} \leftarrow C_{ch\_y}$;
    endif
    $y \leftarrow y + 1$;
  while (($y > Y$) **or**
      $([bf_1(L_{uc}), bf_2(L_{uc}), \ldots, bf_n(L_{uc})] = [(bf_1(L_{ch\_y}), bf_2(L_{ch\_y}), \ldots, bf_n(L_{ch\_y})])$ is found;)
endif

As Algorithm_1 depicts, the learning behavioral features for the unclassified student L_{uc} are contrasted with the records in the error check table T_{check}. If the matched record exists in T_{check}, the class ranks for the unclassified student L_{uc} can be obtained. Since the class ranks for the unclassified student L_{uc} is obtained, the executing process of k-NN classification can be omitted. Hence, the time for computing the distance between an unclassified data and samples decreases.

(2) *Post-Comparison Algorithm*

Post-Comparison algorithm is defined in Algorithm_2 to solve the weakness (III). Post-Comparison algorithm is executed after the executing process of k-NN classification. In Algorithm_2, $Sort(D_{aver\_Cr,}\ S)$ is used to compare and sort the average distance between the unclassified student L_{uc} and the samples belonging to the class c_r, where c_r represents the classes that have the same number of samples in k-NN classification result $S = \{(c_1,R_1),(c_2,R_2),\ldots,(c_m,R_m)\}$, $R_1 + R_2 + \cdots + R_m = k$. The average distance D_{aver_Cr} between the unclassified student L_{uc} and the samples belonging to the class c_r can be derived using

$$D_{aver\_c_r} = \frac{\sum_{z=1}^{Z}[g(c_r,f(L_{s\_z})) \times d(L_{uc},L_{s\_z})]}{\sum_{z=1}^{Z} g(c_r,f(L_{s\_z}))}, \quad 1 \leqslant r \leqslant M, \tag{7}$$

where $g(c_r,f(L_{s\_z}))$ is used to determine whether the sample L_{s_z} belongs to the class c_r. If the result is positive, $g(c_r,f(L_{s_z})) = 1$. Otherwise, $g(c_r,f(L_{s_z})) = 0$.

According to the sorted results derived by $Sort(D_{aver\_Cr,}S)$, $Sort(c_{r,}S)$ sorts the priority order in the class ranks. The class, which average distance is shorter than that of others, has higher priority order in the class ranks.

Algorithm_2: Post_Comparison (S)

Sort(Daver_cr, S);
Sort(c_r, S)

Combined Pre-Contrast algorithm with Post-Comparison algorithm, the enhanced k-NN classification is defined in Algorithm_3. In Algorithm_3, $k\_NN(L_{uc},\ DB_{sample})$ denotes the execution of k-NN classification as Eq. (5) depicts. Before the execution of $k\_NN(L_{uc},\ DB_{sample})$, $Pre\_Contrast(L_{uc},\ T_{check})$ is executed as long as that the error check table T_{check} contains any record. If the matched record cannot be obtained from the error check table T_{check} and the number of samples for each class in DB_{sample} is more than $\lceil \frac{k}{2} \rceil$, $k\_NN(L_{uc},\ DB_{sample})$ is executed to find the class ranks for the unclassified student L_{uc}. If more than one class that has the same number of samples exists in the executed result of $k\_NN(L_{uc},\ DB_{sample})$, $Post\_Comparison(S)$ is executed to sort appropriate class ranks.

Algorithm_3: $Enhanced_k$NN(L_{uc}, T_{check}, DB_{sample})

if ($T_{check} \neq$ empty;) **then**
  $Pre\_Contrast(L_{uc},\ T_{check})$;
endif
if (the match L_{ch_y} does not be found in T_{check} and
  the number of samples for every class in the $DB_{sample} \geqslant \lceil \frac{k}{2} \rceil$) **then**
  $k\_NN(L_{uc},\ DB_{sample})$;
if (more than one class have the same number of samples in S;) **then**
  $Post\_Comparison(S)$;
  endif
endif

3.3. GA-based behavioral feature extraction

This paper employs GA to extract learning behavior features and then reduce the time of computing distance, which is used to solve the weakness (I) as Section 3.2 presents. The execution procedures of GA-based behavioral feature extraction are presented as follows:

Step G1. *Feature value normalizing*: The normalized formula, which normalize the value of behavioral feature of samples between 0 to 1, can be derived as:

$$bf_n(L_{s\_z})' = \frac{bf_n(L_{s\_z}) - \min_{i=1\sim Z}(bf_n(L_{s\_i}))}{\max_{i=1\sim Z}(bf_n(L_{s\_i}) - \min_{i=1\sim Z}(bf_n(L_{s\_i}))}, \tag{8}$$

where $1 \leqslant n \leqslant N$ and $1 \leqslant z \leqslant Z$. $bf_n(L_{s\_z})'$ denotes the normalized value of the nth learning behavioral feature bf_n for the zth sample L_{s_z}. $\min_{i=1\sim Z}(bf_n(L_{s\_i}))$ denotes the minimum of the nth learning behavioral feature in DB_{sample}. $\max_{i=1\sim Z}(bf_n(L_{s\_i}))$ denotes the maximum of the nth learning behavioral feature in DB_{sample}.

Step G2. *Training samples and testing samples divided*: The samples in DB_{sample} are randomly divided into the training sample set $db_{training}$ and the testing sample set $db_{testing}$. In $db_{training}$, the number of training samples for each class must be the same, which needs to contain $\lceil \frac{Z}{2M} \rceil$ training samples. The intersection of the training sample set $db_{training}$ and the testing sample set $db_{testing}$ is empty, i.e., $db_{training} \cap db_{testing} = \emptyset$.

Step G3. *Chromosome encoding and population initialization*: In this paper, a chromosome is denoted as $\{x_1,x_2,\ldots,x_n\}$, $1 \leqslant n \leqslant N$, where N is the number of the learning behavioral feature. x_n denotes a gene, which is a binary value. If a chromosome contains the nth learning behavioral feature, $x_n = 1$. Otherwise, $x_n = 0$. The initial population is randomly generated.

Step G4. *Fitness evaluation*: The fitness function is derived as (Mitchell, 1997)

$$F(BF) = CA_{1\text{-}NN}(db_{training}, BF) - \alpha \times card(BF) \tag{9}$$

where $0 \leqslant CA_{1-NN}(db_{training},\ BF) \leqslant 1$. $CA_{1-NN}(db_{training},\ BF)$ denotes the classification accuracy that using 1-NN classification classifies the samples in the testing sample set $db_{testing}$, which can be derived using

$$CA_{1\text{-}NN}(db_{training}, BF) = \frac{C_{correct}}{C_{total}}\%, \tag{10}$$

where C_{total} denotes the total number of classification times. $C_{correct}$ denotes the number of correct classification times. α is a coefficient that is an optimum value obtained from experimenting. $card(BF)$ denotes the number of learning behavior features that is selected into a chromosome. As Eq. (9) depicts, the fitness function in this paper simultaneously considers the number of behavioral features and the classification accuracy. Hence, employing GA in this paper not only can reduce the number of the learning behavioral features and then speed up the execution of k-NN classification, but also can enhance the classification accuracy of k-NN classification.

Step G5. *Selection*: As Section 2.2 presents, this paper adopts roulette selection that a fitter chromosome is selected to propagate offspring. Given the set of the fitness value for Q chromosomes in a population is $\{f_1(BF), f_2(BF), \ldots, f_q(BF)\}$, $1 \leqslant q \leqslant Q$. The probability $Prob(f_q(BF))$ that the qth chromosome is selected to propagate child chromosomes is defined as

$$Prob(f_q(BF)) = \frac{f_q(BF)}{\sum_{i=1}^{Q} f_i(BF)}, \quad 1 \leqslant q \leqslant Q. \tag{11}$$

Step G6. *Crossover*: This paper adopts two-point crossover, which is depicted in Fig. 2b. The crossover probability is 0.7.

Step G7. *Mutation*: As Fig. 2 in Section 2.2 presents, for performing mutation, two genes are randomly selected to exchange their position in a chromosome. The mutation probability is 0.1.

Step G8. *Stop Criterion*: The evolution population is 100 generations.

3.4. The learning style classification mechanism

This sub-section presents the classification mechanism to classify and identify student's learning styles, which combines the enhanced k-NN classification with GA. Algorithm_4 is defined to carry out the classification mechanism. The involved parameters are defined as follows:

- $A = \{a_1, a_2, \ldots, a_m\}$, $1 \leqslant m \leqslant M$ denotes the set of learning materials, where a_m is the mth learning material that is designed and developed according to the class c_m.
- $GA(DB_{sample}, BF)$ denotes the execution of GA-based behavioral feature extraction, as Section 3.3 presents.
- $CA_{k\text{-}NN}(DB_{sample}, BF)_T$ denotes the present classification accuracy. According to Eq. (10), the present classification accuracy $CA_{k\text{-}NN}(DB_{sample}, BF)_T$ can be derived using

$$CA_{k-NN}(DB_{sample}, BF)_T = \frac{C_{correct}}{C_{total}}\%. \tag{12}$$

- $CA_{k\text{-}NN}(DB_{sample}, BF)_{T\text{-}1}$ denotes the previous classification accuracy.

The executing procedures of Algorithm_4 are presented as follows:

Step 1. Enhanced_kNN(L_{uc}, T_{check}, DB_{sample}) is executed to determine the class ranks for the unclassified student L_{uc}.

Step 2. If the class ranks for the unclassified student L_{uc} can be obtained, the adaptive learning material a_m is provided for L_{uc} to read according to the priority order in the class ranks. If the result is negative, more samples need to be collected. Hence, a_m is randomly selected for L_{uc} to read.

Step 3. After the unclassified student L_{uc} has read the adaptive learning material a_m, the assessments and/or the questionnaires are provided for testing whether or not a_m is suitable for L_{uc}, where the assessments need to regard the adaptive learning material a_m and the questionnaires are to ask a leaner whether or not a_m is suitable for her/him.

Step 4. If a_m is suitable for the unclassified student L_{uc}, both the behavioral features for L_{uc} and the class c_m corresponding to a_m are added into DB_{sample} as a sample. Besides, the behavioral features and the class ranks for L_{uc} are recorded into the error check table T_{check} for a correct record.

```
Algorithm_4: the learning style classification mechanism
Step 1: Enhanced_kNN(L_uc, T_check, DB_sample);
Step 2: if (the class ranks for L_uc can be obtained from Step_1) then
      L_uc ← a_m; // a_m conforms to the class ranks for L_uc
    else
      L_uc ← Random(A)= a_m; // a_m is randomly selected
    endif
Step 3: The assessments and the questionnaires are provided for testing whether or not a_m is suitable for L_uc;
Step 4: if (a_m is suitable for L_uc.) then
    z ← Z + 1;
    L_s_z ← {L_uc, c_m};
    if (the class ranks for L_uc is obtained in Step_1) then
      y ← Y + 1;
      [bf_1(L_ch_y), bf_2(L_ch_y), ..., bf_n(L_ch_y)] ← [bf_1(L_uc), bf_2(L_uc), ..., bf_n(L_uc)];
      C_ch_y ← the class ranks for L_uc;
    endif
  else
      y ← Y + 1;
      [bf_1(L_ch_y), bf_2(L_ch_y), ..., bf_n(L_ch_y)] ← [bf_1(L_uc), bf_2(L_uc), ..., bf_n(L_uc)];
      C_error_y ← the class ranks for L_uc;
```

Algorithm 4 (*continued*)

```
    return to Step2 when L_uc reads the next learning unit;)
  endif
Step 5: if (CA_k-NN(DB_sample, BF)_T ≥ CA_k-NN(DB_sample, BF)_T−1) then
    CA_k-NN(DB_sample, BF)_T−1 ← CA_k-NN(DB_sample, BF)_T;
else
    GA(DB_sample, BF);
    T_check ← empty;
  endif
```

Otherwise, the behavioral features for the unclassified student L_{uc} and her/his class ranks are recorded into the error check table T_{check} for a record that has been classified into incorrect class. In the next learning unit, to find an appropriate learning style that matches the unclassified student L_{uc}, steps 2–4 need to be performed iteratively until an appropriate learning style is found.

Step 5. If the classification accuracy $CA_{k\text{-}NN}(DB_{sample}, BF)_T$ is lower than the classification accuracy in the last time $CA_{k\text{-}NN}(DB_{sample}, BF)_{T-1}$, $GA(DB_{sample}, BF)$ is executed to extract behavioral features. After $GA(DB_{sample}, BF)$ executed, the records in the error check table T_{check} needs to be deleted. For the next unclassified student, the proposed mechanism only considers the extracted behavioral features to classify her/his learning style.

4. Experimental results

This section demonstrates the experimental results for the proposed mechanism, which are based on two datasets: (1) Iris dataset that is available at UCI repository; and (2) actual student data collected from the learning behavioral features of 117 elementary school students.

4.1. Iris dataset experiment

Iris dataset is the most accurate and reliable pattern, which is used widely to verify the research works regarding person identification (Huang, Wang, Tan, & Cui, 2004; Jain, Bolle, & Pankanti, 1999; Wang, 2008). Hence, to find the fitter coefficient α in Eq. (9), this experiment takes Iris dataset with different α values to observe the classification accuracy of the proposed mechanism. Iris dataset contains three classes that comprise respectively 50 samples, i.e., the amount of samples in Iris dataset class is 150. Every sample comprises four real-valued features. For the different α values – 0, 0.005, 0.01, 0.05 and 0.1, the proposed mechanism is executed 10 times. For each experiment among 10 times, 50 samples in Iris dataset are randomly selected to be the testing samples, and the remaining 100 samples are put into the sample database DB_{sample}. According to Eq. (12), the classification accuracy $CA_{k\text{-NN}}(DB_{sample}, BF)$ can be derived using

$$CA_{k-NN}(DB_{sample}, BF) = \frac{C_{correct}}{C_{total}}\% = \frac{C_{correct}}{50}\%, \tag{13}$$

where $C_{correct}$ denotes the number of correct classification times among 50 samples. Since 50 testing samples are classified, the total number of classification times C_{total} is 50. Table 1 depicts the classification results, which contains the average classification accuracy of 10 experiments and the highest classification accuracy among 10 experiments for different α values.

As Table 1 shows, when α equals 0.005, the average classification accuracy is 89.6% and the highest classification accuracy is 94.67%, which are higher than other α values. Hence, α value for actual student experiment is set to 0.005.

4.2. Actual student experiment

The proposed mechanism is implemented on a SCORM-compatible LMS (SCORM, 2004). The implemented LMS is built on a Windows XP environment and developed by JAVA language. Microsoft Access 2003 is adopted to be the sample database DB_{sample}. 117 elementary school students are collected and recorded the following items (Chiu, Chuang, Hsiao, & Yang, 2009): (1) the serial number of each record; (2) the student's identification; (3) the browsed learning unit; (4) the previous browsed unit; (5) the time at which the learning unit is browsed; (6) the IP address of the student; (7) the serial number of the e-learning course; and (8) the identification of the learning unit. By two-stage cluster analysis, 117 elementary school students are classified into three kinds of learning styles (Chiu et al., in press):

(1) *Dilatorily type*: The student belonging to this type takes more time to browse a learning unit than other students. They often review the same learning unit and skip learning units.
(2) *Transitory type*: The student spends the least amount time in browsing and has the least browsing depth. Browsing order is irregular.
(3) *Persistent type*: The browsing depth is the highest and browsing order is regular.

Table 1
The comparison of classification accuracy with different values of α.

α	0	0.005	0.01	0.05	0.1
Average classification accuracy (%)	87.723	89.6	87.732	87.865	87.866
The highest classification accuracy (%)	93.33	94.67	93.33	93.33	93.33

This experiment is divided into three phases to observe the classification accuracy:

(I) *Initial phase*: As the aforementioned description, to execute k-NN classification $k\_NN(L_{uc}, DB_{sample})$, the number of samples for each class in DB_{sample} must contain at least $\lceil \frac{k}{2} \rceil$ samples. If the number of samples for each class is less than $\lceil \frac{k}{2} \rceil$ samples, the learning style of a student is randomly determined, which is called the initial phase.
(II) *Developing phase*: When the number of samples for each class in DB_{sample} is more than $\lceil \frac{k}{2} \rceil$ samples, the enhanced k-NN classification $Enhanced\_k\mathrm{NN}(L_{uc}, T_{check}, DB_{sample})$ is executed to classify the learning style of a student, which is called the developing phase.
(III) *Maturing phase*: The proposed mechanism that combines the enhanced k-NN classification $Enhanced\_k\mathrm{NN}(L_{uc}, T_{check}, DB_{sample})$ with GA-based behavioral feature extraction $GA(DB_{sample}, BF)$ is executed to classify the learning style of a student, which is called the maturing phase.

For each phase, 50 students are randomly selected to be testing samples, and the remaining 67 students are put into the sample database DB_{sample}. For the three phases, the selected 50 testing samples are classified to observe (A) the comparison of classification accuracy, (B) the comparison of classification stability, and (C) the needed number of learning behavioral features.

(A) *The comparison of classification accuracy*. Fig. 3 and Table 2 depict the classification accuracy of the three phases, where the classification accuracy can be derived by Eq. (13). As Table 2 depicts, the highest accuracy in the maturing phase is 96%, which is higher than the 92% in the developing phase and 86% in the initial phase. Similarly, the average accuracy in the maturing phase is 87.4%, which is higher than 81.4% in the developing phase and 76.8% in the initial phase.
(B) *The comparison of classification stability*. The standard deviation σ that is used to observe the classification quality can be derived using

$$\sigma = \sqrt{\frac{\sum_{i=1}^{C_{total}} (CA_{k-\mathrm{NN}}(DB_{sample}, BF)_i - \mu)^2}{C_{total}}}, \tag{14}$$

where $\mu = \frac{\sum_{i=1}^{C_{total}} CA_{k-\mathrm{NN}}(DB_{sample}, BF)_i}{C_{total}}$. C_{total} refers to the total number of the classification times, where $C_{total} = 10$. $CA_{k\text{-}NN}(DB_{sample}, BF)_i$, $1 \leqslant i \leqslant 10$, is the ith classification accuracy. Table 3 shows the computation results. As Table 3 shows, the standard deviation in the maturing phase is 0.038, which is lower than 0.065 in the developing phase and 0.05 in the initial phase. Hence, the classification quality in the maturing phase is more stable than the initial and developing phase.
(C) *The comparison of the needed number of learning behavioral features*: Since the learning style of a student is randomly determined in the initial phase, learning behavioral features in the initial phase do not need to be compared with that of the developing phase and the maturing phase. Fig. 4 depicts the comparison of the number of learning behavioral features in the developing phase and the maturing phase, which refer to the results of 10 experiments. As Fig. 4 depicts, the developing phase considers eight learning behavioral features for each experiment and the maturing phase only considers two to five learning behavioral features. With the needed number of learning behavioral features reduces, the computation complexity also decreases.
As the experimental results present, employing GA not only reduces the needed number of learning behavioral features, but also promotes classification accuracy of the proposed mechanism.

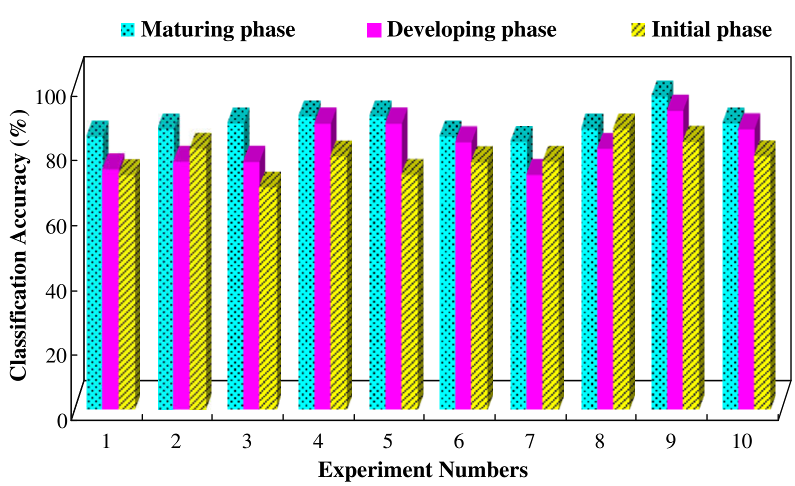

Fig. 3. The classification accuracy of the proposed mechanism.

Table 2
The comparison of classification accuracy with three phases.

Experiment numbers	Classification accuracy($CA_{k\text{-NN}}(DB_{sample}, BF)_i$, $1 \leqslant i \leqslant 10$)										Average
	1	2	3	4	5	6	7	8	9	10	
Maturing phase	84	86	88	90	90	84	82	86	96	88	87.4
Developing phase	74	76	76	88	88	82	72	80	92	86	81.4
Initial phase	72	80	68	78	72	76	76	86	82	78	76.8

Table 3
The comparison of the standard deviation with three phases.

Phase	Maturing	Developing	Initial
Standard deviation (σ)	0.038	0.065	0.05

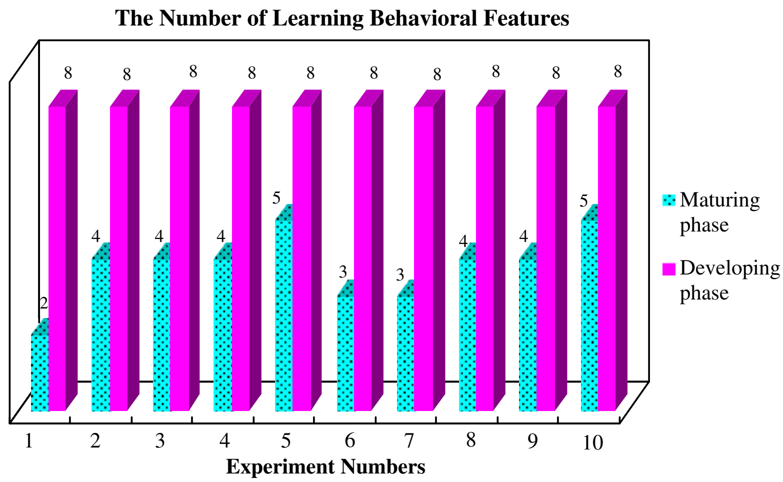

Fig. 4. The comparison of the number of learning behavioral features.

5. Conclusion

This paper has proposed the learning style classification mechanism that combines enhanced k-NN classification and GA, in which enhanced k-NN classification contains Pre-Contrast algorithm with Post-Comparison algorithm to improve the weaknesses of k-NN classification. In the proposed mechanism, while the student studies the adaptive learning material, the student's learning behavioral features are collected as a sample. Hence, the student's samples do not need to be collected in advance.

The proposed mechanism has been implemented on a SCORM-compatible LMS to demonstrate its viability. Based on the Iris dataset and the collected 117 students, the series of experiments have been done to verify the classification accuracy, classification stability and the needed number of learning behavioral features for the proposed mechanism. The experimental results indicate that the proposed mechanism can identify students' learning styles effectively.

Appendix. An illustrative example for the classification mechanism

In this sub-section, an illustrative example is presented to demonstrate how the proposed mechanism works. Given the number of learning styles is three, i.e., $M = 3$, which is represented as $C = \{c_1, c_2, c_3\}$. The number of behavioral features is four, i.e., $N = 4$, which is represented as $BF = \{bf_1, bf_2, bf_3, bf_4\}$. The sample database DB_{sample} contains 17 samples, i.e., $Z = 17$, which is listed in Table 4.

The error check table T_{check} contains three records, i.e., $Y = 3$, which is listed in Table 5. Suppose that the behavioral features of an unclassified student L_{uc} are represented as [5.5,3.5,1.3,0.2]. We take 5-NN for example, i.e., $k = 5$. The executing procedures of the learning style classification mechanism are presented as follows:

Step 1 Enhanced_kNN(L_{uc}, T_{check}, DB_{sample}) is executed to determine the class ranks for the unclassified student L_{uc}. As Algorithm_3 depicts, Pre-Contrast algorithm contrasts the behavioral features [5.5,3.5,1.3,0.2] of the unclassified student L_{uc} with the records in the error check table T_{check} in Table 5. The contrasted results indicate that T_{check} in Table 5 does not contain the matched record

Table 4
An illustrative example for the sample database DB_{sample}.

L_{s_z}	$bf_1(L_{s\_z})$	$bf_2(L_{s\_z})$	$bf_3(L_{s\_z})$	$bf_4(L_{s\_z})$	C_{s_z}
L_{s_1}	5.7	3.8	1.7	0.3	c_1
L_{s_2}	5.8	2.7	5.1	1.9	c_3
L_{s_3}	4.9	2.4	3.3	1	c_2
...					
L_{s_14}	5	2	3.5	1	c_2
L_{s_15}	5.8	2.8	5.1	2.4	c_3
L_{s_16}	5.8	4	1.2	0.2	c_1
L_{s_17}	5.9	3	5.1	1.8	c_3

Table 5
An illustrative example for the error check table T_{check}.

L_{ch_y}	$bf_1(L_{ch\_y})$	$bf_2(L_{ch\_y})$	$bf_3(L_{ch\_y})$	$bf_4(L_{ch\_y})$	C_{error_y}	C_{ch_y}
L_{ch_1}	5.8	2.8	5.1	2.4	c_1, c_3, c_2	c_3, c_1, c_2
L_{ch_2}	5.8	4	1.2	0.2	c_3, c_2, c_1	c_1, c_2, c_3
L_{ch_3}	5.9	3	5.1	1.8	c_1, c_3, c_2	c_3, c_1, c_2

for L_{uc}. Suppose that the number of samples for each class in the sample database DB_{sample} in Table 4 is greater than $\lceil \frac{k}{2} \rceil = \lceil \frac{5}{2} \rceil = 3$. Hence, $k\_NN(L_{uc}, DB_{sample})$ is executed to find the class ranks for the unclassified student L_{uc}. According to Eq.(4), the distance $d(L_{uc}, L_{s\_1})$ between the unclassified student L_{uc} and the sample L_{s_1} in Table 4 can be derived using

$$d(L_{uc}, L_{s\_1}) = \sqrt{\sum_{n=1}^{4} (bf_n(L_{uc}) - bf_n(L_{s\_1}))^2} = \sqrt{(5.5-5.7)^2 + (3.5-3.8)^2 + (1.3-1.7)^2 + (0.2-0.3)^2} \cong 0.55 \tag{15}$$

Similarly, the distance between the unclassified student L_{uc} and the samples in the sample database DB_{sample} can be derived in Table 6. Suppose that $k = 5$ closest distances are $L_{s\_1}, L_{s\_3}, L_{s\_14}, L_{s\_16}$ and L_{s_17}, where L_{s_1} and L_{s_16} belong to c_1, L_{s_3} and L_{s_14} belong to c_2, and L_{s_17} belongs to c_3. Hence, for the unclassified student L_{uc}, the executing results of 5-NN classification is $S = \{(c_1, 2), (c_2, 2), (c_3, 1)\}$. Since the numbers of samples that, respectively, belongs to the class c_1 and c_2 are equal, $Post\_Compared(S)$ is executed to sort appropriate class ranks. According to Eq. (7), the average distance D_{aver_c1} between the unclassified student L_{uc} and the samples belonging to the class c_1 can be derived as:

$$\begin{aligned} D_{aver\_c_1} &= \frac{\sum_{z=1}^{17} [g(c_1, f(L_{s\_z})) \times d(L_{uc\_1}, L_{s\_z})]}{\sum_{z=1}^{17} g(c_1, f(L_{s\_z}))} \\ &= \frac{1 \times 0.55 + 0 \times 4.25 + 0 \times 2.49 + \cdots + 0 \times 2.82 + 0 \times 4.46 + 1 \times 0.59 + 0 \times 4.17}{1+0+0+\cdots+0+0+1+0} \cong 0.57 \end{aligned} \tag{16}$$

Suppose that the average distance D_{aver_c1} between L_{uc} and the samples belonging to the class c_2 is 2.66. Since $D_{aver\_c1} = 0.57$ is less than $D_{aver\_c2} = 2.66$. The class ranks for L_{uc} are listed as $\{c_1, c_2, c_3\}$, i.e., the priority order for L_{uc} is $c_1 > c_2 > c_3$.

Step 2 Since the first priority order in the class ranks for L_{uc} is the class c_1, a learning material a_1 that conforms to c_1 is assigned to L_{uc}.

Step 3 After L_{uc} has read the adaptive learning material a_1 completely, the questionnaire asks L_{uc} whether a_1 is suitable for her/him.

Step 4 If the questionnaire result is positive, the behavioral features for L_{uc} and the class c_1 are added into DB_{sample} as the sample L_{s_18} in Table 7. Besides, the behavioral features and the class ranks $\{c_1, c_2, c_3\}$ are recorded into the error check table T_{check} as the record L_{ch_4} in Table 8.
Otherwise, the behavioral features and the class ranks for L_{uc} are recorded into T_{check} for a record that has been classified into incorrect class, as the record L_{ch_4} in Table 9. For the following steps, this step is supposed that the questionnaire result is positive, i.e., the adaptive learning material a_1 is suitable for L_{uc}.

Step 5 To demonstrate the execution of GA-based behavioral feature extraction $GA(DB_{sample}, BF)$, this step is supposed that the classification accuracy $CA_{k\text{-}NN}(DB_{sample}, BF)_T$ is lower than the classification accuracy in the last time $CA_{k\text{-}NN}(DB_{sample}, BF)_{T-1}$. Hence, $GA_{k\text{-}NN}(DB_{sample}, BF)$ is executed as the following sub-steps:

Table 6
An illustrative example for the computed results of the distance.

L_{s_z}	$bf_1(Ls\_z)$	$bf_2(L_{s\_z})$	$bf_3(L_{s\_z})$	$bf_4(L_{s\_z})$	C_{s_z}	$d(L_{uc}, L_{s\_z})$
L_{s_1}	5.7	3.8	1.7	0.3	c_1	0.55
L_{s_2}	5.8	2.7	5.1	1.9	c_3	4.25
L_{s_3}	4.9	2.4	3.3	1	c_2	2.49
...						
L_{s_14}	5	2	3.5	1	c_2	2.82
L_{s_15}	5.8	2.8	5.1	2.4	c_3	4.46
L_{s_16}	5.8	4	1.2	0.2	c_1	0.59
L_{s_17}	5.9	3	5.1	1.8	c_3	4.17

Table 7
L_{uc} is added to the sample database DB_{sample}.

L_{s_z}	$bf_1(L_{s\_z})$	$bf_2(L_{s\_z})$	$bf_3(L_{s\_y})$	$bf_4(L_{s\_z})$	C_{s_z}
L_{s_1}	5.7	3.8	1.7	0.3	c_1
L_{s_2}	5.8	2.7	5.1	1.9	c_3
L_{s_3}	4.9	2.4	3.3	1	c_2
...					
L_{s_14}	5	2	3.5	1	c_2
L_{s_15}	5.8	2.8	5.1	2.4	c_3
L_{s_16}	5.8	4	1.2	0.2	c_1
L_{s_17}	5.9	3	5.1	1.8	c_3
L_{s_18}	5.5	3.5	1.3	0.2	c_1

Table 8
L_{uc} is added to the error check table T_{check} for a correct classified record.

L_{ch_y}	$bf_1(L_{ch\_y})$	$bf_2(L_{ch\_y})$	$bf_3(L_{ch\_y})$	$bf_4(L_{ch\_y})$	C_{error_y}	C_{ch_y}
L_{ch_1}	5.8	2.8	5.1	2.4	c_1, c_3, c_2	c_3, c_1, c_2
L_{ch_2}	5.8	4	1.2	0.2	c_3, c_2, c_1	c_1, c_2, c_3
L_{ch_3}	5.9	3	5.1	1.8	c_1, c_3, c_2	c_3, c_1, c_2
L_{ch_4}	5.5	3.5	1.3	0.2		c_1, c_2, c_3

Table 9
L_{uc} is added to the error check table T_{check} for an incorrect classified record.

L_{ch_y}	$bf_1(L_{ch\_y})$	$bf_2(L_{ch\_y})$	$bf_3(L_{ch\_y})$	$bf_4(L_{ch\_y})$	C_{error_y}	C_{ch_y}
L_{ch_1}	5.8	2.8	5.1	2.4	c_1, c_3, c_2	c_3, c_1, c_2
L_{ch_2}	5.8	4	1.2	0.2	c_3, c_2, c_1	c_1, c_2, c_3
L_{ch_3}	5.9	3	5.1	1.8	c_1, c_3, c_2	c_3, c_1, c_2
L_{ch_4}	5.5	3.5	1.3	0.2	c_1, c_2, c_3	

Step G1. *Feature Value Normalizing*: As Eq. (8) depicts, the normalized value $bf_1(L_{s\_1})$ of the behavioral feature bf_1 for the sample L_{s_1} can be derived as:

$$bf_1(L_{s\_1})' = \frac{bf_1(L_{s\_1}) - \min_{i=1\sim18}(bf_1(L_{s\_i}))}{\max_{i=1\sim18}(bf_1(L_{s\_i}) - \min_{i=1\sim18}(bf_1(L_{s\_i}))} = \frac{5.7-4.9}{5.9-4.9} = 0.8 \tag{17}$$

where suppose that the minimum of the 1st learning behavioral feature bf_1 in DB_{sample} is 4.9 and the maximum of the 1st learning behavioral feature bf_1 in DB_{sample} is 5.9. Similarly, the normalized value of the samples in the sample database DB_{sample} can be obtained, which are listed in Table 10.

Step G2. *Training samples and testing samples divided*: the samples in DB_{sample} are randomly divided into the training sample set $db_{training}$ and the testing sample set $db_{testing}$. In $db_{training}$, the number of training samples for each class must be the same, where each class needs to contain $\lceil \frac{Z}{2M} \rceil = \lceil \frac{18}{2\times3} \rceil = 3$ training samples. Supposes that the training samples in the training sample set $db_{training}$ is randomly selected, which are $\{L_{s\_1}, L_{s\_2}, L_{s\_3}, L_{s\_11}, L_{s\_12}, \ldots, L_{s\_16}\}$, and the testing sample set $db_{testing}$ is listed, which are $\{L_{s\_4}, L_{s\_5}, \ldots, L_{s\_10}, L_{s\_17}, L_{s\_18}\}$.

Step G3. *Chromosome encoding and population initialization*: In this paper, a gene represents a learning behavioral feature, which is denoted as x_n that is a binary value. A chromosome is denoted as $\{x_1x_2x_3x_4\}$. If a chromosome contains the nth learning behavioral feature bf_n, $x_n = 1$. Otherwise, $x_n = 0$. For example: the 1st chromosome is represented as {1 0 1 0}, which means that the 1st chromosome contains the learning behavioral feature bf_1 and bf_3. Supposes that the number of chromosomes in a population is three, i.e., $Q = 3$. The initial population is randomly generated, which is listed in Table 11.

Step G4. *Fitness evaluation*: As Eq. (9) depicts, $CA_{1\text{-}NN}(db_{training}, BF)$ is defined, 1-NN classification classifies the testing samples $\{L_{s\_4}, L_{s\_5}, \ldots L_{s\_10}, L_{s\_17}, L_{s\_18}\}$ in the testing sample set $db_{testing}$. As Table 11 depicts, for the 1st chromosome {1 0 1 0}, the distance between the testing samples $\{L_{s\_4}, L_{s\_5}, \ldots L_{s\_10}, L_{s\_17}, L_{s\_18}\}$ and the training samples $\{L_{s\_1}, L_{s\_2}, L_{s\_3}, L_{s\_11}, L_{s\_12}, \ldots L_{s\_16}\}$ only needs to consider the behavioral features bf_1 and bf_3. Since the testing set $\{L_{s\_4}, L_{s\_5}, \ldots, L_{s\_10}, L_{s\_17}, L_{s\_18}\}$ contains nine testing samples, the total number of classification is nine, i.e., $C_{total} = 9$.
Suppose that there are two correct classifications, i.e., $C_{correct} = 2$, which contain L_{s_17} belonging to c_3 and L_{s_18} belonging to c_1, as Table 12 depicts. Supposes that coefficient α in Eq. (9) is 0.005. Hence, for the 1st chromosome, the fitness value can be derived as $f_1(BF) = CA_{1\text{-}NN}(db_{training}, BF) - \alpha \times card(BF) = \frac{2}{9} - 0.005 \times 2 \cong 0.21$. Suppose that the fitness value $f_2(BF)$ and $f_3(BF)$ for the 2nd and 3rd chromosomes can be derived as Table 13 depicts.

Step G5. *Selection*: According to Eq. (11), the probability $Prob(f_1(BF))$ that the 1st chromosome is selected to propagate child chromosomes can be derived as:

$$Prob(f_1(BF)) = \frac{f_1(BF)}{\sum_{q=1}^{3} f_q(BF)} = \frac{0.21}{0.21+0.82+0.56} = 0.13. \tag{18}$$

Table 10
An illustrative example for normalizing the behavioral feature value of the samples.

L_{s_z}	$bf_1(L_{s\_z})'$	$bf_2(L_{s\_z})'$	$bf_3(L_{s\_y})'$	$bf_4(L_{s\_z})'$	C_{s_z}
L_{s_1}	0.80	0.90	0.13	0.05	c_1
L_{s_2}	0.90	0.35	1.00	0.77	c_3
L_{s_3}	0.00	0.20	0.54	0.36	c_2
...					
L_{s_14}	0.10	0.00	0.59	0.36	c_2
L_{s_15}	0.90	0.40	1.00	1.00	c_3
L_{s_16}	0.90	1.00	0.00	0.00	c_1
L_{s_17}	1.00	0.50	1.00	0.73	c_3
L_{s_18}	0.80	0.90	0.13	0.05	c_1

Table 11
An illustrative example for initial population.

Chromosomes	Initial population
1st	1010
2nd	0001
3rd	1100

Table 12
An illustrative example for the classified results of the 1st chromosome {1010}.

L_{s_z}	$bf_1(L_{s\_z})'$	$bf_3(L_{s\_z})'$	c_{s_z}	The 1st chromosome {1 0 1 0}		
				$d(\{L_{s\_4}, L_{s\_5}, \ldots, L_{s\_10}\}, L_{s\_i})$	$d(L_{s\_17}, L_{s\_i})$	$d(L_{s\_18}, L_{s\_i})$
L_{s_1}	0.80	0.13	c1	...	0.89	0.22
L_{s_2}	0.90	1.00	c3		0.10	1.02
L_{s_3}	0.00	0.54	c2		1.10	0.79
L_{s_11}				...		
...				...		
L_{s_14}	0.10	0.59	c2	...	0.99	0.75
L_{s_15}	0.90	1.00	c3		0.10	1.02
L_{s_16}	0.90	0.00	c1		1.00	0.30

Table 13
An illustrative example for the fitness value and the selected probability.

Chromosomes	Initial population		
	$\{x_1\ x_2\ x_3\ x_4\}$	Fitness value	$Prob(f_q(BF))$, $1 \leqslant q \leqslant 3$
1st	1 0 1 0	0.21	0.13
2nd	0 0 0 1	0.82	0.52
3rd	1 1 0 0	0.56	0.35

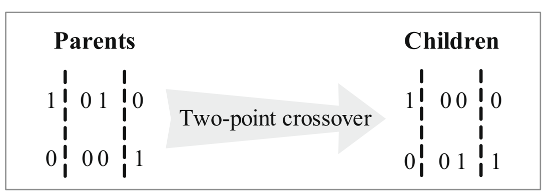

Fig. 5. An illustrative example of crossover execution for the 1st and 2nd chromosomes.

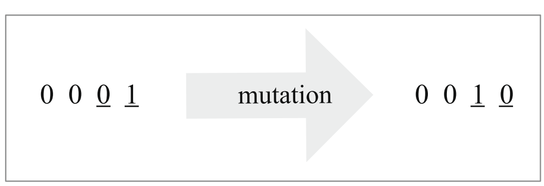

Fig. 6. An illustrative example of mutation execution for the 2nd chromosome.

Similarly, the probability $Prob(f_2(BF))$ and $Prob(f_3(BF))$ for the 2nd and 3rd chromosomes can be derived as Table 13 depicts. Hence, the 2nd chromosome has higher probability than the 1st and 3rd chromosomes to be selected to propagate offspring.

Step G6. *Crossover*: Suppose that the 1st and 2nd chromosomes are randomly selected to execute crossover. Fig. 5 depicts the crossover execution.

Step G7. *Mutation*: Suppose that the 2nd chromosome is randomly selected to execute mutation. Fig. 6 depicts the mutation execution, where the mutated genes are randomly selected and underlined in Fig. 6. Table 14 depicts the 1st generation after the crossover and the mutation execution.

Table 14
an illustrative example for the 1st generation.

Chromosomes	Initial population	1st generation
1st	1 0 1 0	1 0 0 0
2nd	0 0 0 1	0 0 1 0
3rd	1 1 0 0	0 0 1 1

Step G8. *Stop Criterion*: After 100 generations, the appropriate solution that refers to the extracted behavioral features can be obtained. By considering the extracted behavioral features, the proposed mechanism classifies the next unclassified student.

After $GA(DB_{sample}, BF)$ executed, the records in the error check table T_{check} are deleted.

References

Allen, I. E., & Seaman, J. (2003). *Sizing the opportunity: The quality and extent of online education in the United States, 2002 and 2003*. The Sloan Consortium, Technology Report.

Brusilovsky, P., Eklund, J., & Schwarz, E. (1998). Web-based education for all: A tool for development adaptive courseware. *Computer Networks and ISDN Systems, 30*, 291–300.

Brusilovsky, P. (1999). Adaptive and intelligent technologies for web-based education. *KI – Kunstliche Intelligenz, 13*, 19–25.

Brusilovsky, P., & Maybury, M. T. (2002). From adaptive hypermedia to the adaptive web. *Communications of the ACM, 45*(5), 30–33.

Carver, C. A., Howard, R. A., & Lane, W. D. (1999). Addressing different learning styles through course hypermedia. *IEEE Transactions on Education, 42*(1), 33–38.

Chen, C. M., Lee, H. M., & Chen, Y. H. (2005). Personalized e-learning system using Item Response Theory. *Computers and Education, 44*(3), 237–255.

Chen, J. H., Yen, Z. H., & Ho, S. Y. (2004). *Design of optimal nearest neighbor classifier using an intelligent multi-objective evolutionary algorithm*. *Lecture Notes in Computer Science (LNCS)* (Vol. 3157, pp. 262–271). Springer.

Chiu, C. H., Chuang, C. H., Hsiao, H. F., & Yang, H. Y. (in press). Exploring the patterns of computer mediated synchronous collaboration by elementary school students. Computers in Human Behavior.

Dunn, R., Dunn, K., & Price, G. E. (1984). *Productivity environmental preference survey*. Lawrence, KS: Price Systems.

Felder, R., & Silverman, L. (1988). Learning and teaching styles. *Journal of Engineering Education, 78*(7), 674–681.

García, P., Amandi, A., Schiaffino, S., & Campo, M. (2007). Evaluating Bayesian Networks' precision for detecting students' learning styles. *Computers and Education, 49*(3), 794–808.

Gerald, D. E., & Hussar, W. J. (2003). *Projections of education statistics to 2013*. National Center for Education Statistics, Technology Report NCES 2004-013, Washington, DC.

Graf, S., & Kinshuk, P. (2006). An approach for detecting learning styles in Learning Management Systems. In *Proceedings of the sixth international conference on advanced learning technologies (ICALT'06)* (pp. 161–163).

Gregorc, A. F., & Ward, H. B. (1977). Implications for learning and teaching: A new definition for individual. *NASSP Bulletin, 61*(406), 20–26.

Holland, J. H. (1975). *Adaptation in natural and artificial system*. Ann Arbor: The University of Michigan Press.

Huang, J., Wang, Y., Tan, T., & Cui, J. (2004). A new Iris segmentation method for recognition. In *Proceedings of the 17th international conference on pattern recognition (ICPR 2004)* (Vol. 3, pp. 554–557).

Hwang, G. J., Yin, P. Y., Wang, T. T., Tseng, C. R., & Hwang, G. H. (2008). An enhanced genetic approach to optimizing auto-reply accuracy of an e-learning system. *Computers and Education, 51*(1), 337–353.

Jain, A. K., Bolle, R., & Pankanti, S. (1999). *Biometrics: Personal identification in a Networked Society*. MA: Kluwer, Norwell.

Keefe, J. W. (1987). *Learning styles: Theory and practice*. Reston, VA: National Association of Secondary School Principals.

Kolb, D. A. (1984). *Learning style inventory*. Boston: McBerr.

Kuncheva, L. I. (1995). Editing for the k-nearest neighbors rule by a genetic algorithm. *Pattern Recognition Letter, 16*(8), 809–814.

Kuncheva, L. I. (1997). Fitness functions in editing k-NN reference set by genetic algorithms. *Pattern Recognition, 30*(6), 1041–1049.

Kuncheva, L. I., & Jain, L. C. (1999). Nearest neighbor classifier: Simultaneous editing and feature selection. *Pattern Recognition Letters, 20*(11–13), 1149–1156.

Mitchell, T. (1997). *Machine learning*. McGraw-Hill.

Pena, C., Narzo, J., & Rosa, J. (2002). Intelligent agents in a teaching and learning environment on the Web. In *Proceedings of the 2nd IEEE international conference on advanced learning technologies (ICALT2002)* (pp. 9–12).

Rothlauf, F. (2006). *Representations for genetic and evolutionary algorithms*. Springer Verlag.

Schiaffino, S., Garcia, P., & Amandi, A. (2008). ETeacher: Providing personalized assistance to e-learning students. *Computers and Education, 51*(4), 1744–1754.

Sessink, O., Beeftink, R., Tramper, J., & Hartog, R. (2003). Author-Defined storage in the next generation Learning Management Systems. In *Proceedings of the 3rd IEEE international conference on advanced learning technologies (ICALT'03)* (pp. 57–61).

Shang, Y., Shi, H. C., & Chen, S. S. (2001). An intelligent distributed environment for active learning. In *Proceedings of the 10th international conference on World Wide Web* (pp. 308–315).

Shareable Content Object Reference Model (SCORM). (2004). <http://www.adlnet.org/>.

Stangl, W. (2002). Der HALB-Test [The HALB test]. <http://arbeitsblaetter.stangl-taller.at/TEST/HALB/>.

Syswerda, G. (1989). Uniform crossover in genetic algorithms. In *Proceedings of the third international conference on genetic algorithms and their applications* (pp. 2–9).

Trantafillou, E., Poportsis, A., & Demetriadis, S. (2003). The design and the formative evaluation of an adaptive educational system based on cognitive. *Computer and Education, 41*(1), 87–103.

Tseng, C. R., Chu, H. C., Hwang, G. J., & Tsai, C. C. (2008). Development of an adaptive learning system with two sources of personalization information. *Computers and Education, 51*(2), 776–786.

Waits, T., & Lewis, L. (2003). *Distance education at degree-granting postsecondary institutions 2001–2002*. National Center for Education Statistics, Washington, DC, Technology Report NCES 2003-017.

Wang, D. (2008). Fast Constructive-Covering Algorithm for neural networks and its implement in classification. *Applied Soft Computing, 8*(1), 166–173.

Wirt, J., Choy, S., Rooney, P., Provasnik, S., & Tobin, R. (2004). *The condition of education 2004*. National Center for Education Statistics, Washington, DC, Technology Report NCES 2004-077.

Xenos, M. (2004). Prediction and assessment of student behaviour in open and distance education in computers using Bayesian networks. *Journal of Computers and Education, 43*(4), 345–359.

Yao, X. (1999). Evolving artificial neural networks. *Proceedings of the IEEE, 87*(9), 1423–1447.

Yi-Chun Chang received the B.S. degree in Electronic Engineering from Minghsin University of Science and Technology, Hsinchu, Taiwan, R.O.C., in 2000 and the M.S. degree in Computer Science and Information and Engineering from Chaoyang University of Technology, Taichung, Taiwan, R.O.C., in 2004. She is currently a Ph.D. candidate in Department of Computer Science and Information and Engineering, National Cheng Kung University, Tainan, Taiwan, R.O.C. Her research interests include e-learning, personalized learning, applied intelligence, evolutionary algorithms and internet computing.

Wen-Yen Kao received the B.S. degree in Computer Science Engineering from Aletheia University of Science and Technology, Taipei, Taiwan, R.O.C., in 2003 and the M.S. degree in Computer Science and Information and Engineering from National Cheng Kung University, Tainan, Taiwan, R.O.C., in 2005. He is currently an engineer at Avocent Corporation. His research interests include e-learning, genetic algorithm.

Chih-Ping Chu received the B.S. degree in agricultural chemistry from National Chung Hsing University, Taiwan, the M.S. degree in computer science from the University of California, Riverside, and the Ph.D. degree in computer science from Louisiana State University. He is currently a professor in the Department of Computer Science and Information Engineering of National Cheng Kung University, Taiwan, R.O.C. His research interests include high-performance computing, parallel processing, internet computing, e-learning, and software engineering.

Chiung-Hui Chiu received the B.S. degree in Physics from National Taiwan Normal University (NTNU) in Taiwan in 1988, and the M.E. and Ph.D. degrees in Science Education with emphasis in Computer Science from the University of Texas at Austin in 1993 and 1996. She is a Professor of the Graduate Institute of Information and Computer Education at NTNU. She taught at the Department of Information and Learning Technology at the National University of Tainan in Taiwan before coming to NTNU in 2007. Her research interests are in the areas of computer supported collaborative learning, learning technology, and computer science education.