An Intelligent Teaching System Model Based on Web Log Data Mining

Qian Wang

Shanxi Professional College of Finance, Taiyuan 030000, Shanxi, China

Abstract

Based on a fully analysis of the basic principles, characteristics and related models of intelligent teaching system, this paper applies data mining technology to the system, puts forward a research idea of intelligent teaching system based on web and data mining and presents an overall design and implementation scheme of the system.

It focuses on the application of sorting algorithm and its expandability in data mining, employs ID3 algorithm in classification trees and expandability methods which integrate data mining and DBMS closely and classifies students by teaching difficulty successfully. It allows the system to adapt to efficient information mining based on a large number of data. Meanwhile, such kind of research and application makes a new attempt to apply data mining ideas to intelligent teaching system and improve the performance of system.

To study inference mechanisms which demonstrate system intelligence, this paper sets up a productive system of confirmatory inference, adopts an inference control strategy combining forward and backward inference in teaching strategy control and achieves individualized teaching processes dominated by teachers and centered on students.

Key words: INTELLIGENT TEACHING SYSTEM, DATA MINING, COGNITIVE STUDENT MODEL, INDIVIDUALIZED TEACHING, PRODUCTIVE SYSTEM

1. Introduction

With the continuous development and maturity of web technology, web-based intelligent teaching system has received more and more attention. The application of data mining, artificial intelligence and other technology has become more and more widespread and in-depth. Research on intelligent teaching system based on web and data mining will inevitably promote the continuous development and progress of intelligent teaching and related disciplines. It has profound theoretical and practical significance.

Guided by constructivist learning theory, with “artificial intelligence” course teaching as the object, this paper uses dynamic HTML, JSP and other latest web technology, to build an intelligent distance teaching platform. The system is suitable for interdisciplinary pedagogy. It can achieve individualized teaching, present suggestions and guidance for students’ learning and reflect their cognitive abilities. It is a universal intelligent distance teaching platform. The system has been debugged in the campus network and can run efficiently, stably and reliably. Practice has proved that the present study has certain theoretical and practical application value for the intelligent distance teaching system.

The greatest benefit for introducing data mining technology to an educational system is that it can analyze a large number of data gathered in the system, mine useful information for the presentation of course contents and the adjustment of teaching strategies, thus building an informative, easy-to-operate, distinctive and intelligent teaching platform.

2. An Overall Model of the System

Teaching systems based on data mining are developed with a browser/server (B/S) mode. The B/S mode is widely used as a mainstream of current software development. They emphasize servers and ignore browsers [1]. As a result, most work can be done at the server side. The client side only installs browser software. This model is featured with no installation, easy-to-use, simple maintenance, cross-platform and easy to upgrade, etc. Aimed at these advantages of the B/S mode, associating with the actual needs of teaching systems, we build a model of the system based on the B/S mode.

A 3-layer structure of the system includes a user interface layer, a web server layer and a resource layer [2]. When student users enter the system, first of all, their identity will be authenticated. Scheduling modules extract teaching contents customized to their characteristics from the field knowledge base by mining and analyzing data gathered by the system, and adjust these contents dynamically, combining with the users' learning path history [3]. When users use this teaching system, their learning behaviors, such as learning contents, preference for resource types, stay time in course resources, accomplishment and quality of homework, test results, time and frequency of use, etc. will be recorded automatically [4]. If users are freshmen without any learning history, a sorting algorithm for data mining will be used to classify them as one type based on their features and provide teaching contents based on their learning features. The system can adjust teaching contents dynamically by analyzing students' learning process, homework and test results.

If users are teachers, the system will mine and analyze data gathered from their students and provide them with students' learning status and on-line teaching tips [5]. Teachers can adjust the difficulty of teaching contents and the allocation of various types of teaching resources. They can also communicate with some particular students in a variety of forms, inspire their learning enthusiasm and adjust their learning status in a timely manner.

Knowledge base: The sources of teaching contents are mainly various types of teaching resources elaborately organized by teachers, in accordance with teaching needs, including documents, electronic slides, teaching videos and small demo software, etc. They are a cornerstone of teaching systems [6]. Whether they are good or bad have a direct impact on students' learning effect and initiative.

Teaching strategy base: It is a guarantee for systems to carry out teaching activities. This base is mainly organized by teachers according to teaching experience, based on the logical relationships between all knowledge points, well-designed teaching sequences and applicable methods in the form of 2D matrixes [7]. After each student or each group of students has learned for a period of time, a dynamically adjustable teaching strategy matrix will be saved according to their characteristics.

Students' information bank: The students' information bank stores all students' personal profiles and learning history, such as their learning contents, preference for resource types, accomplishment and quality of homework, test results and time and frequency of use, etc. It provides a guarantee for data mining modules to analyze students.

Question bank: It stores questions about various knowledge points and can be used to assign homework or tests [8]. The questions are designed by teachers elaborately and divided into different levels of difficulty. The system can adjust questions automatically, according to the situations of students [9]. Teachers can also adjust questions manually in teachers' module and discriminate different students, lest poor homework or tests dampen their learning enthusiasm.

Data warehouse: The data warehouse stores various data from other databases through ETL, as well as useful features in the "feature extraction" part.

Data mining: It is an intelligent core of any intelligent teaching system based on data mining [10]. It is composed of two sub-modules: correlation and sorting. It analyzes key features extracted by the "feature extraction" part through correlation rules, sorting and other data mining methods and finds out the temporal and spatial relationship between characteristics.

2.1. Process Design of the System

After learners enter the system, they need to register first, fill in their personal information, such as age, gender, educational background and major, etc. and a personal cognitive assessment form [11]. After login, learners can choose majors and subjects. The system will choose suitable teaching strategies from a knowledge base, according to students' learning history and cognitive abilities. It controls their learning progress and learning contents. Students enter a navigation interface and study. In the process of learning, students can select presentation media for teaching contents and sample questions to assist learning [12]. Knowledge points in each chapter are tested, to check students' learning effect. Teaching progress and strategies are adjusted according to students' learning effect. Meanwhile, the results of knowledge point tests in each chapter will be used as materials of data mining in teaching strategy. The results of data mining are some rules about the teaching strategy [13], such as Teaching Strategy X. The result of mining is that if students have a high cognitive ability, then the strategy will be adopted. Any mistakes that students make in knowledge tests in each chapter will be recorded in the history bank, as the source of data mining, to generate a dependency between knowledge points, i.e., learning priority. Students are also allowed to view answers, grades and rankings. Their learning motivation will be inspired (Fig. 1)

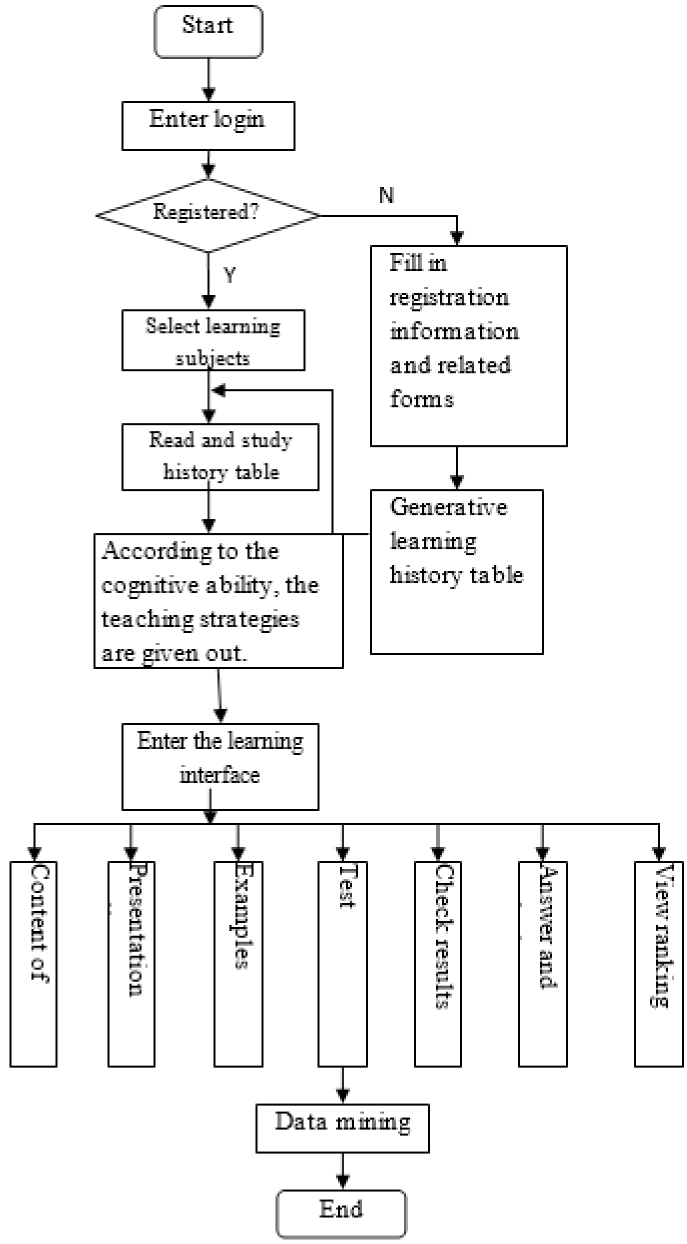

Figure 1. Process design of teaching system model based on Data Mining

2.2. The Design of a Knowledge Base

A knowledge base includes field expert model(s). The field expert model in this system studies the model design between different courses and the model expression of the relationship between knowledge points.

2.2.1. A Design Based on the Relationship between Professional Courses

Learning is a gradual process. Each profession that we learn is composed of several professional courses [14]. Each course touches upon one direction of a profession and knowledge points in this direction. All courses combine organically and form a profession. From the perspective of professions, some courses are fundamental subjects, for example, “advanced math”. They are independent of other courses, while others cannot begin until basic elective courses are completed. For example, software students usually learn “data structure” after “program design fundamentals” and “discrete math” [15]. This kind of priority can be represented with a directed acyclic graph (DAG), i.e. AOV web, as shown in Tables 1 and Fig. 2. The apex of the DAG stands for a course. The arc stands for a prerequisite. If Couse i is an prerequisite for Couse j, then Arc <i, j> exists.

Table 1. Required Courses for Software Majors

Course Codes	Course Names	Prerequisites
C1	Program Design Fundamentals	No
C2	Discrete Math	C1
C3	Data Structure	C1,C2
C4	Assembly Language	C1
C5	Design and Analysis of Language Programs	C3,C4
C6	Computer Principles	C11
C7	Compiling Principles	C5,C3
C8	Operating System	C3,C6
C9	Advanced Math	No
C10	Linear Algebra	C9
C11	General Physics	C9
C12	Numerical Value Analysis	C9,C10,C1

A DAG for course priority aimed at a certain profession exists in the field knowledge base. When students log in, this DAG is copied to their history bank, according to the professions they selected. First of all, the system recommends subjects with 0 degree of admission to students, according to the DAG. If students haven't learned this subject, they will be able to learn it. If they have, they will be tested. If they pass the test, a tail arc of this subject will be deleted. Students can enter any subject with 0 degree of admission and learn. If professional learning is completed, the degree of admission for all nodes in this DAG must be 0. When the degree of admission for all nodes in this DAG is 0, students can learn any subject at will. It can be judged from test results whether students have completed professional learning. There are very similar knowledge point DAGs in students' history bank. Of course, they also come from the expert model.

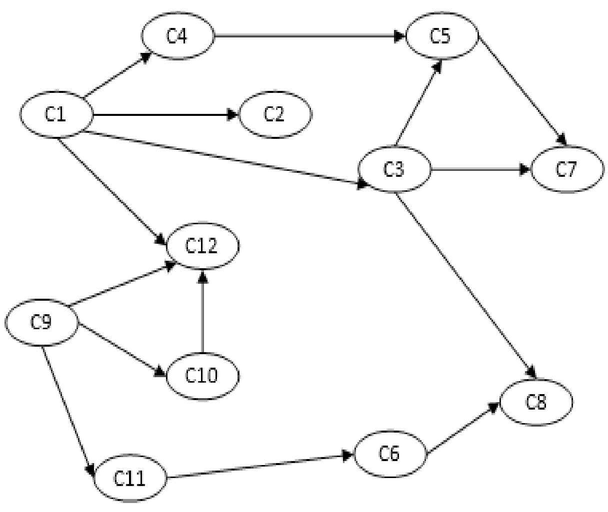

Figure 2. A DAG for Course Priority

3. Algorithm Design

When we finish learning a chapter or a unit, there will be specific tests for the knowledge points in this chapter. When students learn by the priority of dependency in the DAG, if preceding language points are not mastered, it is very likely that subsequent language points are not mastered, either. That is to say, in the test, if a preceding Knowledge Point A is wrong, then it is very likely that a subsequent Knowledge Point B is wrong, too. We take advantage of the necessary correlation between such a priority of mistakes to judge and adjust AOV webs between knowledge points in the expert model and arrive at the following methods for judgment: if Knowledge Point A is wrong, it is very likely that Knowledge Point B is wrong, too. But it is unlikely that Knowledge Points A and B are simultaneously wrong, or it can be inferred from the difficulty of knowledge points that A and B are not simultaneously difficult. It can be concluded that Knowledge Point A is preceding to Knowledge Point B. If Knowledge Points A and B don't have any directed arcs in the expert model, then add this arc to show that the path from A to B is consistent with students' learning and cognitive process. On the contrary, if Knowledge Points A and B already have directed arcs, then whether Knowledge Point A is wrong or not has little impact on Knowledge Point B. It is believed that there is a small dependency between Knowledge Points A and B. The arc between A and B can be deleted. It doesn't matter whether A or B is learned first. Compared with the original DAG, if there is any change, the subsequent DAG will present a more consistent knowledge point dependency with students' learning characteristics and improve teaching sequence. Of course, we should judge whether there is any loop in the subsequent DAG, through topology sequencing. If so, delete the generated DAG. Based on the above analysis, we can select correlation rules for data mining and find out correlations between knowledge points. The basic model of correlation rules is as follows:

Correlation rules: Assume that $I = \{i_1, i_2, \cdots, i_n\}$ is a set of items. D is a set of database transactions. Among them, each transaction T is a set of items. $T \subseteq I$. Assume that A is an item set. Transaction T contains A when and only when $A \subseteq T$. Correlation rules are an implication of $A \Rightarrow B$, where $A \subseteq I$, $B \subseteq I$ and $A \cap B = \varnothing$. Support: transactions in D contain percentages of $A \cup B$, that is, $\text{support}(A \Rightarrow B) = P(A \cup B)$. Confidence: D contains transactions in A and percentages in B, that is, $\text{confidence}(A \Rightarrow B) = P(B|A)$.

Frequent item set: When the support of an item set is greater than or equal to a given minimum support, then the set is called a frequent item set.

Strong rules: Rules that satisfy the threshold of minimum support and the threshold of minimum confidence simultaneously are called strong rules.

The source of data is a list of wrong knowledge points in all students' information base, who complete the course. According to learning chapters and sequence, they are organized into the data source of mining. This paper uses algorithm analogic to AprioriAll to generate frequent item sets.

First of all, aimed at a list of wrong knowledge points of a certain course, a transaction sequence is formed, according to students' learning sequence. All sequences combine and form a sequence for mining.

(2) Items in Candidate Set C1 are sequenced by the topology of the original DAG. A frequent item set L1 is derived from the threshold of support;

(3) The loop generates Ck from the complete connection of item sets in LK-1, until the item sets in LK-1 is null.

(4) Some rules are derived from each LK-1. Among them, some rules are coincident with the original DAG and should be supported. Judge non-coincident parts. If they can satisfy certain requirements, then the DAG is improved and enriched. The algorithm is as follows:

An algorithm analogic to AprioriAll:
Input:
D//transaction sequence base
S//minimum support
Output: sequence mode
Items in C1 are sequenced by the topology of original DAG.
Find L1 in D；
Ln=AprioriAll(D ,S ,L1)
Find maximal reference sequences from Ln
AprioriAll(D,S,L1)
Find maximal refrence sequences from Ln
N=2
Do
For i=1 to length(Ln-1)
For j=2 to length(Ln-1)
If i<>j then Append(ln-1,i(+)ln-1,j)to Cn//(+) stands for perfect crossing.
Next j
Next i
For each c in C
If Supprort(c)inD>=S then Append c to Ln
Next
n=n+1
While Cn-1<>null

The difference between this algorithm and AprioriAll is that the item sequence in C1 is arranged by knowledge points learned by students in the original DAG. Students using the same knowledge point sequence are regarded as the object of mining. The purpose of sequencing is to describe the shadow of the original DAG. Below, a set of instances will be given(Table 2).

First of all, sequence wrong knowledge points, according to a list of students' mistakes in knowledge points. Assume that a DAG for knowledge point dependency is shown in Figure 2. Assume that the support is 50% and confidence is 75%.

Analysis and application of mining results. According to L1, it can be concluded that knowledge points A, B, C and D are difficult to grasp. They need be improved in teaching strategies (e.g. to increase learning time and enrich knowledge contents, etc.)

The correlation between unexpected knowledge points derived from L2 is as follows:

Confidence = support ({B,C})/support ({B})=1

Confidence = support ({B,C})/support({C})=50%

According to the dependency in the original DAG, an unexpected discovery of this mining is that there may be a dependency between B and C. If there is an easy knowledge point between B and C or the support and confidence of mistakes in C and B don't meet the requirements of closed values, then it is believed that there is a dependency between B and C. Learning from B to C in the topology sequence of learning path will achieve better learning effect.

From L3, we can derive a sequence with the most mistakes in knowledge points. Knowledge point combinations in this sequence can be used to improve the difficulty of questions.

Table 2. Sequence for Mining, Candidate Item Sets and Frequent Item Sets

Transaction Sequences	Item Sets
001	ACD
002	BCD
003	ABCD
004	CE

C1: (When K=1, items are sequenced according to a list of students' knowledge points in the original DAG).

Item Sets	Support
A	50%
B	75%
C	100%
D	75%
C2:	
Item Sets	Support
{AB}	25%
{AC}	50%
{AD}	50%
{BC}	50%
{BD}	50%
{CD}	75%
{BA}	0
{CA}	0
{DA}	0
{CB}	0
{DB}	0
{DC}	0
L2:	
Frequent Item Sets	Support
{AC}	50%
{AD}	50%
{BC}	50%
{BD}	50%
{CD}	75%
C3:	
Item Sets	Support
{ACD}	50%
{BCD}	50%
L3:	
Frequent Item Set	Support
{ACD}	50%
{BCD}	50%

Improve the original DAG of knowledge points according to mining results:

(1) Add a directed arc: If two points in the original DAGs don't have direct and indirect dependency or correlation, then the correlation and confidence of mining results meet requirements. If it is judged that there is a dependency between two knowledge points, then add a directed arc between two nodes.

(2) Maintain the original arc. If mining results are not unexpected, for example, AC, BD, CD and AD, etc. in L: reflect the dependency of this DAG, then maintain them. It is also proved that the dependency between these knowledge points is suitable for students to learn knowledge points. There is a strong correlation between B and C, so there is a directed arc between B and C, as shown in the figure below (Figure 3).

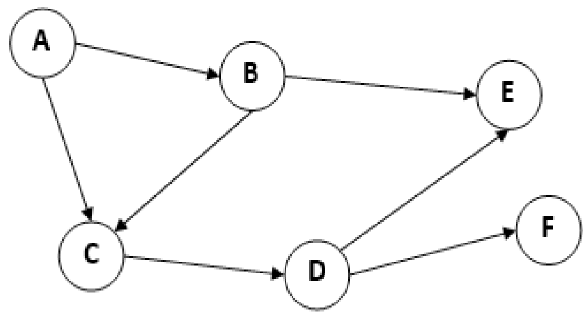

Figure 3. An Improved DAG of Knowledge Points

(3) How to judge whether a generated DAG has a loop: to sequence the topology of a generated DAG. If sequencing result is less than the number of knowledge points, then the DAG is believed to have a loop. Delete this DAG and maintain the original DAG. An analysis of this algorithm according to actual situations: data mined with this mining algorithm are all students completing this course, so each student won't make many mistakes in knowledge points. The mined data are more authoritative, because students can't complete this course until passing the exam. Therefore, databases won't be scanned for too many times.

Of course, this is just a learning sequence of knowledge points. To achieve better teaching effect, we should use different ways and means, i.e., teaching strategies, for each knowledge point, according to students' characteristics.

A Quantitative Assessment of Students' Cognitive Abilities: a Weighted Evaluation Algorithm. In an actual system, we assess students' cognitive abilities from three aspects, i.e., memorization, understanding and application. In order to assess their abilities objectively from three aspects, we employ learning tests. In these tests, corresponding weight levels are set for memorization, understanding and application in each question. These levels are given by teaching experts when designing test questions, according to knowledge points to be tested and form an ability weight matrix (i stands for times and j stands for ability weight. Here j is 3.) Meanwhile, the vector of test results T is composed of students' answers. If students are correct in this question, the corresponding vector is 1. If not, the vector is 0.

$$R=\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet \\ r_{i1} & r_{i2} & r_{i3} \end{bmatrix} T=(t_1,t_2,\cdots t_i)$$

The scores of various cognitive abilities are calculated by the following formula.

$$A_j=\frac{\sum_{i=1}^{\infty} t_i r_{ij}}{\sum_{i=1}^{\infty} r_{ij}} \quad (1)$$

The teachers' assessments for students are fuzzy. The assessment results are fuzzified. Students' cognitive ability values are fuzzified into 3 levels, i.e. P = {Low, Medium, High}. It is defined that $A_j \in [0,0.6)$ is low, $A_j \in [0.6,0.8)$ is medium and $A_j \in [0.8,1)$ is high.

For example, as defined by fuzzification, $A_j \in [0.8,1)$ means students have high memorization and understanding abilities, but medium application ability. Assume that a weight matrix of cognitive abilities is G=(0.6,0.35,0.05), then the overall cognitive abilities are ts=A.G=0.87*0.6+0.8*0.35+0.71*0.05 =0.83. As defined by fuzzification, this student has high overall cognitive abilities. When students have high overall cognitive abilities, they can enter the next chapter to learn.

The following part describes how to realize teaching strategies and mine rules, through a decision tree algorithm. An assessment is given for students' overall cognitive abilities. The assessment results are divided into excellent, good, medium, pass and fail. In order to set up examples and counter examples for training sets, we only select excellent and failed students. Excellent students are examples. Failed students are counter examples. We can set up a general description for Training Model <<S, V, C>, where S is a training set, v is a property set of training sets and c is the classification of training sets. In this system, the meanings of S, V and C are as follows:

1)S is a training set for students. Choose a feature (A) to represent their learning level. $|s|$ stands for the total number of examples.

2) There are 3 properties in S. V0, V1, V2 and V3 stand for teaching methods, memorization, understanding and application abilities in turn. V1, V2 and V3 have three values: high, medium, and low.

3) According to Feature A, S is divided into two categories: 2 values for excellent and fail, P and N represents examples and counter examples.

(1) The implementation of algorithm

We improve the traditional CLS decision tree learning algorithm and control the size of decision tree by pre-trimming. The improved decision tree learning algorithm is as follows: first of all, assess students' cognitive abilities quantitatively. The assessment method is shown in 5.2.1.

1) The initial state of Decision Tree T contains only one root (X, V), where X is a set of all training examples. V is a set of all test properties. V={memorization, understanding, application abilities};

2) If the probability that training examples in X,-belong to the same type (examples or counter examples) is more than 80%, then these training examples are regarded as belong to the same type.

3) If all leaf nodes in T (X,,V,) meet one of the following conditions, then stop running the learning algorithm. The result of learning is T;

1. V' is null.

2. Training examples in X' belong to the same type.

4) Or select a leaf node (X',V') that doesn't have the state mentioned in Step III;

5) For V', select the largest test attribute b of I (X'; b);

6) Assume that different values of Tested Property b are m disjoint subjects Xi, $1 \le i \le m$ stretches out m braches. Each branch stands for a different value of b and forms m new leaf nodes (Xi,V,-{b}), $\le i \le m$. Then turn to Step 2.

When the number of people who have accomplished a teaching strategy using this system is up to m, then the decision tree mining module of the system begins to work. Thus, the teaching rules of system are optimized. They are more suitable for general teaching process.

4. The Productive Generation and Composition of Inference Mechanism

For the purpose of inference, three database lists are set up, i.e., a fact sheet, a rule list and a fact and rule number list. The fact sheet stores information about known facts and intermediate inference. That is to say, intermediate results are obtained from students' assessment results and inference process, through assessment modules. The rule list stores all rules and relevant information. Fact and rule numbers are a series of facts and rules used in the fact sheet and rule list (including rule conditions and conclusions), for ease of the citation of fact sheets and rule lists. Their structures are as follows:

Table 3. Fact Sheet

Field Name	Field Type	FieldDescription
rid	Char(10)	Fact No.

Table 4. Rule List

Field Names	Field Types	Field Descriptions
rid	Char(10)	Rule number
rif	Char(100)	Rule condition
rthen	Char(100)	Rule conclusion
rifcon	Char(10)	Intermediate result or final conclusion
rprino	int	Rule priority
rflag	bull	Rule identifier

Table 5. Fact and Rule Number List

Field Names	Field Types	Field Descriptions
rid	Char(10)	Rule content number
rcontent	Char(50)	Rule content

5. System Architecture

An internet model structure of the web-based intelligent teaching system is shown in Figure 4. The system adopts a browser/server mode. It is not subject to any specific operations and hardware constraints and achieves cross-platform applications. More importantly, the B/S mode reduces the client side's requirements for computer performance and learner constraints. The teaching system is placed in a remote high-performance network server and learners learn through the internet.

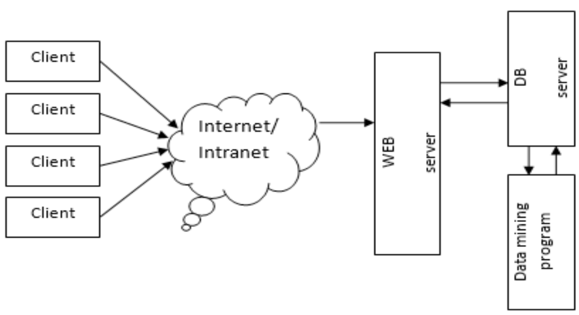

Figure 4. An Internet Model Structure of the Intelligent Teaching System

6. Conclusions

This system is based on constructivist learning theory and applies artificial intelligence technology, data mining technology and web technology, etc. to the system. It can mine potential information fully, optimize correlations between language points, adjust teaching strategies according to each student's cognitive abilities, mistakes in knowledge points, learning requirements, interests and their own characteristics, enable learners to receive customized teaching according to their cognitive abilities and personal information. In the process of teaching, students' test results and cognitive ability assessment can be generated automatically. Teaching strategies and rules can be mined, according to students' learning history, which, to a certain extent, realizes intelligent learning based on constructivism.

The design of this system is centered on teaching process. It pays attention to grasping teaching rules, takes full account of the teachers' teaching characteristics and applies them to the inference rules of system, reflects the guiding role of pedagogical ideas and theories and a combination of education and computers. It is also an important way and means to promote the intelligent teaching system towards a more humane pedagogy.

References

1. Essam Kosba, Vania Dinitrova. Roger Boyle (2007) Adaptive feedback generation to support teachers in wed-based distance education. *User Modeling and User-Adapted interaction*, 17(4), p.p.379-413
2. Jangsoon Yoo, Cen Li, Chrisila Pettey (2005) Adapative teaching strategy for online learning. *Interational Conference on Intelligent User Interfaces*, p.p.266-268
3. K.R. Koedinger , J.R. Anderson , W. Hadley and M. Mark (1997) Intelligent tutoring goes to school in the big city, international. *Journal of Artificial intelligence in Education*, No.8, p.p.30-43
4. P. Brusilovsky (1999) Adaptive and intelligent technologies for wed-based education, special issue on *Intelligent Systems and Tele-teaching*, No.4, p.p.19-25.
5. C.Conati, A.Gerner and K. Vanlehn (2002) Using Bayesian networks to manage uncertainty in student modeling. *User Modeling and User-Adapted interaction*, 12(4), p.p.371-417
6. Colucci P G, Kostandy P, Shrauner W R, et al. (2015) Development and Utilization of a Web-Based Application as a Robust Radiology Teaching Tool (RadStax) for Medical Stu-

dent Anatomy Teaching. *Academic Radiology*, 22(2), p.p.247–255.
7. Patel H I, Levin A V. (2005) Developing a Model System for Teaching Goniotomy. *Ophthalmology*, 112 (6), p.p.968–973.
8. Engvold K J, Hughes J L. (1967) A model for a multifunctional teaching system. *Communications of the Acm*, 10(6), p.p.339-342.
9. Lan T S, Lan Y H, Chen K L, et al. (2013) A Study of Developing a System Dynamics Model for the Learning Effectiveness Evaluation. *Mathematical Problems in Engineering*, p.p.1-6.
10. Dogan B, Camurcu A Y. (2008) Association Rule Mining from an Intelligent Tutor. *Journal of Educational Technology Systems*, 36(4), p.p.433-447.
11. Pahl C. (2003) Managing evolution and change in web-based teaching and learning environments. *Computers & Education*, 40(2), p.p.99-114.
12. Kerfoot B P, Baker HJackson T L, Hulbert W C, et al. (2006) A multi-institutional randomized controlled trial of adjuvant Web-based teaching to medical students. *Academic Medicine*, 81(3), p.p.224-30.
13. Karuppan C M. (2001) Web-based teaching materials: a user's profile. *Internet Research Electronic Networking Applications & Policy*, 11(2), p.p.138-149.
14. Hannafin M J, Kim M C. (2003) In search of a future: A critical analysis of research on web-based teaching and learning. *Instructional Science*, 31(4-5), p.p.347-351.
15. Nguyen A N, Uthman M O, Johnson K A. (2000) A web-based teaching program for laboratory diagnosis of coagulation disorders. *Archives of Pathology & Laboratory Medicine*, 124(4), p.p.588-93.