Creating the self-learning expert system for solving problems with fuzzy logic

V A Bakaev¹, A V Blagov¹ and Yu V Blagov²

¹Samara National Research University, Moskovskoe Shosse 34A, Samara, Russia, 443086
²Volzhsky University after V.N. Tatischev, Leningradskaya str., 16, Togliatti, Russia, 445020

e-mail: alexander.blagov@gmail.com

Abstract. There are many different approaches to building expert systems depending on the problem to be solved and the scope. This article includes a review of traditional approaches to building self-learning expert systems with fuzzy logic. The authors propose an alternative approach based on the developed model of the expert system. The article also provides a description of the generation of synthetic tests and a comparative analysis of the author's approach with existing analogues.

1. Introduction
This article explores classical approaches to building expert systems for a game similar to 20q (or Akinator), proposes an alternative model of a self-learning expert system and a number of heuristics aimed at improving the proposed model, and also compares the quality of the constructed models.

The practical value of such expert systems is to help the user to search for “I don’t know what” (for example, when we are looking for a gift for a loved one, we have only knowledge about this person and nothing more). Such systems can be used in many areas of human life and activity. For example, when preparing students for exams [1] or in the problem of image segmentation [2-4].

Such games as 20q and Akinator [5] act as a testing system, allowing developers to test the model not on "synthetic" data, but on real users. In this paper, since there is no access to a wide audience, a comparison will be made on the basis of data obtained using “synthetic” tests.

It is assumed that the user (player) either knows which object the system should offer (pre-conceived), or if the system offers a relevant object, the user will understand that this is exactly what he wanted to receive.

Then the scenario of user interaction with the system is as follows:

The system asks the user questions regarding the object (questions are selected from a previously prepared knowledge base).

The user chooses one of the answer options: “yes”, “rather yes”, “no”, “rather no” or “I don’t know”.

After several iterations, based on the user's responses, the system suggests the most relevant object. If the assumption turned out to be true, the algorithm is completed, otherwise it returns to step 1 with the accumulated knowledge of the object.
Thus, the system should not only be able to rank objects by relevance based on user responses, but also ask questions in such a way as to minimize the total number of questions asked to the user.

You can draw an analogy with the help in the search for gifts, when instead of a hidden character we have some abstract gift that should be relevant, and questions are asked about the hero of the occasion. Also, we make the assumption that if the user purchased a product, then it is relevant.

The system can be divided into several components:
- the method of choosing the most likely answer option;
- the method of choosing the next question;
- the method of choosing the moment when the system is ready to offer an answer.

2. Traditional approaches to the construction of expert systems

Traditionally, to build expert systems with fuzzy logic, an approach based on decision trees is used, as well as a probabilistic approach based on the Bayes theorem [6].

2.1. Decision Tree Approach

A decision tree is a mathematical model that allows you to visualize the logical structure of decision making. It appears, as a rule, in the form of an acyclic oriented graph, at the vertices of which there are questions, in the edges - possible answers to questions, and in the leaves - the conclusions of the system.

This approach to building expert systems is the most common because of its simplicity and sufficient efficiency to solve most applied problems.

As a rule, the knowledge base of such a system is compiled manually by experts in relevant fields. Such an approach solves the problem of a “cold start”, but it can be extremely costly.

The main disadvantage of this approach is the instability of the system to errors made by users. This means that if the user made a mistake (accidentally or because of ignorance of the correct answer to the question posed), then the system is unable to ask a clarifying question in any way, which leads to false conclusions. Also, the disadvantages include the fact that the tree needs to be rebalanced, there are questions for which there is no definitive answer and, if the system does not find a match in the database, you cannot “poke a finger at the sky” and choose the most similar object.

The advantage of this approach is its simplicity and low computational complexity (the choice of the next question is made in $O(1)$).

2.2. Probabilistic approach based on Bayes’ theorem

The disadvantages of this approach include a high computational complexity when choosing a question, as well as a slight increase in efficiency when using clever modifications [6].

Suppose that the user has already answered a certain number of questions and now the system must put forward an assumption which C object will suit the user. To this end, it is proposed to use the Bayes’ theorem:

$$P(c | Q_1, A_1, ..., Q_n, A_n) = \frac{P(Q_1, A_1, ..., Q_n, A_n | c) P(c)}{\sum_{c'} P(Q_1, A_1, ..., Q_n, A_n | c') P(c')}$$

(1)

where $c$ is an element from the set of objects that can be offered to users as required answer; $P(c)$ is a priori probability that the user will approach the object $c$ (can be calculated as the proportion of sessions in which were chosen the object, among all the sessions); $(Q_i, A_i)$ is a pair consisting of a question that was asked of the user and its corresponding answer.

To simplify the calculations it is assumed conditional independence of the responses to the questions for a given object:

$$P(Q_1, A_1, ..., Q_n, A_n | c) = \prod_{i=1}^{n} P(Q_i, A_i | c).$$

(2)

Substituting the expression (1.2) in the expression (1.1) we get the following formula to calculate the probability of object selection $C$ when you know the answers to the questions:
Based on the history of the games and the answers that the user has already given, we can estimate the probability of each response option with a fixed the question according to the following formula:

\[ P_A = \sum_c P((Q, A) | c)P(c | \langle Q_1, A_1 \rangle, \ldots, \langle Q_n, A_n \rangle), \]

where \( Q \) is fixed issue which chose to ask its user; \( A \) is the answer to the question, whose probability we are going to evaluate.

The idea of the method of selecting the question is: if we played a game with response options "Yes" and "no", the optimal strategy would be the choice of the question that cuts off half of the answers. The author proposes a generalization of this method, in which the system will each time choose a question that minimizes the conditional entropy when a certain response [5]:

\[ H(Q, A) = H[P(c | \langle Q_1, A_1 \rangle, \ldots, \langle Q_n, A_n \rangle, \langle Q, A \rangle)]; \]

\[ Q^* = \arg \min_Q \sum_A H(Q, A) P_A. \]

Thus it is possible to choose the following question during the \( O(|C| + |Q| + |A|) \).

As the moment at which the system should put forward their assumption, the authors propose to use the following condition:

\[ \exists c \epsilon C : P(c | \langle Q_1, A_1 \rangle, \ldots, \langle Q_n, A_n \rangle) \geq 0.8. \]

3. Development of the expert system model for solving problems with fuzzy logic

In the approach proposed by the authors of this article, we deal with a matrix of size \( |Q| \times |C| \):

\[ P = \begin{pmatrix}
    p_{11} & \cdots & p_{1m} \\
    \vdots & \ddots & \vdots \\
    p_{n1} & \cdots & p_{nm}
\end{pmatrix}, \]

where \( p_{ij} \) is the proportion of sessions in which the \( i \)-th question for the \( j \)-th object the answer was "Yes" among all sessions normalized to the interval \([-1; 1]\).

3.1. Basic matrix model

From a practical point of view it is advantageous to store not the actual matrix \( P \), and its components: \( P_+ \) and \( P_- \) are matrices that holds the number of positive and negative answers to each question accordingly for each object. The matrix \( P \) can be found by the following formula:

\[ p_{ij} = \frac{2 p_{+ij}}{p_{+ij} + p_{-ij}} - 1, \quad i \in [0, |Q|], \quad j \in [0, |C|]. \]

We call the proposed model - matrix.

In developing the model, based on the ideas of collaborative filtering [7, 8], were made the following assumptions:

- \( p_{ij} \xrightarrow{K \to \infty} 0 \), if the \( i \)-th question does not affect the choice of the \( j \)-th object;
- \( p_{ij} \xrightarrow{K \to \infty} 1 \), if the \( j \)-th object has the property characterized by a positive response to the \( i \)-th issue;
- \( p_{ij} \xrightarrow{K \to \infty} -1 \), if the \( j \)-th object has the property characterized by a negative response to the \( i \)-th issue;
- the user's answers are presented in the form of a vector-row \( A = (a_1, a_2, \ldots, a_n) \), where \( a_i \in \{-1, 0, 1\} \) and characterizes the user's answer to the \( i \)-th question;
- answer "don't know" is not used in further calculations and is equivalent to a situation in which the question is not asked;
- the responses "probably Yes" and "probably not" are used when searching for personalized recommendations, but is not stored in the matrix \( P_+ \) and \( P_- \) respectively.

The advantages of this method include its simplicity and the ability to add modifications at any step of the algorithm.
Because each object is characterized by its column vector \( P_j = (p_{1j}, p_{2j}, ..., p_{nj})^T \) from the matrix \( P \), we can, using the cosine measure to determine the rule to select the most relevant object. However, since it is important only to maintain the order of ranking in order to save computational resources it is possible to abandon rationing. Then the formula for selecting the most relevant object will look like the following:

\[
C^* = \arg \max_j (A, P_j).
\]

(9)

Thus, we choose the object for which the mean vector of the responses across the set of sessions is closest to the vector of users ‘ responses.

Selecting a next question, we, by analogy with the probabilistic approach based on Bayes ‘ theorem, we will try to choose an issue that cuts away half of the answers. In this case, those issues that do not affect the choice of the majority of objects should have the lowest priority. Based on the described requirements were obtained the following metric to select the next question:

\[
Q^* = \arg \max_i \sum_j \sum_{k=j+1}^m (|p_{ij}| + |p_{ik}|).
\]

(10)

It is obvious that in this approach, the maximum is achieved in the case when the number of positive elements in the string corresponds to the number of negative elements and the absolute values of all line items close to 1. However, the computational complexity of this method equals \( O(|Q| \cdot |C|) \), that is much better than the probabilistic approach.

In the proposed implementation matrix of the model, the system suggests the most likely object if you specified a minimum of 10 questions and at least one of the following conditions:

- it was asked 25 questions in the current attempts;
- there is at least one object not previously proposed, the relevance of which exceeds a predetermined threshold;
- each question was asked 2 times.

To solve a "cold start" is proposed to initialize the vectors of the responses are not zero vectors, and random vectors with uniform or normal distribution. When adding an object you can use one of the following options:

- using trained neural network (for example, word2vec [9]) to find \( m \) the most similar objects name for the new object and initiate its vector of responses, based on vector objects-neighbors;
- to identify groups of objects (one object can belong to multiple groups) and, using probabilistic models (e.g., LDA [10]) to partition the objects in the knowledge base, after which mark a new object and find \( m \) most similar objects from the point of view of the labels.

In both proposed options, the parameter \( m \) selected independently on the basis of experiments with test data.

System logs user session (asked questions, received answers, suggested objects and the user’s choice) into a text file or a specialized DBMS (for example, ClickHouse). Then the learning algorithm of this model is the initial columns of the matrix \( P \) and terms of columns of the matrix \( P \) when new \( k \) records for a single object in the log. In this case, conversion occurs at the \( n \) last sessions (\( n \geq k \)), which allows to take into account the seasonal factor.

3.2. Heuristic methods for improving the quality of the model

Based on the idea of item-based collaborative filtering approach will introduce an additional matrix \( S \) (question-question), containing the indices of similarity of issues. As a measure of similarity of questions we use cosine measure:

\[
S_{ij} = \frac{\sum_{k=1}^{|C|} p_{ik} p_{jk}}{\sqrt{\sum_{k=1}^{|C|} p_{ik}^2} \cdot \sqrt{\sum_{j=1}^{|C|} p_{jk}^2}}
\]

(11)
It is obvious that the diagonal elements of the matrix $S$ equal to 1, and its dimension is equal to $|Q| \times |Q|$. The recommendations list $rec_c$ obtained by multiplying the vector-line answers $a$ user, $S$ $S$ matrix and matrix $P$:

$$rec_c = a \cdot S \cdot P.$$  

(12)

The resulting vector $rec_c$ necessary to equate to zero the elements that the user is already assessed as not relevant and sort the objects in descending order of values.

In addition, it is possible to improve the method of selecting the next question. The choice of question, as described in basic idea of this model, though is able to cut off half of the objects, but it is not consider the answers given by the user during the session. In consequence, regardless of the user's response, with equal matrices $P$ model will offer the same sequence of questions. This can be avoided if you manually cut off some number of the least relevant objects. It is necessary to keep a share of the most relevant objects $rel\_factor$, involved in the selection of the question and for each question to calculate it $split\_factor_i$ – the ratio of the minimum between positive and negative number of responses to question total number of answers (excluding the 0).

The default value $rel\_factor$ is 1. Then, when you select the i-th question, it is multiplied by $split\_factor_i$. Then, the amount of the most relevant objects involved in the question is equal to $[rel\_factor \cdot |C|]$. To eliminate noise caused by false user's answers to certain questions, it is necessary to reduce the influence of conflicting answers. You can use the matrix $S$ and the vector of users' responses $a$.

4. Results and discussions

To test the approaches described in this paper was developed a generator of synthetic tests and the system to test the models.

Based on the testing of the system is the idea proposed by a researcher at Duke University [11]. The algorithm testing system consists of the following:

- in space $\{(x, y) : x \in [0; 1] \land y \in [0; 1]\}$ randomly generated $m$ points denoting objects and $n$ vectors denoting questions. If the j-th point lies to the right of the i-th line, then the answer to the ith question for the j-th object is "Yes", otherwise "No".
- *: Generates a column vector $mistakes$ size $n$ (histogram distribution shown in figure 2). For the i-th question the corresponding element of the vector $mistakes$ specifies the probability with which the system needs to give a false answer.
- The i-th question with probability $nkp = 0,2$ the system responds "don't know", and with probability $mistakes_i$ gives a wrong answer.
- The i-th question with probability $mp = 0,25$ and the system responds with "probably Yes" or "probably not" depending on the answer computed in step 3.
- If the model returned the expected response, then calculated the Euclidean distance from the proposed facility to the target and if it is less than the threshold ($threshold = 0,1$), we consider that the model gave a correct answer, otherwise invalid.

This approach allows you to emulate user error, not knowing the answer to the questions asked and the specific issues on which to give the correct answer is quite difficult. In addition, the objects are initially generated in such a way that they can be divided into clusters so one vector of responses consistent with several relevant objects (not just the one that originally made the system).

4.1. Research models in the absence of noise from the testing system

To study probabilistic, matrix and improved matrix models, we use the tools of statistical analysis. To do this, it is necessary to compare the sample mean and standard deviation of the number of questions asked to the user (in this case, the testing system) and the number of attempts by the expert system to find the most relevant object.

All the models studied showed that the average number of attempts required by a probabilistic model for issuing the most relevant object decreases with time.

Since the testing system requires the values of three parameters (number of sessions, number of objects and number of questions), then to assess the dependence of the result of the initial data we will
conduct a series of experiments in which we will fix two parameters, and the third will remain changeable. Tables 1, 2 and 3 present the sample mean and standard deviation of the number of model attempts with the absence of noise and the changing parameter of the number of objects, the number of questions and the number of sessions, respectively.

Table 1. The comparing of models with no noise, a fixed number of sessions and questions.

| The number of objects | The probabilistic model | The matrix model | The improved matrix model |
|-----------------------|-------------------------|------------------|---------------------------|
| 8                     | 1,2640 ± 0,5462         | 1,0220 ± 0,3186  | 1,0240 ± 0,3278           |
| 16                    | 1,5520 ± 0,9227         | 1,2280 ± 0,6356  | 1,0780 ± 0,5476           |
| 32                    | 1,7280 ± 1,3748         | 1,2300 ± 1,0775  | 1,1340 ± 0,9613           |
| 64                    | 1,7400 ± 1,8134         | 1,3380 ± 1,4628  | 1,3160 ± 1,5862           |
| 128                   | 1,8460 ± 1,6511         | 1,4020 ± 1,1386  | 1,2720 ± 1,1287           |
| 256                   | 2,0300 ± 1,9650         | 1,3620 ± 1,5215  | 1,2780 ± 1,5443           |
| 512                   | 2,0780 ± 2,4397         | 1,4620 ± 1,7597  | 1,4000 ± 1,5672           |

Table 2. The comparison of models in the absence of noise, a fixed number of sessions and objects.

| The number of objects | The probabilistic model | The matrix model | The improved matrix model |
|-----------------------|-------------------------|------------------|---------------------------|
| 4                     | 2,4940 ± 2,0818         | 2,0500 ± 1,9513  | 2,0080 ± 1,7458           |
| 8                     | 1,8560 ± 1,7605         | 1,3500 ± 1,3280  | 1,3280 ± 1,4438           |
| 16                    | 2,2060 ± 3,5122         | 1,4360 ± 1,8016  | 1,4840 ± 2,9621           |
| 32                    | 1,8200 ± 1,5178         | 1,4600 ± 1,1834  | 1,3240 ± 1,0349           |
| 64                    | 1,8180 ± 2,0112         | 1,2540 ± 1,5904  | 1,3740 ± 1,7176           |
| 128                   | 2,2280 ± 2,9278         | 1,4800 ± 2,2860  | 1,5620 ± 2,3896           |

Table 3. The comparison of models in the absence of noise, a fixed number of objects and questions.

| The number of objects | The probabilistic model | The matrix model | The improved matrix model |
|-----------------------|-------------------------|------------------|---------------------------|
| 125                   | 2,1120 ± 2,0049         | 1,5920 ± 1,6642  | 1,4480 ± 1,5720           |
| 250                   | 2,0520 ± 2,2506         | 1,6040 ± 1,7249  | 1,5240 ± 1,7825           |
| 500                   | 1,9280 ± 2,2322         | 1,5120 ± 1,9621  | 1,3980 ± 2,0600           |
| 1000                  | 2,0180 ± 1,9125         | 1,5190 ± 1,3280  | 1,3270 ± 1,3535           |
| 2000                  | 1,9005 ± 1,2906         | 1,2675 ± 0,7937  | 1,1620 ± 0,7468           |
| 4000                  | 1,7213 ± 1,2753         | 1,2443 ± 0,7563  | 1,1343 ± 0,6595           |
| 8000                  | 1,8425 ± 1,2273         | 1,1627 ± 0,5170  | 1,1745 ± 0,5613           |

4.2. The models research with the presence of noise from the testing system

To conduct testing with the presence of noise, we will set the corresponding parameters for launching the testing system.

All the models studied showed that the average number of attempts required by a probabilistic model for issuing the most relevant object decreases with time. In addition, it can be noted that the
probabilistic model is less resistant to noise caused by erroneous responses than the matrix or improved matrix model.

Tables 4, 5 and 6 present the sample mean and standard deviation of the number of model attempts with the presence of noise and the changing parameter of the number of objects, the number of questions and the number of sessions, respectively.

The tables show that the implementation of the matrix model proved to be more effective than the probabilistic model. However, given the computational complexity, choosing the best model may not be so obvious.

**Table 4.** The comparison of models in the absence of noise, a fixed number of sessions in questions.

| The number of objects | The probabilistic model | The matrix model | The improved matrix model |
|-----------------------|-------------------------|------------------|--------------------------|
| 8                     | 1,8440 ± 1,0018         | 1,3840 ± 0,5767  | 1,1980 ± 0,4927          |
| 16                    | 2,5700 ± 1,5624         | 1,3800 ± 0,8072  | 1,2860 ± 0,8533          |
| 32                    | 2,3360 ± 1,5489         | 1,4500 ± 0,8316  | 1,2960 ± 0,8002          |
| 64                    | 2,9700 ± 2,3206         | 1,6460 ± 1,5839  | 1,5060 ± 1,6619          |
| 128                   | 2,8180 ± 2,4333         | 1,7360 ± 1,5147  | 1,5460 ± 1,5569          |
| 256                   | 2,5780 ± 1,9920         | 1,5180 ± 1,1514  | 1,3780 ± 1,0272          |
| 512                   | 2,8980 ± 2,4041         | 1,6660 ± 1,4854  | 1,5220 ± 1,4414          |

**Table 5.** The comparison of models in the absence of noise, a fixed number of sessions and objects.

| The number of objects | The probabilistic model | The matrix model | The improved matrix model |
|-----------------------|-------------------------|------------------|--------------------------|
| 4                     | 3,1760 ± 2,2138         | 2,5380 ± 2,4877  | 2,7400 ± 2,8755          |
| 8                     | 3,4380 ± 3,1531         | 2,7580 ± 2,9556  | 2,4740 ± 2,9178          |
| 16                    | 2,4720 ± 2,1357         | 1,7720 ± 1,5297  | 1,5920 ± 1,7128          |
| 32                    | 3,2980 ± 2,6999         | 1,7000 ± 2,0722  | 1,4800 ± 1,9240          |
| 64                    | 3,0320 ± 2,6561         | 1,6540 ± 1,9022  | 1,5480 ± 1,7910          |
| 128                   | 3,1180 ± 2,8817         | 1,6240 ± 2,2527  | 1,5760 ± 2,3580          |

**Table 6.** The comparison of models in the absence of noise, a fixed number of objects and questions.

| The number of objects | The probabilistic model | The matrix model | The improved matrix model |
|-----------------------|-------------------------|------------------|--------------------------|
| 125                   | 3,6320 ± 3,4906         | 2,0080 ± 2,2996  | 2,0720 ± 2,1398          |
| 250                   | 2,6600 ± 2,9490         | 1,8640 ± 2,4506  | 1,9240 ± 2,5468          |
| 500                   | 2,9120 ± 2,4723         | 1,7120 ± 1,7633  | 1,6960 ± 2,3297          |
| 1000                  | 2,3090 ± 1,7882         | 1,5690 ± 1,7144  | 1,5560 ± 1,6078          |
| 2000                  | 2,3375 ± 1,7876         | 1,4090 ± 1,0372  | 1,3270 ± 0,9675          |
| 4000                  | 1,7123 ± 0,9864         | 1,3793 ± 0,7751  | 1,3493 ± 0,8282          |
| 8000                  | 2,3320 ± 1,6425         | 1,3845 ± 0,8368  | 1,3630 ± 1,0323          |
5. Conclusions
In this article, classical approaches to the construction of expert systems were considered, two versions of the expert system model (basic and improved) were developed and they were experimentally compared with a probabilistic model based on the Bayes theorem.

Both implementations of the matrix model proved to be more effective than the probabilistic model both from the number of attempts required to form personalized recommendations and from the speed of learning. However, unlike the probabilistic model, both implementations of the matrix model are the most computationally complex. All investigated methods are resistant to errors made by users.

The number of attempts required to form recommendations, the usual and improved matrix models differ by a maximum of 7%, however, the computational complexity of the improved model is much higher, since the vector of recommendations must be calculated at each iteration, and not only when issuing recommendations to the user.

6. References
[1] Williams R G, Klamen D L 2015 Twenty Questions game performance on medical school entrance predicts clinical performance Medical Education 49(9) 920-927
[2] Rupprecht C, Peter L and Navab N 2015 Image Segmentation in Twenty Questions The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 3314-3322
[3] Gaidel A V, Kapishnikov A V, Pyshkina Yu S, Kolsanov A V and Khramov A G 2018 Method of nephroscintigraphic dynamic images analysis Computer Optics 42(4) 688-694 DOI: 10.18287/2412-6179-2018-42-4-688-694
[4] Zakani F R, Bouksim M and Arhid K 2018 Segmentation of 3d meshes combining the artificial neural network classifier and the spectral clustering Computer Optics 42(2) 312-319 DOI: 10.18287/2412-6179-2018-42-2-312-319
[5] Akinator and Mathematics URL: https://habr.com/post/84364
[6] Jedynak B, Frazier P and Sznitman R 2012 Twenty questions with noise: Bayes optimal policies for entropy loss Network Journal of Applied Probability 49 114-136
[7] Item-based collaborative filtering do it yourself URL: https://habr.com/ru/company/ivi/blog/232843
[8] Deshpande M, Karypsis G 2004 Item-based top-N recommendation algorithms ACM Transactions on Information Systems (TOIS) 22(1) 143-177
[9] Mikolov T, Sutskever I, Chen K, Corrado G and Dean J 2013 Distributed Representations of Words and Phrases and their Compositionality Advances in Neural Information Processing Systems 26 1-9
[10] Balakrishnama S, Aravind G 1998 Linear discriminant analysis-a brief tutorial Institute for Signal and Information Processing 18 1-8
[11] A Bayesian Strategy to the 20 Question Game with Applications to Recommender Systems 2010 (Duke University Libraries)