Journal of Business and Economics, ISSN 2155-7950, USA
September 2020, Volume 11, No. 9, pp. 1057-1064
DOI: 10.15341/jbe(2155-7950)/09.11.2020/007

http://www.academicstar.us

Classification of Guided Expert Systems According to Their Accuracy, Domain Resolution, and Range

Dan Ophir
(Computer Science Department, Ariel University, Israel)

Abstract: An "Expert System" is based on "trial and error". The configurations' classification is difficult for the human user; however, such categorization may be performed with the assistance of a human experts. The random configurations generated by the computer may be classified and for further computer expert system interrogation by the user.

Imparting knowledge can be performed as follows:

- *Transferring the knowledge* — the instructor shows the collection of elements $A = \{e_1, e_2, \ldots, e_n\}$ to the learning entity, and each value v_i is assigned to the corresponding element e_i ($i = 1, 2, \ldots n$).
- *Using the knowledge* — the learning entity receives an element *e*, $e \in U$ for consideration, where $A \subset U$, and assigns to *e* its corresponding value, *v*. The learning entity scans the values; e_i, finds the element e_j, the closest to *e*, and joins it to the value v_j.

The computerized system is used as a model, enabling formalization and deduction for humans. In the computerized case, it is possible to observe a series of *Expert Systems* in the ascending orders of complexity, defined as follows:

- *Binary* case — the elements may receive only one of two values: "yes" or "no", without the possibility of attaching two different values to the same element.
- *Ternary deterministic* — the elements may receive one of three values "yes", "no", and "maybe".
- *Ternary nondeterministic* — here the same element may appear with various values that are statistically supported.
- *Canonization of elements* — here collection *A* is defined as $A = \{A_1, A_2, \ldots, A_k\}$, where A_l ($l = 1, 2, \ldots,$ k represents: $A_l = \{e_1^l, e_2^l, \ldots, e_{n_l}^l\}$; the computerized system deduces, based on the element e_m^l , the corresponding value, v_l.
- *Element-ranged canonization* — here the collection of elements *A* is defined as follows:

$$A_l = \{\delta_1, \delta_2, \ldots, \delta_{n_l}\} \ (l = 1, 2, \ldots k),\ A = \{A_1, A_2, \ldots, A_{k,}\}$$

$\delta_j = \lfloor \alpha_j, \beta_j \rfloor$ is an interval of integer numbers; the computerized system deduces the corresponding value v_l, relying on elements from the interval A_j.

The following examples, graded according to their complexity and their level of uncertainty, represent the

Dan Ophir, Dr., Senior Lecturer and Researcher, Department of Mathematics and Computer Science, Ariel University. E-mail: dano@ariel.ac.il.

above *Expert System's* classification:

- Determining correct blood transfusion;
- Teaching the Traffic Light rules in two modes:
  - Deterministic
  - Nondeterministic
- Identifying *Deductive* characters
- *Recognizing Color Hues* — based on the colors' RGB ranges.

Key words: expert system; knowledge transferring; structured query language (SQL); determinism; optical character recognition (OCR)

JEL code: C630

1. Introduction

This paper deals with *statistical expert systems* **(**Segura Jason M. & Reiter Albert C., 2012), which are a means of *transferring knowledge*; this refers to knowledge management and not to *computational expert systems* using in-depth learning methods (Bengio Yoshua, 2017).

A simple methodology for managing this transfer is shown. This methodology has to consider various cases such as repetition of information, lack of information, extrapolation, and others. The precision of such operations is estimated by the expert systems introduced.

Knowledge and management are two terms representing mutually connecting operations. Such interoperability is described here. This simple methodology enables students to develop similar expert systems in various domains by themselves.

2. Methodology

The following examples will contain modules, which will be later explained in detail. The paradigm of the "*expert system*" is to supply the human expert with the generated configurations, receive his expertise — *the learning mode*, and save it for further use — *the applying mode*. The learning mode of all the introduced examples has the same frame and infrastructures, but it differs in the set of properties being analyzed and separated into clusters (Aggarwal C. Charu & Reddy K. Chandan, 2014).

2.1 Preparation of Bulk Random Configurations

A set of random configurations and their corresponding acceptation indications are generated and made available to the system's users. The purpose of such generation is to accelerate the data preparation for further demonstration. In reality, however, such data are prepared individually by human experts.

2.2 Data Base Manipulations

A data base appears in a corresponding table showing the human expert's decisions, for each generated configuration. The data base performs, according to the *expert system* type, suitable queries for further operations.

2.3 Simplification

The *Expert System Kernel* performs some operations to simplify the data representing the configurations provided for classifying expertise before storing it in the data base in an equivalent simplified form.

2.4 GUI Representation

The GUI contains two parts:

2.4.1 Learning Mode

This mode is responsible for generating random data for human expert inspection.

2.4.2 Applying Mode

This mode represents facilities enabling the user to formalize a question or data for a *computer expert* to give his opinion. The expert system's opinion may be unambiguous or express some doubt regarding its expertise and reliability, accompanied by a computed probability.

3. Examples

3.1 Transfusion Expert System

3.1.1 GUI

The GUI (Figure 1) has two main parts: *Knowledge Building* and *Testing*. The first part generates random configurations of possible blood groups (Daniels Geoff & Bromilow Imelda, 2013), of the Donor and the Recipient. Each is evaluated by a human expert and his opinion is saved in the data base, for further computer system usage in the testing mode. In the testing mode the user invokes the expert system, which performs some data manipulation and retrieves corresponding expertise from its data base, displaying it in the corresponding text-box of the GUI.

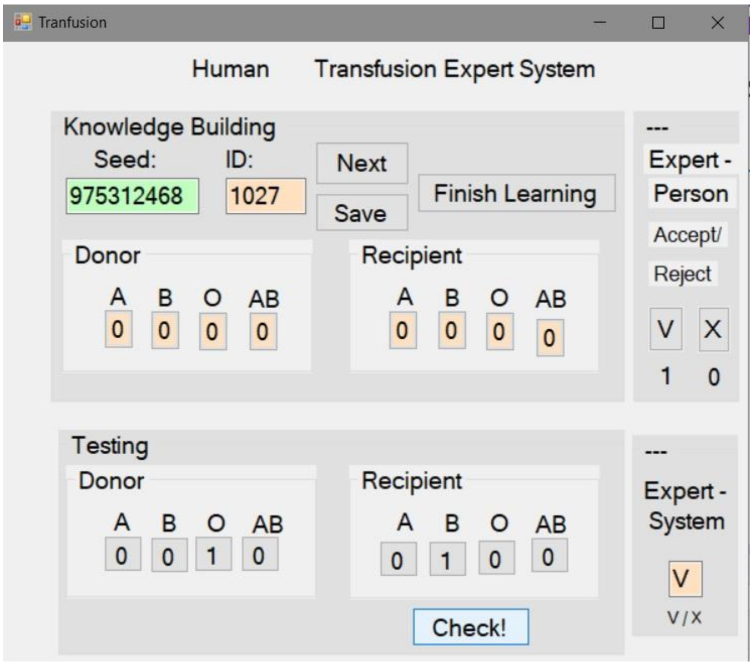

Figure 1 Transfusion Expert System GUI: Transfusion From Donor With "O" Blood Group to Recipient With "B" Blood Group Is Accepted

3.1.2 Generating Configurations

For pedagogical purposes only, the configuration of the blood group's donors and recipients is generated, with an opinion of having the configuration either accepted or rejected. The collection of configurations, with expertise about the suitability for transfusion, provides a fast demonstration of the DB module's manipulations.

This module performs its operation on a worksheet using columns, fields, and computations, and maintains the transfusion rules: each recipient corresponds to only one donor for each transfusion. The correspondence between the donor's blood groups and the recipients is shown in Figure 2.

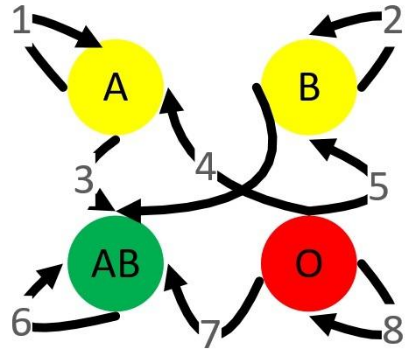

Figure 2 Transfusion Rules With Indicated Groups

3.1.3 Data Base Treatment

The performance, presented in the Table 1, is investigated by SQL queries. For example, Figure 3 represents a query checking whether a donor with the blood group type O could transfuse his blood to a recipient with blood type *A*. This query is applied on a query, that has previously checked the appropriate number of donors and recipients, which should be one.

Table 1 Table Termed T_Random_Base Contains Columns Indicating Whether the Trial of Transfusing Donors' Blood to Recipients May Succed and Is Accepted

T_Random_Base

ID	Don_A	Don_B	Don_O	Don_AB	Rec_A	Rec_B	Rec_O	Rec_AB	Accept	Dec_Value
1	0	1	1	0	0	0	1	0	0	
2	0	0	1	0	1	0	0	1	0	
3	1	1	1	0	0	1	1	0	0	
4	1	1	1	1	1	0	0	1	0	

```
SELECT Q2_Accept.Accept
FROM Q2_Accept
WHERE Don_O=1 AND Rec_A=1;
```

Figure 3 A Query Checking Whether A Donor With the blood Group Type O May Transfuse His Blood to a Recipient With Blood Type A

3.1.4 Traffic Lights

Let us imagine that aliens have landed on earth. They learn how the traffic lights operate (Sano Gina, 2017) the same way that our computerized expert system learns the traffic light rules.

Three variations of this example are given (Figure 4):

- Binary — Black-white case; there are only two cases: Green-Red: save or danger. In this case the elements being investigated may receive only one of two values: "yes" or "no", without the possibility of attaching two different values to the same element.
- Ternary — a yellow light gives a ½ probability of traversing the crossroad safely — the doubt is considered, namely, the investigated elements may receive one of three values: "yes", "no", and "maybe".
- Non-deterministic — a version considering the mistakes of human experts teaching the computerized

expert system. In this case, most opinions can be decided and their probability is computed. In this variant the same investigated element may appear with various values statistically supported.

Figure 4 Traffic Lights GUI

3.2 Character Recognition

3.2.1 GUI

The main GUI window is similar to that of traffic lights (Figure 4). This expert system enables supplying the system with a form (Figure 5a) of taught characters (Rice V. Stephen et al., 1999). The human examines the characters, which are saved for further use by the computerized expert system. Figure 5b represents the same characters as in Figure 5a, but they are in the checking mode. In both cases the main properties of the characters are kept the same; however, the characters may be somehow distorted.

3.2.2 Algorithm

The classical algorithms of character recognition are associated with deep learning (Rice V. Stephen et al., 1999); here a simple self-written algorithm is used independent of the infrastructure.

In the presented algorithm, in the matrices (Figure 5) the columns are analyzed – non-zero substrings search for a string of non-zero substrings in columns (see the bottom of Figure 5). This generates a string of numbers, which is reduced by eliminating successive equal numbers.

The presented algorithm performing the grid-matrix linearization finally leads, in the example (Figure 5) to the string "*121*". This is a pattern standing for the treated character, namely, "*0*". Each such computed pattern is attached to the analyzed character and saved/retrieved /from the data base, during the recognition process.

In order to reduce ambiguity, the algorithm may be upgraded by performing row computing, analogically to the column one.

3.2.3 Mapping Formalization

Canonization of elements is a term used in the presented case. Collection A is defined as $A = \{A_1, A_2, \ldots, A_k\}$, where A_l (collections of grids representing the same value) can be defined:

$A_l = \{e_1^l, e_2^l, ..., e_{n_l}^l\}$ *(l=1,2,..., k)*, the element e_m^l (the grid points — Figure 5), whose corresponding value is v_l (interpreting the corresponding value, namely, "0" Figure 5). The canonical element in the present example is a pattern ("*121*" corresponds to the given character "*0*" — Figure 5) representing A_l. Therefore, here the pattern "*121*" denotes any grid representations of the digit "0". The grids representing "0" in Figure 5a and the distorted grid representing "0" in Figure 5b have the same pattern "121".

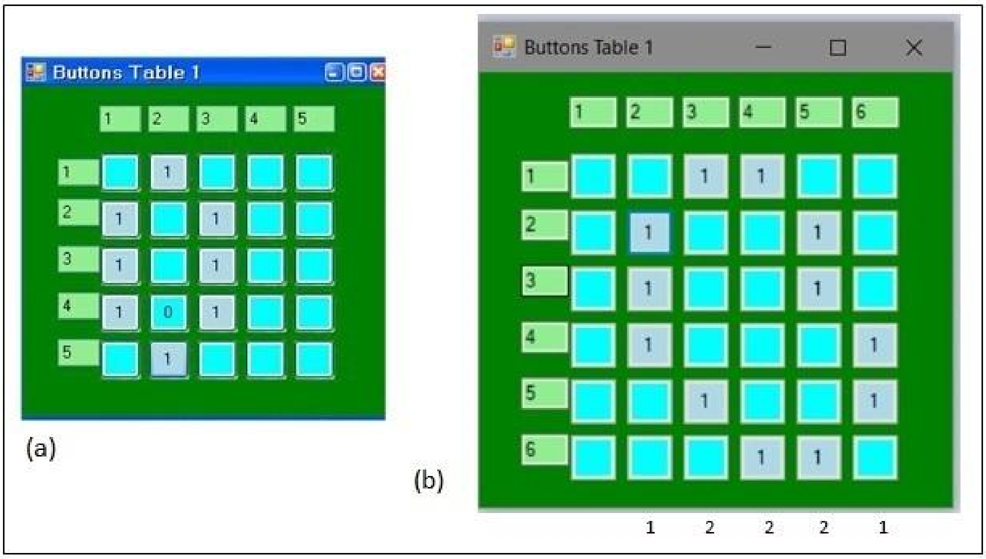

Figure 5 Intermediate Data Structures Participating in the Recognition Process: (a) "0" Is Represented by A Grid in Which the Grid-Point Stands for A Pixel of the Examined Character. A Grid-Point With the Value "*1*" Stands for the Black Pixel, Whereas the Other Represents the Background of the Character. (B) How a Character Is Represented During the "Checking Mode". This Character May Be Distorted But May Be Recognized by the Expert System.

3.3 Color Recognition

The presented system learns some hues and the expert system can deduct other close hues (Lambert Surhone M. 2010). This perfected system performs some kind of deduction by testing the received color. Namely, the color recognition system upgrades the previous systems shown here, which require the exact values of the learned values and the checked values of the investigated object represented by suitable data. This constraint is not required any more.

3.3.1 GUI

The corresponding main GUI window is similar to that of previously presented expert systems; the GUI of the color recognition differs from the others by the secondary GUI windows (Figure 6). This window enables the user to move the horizontal scrollbars and thus govern the RGB components' values and influence the right color container.

3.3.2 Algorithm

The expert system searches the data base for a configuration close to the checked configuration, as represented in Figure 7. Namely, learned RGB configuration values are compared with the corresponding user-given values and then the expert system utilizes its expertise about the name of the presented colour. In this algorithm ϵ denotes any small value that influences the level of finesse — chromatics. The criteria for naming colors are even sharpened and are used instead of comparing the RGB colors' intensity. The proportion of RGB color components in the data base is compared with those in the RGB configuration under investigation.

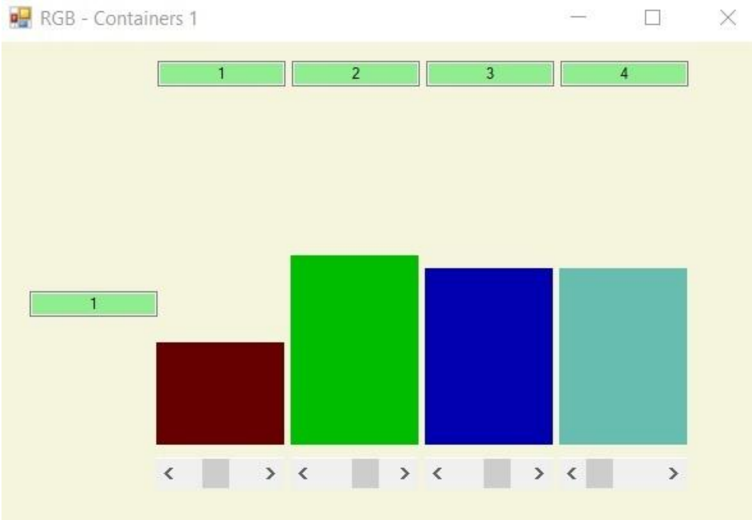

Figure 6 Composing A "Turquoise" Colour From Its Basic RGB Values: (40%, 74%, and 69%) — The Percentage Represents the Maximal Colour Intensity, Namely, 255. There Are Three Left Colour Containers of the Basic Colour; The Right Container in the Illustration Represents A Mixture of the Left Containers

$$\sqrt{(R_{DB} - R_{Checked})^2 + (G_{DB} - G_{Checked})^2 + (B_{DB} - B_{Checked})^2} \quad < \epsilon$$

Figure 7 The Criteria of the Distance Between the RGB Configurations, As Saved In the Data Base and the RGB Configuration Supplied by the User in the Checking Mode

3.3.3 Mapping Formalization

Element-ranged canonization — here the collection of elements in A **(**RGB configuration: $n_l = 3$, δ_1, δ_2, δ_3 correspond to "R", "G", and "B", respectively) are defined as follows:

$$A_l = \{\delta_1, \delta_2, \ldots, \delta_{n_l}\} \text{ (l = 1, 2, …k), A = \{A}_1\text{, A}_2\text{, …, A}_k\text{\}}$$

$\delta_j = \lfloor \alpha_j, \beta_j \rfloor$ (lower and upper limits of the RGB intensities) is an interval of integer numbers that the computerized system deduces, relying on elements from the interval A_j, the corresponding value v_l (the corresponding colour name).

4. Conclusion

A child learns his mother language through examples and counterexamples. The presented article introduces a computer system that imitates that way of learning.

The paradigm of the presented *expert system* is based on statistical deduction in which the computer system acquires knowledge by being exposed to the positive and negative cases of applying that knowledge. The methodology of the *expert system* is shown by applying it to teaching and using its own methodology, namely, by showing a set of examples of its (the methodology's) implementation.

The statistical *expert system* begins to develop by emulating the human being's behavioral learning, and it is used by him to upgrade his knowledge in the same way that the expert system has acquired its knowledge by imitating that of a human, and even an animal's learning methodology is based on its instincts. Namely, some type of *closed loop (*Abouelabbas Ghanaim, 2011) is presented here.

References

Abouelabbas Ghanaim (2011). *Modeling and Control of Closed Loop Networked PLC-systems: Modellierung Und Regelung Von Vernetzten SPS-regelungssystemen*, Shaker Verlag GmbH, Germany.

Aggarwal C. Charu and Reddy K. Chandan (2014). *Data Clustering Algorithms and Application*, CRCP Press.

Bengio Yoshua (2017). *Deep Learning*, MIT Press Ltd.

Daniels Geoff and Bromilow Imelda (2013). *Essential Guide to Blood Groups 3E*, John Wiley & Sons Inc.

Rice V. Stephen et al. (1999). *Optical Character Recognition: An Illustrated Guide to the Frontier*, Springer.

Sano Gina (2017). *Traffic Lights*, *Creative*, Media Partners, LLC.

Segura Jason M. and Reiter Albert C. (2012). *Expert System Software: Engineering, Advantages & Applications*, Nova Science Publishers Inc.

Lambert Surhone M. (2010). *Portable Network Graphics: Raster Graphics, Image File Formats, Lossless Data Compression, Graphics Interchange Format, Recursive Acronym, RGB Color Model*, Usenet Newsgroup: Betascript Publishers.