Connectionist-Based Decision Support Systems and Expert Systems

Nikola Kasabov

Department of Information Science, University of Otago,

P O Box 56, Dunedin, New Zealand,

Tel +64 3 479 8319 Fax +64 3 479 8311

Email nkasabov@otago.ac.nz,

Introduction

A decision support system is an information system that evaluates different options in order to help humans make a decision on a given problem, under given circumstances and constraints. A decision making system, in contrast, is a system that makes the final decision and takes actions, e.g. automated trading systems on the Internet, or systems that grant loans through electronic submissions.

Expert systems are information systems that contain expert knowledge on a particular problem and perform inference when new data is entered that may be partial or inexact. They provide a solution that is expected to be similar to the solution provided by experts in the field. An expert system usually consists of several parts: (1) a knowledge base, where the expert knowledge resides; (2) a data base, where historical and new data is stored; (3) an inference machine that provides different types of inferences; (4) an efficient interface to users; (5) an explanation module that would provide an explanation of *how* and *why* a certain decision was recommended by the system; (6) a module that learns and accumulates new knowledge, based on the system's operation and on new incoming data (see Duda and Shortliffe, 1983; Kasabov,1996).

In the rest of this article we will use the collective term *decision system* (DS) to refer to either a decision support system, or to a decision making system, or to the most sophisticated among them - expert systems.

The complexity and the dynamics of many real-world problems, such as decision making and prediction of economic and financial indexes, decision making in medicine, knowledge discovery in bioinformatics, adaptive intelligent control of industrial processes, on-line decision making based on large amount of continuous and dynamically changing information on the WWW, and many more, set certain requirements to the methods and the tools for building DS. Such requirements are:

- Deal with different types of data and knowledge.
- Adjust incrementally to dynamic changes in the operating environment, accommodating new data, introducing new variables and features if needed without the need to re-design or re-train the whole system; such adaptation may be required to be achieved in an on-line, real-time mode.
- Update the system's knowledge in a dynamic way.
- Explain in an appropriate way what knowledge is contained in the system, or what knowledge has been learned during the system's operation.

Different methods and approaches for building DS have been used so far that facilitate dealing with different types of data and problem knowledge in different modes, e.g. static versus dynamic data, fixed versus adaptive knowledge, off-line versus on-line mode, etc. These methods include: statistical methods, e.g. clustering, principal components, hidden Markov models; mathematical modelling; finite automata; methods of

artificial intelligence, such as logic systems, rule-based systems, case-based reasoning, different types of neural networks; hybrid methods that combine all the above (Duda and Shortliffe,1983; Kasabov,1996; also see relevant articles in this book).

Neural networks (NN), and connectionist-based systems have been applied so far as part of many DS in different ways, some of them listed below:

(a) NN are used as a low-level data processing and pattern matching modules, while rule-based modules are used for a higher-level decision making (Fu, 1989; Kasabov, 1990). NN can be part of a production system or of a first-order logic system for decision support (see Kasabov,1996).

(b) A fixed set of flat rule base is incorporated in a connectionist structure with a pre-defined built-in inference mechanism (Gallant, 1993). No learning is applied.

(c) All elements of a production system, e.g. rules, facts, inference machine, are represented in different connectionist modules (Touretzky and Hinton, 1989; Kasabov and Shishkov- see Kasabov, 1996). No learning is applied on the structure.

(c) Trainable knowledge-based neural networks (KBNN) that have fixed structures in terms of number of neurons and connections, are used to accommodate both knowledge (rules) and data (Fu, 1989;

Towel and Shavlik, 1993). Such KBNN are the fuzzy neural networks (Lin and Lee,1996).

(d) KBNN that develop (evolve) their structure, their functionality and their knowledge in time from incoming data, starting from an initial set of knowledge if such is available, are used for building adaptive, on-line learning DS (Kasabov, 1998,2001).

We will call a DS that has NN modules in its structure a *connectionist-based decision system (CBDS).* The sections below present different architectures and applications of such systems, starting from a general framework of a CBDS.

A general framework of a connectionist-based decision system

A framework of a CBDS is shown in fig.1. It consists of the following parts (fig.1):

- Pre-processing part (e.g. filtering of data, feature extraction).
- Neural network part - NN modules that are trained and incorporate knowledge.
- Higher-level knowledge-based part (e.g. rules for producing final decisions).
- Adaptation part – it evaluates the system decisions and makes changes to the system structure and functionality. Output error can be used to adjust relevant NN modules.

- Rule extraction, explanation part. This part uses both extracted rules from the NN modules and rules from the decision part to explain: (1) *what* the system 'knows' about the problem it is designed to solve; (2) *why* a particular decision for a concrete input vector has been made.

[Figure 1]

Different types of NNs, e.g. multiplayer perceptrons, self-organising maps, radial-basis functions, fuzzy neural networks, etc. (see relevant articles in this book) can be used as part of a CBDS, the most used being the KBNN as discussed in the next sections.

Knowledge-based neural networks

KBNN are pre-structured neural networks to allow for data and knowledge manipulation, including learning from data, rule insertion, rule extraction, adaptation and reasoning (Towel and Shavlik, 1993; Mitra and Hayshi,2000). KBNN have been developed either as a combination of symbolic AI systems and NN, or as a combination of fuzzy logic systems and NN, or as other hybrid systems.

The knowledge represented in KBNN is mainly in the form of different types of IF-THEN. Some of them are listed below:

(1) Simple propositional rules (e.g., IF x1 is A AND/OR x2 is B THEN y is C, where A,B and C are constants, variables, or symbols of true/false type) (Gallant, 1993).

(2) Propositional rules with certainty factors (e.g., IF x1 is A (CF1) AND x2 is B (CF2) THEN y is C (CFc)) (Fu,1989; Touretzky and Hinton, 1989).

(3) Zadeh-Mamdani fuzzy rules (e.g., IF x1 is A AND x2 is B THEN y is C, where A,B and C are fuzzy values represented by their membership functions) (Lin and Lee,1996).

(4) Takagi-Sugeno fuzzy rules (e.g., IF x1 is A AND x2 is B THEN y is a.x1 + b.x2 +c, where A,B and C are fuzzy values and a, b and c are constants) (Lin and Lee, 1996).

(5) Rules that have associated degrees of importance and certainty degrees (e.g., IF x1 is A (DI1) AND x2 is B (DI2) THEN y is C (CFc), where DI1 and DI2 represent the importance of each of the condition elements for the rule output, and the CFc represents the strength of this rule) (Kasabov,1996)

(6) Rules that represent associations of clusters of data from the problem space (e.g., Rule j: IF [an input vector x is in the input cluster defined by its center (x1 is Aj, to a membership degree of MD1j, AND x2 is Bj, to a membership degree of MD2j) and by its radius Rj-in] THEN [y is in the output cluster defined by its center (y is C, to a membership

degree of MDc) and by its radius Rj-out, with Nex(j) examples represented by this rule] (Kasabov, 1998).

(7) Temporal rules (e.g., IF x1 is present at a time moment t1 (with a certainty degree and/or importance factor of DI1) AND x2 is present at a time moment t2 (with a certainty degree/importance factor DI2) THEN y is C (CFc))

(8) Temporal, recurrent rules (e.g., IF x1 is A (DI1) AND x2 is B (DI2) AND y at the time moment (t-k) is C THEN y at a time moment (t+n) is D (CFc))

There are several methods for rule extraction from a KBNN. Three of them are explained below:

(1) Activating a trained KBNN on input data and observing the patterns of activation.

(2) Rule extraction through analysis of the connections in a trained KBNN.

(3) Methods that combine (1) and (2).

In terms of applying KBNN to make new inferences, there are three types of methods that can be used:

(1) Rules extracted from a KBNN are interpreted in another inference machine (e.g. fuzzy inference, production system).

(2) The rule base learning and reasoning modules constitute an integrated connectionist structure, so reasoning is performed in the connectionist structure.

(3) The two options from above are combined in one CBDS.

In terms of learning and rule extraction in a KBNN, we can differentiate the following cases: (1) off-line learning and static rule set extraction – first, learning is performed, and then rules are extracted, which is an one-off process; (2) on-line learning – rules can be extracted at any time of a continuous on-line learning process.

In terms of learning and optimisation in a KBNN, there are three cases: (1) globally optimised KBNN, i.e. for every new example all the connections change during learning; (2) locally optimised KBNN – for a new example only few connections change. Local optimisation in a KBNN would allow for adjusting the KBNN to new data through tuning a small number of elements, and also for extracting locally meaningful rules. The rules can be tuned as the system works.

Fuzzy neural networks and neuro-fuzzy inference systems

Fuzzy neural networks are NN that can be interpreted in terms of fuzzy rules, while neuro-fuzzy inference systems are fuzzy systems that can be structurally represented in a connectionist way (Lin and Lee, 1996; Kasabov, 1996). A review of neuro-fuzzy systems for rule generation is published by Mitra and Hayashi (2000).

Two examples of this class of KBNN are the fuzzy neural network FuNN (Kasabov,1996), and the evolving fuzzy neural network (EFuNN) (Kasabov,1998,2001). Both have similar architectures but different

functionality. The architecture consists of five layers of neurons and four layers of connections (see fig.2). The first layer of neurons receives the input information. The second layer calculates the fuzzy membership degrees to which the input values belong to predefined fuzzy membership functions, e.g. small, medium, or large. The MF can be kept fixed, or can change during training. The third layer of neurons represents associations between the input and the output variables, fuzzy rules. The fourth layer calculates the degrees to which output membership functions are matched by the input data, and the fifth layer does defuzzification and calculates values for the output variables.

While FuNN has fixed number of nodes and connections, EFuNN evolves both nodes and connections over time from the input data stream.

[Figure 2]

An example of a static connectionist-based DSS based on the FuNN architecture is given below.

Example: A CBDS for mortgage approval

A FuNN is trained on data for decision making on mortgage approval (the data is also available from the WWW site: http://divcom.otago.ac.nz/infosci/KEL/data). The following attributes are used: Input1: character (0- doubtful; 1 - good); Input2: total asset; Input3:

equity; Input4: mortgage loan; Input5: budget surplus; Input6: gross income; Input7: debt servicing ratio;Input8: term of loan; Output: decision (disapprove; approve). Two MFs are used both for the input and for the output variables. Ten FuNNs are trained with a modified backpropagation algorithm for 3500 epochs each, on 10 different sets of data, each of them containing both positive (applications are approved) and negative (applications are rejected) examples. Each trained NN is tested on another data of positive and negative examples. The average classification rate on the test data is 100% for the reject class, and 95% for the accept class when a threshold of 0.6 is used to distinguish the reject from the accept class activation of the output of the FuNN. Rules are extracted from the trained FuNN that explain the decision process.

Evolving connectionist systems

Evolving connectionist systems (ECOS) evolve their structure, their functionality, and their knowledge from incoming data, rather than having a pre-defined structure and a fixed set of rules. They learn and improve continuously over time as it is shown in the fig.3. An example of ECOS is the EFuNN architecture.

[Figure 3]

ECOS adapt to a changing environment – possibly in a real time; they can learn in a "lifelong" learning mode; they are able to explain what they have learned in terms of rules and other types of knowledge. An ECOS has a modular, 'open' structure evolving over time. Initially it is a mesh of nodes (neurons) with very little connections between them, pre-defined through *prior* knowledge or “genetic” information. An initial set of rules can be inserted in this structure. Gradually, through self-organisation, the system becomes more and more “wired”. The network learns different patterns (exemplars, prototypes) from the training examples. A node is created and designated to represent an individual example if this example is significantly different from the previous ones (with a level of differentiation set through dynamically changing parameters.

Along with growing, ECOS may have a pruning procedure defined. It allows for removing neurons and their corresponding connections that are not actively involved in the functioning of the ECOS thus making space for new input patterns.

Different modes of learning in ECOS are

(a) Active learning mode - learning is performed when a stimulus (input pattern) is presented and kept active.

(b) Passive (eco, sleep) learning mode - learning is performed when there is no input pattern presented at the input of the ECOS.

On-line connectionist-based decision systems

On-line CBDS are systems that make decisions and adjust their knowledge through an incremental, continuous learning from incoming data. Such systems, for example, learn the dynamic changes in a stock index, or adjust their knowledge to include new gene discoveries.

Example: On-line financial decision making based on on-line prediction of the MIB30 stock index

Fig.4 shows the use of EFuNN for on-line learning and prediction 5 days ahead the moving average values of the MIB30 stock index (the Milan stock index). EFuNN is trained on-line on consecutive data vectors. Inputs to the EFuNN are 5 day moving averages of the following variables: DJ (t), DJ(t-1), MBI30(t), MIB30(t-1), Euro/US$(t), Euro/US$(t-1).

[Figure 4]

At any time of the functioning of the EfuNN, rules can be extracted. The rules represent the current association between the input variables and the output variable as it is illustrated in the rule below.

Rule 2: IF [DJ(t) is (Medium 0.828) and DJ(t-1) is (Medium 0.840) and MIB30(t) is (Low 0.885) and MIB30(t-1) is (Low 0.887)] (receptive field= 0.808)
THEN MIB30(t+5) will be (Low 0.852) (accommodated training examples 182 out of 576)

Connectionist-based decision systems in economics and finance

Beltraffi et al, 1996 describe a variety of CBDS in the area of finance and economics. That include CBDS for solving problems such as: stock trading, portfolio decision, exchange rate prediction, fraud detection, credit scoring, and many more. Some CBDS contain expert rules and make decisions similar to the decision made by experts as illustrated in the next example.

Example: A CBDS for simulation and prediction of decisions made by the European Central Bank (ECB) on interest rate intervention (Rizzi et all, 2001)

The ECB meets regularly to make a decision on the interest rate for the European Union countries. Modelling this decision making process is extremely difficult as it requires both the comprehensive knowledge the ECB members have, and fast adaptation to new situations. The system developed by Rizzi et al (2001) consists of two parts. The first part has several NN that use 6 groups of economic indicators totaling in 17 variables and produces intermediate outputs that include the predicted values for some time series. These values, as well as other economic variables, are fed into a rule –based system that consitutes the second part of the system. The system finally suggests what ECB decisions on the interest rate might be expected at the next 3 consecutive meetings. Fig.5

shows some test results. The system learns and improves over time after each new data is entered.

[Figure 5]

Connectionist-based decision systems in bioinformatics

Processing large amount of data in many areas such as medicine, biochemistry, molecular biology, and making decisions based on this information, is a task where CBDS have been successfully applied. Baldi and Brunak (1998) have demonstrated the application of machine learning techniques, that include NN, for solving difficult problems, such as DNA promoter recognition, RNA splice junction identification (see the example below), secondary structure protein prediction, etc.

Example: CBDS for RNA splice junction identification

Figure 6 shows a simple 3-layer-perceptron NN for identifying a biological feature, such as a splice junction, from an input RNA sequence. The NN has 60 inputs, which is the length of the RNA window sequence, 5 intermediate neurons (nodes) and one output neuron to encode for the feature (1 – present, 0- not present). The data set used for training is available from: http://www.ics.uci.edu/~mlearn/MLRepository.html.

[Figure 6]

The predicted by the trained NN splice junction from new input data should be further analyzed before a final decision is made.

An EFuNN-based on-line training and prediction system for the same problem is available from http://divcom.otago.ac.nz/infosci/kel/CBIIS/GenIn/. The system can also extract rules, such as the rule shown below:

IF -----C--C-C-TCC-G--CTC-GT-C--GGTGAGTG--GGC---C---G-GG-C--CC-

THEN [Junction Exon-Intron] Receptive field =0.216, Examples in the rule 26 out of 1000

Future directions for connectionist-based decision systems

CBDS is a fast growing area. It includes both developing new connectionist techniques for learning, data mining and knowledge discovery, and developing important practical applications in finance and economics, bioinformatics and life sciences, process control and manufacturing, medicine and social sciences.

References

Baldi, P., Brunak, S., 2001, Bioinformatics – A Machine Learning Approach, MIT Press *

Beltraffi, A., Margarita, S., Terna, P., 1996, Neural networks for economics and financial modelling, Int. Thomson Computer Press

Duda, R.O., and Shortliffe, E.H.,1983, Expert systems research, Science, 220:261-268

Fu, L.M., 1989, Integration of neural heuristics into knowledge–based inference, Connection Science, 1:325-340 *

Gallant, S.I.,1993, Neural Network Learning and Expert Systems, MIT Press *

Kasabov, N.K., 1990, Hybrid connectionist rule-based systems, in: Jorrand, P., and Sgurev, V.(eds), Artificial Intelligence – Methodology, Systems, Applications, Amsterdam, North Holland, 227-235

Kasabov, N., 1998, ECOS: A framework for evolving connectionist systems and the ECO learning paradigm, in Proc. of ICONIP'98, Kitakyushu, Japan, IOS Press, vol.3, 1232-1235

Kasabov, N., 2001, Evolving fuzzy neural networks for adaptive, on-line, knowledge –based learning, in IEEE Transactions on Man, Machine and Cybernetics, B, to appear

Kasabov, N., 1996, Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering, CA, MA, MIT Press *

Lin, C.T. and C.S. G. Lee, 1996, Neuro Fuzzy Systems, Prentice Hall *

Mitra, S., and Hayshi, Y., May 2000, Neuro-fuzzy rule generation : survey in soft computing framework, IEEE Transactions on Neural Networks, vol.11, No.3

Rizzi, R., Bazzana, F., Kasabov, N., Fedrizzi, M. and Erzegovesi, 2001, A connectionist-based decision support system for modelling the interest rate intervention made by the European Central Bank, European Journal of Operation Research, to appear

Sima,J., and Cervenka, J., 2000, Neural knowledge processing in expert systems, in: I.Cloete and J.Zurada (eds) Knowledge-based neurocomputing, MIT Press, 419-466

Towell, G.G. and Shavlik, J.W., 1993, Extracting refined rules from knowledge-based neural networks, Machine Learning, 13:71-101

Touretzky, D., and Hinton, G, 1988, A distributed connectionist production system, Cognitive Science, 12:1423-1466

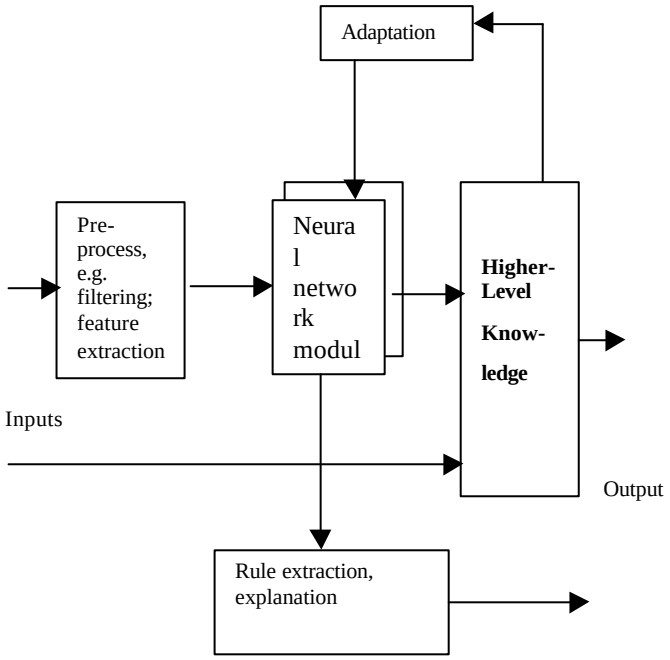

Figure 1. A framework of a connectionist-based decision system

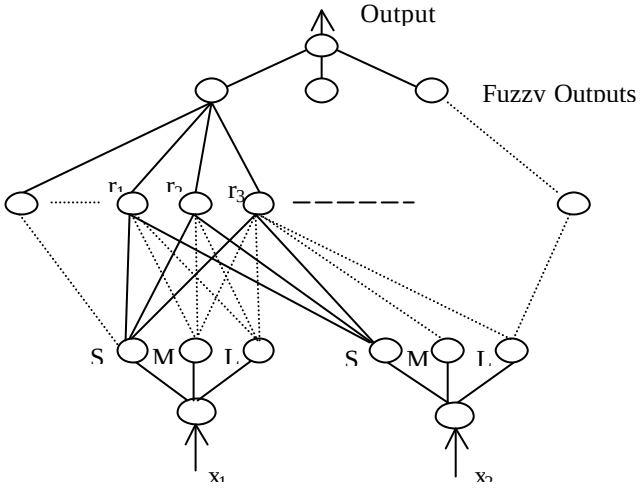

Figure 2. A simple neuro-fuzzy inference system - 2 inputs, 2 membership functions and 1 output. Rule node r1 can be represented as a rule: IF x1 is S and x2 is S THEN Output is S (certain statistical and linguistic parameters are attached)

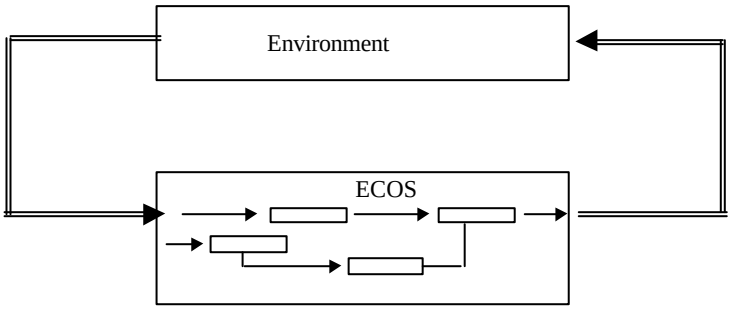

Figure 3. Evolving connectionist systems (ECOS) learn their structure and functionality through interaction with the environment

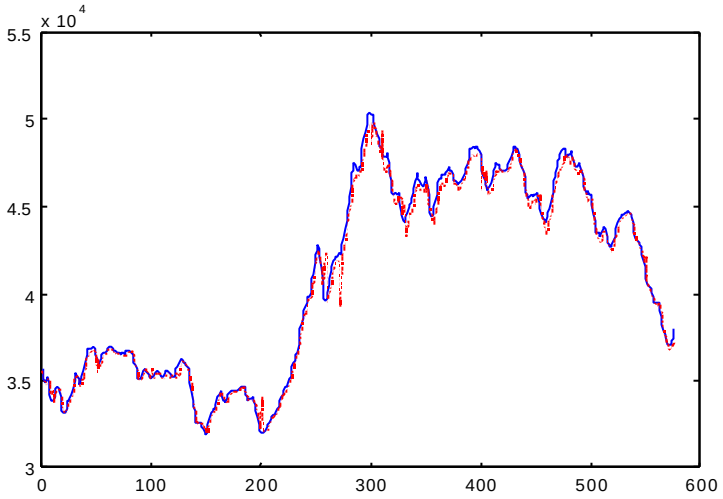

Figure 4. EFuNN: The process of on-line learning and prediction five days ahead of the MIB30 index - the desired versus the predicted by EFuNN in on-line mode output value for 576 consecutive days.

	Forecasts of the expert system					
Date (t)	Date (t+1)		Date (t+2)		Date (t+3)	
3 February 2000	16 March 2000		27 April 2000		8 June 2000	
0.25	0.25	(0.25)	0	(0.25)	0	(0.50)
16 March 2000	27 April 2000		8 June 2000		31 August 2000	
0.5	0.25	(0.25)	0.25	(0.50)	0.25	(0.25)
27 April 2000	8 June 2000		31 August 2000		19 October 2000	
0.5	0.25	(0.5)	0.25	(0.25)	0.25	(0.25)
8 June 2000	31 August 2000		19 October 2000		30 November 2000	
0.5	0.25	(0.25)	0.25	(0.25)	0.25	
31 August 2000	19 October 2000		30 November 2000		18 January 2001	
0.5	0.25	(0.25)	0.25		0.25	

Fig.5. The interest rate decision intervention by the European central Bank (ECB) at five ECB meetings, versus the calculated ahead intervention by the expert system. After each intervention value is made available, the expert system adjusts and then suggests what the ECB decision at three consecutive meetings might be. The real intervention values are the numbers in brackets.

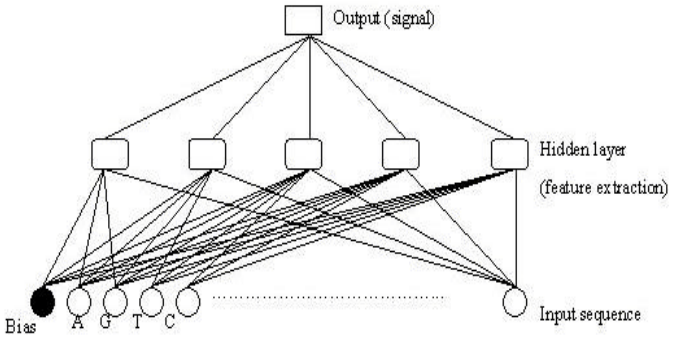

Fig.6. A neural network that takes an input vector of 60 nucleotides from an RNA and suggests the probability for having exon-intron, or intron-exon splice junction in the middle of the sequence, or no junction at all.