The Scientific Temper (2024) Vol. 15 (3): 2481-2488
E-ISSN: 2231-6396, ISSN: 0976-8653
Doi: 10.58414/SCIENTIFICTEMPER.2024.15.3.12
https://scientifictemper.com/

RESEARCH ARTICLE

Stochastic artificial neural network for magdm problem solving in intuitionistic fuzzy environment

P. J. Robinson*, **S. W. A. Prakash**

Abstract

In this work, we have presented the decision-making models based on ANN, which takes argument pairs of the intuitionistic fuzzy values and, defuzzifies the decision matrices and creates stochastic matrices for producing input for computations of ANN. Concepts from Stochastic processes, namely Markov chains and limiting distributions, are discussed in detail in this research work and have been applied for effective decision-making in complex situations. The numerical illustration provided in this work will be solved using the Markov chain models and some linear space techniques and applied in artificial neural network (ANN). A new algorithm is also developed for solving the MAGDM problems by applying the proposed methods. The numerical illustrations are solved with defuzzyfication operators and the results are recorded for effectiveness and comparisons are made with some existing methods. The new method proves to be more effective than the previous methods of ANN for MAGDM problems.

Keywords: Intuitionistic fuzzy theory, Markov chains, Aggregation operators, Weighted geometric operator, Artificial neural network.

Introduction

Intuitionistic fuzzy sets (IFS) is a growing area of research and were introduced by Atanassov, which also registers its importance in the field of artificial intelligence (AI) and machine learning (ML) in the present digital era. Recently, many researchers have concentrated on neural networks with various applications to the industry globally. An aggregation process with a variety of operators is mandatory for multiple attribute group decision-making (MAGDM) problems, which many authors provided. It is obvious that the fields of MAGDM and ANN are closely related, with both techniques including the processing of input data with weight vectors and sharing some common computational methodologies like sorting and filtering of the data set irrespective of their nature. Deriving an appropriate weighting vector for the MAGDM process or the ANN methodology for the available attributes is a herculean task, and in this paper, we have concentrated on the development of an apt weighting vector for the neural network process with the IFS data set. The methodology followed in this work is as follows: Initially, the decision matrices are converted to stochastic matrices and the weights are derived from arriving at the stationary distribution for the Markov chains and in turn, the weights are utilized for the computation of ANN. A special transformation from linear space techniques is also employed in the process of the conversion of the Stochastic matrices. The ANN is run with different levels of iterations and the results are compared for the effectiveness of the proposed methods. The ANN that is being offered in this study is solved using Python programming, and the method of employing MAGDM in conjunction with ANN is new to the domains of artificial intelligence and decision support systems Atanassov, K., (2013), Atanassov, K., (2023), Fullér, R. (2000), Hájek, P., & Olej, V. (2015), Heaton, J. (2015), Kuo, R.J., & Cheng, W. C. (2019), Leonishiya, A, (2023), Leonishiya, A., (2023), Robinson, J., (2024), Robinson, J., (2024), Zadeh, L.A. (1965), Zhao, J., (2016).

Markov Chains and Limiting Distribution

Consider a simple coin-tossing experiment repeated a number of times. The possible outcomes at each trial are two: head with probability, say p and tail with probability q, $p+q=1$. Let us denote head by 1 and tail by 0 and the random variable denoting the result of the n^{th} toss by X_n. Then for n=1,2,… $\Pr(X_n=1)=p$, $\Pr\{X_n=0\}=q$.

Department of Mathematics, Bishop Heber College, Affiliated to Bharathidasan University, Tiruchirappalli, India

***Corresponding Author:** P. J. Robinson, Department of Mathematics, Bishop Heber College, Affiliated to Bharathidasan University, Tiruchirappalli, India, E-Mail: johnrobinson.ma@bhc.edu.in

How to cite this article: Robinson, P. J., Prakash, S. W. A. (2024). Stochastic artificial neural network for magdm problem solving inintuitionisticfuzzyenvironment.TheScientificTemper,**15**(3):2481-2488.
Doi: 10.58414/SCIENTIFICTEMPER.2024.15.3.12

Source of support: Nil

Conflict of interest: None.

Received: 12/07/2024 **Accepted:** 11/08/2024 **Published** : 10/09/2024

Thus we have a sequence of the random variables $X_1, X_2, \ldots$. The trials are independent and the result of the n^{th} trial does not depend in any way on the previous trials numbered $1,2,\ldots,(n-1)$. The random variables are independent. Consider now the random variable given by the partial sum $S_n = X_1 + \ldots + X_n$. The sum S_n gives the accumulated number of heads in the first *n* trials and its possible values are $0,1,\ldots,n$. We have $S_{n+1} = S_n + X_{n+1}$. Given that $S_j = j\left(j = 0,1,\ldots,n\right)$, the random variables S_{n+1} can assume only two possible values: $S_{n+1} = j$ with probability *q* and $S_{n+1} = j+1$ with probability *p*; these probabilities are not at all affected by the values of the variables $S_1,\ldots,S_{n-1}$. Thus $\Pr\{S_{n+1} = j \mid S_n = j\} = p$; $\Pr\{S_{n+1} = j \mid S_n = j\} = q$

We have here an example of a Markov chain, a case of simple dependence that the outcome of $(n+1)^{th}$ trial depends directly on that of n^{th} trial and only on it. The conditional probability of S_{n+1} given S_n depends on the value of S_n and the manner in which the value of S_n was reached is of no consequence.

Definition 1

He stochastic process is $\{X_n, n = 0,1,2,\ldots\}$ is called a Markov chain, if, for $j,k,j_1,\ldots,j_{n-1} \in N$ (one any subset of *I*),

$$\Pr\{X_n = k \mid X_{n-1} = j, X_{n-2} = j_1, \ldots, X_0 = j_{n-1}\}$$
$$= \Pr\{X_n = k \mid X_{n-1} = j\} = p_{jk} \quad (1)$$

Whenever the first member is defined.

The outcomes are called the states of the Markov chain; if X_n has the outcome *j* $(i.e. N = j)$, the process is said to at the state *j* at n^{th} trial. To a pair of state (j,k) at the two successive trials (say, n^{th} and $(n+1)^{st}$ trials) there is an associated conditional probability p_{jk}. It is the probability of transition from the state *j* at n^{th} trial to the state *k* at $(n+1)^{st}$ trial. The transition probabilities p_{jk} are basic to the study of the study of the structure of the Markov chain.

The transition probability may or may not be independent of *n*. If the transition probability p_{jk} is independent of *n*, the Markov chain is said to be homogeneous (or to have stationary transition probabilities). If it is dependent on *n*, the chain is said to be non-homogeneous. Here we shall confine to homogeneous chains. The transition probability p_{jk} refer to the states (j,k) at two successive trials (say, n^{th} and $(n+1)^{st}$ trial); the transition is one-step and p_{jk} is called one-step (or unit step) transition probability. In the more general case, we are concerned with the pair of states (j,k) at two non-successive trials, say, state *j* at the n^{th} trial and state *k* at the $(n+m)^{th}$ trial. The corresponding transition probability is then called *m*-step transition probability denoted by $p_{jk}^{(m)}$, i.e. $p_{jk}^{(m)} = \Pr\{X_{n+m} = k \mid X_n = j\}$.

Transition Matrix (or Matrix of Transition Probabilities)

The transition probabilities p_{jk} satisfy $p_{jk} \ge 0, \ \sum_k p_{jk} = 1$ for all *j*. (2)

These probabilities may be matrix form

$$P = \begin{pmatrix} p_{11} & p_{12} & p_{13} & \cdots \\ p_{21} & p_{22} & p_{23} & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{pmatrix} \quad (3)$$

This is called the transition probability matrix or matrix of transition probabilities of the Markov chain. *P* is a stochastic matrix i.e. a square matrix with non-negative elements and unit row sums.

Stability of a Markov System

Definition

Stationary Distribution

Consider a Markov Chain with transition probabilities p_{jk} and the transition probability matrix $P = p_{jk}$. A probability distribution $\{v_j\}$ is called stationary (or invariant) for the given chain if $v_k = \sum_j v_j p_{jk}$ such that $v_j \ge 0, \sum_j v_j = 1$.

Now the decisive issue is whether a Markov system, regardless of the initial state *j*, reaches a steady or stable state after a large number of transitions or moves. Hence, under what conditions, if any, as n tends to infinity, $p_{jk}{}^{(n)}$ tends to a limit v_k independent of the initial state *j* (i.e., $P^{(n)}$ tends to a stochastic matrix whose rows are identical). This property of limiting the distribution of $p_{jk}{}^{(n)}$ being independent of the initial state *j* is called Ergodicity. When such a limit exists, the probabilities settle down and become stable. Then, the system shows some long-run regularity properties. This mentioned property of the stability of a Markov system is applied to the available decision matrices in order to derive the weights for the input vectors in the artificial neural network algorithm.

Magdm Problem Solving Using Stochastic Ann With Intuitionistic Fuzzy Sets

Pseudo-code for solving Stochastic-ANN

C_n : *n Matrix itemset of size k* x *m*

Input {*Intuitionistic Fuzzy Decision Matrices*}

A_n = *{Collection of n Matrices of size k };*

//* Defuzzyfication Phase*//

Compute {*Defuzzify the decision matrices using* median membership operator, $\frac{1-\mu+\gamma}{2}$}

For (n=1; $A_n \neq \emptyset$; n++) do begin

//* Stochastic matrix generation Phase*//

Generate {Stochastic matrices by normalizing the Defuzzified matrices, $\widetilde{M}_n$}

//* $\widetilde{M}_N$ *is the collection of Individual Preference IF-Decision Matrices* *//

Generate {Moderated decision matrices, $\widetilde{RM}_n$}

Generate {Convert the stochastic matrices into symmetric matrices using $\frac{M+M^T}{2}$, $\widetilde{RSM}_n$}

While $i \le m$ do {*Compute the Eigen values and Eigen vectors of* $\widetilde{RSM}_n$}

Generate {*Input for ANN with the collection of Eigen values,* X_i}

Generate {Target output for the Net using $d_i = 1 - \sqrt{\frac{\sum_{i=1}^{n}(S_i(x))^2}{n}}$}

//* Weight vector generation Phase*//

Generate {weight vector using limiting distribution of Markov chain for $\widetilde{RM}_n$, $\pi p = \pi$; $\sum_{k=0}^{n} \pi_n = 1$}

//* Learning Phase*//

Generate {*Weight Matrix by IF-Delta Rule*}

$net^n = (w^n)^T X_n$, $O^n = \frac{2}{1+\exp(-net^n)} - 1$, $f'(net^n) = \frac{1}{2}[1-(O^n)^2]$,

Update weights for next step {$W^{n+1} = c\,(d_n - O^n) f'(net^n) X_n + W^n$}

Continue the weight updation until the error is minimized to a desired level

//*Activation function*//

Fix {*The Threshold Value-Binary Step Function*}

While *Activated values* ≥ *Threshold* do

Generate {*Binary Matrix for final Decision with values exceeding the Threshold* }

Output{*Best Alternative(s) to be chosen*}

{The final decision variable can be converted into crisp variable and computations can be performed}.

End

Numerical Illustration

Consider the numerical illustration with the following decision matrices as in [10]. The problem is to identify the best alternative out of the available alternatives (5-row wise). The computations are presented as below:

$$\tilde{R}_1 = \begin{pmatrix} (0.4,0.3)(0.5,0.2)(0.2,0.5)(0.1,0.6) \\ (0.6,0.2)(0.6,0.1)(0.6,0.1)(0.3,0.4) \\ (0.5,0.3)(0.4,0.3)(0.4,0.2)(0.5,0.2) \\ (0.7,0.1)(0.5,0.2)(0.2,0.3)(0.1,0.5) \\ (0.5,0.1)(0.3,0.2)(0.6,0.2)(0.4,0.2) \end{pmatrix}$$

$$\tilde{R}_2 = \begin{pmatrix} (0.5,0.4)(0.6,0.3)(0.3,0.6)(0.2,0.7) \\ (0.7,0.3)(0.7,0.2)(0.7,0.2)(0.4,0.5) \\ (0.6,0.4)(0.5,0.4)(0.5,0.3)(0.6,0.3) \\ (0.8,0.1)(0.6,0.3)(0.3,0.4)(0.2,0.6) \\ (0.6,0.2)(0.4,0.3)(0.7,0.1)(0.5,0.3) \end{pmatrix}$$

$$\tilde{R}_3 = \begin{pmatrix} (0.4,0.5)(0.5,0.4)(0.2,0.7)(0.1,0.8) \\ (0.6,0.4)(0.6,0.3)(0.6,0.3)(0.3,0.6) \\ (0.5,0.5)(0.4,0.5)(0.4,0.4)(0.5,0.4) \\ (0.7,0.2)(0.5,0.4)(0.2,0.5)(0.1,0.7) \\ (0.5,0.3)(0.3,0.4)(0.6,0.2)(0.4,0.4) \end{pmatrix}$$

The decision matrices are given as follows, where one additional zero column is added to every matrix in order to transform the matrices into square matrices and then defuzzyfying the entries of the decision matrices using median membership operator $= \frac{1-\mu+\gamma}{2}$, we obtain the following:

$$\tilde{R}_1 = \begin{bmatrix} 0.55 & 0.65 & 0.35 & 0.25 & 0.5 \\ 0.7 & 0.75 & 0.75 & 0.45 & 0.5 \\ 0.6 & 0.55 & 0.6 & 0.65 & 0.5 \\ 0.8 & 0.65 & 0.45 & 0.3 & 0.5 \\ 0.7 & 0.55 & 0.7 & 0.6 & 0.5 \end{bmatrix};$$

$$\tilde{R}_2 = \begin{bmatrix} 0.55 & 0.65 & 0.35 & 0.25 & 0.5 \\ 0.7 & 0.75 & 0.75 & 0.45 & 0.5 \\ 0.6 & 0.55 & 0.6 & 0.65 & 0.5 \\ 0.8 & 0.65 & 0.45 & 0.3 & 0.5 \\ 0.7 & 0.55 & 0.8 & 0.6 & 0.5 \end{bmatrix};$$

$$\tilde{R}_3 = \begin{bmatrix} 0.45 & 0.55 & 0.25 & 0.15 & 0.5 \\ 0.6 & 0.65 & 0.65 & 0.35 & 0.5 \\ 0.5 & 0.45 & 0.5 & 0.55 & 0.5 \\ 0.75 & 0.55 & 0.35 & 0.2 & 0.5 \\ 0.6 & 0.45 & 0.7 & 0.5 & 0.5 \end{bmatrix}.$$

Normalize the values into entries that will form Stochastic matrices.

Normalized matrices are given as follows:

$$\tilde{M}_1 = \begin{bmatrix} 0.239130 & 0.282608 & 0.152173 & 0.108695 & 0.217391 \\ 0.222222 & 0.238095 & 0.238095 & 0.142857 & 0.158730 \\ 0.206896 & 0.189655 & 0.206896 & 0.224137 & 0.172413 \\ 0.296296 & 0.240746 & 0.166666 & 0.111111 & 0.185185 \\ 0.229508 & 0.180327 & 0.229508 & 0.196721 & 0.163934 \end{bmatrix}$$

$$\tilde{M}_2 = \begin{bmatrix} 0.239130 & 0.282608 & 0.152173 & 0.108695 & 0.217391 \\ 0.222222 & 0.238095 & 0.238095 & 0.142857 & 0.158730 \\ 0.206896 & 0.189655 & 0.206896 & 0.224137 & 0.172413 \\ 0.210909 & 0.236363 & 0.163636 & 0.236363 & 0.181818 \\ 0.222222 & 0.174603 & 0.253968 & 0.190476 & 0.158730 \end{bmatrix}$$

$$\tilde{M}_3 = \begin{bmatrix} 0.236842 & 0.289473 & 0.131578 & 0.078947 & 0.263157 \\ 0.218181 & 0.236363 & 0.236363 & 0.127272 & 0.181818 \\ 0.200000 & 0.180000 & 0.200000 & 0.220000 & 0.200000 \\ 0.319148 & 0.234042 & 0.148936 & 0.085106 & 0.212765 \\ 0.218181 & 0.163636 & 0.254545 & 0.181818 & 0.181818 \end{bmatrix}$$

To get the row sum equal to one, distribute the values of the maximum entries with the entries with the least values. Now, we get the moderated values for the decision matrix as follows:

$$\widetilde{RM}_1 = \begin{bmatrix} 0.239130 & 0.282608 & 0.152173 & 0.108698 & 0.217391 \\ 0.222222 & 0.238095 & 0.238095 & 0.142858 & 0.158730 \\ 0.206896 & 0.189655 & 0.206896 & 0.224137 & 0.172416 \\ 0.296292 & 0.240746 & 0.166666 & 0.111111 & 0.185185 \\ 0.229508 & 0.180327 & 0.229508 & 0.196721 & 0.163936 \end{bmatrix}$$

$$\widetilde{RM}_2 = \begin{bmatrix} 0.239130 & 0.282608 & 0.152173 & 0.108698 & 0.217391 \\ 0.222222 & 0.238095 & 0.238095 & 0.142858 & 0.158730 \\ 0.206896 & 0.189655 & 0.206896 & 0.224137 & 0.172416 \\ 0.210909 & 0.207274 & 0.163636 & 0.236363 & 0.181818 \\ 0.222222 & 0.174603 & 0.253968 & 0.190476 & 0.158731 \end{bmatrix}$$

$$\widetilde{RM}_3 = \begin{bmatrix} 0.236842 & 0.289473 & 0.131578 & 0.07895 & 0.263157 \\ 0.218181 & 0.236363 & 0.236363 & 0.127275 & 0.181818 \\ 0.200000 & 0.180000 & 0.200000 & 0.220000 & 0.200000 \\ 0.319148 & 0.234042 & 0.148936 & 0.085109 & 0.212765 \\ 0.218181 & 0.163638 & 0.254545 & 0.181818 & 0.181818 \end{bmatrix}$$

Convert the stochastic matrices into symmetric matrices using $\frac{M+M^T}{2}$, and find the Eigen values of individual column matrices.

For $\widetilde{RM}_1$, $R\tilde{M}_2$, $\widetilde{RM}_3$ we have the transformed symmetric matrices as follows:

$$\widetilde{RSM}_1 = \begin{bmatrix} 0.239 & 0.2525 & 0.1795 & 0.2025 & 0.2235 \\ 0.2525 & 0.238 & 0.214 & 0.192 & 0.1695 \\ 0.1795 & 0.214 & 0.207 & 0.1955 & 0.201 \\ 0.2025 & 0.192 & 0.1955 & 0.111 & 0.191 \\ 0.2235 & 0.1695 & 0.201 & 0.191 & 0.164 \end{bmatrix}$$

$$\widetilde{RSM}_2 = \begin{bmatrix} 0.239 & 0.2525 & 0.1795 & 0.16 & 0.22 \\ 0.2525 & 0.238 & 0.214 & 0.175 & 0.1665 \\ 0.1795 & 0.214 & 0.207 & 0.194 & 0.213 \\ 0.16 & 0.175 & 0.194 & 0.2365 & 0.186 \\ 0.22 & 0.1665 & 0.213 & 0.186 & 0.1585 \end{bmatrix}$$

$$\widetilde{RSM}_3 = \begin{bmatrix} 0.237 & 0.254 & 0.166 & 0.199 & 0.2405 \\ 0.254 & 0.2365 & 0.208 & 0.1805 & 0.1725 \\ 0.166 & 0.208 & 0.2 & 0.1845 & 0.2275 \\ 0.199 & 0.1805 & 0.1845 & 0.085 & 0.1975 \\ 0.2405 & 0.1725 & 0.2275 & 0.1975 & 0.182 \end{bmatrix}$$

Using these corresponding symmetric matrices $\widetilde{RSM}_1$, $\widetilde{RSM}_2$, $\widetilde{RSM}_3$, find the Eigen values and Eigen vectors separately as follows:

Eigen Values and vectors for $\widetilde{RSM}_1$:

$$\lambda_1 = -0.065;\ v_1 = \begin{pmatrix} -0.335 \\ 0.534 \\ -0.075 \\ -1.200 \\ 1 \end{pmatrix};$$

$$\lambda_2 = -0.060;\ v_2 = \begin{pmatrix} -1.865 \\ 1.002 \\ -1.697 \\ 1.906 \\ 1 \end{pmatrix}$$

$$\lambda_3 = 0.029\ ;\ v_3 = \begin{pmatrix} 0.675 \\ -1.142 \\ -0.629 \\ 0.176 \\ 1 \end{pmatrix};$$

$$\lambda_4 = 0.050;\ v_4 = \begin{pmatrix} -1.988 \\ -1.567 \\ 2.452 \\ 0.537 \\ 1 \end{pmatrix};$$

$$\lambda_5 = 1.006;\ v_5 = \begin{pmatrix} 1.159 \\ 1.130 \\ 1.047 \\ 0.947 \\ 1 \end{pmatrix}.$$

Eigen Values for $\widetilde{RSM}_2$:

$$\lambda_1 = -0.071;\ v_1 = \begin{pmatrix} 0.754 \\ 0.607 \\ -0.638 \\ -0.156 \\ 1 \end{pmatrix};$$

$$\lambda_2 = 0.021\ ;\ v_2 = \begin{pmatrix} 2.011 \\ -1.548 \\ -2.583 \\ 1.227 \\ 1 \end{pmatrix};$$

$$\lambda_3 = 0.024\ ;\ v_3 = \begin{pmatrix} -5.043 \\ -3.972 \\ 2.088 \\ 6.831 \\ 1 \end{pmatrix};$$

$$\lambda_4 = 0.103;\ v_4 = \begin{pmatrix} -5.043 \\ -3.972 \\ 2.088 \\ 6.831 \\ 1 \end{pmatrix};$$

$$\lambda_5 = 1.002;\ v_5 = \begin{pmatrix} 1.117 \\ 1.114 \\ 1.064 \\ 1.000 \\ 1 \end{pmatrix}.$$

Eigen Values for $\widetilde{RSM}_3$:

$$\lambda_1 = -0.086;\ v_1 = \begin{pmatrix} -0.774 \\ 0.600 \\ -0.691 \\ -0.142 \\ 1 \end{pmatrix};$$

$$\lambda_2 = -0.075;\ v_2 = \begin{pmatrix} 1.138 \\ 0.152 \\ 1.156 \\ -4.152 \\ 1 \end{pmatrix};$$

$$\lambda_3 = 0.029\ ;\ v_3 = \begin{pmatrix} 1.001 \\ -1.352 \\ -0.893 \\ 0.217 \\ 1 \end{pmatrix};$$

$$\lambda_4 = 0.065;\ v_4 = \begin{pmatrix} -1.378 \\ -1.378 \\ 1.732 \\ 0.295 \\ 1 \end{pmatrix}$$

$$\lambda_5 = 1.007;\ v_5 = \begin{pmatrix} 1.081 \\ 1.038 \\ 0.965 \\ 0.844 \\ 1 \end{pmatrix}.$$

Consider the input values as X_1, X_2, X_3 corresponding to the eigenvalues of $\widetilde{RSM}_1$, $\widetilde{RSM}_2$, $\widetilde{RSM}_3$ respectively:

$$X_1 = \begin{bmatrix} -0.065 \\ -0.060 \\ 0.029 \\ 0.050 \\ 1.006 \end{bmatrix};\quad X_2 = \begin{bmatrix} -0.071 \\ 0.021 \\ 0.024 \\ 0.103 \\ 1.002 \end{bmatrix};\quad X_3 = \begin{bmatrix} -0.086 \\ -0.075 \\ 0.029 \\ 0.065 \\ 1.007 \end{bmatrix}.$$

These above three column matrices will represent the input vectors respectively for the artificial neural network. Now, in this work is proposed a novel formula to find the d values for the target output:

$$d_i = 1 - \sqrt{\frac{\sum_{i=1}^{n} (S_i(x))^2}{n}}. \tag{4}$$

For the input vector $X_1 = \begin{bmatrix} -0.065 \\ -0.060 \\ 0.029 \\ 0.050 \\ 1.006 \end{bmatrix}$,

$d_1 = 1 - \sqrt{\frac{(-0.065)^2+(-0.060)^2+(0.029)^2+(0.050)^2+(1.006)^2}{5}}$,

$d_1 = 0.54762803$,

$d_2 = 1 - \sqrt{\frac{(-0.071)^2+(0.021)^2+(0.024)^2+(0.103)^2+(1.002)^2}{5}}$,

$d_2 = 0.548187871$,

$d_3 = 1 - \sqrt{\frac{(-0.086)^2+(-0.075)^2+(0.029)^2+(0.065)^2+(1.007)^2}{5}}$,

$d_3 = 0.545657398$.

Apply the Learning rule (Delta learning Rule) for the defuzzified individual column matrices to obtain the weight vector. Consider the stochastic matrices $\widetilde{RM}_1, \widetilde{RM}_2, \widetilde{RM}_3$, which will be utilized for the weight determination for ANN. To compute the weights, we can calculate the limiting distribution for the corresponding stochastic matrices as follows:

Utilize the system of linear equations:

$\pi p = \pi$, (5)

$\sum_{k=0}^{n} \pi_n = 1$. (6)

Such that, for $\widetilde{RM}_1$ we have the system of equations as follows:

$0.239130\pi_1+0.282608\pi_2+0.152173\pi_3+0.108698\pi_4+0.217391\pi_5=\pi_1$

$0.222222\pi_1+0.238095\pi_2+0.258095\pi_3+0.142858\pi_4+0.158730\pi_5=\pi_2$

$0.206896\pi_1+0.189655\pi_2+0.206896\pi_3+0.224137\pi_4+0.172416\pi_5=\pi_3$

$0.296292\pi_1+0.240746\pi_2+0.166666\pi_3+0.111111\pi_4+0.185185\pi_5=\pi_4$

$0.229508\pi_1+0.180327\pi_2+0.229508\pi_3+0.196721\pi_4+0.163936\pi_5=\pi_5$

$\pi_1+\pi_2+\pi_3+\pi_4+\pi_5 = 1$

For $\widetilde{RM}_2$ we have the system of equations as follows:

$0.239130\pi_1+0.282608\pi_2+0.152173\pi_3+0.108698\pi_4+0.217391\pi_5=\pi_1$

$0.222222\pi_1+0.238095\pi_2+0.258095\pi_3+0.142858\pi_4+0.158730\pi_5=\pi_2$

$0.206896\pi_1+0.189655\pi_2+0.206896\pi_3+0.224137\pi_4+0.172416\pi_5=\pi_3$

$0.210909\pi_1+0.207274\pi_2+0.163636\pi_3+0.236363\pi_4+0.181818\pi_5=\pi_4$

$0.222222\pi_1+0.174603\pi_2+0.253968\pi_3+0.190476\pi_4+0.158731\pi_5=\pi_5$

$\pi_1+\pi_2+\pi_3+\pi_4+\pi_5=1$

For $\widetilde{RM}_3$ we have the system of equations as follows:

$0.236842\pi_1+0.289473\pi_2+0.131578\pi_3+0.078950\pi_4+0.263157\pi_5=\pi_1$

$0.218181\pi_1+0.236363\pi_2+0.236363\pi_3+0.127275\pi_4+0.181818\pi_5=\pi_2$

$0.200000\pi_1+0.180000\pi_2+0.200000\pi_3+0.220000\pi_4+0.200000\pi_5=\pi_3$

$0.319148\pi_1+0.234042\pi_2+0.148936\pi_3+0.085109\pi_4+0.212765\pi_5=\pi_4$

$0.218181\pi_1+0.163638\pi_2+0.254545\pi_3+0.181818\pi_4+0.181818\pi_5=\pi_5$

$\pi_1+\pi_2+\pi_3+\pi_4+\pi_5=1$

These are stochastic equations of the corresponding three matrices $\widetilde{RM}_1, \widetilde{RM}_2, \widetilde{RM}_3$.

Solving the above set of equations separately using the TORA package, we get the solutions that yield the stationary and limiting distribution as follows:

From the weight calculation, we take the weight of the stochastic matrix of $\widetilde{RM}_1$, since the solution of the other two systems yields the solution as (0.2 0.2 0.2 0.2 0.2), which may not be considered for the consensus process with all 5 equal values.

Hence, the weight vector is: $W=\begin{bmatrix}0.19986\\0.19997\\0.20006\\0.20123\\0.19888\end{bmatrix}$ (from $\widetilde{RM}_1$).

Hence $w^1 = \begin{pmatrix}0.19986\\0.19997\\0.20006\\0.20123\\0.19888\end{pmatrix}$

By applying Delta learning rule to these weights, we get:

When $d_1 = 0.54762803, d_2=0.548187871,$

$d_3 = 0.545657398$

$net^1 = (w^1)^T X_1$

$= (0.19986 \quad 0.19997 \quad 0.20006 \quad 0.20123 \quad 0.19888)$

$\begin{pmatrix}-0.065\\-0.060\\0.029\\0.050\\1.006\end{pmatrix}$ $net^1 = 0.19094742$

$O^1 = \frac{2}{1+\exp(-0.19094742)} - 1$

$= 1.09518467555 - 1$

$O^1 = 0.09518467555$

$f'(net^1) = \frac{1}{2}[1-(O^1)^2]$

$= \frac{1}{2}[1-(0.09518467555)^2]$

$= \frac{1}{2}[1-0.0906012246]$

$= 0.4546993877$

$W^2 = c\,(d_1 - O^1) f'(net^1) X_1 + W^1$

$$= (0.1)(0.54762803 - 0.09518467555)(0.4546993877)\begin{pmatrix}-0.065\\-0.060\\0.029\\0.050\\1.006\end{pmatrix} + \begin{pmatrix}0.19986\\0.19997\\0.20006\\0.20123\\0.19888\end{pmatrix}$$

$$= (0.020572571)\begin{pmatrix}-0.065\\-0.060\\0.029\\0.050\\1.006\end{pmatrix} + \begin{pmatrix}0.19986\\0.19997\\0.20006\\0.20123\\0.19888\end{pmatrix}$$

$$= \begin{pmatrix}-0.001337217\\-0.001234354\\0.000596604\\0.001028628\\0.020696006\end{pmatrix} + \begin{pmatrix}0.19986\\0.19997\\0.20006\\0.20123\\0.19888\end{pmatrix}$$

$$W^2 = \begin{pmatrix}0.198522783\\0.198735646\\0.200656604\\0.202258628\\0.219576006\end{pmatrix}$$

$$net^2 = (w^2)^T X_2$$

$$= (0.198522783 \;\; 0.198735646 \;\; 0.200656604 \;\; 0.202258628 \;\; 0.219576006)\begin{pmatrix}-0.071\\0.021\\0.024\\0.103\\1.002\end{pmatrix}$$

$$= 0.235741884$$

$$O^2 = \frac{2}{1+\exp(-0.235741884)} - 1$$

$$O^2 = 0.1173280764$$

$$f'(net^2) = \frac{1}{2}[1 - (O^2)^2]$$

$$= \frac{1}{2}[1 - (-0.1173280764)^2]$$

$$= 0.4931170612$$

$$W^3 = c(d_2 - o^2)f'(net^2)X_2 + W^2$$

$$= (0.1)(0.548187871 - 0.1173280764)(0.4931170612)\begin{pmatrix}-0.071\\0.021\\0.024\\0.103\\1.002\end{pmatrix} + \begin{pmatrix}0.198522783\\0.198735646\\0.200656604\\0.202258628\\0.219576006\end{pmatrix}$$

$$= (0.048162)\begin{pmatrix}-0.071\\0.021\\0.024\\0.103\\1.002\end{pmatrix} + \begin{pmatrix}0.198522783\\0.198735646\\0.200656604\\0.202258628\\0.219576006\end{pmatrix}$$

$$= (0.021246431)\begin{pmatrix}-0.001759\\-0.001977\\-0.001409\\0.001236\\0.002709\end{pmatrix} + \begin{pmatrix}0.207529\\0.206788\\0.207015\\0.209810\\0.208998\end{pmatrix}$$

$$= \begin{pmatrix}0.197014286\\0.199181821\\0.201166518\\0.20444701\\0.240864929\end{pmatrix}$$

$$net^3 = (w^3)^T X_3$$

$$= (0.197014286 \;\; 0.199181821 \;\; 0.201166518 \;\; 0.20444701 \;\; 0.240864929)\begin{pmatrix}-0.086\\-0.075\\0.029\\0.065\\1.007\end{pmatrix}$$

$$= 0.229792003$$

$$O^3 = \frac{2}{1+\exp(-0.229792003)} - 1$$

$$O^3 = 0.1143930728$$

$$f'(net^3) = \frac{1}{2}[1 - (O^3)^2]$$

$$= \frac{1}{2}[1 - (-0.1143930728)^2]$$

$$= 0.493457112$$

$$W^4 = c\,(d_3 - o^3)f'(net^3)X_3 + W^3$$

$$= (0.1)(0.545657398 - 0.1143930728)(0.493457112)\begin{pmatrix}-0.086\\-0.075\\0.029\\0.065\\1.007\end{pmatrix} + \begin{pmatrix}0.197014286\\0.199181821\\0.201166518\\0.20444701\\0.240864929\end{pmatrix}$$

$$= (0.021281044)\begin{pmatrix}-0.086\\-0.075\\0.029\\0.065\\1.007\end{pmatrix} + \begin{pmatrix}0.197014286\\0.199181821\\0.201166518\\0.20444701\\0.240864929\end{pmatrix}$$

$$= \begin{pmatrix}-0.001830169\\-0.001596078\\0.00061715\\0.001383267\\0.021430011\end{pmatrix} + \begin{pmatrix}0.197014286\\0.199181821\\0.201166518\\0.20444701\\0.240864929\end{pmatrix}$$

$$= \begin{pmatrix}0.195184117\\0.197585743\\0.201783668\\0.205830277\\0.26229494\end{pmatrix}$$

Step 8

Utilize the activation function and identify the best values.

Table 1: Selection of the best alternative from different iterations

S. No.	*Iteration 50*	*Iteration 75*	*Iteration 100*
Values of the Alternatives	A1: 0.1298774281 A2: 0.1629831011 A3: 0.2260207730 A4: 0.2693719259 A5: 1.1511414414	A1: 0.1266662881 A2: 0.1609420101 A3: 0.2272332830 A4: 0.2723026033 A5: 1.1948935760	A1: 0.1257777986 A2: 0.1601059646 A3: 0.2275861055 A4: 0.2729591035 A5: 1.2070637293
Ranking of Alternatives after Activation phase (Descending Order)	A5 A4 A3 A2 A1	A5 A4 A3 A2 A1	A5 A4 A3 A2 A1
	A_5 is the best alternative from 50, 75 and 100th iterations		

Table 2: Comparison of the proposed method with some existing methods of ANN and MAGDM

Type of operator	*IFWG*	*IFWA*	*OWA*	*OWG*	*G-OWA*	*G-OWG*
Inputs as Orthogonal Vectors [10]	A_4, A_5	A_1, A_4	A_1, A_2, A_3	A_2, A_3	A_1, A_2, A_3	A_2, A_3, A_5
Inputs as Orthonormal Vectors [10]	A_4, A_5	A_1, A_2, A_3, A_4	A_1, A_2, A_3	A_2, A_3	A_1, A_2, A_3	A_5
Traditional MAGDM Methods [7,8,9]	METHOD – 1: Score & Accuracy Functions $A_5 > A_2 > A_3 > A_1 > A_4$ Most desirable alternative is A_5.		METHOD – 2: Hamming Distance Function Excluding Intuitionistic Degree $A_1 > A_4 > A_3 > A_2 > A_5$ Most desirable alternative is A_5.		METHOD – 3: Hamming Distance Function Including Intuitionistic Degree $A_1 > A_4 > A_2 > A_3 > A_5$ Most desirable alternative is A_5.	

The computations and comparisons are recorded in the following Tables 1 and 2.

Conclusion

In this work, we have presented the MAGDM problems based on solving them by stochastic artificial neural network methodologies. A combination of some linear space techniques and stochastic models with Markov chain applications has been incorporated in deriving the input for the ANN as well as the weight determination for the same. Finally, a numerical illustration has been given with three different instances of ranking at different iteration levels to show the developed method. The comparison could be extended with a greater number of iterations, which would be out of scope for the current problem of decision-making. A new Algorithm was proposed for the ANN method of solving the MAGDM problems. The new method proves to be more effective than the previous methods since it uses eigenvalues for input creation. Methods based on orthonormal and orthogonal vectors are also employed to create the input for ANN. Numerical illustrations were given for the effectiveness of the proposed methods.

References

Atanassov, K., Sotirov, S., & Angelova, N. (2020). Intuitionistic fuzzy neural networks with interval valued intuitionistic fuzzy conditions. *Studies in Computational Intelligence*, *862*, 99-106. https://doi.org/10.1007/978-3-030-35445-9_9

Atanassov, K., Sotirov, S., & Pencheva, T. (2023). Intuitionistic fuzzy deep neural network. *Mathematics*, *11*(716), 1-14. https://doi.org/10.3390/math11030716

Fullér, R. (2000). Fuzzy neural networks. In *Introduction to Neuro-Fuzzy Systems* (pp. 171–254). https://doi.org/10.1007/978-3-7908-1852-9_3

Hájek, P., & Olej, V. (2015). Intuitionistic fuzzy neural network: The case of credit scoring using text information. In L. Iliadis & C. Jayne (Eds.), *Engineering Applications of Neural Networks. EANN 2015*. Communications in Computer and Information Science, vol 517. Springer, Cham. https://doi.org/10.1007/978-3-319-23983-5_31

Heaton, J. (2015). *Artificial intelligence for humans, volume 3: Deep learning & neural networks*.

Kuo, R. J., & Cheng, W. C. (2019). An intuitionistic fuzzy neural network with Gaussian membership function. *Journal of Intelligent & Fuzzy Systems*, *36*(6), 6731-6741. https://doi.org/10.3233/IJFS-18998

Leonishiya, A., & Robinson, J. P. (2023). A fully linguistic intuitionistic fuzzy artificial neural network model for decision support systems. *Indian Journal of Science and Technology*, *16*(SP4), 29-36. https://doi.org/10.17485/IJST/v16iSP4.ICAMS136

Leonishiya, A., & Robinson, J. P. (2023). Varieties of linguistic intuitionistic fuzzy distance measures for linguistic intuitionistic fuzzy TOPSIS method. *Indian Journal of Science and Technology*, *16*(33), 2653-2662. https://doi.org/10.17485/IJST/v16i33.640-icrsms

Poornappriya, T. S., & Durairaj, M. (2019). High relevancy low redundancy vague set based feature selection method for telecom dataset. *Journal of Intelligent & Fuzzy Systems*, *37*(5), 6743-6760.

Priyadharshini, D., Gopinath, R., & Poornappriya, T. S. (2020). A fuzzy MCDM approach for measuring the business impact of employee selection. *International Journal of Management (IJM)*, *11*(7), 1769-1775

Robinson, J. P., & Leonishiya, A. (2024). Application of varieties of learning rules in intuitionistic fuzzy artificial neural network. In *Lecture Notes in Networks & Systems-831, Vol-1 (Machine Intelligence for Research & Innovations, MAITRI-2023, Springer)*.

Robinson, J. P., & Saranraj, A. (2024). Intuitionistic fuzzy Gram-Schmidt orthogonalized artificial neural network for solving MAGDM problems. *Indian Journal of Science and Technology*, *17*(24), 2529-2537. https://doi.org/10.17485/IJST/v17i24.1386

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, *8*, 338-356.

Zhao, J., Lin, L.-Y., & Lin, C.-M. (2016). A general fuzzy cerebellar model neural network multidimensional classifier using intuitionistic fuzzy sets for medical identification. *Computational Intelligence and Neuroscience*, *2016*, 1-9. https://doi.org/10.1155/2016/8073279