Neural Network Fuzzy Learning Vector Quantization (FLVQ) to Identify Probability Distributions

Warsono*, Gerry Alfa D, Dian Kurniasari, and Mustofa Usman
Department of Mathematics, Faculty of Sciences and Mathematics, University of Lampung, Indonesia

Abstract
A Statistical model is built based on a probability distribution. Classically, probability distribution is identified by some methods for example by using Chi-square goodness of fits, by using graph, by nonparametric goodness of fits test, and by using normal plot to test the normality. The aim of this study is going to discuss the applications of Fuzzy Learning Vector Quantization (FLVQ) model to identify some probability distributions; this model is a merger between neural network and fuzzy set. The results from the application of this FLVQ through a simulation to identify the probability distributions are very good and can be implemented to a real data.

Keywords:
neural network, fuzzy set, FLVQ, codebook vector, goodness of fits test

1. Introduction

In statistical data analysis to know a probability distribution from data is very important. It is because if we want to be built a model from the data and want to do analysis and conduct a statistical test, then it is necessary known the probability distribution of the data so that the assumption of the theory used are attained and whether it is fit with model probability distribution assumption or statistical test. Classical test to know a probability distribution of a data is used nonparametric goodness of fit test such as chi-square or Kolmogorov Smirnov test. However, this classical test sometime lacks accuracy to identify probability distribution, if the sample size is small [1]. The Process of identification of a probability distribution is belongs to the field of "pattern recognition". Neural networks have been widely used in pattern recognition problems and showed they reliability in pattern recognition. Su and Chou [1] shows that neural networks can become a good tool to identify a probability distribution. According to Su and Chou [1], LVQ (Learning Vector Quantization) shows higher accuracy level than nonparametric statistical test to identify probability distribution. In their study, they used eight probability distributions to build neural network model, that is, normal distribution, exponential distribution, Weibull distribution, uniform distribution, chi-square distribution, t distribution, F distributions and lognormal distribution. Nowadays, fuzzy set theory has been widely used in various fields. That is because fuzzy sets can explain vagueness and uncertainty which occurred in various fields. Fuzzy set can be viewed as generalization crisp set which showed its membership value [2].

Fuzzy set theory can be merged with neural network. Fuzzy LVQ (FLVQ) was proposed by Su and Chou [1] and showed high accuracy level than LVQ in pattern recognition. This study was focused on how to build a FLVQ model and use FLVQ to identify nine probability distributions that is normal distribution, exponential distribution, Weibull distribution, uniform distribution, chi-square distribution, Gamma distribution, t distribution, F distribution and lognormal distribution.

2. Learning Vector Quantization

Learning Vector Quantization (LVQ) is neural network with supervised learning methods. It was proposed by Kohonen [3]. LVQ also classified as competitive learning because its neuron input competes each other and the winner will be processed. LVQ has three algorithms, that is LVQ1, LVQ2, and LVQ3. In this study, algorithm LVQ1 is used because its simplicity and accuracy.

LVQ uses codebook vector to classify the input into true class. Codebook vector is assigned to each class of x and x is then determined to belong the same class to which the nearest w_i belongs. Let

$$w_c = \arg\min_i \|x - w_i\| \qquad (1)$$

define the index of nearest w_i to x. Notice that c, the index of the "winner", depends on x and all the w_i. Let $x(t)$ is input and $w_i(t)$ represents sequential values of the w_i in discrete-time domain, $t = 0, 1, 2, \dots, n$. Starting with properly defined initial values, the following equation defines the basic LVQ process; this algorithm is called LVQ1.

$$\begin{aligned}
\boldsymbol{w_c(t+1)} &= \boldsymbol{w_c(t)} + \boldsymbol{\alpha(t)}[\boldsymbol{x(t)} - \boldsymbol{w_c(t)}], && \boldsymbol{x, w_c \in C}\\
\boldsymbol{w_c(t+1)} &= \boldsymbol{w_c(t)} - \boldsymbol{\alpha(t)}[\boldsymbol{x(t)} - \boldsymbol{w_c(t)}], && \boldsymbol{x \in C, w_c \notin C}\\
\boldsymbol{w_i(t+1)} &= \boldsymbol{w_i(t)}, && \boldsymbol{i \neq C}
\end{aligned} \qquad (2)$$

Manuscript received October 5, 2016
Manuscript revised October 20, 2016

where $\alpha(t)$ is learning rate and C is class that belongs to x. Where $\alpha(t) \in (0,1)$ dan $\alpha(t)$is usually made to decrease monotonically in time (t).

3.Fuzzy Sets

Fuzzy sets can be viewed as generalized crisp set. Rutkowska [4] defines a fuzzy set as follows: Let X be a space of points, with a generic element of X denoted by x. A fuzzy set A in X is characterized by a membership function $\mu_A(x)$ which associates with each point x a real number in the interval [0,1] representing the grade of membership of x in A

$$\boldsymbol{A} = \{(\boldsymbol{x}, \boldsymbol{u_A}(\boldsymbol{x})); \boldsymbol{x} \in \boldsymbol{X}\} \quad (3)$$

Where

$$\boldsymbol{\mu_A}(\boldsymbol{x}): \boldsymbol{X} \rightarrow [\boldsymbol{0}, \boldsymbol{1}] \quad (4)$$

The closer value of $\mu_A(x)$ to one, the higher grade of membership of x in A. If $\mu_A(x) = 1$, then x fully belongs to A. If $\mu_A(x) = 0$, then x does not belongs to A. Space X is called universe of discourse.

A fuzzy subset of real line that has some additional properties is called fuzzy numbers. So that it can be stated that fuzzy numbers are a generalization of a classical real numbers. Rutkowska [4] defines a fuzzy numbers as follows: A fuzzy set A is a fuzzy number if the universe of discourse X in R and the following criteria are fulfilled:

a. The fuzzy set is convex
b. The fuzzy set is normal (maximum membership of function is one)
c. The membership function of fuzzy set is piecewise continue
d. The core of fuzzy set consists of one value only.

Generally, fuzzy numbers are classified into 3 types, that is, trapezoidal fuzzy number; triangular fuzzy number and Gaussian fuzzy number (see [5]).

4. Fuzzy Learning Vector Quantization (FLVQ)

Fuzzy LVQ (FLVQ) was originally proposed by Sakuraba, et al., [6]. In this study, modification will be made in original FLVQ. FLVQ is based on LVQ, so its algorithm is the same as LVQ. As already mentioned in section 2, the algorithm which is used for this study is LVQ1.

4.1 The Input Vector

The Input vector in FLVQ models must be in a fuzzy form. Vector in fuzzy form is called fuzzy vector.

Fuzzy vector is a vector where each element of vector is fuzzy numbers. Suppose y is vector with size $n \times 1$, that is

$$\boldsymbol{y} = [\boldsymbol{y_1}, \boldsymbol{y_2}, \dots, \boldsymbol{y_j}, \dots, \boldsymbol{y_n}] \quad (5)$$

Then fuzzy vector of y denoted by $y^*$is defined as

$$\boldsymbol{y}^* = [\boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \dots, \boldsymbol{y}_j^*, \dots, \boldsymbol{y}_n^*] \quad (6)$$

where $y_1^*, y_2^*, \dots, y_n^*$ is fuzzy number and $\mu_{y_j^*}(y_j)$ is membership function of y_j^*. In this study, triangular fuzzy numbers are used because they can state possibility numbers around the numbers with membership function value is one. Fuzzy vector input can be represented by a graphic as seen in Figure 1.

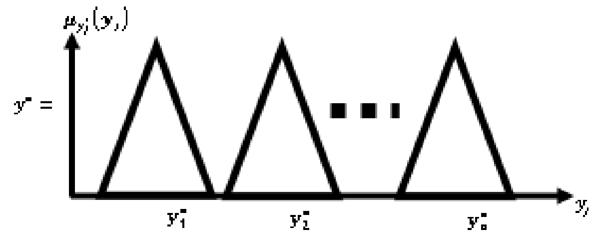

Fig. 1. Fuzzy Input Vector

4.2 Codebook Vector

In LVQ, codebook vector is used as an indicator to pattern recognition. FLVQ also has codebook vector that has fuzzy vector form. Suppose w is codebook vector defined as

$$w = [w_1, w_2, \dots, w_j, \dots, w_n] \quad (7)$$

then fuzzy codebook vector of w denoted by w^* is defined as

$$\boldsymbol{w}^* = [\boldsymbol{w}_1^*, \boldsymbol{w}_2^*, \dots, \boldsymbol{w}_j^*, \dots, \boldsymbol{w}_n^*] \quad (8)$$

where $\mu_{w_j^*}(w_j)$ is membership function of w_j^* . Fuzzy codebook vector also uses triangular fuzzy number. Figure 2 represents fuzzy codebook vector.

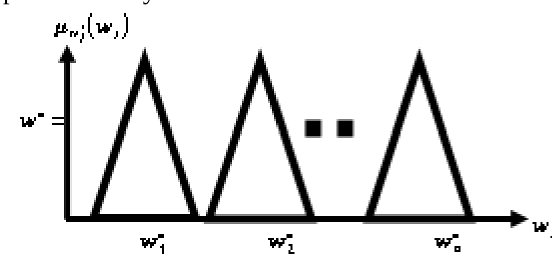

Fig. 2 Fuzzy Codebook Vector

4.3 Processing element in FLVQ

FLVQ has base processing element which is the same as LVQ. However, FLVQ processing element has to follow the fuzzy set operation. if LVQ uses Euclidean distance to pattern recognition from input data, then FLVQ uses fuzzy similarity measure to pattern recognition. Fuzzy similarity measure (see [7]) is used to measure similarity between input vector and codebook vector. If the fuzzy similarity is high, then the input vector and codebook vector are similar. If the fuzzy similarity is low, then the input vector and codebook vector are not similar. This study uses fuzzy similarity measure

$$\boldsymbol{M(A,B)} = \frac{\sum_{i=1}^{n} \min(\mu_A^i, \mu_B^i)}{\sum_{i=1}^{n} \max(\mu_A^i, \mu_B^i)} \quad (9)$$

where μ_A^i and μ_B^i are membership function of fuzzy set A and B. If $M(A,B)$ goes to 1, then it is similar. If $M(A,B)$ goes to 0, then it is not similar.

Thus processing element in FLVQ by using fuzzy similarity measure is defined as follows:

$$\boldsymbol{m_c = \max_i \frac{1}{n} \sum_{j=1}^{n} M(y_j^*, w_{ij}^*)} \quad (10)$$

where y_j^* is the jth element of vector y^* and w_{ij}^* is the ith fuzzy codebook vector (i is assigned class in codebook vector) and jth is input vector element.

4.4 Learning in FLVQ

Learning in FLVQ has two phases. Phase one, move codebook vector close to input vector if classification is true and move codebook vector far from input vector if classification is false. Phase two, update membership function of codebook vector. Update membership function is done by multiplication between fuzzy number and scalar. Suppose $s = (a_0^-, a_1, a_0^+)$ is triangular fuzzy number. Symbol a_1 denotes the core of fuzzy number and a_0^-, a_0^+ denotes upper and lower bound of triangular fuzzy number. Multiplication operation in membership function is defined as follows:

$$\boldsymbol{\lambda \cdot s = (\lambda a_0^-, a_1, \lambda a_0^+)} \quad (11)$$

where λ is a real number. If $\lambda > 1$ then membership function of fuzzy number will extend that means fuzziness level of fuzzy number becomes bigger, whereas if $\lambda < 1$ then membership function of fuzzy number will shrink that means fuzziness level of fuzzy number becomes smaller. Thus, the extend of membership function of codebook vector if classification is true and shrink membership function of codebook vector if classification is false.

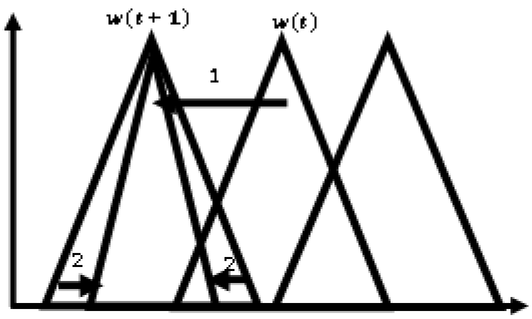

Fig. 3. Update Codebook Vector When Classification is False

Figure 3 and Figure 4 give an illustration of FLVQ learning processes. Arrow 1 shows codebook vector moves closer to input vector or moves far from input vector, whereas arrow 2 shows membership function of codebook vector which is extended or shrunk.

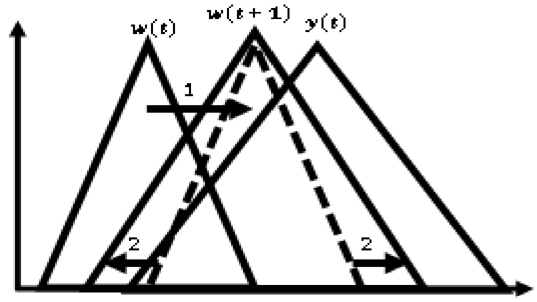

Fig. 4. Update Codebook Vector when Classification is true

Learning processes above can be expressed mathematically as follows: The fuzzy similarity is measured between $x(t)$ and $w_c(t)$ and can be calculated by using equation 4, and learning rule can be defined as follows:

$$\boldsymbol{w_c(t+1) = \theta(w_c(t) + \alpha(t)[x(t) - w_c(t)])}, \quad \boldsymbol{x, w_i \in C}$$

$$\boldsymbol{w_c(t+1) = \beta(w_c(t) - \alpha(t)[x(t) - w_c(t)])}, \quad \boldsymbol{x \in C, w_i \notin C}$$

$$\boldsymbol{w_i(t+1) = \eta w_i(t)}, \quad \boldsymbol{m_c = 0} \quad (12)$$

where c is class winner and C is class that belongs to $\boldsymbol{x(t)}$. The value of α is open interval between 0 and 1 and also decrease monotonically in time. For initial value of α, it is recommended to use small value, smaller than 0.1. In this study, the equation below is used for making α decrease monotonically in time.

$$\boldsymbol{\alpha(t) = 0.999\alpha(t-1)} \quad (13)$$

The value of β is in open interval between 0 and 1 and also decrease monotonically in time. For initial value of α, it is recommended to use high value, greater than 0.8. In this study, this equation below is used for making α decrease monotonically in time.

$$\boldsymbol{\beta(t) = 0.999\, \beta(t-1)} \quad (14)$$

The value of θ is greater than 1 and depends on the value of β. In this study, equation of θ is defined as follows:

$$\boldsymbol{\theta(t) = 2 - \beta(t)} \quad (15)$$

The value of η is greater than 1 and not time dependency. For initial value of α, it is recommended to use greater value than 1 and smaller than 1.5. All operation process in learning rule is based on fuzzy arithmetic (see [5]) and expects change of fuzziness level. It uses equation (3) to change fuzziness level.

4.5 Simulation of FLVQ

In this section, we discus the application of FLVQ to identify probability distribution by using R software.

(1). Algorithm

There are several steps to identify probability distribution.

a. Creating Database

In this study, database is created by generating data based on normal, lognormal, F, t, χ^2, Gamma, Exponential, Weibull and Uniform distribution. The sample size in each distribution is 5, 10, 20, 30 and 100 with 300 samples in each distribution. So, in each sample size contains 2700 samples.

b. Creating Training Data and Testing Data

Training data and testing data are created based on database. Training data and testing data contain 1800 and 900 samples. Sample selection in training data and testing data are conducted by using simple random sampling. So, training data and testing data have 200 samples and 100 samples in each distribution.

c. Creating initial codebook vector

In this study, there are nine probability distributions as mentioned above. It means that there are nine classes of codebook vector is used to represent each probability distribution. Initial value of these codebook vectors are determined by using K-NN (K-Nearest Neighborhood) method (see [8]) in training data.

d. Training FLVQ

Before FLVQ is trained, initial codebook vector and input vector must be in fuzzy vector. It is done by using fuzzification process. Fuzzification is a process to transform common data to fuzzy data. As mention in section 2, fuzzy vector element is triangular fuzzy number. In this study, upper bound and lower bound from triangular fuzzy numbers is defined by

$$\boldsymbol{y_j \pm \sigma_y}$$

where y_j is jth observation and σ_y is standard deviation of y. Then membership function of y_j^* that is $\mu_{y_j^*}(y_j)$ is defined by

$$\boldsymbol{\mu_{y_j^*}(y_j)} = \begin{cases} \boldsymbol{0} & , \boldsymbol{t} < a \\ \frac{\boldsymbol{y_j - a}}{\boldsymbol{b-a}}, & \boldsymbol{a \leq t} < b \\ \boldsymbol{1} & \boldsymbol{b} \\ \frac{\boldsymbol{c - y_j}}{\boldsymbol{c-b}}, & \boldsymbol{b} < t \leq c \\ \boldsymbol{0} & , \boldsymbol{t} > c \end{cases} \quad (16)$$

where $a = y_j - \sigma_{\boldsymbol{y}}$, $b = y_j$ and c=$y_j + \sigma_{\boldsymbol{y}}$.

The purpose of training FLVQ is to obtain optimal fuzzy codebook that can represent the 9 probability distributions.

e. Testing FLVQ

In this step, optimal fuzzy codebook vector is used for testing classification ability of FLVQ.

The results of the identification probability distribution are shown in Table 1. Based on Table 1, FLVQ model shows a very good accuracy to identify probability distributions. It is shown that the accuracy is above 50% for all probability distributions and most of them is close to 100 %.

Table 1. Accuracy of FLVQ model in probability distributions identification

Distribution	Accuracy (%)				
	Sample (n=5)	Sample (n=10)	Sample (n=20)	Sample (n=30)	Sample (n=100)
Normal	100	100	100	100	100
Lognormal	99	96	100	100	100
F	100	100	100	100	100
t	100	100	100	98	100
Chi-Square	82	100	99	100	100
Gamma	52	98	94	71	99
Exponential	91	82	99	100	100
Weibull	100	99	100	100	100
Uniform	100	100	100	100	100

5. Conclusion

The FLVQ model has a very good accuracy in identification of probability distributions. It can be shown that most of probability distribution has accuracy close to 100% in each sample size. In the future study it can consider for other different probability distributions.

References

[1] C.-T. Su & C.-J, CHou., A Neural Network Based Approach for Statistical Distribution Recognition. Quality Engineering, Vol. 18 (3), p293-297, 2007.

[2] Klir, G. J., & Yuan, B., Fuzzy Sets and Fuzzy Logic: Theory and Application. New Jersey: Prentice Hall, 1995.

[3] Kohonen, T., Improved Version of Learning Vector Quantization. International Joint Conference on Neural Network. Baltimore MD, 1990.

[4] Rutkowska, D., Neuro-Fuzzy Architectures and Hybird Learning. Berlin: Springer Verlag, 2002.

[5] Bede, B., Mathematics of Fuzzy Set and Fuzzy Logic. New York: Springer Verlag, 2013.

[6] Sakuraba, Y., Nakamoto, T., & Moriizumi, T., New Method of Learning Vector Quantization using fuzzy theory. Systems and Computers in Japan, Vol.22 (13),p93-103,1991.

[7] Baccour, L., Alimi, A. M., & John, R. I., Some Notes on Fuzzy Similarity Measures and Applications to Classification of Shapes, Recognition of Arabic Sentences and Mosaic. IAENG Int.J Comput.Sci, Vol. 41, p81-90,2014.

[8] Rencher,A.C., and Cristensen,W.F., Methods of Multivariate Analysis (3rd. Ed.), New Jersey: John Wiley& Sons, 2012.