DIABETIC PREDICTION USING FUZZY BACK PROPAGATION AND ANALYSIS

*Lohit Mittal and Anbarasi Masilamani

Scope, Vit University, Vellore, India

DOI: http://dx.doi.org/10.24327/ijrsr.2017.0803.0062

INTRODUCTION

The diabetic disease among the pregnant women are major concern issue as it not only affect themselves it also effect the newborn child, hence a proper diagnosis of the disease is necessary. There are many factors that affect the diabetic condition among the women which are determined by the optimized algorithm. The “organizations” estimated that 11 million deaths occur globally due to diabetic disease. Due to advancement in the medical and health industry it has become easier to determine the risk factors which may be the most probable reason for the diabetic disease, but there are still many cases of improper treatment and misuse of the resources have been seen. Medical Diagnosis is necessary procedure but it is complicated and need to handle out very carefully and due to increasing population the case for diabetic disease is increasing alarmingly and due to the shortage of medical staff it will be very beneficial if there is an automated system which can provide with efficient and accurate result. The expertise required to diagnosis is not posses by all the doctors. Hence we need an automated system to overcome the problem.

These study implements algorithms such as Fuzzy back propagation using triangular membership function, standard artificial neural network and naive Bayes algorithm and compare them. The analysis of fuzzy back propagation will also be done by epoch, N and M (memory constant) will be varied. The best results will be taken under consideration for the proposed system and for further implementation. The results which are taken and the graphs have been plotted and variation of results have been seen in the study.

LITERATURE REVIEW

The first Fuzzy Back Propagation neural network was purposed in the year 1994 by Stefka Stoeva, Alexander Nikov. They showed the importance of fuzzy approach as compared to the standard neural network. They approach simulated hierarchal approach towards the implementation of the fuzzy neural network. The next application of Fuzzy Back Propagation was done by Li, Liu, et al for early diagnosis of hypoxic ischemic encephalopathy in newborns. The next approach was taken by M. Anbarasi, M. A. Saleem Durai for prediction of protein folding kinetics in the year 2015.

The study of standard ANN goes back to 1980s by Frank Rosenblatt (1958) who developed first perceptron algorithms and the outcome of the study was used to develop smart automated software and systems. Later on it became the most important algorithm in the field of supervised learning and prediction. It was used in the application credit risk evaluation were it was compared with Naïve Bayes and SVM by Lohit et al in the year 2016. Naïve Bayes is one more important approach in terms of probabilistic approach. It has proven to give better results when compared with the algorithms such as PSO,
Random forest, ANN in some cases and other algorithms, hence it is necessary to compare the results with Naïve Bayes algorithm.

Jeroen Eggermont and Joost N. Kok and Walter A. Kosters were the first to work with the dataset. They used genetic algorithm. In the year 2004 Hastie, Trevor, et al tried the same dataset with the same data set and got a decent results. Till now no one has implemented Fuzzy Back Propagation Using Triangular Membership function on this dataset. In this study variation of parameters such as no. of epoch, N and M (memory constant) will be varied.

Many works on Naïve Bayes classifiers have been also done in this context.

**Dataset**

This study is based upon the data set on which the research has been carries out and the attributes used for prediction. The data set was donated by National Institute of Diabetes and Digestive and Kidney Diseases. A total of 760 records are taken and worked upon. As the missing values of the dataset were replaced by 0 so the dataset required some preprocessing. The method adopted is replacing the 0 value with the global mean of the following column. The data set was again preprocessed as it needed to be converted to binary data set where the output can be either 1 or 2. This 1 or 2 represent the label rather than their usual value, where 1 represents credit risk and 2 represents credit no risk. The output given by is in range of 0 to 1. As given by, the output is marked 2 if value is greater than 0.75 and 1 if the output is lower than 0.75, this result in a binary data set as shown in Table 1.

| S.NO | CLASS | OUTPUT | DISEASE |
|------|-------|--------|---------|
| 1    | >0.5  | 1      | DIABETIC |
| 2    | <0.5  | 2      | NON-DIABETIC |

There are eight predictor attributes in the data set that are 1. Number of times pregnant 2. Plasma glucose concentration a 2 hours in an oral glucose tolerance test 3. Diastolic blood pressure (mm Hg) 4. Triceps skin fold thickness (mm) 5. 2-Hour serum insulin (mu U/ml) 6. Body mass index (weight in kg/(height in m)^2) 7. Diabetes pedigree function 8. Age (years) 9. Class variable (0 or 1). To increase the accuracy hope the above algorithms, we need to preprocess the data as different algorithm required different preprocessing in Fuzzy algorithms data preprocessing is done using membership function and in normal algorithms preprocessing is done using average value in non-fuzzy algorithms that data is divided into classes. Each classes can be varied from maybe 3-5.

Classes given a label for example high medium low but fuzzy Back propagation the membership function divides 1 data set into sets of 3 using triangular membership function. In naive Bayesian the dataset also the data are divided using membership function Data set normalisation need to be done where the missing values or the zero values need to be replaced by the Global mean The data was stored in a CSV (comma delimiter) and was stored in mysql database using PHP Myadmin for all the ALGORITHMS. The data set is derived using java JDBC driver and loaded for further processing. The data set are divided into group of two. One group is used for training of the dataset and the other group is used for testing of the data set.

**Table 2 Dataset**

| Parameter          | Range  |
|--------------------|--------|
| Number of times pregnant | 0-17   |
| Diastolic blood pressure (mm Hg) | 44-199 |
| Plasma glucose concentration a 2 hours in an oral glucose tolerance test | 24-122 |
| Triceps skin fold thickness (mm) | 7-99   |
| 2-Hour serum insulin (mu U/ml) | 14-846 |
| Body mass index (weight in kg/(height in m)^2) | 18-67  |
| Diabetes pedigree function | 0.078-2.42 |
| AGE                | 21-81  |
| Class variable     | 0-1    |

**METHODOLOGY**

**Artificial Neural Network**

ANN is an algorithm which is used to model the human brain using the concept of perceptron network. It is a learning based algorithm in which the weights are trained first again and again unless they measure upon the output. This is called the training phase. It is a standard algorithm which has same proceedings as the fuzzy ANN but its simple and less complex.

The figure above shows the architecture of ANN as first it consists of first the input neuron to which corresponding data are inserted. Then we have hidden layers whose input is calculated using the summation function of weights

\[ u = \sum_{j=1}^{m} w_{ij} x_j \]

Now the hidden layers are calculated and fed into output neuron with a bias and the error is calculated. Then weight updation formula are applied and then again checked till we get the best accuracy and weight set is noted and used for testing of the data, to which the application is formed.

**Fuzzy Back propagation Using Triangular Membership Function**

Algorithm for Fuzzy Back propagation using triangular membership function:

1. Generate weights for the hidden input and output layer randomly between the range from 0.1 to 0.9. Here input weights are \( W_{ij} = (W'\alpha_{ij},W'\beta_{ij},W'\gamma_{ij}) \) and output hidden layer weights \( W'_{ij} = (W'\alpha_{ij},W'\beta_{ij},W'\gamma_{ij}) \)
2. Now the input be \( P = (P_1,P_2,\ldots,P_n) \) where \( N \) is the number of input layers. In Fuzzy all the input are convert to fuzzy number using fuzzy triangular membership function as \( P_i = (P_1\alpha_{ij},P_1\beta_{ij},P_1\gamma_{ij}) \)
3. The function above is used to convert normal data set to fuzzy dataset.
4. Now initialize the momentum factor \( \alpha = 0.9 \) and learning factor \( \eta = 0.9 \) and \( \Delta W(k-1) = 0 \) and \( \Delta W'(k-1) = 0 \) where \( k \) is iteration number.
5. Now compute \( T_{np} = f(TOT_{rp}), j=1,2,\ldots,n \) where
\[ T_{ijp} = \text{NE}\left(\sum_{i=0}^{n} W_{ij}^p T_{jp}\right) \]
\[ \text{NE}(x) = I_x + 0.33x (I_x - 1) \]
\[ f(T_{ijp}) = \frac{1}{1 + \exp(-T_{ijp})} \]

5. Compute step 4 similar for the output neuron again.

6. Compute the error as \( \Delta W'(k-1) \) and compute
\[
\nabla X_p = \left( \frac{\partial X_p}{\partial W'\alpha_{ij}}, \frac{\partial X_p}{\partial W'\beta_{ij}}, \frac{\partial X_p}{\partial W'\gamma_{ij}} \right)
\]
where
\[
\frac{\partial X_p}{\partial W'\alpha_{ij}} = -(A_{ij} - T_{ijp}').T_{jp}'.(1 - T_{ijp}').1 . T_{jk}
\]
\[
\frac{\partial X_p}{\partial W'\beta_{ij}} = -(A_{ij} - T_{ijp}').T_{jp}'.(1 - T_{ijp}').(-1/3).T_{jk}
\]
\[
\frac{\partial X_p}{\partial W'\gamma_{ij}} = -(A_{ij} - T_{ijp}').T_{jp}'.(1 - T_{ijp}').(1/3).T_{jk}
\]

7. Compute step 6 similar for the output neuron again.

8. \( \Delta W' = -\eta \nabla X_p + \alpha \Delta W'(k-1) \) Where \( \eta, \alpha \) are learning factor and memory constant.

9. \( \Delta W = \Delta W'(k-1) + \Delta W' \) Statement should be used for the weight updation.

10. Iterate from step 2 to 9 unless, we receive less errors.

**Naïve Bayes**

The algorithm determines the occurrence of a state using probabilistic prediction. Works on class labels. Attribute are predicted to be conditionally independent of one another. As the algorithm works on discrete data, in order to implement upon on continuous data, we need to divide the attributes among classes. We can use Gaussian, mean and fuzzy classifier. Gaussian produce the very poor result, hence we will be using mean and fuzzy classifiers.

\[
P(A | B) = \frac{P(B | A)P(A)}{P(B)}
\]

The formula above is used in which data is already divided in sets of classes and are determined whether they are co-dependent or independent. Another important classifier for Naïve Bayes is
\[
p(X_k | x_1, \ldots, x_n) = \frac{1}{Z} P(X_k) \prod_{i=1}^{n} p(x_i | X_k)
\]

Naïve Bayesian classifier is very effective in case of cost; it reduces the calculation cost. It is best for the problems where there is a strong relation between the variables and the data given.

**RESULTS AND DISCUSSION**

The artificial neural network and fuzzy back propagation using triangular membership function and naïve Bayes are all implemented in JDK 8.0, JSP (Java Servlet Package) and data set is managed using Mysql. The csv file has been converted to SQL file using PHP my admin and using Netbeans it has been past parsed. In fuzzy back propagation results have been obtained using variation of parameters. There are three main parameters Number of iterations, learning factor and momentum factor. Iteration has been varied between 100 to 500000. Learning factor have been varied between 0.2 to 0.9 and momentum factor have also been varied between 0.1 to 0.95 and results are obtained as follows

| Epoch | V | Momentum |
|-------|---|----------|
| 300   | 65 | 63       |
| 1000  | 55 | 52       |
| 5000  | 45 | 42       |
| 10000 | 46 | 43       |

Table 3: Epoch V Momentum

In the above table 3 the comparison is done by taking iteration and momentum factor. Momentum factor is varied from 0.2 to 0.9 and Iteration from 300 to 10000 and their corresponding error has been noted.

![Figure 1: System model](image-url)
It has been observed that the increasing iteration results in better accuracy same goes momentum factor as it increases the error decreases.

**Table 4 Epoch V Rate**

| Iter | Learning 0.2 | 0.4 | 0.5 | 0.6 | 0.8 | 0.9 |
|------|--------------|-----|-----|-----|-----|-----|
| 300  | 38           | 33  | 40  | 44  | 49  | 55  |
| 1000 | 38           | 35  | 42  | 44  | 48  | 50  |
| 5000 | 24           | 21  | 29  | 35  | 43  | 45  |
| 10000| 24           | 22  | 28  | 36  | 47  | 46  |

The next table is between iteration and learning factor as it can be observed that the increase in iteration leads to decreasing in error but increase of learning factor leads to increase in error.

**Table 5 Learning V Momentum**

| Learning/momentum | 0.2 | 0.4 | 0.5 | 0.6 | 0.8 | 0.9 |
|-------------------|-----|-----|-----|-----|-----|-----|
| 0.2               | 45  | 42  | 38  | 26  | 22  | 20  |
| 0.35              | 46  | 43  | 37  | 24  | 23  | 21  |
| 0.5               | 49  | 47  | 44  | 33  | 29  | 26  |
| 0.65              | 51  | 50  | 46  | 35  | 30  | 28  |
| 0.75              | 52  | 48  | 45  | 42  | 36  | 33  |
| 0.9               | 54  | 52  | 49  | 45  | 30  | 34  |

The next table is between learning factor and momentum factor. It can be seen that learning factor doesn't affect that much as compared to momentum factor. The results comply with the above observations.

The FIGURE 2 shows relation between error and learning factor, as it can be seen that increases in learning factor mostly leads to increase in error, hence in order to achieve better accuracy learning factor should be kept low.

**Table 3 Fuzzy Bp CM**

| Learning 'yes' | Predicted 'yes' | Predicted 'no' |
|----------------|-----------------|---------------|
| Actual 'yes'  | 180             | 22            |
| Actual 'no'   | 20              | 70            |

**Table 4 Naïve Bayes CM**

| Learning 'yes' | Predicted 'yes' | Predicted 'no' |
|----------------|-----------------|---------------|
| Actual 'yes'  | 200             | 10            |
| Actual 'no'   | 20              | 70            |
The Figure 3 is between learning rate and Momentum factor, it can be seen as the moment factor increases the error decreases.

Table 5 Ann CM

|                | Predicted‘yes’ | Predicted‘no’ |
|----------------|---------------|---------------|
| Actual ‘yes’   | 170           | 19            |
| Actual ‘no’    | 40            | 73            |

The Figure 4 is between number of iteration and error rate it can be seen as we increase our iteration for certain time the error decreases but after a certain iteration the error rate remain constant.

Now the accuracy of the dataset is taken by three algorithms and their confusion matrix is taken. A set of 300 values have been taken to determine testing.

As it can be seen from the above results Naïve Bayes tends to give better accuracy than the fuzzy ANN and ANN but fuzzy ANN shows better accuracy than the standard ANN.

CONCLUSION

The novelty of the work is to make comparison among machine learning algorithm and to determine the best which we obtained was Naïve Bayes as it more efficient in classification than the other two, although the other algorithm were slightly less they can also be used to determine the diabetic diseases prediction. The results obtained can be base for implementation of a system for diabetic diseases prediction.

Reference

1. Rosenblatt, F. (1958). "The Perceptron: A Probabilistic Model For Information Storage And Organization In The Brain". Psychological Review 65 (6):386-408. doi:10.1037/h0042519. PMID 13602029.
2. Stoeva, Stefka, and Alexander Nikov. "A fuzzy back propagation algorithm." Fuzzy Sets and Systems 112.1 (2000): 27-39.
3. Li, Liu, et al. "The use of fuzzy backpropagation neural networks for the early diagnosis of hypoxic ischemic encephalopathy in newborns." BioMed Research International 2011 (2011).
4. Mittal, Lohit, Tarang Gupta, and Arun Kumar Sangaiah. "Prediction of Credit Risk Evaluation Using Naive Bayes, Artificial Neural Network and Support Vector Machine.”
5. Anbarasi M. DOI: 10.1007/978-3-319-30348-2_36 · In book: Proceedings of the 3rd International Symposium on Big Data and Cloud Computing Challenges (ISBCC - 16’), pp.419-443
6. Jang, J.-S., & Sun, C.-T., (1993). Functional equivalence between radial basis function networks and fuzzy inference systems. IEEE Transactions on Neural Networks 4 (1), 156-158.
7. Hawley, D.D., Johnson, J.D. & Raina D., (1990). Artificial neural systems: A new tool for financial decision-making. Fin. Anal. J., 46: 63-72. DOI: 10.2307/2F4479380 Huang
8. Minsky, M.; S. Papert (1969). Perceptrons. MIT Press.
9. Odom, M., & Sharda, R. (1990). A neural network model for bankruptcy prediction. Proceedings of the International Joint Conference on Neural networks, pp. 163-168.
10. Rumelhart, D.E; James McClelland (1986). Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Cambridge: MIT Press.
11. Angelini, E. di Tollo, G. & Roli, A. (2008). A neural network approach for credit risk evaluation. The Quarterly Review of Economics and Finance, Vol. 48 No. 4, pp. 733-755
12. Mohsen, N., & Mojtaba, A. (2013). Measuring credit risk of bank customers using artificial neural network. Journal of Management Research., Vol. 5, No. 2, Pp. 17.
13. Aeh, H., Lee, K., & Kim, K.: Global Optimization of Support Vector Machines Using Genetic Algorithms for Bankruptcy. Lecture Notes in Computer Science, vol 4234(5), pp. 420-429. Springer. (2006)
14. Wu, C., Tzeng, G., Goo, Y., Fang, W.: A real-valued genetic algorithm to optimize the parameters of support vector machine for predicting bankruptcy. Expert Systems with Applications. Vol. 32, pp. 397-408 (2007).
15. Zhang, D., Chen, Q., Wei, L.: Building Behavior Scoring Model Using Genetic Algorithm and Support Vector Machines. Lecture Notes in Computer Science. Vol. 4488, pp. 482-485 (2007).
16. Hao, P.-Y., Lin, M.-S., Tsai, L.-B.: A New Support Vector Machine with Fuzzy Hyper-Plane and Its Application to Evaluate Credit Risk. 2008 Eighth International Conference on Intelligent Systems Design and Applications, pp. 83-88 (2008).

How to cite this article:
Lohit Mittal and Anbarasi Masilamani.2017, Diabetic Prediction Using Fuzzy Back Propagation and Analysis. Int J Recent Sci Res. 8(3), pp. 16044-16048.