EUROPEAN JOURNAL OF PURE AND APPLIED MATHEMATICS
Vol. 6, No. 4, 2013, 377-386
ISSN 1307-5543 – www.ejpam.com

Binary Classification for Hydraulic Fracturing Operations in Oil & Gas Wells via Tree Based Logistic RBF Networks

Oguz Akbilgic

Department of Chemical and Petroleum Engineering, Schulich School of Engineering, University of Calgary, AB, Canada
Department of Quantitative Methods, Istanbul University School of Business, Istanbul, Turkey

Abstract. In this paper we develop a novel tree based radial basis function neural networks (RBF-NNs) model incorporating logistic regression. We aim to improve the classification performance of logistic regression method by pre-processing the input data in RBF-NN frame. Although the scope of our proposed method is binary classification in this paper, it is easy to generalize it for multi-class classification problems. Furthermore, our model is very convenient to adapt for $n < p$ classification problem that is very popular yet difficult topic in statistics. We show the generalization and classification performance of our model using simulated data. We have also applied our model on a real life data set gathered from hydraulic fracturing in Oil & Gas wells. The results show the high classification performance of our model that is superior to logistic regression. We have coded our model on R software. Logistic Regression applications were carried out using IBM SPSS Version 20.

2010 Mathematics Subject Classifications: 62M45, 03C45, 62J12

Key Words and Phrases: Hydraulic Fracturing, Radial Basis Function Neural Networks, Classification and Regression Trees, Logistic Regression

1. Introduction

Radial basis function (RBF) neural networks (NNs) [5] are one of the techniques to handle binary classification problems. Using fewer assumptions to compare parametric statistical techniques is one of the reasons that RBF-NNs has become very popular for applications for real life data [6]. However, there are some problems accompanying RBF networks such as over fitting, randomness in parameter detection, and using a gradient approach that might lead networks to fixate on a local optimum. Furthermore, there is no theoretical justification for determining the number of hidden neurons in the network. In this paper, we develop a novel tree-based RBF-NN model incorporating logistic regression in order to avoid such problems as well as to improve the classification performance of the logistic regression.

Email address: oguzakbilgic@gmail.com

There are several studies in the literature combining RBF networks with other statistical techniques to improve its performance. Kubat [11] introduced the idea of initializing RBF networks with decision trees. Following Kubat [11], Orr [12] used classification and regression trees to determine the center and radius parameters of RBF functions. Akbilgic and Bozdogan [2, 3] improved Orr's [12] work by bringing ridge regression and variable selection scheme into the RBF-NN frame.

In our model, we use classification and regression trees (CART) [4] to determine the center and radius parameters of radial basis functions taking place in the hidden layer of RBF networks. Each terminal node obtained by CART corresponds to a hidden neuron in the RBF network [11, 12]. Thus the number of hidden neurons on a hidden layer is automatically determined. At this point, each hidden neuron carries out a nonlinear transformation of predictors via RBFs into the hidden layer.

Usage of the new data that is a nonlinear transformation of original data by the hidden layer, as predictors highly tends to cause singularity problem of design matrix. To be able to handle the possible singularity problem, we reduce the dimensionality of the hidden layer by using the AIC [1] based stepwise technique for logistic regression. By doing so, not only do we handle the singularity problem in logistic regression, but we also reduce the size of the RBF-NN model, which prevents the model from the over fitting problem. Furthermore, the proposed model leads us to obtain completely continuous predictors in hidden layer by processing original categorical predictors via tree-based RBFs. Hence, the data become very suitable to be processed using logistic regression in terms of continuity assumption of predictors.

In Section 2, we described our proposed method right after giving brief definitions of RBF-NNs, CART, and the logistic regression. Section 3 is to show our proposed model's classification and generalization performance. To do so, we apply our model on simulated data and a real life data set derived from hydraulic fracturing operations in Oil & Gas wells. We discuss our model and the results in the Conclusions section.

2. Tree Based Logistic RBF Network Model with Logistic Regression

In this paper we developed a novel approach for binary classification problems by embedding the classification, regression trees, and the logistic regression into the RBF-NN frame. Brief definitions of the techniques used, as well as their role in RBF-NN frame, are described in following subsections.

2.1. Radial Basis Function Neural Networks

Radial basis function neural networks are special types of neural networks. RBF-NNs are distinguished by having only one hidden layer using RBFs as an activation function of hidden neurons. Furthermore, input data is directly sent to the hidden layer without being weighted unlike the other multi layer feed forward neural network models [7]. The simple presentation of RBF-NN is given by (1) where the outputs are modeled as a sum of m weighted radial basis

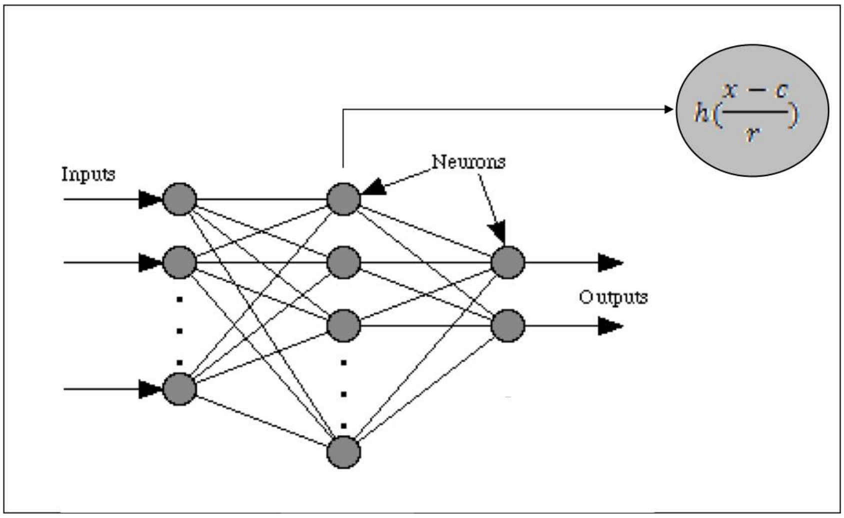

Figure 1: Radial Basis Function Neural Network.

functions $h(\cdot)$ with centers c_j, radii r_j, and weights w_j $(j = 1 \ldots m)$:

$$\hat{y_i} = f(x; w, c, r) = \sum_{j=1}^{m} w_j h(\frac{x_i - c_j}{r_j}) \quad i = 1, \ldots, n \tag{1}$$

In (1), $h_j(x)$ are RBFs, transferring input space nonlinearly to hidden layer, and w are the parameters connecting hidden layer to output layer. The Figure 1 simply presents the structure of RBF-NN.

Radial basis functions produce monotonically increasing or decreasing values by moving away from a center which is a parameter in RBF-NN model. By the strong localization property of RBFs, the RBF-NN model possesses the properties of best approximation [13]. Although there are several RBFs seen in the literature, in this study we use the most common one, Gaussian RBF in 2 [9].

$$h(x; c, r) = e^{-\left(\frac{x-c}{r}\right)^2} \tag{2}$$

During the RBF learning, there are basically three parameters to be determined; w, c, and r. Furthermore, the number of hidden neurons (m) in the hidden layer can be considered as another parameter. As explained in the following subsections, m, c, and r parameters are determined by classification and regression trees while the w is determined by using logistic regression.

2.2. Classification and Regression Trees for the Proposed Model

Classification and Regression Trees (CART) [4] is one of the tree-based statistical techniques for prediction and classification problems depending on the type of target variables, either continuous or categorical. The idea of the CART algorithm is based on recursively splitting input space into two hyper rectangles, minimizing a fitness value such as prediction or classification error, and turning input space to smaller hyper-rectangles including data points. The end nodes, called terminal nodes, are the hyper-rectangles where there is no more split. Following Kubat [11], and Orr [12], we obtained center and radius parameters of RBFs from terminal nodes. At this point, each terminal node corresponds to a hidden neuron in RBF-NN structure that solves the problem of determining the number of hidden neurons in the hidden layer of the network. Center coordinates of the terminal nodes correspond to the center parameters (c) of RBFs while the widths (s) of the terminal nodes are used to obtain radius parameter (r) of RBFs by scaling with a α parameter, $r = \alpha s$. Here, α is another parameter of our model that needs to be optimized. Figure 2 illustrates how the two-dimensional input space is split into hyper-rectangles including data points. The beauty of classification and regression trees is to be able to handle the different input-output relationships in different sub spaces of overall input space as can be seen in Figure 2.

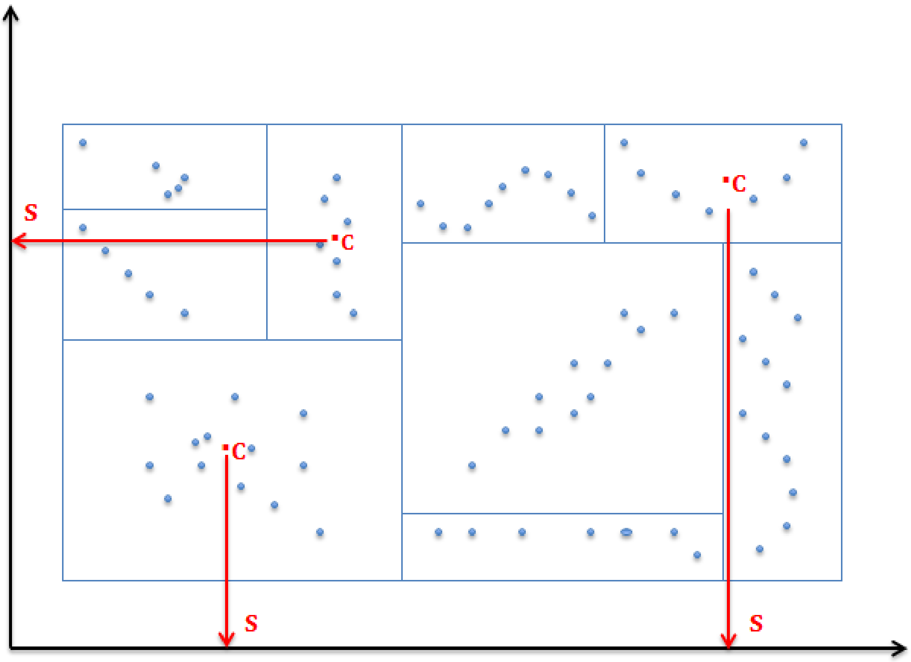

Figure 2: Splitting the data space into sub-spaces using CART

2.3. Logistic Regression for the Proposed Model

Logistic regression is one of the most commonly used classification methods in the literature. For a given input variables X, the logistic regression function is given by (3) [14], where the regression parameters β are determined by numerical optimization, such as in Newton's Method.

$$y = \frac{1}{1 + e^{-\sum X\beta}} \tag{3}$$

We use logistic regression to estimate the hidden layer weights w_j, and for classification in the output layer of the RBF-NN frame. In our model, the **H** matrix, which is the nonlinear transformation of predictors under the RBF-NN frame, replaces X in (3).

$$\mathbf{H}_{n\times(m+1)} = \begin{bmatrix} 1 & w_1 h(\frac{X_{i1}-c_1}{r_1}) & \dots & w_m h(\frac{X_{im}-c_m}{r_m}) \\ 1 & w_1 h(\frac{X_{21}-c_1}{r_1}) & \dots & w_m h(\frac{X_{2m}-c_m}{r_m}) \\ & & \vdots & \\ 1 & w_1 h(\frac{X_{n1}-c_1}{r_1}) & \dots & w_m h(\frac{X_{nm}-c_m}{r_m}) \end{bmatrix} \tag{4}$$

2.4. Proposed Model: Data Flow Algorithm

Our proposed method is based on pre-processing data by using classification and regression trees before applying logistic regression in the RBF-NN frame. The data flow of proposed method is explained step by step. Note that there are two conditional steps depending on the problem specifications.

Step 1 Use CART to split the original p-dimensional input space into m hyper-rectangles.

Step 2 (Conditional) If $n < p$ and/or $n < m$, prune the tree until m is small enough to avoid a design matrix that is rank deficient.

Step 3 From each terminal node, compute c_j and $r_j = \alpha s_j$, and place them in the m neurons of the RBF hidden layer.

Step 4 With the weights initialized to 1, transform the original $n \times p$ matrix X to the $n \times m$ matrix **H** using the parameterized radial basis functions.

Step 5 Using the initial **H** matrix and the original $n \times 1$ matrix of binary class labels, y, perform logistic regression to estimate the optimal weight parameters, w_j.

Step 6 (Conditional) Use stepwise logistic regression to avoid singularity of the design matrix and to avoid over fitting.

Step 7 Obtain the final classifier $\hat{y}$ by combining (3) and (4), with the weights from Step 5.

Following the given steps of the proposed model, the final output of the model is given by (5).

$$y = \frac{1}{1 + e^{-\sum_{j=0}^{m} H(x;c,r)w}} \tag{5}$$

As can be seen from the steps of the proposed method, our model gives us options to solve $n < p$ classification problems that is one of the popular and most difficult problems in statistics and machine learning.

Applications of our models on simulated and real data are documented in the following section.

3. Applications

In this section we first apply our method on simulated data in order to show the generalization performance of the proposed method. We compare our results with classical logistic regression. After showing the generalization performance of our method, we applied it using real data. All calculations for the proposed model are carried out using R [8] while for the logistic regression, SPSS 20 is used.

3.1. Application of Proposed Method on Simulated Data

We aimed to explore the generalization performance of our proposed method on simulated data. Simulation data included a mixture of two bivariate normal distribution, $N(\mu, \Sigma)$, which are highly overlapped as it is seen in Figure 3. The number of observations of each group are specified as 250 each, and parameters of the distributions for each group are defined below.

We ran our model for generated data by chancing the α parameter from the set $\{.15, .20, .25, \ldots, 2.5\}$. We find best α parameter as $alpha = 0.9$ giving the highest classification accuracy. The confusion matrix obtained for $alpha = 0.9$ is shown in Table 1.

$$\mu_1 = \begin{pmatrix} 2.0 \\ 1.0 \end{pmatrix} \quad \Sigma_1 = \begin{pmatrix} 1.2 & 0.1 \\ 0.1 & 0.25 \end{pmatrix},$$

$$\mu_2 = \begin{pmatrix} 3.0 \\ 2.0 \end{pmatrix} \quad \Sigma_2 = \begin{pmatrix} 0.5 & -0.1 \\ -0.1 & 0.3 \end{pmatrix}$$

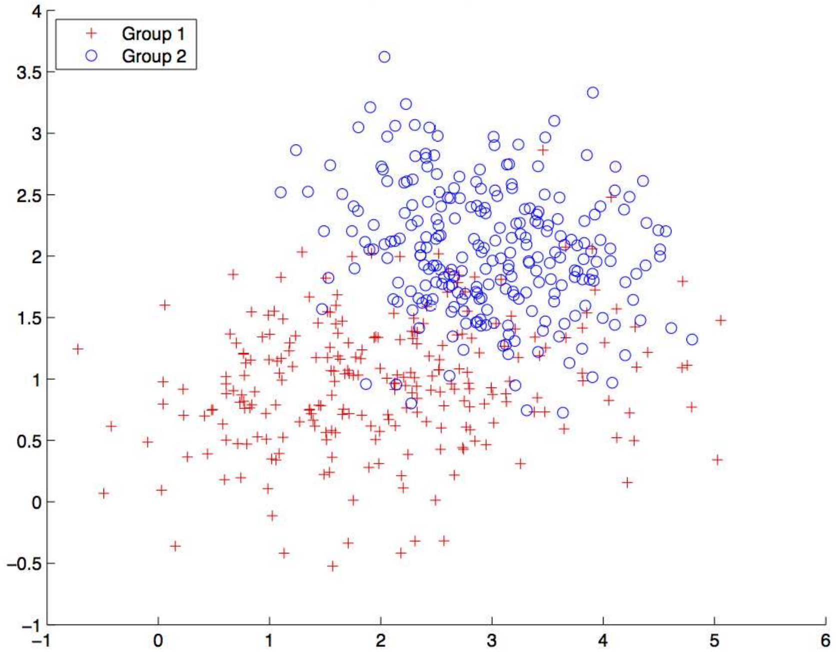

Figure 3: Highly Overlapped Structure of Simulated Data.

Table 1: Confusion Matrix for Log-RBF-NN Model

Group Label	0	1	Total	Accuracy
0	219	31	250	87.60%
1	18	232	250	92.80%
Total			500	90.20%

Table 2: Confusion Matrix for Logistic Regression

Group Label	0	1	Total	Accuracy
0	215	35	250	86.00%
1	26	224	250	89.60%
Total			500	87.80%

Table 3: Confusion Matrix for Train Data

Group Label	0	1	Total	Accuracy
0	174.33	25.61	199.94	87.19%
1	15.85	184.21	200.06	92.07%
Total			400.00	89.64%

Table 4: Confusion Matrix for Test Data

Group Label	0	1	Total	Accuracy
0	42.99	7.07	50.06	85.85%
1	4.71	45.23	49.94	90.47%
Total			100.00	88.22%

As is seen from Table 1, the proposed model shows high classification accuracy, although the groups are highly overlapped. We ran logistic regression for the same data set and show the confusion matrix in Table 2 comparing our results. Comparing Table 1 and Table 2, we can see that the proposed method is superior to the Logistic Regression in terms of classification accuracy.

Although our model uses the RBF-NN frame, there is no random process to determine the network parameters. At this point, it is a question of whether the proposed model requires cross validation as the classical neural network methods or not. To make sure about the generalization performance of our model, we randomly split data into train (80%) and test (20%). We repeated this splitting process 100 times and document the average results in Table 3 and Table 4.

During the cross validation process, we determined network parameters using train data and carried out classification for test data with pre-determined parameters. At this point, classification accuracy should not be decreased for test data to be able to claim that the proposed method offers good generalization performance. Table 3 and Table 4 show the high generalization performance of our model in terms of classification accuracy where there is no significant drop off in test data. Moreover, the classification accuracy for train and test data is very close to that obtained without splitting data. The small difference between classification accuracy between "with and without" splitting can be explained with the number of observations which are obviously more for non-split data.

3.2. Pre-Determination of Hydraulic Fracturing Failure for Oil & Gas Wells

Hydraulic fracturing (fracking) is an unconventional Oil & Gas extraction technique [10, 15]. Fracking operation is basically injecting a high-pressured solution of chemicals into the ground around the walls of oil or gas wells [5, 6]. Fracking operations are not allowed in some countries because of the chemicals injected into ground, which contain very hazardous elements. Furthermore, fracking is a very costly operation costing approximately $\$500,000$ or more for each operation. Due to these two reasons, it is crucial to make an educated decision regarding applying a fracking operation within a specific zone.

In this study, we applied our proposed method on historical data obtained from an international oil company operating in Turkey under a non-disclosure agreement. Our data includes 50 hydraulic fracturing operations from 38 different natural gas wells. For each fracking operation, our input data contains 18 different well-log measurements, such as Gamma Ray, Resistivity Logs, Porosity Logs, etc. We also have the results from fracking operations as to whether gas was produced or not. The goal is to find a classification model to determine, before the operation, which zones are going to produce gas by fracking. Thus, the companies can avoid the unnecessary cost of the operation as well as ensuring that hazardous chemicals would not unnecessarily affect the environment.

We performed our tree based Log-RBF-NN model to carry out classification on hydraulic fracturing data. Again, we ran the proposed model for different α parameters from the set $\{.15, .20, .25, \ldots, 2.5\}$. We found the best performing α value as $\alpha = 0.85$ which is very close for the best operating α parameter for simulated data. Note that, the classification accuracy of the model was increasing until the best performing α values while it was decreasing with the larger values for both simulated and real data.

Table 5 shows the results of the confusion matrix for the classification of hydraulic fracturing data. Since our sample size is very small, we did not divide our data into train and test. Instead, we pruned the classification and regression tree so that it produced only five terminal nodes returning five hidden neurons in the hidden layer considering the number of observations, $n = 50$. Note that the pruning operation is basically selecting the best performing hidden neurons in the hidden layer using the AIC-based variable selection scheme for logistic regression.

Table 5 shows the high classification accuracy of our proposed method on fracking data. Using this model as a decision support system in the Oil & Gas industry might result in saving time and potentially millions of dollars. Furthermore, the environment would be protected from the hazardous effects of chemicals used in fracking operations. Note that classification accuracy obtained by logistic regression for same data is 78.00%, which is very low compared to our method's accuracy of 94.00%. On the other hand, the accuracy of the engineers who are decision makers making decisions on hydraulic fracturing operations is 66%.

Table 5: Confusion Matrix for Hydraulic Fracturing Data Set

Group Label	0	1	Total	Accuracy
0	15	2	17	88.24%
1	1	32	33	96.97%
Total			50	94.00%

4. Conclusions

In this paper we developed a novel tree-based logistic RBF-NN model for binary classification problems. We avoided using any techniques using random process during the determination of network parameters. As a result, our model reflects very good generalization properties, which are supported by an example using highly overlapping simulation data. Furthermore, our model gives high classification accuracy using real life data as well.

Since our model is a novel approach it is open for improvement in different ways. Although there are no examples taking place in the paper, we have applied our model on $n < p$ classification problems and obtained promising results. We have also tested our model on classification problems with more than two classes and again our results were encouraging in terms of classification accuracy. An additional benefit to using our model is that it is possible to reduce dimension by excluding the predictors that are not used in splitting data by classification and regression trees algorithms.

While the real data used in this study is related to obtaining natural gas by hydraulic fracturing, this data does not clearly indicate the true economic value. While the fracking may be deemed successful, it may cost more to extract than the value of the gas obtained or produced. If we were to define successful fracking as obtaining economical natural gas, the success of the company decreases from 66% to slightly less than 40%. This opens up new areas to examine, which we will be focusing on in our future studies.

ACKNOWLEDGEMENTS The author would like to thank Dr. J.Andrew Howe and Clarissa Stevens-Guille for reviewing the paper and for the grammatical corrections.

References

[1] H. Akaike. Information theory and an extension of the maximum likelihood principle. In B.N. Petrox and F. Csaki, editors, *Second International Symposium on Information Theory.*, pages 267–281, Budapest, 1973. Academiai Kiado.

[2] O. Akbilgic and H. Bozdogan. Predictive Subset Selection Using Regression Trees and

RBF Neural Networks Hybridized with the Genetic Algorithm. *European Journal of Pure and Applied Mathematics*, 4:467–485, 2011.

[3] O. Akbilgic, H. Bozdogan, and M.E. Balaban. A novel Hybrid RBF Neural Networks model as a forecaster. *Statistics and Computing*, 2013.

[4] L. Breiman, J. Freidman, J. C. Stone, and R. A. Olsen. *Classification and Regression Trees*. Chapman & Hall, 1984.

[5] J. Caers. *Petroleum Geostatistics*. Society of Petroleum Engineers, 2005.

[6] D.M. Carson. *Our Petroleum Challenge: Sustainability into 21st Century*. Canadian Centre for Energy Information, 7th edition, 2009.

[7] S Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall,, New Jersey, 1999.

[8] Kurt Hornik. The R FAQ, 2013.

[9] R.J. Howlett and L.C. Jain. *Radial Basis Function Networks 1: Recent Developments in Theory and Applications*. Physica Verlag, New York, 2001.

[10] J.R. Jones and L.K. Britt. *Design and Appraisal of Hydraulic Fractures*. Society of Petroleum Engineers, 2009.

[11] M. Kubat. Decision trees can initialize radial basis function networks. *Transactions on Neural Networks*, 9:813–821, 1998.

[12] M. Orr. Combining Regression Trees and RBFs. *International Journal of Neural Systems*, 10:453–465, 2000.

[13] T. Poggio and F. Girosi. Regularization algorithms for learning that are equivalent to multilayer networks. *Science, New-Series*, 247:978–982, 1990.

[14] I.H. Witten, E. Frank, and M.A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufman, 3rd Edition, 2011.

[15] C.H. Yew. *Mechanics of Hydraulic Fracturing*. Gulf Professional Publishing, 1st edition, 1997.