2009 Ninth International Conference on Intelligent Systems Design and Applications

Risk Factor Identification and Classification of Macrosomic Newborns by Neural Networks

A. Guillen, A. M. Trujillo, S. Romero, G. Rubio, I. Rojas, H. Pomares, L. J. Herrera

Department of Computer Architecture and Technology
University de Granada.
Spain
Email: aguillen@atc.ugr.es

J.F. Guillen

Department of Preventive Medicine
University of Granada.
Spain

Abstract

This paper presents a first approach to try to determine if a newborn will be macrosomic before the labor, using a set of data taken from the mother. The problem of determining if a newborn is going to be macrosomic is important in order to plan cesarean section and other problems during the labor. The proposed model to classify the weight is a Neural Network whose design is based recent algorithms that will allow the networks to focus on a concrete class. Before proceeding with the design methodology to obtain the models, a previous step of variable selection is performed in order to indentify the risk factors and to avoid the curse of dimensionality. Another study is made regarding the missing values in the database since the data were not complete for all the patients. The results will show how useful the addition of the missing values into the original data set can be in order to identify new risk factors.

***Keywords*-Macrosomy, weight prediction, newborn, classification, Mutual information**

I. Introduction

The problem of identifying if a newborn will be macrosomic or not consists of the determination of the final weight, if the baby is over 3.9 or 4 Kilograms, the newborn is considered macrosomic [18].

The most important information that could be obtained is the fetal macrosomia, this is, a birth weight of more than 4 Kilograms. Macrosomia is difficult to predict and clinical and ultrasonographic estimates tend to have errors [1]. Furthermore, the weight of the fetus is a risk factor for several diseases such as gestational diabetes mellitus [4]. Therefore, if we are able to determine if the newborn is macrosomic, we will know in advance one of the many elements that are used to identify diseases.

The work carried out also considered the identification of the risk factors that can determine if the newborn will be macrosomic. The risk factors identification helps also to design the models since it alleviates the $curse$ of dimensionality [11]. In addition to this, a priori treatment of the data was performed: missing values were filled using recent methodologies, allowing a more complete analysis of the data.

The rest of the paper is organized as follows, Section 2 describes the model, tools and algorithms used to design the classifiers and to perform a pre-treatment of the data. Then Section 3 shows the results of the experiments considering different subsets of data. Finally, in Section 4, conclusions are drawn.

II. Tools and models applied

This section describes first the type of neural network employed for the classification task. Then, the Mutual Information (MI) theory used to reduce the dimensionality is

978-0-7695-3872-3/09 $26.00 © 2009 IEEE
DOI 10.1109/ISDA.2009.251

described. Afterwards, the procedure to deal with missing values is depicted.

A. Radial Basis Function Neural Networks (RBFNN) Description

A RBFNN (Figure 1) $\mathcal{F}$ with fixed structure to relate a set of n inputs $X = [\vec{x}_i]; i = 1...n$ with an output $Y = [y_i]; i = 1...n$ has a set of parameters to be optimized:

$$\mathcal{F}(\vec{x}_k; C, R, \Omega) = \sum_{j=1}^{m} \phi(\vec{x}_k; \vec{c}_j, r_j) \cdot \Omega_j \qquad (1)$$

where $C = \{\vec{c}_1, ..., \vec{c}_m\}$ is the set of RBF centers, $R = \{r_1, ..., r_m\}$ is the set of values for each RBF radius, $\Omega = \{\Omega_1, ..., \Omega_m\}$ is the set of weights and $\phi(\vec{x}_k; \vec{c}_j, r_j)$ represents an RBF. These networks are widely used in regression using gaussian neurons [2], [16] although they have been also applied to classification problems with unbalanced data sets [**?**], as the problem described in this paper.

The procedure to design an RBFNN starts by setting the number of RBFs in the hidden layer, then the RBF centers $\vec{c}_j$ must be placed and a radius r_j has to be set for each of them. Finally, weights Ω_j can be optimally calculated by solving a linear equations system [6].

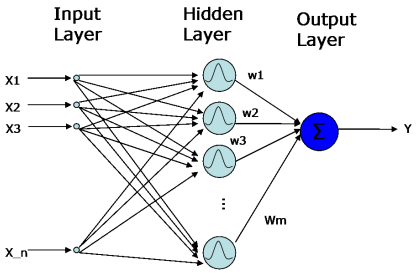

Figure 1. A Radial Basis Function Neural Network

The algorithms applied in the experiments section [7], [8], although originally designed for function approximation, have the ability to weight the output in such a way that the network is able to establish a preference in the classification. Thus, we can make the network more specific and focused on the identification of the macrosomic infants instead of the normal newborns.

B. Reducing the dimensionality

In order to reduce the dimensionality, the Mutual Information (MI) theory has been used. Given a single-output multiple input function approximation or classification problem, with input variables $X = [x_1, x_2, \ldots, x_n]$ and output variable $Y = y$, the main goal of a modelling problem is to reduce the uncertainty on the dependent variable Y. According to the formulation of Shannon, and in the continuous case, the uncertainty on Y is given by its entropy defined as

$$H(Y) = -\int \mu_Y(y) \log \mu_Y(y) dy, \qquad (2)$$

considering that the marginal density function $\mu_Y(y)$ can be defined using the joint probability density function $\mu_{X,Y}$ of X and Y as $\mu_Y(y) = \int \mu_{X,Y}(x, y)dx$. Given that we know X, the resulting uncertainty of Y conditioned to known X is given by the conditional entropy, defined by

$$H(Y|X) = -\int \mu_X(x) \int \mu_Y(y|X = x) \log \mu_Y(y|X = x) dy dx. \qquad (3)$$

The joint uncertainty on the $[X, Y]$ pair is given by the joint entropy, defined by

$$H(X, Y) = -\int \mu_{X,Y}(x, y) \log \mu_{X,Y}(x, y) dx dy. \qquad (4)$$

The mutual information (also called cross-entropy) between X and Y can be defined as the amount of information that the group of variables X provide about Y, and can be expressed as $I(X, Y) = H(Y) - H(Y|X)$. In other words, the mutual information $I(X, Y)$ is the decrease of the uncertainty on Y once we know X. Due to the mutual information and entropy properties, the mutual information can also be defined as $I(X, Y) = H(X) + H(Y) - H(X|Y)$, leading to

$$I(X, Y) = \int \mu_{X,Y}(x, y) \log \frac{\mu_{X,Y}(x, y)}{\mu_X(x)\mu_Y(y)} dx dy. \qquad (5)$$

Thus, only the estimate of the joint Probability Density Function (PDF) between X and Y is needed to estimate the mutual information between two groups of variables.

Estimating the joint probability distribution can be performed using a number of techniques. As mentioned already, histograms and kernel density estimators have been used for this purpose although this paper uses the method based on the k-nearest neighbours presented in [13]. As it is recommended in [9] for a tradeoff between variance and bias, in the examples, a mid-range value for k (k = 6) will be used.

C. Dealing with missing values

The incomplete-data problem, in which certain feature values are missing from particular observations, exists in a wide range of fields, including social sciences [14], biological systems [15], [12], [3], and remote sensing [10]. Missing data are often avoided by filling with specific values.

In this paper we use a novel proposed imputation framework [5] which aims to improve the performance of single imputation methods. It is the combination of four main modules: mean pre-imputation, base imputation, confidence intervals and boosting. This process is graphically depicted in Figure 2.

In module 1, the missing values are temporarily filled with the mean (for numerical attributes) or mode (for nominal attributes) of the corresponding attribute. Then, in module 2, each missing pre-filled value is completed by using a base imputation method, which in our case is Hot Deck [17]. The next step is to perform a filtering process of the filled values by using confidence intervals (module 3). This filter selects those filled values that have high probability of being correct, which are close to the mean or mode of an attribute, and removes possible outlier imputations. Finally, the boosting module (module 4) accepts or rejects the imputed values, based on a threshold. This threshold is defined as the average distance between the records with missing data and the records from which the imputed values were taken (closest records). As a result, a partially filled database is created and fed back to the base imputation algorithm. The process repeats until 10 iterations when the completed database is produced.

III. Experimental Results

The data used for the experiments were provided by the Preventive Medicine Department at the University of Granada, and consists of a cohort of 1962 pregnant women considering 50 variables. These variables were measured by doctors during the periodic visits of the pregnant women.

A. Considering missing values

The method for handling missing values has been implemented taking into account information provided by the target class attribute of supervised database used, with the aim of improving the accuracy of the imputation. So, the distances calculated in the Hot Deck method and the confidence intervals were computed from records of the same class.

In our database, the target class attribute is a continuous variable, so two classes have been defined allowing the imputation method to be applied from the viewpoint of the classification. This fact produces two types of records, those in which the weight of the fetus is less than 3.9 Kilograms (class **a**) and those in which the weight is equal to or greater than 3.9 Kilograms (class **b**).

After this transformation, the database is composed of 1009 complete records (908 in class **a** and 100 in class **b**) and 954 records with missing values (877 in class **a**, and 77 in class **b**).

The number of boosting iterations should be the least giving enough accuracy, assuming that an increase in the number of iterations leads to more computations. In our experiment, the selected number of iterations has been 10. This selection is based on comparison of results when the method is boosted at different number of times. It has been observed that after approximately 10 iterations the number of accepted filled values don't increase because their quality is not appropriate under the threshold set.

Finally, the database is composed of 1570 complete records (1421 in class **a**, and 148 en class **b**) and 392 records with missing values (364 in class **a**, and 28 in class **b**). The remaining completed values that have been rejected are filled with the value obtained by the initial pre-imputation mean process.

Thus, all missing values have been filled and a completed database is obtained.

B. Identifying risk factors

The values of MI for each variable and the output were computed with both data sets: *filled*, having the missing values added (1962 instances), and *original*, which consist in a subset (1008 instances) of the original instances without the vectors containing missing values.

The absolute values of MI are shown in Tables I and II. In order to make the table easily readable, only the variables with a MI value over 0.01 are shown. Figure 3 represents the normalized (between 0 and 1) values of MI for the $original$ and $filled$ data sets. A remarkable fact is that the MI values for the $filled$ data set are much higher than for the original one. Furthermore, among the 15 most important variables, they only match in 3 variables: 7 (pregnancy duration), 21 (inital mother's weight), and 28 (final mother's weight).

These results were analyzed by a medical expert, who agreed with the variables selected using the $filled$ data set. This indicates that the addition of the missing values was succesfully performed.

C. Classification accuracy

After the preprocessing of the input, the networks were designed to perform the classification. Due to the scarce number of input vectors for macrosomic infants, several training and test sets were defined using a Leave One Out (LOO) validation, but only considering the macrosomic instances. The results are shown in Table III.

The results show how difficult is the identification of macrosomic babies. However, the accuracy in the classifications during the training process was low, as the

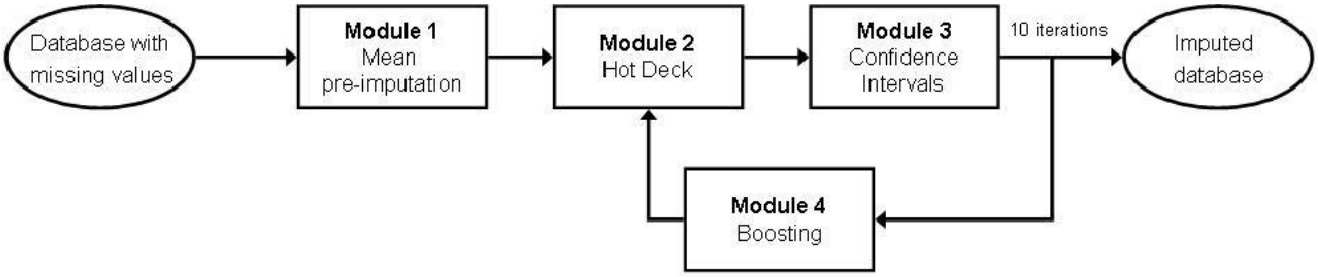

Figure 2. Missing values treatment procedure.

MI value	Var number
0.0154	36
0.0155	44
0.0191	48
0.0197	29
0.0206	3
0.0207	14
0.0241	22
0.0279	25
0.0282	21
0.0284	40
0.0288	38
0.0294	46
0.0311	39
0.0326	15
0.0326	49
0.0334	27
0.0351	11
0.0355	26
0.0392	28
0.0400	37
0.0404	7
0.0450	16
0.0460	47
0.0493	17

Table I. MI value ($>$ 0.01) for each variable for the $filled$ data set.

MI value	Var number
0.0111	10
0.0115	20
0.0122	21
0.0161	29
0.0223	36
0.0466	28

Table II. MI value ($>$ 0.01) for each variable for the $original$ data set.

data set	Training	Test
$original$	90.9 %	10 %
$filled$	91.2 %	11.3 %

Table III. Classification accuracy using the different data sets.

networks seem to overfit the data. This is a consequence of having a low number of input vectors of macrosomic infants.

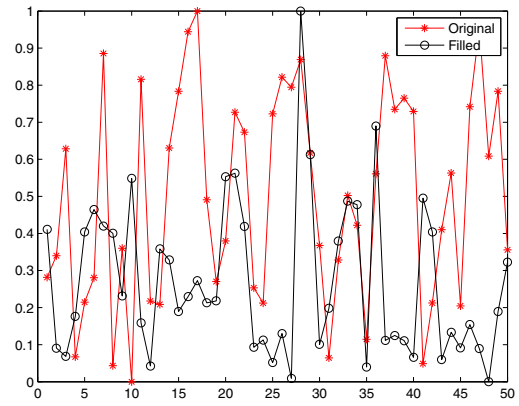

Figure 3. Normalized Mutual Information values for the $original$ and $filled$ data sets.

IV. Conclusions

This work has presented an application of RBFNNs to a real world problem: the classification of newborns to predict if they will be or not macrosomic. This task could be quite useful in the preventive treatment of the labor and its planning, considering hospitalization and cesarean section. The methodology used considered the addition of missing values and a previous step of variable selection that could provide several indicators to determine the risk factors. The accuracy of the results was not as good as desired due to the highly unbalanced data set, even after the addition of the missing values. Therefore, the networks focus in the identification of normal infants. However, it was noted how the use of specific algorithms can improve the performance. Further work will consider new data where the two classes are more balanced. Nonetheless, it is remarkable to check that the addition of the missing values was useful to determine the risk factors through the variable selection.

Acknowledgment

This work has been partially supported by the Spanish CICYT Project TIN2007-60587 and Junta Andalucia Project P07-TIC-02768.

References

[1] J. Berard, P. Dufour, D. Vinatier, D. Subtil, S. Vanderstichele, JC. Monnier, and et al. Fetal macrosomia: risk factors and outcome. A study of the outcome concerning 100 cases $>$ 4500 g. *Eur J Obstet Gynecol Reprod Biol*, 77:51–59, 1998.

[2] A. G. Bors. Introduction of the Radial Basis Function (RBF) networks. *OnLine Symposium for Electronics Engineers*, 1:1–7, February 2001.

[3] Naisyin Wang Danh V. Nguyen and Raymond J. Carroll. Evaluation of missing value estimation for microarray data. *Journal of Data Science*, 2:347–370, 2004.

[4] R. Dyck, H. Klomp, L.K. Tan, R.W. Turner, and M.A. Boctor. A comparison of rates, risk factors, and outcomes of gestational diabetes between aboriginal and non-aboriginal women in the Saskatoon Health District. *Diabetes Care*, 25:487–493, 2002.

[5] A. Farhangfar, L. Kurgan, and W. Pedrycz. A novel framework for imputation of missing values in databases. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, 37(5):692–709, 2007.

[6] J. González, I. Rojas, J. Ortega, H. Pomares, F.J. Fernández, and A. Díaz. Multiobjective evolutionary optimization of the size, shape, and position parameters of radial basis function networks for function approximation. *IEEE Transactions on Neural Networks*, 14(6):1478–1495, November 2003.

[7] A. Guillén, J. González, I. Rojas, H. Pomares, L.J. Herrera, O. Valenzuela, and A. Prieto. Improving Clustering Technique for Functional Approximation Problem Using Fuzzy Logic: ICFA algorithm. *Neurocomputing*, DOI:10.1016/j.neucom.2006.06.017, June 2007.

[8] A. Guillén, J. González, I. Rojas, H. Pomares, L.J. Herrera, O. Valenzuela, and F. Rojas. Output Value-Based Initialization For Radial Basis Function Neural Networks. *Neural Processing Letters*, DOI:10.1007/s11063-007-9039-8, June 2007.

[9] H. Stogbauer, A. Kraskov, S. A. Astakhov and P. Grassberger. Least dependent component analysis based on mutual information. *Physics Review*, December 2004.

[10] Mihail Halatchev and Le Gruenwald. Estimating missing values in related sensor data streams. In *COMAD*, pages 83–94, 2005.

[11] L.J. Herrera, H. Pomares, I. Rojas, M. Verleysen, and A. Guillen. Effective Input Variable Selection for Function Approximation. *Lecture Notes in Computer Science*, 4131:41–50, 2006.

[12] Hyunsoo Kim, Gene H. Golub, and Haesun Park. Missing value estimation for dna microarray gene expression data: local least squares imputation. *Bioinformatics*, 21(2):187+.

[13] A. Kraskov, H. Stogbauer, and P. Grassberger. Estimating mutual information. *Physics Review*, June 2004.

[14] Kamakshi Lakshminarayan, Steven A. Harp, and Tariq Samad. Imputation of missing data in industrial databases. *Applied Intelligence*, 11(3):259–275, 1999.

[15] Erkki Pesonen, Matti Eskelinen, and Martti Juhola. Treatment of missing data values in a neural network based decision support system for acute abdominal pain. *Artificial Intelligence in Medicine*, 13(3):139–146, 1998.

[16] I. Rojas, M. Anguita, A. Prieto, and O. Valenzuela. Analysis of the operators involved in the definition of the implication functions and in the fuzzy inference proccess. *International Journal of Approximate Reasoning*, 19:367–389, 1998.

[17] G. Sande. Hot deck imputation procedures. *Incomplete Data in Sample Surveys*, 3:339–349, 1983.

[18] M. A. Zamorski and W.S. Biggs. Management of Suspected Fetal Macrosomia. *American Family Physician*, 63(2), January 2001.