A Neurofuzzy Approach for Property Value Prediction

Merisandra Côrtes de Mattos[1], Priscyla Waleska Simões[1], Evelise Zancam[1], Niria B.Ferreira[1], and Cristian Cechinel[2]

[1] Universidade do Extremo Sul Catarinense, Av.Universitária 1105
Bairro Universitário, 88806-000, Criciúma, SC, Brazil
{mem,pri,evelise}@unesc.net, niriaborges@hotmail.com
www.unesc.net

[2] Universidade Federal do Pampa, Unipampa/Bagé, Rua Carlos Barbosa S/n
Bairro Getúlio Vargas, 96412-420, Bagé, RS, Brazil
contato@cristiancechinel.pro.br
www.unipampa.edu.br

Abstract. This work presents a neurofuzzy network developed for property value prediction. The solution has an hybrid architecture with 6 layers and five linguistic variables (square footage, number of bedrooms, location, quality and price). The data set used to train and test the network was acquired during the months of January and April 2006 and contained informations about 200 apartments in the city of Criciúma (Brazil). Among these cases, 150 were used to train the neurofuzzy network, and 55 to test it. During the training and testing processes the model achieved a medium error of 7.06% when using a 0.5 learning rate and training periods of 2000. Due to the small amount of data used to run the tests, it is still not possible to validate the network, and more work towards this direction is needed.

Key words: Neural Networks, Fuzzy Sets, Neurofuzzy Networks, Property Value Prediction

1 Introduction

Property value prediction is the process of assigning market value to a property for specific purposes [14], such as helping on: setting up tax rates, buying/ selling/ locating transactions, insurance contract analysis, and financing and credit evaluation [2], [1]. Different techniques for an acuratte prediction of property prices have been developed and tested over the years, and this subject is still under the attention of the academic research community. The very first price model that could be mentioned here is called Related Sales Prices Indexes [9] and consists of a comparative method, where the predicted value of some property is calculated based on the prices of similar properties recently sold. A more accurate method for house prices valuation is the Hedonic Method which uses the attributes of several properties to extract an equation (hedonic equation)

XXXIV Conferencia Latinoamericana de Informática (CLEI 2008), pp. 72-81, 2008.

mostly through Multiple Regression Analysis(MRA) [9]. This method assumes that a property can be viewed as an aggregation of individual attributes (such as: the number of rooms, geographical area, neighboorhood, square footage, age, conservation) that have different impact over its price, and once the hedonic equation is generated, it can be used to predict the prices of other properties using as inputs the avaiable attributes[8]. The Hedonic Method approach using MRA tends to be popular to end-users because their models are generally considered easy to specify, estimate, and extrapolate [12]. On the other hand, this linear and parametric approach carries with it some handicaps (and restrictions) such as: statistical and econometric weakness, non-normality of most variables [3] and exposure to sampling errors [12]. These problems are directly related to the fact that House Prices Valuation is (in most of the cases) a non-linear problem, and the parametric statistics, such as MRA, work better for linear situations. Regarding this, several works ([15],[16],[14],[12],[5]) have proposed an alternative Hedonic Model based on Artificial Neural Networks(ANNs). They claim ANNs to be an efficient non-parametric approach to this problem due to their ability in the task of learning, generalisation and recognition of complex patterns in multivariate data. Despite the advantages of its use, some authors have argued that ANNs do not always outperform the MRA, and the use of the first in preference to the second is still a controversial issue. For instance, [13] suggests that ANNs perform well for smaller data sets, while MRA is superior for larger data sets, [17] highlights ANNs are not easy to use and present inconsistent results between different neural network packages and between runs of the same neural network software. On the other hand, [11] has made performance comparisions between MRA and ANNs, and defends that if the sufficient data training size and ANN parameters (model specification, number of hidden layers, learning rate, number of training cycles) are provided (which is not an easy task), the ANN performs better than MRA for any amount of data. At the same time, [11] also calls attention to the fact that ANNs have a black-box nature, i.e., the internal ANN structure is hidden, and consequently it is difficult to explain the achieved results. In order to minimize this problem, some works [6],[7] have proposed a hybrid approach which uses ANNs and Fuzzy Logic, the Neurofuzzy Network approach (NFN). In these solutions they try to combine the Fuzzy Logic capability of explicit represent imprecision and vagueness (presents in House Prices Valuation problem) with the ANN hability of learning from data.

This paper describes a first attempt towards the development of a NFN for House Prices Valuation problem. The following sections present a brief review about NFNs, the NFN development process, the partial results obtained, and some conclusions about the work.

2 Neurofuzzy Networks

Neurofuzzy networks is the denomination given to models which combine ANN and Fuzzy Logic. The main goal of using this approach is to combine the advantages of each model in a manner to overcome their individual limitations. For instance, while fuzzy logic systems are good to represent the knowlege in a explicit way (which gives them the possibility to explain their decisions), they are not able to automatically acquire this knowledge. At the same time, ANNs can easily automatically extract knowledge and rules from data, but are not good to explain how they reach their decisions. [4]. The combination of both in the same model can help to enhance the performance of intelligent systems; in our case, systems to predict residential prices.

There are basically two general kinds of NFN models: the cooperative model and the hybrid model.

2.1 The Cooperative Approach

This approach uses ANNs to optimize the parameters of a normal Fuzzy System, to preprocess or postprocess the inputs and/or outputs of the Fuzzy System, and also to extract fuzzy rules from data [4]. Four possible kinds of models for this approach are described by [10], when the ANN: 1) derives the membership functions from training data, 2) derives linguistic control rules from training data, 3) adapts parameters of the fuzzy sets, and 4) learns weight factors to the fuzzy rules. Because learning happens only in the ANN part, the cooperative approach is not always considered (or classified) as a Neurofuzzy model - sometimes it is called as pre or postprocessing approach[10].

2.2 The Hybrid Approach

On this model, the ANN and the Fuzzy parts are totally integrated in an homogeneous architecture, where the Fuzzy System can be interpreted as an ANN. According to [10], the ANN weights are represented by the fuzzy sets, and the neurons are represented by the input and output variables and the fuzzy rules. This architecture is represented by a three-layer network, where each layer consists of one fuzzy system reasoning step: fuzzyfication, fuzzy rules evaluation, and defuzzification. During the **fuzzification** step, each fuzzy-neuron receives the input values and, through a membership function, calculates the membership values of them. These values will be the given to the if-clauses of the fuzzy-rules of the next layer. The **fuzzy rules** layer represents the system rules base, where each fuzzy-neuron of the layer corresponds to a fuzzy rule. This layer calculates the then-clause value through a T-norm function, and gives the result to the defuzzification layer. The **defuzzification layer** gives the final output, calculating the product between sums of the firing strenghts of each neuron from the previous layer and the rules consequent values.

3 The Neurofuzzy Network Development

The development of the Neurofuzzy Network for House Prices Valuation is described in the following steps: 1) Linguistic Sets Definition, 2) Fuzzy Rules Extraction, 3) The Architecture Generation, 4) Data Description, 5) Training and, 6) Testing.

3.1 Linguistic Sets Definition

This step can also be seen as the knowledge acquisition process. In this process some interviews with the expert in the domain (a civil engineering) were made to discover the most important factors involved on the house prices valuation problem. Although there are severall variables which influence the price of a residency, this study has selected just five to model the neurofuzzy network:

$$Sets = \{square\ footage, number\ of\ bedrooms, location, quality, price\}$$

Each linguistic set was divided into three fuzzy sets, and each fuuzy set was defined by a trapezoidal membership function (because of its simplicity and efficiency with respect to computability). The equation 1 shows a generic trapeizodal function:

$$\mu(x,a,b,c,d) = \begin{cases} 0 & \text{if} \quad x < a, x > d \\ (x-a)/(b-a) & \text{if} \quad a \leq x \leq b \\ 1 & \text{if} \quad b < x < c \\ (d-x)/(d-c) & \text{if} \quad c \leq x \leq d \end{cases} \qquad (1)$$

Where x is the input value, a and d are the left and right base points of the trapezoid, and b and c the top of it. The figure 1 shows a graphical representation of this function:

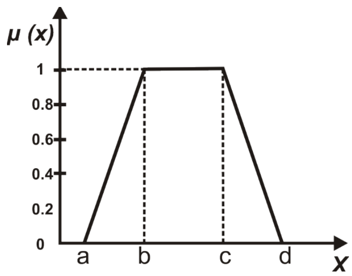

Fig. 1. Graphical Representation of Trapezoidal Membership Function

The description of each linguistic variable, its fuzzy sets and their respective membership functions are given next:

1. **Square Footage (SF)**: which represents the size of the residency in square meters (m^2). This variable was defined in three fuzzy sets (Small, Average and Large) in the range $[40, 1000]\, m^2$ by the following membership functions:

   $\mu_{SF=Small}(x, 40, 40, 60, 100)$, $\mu_{SF=Average}(x, 60, 100, 150, 200)$, and $\mu_{SF=Large}(x, 150, 200, 1000, 1000)$.

2. **Number of Bedrooms (NB)**: defined in the range $[1, 6]$. If the bedroom has a suite, it must be added 0.5 in its value. The fuzzy sets for this variable are: Few, Average and Many and their respective membership functions are:

$$\mu_{NB=Few}(x,1,1,1.5,2.5),\ \mu_{NB=Average}(x,1.5,2.5,3.5,4.5),\ \text{and}\ \mu_{NB=Many}(x,3.5,4.5,6,6).$$

3. **Location (L)**: the minor distance (in meters) between the residency and some of the most valious neighborhoods of the city in the range $[0,\infty]\,m$. The fuzzy sets defined for this variable are: Near, Medium and Far, and their membership functions are:

$$\mu_{L=Near}(x,0,0,100,300),\ \mu_{L=Medium}(x,100,300,500,1000),\ \text{and}\ \mu_{L=Far}(x,500,1000,\infty,\infty).$$

4. **Quality(Q)**: identifies the quality of the property with values defined in the range $[0,\infty]$ points. These points were defined by the expert, and they are calculated through the product between the standard of the property (High, Normal and Low), and the conservation index (Bad, Regular, Good and Great). The fuzzy sets for this variable are Low, Medium and High, and their membership functions are:

$$\mu_{Q=Low}(x,0,0,100,200),\ \mu_{Q=Medium}(x,100,200,300,400),\ \text{and}\ \mu_{Q=High}(x,300,400,\infty,\infty).$$

5. **Price (P)**: it is the output of the system, with values defined in the range $[20000,\infty]$. The fuzzy sets for this output variable are: Low, Average and High, and their membership functions are:

$$\mu_{P=Low}(x,20000,20000,40000,80000),\ \mu_{P=Average}(x,40000,80000,150000,200000),\ \text{and}\ \mu_{P=High}(x,150000,200000,\infty,\infty).$$

3.2 Fuzzy Rules Extraction

From the combination of the different fuzzy sets of each linguistic variable, 243 possible fuzzy rules were generated. Following the knowledge engineer orientation (a civil engineer), 72 rules were selected to be put on the rules base. The table 1 shows some examples of the selected rules:

Table 1. Fuzzy Rules Sample

If SF is Small	and NB is Few	and L is Near	and Q is Low	then P is Low
If SF is Average	and NB is Average	and L is Far	and Q is High	then P is High
If SF is Average	and NB is Many	and L is Medium	and Q is Medium	then P is Average
If SF is Large	and NB is Few	and L is Near	and Q is Medium	then P is Average

3.3 The Neurofuzzy Network Architecture Generation

To model the neural part of the neurofuzzy network, a multilayer perceptron ANN architecture was used. After the definition of the fuzzy sets and of the fuzzy rules, a six layers neurofuzzy network was created. These layers are: 1) Input Layer, 2) Input Fuzzy Sets Layer, 3) Fuzzy Rules Inference Layer, 4) Normalisation Layer, 5) Output Fuzzy Sets Layer, and 6) Output Layer. Each layer and its functioning is described next:

Input Layer This layer just receives the input data and sends them to the neurons of the next layer. The data received can be for purposes of training, testing or using.

Input Fuzzy Sets Layer The neurons of this layer receive the values given by the previous layer and calculate the membership degrees of each value using the trapeizodal membership functions described in the section 3.1. This layer represents the fuzzification layer mentioned in the section 2.2.

Fuzzy Rules Inference Layer Each neuron of this layer represents a fuzzy rule. The membership degree of the consequent part of each rule is calculated by the intersection of the antecedent values of it. In other words, the membership degree of the consequent part is the minimum value between the antecedent values, as described in 2:

$$S_n = \min\left[\mu_{SF}(x), \mu_{NB}(x), \mu_L(x), \mu_Q(x)\right] \; for \; n = 1, 2, 3, \ldots, N \tag{2}$$

where $\mu_{SF}(x)$ means the membership function of x in $square\ footage$, and N is the number of unchained rules. At the end, the results are send to every neuron of the next layer.

Normalisation Layer The neurons of this layer are responsible for the normalisation of the values received from the previous layer. To normalise consists on dividing each membership value by the sum of all membership values, as described in 3:

$$S'_n = \frac{S_n}{S_1 + S_2 + S_3 + ... + S_n} \tag{3}$$

where, S'_n is the output value of the neuron, and S_n are the values given by the neurons from the previous layer. The S'_n value is then sent to just one neuron of the next layer, i.e., to the neuron which corresponds to the consequent part of the fuzzy rule from the Fuzzy Rules Inference Layer.

Output Fuzzy Sets Layer Each neuron of this layer applies the union operation between the membership values received, i.e., it chooses the maximum membership value between them, as described in 4

$$A(x) = \max\left[S'_1 + S'_2 + S'_3 + ... + S'_n\right] \tag{4}$$

The value $A(x)$, obtained in 4, is then used in the defuzzification process. This process uses the centroid method and calculates the product between $A(x)$ and the consequent part of the fuzzy rule, as shown in 5

$$hn = A(x).C_A \tag{5}$$

where C_A is the mean between the left and right base points (a and d) of the membership function (equation 1) which represents the fuzzy set of the consequent part of the fuzzy rule.

Output Layer This layer gives the system output through the sum of the values given by the previous layer, as shown in 6

$$output = h_1 + h_2 + h_3 + \cdots + h_n \tag{6}$$

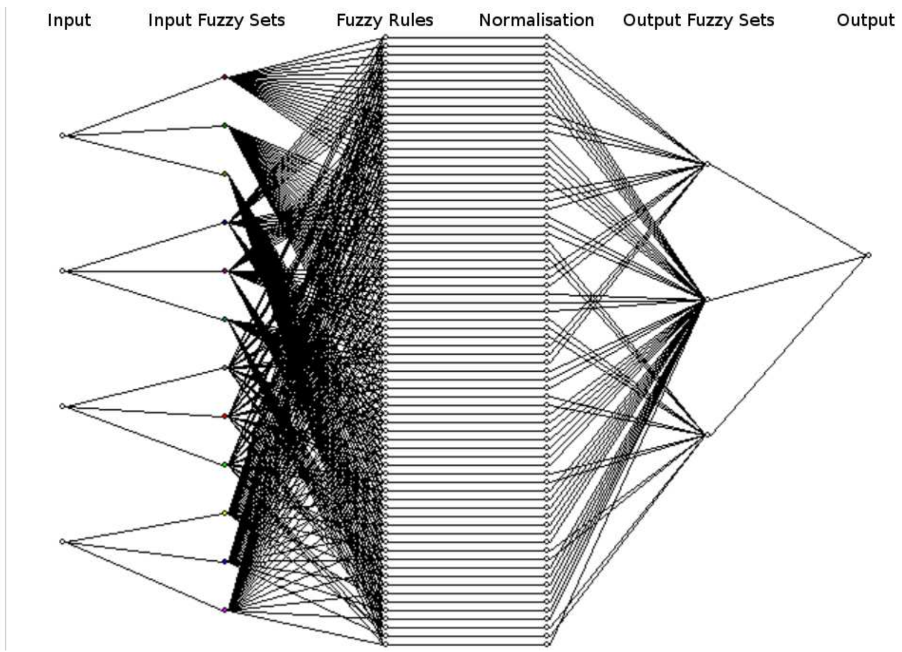

Fig. 2. The Neurofuzzy Network Architecture

3.4 Data Description

The data set used in the present work was acquired during the months of January and April 2006 and contained informations about the number of rooms, the geographical area, the square footage, the conservation, the standard of the property, and the price of 200 apartments in the city of Criciúma (Brazil). Among these cases, 150 were used to train the neurofuzzy network, and 55 to test it.

3.5 Training

The network was trained using the backpropagation algorithm combined with a supervised error-correction learning rule described in 7

$$e_j(n) = d_j(n) - y_j(n) \tag{7}$$

where j is the output neuron, d_j the desired output value, and y_j the actual network output value. In order to minimize processing costs, the equation 7 it was adapted, dividing its result by C_A, as shown in 8

$$e_j(n) = \frac{d_j(n) - y_j(n)}{C_A} \tag{8}$$

The error value e_j is used to adjust the network weights, as shown in 9

$$\Delta w_j = \eta.(e_j(n).y_j(n)) \tag{9}$$

where w_j is the weight of the neuron j, Δw_j represents the delta correction and η represents the learning rate. After the delta correction calculation, the weights of the network are adjusted by 10

$$w_j = w_j + \Delta w_j \tag{10}$$

This process is made during several training periods to adjust the weights of the neurons from the Input Fuzzy Sets Layer and the Normalisation Layer.

3.6 Testing

After several simulations, using learning rates in the interval between 0.1 and 0.5 and fixed training periods of 2000, the network reached the best medium error with a 0.5 learning rate. The results of the tests are described in table 2.

The network was also trained with smaller amounts of data due to evaluate the influence of the sample size in the training process. The table 3 shows that as the sample size gets bigger, the medium error reduces significantly.

Table 2. Neurofuzzy Network Results for Different Learning Rates

Learning Rate	*Medium Error*	*Medium Error Percentage (%)*
0.1	0.121258	12.12
0.2	0.088933	8.89
0.3	0.077115	7.71
0.4	0.079634	7.96
0.5	0.070602	7.06

Table 3. Neurofuzzy Network Results for Different Sample Sizes

Sample Size	*Medium Error*	*Medium Error Percentage (%)*
20	0.079799	7.97
50	0.078590	7.85
70	0,077453	7.74
100	0.076894	7.68
150	0.070602	7.06

4 Partial Results and Discussion

Different ways of training the network were used regarding specifically which were the best layers to have their neuron weights adjusted during training process. After severel attempts, the best results (Medium Error of 7.06%) were achieved when: 1) the weights modification is made only on the Input Fuzzy Sets Layer and Normalisation Layer; 2) it is used training periods of 2000; and 3) it is used learning rates of 0.5. Although these results can be considered good, more work towards validation is still needed. The reduced amount of real data to test and train the network is the most significant limitation for the validation of the present work. As can be seen in table 3 the amount of data influences directly on the network error, and it is necessary to increase this amount of data to measure the network prediction accuracy.

5 Final Remarks

Hybrid systems have been used as an alternative to solve several kinds of problems. In this work it was presented a hybrid neurofuzzy network for property value prediction. This approach has as an advantage the combination of fuzzy logic capability of explicit represent and explain reasoning with the ANN hability of learning from data. Because of the small amount of data gathered to train and test the network, the results obtained with the developed model are still inconclusive. The next steps of this research are focused on collect more data about house prices, and use this data to run more tests related to the network performance.

References

1. Abunahman, S.: Curso básico de engenharia legal de avaliações. Pini, São Paulo (1999)
2. Dantas, R.A.: Engenharia de Avaliações: Uma introdução a Metodologia Científica. Pini, São Paulo (1998)
3. Fischer,D., Lai, P.P.: Artificial Neural networks and Computer Assisted Mass Appraisal. In: 12th Annual Conference of the Pacific Rim Real Estate Society (PRRES). Auckland, New Zealand (2006)
4. Fullér, R.: Introduction to Neuro-Fuzzy Systems. Series: Advances in Soft Computing, Vol. 2. Springer Verlag, New York (2000)
5. Ge, X.J., Runeson, G.: Modeling Property Prices Using Neural Network Model for Hong Kong. International Real Estate Review (IRER). Vol.7, N.1, 121–138 (2004)
6. González, M.A.S., Soibelman, L., Formoso, C.T.: Explaining Results in a Neural-Mass Appraisal Model. In: 9th European Real Estate Society Conference (ERES). Glasgow, Scotland (2002)
7. Guan, J., Zurada, J., Levitan, A.S.: An Adaptive Neuro-Fuzzy Inference System Based Approach to Real Estate Property Assessment. The Journal of the Real Estate Research. Accepted Papers for Future Publication (2008)
8. Limsombunchai, V., Gan, C., Lee, M.: House Price Prediction: Hedonic Price Model vs. Artificial Neural Network. American Journal of Applied Sciences. Vol.1, N.3. 193–201 (2004)
9. Malpezzi, S.: Hedonic Pricing Models: A Selective and Applied Review. Wisconsin-Madison CULER working papers, University of Wisconsin Center for Urban Land Economic Research, N.5 (2002)
10. Nauck, D., Kruse, R.: Choosing Appropriate Neuro-Fuzzy Models. In: Second European Congress on Fuzzy and Intelligent Technologies (EUFIT94), pp. 552–557. Aachen, Germany (1994)
11. Nguyen, N., Cripps, A.: Predicting Housing Value: A Comparison of Multiple Regression Analysis and Artificial Neural Networks. Journal of Real Estate Research (JRER), Vol.22 N.3, 314–336 (2001)
12. Peterson, S., Flanagan, A.B.: Neural Network Hedonic Pricing Models in Mass Real Estate Appraisal. The Journal of the Real Estate Research. Accepted Papers for Future Publication (2008)
13. Rossini, P.: Improving the Results of Artificial Neural Network Models for Residential Valuation. In: 4th Annual Conference fo the Pacific Rim Real Estate Society (PRRES). Perth, Australia (1998)
14. Sarip, A.G.: Integrating Artificial Neural Networks and GIS for Single-Property Valuation. In: 11th Annual Conference of the Pacific Rim Real Estate Society (PRRES). Melbourne, Australia (2005)
15. Wilson, I.D., Paris, S.D., Ware, J.A., Jenkins, D.H.: Residential Property Price Time Series Forecasting with Neural Networks. Journal of Knowledge-Based Systems. Vol. 15, N.5, 335–341 (2002)
16. Wilson, I., Paris, S., Ware, A., Jenkins, D.: Residential Price Forecasting at National and Regional Levels. In: XXII International Federation of Surveyors (FIG Congress). Washington, DC, USA (2002)
17. Worzala, E., Lenk, M., Silva, A.: An Exploration of Neural Networks and Its Application to Real Estate Valuation. The Journal of the Real Estate Research. Vol.10, N.2, 185–201 (1995)