Non-Conventional Approaches To Property Value Assessment

Jozef M. Zurada, University of Louisville
Alan S. Levitan, (E-mail: levitan@louisville.edu), University of Louisville
Jian Guan, University of Louisville

ABSTRACT

Lack of precision is common in property value assessment. Recently non-conventional methods, such as neural networks based methods, have been introduced in property value assessment as an attempt to better address this lack of precision and uncertainty. Although fuzzy logic has been suggested as another possible solution, no other artificial intelligence methods have been applied to real estate value assessment other than neural network based methods. This paper presents the results of using two new non-conventional methods, fuzzy logic and memory-based reasoning, in evaluating residential property values for a real data set. The paper compares the results with those obtained using neural networks and multiple regression. Methods of feature reduction, such as principal component analysis and variable selection, have also been used for possible improvement of the final results. The results indicate that no single one of the new methods is consistently superior for the given data set.

INTRODUCTION

Assessing the value of real estate is required frequently, for such events as sales, exchanges, donations, or estate taxes. Most commonly, municipalities must establish objective and reliable values for property taxation. Customarily, assessors use a multiple regression program, which assigns weights to numerical and yes/no variables. However, interest in the non-conventional methods for real estate property valuation has increased in the last decade. Most studies looked at neural networks and the results are mixed but the interest in non-conventional methods has clearly been increasing (Guan et al, 1996; Worzala et al., 1995; McGreal et al., 1998; Nguyen et al, 2001; Connellan et al, 1998; Goh Bee-Hua, 2000). More recently some research has suggested use of fuzzy logic-based methods although there has been no work that shows the application of fuzzy logic to assessing real estate property values (Dilmore, 1993; Bagnoli et al, 1998).

Given the apparent academic interest in non-conventional methods for assessing property values more work in this direction is obviously needed. This paper presents the results of using two new non-conventional methods, fuzzy logic and memory-based reasoning, in evaluating residential property values for a real data set. The paper compares the results with those obtained using neural networks and multiple regression. Methods of feature reduction, such as principal component analysis and variable selection, have also been used for possible improvement of the final results. The results indicate that no single one of the new methods is consistently superior for the given data set.

PRIOR RESEARCH

Most research on non-conventional property assessment has focused on the use neural network (NN) based approaches. Guan and Levitan (1996) studied the results of two different NN architectures and those of multivariate linear regression analysis in predicting actual sales values of residential properties. As they noted, it is the multivariate linear regression program CAMA (computer-assisted mass appraisal) that is used by many government authorities to assign comparable market values to properties. They found the NN results to be very similar to those obtained through regression. Moreover, they felt that NN has theoretical appeal, since it does not depend on assumptions about the data (e.g., normality, linearity) and may better replicate a home buyer's heuristic thought processes. Worzala et al.

(1995) compared the results in estimating sales prices for Colorado residential properties of two NN models and a traditional multiple regression model. They did not find the NN models to produce superior results, and furthermore found inconsistent outcomes between different runs of the same NN package. Later, McGreal et al. (1998) found similarly unsatisfactory predictive significance with NN, achieving a value within 15% of the actual sales price in only 80% of the residential properties in their Northern Ireland sample. This was despite the inclusion of supplemental census and environmental attributes, such as traffic noise, view, and attractiveness. Nguyen and Cripps (2001) compared a back propagation NN against the traditional multiple regression (MR) analysis, using a population of 3,906 sold single-family residential properties in Tennessee. NN performed better than MR when a moderate to large sample size was used. While the NN results generally improved with sample size, the MR results remained more constant. Moreover, a larger sample size was needed to obtain useful results from NN as the model functional specification complexity increased. Connellan and James (1998) used NNs to project values of commercial real estate longitudinally forward in time. With 12 input nodes of previous values, they obtained prediction results with less than one percent divergence from the actual success using back propagation in deriving the output node of the most recent value. Instead of actual property values, Goh Bee-Hua (2000) applied NN and genetic algorithm (GA) tools to forecast demand for residential construction in Singapore. He found both models able to produce forecasts with mean absolute percentage errors within 10%. But a combined NN-GA model produced superior results.

Past studies of NN based methods for real estate property assessment have yielded mixed results. While some studies have found no difference between the traditional, multi-regression based methods and the NN based methods, others have found NN based methods to be unsatisfactory. One study has shown better results using NN than those using multiple regression provided the sample size is sufficiently large.

A fuzzy logic framework has also been proposed as an alternative to the traditional, probability-based methods for property assessment (Dilmore, 1993; Bagnoli et al, 1998). Byrne (1995) demonstrated the application of fuzzy analysis software to real estate appraisal. Using a hypothetical case, he compared fuzzy logic and Monte Carlo simulation, finding that both could reduce uncertainty to a limited extent. However, the two studies above were purely conceptual and did not apply fuzzy logic to any real data set.

To the best of our knowledge, neural networks and fuzzy logic are the only two non-conventional approaches that have been applied or suggested for real estate assessment, although such other approaches have been applied to different business problems. One such approach is memory-based reasoning. Memory-based reasoning (MBR) first analyzes the historical data using a rough but efficient model to retrieve a set of relevant similar instances, to which more sophisticated local modeling can be applied. It is explained as one of several major algorithms (along with neural networks) used for market-oriented data mining (Radding 1997). Radding suggests MBR as a tool for real-time fraud detection in long-distance telephone services, using variables such as frequency of calls, time of day, duration, and geography. Radding maintains, however, that, "No technique solves every problem and even the experts aren't always sure which technique will work best in a given situation. Experimentation with multiple techniques is the rule."

Pequeno (1997) also suggests having several options, including NN and MBR, available for detecting telecommunications fraud. Different algorithms will have different levels of success, depending on the particular problem and the data. While NN combines evidence contained in the data in a record, MBR will look for the most closely matching record in a large historical database of records already classified.

Several studies found ensemble models to be more powerful predictors than single models. In forecasting foreign exchange rates, ensemble models consisting of different NN structures consistently outperformed those using the single best network (Zhang and Berardi 2001). And in a medical diagnostic decision support system, ensembles were significantly more accurate than 23 single models for four of five applications studied (Mangiameli et al. 2004).

THE DATA SET

The data used in this study consists of 360 single family home sales in Louisville, Kentucky from 1982 to 1992. These home sales were all from the same neighborhood. Table 1 lists the fields in each sale record together with a sample sale record.

Table 1: Sale Record Structure and Sample Record

Fields in a Sale Record	Sample Sale Record
Street name	Elm Street
Street address	421 Elm Street
Sale price	$68,500
Id	22040700230000
Sale date	00/00/1983
Neighborhood	537
Lot size	1
Construction type	3
Wall type	1
Year built	30
Basement	0
Square footage on the first floor	1373
Square footage on the second floor	667
Upper area	0
Number of baths	3
Presense of central air	0
Number of fireplaces	1
Basement type	1
Garage type	2
Garage size	2

All fields in a property record can potentially serve as input fields except the price. Sixteen of these fields were selected as input (see Table 2). The street name is an ASCII string in the original data record, but it was coded by a unique number in the input record. Table 2 also contains an example of a sale record before it was preprocessed (the middle column) and the corresponding preprocessed record in the right column.

The sale prices in the data set are inflation adjusted before they are used in the study. The following formula was used in adjusting the prices for inflation:

$$Inflation\ Adjusted\ \text{Price} = \text{Price} \times \frac{88798}{Average\ of\ that\ Year}$$

where the average price of each year is as given in Table 3 and 88798 is the average price of the last year in the data set. The sample record in Table 1 has an original sale price of $68,500 and its inflation adjusted price is $107,818 as computed by the above formula.

Table 2: Sample Input Record

Fields Selected As Input	Sample Input Record	Preprocessed Input Record
Street name	Elm Street	1
Neighborhood	537	537
Lot size	1	1
Construction type	3	3
Wall type	1	1
Year built	30	30
Basement	0	0
Square footage on the first floor	1373	1373
Square footage on the second floor	667	667
Upper area	0	0
Number of baths	3	3
Presence of central air	0	0
Number of fireplaces	1	1
Basement type	1	1
Garage type	2	2
Garage size	2	2

Table 3: Inflation Adjustment of Prices

Year	Average Price
1982 and before	$53,092
1983	$56,416
1984	$58,381
1985	$61,376
1986	$62,636
1987	$67,214
1988	$71,699
1989	$76,871
1990	$79,699
1991	$82,892
1992	$88,798

COMPUTER SIMULATION ARCHITECTURE, DATA PREPARATION AND TRANSFORMATION, VARIABLE REDUCTION METHODS, AND BUILDING AND TESTING THE MODELS

Computer Simulation Architecture

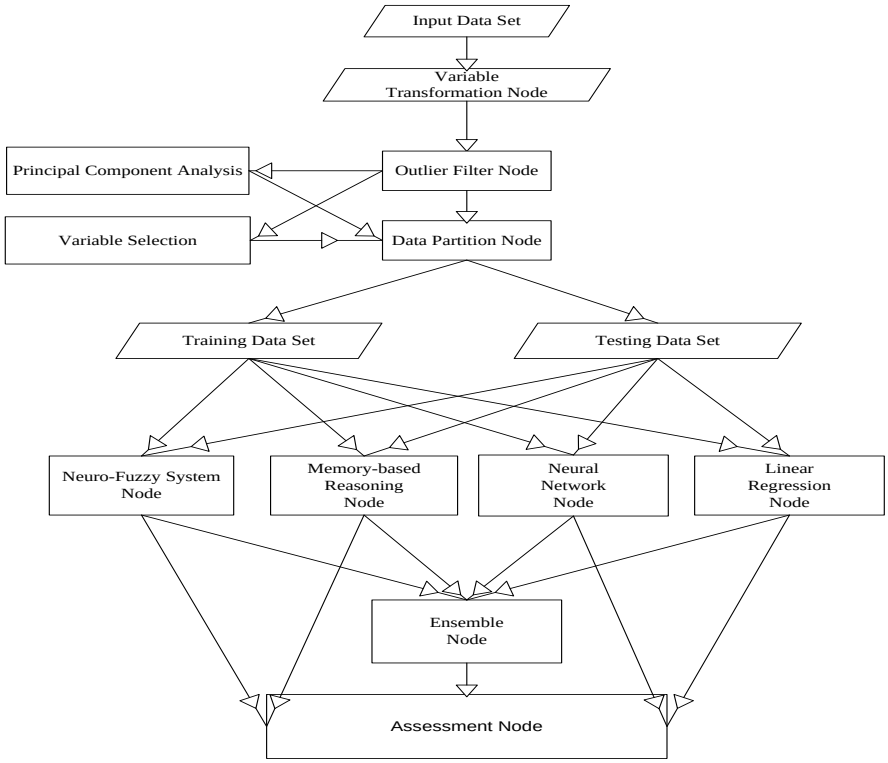
SAS Enterprise Miner has been used to perform the data preparation and transformation, including variable reduction, value reduction for variables and sampling (*k*-fold cross validation), as well as implementation of the three methods: linear regression, neural networks, and memory-based reasoning. The fourth method – based on the neuro-fuzzy system has been constructed using MatLab Fuzzy Logic Toolbox. The ensemble method (combined model) is technically not a data mining method. It is implemented in SAS Enterprise Miner as a separate node. This method simply averages the prediction results from the four previous methods. Figure 1 shows the general architecture of the simulation.

Data Preparation and Transformation

Data preparation and transformation is essential in data analysis using both traditional statistics and data mining methods. It enables one to reduce the dimensionality of the data set and it typically leads to more accurate and stable prediction results (Kantardzic, 2003). The Input Data node reads the original data set containing 360 cases and defines the attributes of the data source for analysis (Figure 1). This node also allows one to examine the distribution

of values for all 16 input variables presented in Table 2. After careful examination of these variables, we removed two input variables from further analysis, i.e., Street Name and Lot Size. The former is a nominal variable that had 26 different categories and the value of the second variable removed, Lot Size, was 1 for all cases but four. The proper coding of the variable Street Name would require 26 additional dummy input variables. This would substantially increase the dimensionality of the data set. In fact, all houses under evaluation were essentially from the same neighborhood of St. Matthews in Louisville. As a result, we concluded that the Street Name variable was redundant. In addition, we eliminated two variables Square Footage on the first floor and Square Footage on the second floor. Instead, we created a new variable called Square Footage to represent the combined square footage on the first and the second floor. The rationale for this transformation can be explained by the fact that only few houses in the data set have a second floor. As a result, the original number of 16 input variables has been reduced to 13 variables.

Figure 1: Architecture of Simulation Model



Next we applied the Variable Transformation node in the SAS Enterprise Miner that enabled us to group values for some of the variables into bins. For example, the Number of baths variable takes values of 1 or 2 for the vast majority of the cases in the data set. In a few cases, however, this variable takes on values within the [3-6] range. The range [3-6] has been collapsed to the value of 3 which means "3 or more bathrooms". Similar transformations have also been performed for the following variables: Neighborhood, Construction type, Number of Fireplaces, Basement Type, and Garage Type.

The Filter Outliers node identifies and removes outliers from a data set. Filtering extreme values from the data tends to produce better models because the parameter estimates are more stable. We identified 8 outliers for which the average Inflation Adjusted Price was outside the range of its ±3 standard deviations. As a result 8 cases were removed so the actual sample size contains 352 cases.

Two different methods have been used for feature/variable reduction and they are represented by the Principal Component Analysis node and the Variable Selection node. This architecture also allows data to be used by the four methods tested for the 13 variables without any feature reduction.

Variable Reduction Methods

Two methods of variable reduction are used and they are R^2 variable reduction and principal component analysis (PCA).

R^2 Variable Reduction Method

In the R^2 variable reduction method an R-square selection criterion is used to remove independent variables that are unrelated to the dependent/target variable. This criterion provides a preliminary variable assessment and facilitates the development of predictive models. It is possible to identify input variables which are useful for predicting the target variable(s) based on a linear model's framework.

The following two-step process is performed when R^2 variable selection criterion is applied:

1. Compute the squared correlation for each variable and then reject those variables that have a value less than the cutoff criterion.
2. Evaluate the remaining significant variables using a forward stepwise regression. Reject variables that have a stepwise R^2 improvement less than the cutoff criterion.

More specifically, the R^2 variable reduction method works as follows. The squared correlation coefficient (simple R^2) between each input variable and the target variable is computed and compared to the squared correlation cutoff criterion of .02 chosen arbitrarily. If the squared correlation coefficient for an input variable is less .02, then the input variable is removed. The squared correlation coefficient is the proportion of target variation explained by a single input variable; the effect of the other input variables is not included in its calculation. It is also referred to in statistics as the coefficient of determination, which ranges from 0 (no linear relationship between an input and the target) to 1 (the input explains all of the target variability). The R^2 selection criterion performs a simple linear regression to obtain the squared correlation coefficient value for interval variables, such as Square Footage of the basement or the Square footage of the upper area of the house. For class (group) variables, such as the Number of baths, the R^2 selection method performs a one-way analysis of variance to calculate the squared correlation coefficient value.

After computing the squared correlation coefficient for each variable, the remaining significant variables are evaluated using a forward stepwise R^2 regression. The sequential forward selection process starts by selecting the input variable that explains the largest amount of variation in the target. This is the variable that has the highest squared correlation coefficient. At each successive step, an additional input variable is chosen that provides the next largest incremental increase in the model R^2. The stepwise process terminates when no remaining input variables can meet the R^2 cutoff criterion (SAS).

We chose the squared correlation cutoff = .02 and stepwise R^2 improvement cutoff=.02. As a result, the 13 input variables were reduced to 6 variables. Obviously, the lower and higher cutoff values retain more and less original input variables, respectively. The .02 cut-off points were mainly dictated by the fact that neuro-fuzzy systems require a more sensible and smaller number of variables to build (train) and operate well (Mathworks). Table 4 presents the 6 variables (in bold) retained for further analysis along with the list of rejected variables (denoted by x) for the 10 different partitions of the training sets. It is not surprising that variables such as Square footage of the

basement and Square footage of the upper floor turned out to be redundant. An analysis of the distribution of these variables reveals that a substantial number of houses under consideration did not have the basement or the upper floor.

Table 4: R^2 variable selection criterion

Cut-off point=0.02. Variable names versus partition numbers used for training. Variables removed are indicated by x and variables retained are in bold.

Variable Name	**Partition #s Used for Training**									
	2-10	1, 3-10	1-2, 4-10	1-3, 5-10	1-4, 6-10	1-5, 7-10	1-6, 8-10	1-7, 9-10	1-8, 10	1-9
Neighborhood	x	x	x	x	x	x	x	x	x	x
Construction type										
Wall type	x	x	x				x	x		
Year built	x	x	x	x	x	x	x	x	x	x
Basement	x	x	x	x	x	x	x	x	x	x
Square footage of the 1st and 2nd floor										
Upper Area	x	x	x	x	x	x	x	x	x	x
Number of baths										
Presence of central air										x
Number of fireplaces										
Basement type	x	x	x	x	x	x	x		x	x
Garage type	x	x			x	x	x	x	x	x
Garage size								x		

4.3.2 Principal Component Analysis

Principal component analysis, known also as the Hôtelling transform, is a linear analysis technique that finds the most efficient representation (in the least-squares sense) of a data set in several dimensions. It is often employed in data representation and data compression tasks, where representing a large data set in a smaller number of dimensions may be desirable. In general, with a set of M vectors x_m, each with N elements such that $x_{m,n}$ represents the n-th element of the m-th vector, PCA finds the linear combinations of the dimensions that encode the greatest proportion (in the least squares sense) of the variance in the data set $\{x_1,\ldots,x_M\}$. PCA computes the zero-mean data set by subtracting the average vector from all data and then finds the covariance matrix C where element $C_{i,j}$ of the matrix is defined as the expected value of the product of elements i and j in any individual vector x_m:

$$C_{i,j}=\langle x_{m,i} \cdot x_{m,j} \rangle$$

The normalized eigenvectors of this matrix, ranked by their corresponding eigenvalues, are the principal components of the data set. Projection onto the first p principal components is the most efficient linear representation of the data in p dimensions. Given a data set of moderate size, this algorithm is relatively robust to noise and is useful in its capacity to combine unrelated measurements into a common statistical framework (Mitchell, 1997).

The 13 input variables were reduced to only 3 principal components. The 3 principal components accounted for 99.98% of the total variation in the data set.

Building and Testing the Models

The Data Partition node partitions the input data into the training and test data sets. The training set is used for preliminary model fitting. The test set is used to obtain a final, unbiased estimate the model.

For problems where the number of samples in the data set is relatively small, the *k*-fold cross validation method has been widely used in practice (Han and Kamber, 2001; Kantardzic, 2003). The method divides the available data set into *P* disjoint partitions/subsets, where $1 \le P \le n$. (*P-1*) subsets are used for training and the remaining subset for testing. This is repeated *P* times. Training and testing subsets are independent and the error estimation is pessimistic. This approach also allows one for better generalization of the results.

In this paper, we used *10*-fold cross validation. First, the order of samples in the data set containing 352 samples was randomized. Next the data set was divided into 10 disjoints partitions/subsets of 35-36 samples each. Nine partitions/subsets (315-317 samples) were used for training and the remaining partition/subset containing approximately 35-36 samples for testing. This has been repeated 10 times. For example, partition 1 contains cases 36-352 for training and cases 1-35 used for testing, whereas partition 10 includes cases 1-317 for training and cases 318-352 for testing. It is clear that different partitioning will give different estimates of error. However, a repetition of the process, with different training and testing sets, and averaging the error results for the testing sets improves the estimate of the models. *10*-fold cross validation technique also improves the robustness of the models and their generalization abilities. The technique has been implemented in the Data Partition node in SAS.

The next four nodes, Neuro-Fuzzy System, Memory-based Reasoning, Neural Network, and Regression, are used to build estimation models. Out of the four, the Neuro-Fuzzy system is the only model implemented in MatLab. The remaining three are constructed using SAS Enterprise Miner. The Memory-based Reasoning node uses a *k*-nearest neighbor algorithm to estimate the given cases. The Neural Network node is used to build a multilayer feed-forward neural network model. The Regression node is used to fit a linear regression model. The Ensemble node is technically not a new model; it averages the estimated values from the four models. The Assessment node provides a common framework to compare models and predictions from the Neuro-Fuzzy System, Memory-based Reasoning, Neural Network, Regression, and Ensemble Model nodes.

We present the results of using four different methods for the above data set. The first is a fuzzy logic method; the second is a memory-based reasoning method; the third is a neural networks method; and the last is a multiple linear regression method. Finally, an ensemble method, a combination of the above four methods, is also tested.

METHODS USED

Fuzzy Logic Method

Fuzzy logic is a theory primarily concerned with quantifying and reasoning using natural language in which words have ambiguous meanings, such as tall, hot, a little, very, etc. Fuzzy logic is a development from the basic theory of fuzzy sets first stated by Lotfi Zadeh (1965). In fuzzy sets an object *x* may belong partially to a set *A*. A fuzzy set *A* is defined by a set of ordered pairs

$$A = \{(x, \mu_A(x)) \mid x \in A, \mu_A(x) \in [0,1]\}$$

where $\mu_A(x)$ is a membership function that specifies the grade or degree to which any element *x* in *A* belongs to the fuzzy set *A*. The definition above associates with each element *x* in *A* a real number $\mu_A(x)$ in the interval [0,1]. *A* is a subset of the set *U* of all objects under consideration called the universe of discourse. In a similar manner, one can define a fuzzy set *B* on the universe of discourse *U*. Membership functions are frequently of a triangular, trapezoidal or Gaussian shape.

The most typical operations on fuzzy sets are the intersection and union operations denoted as $A \cap B$ and $A \cup B$, respectively. For two fuzzy sets *A* and *B*, the intersection operation (AND) and the union operation (OR) would be formally defined as

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)), x \in U$$

and

$$\mu_{A\cup B}(x) = \max(\mu_A(x), \mu_B(x)), x \in U$$

A single fuzzy if-then rule assumes the form

If *x* is *A* then *y* is *B*

where *A* and *B* are linguistic values defined by fuzzy sets on the ranges (universes of discourse) *X* and *Y*, respectively. In a real-estate application, an example of such rule containing several antecedents may be:

If Number of baths is Large AND Square footage of the first and second floor is Large AND …. Then Price is High

In fuzzy logic based-systems, mapping of inputs to outputs is accomplished by using fuzzy sets and fuzzy rules (in which knowledge is encoded), and the process is called fuzzy inference (Zadeh, 1965). It is important to realize that inputs to and output(s) from fuzzy systems are crisp numbers. Using fuzzy inference process, which includes fuzzification and defuzzification, a fuzzy logic-based system can very effectively map inputs to output(s), even if the relationship between them is complex and nonlinear.

In simple applications which (1) involve few variables, (2) a predetermined model structure based on the characteristics of variables is known, and (3) input/output data are not readily available; one can build arbitrary membership functions and fuzzy rules using common sense, intuition and domain knowledge. In cases, as in estimating real estate property values, when a collection of input/output data is available and relationships between variables are complex, one cannot just look at the data and discern what membership functions and fuzzy rules should look like. The technique used to compute the membership functions parameters and build rules based on the given input/output data is called adaptive neuro-fuzzy inference system. MatLab Fuzzy Logic Toolbox has been used to construct the fuzzy inference system for predicting prices of real estate properties in the data set.

The fuzzy inference, or Adaptive-Network-Based Fuzzy Inference System (ANFIS), has a typical structure as shown in Figure 2 (Jang, 1993). Layer 1 consists of membership functions described by generalized bell functions:

$$\mu(X) = (1 + (\frac{X - c}{a})^{2b})^{-1}$$

where *a*, *b* and *c* are adaptable parameters. Layer 2 estimates the firing strength of a rule by ANDing the incoming values. Layer 3 sums and normalizes the firing strength from the previous layer. Layer 4 contains adaptive nodes that are linear combinations of the inputs. Information is propagated forward till layer 4. The results obtained in layer 4 are used to modify the parameters at layer 2. Finally layer 5 produces the output of the ANFIS system through summarization of the inputs from layer 4. The only user specified information is the number of membership functions.

Using ANFIS the system learns from the data it is modeling and builds fuzzy rules and membership functions for each input and output variable using either a back-propagation algorithm alone, or in combination with a least squares type of method. Although we tried approaches with several fuzzy rules and several membership functions for each variable, the resulting neuro-fuzzy system has three rules, in which normalized input variables are connected with the AND logical operator, and each input variable is represented by three overlapping Gaussian membership functions. This approach seems to produce the best prediction results on the test sets.

Memory-based Reasoning Method

Memory-based reasoning is a type of case-based reasoning (CBR). Broadly construed, it is the process of solving new problems based on the solutions of similar past cases. Case-based reasoning stores cases with known solutions in their original or slightly modified form. In solving a new case, a case-based approach retrieves a case it deems sufficiently similar and uses that case as a basis for solving the new case. The new case is solved through a mapping of the new case in the problem domain to a case with a known solution in the solution domain.

One of the simplest methods for mapping a new case to a known case is called the Nearest Neighbor method (Mitchell, 1997). In this approach it is assumed that all cases correspond to points in the *n*-dimensional space R^n. The nearest neighbors of an instance are defined in terms of the standard Euclidean distance. More precisely, let a case *x* be described by the feature vector

$$\{a_1(x), a_2(x), ...a_n(x)\}$$

where $a_r(x)$ denotes the value of the *r*-th attribute of case *x*. First the feature values have been normalized to the [0,1] range using the following formula.

$$a_{r\,norm}(x) = \frac{a_r(x) - \min(a_r(x))}{\max(a_r(x)) - \min(a_r(x))}$$

Then the normalized distance between two cases x_i and x_j is defined to be $d_{norm}(x_i, x_j)$, where

$$d_{norm}(x_i, x_j) \equiv \sqrt{\sum_{r=1}^{n} (a_{r\,norm}(x_i) - a_{r\,norm}(x_j))^2}$$

where *n* is the number of attributes.

The training of a case-based learning process is simply a process of adding the training examples to the list of training examples. For a given case x_q, let $\hat{f}(x_q)$ be the estimate for $f(x_q)$. Then the *k* nearest neighbors for x_q is given as follows

$$\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^{k} w_i f(x_i)}{\sum_{i=1}^{k} w_i}$$

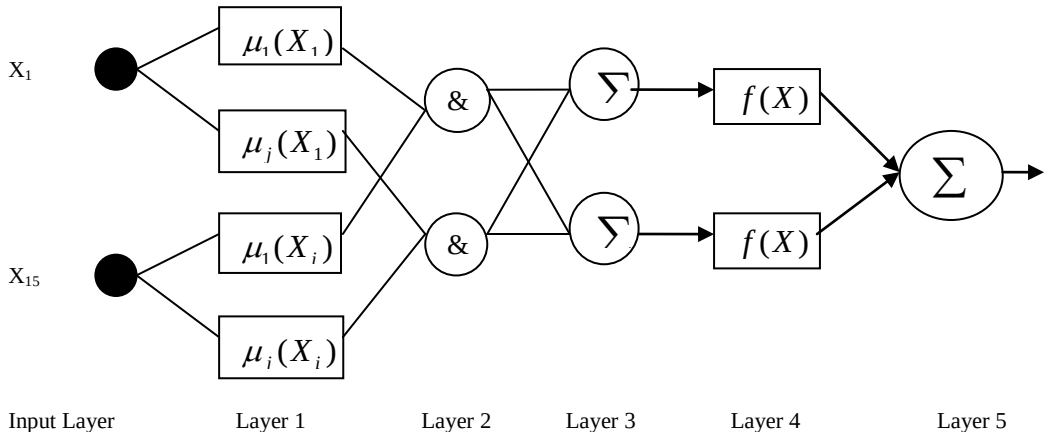
where

$$w_i = \frac{1}{d_{norm}(x_q, x_i)^2}$$

w_i is used to weight the contribution of each of the *k* neighbors according their distance x_q, giving a greater weight to closer neighbors. The denominator is a constant that normalizes the contributions of the various weights.

The memory-based reasoning method requires no model to be fitted, or function to be estimated. Instead it requires all observations to be maintained in memory, and when a prediction is required, the method recalls items from memory and predicts the value of the dependent variable. Two crucial choices in the nearest neighbor-method are the distance function and the cardinality *k* of the neighborhood. After performing several experiments, we chose *k*=10 because this value of *k* seemed to give the lowest root mean square error (RMSE). This means that 10 most similar cases (neighbors) were used to predict the value of the dependent variable.

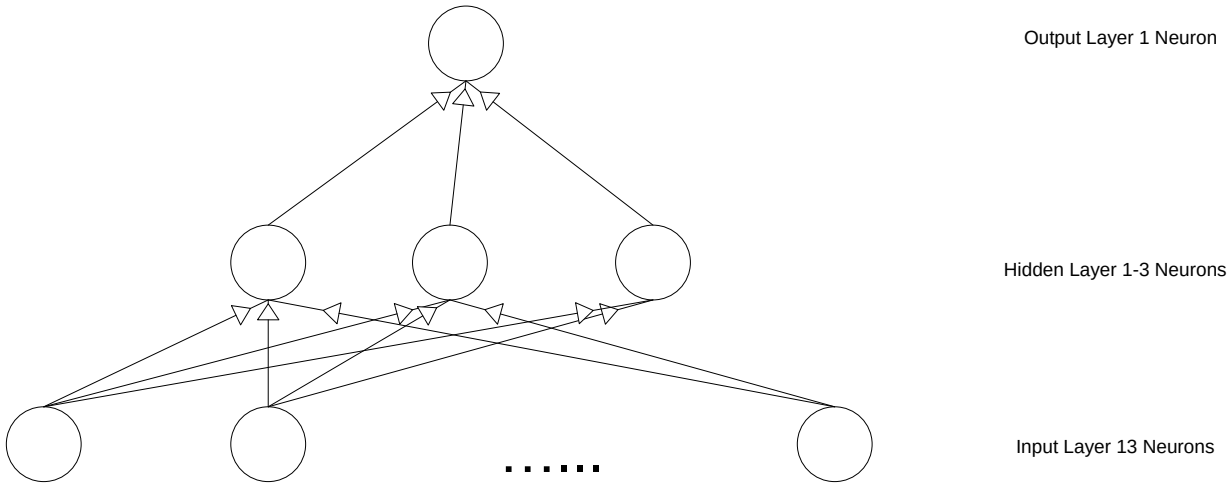
Figure 2: ANFIS Structure



5.3. Neural Network Method

Neural networks methods have been applied extensively to various economic and financial problems. The structure of neural networks models can vary widely. Neural networks, which mimic the way the human brain functions and processes information, are built of nodes or neurons connected by small numerical values called weights. Each neuron is built of the summation node and the activation function. Weights encode knowledge and express the strength of connections between neurons. Neural networks learn through their repeated adjustment. Neural networks are typically organized in three layers: an input layer, a hidden layer, and an output layer. The neural network used in this study is a fully connected perceptron network (see Figure 3) with one hidden layer. The output layer contains just one node, representing the estimated price. The input layer has 13 nodes representing the 13 input values as shown in Table 2. We tested several neural networks with different numbers of neurons in the hidden layer and one neuron in the output layer representing the estimated price. It turned out that the network with 3 neurons in the hidden layer produced the lowest error on the test sets. The hyperbolic tangent activation function was used for neurons. The standard deviation normalization was used for the variables. This normalization subtracts the mean and divides by the standard deviation, so that the resulting values have a mean of zero and a standard deviation of one.

Figure 3: Neural Network



Linear Regression

Linear regression is a time-tested statistical method for determining the relationship between one or more independent variables and a dependent variable. In simple linear regression, built into all popular electronic spreadsheet packages, only one independent variable is used, while in multiple linear regression there are more than one. In all cases, the best linear equation is found, as measured by the least squares method, even if the relationship is actually nonlinear. But the coefficient of determination, R^2, will measure how well the line fits the data points after finding the one which minimizes the sum of the squares of the vertical distances – the residuals – between the line and the points. An R^2 of 100% indicates that the equation explains 100% of the variation in the dependent variable around its mean within the relevant range of the sample. An R^2 of zero indicates that regression can find no relationship between the dependent variable and the independent one(s), or no line that fits any better than any other one.

In general, multiple regression analysis will yield an intercept and a coefficient and a standard error for each of the independent variables. The smaller a variable's standard error relative to its coefficient, the more valuable the corresponding variable is considered to be in its relationship to the independent variable, or the more likely it is that the actual coefficient differs from zero. The ratio of the coefficient to its standard error is called that variable's t-value, the absolute value of which should be at least 2.

A major limitation of regression, compared to the non-conventional approaches discussed, is the strict set of assumptions on the data required in order to use it. First, of course, is that the relationships to the dependent variable must be linear, which is highly unlikely for these real estate variables.

Then, there are three other specifications in the data that should be met in order to use regression. One is constant variance of residuals, meaning that the distances of the points from the line should not be greater in some ranges than in others. That is, the data should not exhibit heteroscedasticity. Another is independence of the residuals, referring to the absence of autocorrelation among proximate points. And finally there is normality of the residuals, meaning that the data points should be denser near the line and sparser at greater distances.

The Ensemble Method

As mentioned above, the ensemble model in SAS Enterprise Miner is represented as an independent node. Technically is not a new method/model; it just averages the predicted values from multiple models. SAS designers consider it, however, as a new model which is then used to score new data. It is important to note that the ensemble model can only be more accurate than the individual models if the individual models disagree with one another. The ensemble or combined model is typically used to improve the stability of disparate non-linear models such as neural network, fuzzy logic, and memory-based reasoning models, and linear models such as regression which were used in the computer simulation (SAS).

RESULTS

To examine the results of the study, two measures are used in comparing the different models. The first measure is the Root Mean Squared Error (RMSE), which is defined as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Actual\_Price_i - Estimated\_Price_i)^2}{N}}$$

where *N* is the number of test records. The second measure is the Maximum Absolute Error (MAE), which is defined as follows:

$$MAE = \max_i |(Actual\_Price_i - Estimated\_Price_i)|$$

where *N* is the number of test records and *i* changes from 1 to *N*.

The results of the simulations using the five methods for the three scenarios (All 13 variables, R^2 for 6 variables, and PCA for 3 principal components) are shown in Table 5. Table 5 presents the average results (RMSE and MAE) for 10 separate test data sets, each containing 35-36 cases. We ran a 2-tailed t-test to check if the average RMSE and MAE values across the different five methods and across the three scenarios are statistically different from each other. As far as RMSE is concerned, we found that the differences in values are not statistically significant. However, some values for MAE appear to be statistically different at .1 significance level (α=.1). Specifically, the values of MAE for the Memory-based reasoning and Neural network methods for the scenario that includes all 13 variables are significantly different. Similarly, the MAE values for the Memory-based reasoning and Linear regression models appear to be statistically different as well for the R^2 scenario containing 6 variables. The results indicate that these non-traditional methods are not superior to multiple regression analysis. A few studies have reported similar results with neural networks (Worzala et al. 1995; Guan et al, 1996).

Table 5: The average RMSE and MAE for the Test Data Set for the Five Methods and Three Scenarios:

All Variables, R^2, and PCA. * - MAE for MBR and NN is significantly different at α=.1 level. # - MAE for MBR and REG is significantly different at α=.1 level. Two-tailed t-test has not shown any significant difference in terms of RMSE across the 5 methods and across the 3 scenarios. (N-FS – neuro-fuzzy system, MBR – memory-based reasoning, NN – neural network, REG – multiple linear regression, ENS – ensemble model. The neuro-fuzzy model was only created for the reduced number of variables, i.e., 6 and 3.)

	RMSE			MAE		
	13 Variables	R^2	PCA	13 Variables	R^2	PCA
N-FS	---	$13,829	$13,620	---	$32,306	$30,512
MBR	$13,576	$14,232	$14,135	$29,846*	$33,908#	$30,621
NN	$14,192	$13,907	$13,830	$34,385*	$31,841	$31,391
REG	$13,538	$13,490	$13,462	$30,745	$30,562#	$31,047
ENS	$13,721	$13,628	$13,518	$31,980	$30,828	$30,286

Below we present a sample regression equation for a model with the reduced number of variables.

Predicted Price = 54718 + 759 * Number of baths + 10.72 * Square footage of the First and Second Floor + 2175 * Garage size – 1694 * Presence of central air (0) + 5084 * Number of fireplaces – 2714 * Construction type (1) – 816 * Garage type (1) + 2024 * Garage type (2).

In the above equation the variables Presence of central air, Construction type, and Garage type are qualitative variables either on the binary or ordinal scale. As a result, in the regression model SAS EM created two dummy variables for the first variable and three dummy variables for the second and third variable each. The presence of central air is coded as 1 and absence as 0; Construction type is coded as 1 or 2; and Garage type takes three values: 1, 2, and 3. For example, if there is no central air (0), construction type is (1) and garage type is 2, $1694 is subtracted, $2714 is subtracted, and $2024 added, respectively, to the predicted price of the property.

CONCLUSIONS

Our results show that there is no single obvious non-conventional method that can be expected to consistently outperform traditional multivariate linear regression in predicting residential real estate sales prices. The results in this study and others before it suggest that, in the least, the non-conventional methods may be used as a complement to the traditional, multiple regression based methods. The results also point to the need for further research in the use of non-conventional, AI methods in assessing real estate property values.

There are several areas that merit further research. The size of the data set is important to effective use of these non-conventional, AI-based methods. Increasing the size of the data set is likely to improve the prediction results. Moreover, we used only the variables included in the local property valuation assessor's regression model. A significant strength of the non-conventional methods is their ability to exploit less quantifiable data, such as the attractiveness of the view from the front window. The inclusion of such "fuzzy" data might well improve the ability to predict selling prices.

REFERENCES

1. Bagnoli, C., Smith, and C. Halbert. 1998. The theory of fuzzy logic and its application to real estate valuation, *The Journal of Real Estate Research* 16 (2): 169-200.
2. Bee-Hua, B. 2000. Evaluating the performance of combining neural networks and genetic algorithms to forecast construction demand: the case of the Singapore residential sector. *Construction Management and Economics* 18 (2): 209-218.
3. Byrne, P. 1995. Fuzzy Analysis: a vague way of dealing with uncertainty in real estate analysis? *Journal of Property Valuation & Investment* 13 (3): 22-41.
4. Connellan, O. and H. James. 1998. Estimated realization price by neural networks: forecasting commercial property values. *Journal of Property Valuation & Investment* 16 (1): 71-86.
5. Dilmore, G. (1993), Fuzzy set theory: an introduction to its application for real estate analysis, American Real Estate Society Annual Conference, Key West, Florida , April 1993.
6. Do, Q. and G. Grudnitski 1992. A neural network approach to residential property appraisal. *The Real Estate Appraiser* 58: 38-45
7. Do, Q. and G. Grudnitski. 1993. A neural network analysis of the effect of age on housing values. *Journal of Real Estate Research* 254-64.
8. Guan, J. and A. Levitan. 1996. Artificial neural network-based assessment of residential real properties: a case study. *Accounting Forum* 20 (3-4): 311-326.
9. Han, J. and M. Kamber. 2001. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers.
10. Jang, J.S.R., 1993. ANFIS: adaptive-network-based fuzzy inference systems. *IEEE Transactions on Systems, Man, and Cybernetics* 23 3, pp. 665–685
11. Kantardzic, M. 2003. *Data Mining: Concepts, Models, Methods, and Algorithms*. IEEE Press/Wiley
12. Mangiameli, P., D. West, and R. Rampal. 2004. Model selection for medical diagnosis decision support systems. *Decision Support Systems* 36 (3): 247-259.
13. Mathworks www.mathworks.com
14. McCluskey, W. and *R.* Borst. 1997. An evaluation of MRA, comparable sales analysis, and ANNs for the mass appraisal of residential properties in Northern Ireland, *Assessment Journal* 4(1): 47-55.
15. McGreal, S., A. Adair, D. McBurney, and D. Patterson. 1998. Neural networks: the prediction of residential values. *Journal of Property Valuation & Investment* 16 (1): 57-70.
16. Mitchell, T. 1997. *Machine Learning*. New York: McGraw-Hill.
17. Nguyen, N. and A. Cripps. 2001. Predicting housing value: a comparison of multiple regression analysis and artificial neural networks. *The Journal of Real Estate Research* 22 (3): 313-336.
18. Pequeno, K. 1997. Real-time fraud detection: Telecom's next big step. *Telecommunications* 31 (5): 59-60.
19. Radding, A. 1997. Unpacking the mystery of the black box. *Software Magazine*, December 1997: S8-S9.
20. SAS www.sas.com
21. Shank, R. 1982. *Dynamic Memory: A Theory of Learning in Computers and People*. New York: Cambridge University Press.
22. Worzala, E., M. Lenk, and A. Silva. 1995. An exploration of neural networks and its application to real estate valuation. *The Journal of Real Estate Research* 10 (2): 185-201.
23. Zadeh, L 1965. Fuzzy sets, *Information and Control* 8(3): 338-353.
24. Zhang, G. and V. Berardi. 2001. Time series forecasting with neural network ensembles: An application for exchange rate prediction. *The Journal of the Operational Research Society* 52 (6): 652-664.