Department of Computer Science
Georgia State University

Data Mining Course Project

House Price Prediction

8 April 2015

Prepared by:

Prakash Chourasia 002-13-7115

Shrada Pradhan 002-23-3231

Problem Definition:

The aim of the project was to predict the price of the houses on the basis of nine attributes and then determine whether the house price is expensive or not. The problem is relevant in real world application as real-state is a large industry and it requires a lot of analysis and evaluation. One of the many functions of a real estate agents and companies is to suggest best matching investment as per to get best ROI. The number of possibilities can be numerous and finding the matching property can be time consuming. In this case, prediction of ROI and other factors may be helpful. In order to make the prediction many data mining techniques can be implemented amongst which the two of them we implemented.

Once the house prices are predicted it can be seen that price was expensive and whether a customer would buy or not.

Problem Formulation:

We extracted freely available data from http://lib.stat.cmu.edu/datasets/ under the subtitle: houses. Zip [1]. Assuming that those 20000 instances of data were preprocessed we implemented KNN algorithm to predict the price of the house and later implemented ID3 algorithm to construct a decision tree classifying whether a person will buy that house or not on the basis of the expensiveness.

For KNN algorithm, a series of different values of k were tested and the results were compared amongst one another.

Data Description

The 20000 instances of data had eight attributes in which latitude and longitude attributes were removed as they didn't contribute much in the analysis of the problem. Those eight attributes are listed as:

1. Median income of individuals living in block. Number merely correlates to income
2. Salary
3. Housing median age i.e. median house age,
4. Total rooms in block,
5. Total bedrooms in block,
6. Population in block,
7. Number of households in block
8. Latitude
9. 8. Longitude

Those eight attribute data are numeric data. As the dataset were collected of a particular zone of California, the values for Latitude and Longitude were similar. And, therefore it did not

contributed much with the processing and analysis of the problem. So, during the classification problem formulation it was discarded.

In the above data, the price value is already averaged. Nevertheless, each block was viewed as a unit itself and not each house unit. Due to this the house prices were categorized into two groups i.e. expensive or cheap, thereby saying whether a person will buy the house or not. So, we added another attribute say 'Classifier' in the dataset for classification algorithm implementation. Thereby, giving 1 nominal class and 8 numeric attributes.

Algorithms implemented

1. **KNN algorithm**
2. **ID3 algorithm for Decision tree**
3. **Neural Network**

KNN algorithm: Imputation of the data with k-nearest neighboring algorithm.

- K-nearest neighbor can predict both discrete attributes (the most frequent value among the k nearest neighbors) and continuous attributes (the mean among the k nearest neighbors).
- There is no necessity for creating a predictive model for each attribute with missing data. Actually, the k-nearest neighbor does not create explicit models (like a decision tree or a set of rules). Thus, the k-nearest neighbor can be easily adapted to work with any attribute as class, by just modifying which attributes will be considered in the distance metric. Also, this approach can easily treat examples with multiple missing values.

The limitation of KNN algorithm:

- Whenever the k-nearest neighbor looks for the most similar instances, the algorithm searches through all the data set. This limitation can be very critical for KDD, since this research area has, as one of its main objectives, the analysis of large databases.

ID3 algorithm for Decision Tree:

Decision tree is a flowchart like structure, where each internal node (nonleaf node) denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (or terminal node) holds a class label. The top-most node is the root node. One of the algorithms to build a decision tree is ID3 algorithm. It uses top down recursive divide and conquer approach which begins with training set of tuples and their associated class. The training set is recursively partitioned into smaller subsets as the tree is being built [2] based on the selected attributes. Test attributes are selected on the basis of a heuristic or statistical measure.

ID3 algorithm:

- Select the attribute with the highest information gain

- Let p_i be the probability that an arbitrary tuple in D belongs to class C_i, estimated by $|C_{i,D}|/|D|$
- Expected information (entropy) needed to classify a tuple in D:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

- Information needed (after using A to split D into v partitions) to classify D:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

- Information gained by branching on attribute A

$$Gain(A) = Info(D) - Info_A(D)$$

For the continuous values, split point was evaluated and classification was done based on those best split points.

The implementation of this algorithm was done in C#.

Neural Network

Neural networks are the computational models which work as biological neural network, and they are used for approximation functions that are generally known. It has set of input nodes, hidden nodes and output node. Neural network is generally used when there are huge set of data to be processed and some output is expected.

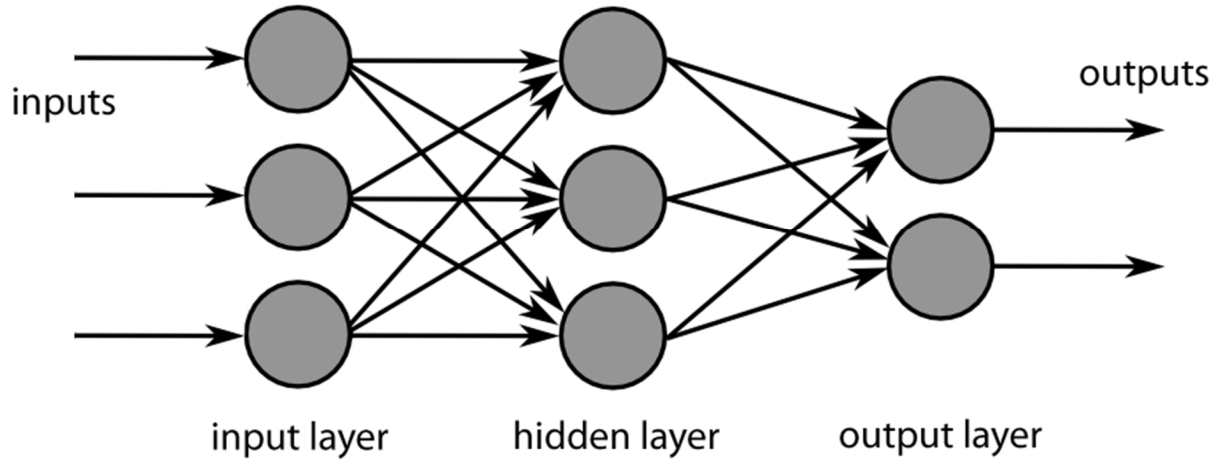

Figure 1. Multilayer neural network

For the implementation of Neural network, input data of n*m cases was taken in which 15% were considered for training, and 15% of them for validating and 1*m size data was used as target data i.e. for the prediction of house price. Different cases were tested, like with the number of hidden neurons, and with different sets of attributes.

Evaluation Methods

Evaluation method for KNN

There are three possible sets really: training set (training_data.txt), validation set (Expected Result.txt) and testing set (Test_Data.txt). We divided the training data in two files for ease of simplicity (training_data.txt and Training_Data_Attprice.txt). At this point no validation will be used. It is our belief that this would be superfluous.

In the evaluation step, each imputed data set from the imputation step is compared to the original data set in order to measure the performance of the imputation. Three separate metrics are used: one ability metric and two quality metrics. The two quality metrics differ both in what they measure and how they measure it. The first quality metric is a measure of how many of the imputed attribute values that were imputed correctly. In other words, it is a precision metric. The second quality metric is a measure of how much those that were not imputed correctly differ from their correct values, which makes it a distance metric.

We define the ability metric as

R= A"/A'
which equals 0 if all incomplete cases were lost during the imputation (in step 2-1), and 1 if all incomplete cases were imputed.

To define the precision metric, let B' be the number of matching imputed attribute values. Then, the metric can be expressed as
Q = B'/B {if B > 0}
And Q = undefined {if B = 0}
Q equals 0 if all the imputed attribute values are incorrect, and 1 if all are correct.

Finally, we calculate the mean square error of the incorrectly imputed attribute values as

$$MSE = \begin{cases} \dfrac{\sum_i (x_i - \hat{x}_i)^2}{B - B'} & \text{if} \quad B > 0, B' < B \\ \text{undefined} & \text{if} \quad B = 0 \text{ or } B' = B \end{cases}$$

Where x_i is the correct value and is the imputed value of the ith incorrectly imputed attribute value. Since B = 0 when R = 0, it is apparent that both the precision measure and the mean square error are invalid when the ability measure is zero. Moreover, the mean square error becomes invalid when Q = 1. Consequently, the three metrics need to have different priorities: R is the primary performance metric, Q is the secondary, and MSE is the tertiary. Recognizing that it would be difficult to create one single metric for measuring the performance, no attempts to accomplish this have been made. Average values of R, Q and MSE are presented in the results, because several imputations are performed with identical parameters (value of k).

Experimental Results / Performance Matrix

K	Precision Metrics Q	Mean Square Error MSE
3	168/600	2.30
5	186/600	1.96
10	197/600	1.65
20	210/600	1.52
40	232/600	1.44

Table 1. Experimental results for KNN

Evaluation of ID3 algorithm:

To evaluate the ID3 algorithm we chosen WEKA. Different test cases for different data sets were created and implemented by the algorithm. For purchasing a house different attributes were taken into consideration and not just the price. So, from the particular sets of data amongst different datasets, following decision tree was deduced.

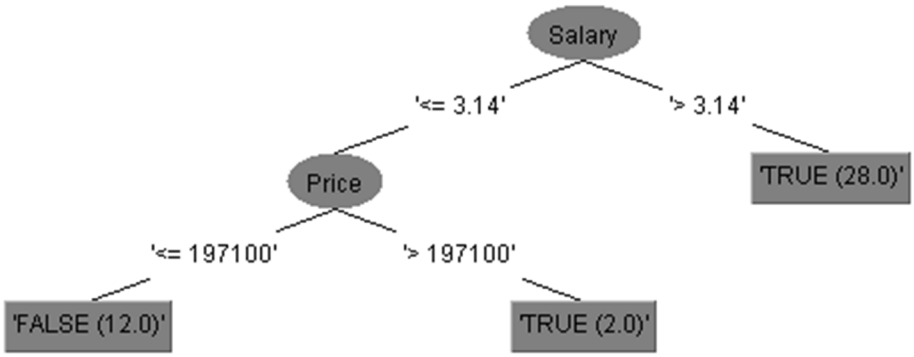

Figure 2 Experimental results by WEKA

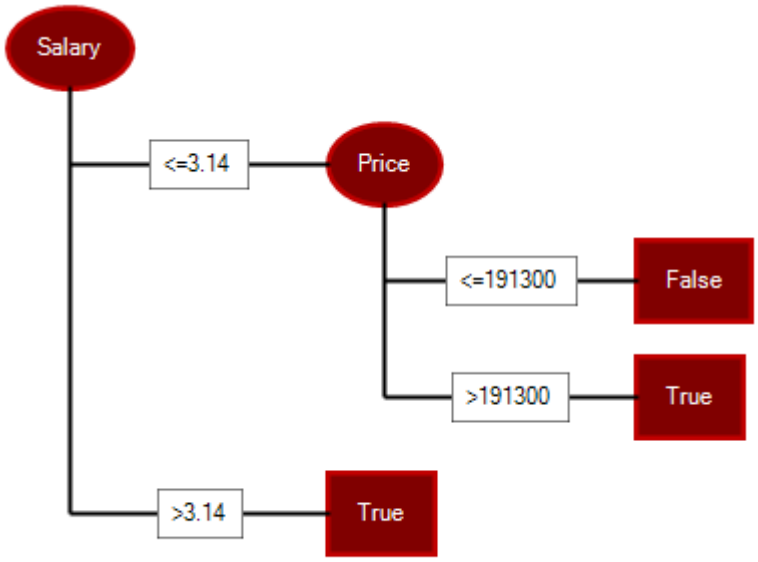

Figure 3. Decision Tree from our implementation

Results of Neural network implementation

No. of Attributes	Hidden Layers	Normalized rmse
9	20	0.10402
	40	0.10342
	100	0.10144
7	20	0.10263
	40	0.10513
	100	0.1032

Conclusion

We came to conclusion that for KNN, with increasing the value of k, MSE is reduced and Qis being increased. The data above is considered with +|- 15%, After testing the various learning algorithms, including our own code, one can confidently say that data mining can be used to successfully predict house values in California from the nine attributes described above. Also, successful deduction of Id3 algorithm was deduced which generated results similar to that generated by WEKA. Different sets of cases were considered for neural network to predict the values of house price. We came to a conclusion that with the higher number of hidden layers lower root mean square was generated thereby giving better results.

Task	Lead Member
Data collection	Prakash Chourasia
Data preprocessing	Shrada Pradhan
Algorithm KNN Impute in MATLAB implementation	Prakash Chourasia
Algorithm evaluation KNN	Prakash Chourasia
Algorithm ID3 in c# implementation	Shrada Pradhan
Algorithm evaluation ID3	Shrada Pradhan
Neural Network implementation in MATLAB	Shrada Pradhan
Writing Report	Prakash | Shrada

Tasks mentioned above in table were led by associated members, however there was equal participation of each member in every task mentioned above.

References:

1. We get the data from http://lib.stat.cmu.edu/datasets/ house.zip
2. http://www.cs.gsu.edu/zcai/course/4740-6740/slides
3. An evaluation of k-nearest neighbor imputation using Likert data, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.14.3558&rep=rep1&type=pdf
4. http://natureofcode.com/book/chapter-10-neural-networks/
5. Batista and M. C. Monard. K-Nearest Neighbour as Imputation Method: Experimental Results (in print). Technical report, ICMC-USP, 2002. ISSN-0103-2569.
   Mental Results (in print). Technical report, ICMC-USP, 2002. ISSN-0103-2569.