

ANALYSIS AND COMPARISON STUDY OF DATA MINING ALGORITHMS USING RAPIDMINER

Thirunavukkarasu K¹ and Dr.Manoj Wadhawa²

¹Galgotias University,India

²Echelon Institute of Technology, India

ABSTRACT

Comparison study of algorithms is very much required before implementing them for the needs of any organization. The comparisons of algorithms are depending on the various parameters such as data frequency, types of data and relationship among the attributes in a given data set. There are number of learning and classifications algorithms are used to analyse, learn patterns and categorize data are available. But the problem is the one to find the best algorithm according to the problem and desired output. The desired result has always been higher accuracy in predicting future values or events from the given dataset. Algorithms taken for the comparisons study are Neural net, SVM, Naïve Bayes, BFT and Decision stump. These top algorithms are most influential data mining algorithms in the research community. These algorithms have been considered and mostly used in the field of knowledge discovery and data mining.

KEYWORDS

Data mining, Machine Learning (ML), Learning algorithms, Classification algorithms Neural Net, SVM, Naive Bayes, BFT, Decision Stump

1. INTRODUCTION

Machine Learning (ML) is improving with its own way that can play a key role in a wide range of critical applications, such as data mining, natural language processing, image recognition, expert systems and decision making system. ML provides potential solutions in all these domains and very much required for our future civilization. It is a subdivision of artificial intelligence and plays a vital role in data mining. However, machine learning can be divided into two subcategories, representation and generalization. Representation refers to defining all instances and functions pertaining to a particular instance. Generalization on the other hand defines the accuracy of a machine to perform well on unseen data after having experienced from a learning data set. In order for a machine to learn from a given data set, it has to be trained in order to accomplish the task. A learning model is developed based on one or more algorithms. The model is given a training dataset to learn and an original data set to perform actual function. Depending on the probability distribution of the data and the frequency the data can be classified and predicted.

Databases have become more complex with the rise of XML trees, spatial data and GPS temporal data. Adaptive techniques should be used in order to effectively process and learn from these types of data sets. However, the question arises on the precision and accuracy of the prediction and classification. The performance of learning algorithms varies on the same dataset input and on similar configuration.

In this paper we have used a single dataset as input to compare the performance with learning algorithms in the same category. The results contrast to a certain extent. This gives us an insight on the effectiveness on learning techniques with preferable configuration. The learning techniques often use the concept of data window which basically limits the amount of data to be processed based on different characteristics. The sensitive part is to know how much data to be processed in order to produce optimum results. Data window comprises of four types: fixed sliding window, adaptive window, landmark window, damped window. The preferred data window is the adaptive window which resizes dynamically on incoming input data.

To process and represent complex input and output relations neural network model proves to be a powerful model. Its major ability is to represent both linear and non-linear relationships from the model data. The neural net model is first trained using a set of data, after which it is capable of performing classification, prediction and analysis.

The Artificial Neural Network [1] is one of the field of study that gains knowledge by mapping and learning data similar to the human brain. The network is interconnected through various nodes known as neurons. However, the performance of the ANN solely depends on the type of network and the number of neurons present in the network. The fewer the number of neurons the higher is the performance of the system.

Support Vector Machine (SVM) [2] has also gained major importance in learning theory because of its ability to represent non-linear relationships efficiently. It focuses on statistical learning theory. SVM maps the data Y into a feature space M of high dimensionality. This technique can be applied to both classification and regression. In the case of classification an efficient hyperplane is found that separates the data in two different classes. Whereas, in the case of regression a hyperplane is constructed which is close to maximum number of data points.

We have taken the issue of weather forecasting to compare Neural net and SVM. The challenge behind weather forecasting [3] is the unpredictability and dynamic change in the atmospheric temperature. There is no particular pattern that can be modelled to predict the temperature from historic data. However, multiple experiments have resulted that neural net and SVM [4], [5] have the capability to model and capture complex non-linear relationships that contribute to a particular temperature.

Naive Bayes classifier [6] is based on Bayes theorem. It simplifies the learning technique by assuming that the features in a particular class are independent. Although, this may not be an optimal method for classification generally, but it proves to be a competent algorithm compared to other sophisticated classification techniques. Naive Bayes has been successful in applications concerning text classification, system performance management and medical diagnosis. Best First Decision tree (BFT) [7] works with the concept of expanding nodes in best first order rather than a fixed order.

Decision stump [8] is a one level decision tree with only a single split. It means there is only one internal node which is connected to leaf nodes. Therefore, a prediction is made based on a single feature of the data set.

In this paper we have compared and analysed classification and learning algorithms in Rapidminer 5. Rapidminer [9] is a powerful open-source tool for data mining, analysis and simulation. The tool provides the user with an environment of rapid application development and appropriate data visualizations.

2. ANN FRAMEWORK

Feed forward neural network using back propagation algorithm [10] is generally used for prediction and classification. Back propagation algorithm is used for simple pattern recognition and mapping. Consider a hidden layer neuron X and output neuron Y and a connection weight WAB between them. There is also a connecting link between neuron A and Z. The algorithm would function as follows:

1. The initial weight being random number, provide the system with appropriate inputs and process the output.
2. Calculate the error for neuron B.

$$\text{Error Y} = \text{Output Y} (1 - \text{Output Y}) (\text{Target Y} - \text{Output Y})$$

3. The weight is adjusted. Let W+XY be the trained weight and WXY be the initial weight.

$$W + XY = WXY + (\text{Error Y} * \text{Error X})$$

4. Calculate the errors for the hidden layer neurons. We Back propagate them from the output layer. This is done by taking the errors from the output neurons and running them back through the weights to get the hidden layer errors.

$$\text{Error X} = \text{Output X} (1 - \text{Output X}) (\text{Error Y} WXY + \text{Error Z} WXZ)$$

5. After calculating the errors of the hidden layer neurons, go to step 3 and repeat the process in order to train the network.

3. SUPPORT VECTOR MACHINE (SVM)

A standard SVM [11] basically takes an input data and predicts which of the two classes would comprise the input. Support vectors are basically essential training tuples and margins (defines by these support vectors). Therefore this makes the SVM a non-probabilistic binary linear classifier. It uses non-linear mapping to transform the training data points into a higher dimensional feature space. SVM constructs hyperplanes in a high dimensional space which can be used for classification. An optimal classification can be achieved when the distance between the hyperplane and closest training data points is largest. The reason is low generalization error of the classifier.

Once we have trained the support vector machine, the classification of data is done on the basis of Lagrangian formulation. The maximum distance hyperplane boundary can be achieved by

$$d(AT) = \sum_{i=1}^l y_i \alpha_i A_i AT + c_0$$

where y_i is the class label for the support vector A_i . AT is the test tuple. α_i (Lagrangian multiplier) and c_0 are numeric parameters which are determined by the SVM algorithm and l is the number of support vectors.

4. NAIVE BAYES CLASSIFIER

Naive Bayes [12] is a simple probabilistic classifier based on Bayesian theorem. It assumes independent functions or features of the data set. The feature in a class is independent of the presence and absence of any other feature. The main advantage of this classifier is that it requires a small amount of training data in order to calculate the mean and variance of the variable for classification. The variances of the label are determined rather than the entire covariance matrix since the features of the class are independent of each other.

Let A be a data tuple and S be a hypothesis such that data tuple A belongs to a class C . In order to classify, $P(S|A)$ has to be determined.

$$P(S|A) = \frac{P(A|S) P(S)}{P(A)}$$

5. DECISION STUMP AND BEST FIRST TREE

A decision tree is a flowchart branched structure. Each internal node represents the test on the attribute. The branch nodes represent the outcome of the test and the leaf nodes hold the class label.

Given a tuple T , with an unknown class label, the attribute values are tested with respect to the decision tree. The traced path from the root node to the leaf node holds the prediction of the tuple. The popularity of the decision tree is based on the construction of tree classifiers without any domain knowledge or any significant parameter setting.

Decision stump is a one-level decision tree with a single split. There is only one internal node and two leaf nodes. Prediction is based on a single input feature. Although, it is a weak learning technique, but it proves to be a fast classification technique when used with boosting [13].

BFT works similar to a standard decision tree in the depth first expansion, but the technique helps us know new tree pruning methods using cross validation. Best first tree order is used for tree construction rather than a predefined fixed order. The best node maximally reduces impurity among other nodes available for splitting (i.e. non labelled leaf nodes).

6. EXPERIMENT AND RESULT

We have performed analysis, simulation and forecasting performance evaluation for the learning algorithms in Rapid miner 5.0 [14]. It is efficient open-source software used for modelling and validating various classifications, learning and rule based algorithms. The Rapid Application Development environment and graphical user interface with flow design makes modelling and simulation easier Rapidminer. The internal process and functions are also and described in XML.

We first compare the performance of Neural net and SVM. The dataset considered consists of average annual and monthly temperature of India from the year 1901 to the year 2012. The dataset is downloaded from the Delhi Meteorological department website [15].

There are six attributes in the dataset, YEAR, ANNUAL, JAN-FEB, MAR-MAY, JUN-SEP, OCT-DEC. The attributes ANNUAL, consists the average annual temperature. Attributes JAN-FEB, MAR-MAY, JUN-SEP, OCT-DEC are the average annual monthly temperature.

6.1. Neural Net Learning model

The following is the neural net model designed in Rapid miner

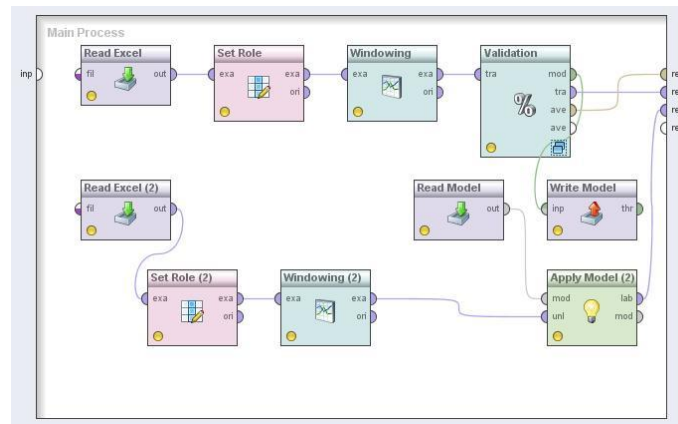


Fig. 1: Model design for Neural Net training in Rapidminer

The input to the model is given in xls format and there is also a training dataset given to the network in order for the system to learn from the given data.

The set role operator changes the role of one or more attributes. In this case we have assigned the YEAR attribute as an ID and ANNUAL as the label. The windowing operator transforms a set of examples into a new single valued example set. The parameters consist of series representation, window size and step size. The series representation is set to the default value (encode series by example). The window size gives the width of the window or the number of examples considered at a time. Window size is set to a value of 2. The step size is basically the gap between the first values. The step size is set to one (default value).

Next, the sliding window validator is used to encapsulate the training and test window examples in order to estimate the performance of a prediction operator. The parameters consist of training window width, training window step size and the test window width and the horizon. The training window width and test window width is set to 7 and the training window size and the step size is set to 1. The horizon gives us the incremented value from first to the last example. In this case horizon value 1 means the prediction of the next example, i.e the prediction for the next average temperature.

The validation operator consists of 2 phases, training and testing. The training phase contains the neural net operator to learn from the data examples and the testing phase applies the model and results the average performance of the system.

The learning rate of the neural net is tuned to 0.8 and we trained the system to tune of 500 cycles. The following was the analysis graph of the predicted and the observed annual temperature.

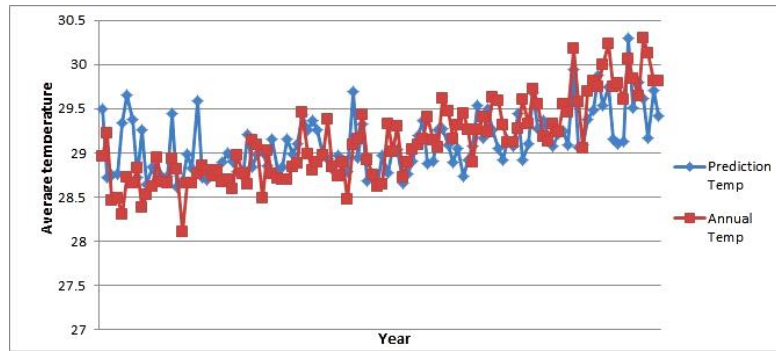


Fig. 2: Comparison graph of predicted temperature by neural Net operator and the original Annual temperature

There were some changes found in the forecasting performance of the neural net model when we tuned the learning rate of the neural net operator.

Table 1 The table projects the learning rate and the corresponding forecasting performance of the Neural Net model.

| Learning rate | Forecasting performance (%) |
|---------------|-----------------------------|
| 0.3 (default) | 67.9 |
| 0.4 | 67.7 |
| 0.5 | 67.5 |
| 0.6 | 68.4 |
| 0.7 | 68.2 |
| 0.8 | 68.7 |

The observed forecasting performance was found to follow a linear trend on changing the learning rate of the neural net. The performance of the system although decreased minutely (from 0.3 to 0.5) but increased by a certain factor thereafter. The graph depicted below shows the performance variation with respect to the learning rate.

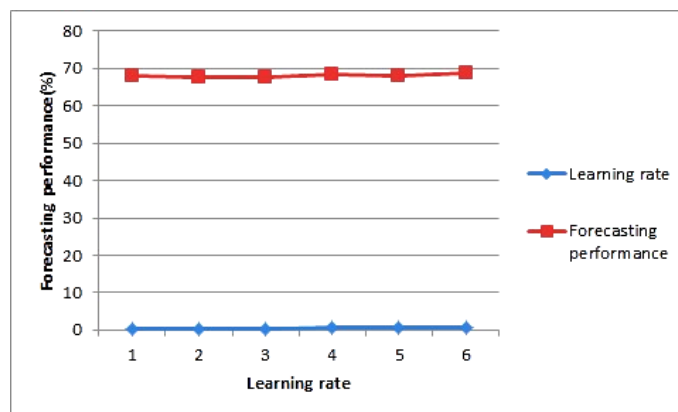


Fig. 3: Learning rate graph for the Neural net model.

6.2. SVM learning model

SVM, initially introduced by Vapnik is basically partitioned into two categories i.e Support Vector Classification (SVC) and Support Vector Regression (SVR). To unveil the prediction and pattern recognition functions from the support vector we make use of Support Vector Regression. The main aim is to minimize the generalization error bound in order to achieve generalized performance. In order to understand the learning mechanism in detail and its origin one can refer to [16].

The overall model of the SVM learner is similar as depicted in Fig.1. But the validation phase holds the SVM regression operator instead of the Neural Net operator. The setup for the general model is same as discussed earlier in section VI. The SVM operator is although tuned on certain parameters. The kernel function of the operator is set to dot type. The kernel function actually helps the data to be mapped into a higher dimensional space. In this case, the dot kernel is defined by the inner product of the support vectors i.e $k(x,y)=x*y$. The SVM complexity constant (C constant) depicts the tolerance level for the misclassification. Higher values for this parameter lead to softer boundaries and lower values lead to harder boundaries. The parameter has to be set carefully since values above optimum level can result in over generalization of data points.

The C constant has been set to a value of 0.1 from the default value of 0.0. We observed that on increasing the value of C constant the forecasting performance slightly reduced due to over generalization. However tuning the value of C higher than 0.2 the forecasting performance slightly declined but followed a linear trend.

Table 2: Table depicts the C constant and the forecasting performance of the SVM model

| C constant | Forecasting performance (%) |
|------------|-----------------------------|
| 0.1 | 71.5 |
| 0.2 | 69.1 |
| 0.3 | 69.1 |
| 0.4 | 68.9 |
| 0.5 | 68.7 |
| 0.6 | 68.4 |

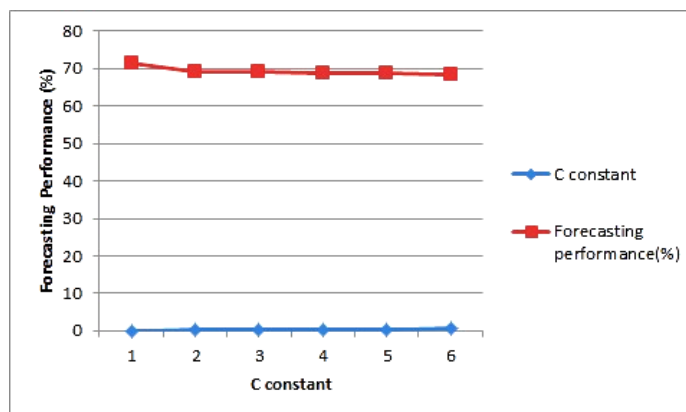


Fig. 4: Forecasting performance graph with respect to C constant in SVM model

On execution of the model with a C constant tuned to 0.1, the forecasting performance was observed to be 71.5% which is higher than the highest performance of the neural net operator.

The following trend was observed for SVR prediction function

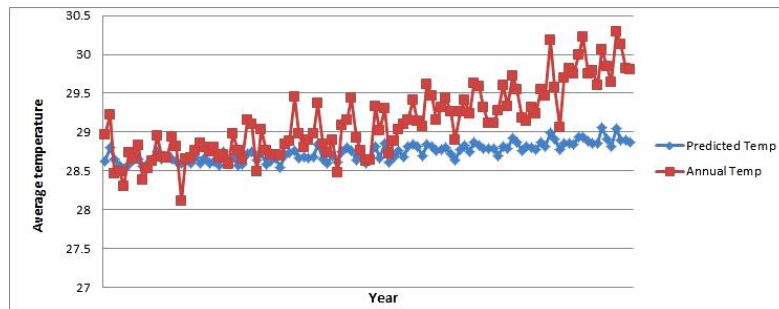


Fig. 5: Comparison graph between predicted temperature of SVM and the actual Annual temperature.

However, the prediction may not be very accurate in determining the average temperature due to lack of a proper pattern and unpredictability of temperature change. But the mapping of non-linear data into a higher dimensional space and the ability to determine patterns is more proficient in SVM than Neural Net.

6.3. Naive Bayes model

For the comparison of classification model we have selected the default dataset available in Rapidminer i.e Golf dataset. The dataset is a sample dataset which can be used for Boolean classification.

The dataset consists of 5 attributes Play, Outlook, Temperature, Humidity, Wind. The attributes Play and Wind consist of Boolean value(True and Flase).Outlook contains 3 values overcast, rain and sunny. Temperature and Humidity attributes consist of real values. According to the dataset, if the day is windy there will be no game that particular day (Play=No if Wind=True). The attributes which are to be classified are first selected from the dataset. In this case the attributes Wind and Outlook are selected in order to identify and classify the trend. The algorithm for Naive Bayes processes and calculates the probability for all labels but it selects only maximum valued label from the dataset. For instance the calculation for label= yes would be following: posterior probability of label = yes (i.e. 9/14) value from distribution table when Outlook = rain and label = yes (i.e. 0.331) value from distribution table when Wind = true and label = yes (i.e. 0.333) Thus the answer = $9/14 * 0.331 * 0.333 = 0.071$.

The Naive Bayes model design is depicted below

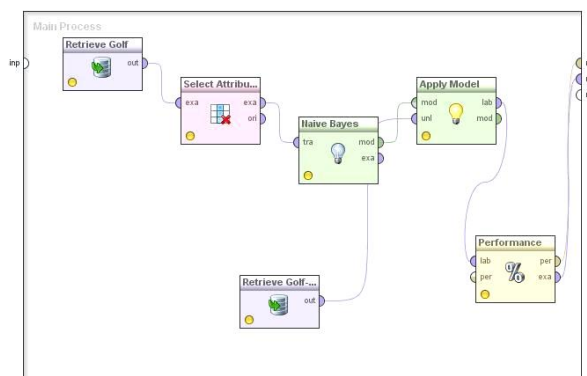


Fig. 6: Naive Bayes process model in Rapidminer.

The Select Attribute operator selects just Outlook and Wind attributes. The I Bayes operator processes it and the resulting model is applied on the 'Golf-testset' data set. A breakpoint is inserted after the I Bayes operator. In the given dataset 9 out of 14 examples of the training set have label = yes, thus the posterior probability of the label = yes is 9/14. Similarly the posterior probability of the label = no is 5/14. However, in the testing set, the attributes of the first example are Outlook = sunny and Wind = false. The following is the performance vector for the Naive Bayes operator. Here the confusion matrix [17] depicts the actual and predicted classification done by the classification operator.

PerformanceVector

```

PerformanceVector:
accuracy: 57.14%
ConfusionMatrix:
True:  no   yes
no:    2    3
yes:   3    6
precision: 66.67% (positive class: yes)
ConfusionMatrix:
True:  no   yes
no:    2    3
yes:   3    6
recall: 66.67% (positive class: yes)
ConfusionMatrix:
True:  no   yes
no:    2    3
yes:   3    6
AUC (optimistic): 0.667 (positive class: yes)
AUC: 0.433 (positive class: yes)
AUC (pessimistic): 0.333 (positive class: yes)
    
```

Fig. 7: Performance vector statistics for Naive Bayes process model.

The calculation process for the accuracy, recall, confusion matrix, AUC (optimistic and pessimistic can be referred from [17]. Overall Naive Bayes is an easy algorithm to implement. It performs surprisingly well assuming that labels of class are independent. But, the performance of the algorithm is appreciable in case when the training data is small.

6.4. Best First decision Tree model (BFT)

The design of BFT model is similar to the Naive Bayes model. The input dataset is the earlier used Golf dataset.

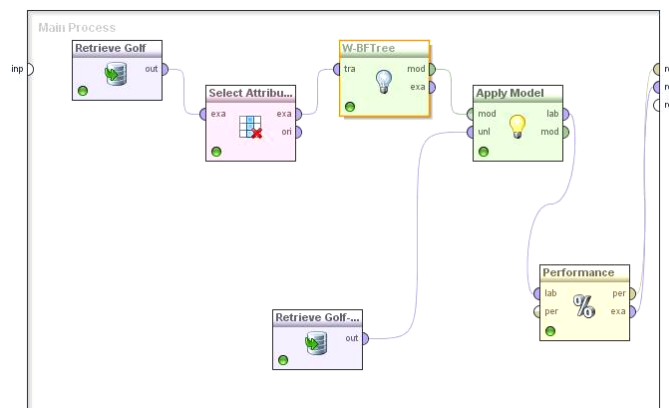


Fig. 8: BFT process model in Rapidminer.

BFT operator is a class for building a best-first decision tree classifier. This class uses binary split for both nominal and numeric attributes. For missing values, fractional instance method is used.

PerformanceVector

```

PerformanceVector:
accuracy: 64.29%
ConfusionMatrix:
True:  no   yes
no:    0    0
yes:   5    9
precision: 64.29% (positive class: yes)
ConfusionMatrix:
True:  no   yes
no:    0    0
yes:   5    9
recall: 100.00% (positive class: yes)
ConfusionMatrix:
True:  no   yes
no:    0    0
yes:   5    9
AUC (optimistic): 1.000 (positive class: yes)
AUC: 0.500 (positive class: yes)
AUC (pessimistic): 0.000 (positive class: yes)
    
```

Fig. 9: Performance vector statistics for Best-First decision tree process model.

6.5. Decision stump model

Our next experiment is to compare decision stump. A decision stump is basically a one-level decision tree. It consists of a single node and 2 leaves. The prediction is made on the basis of just a single input feature.

In this experiment we take a sample dataset from Rapidminer called Labour-Negotiations. The dataset consists of 17 attributes with some missing values. The decision tree splits the table into 2 classes good and bad according to the wage, long-term disability assistance, working hours and statutory holidays. The minimal gain i.e the parameter that controls the size of the tree is set to a value of 0.1. The gain of a node is calculated before splitting a tree. A very high value in this parameter can lead to very few splits but can reduce the performance of the classification. When we split the dataset in the form of a decision tree with a limit of 4 splits and 2 minimal leaves, the following decision tree is formed

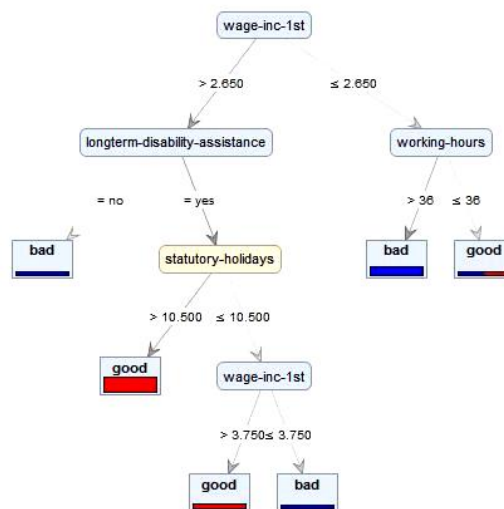


Fig. 10: Decision tree structure for Labour-Negotiations dataset.

Now, feeding the same dataset to the decision stump operator would result in the following decision tree

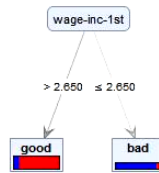


Fig. 11: Decision stump structure for Labour-negotiations dataset with classes ‘good’ & ‘bad’

In the first decision tree the wage-inc-1st is classified into various levels of long-term-disability, working hours, statutory holidays. The classification of labels is performed according to values that are greater, equal or smaller to a certain limit given in the dataset.

The decision stump clearly generalizes, gives the prediction and classifies the data points to predefined classes. The classification is done on the basis of an input feature. The input feature can be selected by the user or it can be suggested by other learning techniques. This classification technique can be used in association with other learning techniques to analyse, predict and classify time series and patterns. This technique is much faster, generalized and easy to implement compared to other types of decision trees.

7. CONCLUSIONS

We have performed experiments and compared learning algorithms including SVM, Naive Bayes, BFT and Decision stump. The results portrayed that both Neural Net and SVM although are capable to model both linear and non-linear datasets. The Neural Net is more proficient when it comes to model complex datasets whereas SVM is more successful in classifying and modelling non-linear relationships. The mapping of features into a higher dimensional space is efficient yet useful while predicting and recognizing patterns in SVM. The performance of SVM was found to be better than Neural Net while predicting. Naive Bayes is a simple classifier based on Bayes theorem. It is simple to implement and proves to be successful and competent against various classification algorithms. The accuracy of Naive Bayes was slightly lower than BFT. Naive Bayes though is easy to implement and efficient, it can sometimes deviate from optimum performance when the size of the dataset exceptionally increases. Decision Stump is perhaps a generalizing tree structure that uses a single node and 2 leaves to represent and classify data points. It proves to be quite efficient if the feature selection by the user is apt for optimum classification. These algorithms, if allowed to perform in the right combination can be more efficient in learning, feature selection and classification.

We plan to analyse and optimize learning algorithms of fuzzy datasets. This will help researcher further to plan and provide a framework for learning and decision making from complex datasets.

ACKNOWLEDGEMENTS

I would like to thank Dr. Manoj Wadhawa for being my guide and mentor and sincere and dedicated student Mr. Sharavan Vishwanathan, now working for Infoys.

REFERENCES

- [1] Lippmann, Richard. "An introduction to computing with neural nets." *ASSP Magazine, IEEE* 4.2 (1987): 4-22.
- [2] Hearst, Marti A., et al. "Support vector machines." *Intelligent Systems and their Applications, IEEE* 13.4 (1998): 18-28.
- [3] Firth, W. J. "Chaos--predicting the unpredictable." *BMJ: British Medical Journal* 303.6817 (1991): 1565.
- [4] Baboo, S. Santhosh, and I. Kadar Shereef. "An efficient weather forecasting system using artificial neural network." *International journal of environmental science and development* 1.4 (2010): 2010-0264. R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, "High-speed digital-to-RF converter," U.S. Patent 5 668 842, Sept. 16, 1997.
- [5] Radhika, Y., and M. Shashi. "Atmospheric temperature prediction using support vector machines." *International Journal of Computer Theory and Engineering* 1.1 (2009): 1793-8201.
- [6] Rish, Irina. "An empirical study of the naive Bayes classifier." *IJCAI 2001 workshop on empirical methods in artificial intelligence*. Vol. 3. No. 22. 2001. FLEXChip Signal Processor (MC68175/D), Motorola, 1996.
- [7] Shi, Haijian. *Best-first decision tree learning*. Diss. The University of Waikato, 2007.
- [8] Witten, Ian H., et al. "Weka: Practical machine learning tools and techniques with Java implementations." (1999).
- [9] RapidMiner, RapidMiner. "Open Source Data Mining." (2009).
- [10] White, Halbert. "Learning in artificial neural networks: A statistical perspective." *Neural computation* 1.4 (1989): 425-464.
- [11] T. Euler. *Publishing Operational Models of Data Mining Case Studies*. In *Proceedings of the Workshop on Data Mining Case Studies at the 5th IEEE International Conference on Data Mining (ICDM)*, pages 99–106, Houston, Texas, USA, 2005.
- [12] Dr. Rajeshwari S. Mathad, "Supervised Learning in Artificial Neural Networks", *International Journal of Advanced Research in Engineering & Technology (IJARET)*, Volume 5, Issue 3, 2014, pp. 208 - 215, ISSN Print: 0976-6480, ISSN Online: 0976-6499.
- [13] Dr. Rajeshwari S. Mathad, "Supervised Learning in Artificial Neural Networks", *International Journal of Advanced Research in Engineering & Technology (IJARET)*, Volume 5, Issue 3, 2014, pp. 208 - 215, ISSN Print: 0976-6480, ISSN Online: 0976-6499.
- [14] Sandip S. Patil and Asha P. Chaudhari, "Classification of Emotions from Text using SVM Based Opinion Mining", *International Journal of Computer Engineering & Technology (IJCET)*, Volume 3, Issue 1, 012, pp. 330 - 338, ISSN Print: 0976 - 6367, ISSN Online: 0976 - 6375.
- [15] Tarun Dhar Diwan and Upasana Sinha, "The Machine Learning Method Regarding Efficient Soft Computing and ICT using SVM", *International Journal of Computer Engineering & Technology (IJCET)*, Volume 4, Issue 1, 2013, pp. 124 - 130, ISSN Print: 0976 - 6367, ISSN Online: 0976 - 6375.
- [16] V.Anandhi and Dr.R.Manicka Chezian, "Comparison of the Forecasting Techniques –ARIMA, ANN and SVM - A Review", *International Journal of Computer Engineering & Technology (IJCET)*, Volume 4, Issue 3, 2013, pp. 370 - 376, ISSN Print: 0976 - 6367, ISSN Online: 0976 - 6375.

AUTHORS

Thirunavukkarasu K., is an Assistant Professor at Galgotias University, Greater Noida, Delhi-NCR. He is pursuing PhD in CSE in the research area of Spatial Database. He has more than 14 years of experience in Teaching and 3 years in software industry. He has taught for APIIT, (affiliated to Staffordshire University, UK) at Panipat, Vijaya College, Surana College and KKECS College, Bangalore University, Bangalore and worked as Software Engineer for I2 Technology, at Bangalore. He has involved in various academic activities like BoE member and Assistant Custodian for PG-Unit, Bangalore University, Bangalore. He has wide research interests that include Knowledge Engineering, Data Mining, and Databases Technology. He is a Member of IEEE, CSI and Life Member of ISTE. He has published many research papers in international as well as national level.



Prof (Dr.) Manoj Wadhawa, is working as Professor and HoD of Computer Science & Engineering, Echelon Institute of Engineering and Technology, Faridabad, Delhi-NCR and has more than 15 years of experience in teaching and research. He has obtained his Ph.D and M.Tech. from Kurukshetra University. He is a professional member of IACSIT, Singapore, Life Member of Indian Society of Technical Education, Delhi (ISTE), Member of Institute of Electrical and Electronics Engineers, USA (IEEE) and Member of Computer Society of India (CSI). He has published many research papers in international as well as national level. He has a book titled “Fundamental of Computers” ISBN 9789381355147 by International Book House Pvt Ltd., Delhi.

