Advances in Computational Research

Advances in Computational Research
ISSN: 0975–3273 & E-ISSN: 0975–9085, Volume 3, Issue 1, 2011, PP-25-30
Available online at http://www.bioinfo.in/contents.php?id=33

A HYBRID STACKING ENSEMBLE FRAMEWORK FOR EMPLOYMENT PREDICTION PROBLEMS

SUDHEEP ELAYIDOM[1]*, SUMAM MARY IDIKKULA[2] AND JOSEPH ALEXANDER[3]
[1]Cochin University of Science and Technology, Computer Science and Engineering Division, School of Engineering,
[2]Cochin University of Science and Technology, Department of Computer Science,
[3]Project officer NODAL Centre, Cochin University of Science and Technology,
*Corresponding Author Email:-sudheepelayidom@hotmail.com

Received: November 16, 2011; Accepted: December 01, 2011

Abstract- In this paper we put forward a hybrid stacking ensemble approach for classifiers which is found to be a better choice than selecting the best base level classifier. This paper also describes and compares various data mining methodologies for the domain called employment prediction. The proposed application helps the prospective students to make wise career decisions. A student enters his Entrance Rank, Gender (M/F), Sector (rural/urban) and Reservation category. Based on the entered information the data mining model will return which branch of study is Excellent, Good, Average or poor for him/her. Various data mining models are prepared, compared and analyzed.
Key words - Confusion matrix, Data Mining, Decision tree, Neural Network, stacking ensemble, voted perceptron

Introduction

Majority of students join a course in engineering for securing a good job. Therefore taking a wise career decision regarding the selection of a particular course or branch is crucial in a student's life. An educational institution contains a large number of student records. Therefore finding patterns and characteristics in this large amount of data is a difficult task. So data mining techniques can be applied using neural network, Decision tree and Naïve Bayes classifier to interpret potential and useful knowledge.

With the help of this knowledge a student enters his/her rank, branch, location etc. and on the basis of which the placement chances for different streams of study are calculated. Now a student on the basis of this inference may decide to opt for branch giving excellent chances of placement.

It has been an active research area in data mining to select the most suited data mining model for a given problem. At the same time in the area of supervised learning it has been an active debate whether combining classifiers gives a better performance than selecting the best base level classifier. This paper is an attempt to explore this research problem in the area of technical manpower analysis and interestingly it is being observed that an ensemble of classifiers is giving a better performance than selecting the best base level classifier.

There are some algorithms for extracting comprehensible representations from neural networks. [1] Describes research to generalize and extend the capabilities of these algorithms. The application of the data mining technology based on neural network is vast. One such area of application is in the design of mechanical structure.[2] introduces one such application of the data mining based on neural network to analyze the effects of structural technological parameters on stress in the weld region of the shield engine rotor in a submarine. Prediction of Beta-Turns using global adaptive techniques from multiple alignments in Neural Networks has been studied in [3] in study of proteins. This also introduces global adaptive techniques like Conjugate gradient method, Preconditioned Conjugate gradient method etc. This paper is an attempt that uses the neural network based on back propagation training for placement prediction which uses the above said concepts with more application in the domain of data mining.

Decision trees have proved to be valuable tools for the description, classification and generalization of data. Work on constructing decision trees from data exists in multiple disciplines such as statistics, pattern recognition, decision theory, signal processing, machine learning and artificial neural networks. [5] Surveys existing work on decision tree construction, attempting to identify the important issues involved, directions the work has taken and the current state of the art. Studies have been conducted in similar area such as understanding student data as in [6]. There they apply and evaluate a decision tree algorithm to university records, producing graphs that are useful both for predicting graduation, and finding factors that lead to graduation. It's always been an active debate over which engineering branch is in demand .So this work gives a scientific solution to answer these [7, 8] are papers which explain the possibilities of combining

data mining models to get better results. The method in [9] is used for performance evaluation of the system using confusion matrix which contains information about actual and predicted classifications done by a classification system. [10, 11] are publications by same authors describing the data preprocessing and applicability of various data mining models.

Problem Statement

To propose the most suitable data mining model which can predict the most suited branch for a student who supplies his information. The problem includes deciding which the attributes that decide placement chances are. Various data mining models are to be trained and tested for this problem. Their performances are to be compared based on statistical measures. It is to be investigated whether combining models is better than selecting the best base model and if so, propose the best ensemble.

Stacking framework

Stacking is the combining process of multiple classifiers generated by using different learning algorithms $L_1 \ldots L_n$ on a single dataset. In the first phase a set of base level classifiers C_1, $C_2 \ldots C_n$ is generated. In the second phase a Meta level classifier is learned that combines the base level classifiers. The basic difference between stacking and voting is that in voting no learning takes place at the meta level, as the final classification is by votes casted by the base level classifiers whereas in stacking learning takes place in the meta level. The WEKA data mining package is used for implementing and testing the stacking approach.

The two active research areas in data mining nowadays are optimization problems which try to optimize a single data mining model which is already existing and the other being combining those models in an optimized way. The goal of both these approaches is improving model performance. Usually these problems are studied on a particular domain and results are consolidated.

A methodology on how models can be combined for customer behavior is described in [13]. EC companies are eager to learn about their customers using data mining technologies. But the diverse situations of such companies make it difficult to know which the most effective algorithm for the given problems is. Recently, a movement towards combining multiple classifiers has emerged to improve classification results. In [13], a method for the prediction of the EC customer's purchase behavior by combining multiple classifiers based on genetic algorithm, have been proposed.

One approach in combining models is called Meta decision trees, which deals with combining a single type of classifier called decision tree. [14] Introduces Meta decision trees (MDTs), a novel method for combining multiple models. Instead of giving a prediction, MDT leaves specify which model should be used to obtain a prediction.

Another approach in improving classifier performances have been in studies of applying special algorithms like genetic algorithms[13] and fuzzy logic[15] concepts into classifiers, which were found to be successful in improving accuracies as explained in the literature.

In this paper more focus is on how classifier performance can be improved using a stacking approach. Conventional data mining research focuses on how we can improve a single model, rather here it focuses more on heterogeneous classifiers and combining them. It has been also observed that this approach yields better accuracy in the domain of employment prediction problems.

There are many strategies for combing classifiers like voting, bagging and boosting each of which may not involve much learning in the Meta or combing phase. Stacking is a parallel combination of classifiers in which all the classifiers are executed parallel and learning takes place at the Meta level. Cascading involves sequential combination, but owing to its accuracy stacking has become more popular. To decide which model or algorithm performs best at the Meta level for a given problem is also an active research area, which is addressed in this paper. It has been always a debate that whether an ensemble of homogenous or heterogeneous classifiers yields good performance. [17] Proposes that depending on a particular application an optimal combination of heterogeneous classifiers seems to perform better than the homogenous classifiers.

When we select only the best classifier among the base level classifiers, the valuable information provided by other classifiers are being ignored. In classifier ensembles which are also known as combiners or committees, the base level classifier performances are combined in some way such as voting or stacking. In [19], it is being observed that when we prepare ensembles the number of base level classifiers is not that much influencing, and usually researchers select 3 or 7 at random.

In [16], Dietterich gave three fundamental reasons for why ensemble methods are able to outperform any single classifier within the ensemble — in terms of statistical, computational and representational issues. Besides, plenty of experimental comparisons have been performed to show significant effectiveness of ensemble. Mathematically, classifier ensembles provide an extra degree of freedom in the classical bias/variance trade off, allowing solutions that would be difficult (if not impossible) to reach with only a single classifier.

The paper is organized as two main parts, one describing the base level classifier modeling and the second one, the combined stacking framework implementation and analysis.

Mathematical insight into stacking ensemble

If an ensemble has M base models having an error rate $e < 1/2$ and if the base models' errors are independent, then the probability that the ensemble makes an error is the probability that more than M/2 base models misclassify the example. This is precisely $P(B > M/2)$, where B is a Binomial (M, e) random variable. In a three-model example, if all the networks have an error rate of

0.3 and make independent errors, then the probability that the ensemble misclassifies a new example is 0.21[18]. The simple idea behind stacking is that if an input–output pair (x, y) is left out of the training set of hi, after training has been completed for hi, the output y can still be used to assess the model's error. In fact, since (x, y) was not in the training set of hi, hi(x) may differ from the desired output y. A new classifier then can be trained to estimate this discrepancy, given by y - h(x). In essence, a second classifier is trained to learn the errors the first classifier has made. Adding the estimated errors to the outputs of the first classifier can provide an improved final classification decision [18].

Base level classifier modeling using decision trees

A decision tree is a popular classification method that results in a tree like structure where each node denotes a test on an attribute value and each branch represents an outcome of the test. The tree leaves represent the classes. The tree consists of zero or more internal nodes and one or more leaf nodes with each internal node being a decision node having two or more child nodes. The decision tree can be modeled using WEKA package, but since the NODEL CENTRE authorities are interested to implement a web site it have been decided to use conventional Web technologies like mySQL, php etc itself to model a decision tree.

Given a set of examples (training data) described by some set of attributes (ex. Sex, rank, background) the goal of the algorithm is to learn the decision function stored in the data and then use it to classify new inputs.

The Initial database provided by Nodal Center, was in FoxBase format, hence it has to be converted to some latest DBMS like mySQL to make the approach efficient and faster. First FoxBase data was converted to CSV files (Comma Separated files) and this file was loaded to MS Excel. Then from this Excel format using xls-mySQL converter it was converted to mySQL format. These attributes were fed into mysql through sql queries and each of these entities and two databases, one containing records of students from the year 2000-2002 and another for year 2003, were created.

List of attributes extracted:

RANK: Rank secured by candidate in the engineering entrance, Rank is distributed between 1 and 3000
CATEGORY: Social background. Range: {General, Scheduled Cast, Scheduled Tribe, Other Backward Class}
SEX: Range {Male, Female}
SECTOR: Range {Urban, Rural}
BRANCH: Range is distributed between A to J
ACTIVITY: Indicator of whether the candidate is placed.

All these attributes have been found to be deciding the placement chance which has been analyzed using chi-square based statistical dependency analysis. There have been certain irrelevant non dependent attributes like "whether training attended", "whether had financial assistance" etc, which were removed after chi-square test. This test is carried out in the statistical package called SPSS. Now the actual training data for the mining process have to be prepared. A new table is created in which the placement chance for each possible input combination is stored. For e.g., if RANK (1-200) SECTOR (U) SEX (M) CATEGORY (GEN), we compute how much percentage of students having these criteria are placed in history database. If this percentage is greater than 95% it is called as "Excellent" chance. Similarly other grades are also assigned corresponding to placement percentages.

Base level classifier modeling using neural networks

Neural Network has the ability to realize pattern recognition and derive meaning from complicated or imprecise data that are too complex to be noticed by either humans or other computer techniques.

The output data used for training is derived from ACTIVITY attribute. Instead of representing the output on 0 to 1 scale basis, a four-fold classification has been used; a 4 value code has been assigned with each record. The data processing is very similar to that used for decision trees. But since neural networks need numeric inputs slight modifications were done.

A code value of 1000 represents a 'excellent' chances of getting a student placed, a code value of 0100, 0010, 0001 represents 'good', 'average' and 'Poor' chances of placement of a student respectively these codes are calculated by following the steps

Step 1:

Calculate the probability of each test case for getting a student placed. It is calculated as

Probability (P) = Number Placed/ Total Number

Where 'Number Placed' is the number of students placed in a particular class of inputs and 'Total Number' is the total number of students in that class. For example let a class of records has: RANK = 0.72, SEX = 1, CATEGORY = 1, SECTOR = 0, BRANCH = 0.47.

Let the number of records be 98 and the number of records having ACTIVITY as 1 (i.e. student is placed) be 79, then probability is given by P = 79/98 = 0.80.

Step 2: Assign the output code to each record as explained above.

This is the most important step in the data mining. A proper selection of algorithm is made on the basis of the required objective of the work. MATLAB has been used to model the neural network.

One of the most popular Neural Network model is Perceptron, but this model is limited to classification of linearly separable vectors [4]. The input data obtained from NTMIS, may contain variations resulting in Non-Linear data. To deal with such inputs data cleaning alone is not sufficient. Therefore we go for multilayer perceptron network with supervised learning which gives back propagation Neural Network. A BP neural network reduces the error by propagating the error back to the network. Appropriate model of BP neural network is selected and repeatedly trained with the input data until the error reduces to a fairly low value. At the end of training we get a set of thresholds and weights which determines the architecture of the neural network.

A back propagation neural network model is used consisting of three layers namely input, hidden and output layers. The number of input neurons is 5, which depends upon the number of the input attributes. The number of neurons used in hidden layer is 5; this number is obtained by value based on observations. The transfer function used in the Hidden layer is Log- Sigmoid while that in the output layer is Pure Linear. Training is done by using one of the Back propagation Conjugate Gradient algorithm, Powell-Beale Restarts [4]. This algorithm provides faster convergence in comparison to conventional basic Back propagation algorithm by performing a search along the conjugate direction to determine the step size which minimizes the performance function along that line.

Base level classifier modeling using Naïve Bayes Classifier

Naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. An advantage of the Naive Bayes is that it requires a small amount of training data to estimate the parameters (means and variances of variables) necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix.

The classifier is based on Bayes theorem, which is stated as:

$$P(A|B) = P(B|A)*P(A)/P(B)$$

Each term in Bayes' theorem has a conventional name

*P (*A*) is the prior probability or marginal probability of *A*. It is "prior" in the sense that it does not take into account any information about *B*.

*P (A|B) is the conditional probability of A, given B. It is also called the posterior probability because it is derived from or depends upon the specified value of B.

*P (B|A) is the conditional probability of B given A.

*P (B) is the prior or marginal probability of B, and acts as a normalizing constant.

Weka is a collection of machine learning algorithms for data mining tasks. It is used to model the naïve Bayes classifier. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes. The Initial database provided by Nodal Center was loaded to MS Excel and converted to CSV files (Comma Separated files) .This file has been loaded in Weka Knowledge Flow interface and converted into ARFF files (Attribute-Relation File Format). The data processing remains same as in the decision tree.

Experimental Results

The data used in this project is the data supplied by National Technical Manpower Information System (NTMIS) via Nodal centre. Data is compiled by them from feedback by graduates, post graduates, diploma holders in engineering from various engineering colleges and polytechnics located within the state during the year 2000-2003. This survey of technical manpower information was originally done by the Board of Apprenticeship Training (BOAT) for various individual establishments. The collected data has been entered into FoxBASE data base system, which is a pretty old data base technology and this practice have been there in Nodal centre since inception. This format has to be completely ported to respective formats needed for various data mining models. Each prediction model is prepared from training data during the year 2000-2002 and tested with data from the year 2003. In nodal centre during 2000-2002 data records of 6096 students were collected for analysis and during 2003, 2428 records were collected for analysis. From this data processing has to be done as explained in previous section corresponding to that of decision trees.

Training and testing has been conducted separately for the projects based on the neural network, decision tree and Naïve Bayes classifies models. Accuracy, confusion matrix, and all performance parameters were separately computed. The same test data set has been used for the above three data mining techniques. The negative cases here are when the prediction was Poor /average and the corresponding observed values were Excellent/good and vice versa.

Combining models- A hybrid stacking ensemble approach

Bagging is a voting scheme in which n models, usually same type are constructed and for an unknown instance for each models predictions are recorded. Assign to that class which is having the maximum vote among the predictions from models.

Boosting is very similar to bagging in which only the model construction phase differs where every time those instances which are most misclassified are allowed to participate in training more.....there will be n classifiers which themselves will have individual weights for their accuracies...that class is assigned which is having maximum weight. An example is Adaboost algorithm. Bagging is better than boosting as boosting suffers from over fitting. There are 2 approaches for combining models. One of them uses voting in which the class predicted by majority of the models is selected, whereas in stacking the predictions by each different model for each class for a given instance is given as input for a meta level classifier whose output is the final class.

Stacking takes place in two phases. In the first phase each of the base level classifiers takes part in the j- fold cross validation training where a vector is returned in the form $<(y'_0 \ldots y'_m), y_j >$ where y'_m is the predicted output of the m^{th} classifier and y_j is the expected output for the same . In the second phase this input is given for the Meta learning algorithm which adjusts the errors in such a way that the classification of the combined model is optimized. This process is repeated for k-fold cross validation to get the final stacked generalization model.

It has been found that stacking method is particularly better suited for combining multiple different types of models. Stacked generalization provides a way for this

situation which is more sophisticated than winner-takes-all approach. Instead of selecting one specific generalize out of multiple ones; the stacking method combines them by using their output information as inputs into a new space. Stacking then generalizes the guesses in that new space. The winner-takes-all combination approach is a special case of stacked generalization. The simple voting approaches have their obvious limitations due to their abilities in capturing only linear relationships. In stacking, an ensemble of classifiers is first trained using bootstrapped samples of the training data, producing level-0 classifiers. The outputs of the base level classifiers are then used to train a Meta classifier. The goal of this next level is to ensure that the training data has accurately completed the learning process. For example, if a classifier consistently misclassified instances from one region as a result of incorrectly learning the feature space of that region, the Meta classifier may be able to discover this problem. Using the learned behaviors of other classifiers, it can improve such training deficiencies.

It is always an active research area that whether combining data mining models gives better performance than selecting that model with best accuracy among base level classifiers. In this research also, in pursuit for finding the best model suitable for this problem, this possibility has been explored. In combining of the models usually the models in level 0(base level classifiers) are operated in parallel and combined with another level classifier called as Meta level classifier. In this work, keeping the three base level classifiers namely decision trees, neural networks and Naïve Bayes classifier, various Meta level classifiers has been tested and it has been observed that voted perceptron Meta level classifier performed best among others. Although numerous data mining algorithms have been developed, a major concern in constructing ensembles is how to select appropriate data mining algorithms as ensemble components.

A challenge is that there is not a single algorithm that can outperform any other algorithms in all data mining tasks, i.e. there is no global optimum solution in selecting data mining algorithms although much effort has been devoted to this area. An ROC (Receiver Operating Characteristics) analysis based approach was approved to evaluate the performance of different classifiers. For every classifier, we can calculate its TP (True Positive) and FP (False Positive) and map it to a two dimensional space with FP on the x-axis and TP on the y-axis. The most efficient classifiers should lie on the convex hull of this ROC plot since they represent the most efficient TP and FP trade off. But ROC analysis requires a two-class classification problem. The three base level classifiers decision tree (ROC=0.79, ACC=79.39), neural networks (ROC=0.80, ACC= 75.54) and naive Bayes (ROC=0.84, ACC=78.29) are the best choices among base level classifiers for this domain. Hence the data is to be modelled as a two class problem by combining classes “excellent”, “average” to output class “E” and the latter two as output class “P”. The 3 models are now completely designed and developed in WEKA. It has been clear that by combing classifiers with stacking, an accuracy of (82.218) has been observed which is clearly better than selecting best among base level classifier accuracy which is 80 in this work.

The voted perceptron algorithm- A variation of perceptron

In the voted-perceptron algorithm, more information is stored during training and then this elaborate information has been used to generate better predictions on the test data. The algorithm is detailed below. The information maintained during training is the list of all prediction vectors that were generated after each and every mistake. For each such vector, we count the number of iterations it survives until the next mistake is made; we refer to this count as the weight of the prediction vector. To calculate a prediction we compute the binary prediction of each one of the prediction vectors and combine all these predictions by a weighted majority vote. The weights used are the survival times described above. This makes intuitive sense as good prediction vectors tend to survive for a long time and thus have larger weight in the majority vote. One application of this algorithm is explained in [12].

Input: A labeled training set $<(x_1,y_1)\ldots.(x_m, y_m)>$ where $x_1 \ldots x_m$ are feature vector instances and $y_1 \ldots y_m$ are class labels to which the training instances have to be classified, T is the no of epochs.

Output A list of weighted perceptrons $<(w_1,c_1)\ldots(w_k,c_k)>$ where $w_1 \ldots w_k$ are the prediction vectors and $c_1 \ldots c_k$ are the weights.

```
K=0
w1 = 0
c1 = 0
Repeat T times
        For i = 1 to m
                If (xi, yi) is misclassified:
                        wk+1 = wk + yi xi
                        ck+1 = 1
                        k = k + 1
                Else
                        ck = ck + 1
```

At the end, a collection of linear separators w_0, w_1, w_2, …, along with survival times: c_n = amount of time that w_n survived.

This c_n is a good measure of the reliability of w_n.

To classify a test point x, use a weighted majority vote

y'=sgn(S) where S is shown as below

$$\operatorname{sgn}\left\{\sum_{n=0}^{N} c_n \operatorname{sgn}(w_n \cdot x)\right\}$$

Conclusion and future directions

This paper puts forward a hybrid stacking ensemble based framework for employment prediction problems that uses voted perceptron algorithm at the Meta level. It

can be verified that a hybrid stacking ensemble approach outperforms the single base level classifier. The various future directions can be improving the stacking framework for multi-class problems involving higher number of output classes and the applicability of the hybrid stacking ensembles in other functional domains such as distributed systems.

References

[1] Antony Browne, Brian D. Hudsonb, David C. Whitley, Martyn G. Ford, Philip Picton. (2004) *Elsevier*, 57(1), 275-293.

[2] Wang L., Sui T. Z. (2007) *Proceedings of wireless communications, networking and mobile computing*, 5544 – 5547.

[3] Zarita Zainuddin, Chan Siow Cheng, Lye Weng Kit. (2008) *Malaysian Journal of Mathematical Sciences* 185-194.

[4] Yoav Freund, Robert E. Schapire (1999) *Machine Learning* 37(3):277-296.

[5] Sreerama K. Murthy. (1998) *Data Mining and Knowledge Discovery*, 345-389.

[6] Elizabeth Murray. (2005) *Proceedings of the 22nd International Conference on Machine Learning, Bonn, Germany.*

[7] Chan P.K., Stolfo S.J. (1995) *Proceedings of the International Conference on Machine Learning*, 90--98.

[8] Sholomo Hershkop, Salvatore J.Stolfo. (2005) *Proceedings of international conference on knowledge discovery in data bases,* 98-107.

[9] Kohavi R. and Provost F. (1998) *Machine Learning* 30(1), 271-274.

[10] Sudheep Elayidom M., Sumam Mary Idicula, Joseph Alexander (2009) *Proceedings of international conference on advances in computing, control and telecommunication technologies, in India*, 669-671.

[11] Sudheep Elayidom, Sumam Mary Idicula, Joseph Alexander. (2010) *Global Journal of Computer Science and Technology, GJCST* 10(10).

[12] Freund Shapire (1998) *Proceedings of the eleventh annual conference on Computational learning theory, USA*, 148-156.

[13] Eunju Kim, Wooju Kim, Yillbyung Leev (2002) *Elsevier Decision Support Systems* 34, 167– 175.

[14] Todorovski, Sašo Džeroski. (2000) *Principles of Data Mining and Knowledge Discovery, Lecture Notes in Computer Science, Springer link, Volume* 1910/2000, 69-84.

[15] Cezary Z. Janikow (1998) *IEEE Transactions on Systems, Man, and Cybernetics, Part* B 28(1): 1-14.

[16] Dietterich T.G. (2000) *In Proceedings of Multiple Classifier Systems*, 1–15.

[17] Magdalena Graczyk, Tadeusz Lasota, Bogdan Trawiński and Krzysztof Trawiński (2010) *Lecture Notes in Computer Science, Volume 5991, Intelligent Information and Database Systems,* 340-350.

[18] NikunjC.Oza, Kagan Tumer (2008) *Science Direct Information Fusion* 9(1), 4-20.

[19] Zenko B., Dzeroski S. (2004) *ACM Journal of machine learning*, 54(3), 255-273.