IJCSMA

Sweety R. Patel *et al*, International Journal of Computer Science and Mobile Applications,
Vol.1 Issue. 4, October- 2013, pg. 7-16 ISSN: 2321-8363

A Review on Ensemble of Classifier Using Artificial Neural Networks as Base Classifier

Sweety R. Patel[1], Mittal C. Patel[2], A. N. Nawathe[3]
[1]PG Student, [2]Assistant Prof (IT), [3]Associate Prof Computer engineering

Keywords

Artificial Intelligence; Classification; Data Mining; Ensemble Method; Neural Network; Noisy data

ABSTRACT

Classification is application areas of neural networks. To generate an algorithm to classify multiclass and single class datasets to achieve high diversity and more accuracy. Ensemble Data Mining Methods provides the power of multiple classifiers to achieve better prediction accuracy than any of the individual classifier could on their own. An ensemble approach involves employment of multiple classifiers and combination of their predictions. Artificial Neural networks are very flexible with respect to incomplete, missing and noisy data and also makes the data to use for dynamic environment. Diversity in an ensemble of neural networks can be handled by manipulating either input data or output data. The paper will help the better understanding of different directions in which research of ensembles has been done in field of noisy data collection

I. INTRODUCTION

In supervised learning, different algorithms are employed to find out the association between independent variables (attributes) and target dependent variable (class). The supervised learning algorithms can be used in two different modes: classification and regression. In classification, the algorithms map the input space to set of predefined class labels whereas in regression, it maps input space to domain of real values. In this text, we limit our study to classification problems. For example, Boosting is an ensemble method that learns a series of “weak” classifiers ach one focusing on correcting the errors made by the previous one; and it is currently one of the best generic inductive classification methods. Ensembles and/or hybrid classifiers remained the problems. Ensembles and/or hybrid classifiers remained the focus of research community since last decade. The concept of ensembles is to employ multiple classifiers and their individual predictions are combined in some way to obtain reliable and more accurate predictions Ensembles have been successfully applied to improve the performance of classifier in many fields e.g. finance [3], bioinformatics [4], medicine [5], information security [6, 7], Information Retrieval [8] etc. . Many researchers report that ensembles often outperform the individual best base classifier [4, 9-17]. Many researchers proposed different concepts to describe improved performance, reduced generalization error and successful applications of ensembles to different fields over individual classifier. For example Allwein et al. (2000) [18] interpreted the improved performance in the framework of large margin classifiers [18, 19]

II. Basic concept of Data Mining

Data mining process (the analysis step of the knowledge discovery in databases process, or KDD), a field of computer science is the process of discovering new patterns from large data sets involving methods such as artificial intelligence, machine learning, statistics and database systems. The goal of data mining is to extract knowledge from a data set in a human-understandable structure and involves database and data management, data preprocessing, model and inference considerations, interestingness metrics, complexity considerations, post-processing of found structure, visualization and online updating. The following steps are used to preprocess the large dataset.

* Selection: Obtain data from various sources.

*Preprocessing: Cleanse data.

* Transformation: Convert to common format. Transform to new format.

* Data Mining: Obtain desired results.

* Interpretation/Evaluation: Present results to user in meaningful manner.

* Knowledge presentation: where visualization and knowledge representation techniques are used to present the mined knowledge to the user. Databases are rich with hidden information that can be used for making intelligent business decisions. Classification and prediction are two forms of data analysis which can be used to extract models describing important data classes or to predict future data trends. Whereas classification predicts categorical labels (or discrete values), prediction models continuous-valued functions.

Data classification is a two step process In the first step, a model is built describing a predetermined set of data classes or concepts. The model is constructed by analyzing database tuples described by attributes. Each tuple is assumed to belong to a predefined class, as determined by one of the attributes, called the class label attribute. In the context of classification, data tuples are also referred to as samples, examples, or objects. The data tuples analyzed to build the model collectively form the training data set.

The individual tuples making up the training set are referred to as training samples and are randomly selected from the sample population. Since the class label of each training sample is provided, this step is also known as supervised learning (i.e., the learning of the model is 'supervised' in that it is told to which class each training sample belongs). It contrasts with unsupervised learning (or clustering), in which the class labels of the training samples are not known, and the number or set of classes to be learned may not be known in advance.

2. 1 Ensemble classifiers

The ensemble classifier involves the employment of multiple classifiers and combines their predictions to obtain reliable and more accurate predictions. To better exploit information, an ensemble of individuals is a more promising choice because information that is derived from combining a set of classifiers might produce higher accuracy than merely using the information from the best classifier among them

Dietterich (2000b) [1] listed three specific reasons for benefits of ensembles: statistical, computational and representational. Other reasons for combining different classifiers include [39]:

1) A designer may have access to a number of different classifiers, each developed in a different context and for an entirely different representation/description of the same problem. An example is the identification of persons by their voice, face, as well as handwriting;

2) Some times more than a single training set is available, each collected at a different time or in a different environment. These training sets may even use different features;

3) Different classifiers trained on the same data may not only differ in their global performances, but they also may show strong local differences. Each classifier may have its own region in the feature space where it performs the best;

4) Some classifiers such as neural networks show different results with different initializations due to the randomness inherent in the training procedure. Instead of selecting the best network and discarding the others, one can combine various networks.

Langin and Rahimi (2010) [23] proposed three different strategies to combine base classifiers namely

1) consecutive combination: A consecutive combination uses methods in order, first one, and then the next;

2) Ensemble combinations: An ensemble combination has methods which are run in parallel with an additional method at the end which provides a single output from multiple potential outputs;

3) Hybrid combinations: A hybrid combination is an offspring of two different parents which implies an interaction of some sort as opposed to being consecutive or parallel. A hybrid strategy can loop-back and forth multiple times between methods or can embed one method within another method.

Ensemble can be generated by using various methods that may include:

1) Modification of structure and characteristics of input data;

2) Aggregation of classes;

3) Selection of base classifiers specialized for specific input regions;

4) Selection of proper set of base classifiers evaluating their performance;

5) Selection of randomly base classifiers;

6) Exploiting problem characteristics e.g., hyperlink ensembles etc. Keeping in view, the popularity and successful applications of ensembles in different fields, various methods are proposed in literature for creating ensembles. Much taxonomy is proposed to categorize different ensembles into different categories. Since research of ensemble is continuously evolving, there is no existing taxonomy that covers every aspect of ensembles.

Kuncheva (2004) proposed a simple taxonomy that is widely used for combining different classifiers [2]. She proposed a classification of ensembles in four levels. The levels are:

A. Combination level – it refers to different ways of combining the classifier predictions.

B. Classifier level – it determines which base classifiers are used to constitute the ensemble.

C. Feature level – it refers to different feature subsets that can be used for the classifiers.

D. Data level – it indicates which dataset is used to train each base classifier.

Kuncheva (2004) [2] also proposed that there are two types of methods to develop ensembles.

1 Decision optimization: it refers to methods to choose and optimize the combiner for a fixed ensemble of base classifiers. This method corresponds to level A (combination level as described above).

2 Coverage optimization: it refers to methods for creating diverse base classifiers assuming a fixed combiner. This method corresponds to level B, C, and D.

2.2 Ensemble classifiers accuracy

Estimating classifier accuracy is important in that it allows one to evaluate how accurately a given classifier will correctly label future data, i.e., data on which the classifier has not been trained. For example, if data from previous sales are used to train a classifier to predict customer purchasing behavior, we would like some estimate of how accurately the classifier can predict the purchasing behavior of future customers. Accuracy estimates also help in the comparison of different classifiers Using training data to derive a classifier and then to estimate the accuracy of the classifier can result in misleading over-optimistic estimates due to overspecialization of the learning algorithm (or model) to the data Holdout and cross-validation are two common techniques for assessing classifier accuracy, based on randomly- sampled partitions of the given data

2.2.1 Holdout Method

In the holdout method, the given data are randomly partitioned into two independent sets, a training set and a test set. Typically, two thirds of the data are allocated to the training set, and the remaining one third is allocated to the test set. The training set is used to derive the classifier, whose accuracy is estimated with the test set .The estimate is pessimistic since only a portion of the initial data is used to derive the classifier. Random subsampling is a variation of the holdout method in which the holdout method is repeated k times. The overall accuracy estimate is taken as the average of the accuracies obtained from each iteration.

2.2.2 Cross-validation

In k-fold cross validation, the initial data are randomly partitioned into k mutually exclusive subsets or folds", s1, s2,…sk, each of approximately equal size. Training and testing is performed k times. In iteration i, the subset Si is reserved as the test set, and the remaining subsets are collectively used to train the classifier. That is, the classifier of the first iteration is trained on subsets S2,…, Sk, and tested on S1; the classifier of the section iteration is trained on subsets S1; S3; ::; Sk, and tested on S2; and so on. The accuracy estimate is the overall number of correct classifications from the k iterations, divided by the total number of samples in the initial data. In stratified cross-validation, the folds are stratified so that the class distribution of the samples in each fold is approximately the same as that in the initial data.

2.2.3 Bootstrap

Other methods of estimating classifier accuracy include bootstrapping, which samples the given training instances uniformly with replacement, and leave-one-out, which is k-fold cross validation with k set to s, the number of initial samples. In general, stratified 10-fold cross-validation is recommended for estimating classifier accuracy (even if computation power allows using more folds) due to its relatively low bias and variance.

III. Increasing classifier accuracy

Bagging (or bootstrap aggregation) and boosting are two such techniques. Each combines a series of T learned classifiers, C1; C2;…..CT, with the aim of creating an improved composite classifier, C*.

3.1 Bagging

Given a set S of s samples, bagging works as follows. For iteration t (t = 1,2,…T), a training set St is sampled with replacement from the original set of samples, S. Since sampling with replacement is used, some of the original samples of S may not be included in St, while others may occur more than once. Each bootstrap sample Si contains approx. 63.2% of the original training data. Remaining (36.8%) are used as test set. A classifier Ct is learned for each training set, St. To classify an unknown sample, X, each classifier Ct returns its class prediction, which counts as one vote. The bagged classifier, C*, counts the votes and assigns the class with the most votes to X. Bagging can be applied to the prediction of continuous values by taking the average value of each vote, rather than the majority.

Advantages

Bagging works well if the base classifiers are unstable. It Increased accuracy because it reduces the variance of the individual classifier. Bagging seeks to reduce the error due to variance of the base classifier. Noise-tolerant, but no t so accurate

3.2 Boosting

In boosting, weights are assigned to each training sample. A series of classifiers is learned. After a classifier Ct is learned, the weights are updated to allow the subsequent classifier, Ct+1, to “pay more attention" to the misclassification errors made by Ct. The final boosted classifier, C_, combines the votes of each individual classifier, where the weight of each classifier's vote is a function of its accuracy. The boosting algorithm can be extended for the prediction of continuous values.

Advantage

Boosting tends to achieve more accuracy than bagging Boosting focuses on misclassified tuples so it risks over fitting.

Limitation

Boosting can fail to perform well given insufficient data. This observation is consistent with the Boosting theory. Boosting also does not perform well when there is a large amount of classification noise (i.e. training and test examples with incorrect class labels).Boosting is also very susceptible to noise in the data.

Comparison between Bagging and Boosting

Bagging is noise-tolerant, produce better class probability estimates. It is not so accurate. It is related to random subsampling. While Boosting is very susceptible to noisy data, produces rather bad class probability estimates. It is related to windowing.

REFERENCES

[1] T.G. Dietterich, Ensemble methods in machine learning, In J. Kittler and F. Roli, editors, Multiple Classifier Systems. First International Workshop, MCS 2000, Cagliari, Italy, volume 1857 of Lecture Notes in Computer Science, pages 1-15. Springer-Verlag, 2000b.

[2] L.I. Kuncheva Combining Pattern Classifiers: Methods and Algorithms, Wiley-Interscience, New York, 2004.

[3] W. Leigh, R. Purvis, J. M. Ragusa, Forecasting the NYSE composite index with technical analysis, pattern recognizer, neural networks, and genetic algorithm: a case study in romantic decision support, Decision Support Systems 32(4) (2002) 361-377.

[4] A. C. Tan, D. Gilbert, Y. Deville, Multi-class Protein Fold Classification using a New Ensemble Machine Learning Approach, Genome Informatics, 14:206-217, 2003.

[5] P. Mangiameli, D. West, R. Rampal, Model selection for medical diagnosis decision support systems, Decision Support Systems, 36(3) (2004) 247-259.

[6] E. Menahem, A. Shabtai, L. Rokach, Y. Elovici, Improving malware detection by applying multi-inducer ensemble. Computational Statistics and Data Analysis, 53(4) (2009) 1483-1494.

[7] R. Moskovitch, Y. Elovici, L. Rokach, Detection of unknown computer worms based on behavioral classification of the host, Computational Statistics and Data Analysis, 52(9) (2008) 4544-4566.

[8] Y. Elovici, B. Shapira, P. Kantor, Using the Information Structure Model to Compare Profile-Based Information Filtering Systems. Information Retrieval Journal 6-1 (2002) 75-97.

[9] E. Bauer, R. Kohavi, An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. Machine Learning, 36, (1999)105–139. ISSN 0885-6125.

[10] G. Brown, J. Wyatt, R. Harris, X. Yao, Diversity creation methods: A survey and categorisation. Journal of Information Fusion, 6 (2005) 5–20.

[11] T.G. Dietterich, G. Bakiri, Error - correcting output codes: A general method for improving multiclass inductive learning programs, In Proceedings of AAAI-91, pp. 572-577. AAAI Press / MIT Press, 1991.

[12] A.C. Tan, D. Gilbert, Ensemble machine learning on gene expression data for cancer classification. Applied Bioinformatics, 2 (2003) 75–83.

[13] V. Engen, Machine Learning For Network Based Intrusion Detection, PhD thesis, Bournemouth University, June 2010.

[14] R.E. Banfield, L.O. Hall, O. Lawrence, K.W. Bowyer, W. Kevin, P. Kegelmeyer, A comparison of decision tree ensemble creation techniques, IEEE Transaction on Pattern Analysis and Machine Intelligence, 29(1) (2007) 173- 180.

[15] S.Y. Sohna, H.W. Shinb, Experimental study for the comparison of of classifier combination methods, Pattern Recognition, 40 (2007) 33-40.

[16] G. Kumar, K. Kumar, AI based supervised classifiers an analysis for intrusion detection, In Proceedings of International conference Advances in Computing and Artificial Intelligence, ACM New York, NY, USA, pp. 170-174, 2011. DOI: 10.1145/2007052.2007087.

[17] G. Kumar, K. Kumar, M. Sachdeva, The Use of Artificial Intelligence based Techniques For Intrusion Detection – A Review, Artificial Intelligence Review, 34-4 (2010) 369-387 Springer, Netherlands, DOI: 10.1007/s10462-010-9179-5 ISSN: 0269-2821.

[18] E.L. Allwein, R.E. Schapire, Y. Singer, Reducing multiclass to binary: a unifying approach for margin classifiers. Journal of Machine Learning Research, 1 (2000) 113-141.

[19] R.E. Schapire, Y. Freund, P. Bartlett, W. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods, The Annals of Statistics, 26(5) (1998) 1651-1686.

[20] E.M. Kleinberg, On the Algorithmic Implementation of Stochastic Discrimination, IEEE Transactions on Pattern Analysis and Machine Intelligence, 22-5 (2000) 473-490.

[21] L. Breiman. Bias, variance and arcing classifiers, Technical Report TR 460, Statistics Department, University of California, Berkeley, CA, 1996.

[22] M. Govindarajan, R. M. Chandrasekaran, Intrusion detection using neural based hybrid classification methods, Computer Networks 55 (2011) 1662–1671.

[23] Chet Langin, Shahram Rahimi, Soft computing in intrusion detection: the state of the art, Journal Ambient Intelligence Human Computing 1 (2010) 133–145, DOI 10.1007/s12652-010-0012-4.

[24] G. Wang, H. Jinxing, M. Jian, H. Lihua, A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering, Expert Systems with Applications (2010), doi:10.1016/j.eswa.2010.02.102.

[25] A. Zainal, MA Maarof, SM Shamsuddin, Ensemble classifiers for network intrusion detection system. Journal of Information Assurance and Security 4 (2009) 217–225.

[26] C. Xiang, P.C. Yong, L.S. Meng, Design of multiple-level hybrid classifier for intrusion detection system using Bayesian clustering and decision trees. Pattern Recogn. Lett., 29,(2008) 918–924, ISSN 167-8655.

[27] N.B. Anuar, H. Sallehudin, A. Gani, O. Zakari, Identifying false alarm for network intrusion detection system using hybrid data mining and decision tree, Malaysian Journal of Computer Science, 21 (2008) 101–115.

[28] F. Gharibian, A.A. Ghorbani, Comparative Study of Supervised Machine Learning Techniques for Intrusion Detection. In CNSR '07: Proceedings of the Fifth Annual Conference on Communication Networks and Services Research, pages 350–358,Washington, DC, USA, 2007. IEEE Computer Society. ISBN 0-7695-2835-X.

[29] A. N. Toosi and Mohsen Kahani, A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers, Computer Communications 30 (2007) 2201–2212.

[30] L. Khan, M. Awad, B. Thuraisingham, A new intrusion detection system using support vector machines and hierarchical clustering. VLDB J 16 (2007). [31] Y. Chen, A. Abraham, B. Yang, Hybrid flexible neural-tree-based intrusion detection systems, International Journal of Intelligent Systems, 22-4 (2007) 337–352, DOI: 10.1002/int.20203.

[32] S. Peddabachigari, A. Abraham, C. Grosan, J. Thomas, Modeling Intrusion Detection Systems Using Hybrid Intelligent Systems, Journal of Network and Computer Applications, 30 (2007) 114–132.

[33] T. S. Hwang, T-J Lee, Y-J Lee, A three-tier IDS via data mining approach, Workshop on mining network data (MineNet), 2007.

[34] A. Abraham, J. Thomas, Distributed intrusion detection systems: a computational intelligence approach, in: H. Abbass, D. Essam (Eds.), Applications of Information Systems to Homeland Security and Defense, Idea Group Inc., USA, 2005, pp. 105–135, chapter 5.

[35] S. Chebrolu, A. Abraham, J. P. Thomas, Feature Deduction and Ensemble Design of Intrusion Detection Systems. Computers and Security, 24 (2005) 295–307.

[36] Z.S. Pan, S.C. Chen, G.B Hu, D.Q. Zhang, Hybrid Neural Network and C4.5 for Misuse Detection, In Machine Learning and Cybernetics, pp. 2463–2467. Xi'an, 2003.

[37] M. Sabhnani, G. Serpen, Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset within Misuse Detection Context. In Proceedings of the International Conference on Machine Learning, Models, Technologies and Applications (MLMTA 2003), volume 1, pp 209–215, 2003.

[38] G. Giacinto, F. Roli, An approach to the automatic design of multiple classifier systems, Pattern Recognition Letters, 22-1(2001) 25-33.

[39] A. K. Jain, R. P. W. Duin, J. Mao, Statistical pattern recognition: A review, IEEE Transactions on Pattern Analysis and Machine Intelligence 22-1 (2000) 4–37.

[40] Yu Yan, Huang Hao, An ensemble approach to intrusion detection based on improved multi-objective genetic algorithm, Journal of Software, 18(6) (2007) 1369−1378.

[41] J. R. Quinlan, C4.5 Programs for machine learning. Morgan Kaufmann San Mateo Ca, 1997.

[42] G. H. John, P. Langley, Estimating continuous distributions in Bayesian classifiers. In: Proceedings of the conference on uncertainty in artificial intelligence, pp 338–345, 1995.

[43] DA Kibler, Instance-based learning algorithms. Mach Learn (1991) 37–66.

[44] G. D. Guvenir, Classification by voting feature intervals. In: Proceedings of the European conference on machine learning, pp 85–92, 1997.

[45] R. Holte, Very simple classification rules perform well on most commonly used datasets. Mach Learn 11 (1993) 63–91.

[46] E. Menahem, L. Rokach, Y. Elovici, Troika – An Improved Stacking Schema for Classification Tasks, Information Sciences, 179-24 (2009a) 4097-4122.

[47] J. Demsar, Statistical Comparisons of Classifiers ver Multiple Data Sets Journal of Machine Learning Research, 7 (2006) 1-30.

[48] Z Muda, W Yassin, M N Sulaiman, N I Udzir, A K means and Naïve Bayes leaning approach for better intrusion detection, information technology journal 10(3) (2011) 648-255, ISSN 1812-5638, DOI: 10.3923/itj.2011.648.655.

[49] G. Brown, J. Wyatt, R. Harris, X. Yao, Diversity creation methods: A survey and categorisation. Journal of Information Fusion, 6 (2005) 5–20.

[50] E.K. Tang, P.N. Suganthan, X. Yao, An Analysis of iversity Measures, Machine Learning, 65 (2006) 247–271.