Applying WEKA towards Machine Learning With Genetic Algorithm and Back-propagation Neural Networks

Zeeshan Ahmed1,2* and Saman Zeeshan2

1Department of Neurobiology and Genetics, Biocenter, University of Wuerzburg, Germany
2Department of Bioinformatics, Biocenter, University of Wuerzburg, Germany

Abstract

Machine learning aims of facilitating complex system data analysis, optimization, classification and prediction with the use of different mathematical and statistical algorithms. In this research, we are interested in establishing the process of estimating best optimal input parameters to train networks. Using WEKA, this paper implements a classifier with Back-propagation Neural Networks and Genetic Algorithm towards efficient data classification and optimization. The implemented classifier is capable of reading and analyzing a number of populations in giving datasets, and based on the identified population it estimates kinds of species in a population, hidden layers, momentum, accuracy, correct and incorrect instances.

Keywords: Back Propagation Neural Network; Genetic Algorithm; Machine learning; WEKA

Introduction

Machine learning [1] is a branch of Artificial Intelligence, facilitating probabilistic system development for complex data analysis, optimization, classification and prediction. Different learning methods have been introduced e.g. supervised learning, unsupervised learning, semi supervised learning, reinforcement learning, transduction learning and learning to learn etc.

Several statistical algorithms (e.g. Genetic Algorithm [2], Bayesian statistics [3], Case-based reasoning [4], Decision trees [5], Inductive logic programming [6], Gaussian process regression [7], Group method of data handling [8], k-NN [9], SVMs [10], Ripper [11], C4.5 [12] and Rule-based classifier [13] etc.) have been proposed for the learning behavior implementation. The criterion for choosing a mathematical algorithm is based on the ability to deal with the weighting of networks, chromosome encoding and terminals.

Different machine learning approaches have been proposed towards the implementation of adaptive machine learning systems and data classification e.g. Fast Perceptron Decision Tree Learning [14], Massive Online Analysis (MOA) [15], 3D Face Recognition Using Multi view Key point Matching [16], Evolving Data Streams [17], Classifier Chain [18], Multi-label Classification [19], Multiple-Instance Learning [20], Adaptive Regression [21], nearest neighbor search [22], Bayesian network classification [23,24], Naive Bayes text classification [25], ML for Information Retrieval [26], Probabilistic unification grammars [27], Instance Weighting [28], KEA [29] and Meta Data for ML [30] etc. Apart from the fact of existence of these referred valuable approaches, we have decided to implement our own software application during this research and development, consisting of different methodology.

In this research, we are interested in finding the most suitable algorithm to establish the process of estimating best optimal input parameters and on the basis the selected parameters, train network to best fit with the use of suitable learning techniques. We discuss a script implementing the Genetic Algorithm for data optimization and back propagation neural network algorithm for the learning behavior. The objective is to analysis different datasets based on the number of attributes, classes, instances and relationships.

Following the agenda (Section 1), this short paper is organized in the upcoming sections: data classifier and its methodology explain in section 2, validation is performed in section 3 and observed results are concluded in section 4.

Optimal Data Classifier

The implemented classifier is proficient in reading and analyzing a number of populations in giving datasets. The classifier is developed using Java programming language and WEKA library [31,32]. The WEKA library provides built-in classes and functions to implement and utilize different mathematical and statistical algorithms (e.g. genetic algorithm and back propagation algorithm etc.) for data classification.

Based on the number of identified population, it estimates following results: kinds of species in a population (if there are more than 1), correctly classified instances, incorrectly classified instances, hidden layers, momentum and accuracy (optimized, weighted results).

The classifier is capable of processing standard Attribute Relation File Format (ARFF) dataset files, which describes the list of instances sharing a set of attributes, especially used to develop for machine learning projects. The classifier’s workflow starts with the analysis of inputted data and extraction of attributes, classes, instances and relationships. In the next step classifier extracts the information about number of hidden layers, learning rate and momentum to identify correctly and incorrectly classified instances. At the final step, classify the data using Back Propagated Neural Network for Multilayer Perceptron and optimize results using Genetic Algorithm.

During data classification using the genetic algorithm; first chromosomes are estimated, then learning rate and momentum is set.

*Corresponding author: Zeeshan Ahmed, Department of Neurobiology and Genetics, University of Wuerzburg, Germany, Tel: +49-931-31-81917; Fax: +49-931-31-84452; E-mail: zeeshan.ahmed@uni-wuerzburg.de

Received May 20, 2014; Accepted August 26, 2014; Published September 02, 2014

Citation: Ahmed Z, Zeeshan S (2014) Applying WEKA towards Machine Learning With Genetic Algorithm and Back-propagation Neural Networks. J Data Mining Genomics Proteomics 5: 157. doi:10.4172/2153-0602.1000157

Copyright: © 2014 Ahmed Z, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

J Data Mining Genomics Proteomics
ISSN: 2153-0602 JDMGP, an open access journal

Volume 5 • Issue 2 • 1000157
to perform cross over using a pair of the best results (Figure 1). The next mutation of two offspring is performed on the basis of obtained accuracies of two previously estimated offspring. The offspring with lower values are replaced with two new offspring. In the last steps, after cross validation, the individual and commutative weights of instances are calculated. The obtained results are validated and final output is presented to the user in the end. The measurement and prediction procedure can be repeated until the satisfactory results are achieved.

Validation

We have validated the classifier using two different data sets: Zoo database (http://www.hakank.org/weka/zoo.arff) and Labor database...
Table 1: Results of Data classification.

| Zoo Database | Labor Database |
|--------------|----------------|
| 1617 Mammals, 539 Birds, 0 Reptile, 637 Fish, 0 Amphibian, 490 Insects and 49 Invertebrate from the whole population of 3332 species in dataset. | 49 Good and 441 Bad of all 490 Population. |
| 68 instances are correctly classified and rest 33 are incorrectly classified from all 101 instances | 10 instances are correctly classified and rest 47 are incorrectly classified from all 57 instances |
| No Hidden layer | No Hidden layer |
| 0.3 Learning rate | 0.1 Learning rate |
| 0.1 Momentum | 0.5 Momentum |
| 0.6732673267326733 Accuracy | 0.17543859649122806 Accuracy |

Both datasets are analyzed using implemented classifier, using WEKA explorer (Figure 2A and 2C). The observed results are (Figure 2B and 2D) are presented in Table 1. We have validated the classifier in three ways: (1) by increasing the learning rate and placing the momentum constant, (2) by increasing both learning rate and momentum and (3) by randomly changing the weight. During the validation process the size of the chromosome was 6 bits, 3 bit decimal value (0-10/10=value) by randomly changing the weight. During the validation process the size of the chromosome was 6 bits, 3 bit decimal value (0-10/10=value) by randomly changing the weight. During the validation process the size of the chromosome was 6 bits, 3 bit decimal value (0-10/10=value) by randomly changing the weight. During the validation process the size of the chromosome was 6 bits, 3 bit decimal value (0-10/10=value) by randomly changing the weight. During the validation process the size of the chromosome was 6 bits, 3 bit decimal value (0-10/10=value) by randomly changing the weight. During the validation process the size of the chromosome was 6 bits, 3 bit decimal value (0-10/10=value) by randomly changing the weight.

Conclusions

We have observed during the validation process that by keeping the default weight of instance, the results become stable but by increasing the weight of instance the size of results increases. The findings lead to the outcome that mutation can affect the accuracy by increasing and decreasing it. Moreover, we have also observed that classifier produces results in minimum possible time with value 1, and if we will increase the value of classifier it will take more time.

Acknowledgement

We would like to thank Deutsche Forschungsgemeinschaft (DFG SFB 1047/Z) for funding and University of Wuerzburg Germany for the support. We thank anonymous reviewers for helpful comments on the manuscript and the Land of Bavaria Germany.

References

1. Smola A, Vishwanathan SVN (2008) Introduction to Machine Learning, Cambridge University Press, USA.
2. Man KF, Tang KS, Kwong S (1996) Genetic algorithms: concepts and applications. IEEE Transactions of Industrial Electronics.
3. Gelman A, Shalizi CR (2013) Philosophy and the practice of Bayesian statistics. British Journal of Mathematical and Statistical Psychology 66: 8-38.
4. Kohodner JL (1992) An Introduction to Case-Based Reasoning. Artificial Intelligence Review 6: 3-34.
5. Quinlan JR (1986) Induction of Decision Trees. Machine Learning 1: 81-106.
6. Raedt LD, Frasconi P, Kersting K, Muggleton S (2008) Probabilistic Inductive Logic Programming - Theory and Applications. Springer Lecture Notes in Computer Science.
7. Wilson AG, Knowles DA, Ghahramani Z (2012) Gaussian Process Regression Networks - In the Proceedings of the 29th International Conference on Machine Learning, Scotland, UK.
8. Mehra RK (1977) “Group method of data handling (GMDH): Review and experience”. In the 16th Symposium on Adaptive Processes and A Special Symposium on Fuzzy Set Theory and Applications.
9. Hajebi K, Abbasi-Yadkori Y, Shahbazi H, Zhang H (2011) Fast Approximate Nearest-Neighbor Search with k-Nearest Neighbor Graph”, In the Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence 2: 1312-1317.
10. El-Naqi A, Yang Y, Wernick MN, Galatsanos NP, Nishikawa RM (2002) "A Support Vector Machine Approach for Detection of Microcalcifications”. IEEE Transactions on Medical Imaging 21: 1552-1563.
11. Qin B, Xia Y, Prabhakar S, Tu Y (2009) A Rule-Based Classification Algorithm for Uncertain Data In the Proceedings of IEEE International Conference on Data Engineering.
12. Cao R, Xu L (2009) Improved C4.5 Algorithm for the Analysis of Sales. In the Proceedings of Web Information Systems and Applications Conference 173-176.
13. Bitlet A, Holmes G, Pfahringer B, Eibe F (2010) Fast perception decision tree learning from evolving data streams. In the Proceedings of 14th Pacific-Asia Conference on Knowledge Discovery and Data Mining 299-310.
14. Bitlet A, Holmes G, Pfahringer B, Eibe F (2010) MOA: Massive online analysis. Journal of Machine Learning Research 11: 1601-1604.
15. Mayo M, Zhang E (2009) 3d face recognition using multiview keypoint matching. In the Proceedings of 6th International Conference on Advanced Video and Signal Based Surveillance 290-295.
16. Bitlet A, Holmes G, Pfahringer B, Eibe F, Galavarda R (2009) New ensemble methods for evolving data streams In the Proceedings of 15th International Conference on Knowledge discovery and data mining.
17. Read J, Pfahringer B, Holmes G, Frank E (2009) Classifier chains for multi-label classification. Machine Learning and Knowledge Discovery in Databases Lecture Notes in Computer Science 5782: 254-269.
18. Read J, Pfahringer B, Geoffrey H (2008) Multi-label classification using ensembles of pruned sets. In the Proceedings of 8th IEEE International Conference on Data Mining 995-100.
19. Foulis J, Frank E (2008) Revisiting multiple-instance learning via embedded instance selection. AI 2008: Advances in Artificial Intelligence Lecture Notes in Computer Science 5360: 300-310.
20. Frank E, Hall M (2008) Additive regression applied to a large-scale collaborative filtering problem. In the Proceedings of 21st Australasian Joint Conference on Artificial Intelligence, New Zealand.
21. Arya S, Mount DM, Netanyahu NS, Silverman S, Wu AY (1998) An optimal algorithm for approximate nearest neighbor searching fixed dimensions. Journal of the ACM 45: 891-923.
22. Bouckaert RR (2006) Voting massive collections of bayesian network classifiers for data streams. AI 2006: Advances in Artificial Intelligence Lecture Notes in Computer Science 4304: 243-252.
23. Frank E, Bouckaert RR (2006) Naive bayes for text classification with unbalanced classes In the Proceedings of 10th European Conference on Principles and Practice of Knowledge Discovery in Databases 503-510.
24. Bouckaert R (2004) Bayesian network classifiers in weka. Department of Computer Science.
25. Cunningham SJ, Littin JN, Witten IH (1997) Applications of machine learning in information retrieval. Department of Computer Science.
26. Smith TC, Cleary JG (1997) Probabilistic unification grammars. In the Proceedings of Australasian Natural Language Processing Summer Workshop.

27. Ting KM (1997) Inducing cost-sensitive trees via instance-weighting. Department of Computer Science.

28. Witten IH, Paynter WG, Frank E, Gutwin C, Neville-Manning CG (1999) KEA: Practical automatic keyphrase extraction. In the Proceedings of 4th ACM conference on Digital Libraries 254-255.

29. Cleary JG, Holmes G, Cunningham SJ, Witten IH (1996) Metadata for database mining. In the Proceedings of IEEE Metadata Conference.

30. Cunningham SJ (1996) Dataset cataloguing metadata for machine learning applications and research. Computer Science Department.

31. Amini J (2008) Optimum Learning Rate in Back-Propagation Neural Network for Classification of Satellite Images (IRIS-ID). Scientia Iranica 15: 558-567.

32. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, et al. (2009) The WEKA data mining software: an update. ACM SIGKDD Explorations Newsletter Explorations 11: 10-18.