# Towards Efficient Data Classification and Optimization using WEKA

Z. Ahmed
University of Wuerzburg
Germany
zeeshan.ahmed@uni-wuerzburg.de

***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 population 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

**Received:** 12 March 2014, Revised 19 April 2014, Accepted 25 April 2014



## 1. 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], *C*4.5 [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], 3*D 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.

**2. Optimal Data Classifier**

The implemented classifier is proficient in reading and analyzing a number of population in giving datasets. 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 Perception and optimize results using Genetic Algorithm. The software, scripting is performed in the Java programming language and with the help of WEKA [31], [32].

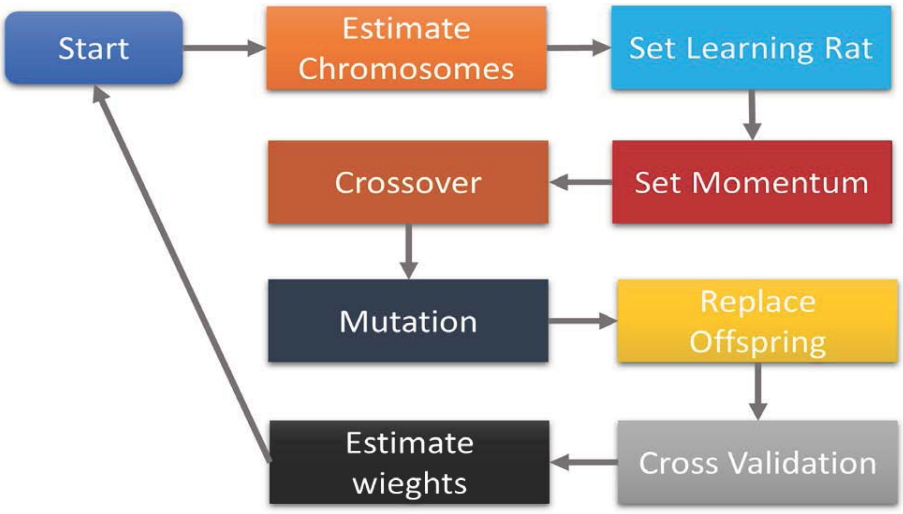
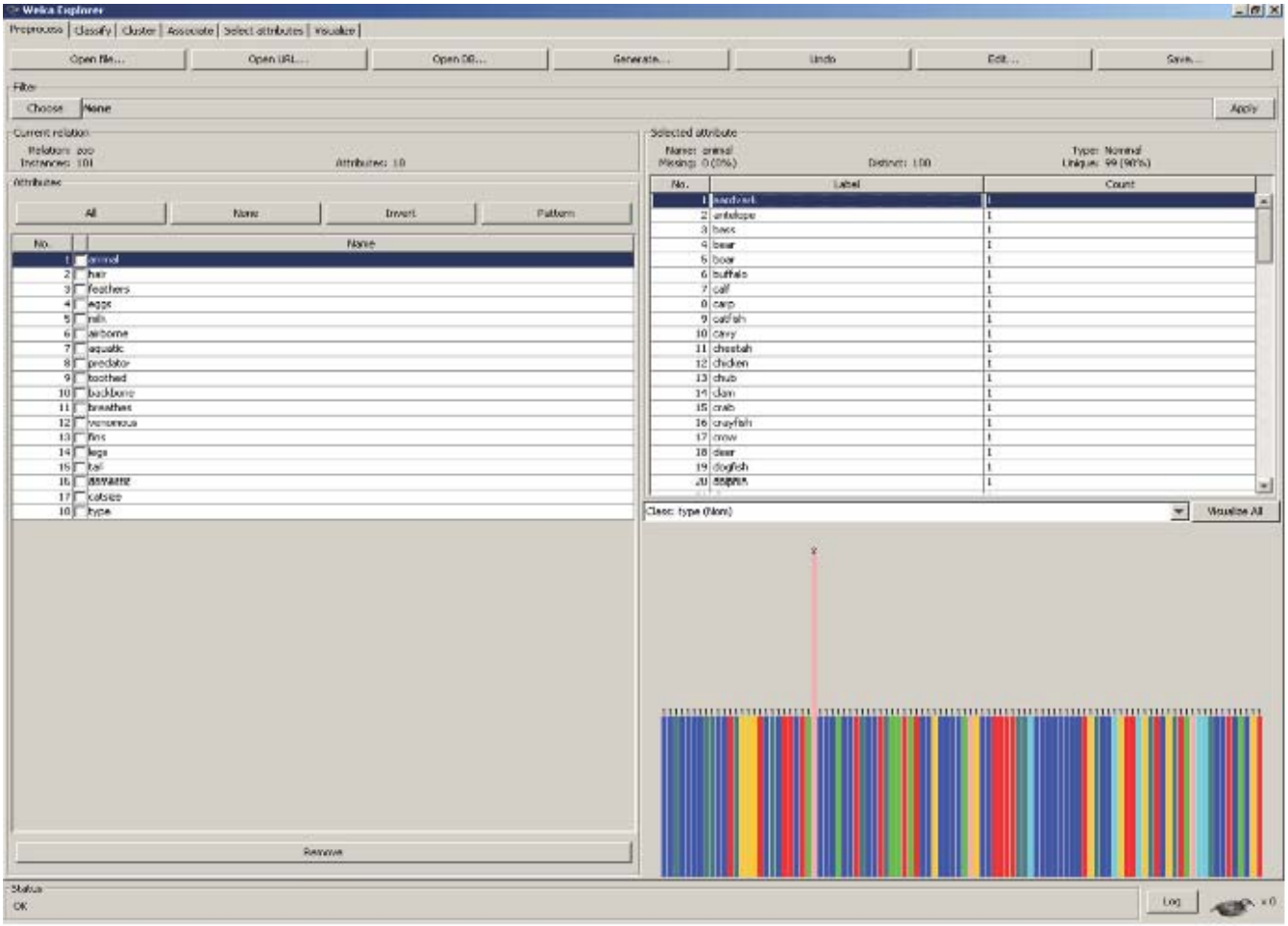



Figure 1. Data Classification

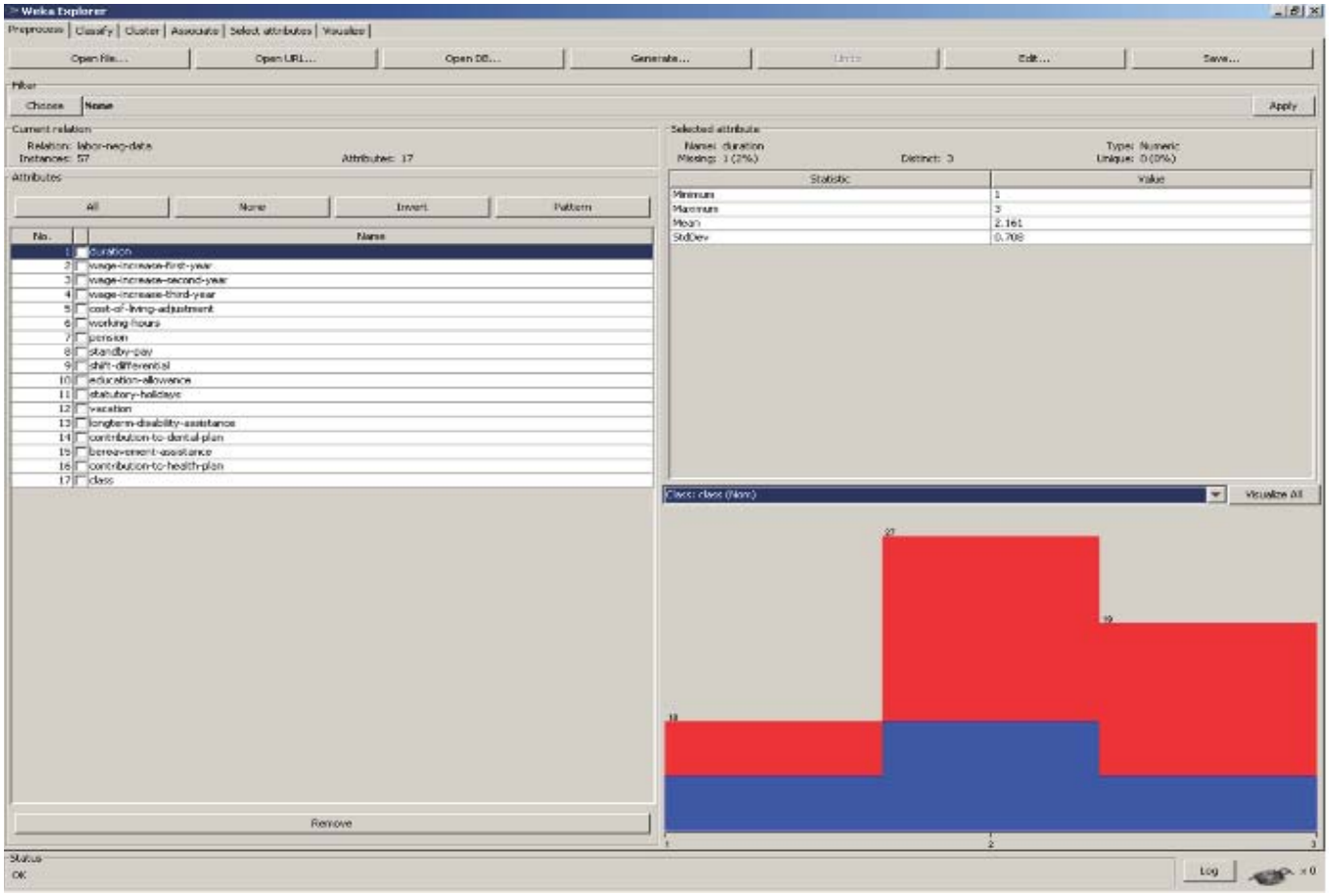
The Figure 1 presents the application of Genetic Algorithm for data classification. The method estimates chromosomes, sets learning rate and momentum based calculated chromosomes, crosses over using pair of best chromosomes, mutates new off springs, replaces offspring, perform cross validation, calculates individual and commutative weights of all instances.



(a)

```
Mamals              1617
Birds               539
Reptiles            0
Fish                637
Amphibian           0
Insects             490
Invertable          49
Classified Instance68
Non Classified Instance33
Updating Second Parent information
Hidden Layers                   :null
Leaning Rate                    :0.3
Momentum                        :0.1
Correctly classified            : 68.0
Incorrectly classified          : 33.0
Accuracy                        : 0.6732673267326733
Making new population
Evaluation of Best Final result
**********Optimal Results**********
  First Best Corrent Instance    : 68.0
  First Best Incorrect Instance  : 33.0
  First Best Learning Rate       : 0.1
  First Best Momentum            : 0.3
  First Best Weight              : 0.0
  First Accuracy                 : 0.6732673267326733
```

(b)




(c)

```
GOOD              49
BAD                    441
Classified Instance              10
Non Classified Instance 47
Updating Second Parent information
Hidden Layers                  :null
Leaning Rate                   :0.1
Momentum                       :0.5
Correctly classified           : 10.0
Incorrectly classified         : 47.0
Accuracy                       : 0.17543859649122806
Making new population
Evaluation of Best Final result
**********Optimal Results**********
  First Best Corrent Instance    : 10.0
  First Best Incorrect Instance  : 47.0
  First Best Learning Rate       : 0.5
  First Best Momentum            : 0.1
  First Best Weight              : 0.0
  First Accuracy                 : 0.17543859649122806
```

(d)

Figure 2. WEKA Graphical User Interface

| Zoo Database | Labor Database |
|---|---|
| 1617 Mammals, 539 Birds, 0 Reptile, 637 Fish, 0 Amphibian, 490 Insects and 49 Invertible 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 |
| 0.67326732673267326733 Accuracy | 0.17543859649122806 Accuracy |

Table 1. Results of Dataclassification

During data classification using the genetic algorithm; first chromosomes are estimated, then learning rate and momentum is set to perform cross over using a pair of the best results (Figure 1). The next the 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.

**3. Validation**

We have validated the classifier using two different data sets: *Zoo database* (http://www.hakank.org/weka/zoo.arff) and *Labor database* (http://www.hakank.org/weka/labor.arff).

Zoo database contains 101 Instances with of 18 Attributes; 2 numeric attributes (animal and legs) and 16 Booleans attributes (hair, feather, eggs, milk, airborne, aquatic, predator, toothed, backbone, breathes, venomous, fins, legs, tail, domestic, cat size and type). Whereas the Labor database comprises of 57 Instances including of 16 Attributes; 13 numeric attributes (duration, wage increase first year, wage increase second year, wage increase third year, cost of living adjustments, working hours, pension, standby pay, shift differential, statutory holidays, vacations, contribution dental plan and contribute to health plan) and 3 Boolean attributes (bereavement assistance, long term disability assistance and education alliance).

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 Table1. 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) for learning rate and 3 bit decimal values for momentum.

The Figure 2 (A) presents the example data set Zoo Database being processed using WEKA Explorer and (2B) presents the obtained results. Whereas the Figure 2 (C) presents the example data set Labor Database being processed using WEKA Explorer and (2C) presents the obtained results.

**4. 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.

**5. Acknowledgements**

Special thanks the FUNDING, UNIVESRITY and the anonymous reviewers for helpful comments on the manuscript.

**References**

[1] Smola, A., Vishwanathan, S. V. N. (2008). Introduction to Machine Learning, Cambridge University Press.

[2] Man, K. F., Tang, K. S., Kwong, S. (1996). Genetic algorithms: concepts and applications, IEEE Transactions of Industrial Electronics, 43 (5).

[3] Gelman, A., Shalizi, C. R. (2013). Philosophy and the practice of Bayesian statistics, *British Journal of Mathematical and Statistical Psychology*, 66, p.8-38.

[4] Kolodner, J. L. (1992). An Introduction to Case-Based Reasoning, *Artificial Intelligence Review*, 6, p.3-34.

[5] QUINLAN, J. R. (1986). Induction of Decision Trees, *Machine Learning*, 1, p.81-106.

[6] Raedt, L. D., Frasconi, P., Kersting, K., Muggleton, S. (2008). Probabilistic Inductive Logic Programming-Theory and Applications. Springer Lecture Notes in Computer Science.

[7] Wilson, A. G., Knowles, D. A., Ghahramani, Z. (2012). Gaussian Process Regression Networks, *In*: Proceedings of the 29th International Conference on Machine Learning, Scotland, UK.

[8] Mehra, R. K. (1977). Group method of data handling (GMDH): Review and experience, *In*: 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*: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence.

[10] El-Naqa, I., Yang, Y., Wernick, M. N., Galatsanos, N. P., Nishikaw, R. M. (2002). A Support Vector Machine Approach for Detection of Microcalcifications, *IEEE Transactions on Medical Imaging*, 21 (12).

[11] Qin, B., Xia, Y., Prabhakar, S., Tu, Y. (2009). A Rule-Based Classification Algorithm for Uncertain Data, *In*: Proceedings of IEEE International Conference on Data Engineering.

[12] Cao, R. (2009). Improved C4.5 Algorithm for the Analysis of Sales, *In*: Proceedings of Web Information Systems and Applications Conference.

[13] Bifet, A., Holmes, G., Pfahringer, B., Eibe Frank. (2010). Fast perceptron decision tree learning from evolving data streams, *In*: Proceedings of 14th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Hyderabad, India.

[14] Bifet, A., Holmes, G., Kirkby, R., Pfahringer, K. (2010). MOA: Massive online analysis, *Journal of Machine Learning Research*, 11, p. 1601-1604.

[15] Mayo, M., Edmond Zhang. (2009). 3D face recognition using multiview keypoint matching. *In*: Proceedings of 6th International Conference on Advanced Video and Signal Based Surveillance, Italy.

[16] Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., Gavaldà, R. (2009). New ensemble methods for evolving data streams, *In*: Proceedings of 15th International Conference on Knowledge discovery and data mining, USA.

[17] Read, J., Pfahringer, B., Holmes, G., Frank, E. (2009). Classifier chains for multi-label classification. *In*: Proceedings of 13th European Conference on Principles and Practice of Knowledge Discovery in Databases and 20th European Conference on Machine Learning, Slovenia.

[18] Read, J., Pfahringer, B., Geoffrey Holmes, R. (2008). Multi-label classification using ensembles of pruned sets. *In*: Proceedings of 8th IEEE International Conference on Data Mining, Italy.

[19] Foulds, J., Frank, E. (2008). Revisiting multiple-instance learning via embedded instance selection, *In*: Proceedings of 21st *Australasian Joint Conference on Artificial Intelligence*, Auckland, New Zealand.

[20] Frank, E., Hall, M. (2008). Additive regression applied to a large-scale collaborative filtering problem, *In*: Proceedings of 21st *Australasian Joint Conference on Artificial Intelligence*, New Zealand.

[21] Arya, S., Mount, D. M., Netanyahu, N. S., Silverman, R., Wu, A. Y. (1998). An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM*, 45 (6) 891-923.

[22] Bouckaert, R. R. (2006). Voting massive collections of bayesian network classifiers for data streams, *In*: Proceedings of 19th Australian Joint Conference on Artificial Intelligence.

[23] Frank, E., Bouckaert, R. R. (2006). Naive bayes for text classification with unbalanced classes, *In*: Proceedings of 10th European Conference on Principles and Practice of Knowledge Discovery in Databases, Germany.

[24] Bouckaert, R. (2004). Bayesian network classifiers in weka. Technical Report 14/2004, The University of Waikato, Department of Computer Science, Hamilton, New Zealand.

[25] Cunningham, S. J., Littin, J. N., Witten, I. H. (1997). Applications of machine learning in information retrieval, Technical Report 97/6, University of Waikato, Department of Computer Science, Hamilton, New Zealand, February.

[26] Smith, T. C., Cleary, J. G. (1997). Probabilistic unification grammars, *In*: Proceedings of Australasian Natural Language Processing Summer Workshop.

[27] Ting, K. M. (1997). Inducing cost-sensitive trees via instance-weighting, Technical Report 97/22, University of Waikato, Department of Computer Science, Hamilton, New Zealand, September.

[28] Witten, I. H., Paynter, W. G., Frank, E., Gutwin, C., Nevill-Manning, C. G. (1999). KEA: Practical automatic keyphrase extraction, *In*: Proceedings of 4th ACM conference on Digital Libraries, Berkeley, CA.

[29] Cleary, J. G., Holmes, G., Cunningham, S. J., Witten, I. H. (1996). Metadata for database mining, *In*: Proceedings of IEEE Metadata Conference, April.

[30] Cunningham, S. J. (1996). Dataset cataloguing metadata for machine learning applications and research. Technical Report 96/26, University of Waikato, Computer Science Department, Hamilton, New Zealand, October.

[31] Amini, J. (2008). Optimum Learning Rate in Back-Propagation Neural Network for Classification of Satellite Images (IRS-1D). *Scientia Iranica*, 15 (6) 558-567.

[32] Hall, M., Frank, E., Holmes, G., Pfahringer P. Reutemann, B., Witten, I. H. (2009). The WEKA data mining software: an update. *ACM SIGKDD Explorations*, 11 (1) 10-18.