# Darwin, Lamarck, or Baldwin: Applying Evolutionary Algorithms to Machine Learning Techniques

Andreas Holzinger, David Blanchard
Marcus Bloice, Katharina Holzinger
Research Unit HCI
Institute for Medical Informatics,
Statistics and Documentation
Graz, Austria
Email: a.holzinger@hci4all.at

Vasile Palade
Department of Computer Science
University of Oxford, UK
and
Department of Computing
Coventry University, UK
Email: vasile.palade@cs.ox.ac.uk

Raul Rabadan
Columbia University
Columbia Medical Center
Department of Biomedical Informatics,
New York,
USA
Email: rabadan@dbmi.columbia.edu

*Abstract*—**Evolutionary Algorithms (EAs), inspired by biological mechanisms observed in nature, such as selection and genetic changes, have much potential to find the best solution for a given optimisation problem. Contrary to Darwin, and according to Lamarck and Baldwin, organisms in natural systems learn to adapt over their lifetime and allow to adjust over generations. Whereas earlier research was rather reserved, more recent research underpinned by the work of Lamarck and Baldwin, finds that these theories have much potential, particularly in upcoming fields such as epigenetics. In this paper, we report on some experiments with different evolutionary algorithms with the purpose to improve the accuracy of data mining methods. We explore whether and to what extent an optimisation goal can be reached through a calculation of certain parameters or attribute weightings by use of such evolutionary strategies. We provide a look at different EAs inspired by the theories of Darwin, Lamarck, and Baldwin, as well as the problem solving methods of certain species. In this paper we demonstrate that the modification of well-established machine learning techniques can be achieved in order to include methods from genetic algorithm theory without extensive programming effort. Our results pave the way for much further research at the cross section of machine learning optimisation techniques and evolutionary algorithm research.**

## I. Introduction and Motivation

The original idea to use algorithms inspired by evolutionary paradigms in the sense of Darwin, Lamarck or Baldwin, reaches back to the early days of computer science [1] and gained momentum with the introduction of adaptive systems by John H. Holland (1962) [2]. Since the 1980s, EAs have been used to address optimisation problems due to their robustness and flexibility, especially in fields where traditional greedy algorithms did not provide satisfactory results. Traditional evolutionary paradigms are usually divided into two groups according to the principle invoked to explain the biological change: While Lamarck proposed the inheritance of acquired characteristics, Darwin underlines the role of selection on random genetic variation, and Baldwin proposed a mechanism for specific selection for general learning abilities. The most commonly used EAs are inspired by the evolution theory of Darwin and, whilst the theories of Lamarck or Baldwin were questioned a lot in the past, recent research has found much potential, particularly in epigenetics. In this paper, we report on exploring the possibilities of optimising data mining methods, being inspired by these theories, which has much potential for further research e.g. in classification optimisation [3], or imbalanced data [4], or in feature selection [5], to mention only a few.

## II. Evolutionary Algorithms inspired by Darwin, Lamarck, and Baldwin

### A. Background

A recent overview on EA's and particularly a discussion on the differences between the theories of Charles Darwin (1809–1882) versus Jean-Baptiste de Lamarck (1744–1829) can be found in [6]. Darwinism was not the only theory of evolution of the time, and less known to computer scientists may be Lamarckism, which states, unlike Darwinism, that selection is *not* the driving force of evolution, but the inheritance of acquired characteristics or inherited "effort" of the organisms themselves. It was assumed that appropriate characteristics arise from the desire of the organisms to achieve them (strive for perfection). So, unlike Darwinism, where evolution is only a result of competition and selection, in Lamarckism the organisms themselves control evolution. This is accomplished through practice and training. Although in biology, Lamarckism would be possible if there was a mechanism that translates phenotypic changes into the sequence of the responsible gene, this theory was mostly rejected. However, Lamarckism might provide some answers, especially for modern genetics and medicine. In epigenetics, which very early dealt with questions of evolution, it was found that there are special traits which can be inherited without being part of the genetic code. The possibility that acquired behaviour or marks can be passed from parents to children is currently in serious debate and the advent of epigenetics is hailed as a profound shift in our understanding of inheritance, i.e. that genes also possess a type of “memory” [7], [8]. Research in epigenetics is on the increase in the medical field [9]; a very recent example in cancer research can be found in [10].

Even lesser known to computer scientists may be a theory suggested by James Mark Baldwin (1861–1934) in 1896 [11], named the *Baldwin effect* by Simpson (1953) [12] and which is based on the following observations [13]: If a species is evolving in a changing environment, there will be evolutionary pressure to favour individuals with the capability to learn during their lifetime. For example, if a new predator appears in the environment, then individuals capable of learning to avoid the predator will be more successful than individuals who cannot learn. In effect, the ability to learn allows an individual to perform a small local search during its lifetime to maximise its fitness. In contrast, non learning individuals whose fitness is fully determined by their genetic makeup will operate at a relative disadvantage. Those individuals who are able to learn many traits will rely less strongly on their genetic code to “hard-wire” traits. As a result, these individuals can support a more diverse gene pool, relying on individual learning to overcome the “missing” or “not quite optimised” traits in the genetic code. This more diverse gene pool can, in turn, support more rapid evolutionary adaptation. Thus, the ability of individuals to learn can have an indirect accelerating effect on the rate of evolutionary adaptation for the entire population.

### B. Basic principles of evolutionary algorithms

The standard approach applied by most genetic algorithms is to generate solution candidates which are a symbolic representation of chromosomes. Like their biological counterparts, these candidates consist of multiple entities which can be seen as the individual genes of the chromosome. While real genes offer a very complex structure, in evolutionary algorithms they are usually represented by primitive numeric types[14].

In the beginning, a population is initialised and different functions based on evolutionary principles are applied to that population in a loop, which is called a generation, until a specified termination criterion is reached.

The procedure for such algorithms usually follows similar methodologies and can be summarised as follows [14]:

- **Initialize population:** The encoded solution candidates are generated randomly or by a given function.
- **Evaluate fitness:** Evaluates similar to a cost function how close each one of the candidates is to the optimal solution.
- **Selection and Crossover:** The chromosomes most removed from the desired solution are removed while children will be produced from the best chromosomes in the population.
- **Mutation:** To avoid the possibility of getting stuck near a local optimum, a random factor is introduced and genes of certain chromosomes will be altered in a random manner.

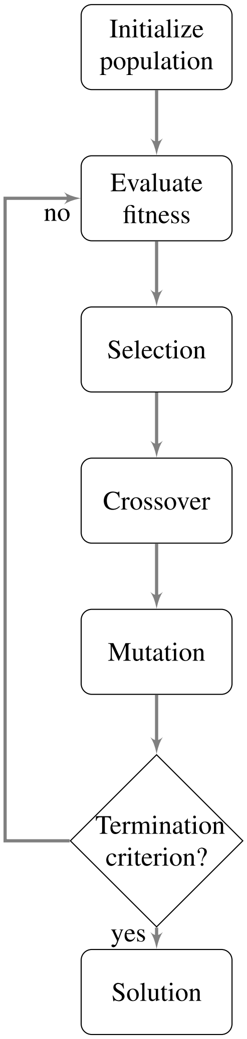

Fig. 1: Flow chart of a Darwinian evolutionary algorithm.

### C. Differences in the implementation of the theories

There are certain changes that need to be made to adapt an implementation of a evolutionary algorithm based on the Theory of Darwin to represent the theories of Lamarck or Baldwin. We continue with the basic structure of these algorithms and their differences. The implementations we will explore in the following sections are only figurative ways to apply these ideas to the field of computer science.

A description of Darwinian Theory can be seen in Figure 1, while a description of the Lamarckian/Baldwinian Theory can be seen in Figure 2 [15].

This strategy uses an adaptation tactic to modify certain chromosomes in order to transform them to become a fitter solution candidate in their environment. Either a certain percentage of chromosomes or the entire population can be target of this modification process.
These tactics have proven to offer great advantages trough repairing unfit chromosomes[16]. These adaptations can be carried out additionally or instead of the mutation processes, seeing that adapting to an environment is a form

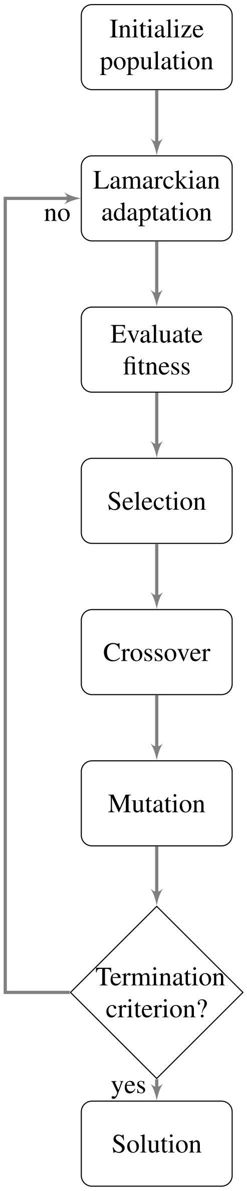

Fig. 2: Flow chart of an exemplary Lamarck evolutionary algorithm. The Lamarckian adaptation function can also be performed instead of the mutation.

of mutation[17].
The adaptation of individual chromosomes can be performed trough a local search optimisation, for example with an Hill Climbing algorithm[17].

The theory of Baldwin proposes a more passive implementation compared to the Lamarckian method and stays closer to the Darwinian theory. A local search optimisation is performed on certain individuals before their fitness can be evaluated, but the changes are not encoded back into their chromosomes [18].
Through this mechanic, the adaptation to the environment still produces an advantage for certain chromosomes but these changes will not directly get passed on to the next generations. An advantage will only be passed on indirectly, making this method a passive Lamarckian algorithm.

## III. Exemplary optimisation of a Naive Bayes Classifier

Naive Bayes is a very effective classifier used in different forms of data mining. A weighted Naive Bayes is an extension of this classifier presenting the possibility to improve its accuracy by applying different weightings to attributes[19].

### A. *Optimisation experiments requirements*

The basic classification in our experiments was executed with a simple multinomial Naive Bayes using evolutionary algorithms to improve the result calculated through a normal density function. The experiments were performed with the help of the Weka[20] engine, a collection of various machine learning methods and data sets. The experiments were performed on an open data set of patients diagnosed with diabetes.

The dataset used to evaluate the algorithms was obtained from the University of California, Irvine's Machine Learning Repository [21]. The repository contains over 280 datasets that are made available as a service for the machine learning community for the purposes of experimentation, evaluation, and benchmarking. The Pima Indians Diabetes dataset chosen by our group for the algorithm comparison. The dataset consists of n=768 instances (patients) and m=8 integer and real number attributes. All patients were female and at least 21 years of age. The eight attributes for each patient are as follows (the ninth item being the vector's label):

1) Number of times pregnant
2) Plasma glucose concentration
3) Diastolic blood pressure in mm of Hg
4) Triceps skin fold thickness in mm
5) Two-hour serum insulin in $\mu$ U/ml
6) Body mass index
7) Diabetes pedigree function
8) Age in years
9) Class variable (diabetic or non-diabetic)

The dataset does not define a training set and test set, the results of the BayesNet algorithm mentioned in this article was the result of a 10-fold cross-validation. A quick comparison between well-established methods and evolutionary methods can be performed more easily through the use of the Weka software.

### B. *Encountered problems*

We experimented with Bayesian Networks and multinomial Bayes classifiers and researched on the possibility of how to optimise their classification accuracy. Most of the well established algorithms already offer good optimisation and parametrisation techniques. Because of this an improvement in classification accuracy was unlikely, especially considering the possibility of applying different algorithms consecutively to achieve optimal results.

### *C. Implementation*

The implementation of the evolutionary algorithms was performed in Java and based on the Naive Bayes algorithm implemented in Weka[22] to illustrate the mechanisms. The weights of the different attributes used to calculate the probability distribution were initialised at random and improved by the evolutionary mechanism until a certain generation was reached. This phase was performed against a training set. Each individual chromosome consists of the necessary parameters to calculate the density function need for the normal distribution class membership prediction:

$$f(x) = \frac{e^{(-(x-\mu)^2/(2\sigma^2)}}{\sigma\sqrt{2\pi}}.$$

The fitness or accuracy of prediction for each individual chromosome is evaluated by calculating the probabilities of class memberships for each instance of the training set:

**Algorithm 1** Fitness function

```
procedure FITNESS FUNCTION(weightings[], List trainingSet)
    for all instances of trainingset do
        for i = 1 to NumberOfClasses do
            for all attribute to MaxNumberAttributes do
                probability[i] *= NORMDISTRIBUTION(attribute + weightings[attribute])

        index ←INDEX OF MAX(probability[])

        if index == CLASS OF(instance) then
            INCREMENT(fitness)
        else
            DECREMENT(fitness)

    RETURN fitness
```

After a ranking based on the quality of the different solutions is concluded, the evolutionary functions like mutation, crossover and selection, described in the preceding section, can be applied. The different evolutionary functions can easily be implemented to fit the purpose of this optimisation task.

### *D. Results*

TABLE I: Weka Bayesian network prediction performance on the diabetes dataset.

| Metric | Score |
|---|---|
| Correctly Classified Instances | 586 (76.3021%) |
| Incorrectly Classified Instances | 182 (23.6979%) |
| Kappa statistic | 0.4664 |
| Mean absolute error | 0.2841 |
| Root mean squared error | 0.4168 |
| Relative absolute error | 62.5028% |
| Root relative squared error | 87.4349% |
| Total Number of Instances | 768 |

TABLE II: Weka Bayesian network confusion matrix on the diabetes dataset.

| a | b | ← | classified as |
|---|---|---|---|
| 422 | 78 | | a = testedNegative |
| 104 | 164 | | b = testedPositive |

In our experiments the fitness of an individual solution candidate was determined by the number of correctly classified instance in the training set of the diabetes dataset. The adaptation was carried out trough an hill climbing strategy using a standard mutation process to improve randomly selected chromosomes before evaluation would take place. The adaptation function used the same mechanisms as the fitness function to measure the success of the mutation in the adaptation process. To measure the results the testing classification algorithm was conducted in Java, and the classification success was compared to other algorithms already available in the Weka software on the same diabetes dataset.
The usage of an adaptation instead of the classical Darwinian mutation resulted in no significant differences. While Naive Bayes is usually a very efficient classifier (see Table I and Table II), it performed only average on our diabetes data set.

Methods based on Naive Bayes algorithm achieved around 75%. The introduction of attribute weighting trough evolutionary algorithms only resulted in around 68% accuracy.

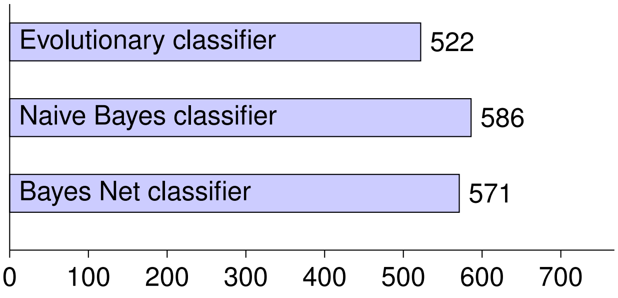

Fig. 3: Correctly identified instances.

### *E. Future challenges*

The application of evolutionary algorithms to well-established machine learning algorithms yielded relatively unimpressive results. However, their advantage of being easily adaptable to a multitude of optimisation tasks offers a huge variety of possibilities to improve machine learning algorithms. We conclude from these at first sight rather negative results, that an optimisation of feature selection algorithms on a data set with a large numbers of instances could result in much better accuracy ratings. We encountered the conceptual problem to identify machine learning methods that would provide a useful application of evolutionary algorithms to gain real advantages in classification accuracy. Regarding this unanticipated limitation we were not able to perform a direct optimisation on those algorithms. Based on the research problems we encountered on the direct optimisation of machine learning methods we propose the future research suggestion to explore the usage of evolutionary algorithms in more diverse fields such as feature selection, which indeed has much potential and opens a lot of research directions.

## IV. Conclusion

In this paper we have discussed the use of EAs for the weighting of nodes in a Bayesian network. While the use of EAs for the weighting values of nodes in a Naïve Bayes did not result in improved results when compared with a more traditional method, this paper has shown that the implementation of evolutionary algorithms in machine learning can be achieved without extensive effort. The implementation of a basic genetic algorithm in Java is relatively straightforward, while altering the Weka Bayesian network machine learning algorithm was also not a significantly large undertaking.

In future, we will concentrate on altering several aspects of the machine learning process using genetic algorithms. The use of an evolutionary algorithm as a replacement for a gradient decent optimisation function, for example, is only *one* potential aspect of machine learning that could be complemented using a genetic algorithm. However, as pointed out emphatically by DeJong, Evolutionary Algorithms are not just function optimisers [23], and future work shall concentrate on using techniques such as genetic algorithm theory to design new methods for feature selection, for example. In this paper, we purposely focused on Bayesian networks as they work using weighted nodes, which makes it convenient for alteration using a genetic algorithm, yet there are dozens of machine learning techniques such as meta-learning (e.g. AdaBoost), decision trees (e.g. Random Tree), or Support Vector Machines, all of which allow for customization and/or adaptation through the use of genetic and evolutionary algorithms.

Three main avenues could be explored in future: the use of such algorithms in 1) optimisation techniques (replacing gradient decent for example), 2) parametrization, and 3) feature selection. As we have shown here, the implementation of EAs in machine learning can be achieved in a relatively short time, meaning much experimentation can be performed quickly in order to discover novel areas where genetic algorithms can compliment machine learning algorithms to achieve better classification results.

We are confident that the alteration of machine learning algorithms through the use of evolutionary algorithms using genetic programming, genetic algorithms, and the like, will be an area of much interest and a promising area of research in the coming years.

## References

[1] G. E. Box, "Evolutionary operation: A method for increasing industrial productivity," *Applied Statistics*, vol. 6, no. 2, pp. 81–101, 1957.

[2] J. H. Holland, "Outline for a logical theory of adaptive systems," *Journal of the ACM*, vol. 9, no. 3, pp. 297–314, 1962.

[3] Z. Zhang, G. Gao, J. Yue, Y. Duan, and Y. Shi, "Multi-criteria optimization classifier using fuzzification, kernel and penalty factors for predicting protein interaction hot spots," *Applied Soft Computing*, vol. 18, no. 0, pp. 115–125, 2014.

[4] J. W. Grzymala-Busse, J. Stefanowski, and S. Wilk, "A comparison of two approaches to data mining from imbalanced data," *Journal of Intelligent Manufacturing*, vol. 16, no. 6, pp. 565–573, 2005.

[5] J. Xie, J. Lei, W. Xie, X. Gao, Y. Shi, and X. Liu, *Novel hybrid feature selection algorithms for diagnosing erythemato-squamous diseases*. Berlin Heidelberg: Springer, 2012, pp. 173–185.

[6] K. Holzinger, V. Palade, R. Rabadan, and A. Holzinger, *Darwin or Lamarck? Future Challenges in Evolutionary Algorithms for Knowledge Discovery and Data Mining*. Heidelberg, Berlin: Springer, 2014, pp. 35–56.

[7] A. Bird, "Perceptions of epigenetics," *Nature*, vol. 447, no. 7143, pp. 396–398, 2007.

[8] A. Handel and S. Ramagopalan, "Is lamarckian evolution relevant to medicine?" *BMC Medical Genetics*, vol. 11, no. 1, p. 73, 2010.

[9] P. A. Kiberstis, "All eyes on epigenetics," *Science*, vol. 335, no. 6069, p. 637, 2012.

[10] F. Emmert-Streib, R. de Matos Simoes, G. Glazko, S. McDade, B. Haibe-Kains, A. Holzinger, M. Dehmer, and F. Campbell, "Functional and genetic analysis of the colon cancer network," *BMC Bioinformatics*, vol. 15, no. Suppl 6, p. S6, 2014.

[11] J. M. Baldwin, "A new factor in evolution," *The American Naturalist*, vol. 30, no. 354, pp. 441–451, 1896.

[12] G. G. Simpson, "The baldwin effect," *Evolution*, vol. 7, no. 2, pp. 110–117, 1953.

[13] T. M. Mitchell, *Machine learning*. Boston (MA): McGraw-Hill, 1997, p. 267.

[14] R. Kruse, C. Borgelt, F. Klawonn, C. Moewes, M. Steinbrecher, and P. Held, *Computational Intelligence: A Methodological Introduction*. Springer, 2013.

[15] E. V. Koonin and Y. I. Wolf, "Is evolution darwinian or/and lamarckian?" *Biology direct*, vol. 4, no. 1, p. 42, 2009.

[16] D. Orvosh and L. Davis, "Using a genetic algorithm to optimize problems with feasibility constraints," in *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*. IEEE, 1994, pp. 548–553.

[17] B. J. Ross, "A lamarckian evolution strategy for genetic algorithms," *Practical handbook of genetic algorithms: complex coding systems*, vol. 3, pp. 1–16, 1999.

[18] D. Whitley, V. S. Gordon, and K. Mathias, "Lamarckian evolution, the baldwin effect and function optimization," in *Parallel Problem Solving from NaturePPSN III*. Springer, 1994, pp. 5–15.

[19] H. Zhang and S. Sheng, "Learning weighted naive bayes with accurate ranking," in *Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on*. IEEE, 2004, pp. 567–570.

[20] G. Holmes, A. Donkin, and I. H. Witten, "Weka: A machine learning workbench," in *Intelligent Information Systems, 1994. Proceedings of the 1994 Second Australian and New Zealand Conference on*. IEEE, 1994, pp. 357–361.

[21] K. Bache and M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[22] R. O. Duda, P. E. Hart *et al.*, *Pattern classification and scene analysis*. Wiley New York, 1973, vol. 3.

[23] K. A. De Jong, "Genetic Algorithms are NOT Function Optimizers." *FOGA*, 1992.