

# PREDICTION OF STELLAR ATMOSPHERIC PARAMETERS USING INSTANCED-BASED MACHINE LEARNING AND EVOLUTIONARY ALGORITHMS

J. FEDERICO RAMÍREZ AND OLAC FUENTES  
Instituto Nacional de Astrofísica, Óptica y Electrónica  
Luis Enrique Erro # 1  
Santa María Tonanzintla, Puebla, 72840, México  
framirez@cseg.inaoep.mx, fuentes@cseg.inaoep.mx

## ABSTRACT

In this paper we show how evolutionary algorithms can improve the predictive accuracy of the nearest neighbor classifier applied to the problem of predicting stellar atmospheric parameters from stellar spectral indices. We use a genetic algorithm in order to find a subset of features that improves the prediction accuracy of the k-nearest neighbors method (KNN). We also implemented an evolution strategy in order to find a weight vector that indicates the relevance of the spectral indices and improves the classifier's predictive accuracy. Our experimental results show that the feature selection performed by the genetic algorithm reduces the running time of KNN by 42% and the error by 5%, while the use of evolution strategies yields an error reduction of 12%.

## KEY WORDS

Applications: Science, Genetic Algorithms, Machine Learning, Optimization.

## 1. Introduction

With the new generation of large spectroscopic surveys, and the rapid development of the Internet, astronomers will have at their disposal enormous amounts of high-quality information. For example, the Sloan Digital Sky Survey, which will map half the northern sky in five different wavelengths, from UV to the near infrared, will gather data for more than 200 million objects requiring an archive of approximately 40 terabytes. The analysis of these databases will benefit the field of stellar population studies, since its concept is based on statistical analyses of properties of individual components that form aggregate populations. To take advantage of all the available information, new tools for intelligent automated data analysis will have to be developed.

In recent years, various techniques developed in the field of artificial intelligence have been applied to the analysis of astronomical data, in an attempt to cope with the problem posed by information overload.

By far the most commonly used approach has been artificial neural networks, which have been used for spectral classification of stars [15], [3], for spectral classification of galaxies [14], for morphological classification of galaxies [16], [1], [10], and for discriminating stars and galaxies in deep-field photographs [11]. In this paper we propose evolutionary algorithms, which have received little attention from the astronomical research community, as an alternative to neural networks for astronomical data analysis.

The organization of the remainder of this paper is as follows: Section 2 gives a brief overview of genetic algorithms, evolution strategies and instance-based learning, Section 3 presents the data used for our experiments, Section 4 presents the algorithm used for the selection and weighting of features. Section 5 presents experimental results and discussion, and Section 6 presents conclusions and outlines directions for future work.

## 2. Instance Based Learning and Evolutionary Algorithms

One of the simplest and most commonly used machine learning methods is k-Nearest-Neighbor (KNN). In this algorithm, we simply store all the training examples, and, when a query is presented, we find the training examples that are most similar to it (its k nearest neighbors), and assign to it an output parameter that corresponds to the average, or weighted average, of the parameters of its neighbors.

KNN assumes all instances correspond to points in an n-dimensional space. The standard Euclidean distance (equation 1) is defined as a measure of similarity between each of the training examples  $x_i$  and the query point  $x_q$ . Given a query point, KNN returns the average target function value of its neighbors  $\hat{f}(x_q)$ , when the target function is continuous, and the most common value if the target function is discrete (equation 2).

$$d(x_q, x_i) = \sqrt{\sum_{r=1}^n (a_r(x_q) - a_r(x_i))^2} \quad (1)$$

where  $a_r$  is the value of  $r$ th attribute of the instance  $x$ .

$$\hat{f}(x_q) = \frac{\sum_{i=1}^k w_i f(x_i)}{\sum_{i=1}^k w_i} \quad (2)$$

where

$$w_i = \frac{1}{d(x_q, x_i)}$$

Despite its simplicity, k-NN is commonly competitive with other more sophisticated machine learning methods, such as neural networks or decision trees. However, one disadvantage of k-NN is that it typically considers all of the attributes of an instance when attempting to retrieve similar training examples from memory. If the target concept depends on only a few of the potentially many available attributes, then the instances that are more relevant to the prediction of the output function may well be a large distance apart. One approach to overcoming this problem is to weight each attribute differently when computing the distance between two instances as is shown in equation 3. This process of stretching the axes in order to optimize the performance of KNN provides a mechanism for suppressing the impact of irrelevant attributes. There are several ways to carry out this stretching as mentioned in [17], in this work we propose that the amount by which each axis can be stretched can be determined automatically using evolution strategies.

$$d(x_q, x_i) = \sqrt{\sum_{r=1}^n w_r (a_r(x_q) - a_r(x_i))^2} \quad (3)$$

Evolutionary Algorithms are a class of probabilistic search algorithms loosely based on biological evolution. They work on a population of individuals, where each of them represents a search point in the space of potential solutions to a given problem. Initially, the algorithm randomly generates a population of individuals; subsequently this population is updated by means of randomized processes of recombination, mutation, and selection. Each individual is evaluated according to the quality of the information it contains (fitness function). The selection process favors the most fit individuals from the current population to reproduce more often than unfit individuals. The recombination process allows to combine information from different members of a population, creating offspring from them. Mutation is an asexual operator that generates random changes to an individual and often provides new relevant information.

Two major representative algorithms of the class of evolutionary algorithms are Genetic Algorithms (GA) [4], [6], and Evolution Strategies (ES) [12], [13]. In the first, the genetic information (chromosome) is represented by a bit string and sets of bits encode the solution. The bit string may be of variable length. The recombination process is called crossover in this algorithm. It is a sexual operation that creates two offspring strings from two parent strings copying selected bits from each parent. The crossover operation is repeated as often as desired, usually until the new generation is completed. Mutation is carried out by randomly changing the value of a single bit (with small probability) from the bit strings. There are four selection schemes commonly used in genetic algorithms [5]: 1) proportional reproduction or roulette wheel selection, 2) ranking selection, 3) tournament selection and 4) Genitor ( or "steady state") selection. Commonly, some of the best individuals are copied into the next generation population intact. This operation is known as elitism.

In the evolution strategy, as opposed to the genetic algorithms, each individual is represented by a vector of real numbers. This is a good representation when the problem at hand deals with continuous parameters. Tomas Bäck in [2] shows a variety of different recombination mechanisms used in evolution strategies. Typical examples of them are discrete recombination (which is comparable to uniform crossover in genetic algorithms), and intermediate recombination (which it is commonly used as an arithmetic average with some variants). These operators can be used in sexual or panmictic form. In the sexual form, every element of an offspring is the result of recombination between two individuals randomly chosen from the parent population. In panmictic form, each element of an offspring may be the result of recombination among one individual and several other individuals randomly chosen from the parent population.

Each element of an individual has associated to it a standard deviation. The mutation operator is applied independently to each of elements of an individual. It carries out as shown in equation 5. The standard deviation may be mutated using a multiplicative, logarithmic normally distributed process as shown in equation 4. A theoretically confirmed rule proposed by Rechenberg for a deterministic adjustment of standard deviation during the evolution is called the 1/5-success rule, which reflects that, on average, one out of five mutations should cause an improvement in the objective function values to achieve best convergence rates: The ratio of successful mutations to all mutations should be 1/5, if it is greater than 1/5, increase the standard deviation, if it is smaller, decrease the standard deviation.

$$\sigma'_i = \sigma_i \cdot \exp(N(0, 1) + N_i(0, 1)) \quad (4)$$

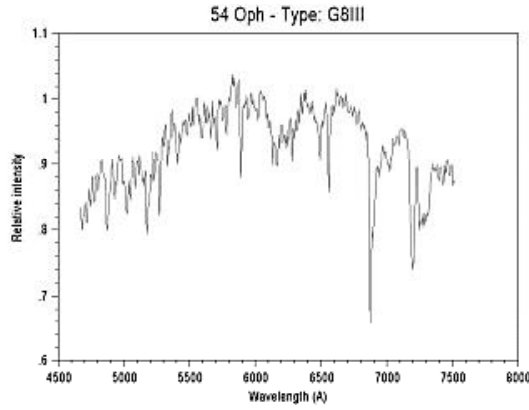


Figure 1. Sample of stellar spectrum

$$x'_i = x_i + \sigma'_i \cdot N(0, 1) \quad (5)$$

Where  $N(0, 1)$  is a normally distributed random variable having an expectation of zero and a standard deviation of one,  $N_i(0, 1)$  indicates that the random variable is sampled anew every time the index  $i$  changes.

The selection operator used in evolution strategies is completely deterministic. In  $(\square + \square)$ -selection, this operator selects the  $\square$  best individuals out of the union of parents and offspring to form the next parent generation, and in  $(\square, \square)$ -selection this operator selects the  $\square$  best individuals out of the offspring only; for this  $\square \square \square$  is required. We can apply different strategies to the parent population to obtain a new generation.

### 3. Data

Astronomers have a way to determine the chemical composition and physical nature of stars by analyzing their spectra, which is a plot of flux density as a function of wavelength, as shown in figure 1. Stellar spectra consist of a continuous spectrum, or continuum, with narrow discontinuities superimposed. These discontinuities are called absorption lines and are caused by the presence of certain atoms in the star's atmosphere. Each absorption line is presented as a valley in a stellar spectrum. The depth of each line indicates its strength. The width of each line indicates the range of wavelengths. Finally, each line has a specific shape. All of these characteristics convey information about the stars. An expert astronomer can analyze these lines and estimate with good accuracy several of the most important properties of the star including its temperature, surface gravity and metal content.

Instead of using the spectra as input data, a very large degree of compression can be attained if we use a measurement of the strength of several selected ab-

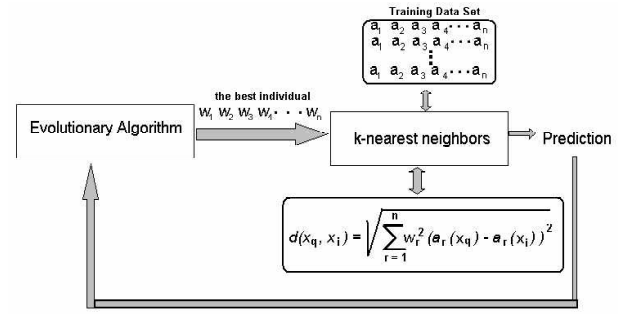


Figure 2. Feature weighting using evolution algorithms

sorption lines that are known to be important for predicting the stellar atmospheric parameters. In this work we use a library of such measurements, which are called spectral indices in the astronomical literature, due to Jones [8]. The dataset used in our experiment consists of 24 spectral indices for 651 stars together with their estimated effective temperatures, surface gravities and metallicities. It was observed at Kitt Peak National Observatory and has been made available by the authors at an anonymous ftp site at the National Optical Astronomy Observatories.

### 4. The Methods

Figure 2 shows the structure of the wrapper system[7] that was implemented. It uses evolutionary algorithms to find a vector of weights (real-valued or binary) to improve the accuracy in the prediction of stellar atmospheric parameters from spectral indices. It uses the KNN learning algorithm to evaluate each set of weights provided by the evolutionary algorithm. We use a genetic algorithm (GA) to find the best binary-valued weight vector that represents the relevant spectral indices. The fitness function of the GA is shown in equation 6. The goal of this function is to maximize the predictive accuracy of KNN and to reduce the size of spectral indices dataset.

$$\text{GA fitness} = \sqrt{\frac{\sum_{i=1}^n \square (f(x_i) - f(x_i))}{\square}} + \square \frac{\square'}{\square} \quad (6)$$

where  $\square$  is the number of spectral indices in the original dataset,  $\square'$  is the number of spectral indices of the new subset found by the genetic algorithm, and  $\square$  is an adjustment constant. The first term of the fitness function corresponds to the prediction accuracy and the second term corresponds to a penalty favoring individuals with fewer attributes.

Each bit string in the GA population represents a possible selected data subset with  $\square'$  relevant spectral

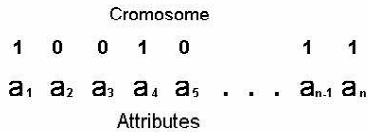


Figure 3. Encoding of chromosome in the genetic algorithm

indices as shown in figure 3. Each bit position represents a spectral index, if the bit is one ('1') the spectral index is considered in the subset and conversely if the bit is zero ('0'), it is not considered. It means that KNN only considers the spectral indices chosen by the genetic algorithm bit string when it computes the distance between two instances.

We use an evolution strategy to find a real-valued weight vector that multiplies the spectral indices. That corresponds to stretching the axes in Euclidean space, shortening the axes that correspond to less relevant attributes, and lengthening the axes that correspond to more relevant attributes [1]. This algorithm has as fitness function the absolute mean error of the prediction (equation 7).

$$\text{ES fitness} = \sqrt{\frac{\sum_{i=1}^n |f(x_i) - \hat{f}(x_i)|}{n}} \quad (7)$$

## 5. Experimental Results

These methods have been applied to Jones's catalog data set with 651 stars and 24 spectral indices together with their estimated stellar atmospheric parameters effective temperature, surface gravity and metallicity.

We implemented the GA with a population size of 100 individuals per generation and the following parameters and genetic operators: elitism, retaining the two best individuals of each generation; two-tournament selection operation; uniform crossover operation and mutation operation with a probability of 0.01 per bit. We used 3-NN as predictive classifier. For the experiments reported we used five-fold cross validation. In five-fold cross validation, the training set is randomly divided into 5 disjoint equal-sized subsets. The experiment is applied 5 times, each time using one of the subsets for testing and the other four for training.

The parameters of evolution strategies used in this experiment were: a population size of 50 individuals, discrete recombination on the whole vector on 20% of the parent population, discrete recombination on object parameters and panmictic-intermediate-generalized on strategy parameters on another 20% of

	Teff Error [K]	Log Error [dex]	[Fe/H] Error [dex]
KNN	124.64	0.233	0.126
BW-KNN	118.07	0.214	0.122
RW-KNN	110.27	0.213	0.114

Table 1. Mean absolute errors in the prediction of stellar atmospheric parameters.

the parent population, and mutation on the remaining individuals. We also used the 1/5-success rule on the strategy parameters of the selected individuals of the new parent population.

Table 1 presents the experimental results obtained applying the nearest neighbors method to the data in order to predict the stellar parameters temperature, surface gravity and metallicity from spectral indices. The difference among the methods is the term  $w_i$  of the distance metrics mentioned in equation 3 used by the algorithm. In KNN,  $w_i = 1$  for all  $i$  indicating that we should consider all the attributes as equally relevant. We can see in the table that it yields the poorest result due possibly to the irrelevance of some attributes. Figure 5 shows a plot of the results obtained applying 3NN to the original dataset. In BW-KNN,  $w_i \in [0, 1]$  as indicated by the best individual provided by the genetic algorithm. This means that we should consider as relevant those attributes that the individual marks as 1's. The genetic algorithm found 14 spectral indices as the relevant features, reducing slightly the prediction error and reducing by 42% the classification run time. Figure 6 shows a plot of the results obtained applying BW-3NN to the original dataset. Finally, in RW-KNN,  $w_i$  are normalized real numbers between 0 and 1 provided by the evolution strategy indicating the relevance of the attributes. This algorithm provides a set of values to weight the attributes as shown in figure 4 where we observe that there are few attributes that really are relevant. Figure 7 shows the results obtained applying RW-3NN using the weights provided by the evolution strategy to the original dataset. This method slightly increases the running time but improves the prediction accuracy better than the other methods.

## 6. Conclusions

In this paper we have presented an approach to improve the prediction accuracy of k-Nearest Neighbor using evolutionary algorithms. A genetic algorithm generates a set of binary values for feature selection from a dataset while an evolution strategy generates a set of real values to weight the dataset features (spectral indices). Our experimental results on astronomi-

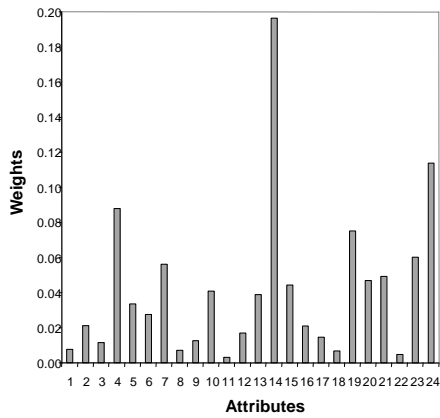


Figure 4. Weights provided by the evolution strategy.

cal data show that using this method over a large data set with many features provides the following advantages.

1. The GA reduces size data set so that k-NN can classify faster.
2. The GA identifies relevant spectral indices, so that the data are easier to understand, which may be useful for other applications.
3. Both methods increase the predictive accuracy due to the elimination of noisy or irrelevant attributes.

Future work will attempt to combine evolutionary algorithms with other machine learning algorithms.

## References

- [1] A. Adams and A. Woolley. Hubble classification of galaxies using neural networks. *Vistas in Astronomy*, 38:273–280, 1994.
- [2] T. Bäck. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, 1996.
- [3] C. A. L. Bailer-Jones, M. Irwin, and T. von Hippel. Physical parametrization of stellar spectra: the neural network approach. *Monthly Notices of the Royal Astronomical Society*, 292(1):157–166, 1997.
- [4] D. Golberg. *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley Publishing Company, 1989.
- [5] D. Goldberg and K. Deb. A comparative analysis of selection schemes used in genetic algorithms. In G. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 6–13. Morgan Kaufmann, Berlin, 1991.
- [6] J. Holland. *Adaptation in Natural and Artificial Systems*. MIT Press, 1975.
- [7] G. H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. *Proceeding of the Eleventh International Conference on Machine Learning*, pages 121–127, 1994.
- [8] L. A. Jones. PhD thesis, University of North Carolina, Chapel Hill, North Carolina, 1996.
- [9] T. M. Mitchell. *Machine learning*. Mc. Graw-Hill, 1997.
- [10] A. Naim, O. Lahav, L. S. Jr., and M. C. Storrie-Lombardi. Spectral classification with principal component analysis and artificial neural networks. *Monthly Notices of the Royal Astronomical Society*, 1995.
- [11] S. C. Odewahn and M. L. Nielsen. Star-galaxy separation using neural networks. *Vistas in Astronomy*, 38:281–285, 1994.
- [12] I. Rechenberg. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog Verlag, Stuttgart, 1973.
- [13] H.-P. Schwefel. *Evolution and Optimum Seeking*. Wiley, New York, 1995.
- [14] L. Sodr e and H. Cuevas. Spectral classification of galaxies. *Vistas in Astronomy*, 38:286–291, 1994.
- [15] M. C. Storrie-Lombardi, M. J. Irwin, T. von Hippel, and L. J. Storrie-Lombardi. Spectral classification with principal component analysis and artificial neural networks. *Vistas in Astronomy*, 38:331–340, 1994.
- [16] M. C. Storrie-Lombardi, O. Lahav, L. Sodr e, and L. J. Storrie-Lombardi. Morphological classification of galaxies by artificial neural networks. *Monthly Notices of the Royal Astronomical Society*, 258(8):12, 1992.
- [17] D. Wettschereck, D. W. Aha, and T. Mohri. A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review*, 11(1-5):273–314, 1997.

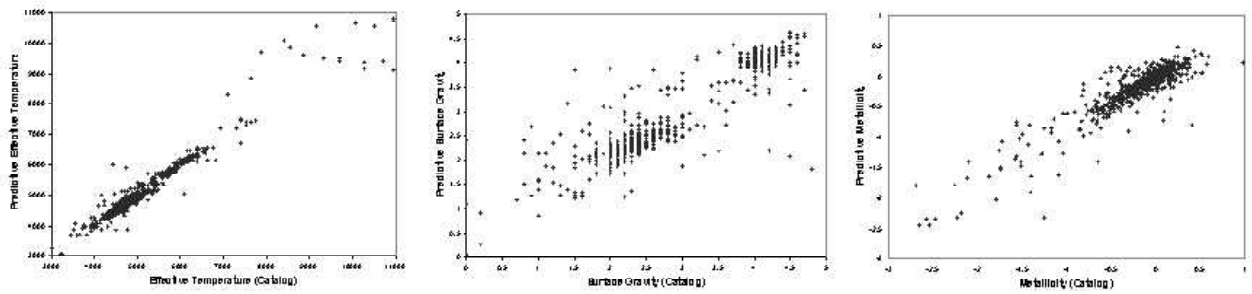


Figure 5. Catalog versus predicted parameters using KNN with the original dataset

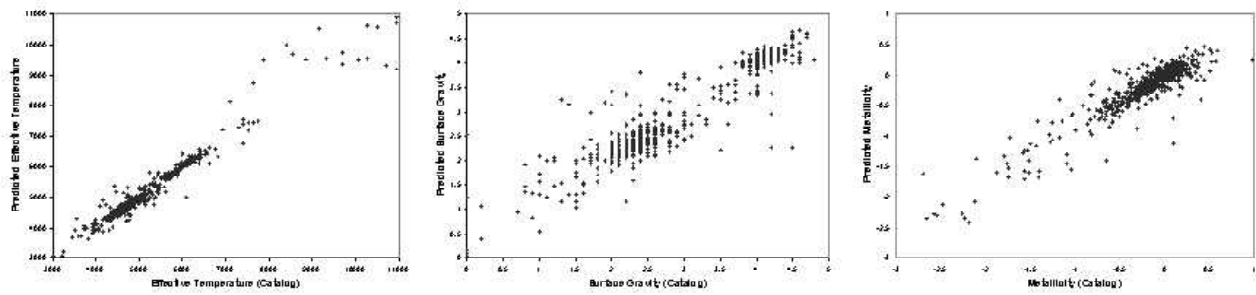


Figure 6. Catalog versus predicted parameters with binary-valued weighted KNN

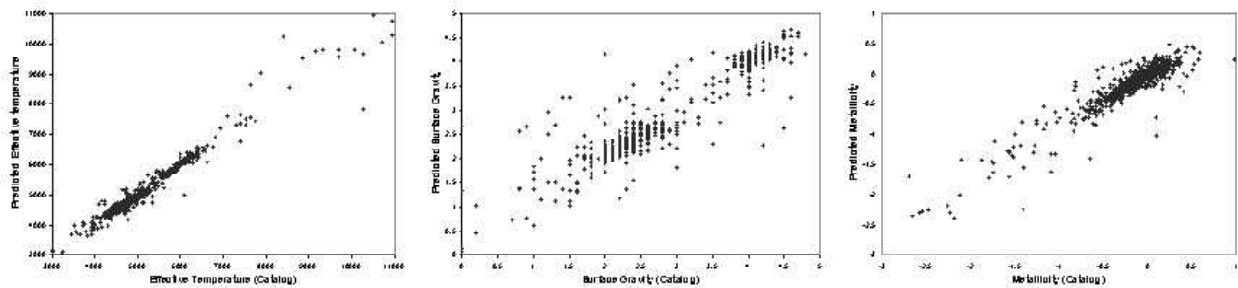


Figure 7. Catalog versus predicted parameters with real-valued weighted KNN.



ERROR: undefined  
OFFENDING COMMAND: From  
STACK: