Prediction of Stellar Atmospheric Parameters Using Instance-Based Machine Learning and Genetic Algorithms

J. Federico Ramírez (framirez@cseg.inaoep.mx), Olac Fuentes (fuentes@cseg.inaoep.mx) and Ravi K. Gulati (gulati@inaoep.mx)
Instituto Nacional de Astrofísica, Óptica y Electrónica
Luis Enrique Erro # 1
Santa María Tonanzintla, Puebla, 72840, México

Abstract. In this paper we present a method for the automated prediction of stellar atmospheric parameters from spectral indices. This method uses a genetic algorithm (GA) for the selection of relevant spectral indices and prototypical stars and predicts their properties using the k-nearest neighbors method (KNN). We have applied the method to predict the effective temperature, surface gravity, metallicity, luminosity class and spectral class of stars from spectral indices. Our experimental results show that the feature selection performed by the genetic algorithm reduces the running time of KNN up to 92%, and the predictive accuracy error up to 35%.

Keywords: Prediction, Genetic Algorithms, Machine Learning, Optimization.

1. Introduction

With the new generation of large spectroscopic surveys and the rapid development of the Internet, astronomers 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 the UV to the near infrared, will gather data for more than 200 million objects, requiring an archive of approximately 40 terabytes (Szalay et al., 2000). To take advantage of all the available information, new tools for intelligent automated data analysis have to be developed. In recent years, various artificial intelligence techniques have been applied to the analysis of astronomical data, in an attempt to cope with the problem posed by the information overload created by the presence of numerous and sophisticated astronomical data collection devices. By far the most commonly used approach has been artificial neural networks (ANNs) (Rumelhart et al., 1986). Neural networks have been used for spectral classification of stars (Storrie-Lombardi et al., 1994; Gulati et al., 1994; Bailer-Jones et al., 1997), for spectral classification of galaxies (Sodré and Cuevas, 1994), for morphological classification of galaxies (Storrie-Lombardi et al., 1992; Adams and Woolley, 1994; Naim et al., 1995) and for discriminating stars and

© 2002 *Kluwer Academic Publishers. Printed in the Netherlands.*

galaxies in deep-field photographs (Odewahn and Nielsen, 1994). While ANNs have had remarkably good results in some problem domains, they present some drawbacks that make the investigation of alternative and complementary automated methods desirable. One problem with neural networks is their large training time. Also, since they perform local search in parameter space, they often converge to local minima. A final drawback is that once trained, neural networks are black boxes, in the sense that it is very difficult for humans to interpret the rules ANNs learn. Few works in Astronomy have used genetic algorithms, but they have potential applications to a wide range of problems in this area because they are robust and easy to code and modify, compared to conventional optimization methods, as described in (Charbonneau, 1995).

In this paper, a genetic algorithm combined with the nearest neighbor algorithm is used to extract a subset of relevant spectral indices and prototypical stars from a catalog to predict their stellar atmospheric parameters and spectral classification. Our experimental results show that the elimination of redundant and irrelevant information by means of the GA reduces the running time of KNN up to 92%, and the predictive accuracy error up to 35%.

The organization of the remainder of this paper is as follows: Section 2 gives a brief overview of GAs and instance-based learning, Section 3 presents the data used for our experiments, Section 4 presents the algorithm used to select the spectral indices and prototype stars, Section 5 presents experimental results and discussion, and Section 6 presents conclusions and outlines directions for future work.

2. Instance-Based Learning and Genetic Algorithms

One of the simplest and most commonly used machine learning methods is *k-Nearest-Neighbor (KNN)* (Fix and Hodges, 1951). This algorithm simply stores all the training examples $(x, f(x))$ in memory, where x is the input and is defined by an n-dimensional feature vector $(a_1, a_2, a_3, \ldots, a_n)$ and $f(x)$ is the output. Generalization beyond these training examples is postponed until a new instance has to be classified.

Given a query x_q, KNN finds the k training examples that are most similar to it (its k nearest neighbors) using the standard Euclidean distance as a measure of similarity between each training example x_i and the query point x_q, (as shown in equation 1).

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

where a_r is the value of the rth attribute of the instance x.

For a discrete-valued target function, KNN returns the most common target function value among the neighbors of the query point. In the case of a real-valued target function, KNN returns the weighted average target function value $\hat{f}(x_q)$ of the neighbors:

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

where

$$w_i = \frac{1}{d(x_q, x_i)^2}$$

Thus, points that are closer to the query point are weighted more heavily.

Despite its simplicity, KNN is highly competitive with other more sophisticated machine learning methods, such as ANNs or Decision Trees (Quinlan, 1986). However, one disadvantage of KNN is its computation time, due to the search for the k nearest training examples in memory performed each time a query needs to be answered. Another disadvantage is that KNN considers all of the attributes to compute the distance between a query point and the training examples; if the target concept depends only on a few features and there are many irrelevant features the distance may mislead KNN's prediction.

One approach to overcome this problem is to select relevant attributes for computing the distance between two instances:

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

Here $b_r \in [1, 0]$ and $[b_1,...,b_n]$ is a binary vector that indicates the subset of features to be considered in the prediction. A suitable choice of b can suppress the impact of irrelevant attributes. In this work we propose to use GAs for searching the feature and example subsets to improve the predictive accuracy of the KNN method.

GAs (Holland, 1975; Golberg, 1989) are a class of probabilistic search algorithms loosely based on biological evolution. GAs work on a population of individuals, where each of them represents a search point in the space of potential solutions to a given problem. A problem is represented by a set of parameters identifying it (*phenotype*); these parameters are encoded in *structures* (*genotypes*) to be manipulated by the GA; these structures form the *individual*. Often these individuals are also called *chromosomes*. Chromosomes are made of units (*genes*) arranged in a linear form or *bit string*, which may be of variable length.

```
GA
  begin
      t ← 0                              t is generation number
      Initialize P(t)                    P is the population
      fitness P(t) ← evaluate P(t)
      while (best_fitness P(t) < fitness_threshold) do
      begin
          t ← t + 1
          select P(t) from P(t − 1)
          crossover P(t)
          mutate P(t)
          fitness P(t) ← evaluate P(t)
      end
      return individual with best_fitness from P
  end
```

Figure 1. Genetic Algorithm

The position (*loci*) of these units and their possible states (*alleles*) define a property of an individual. The genes control the inheritance material.

Initially, the algorithm randomly generates a population of individuals; subsequently, this population is updated by means of randomized processes of recombination, mutation, and selection as shown in Figure 1. Each individual is evaluated according to a *fitness function*, which indicates the quality of the information it contains. The selection process favors fit individuals from the current population to reproduce more often than unfit individuals. The recombination process combines information from different members of a population, creating offspring from them. Mutation is an operator that generates random changes to an individual and often provides new relevant information.

The recombination process is called crossover. It is a sexual operation that creates two offspring strings from two parent strings by copying selected bits from each parent. There are three common ways of carrying out the crossover operation: single-point crossover, two-point crossover, and uniform crossover. In single-point crossover, offspring are formed by randomly choosing a bit position that divides the parents and offspring strings into two parts as shown in Figure 2(a). The first part of parent A is copied into the first part of the offspring A and the second part of parent B is copied into the second part of the offspring A. Another offspring B can be formed by copying the first part of parent B into the first part of offspring B and the second part of the

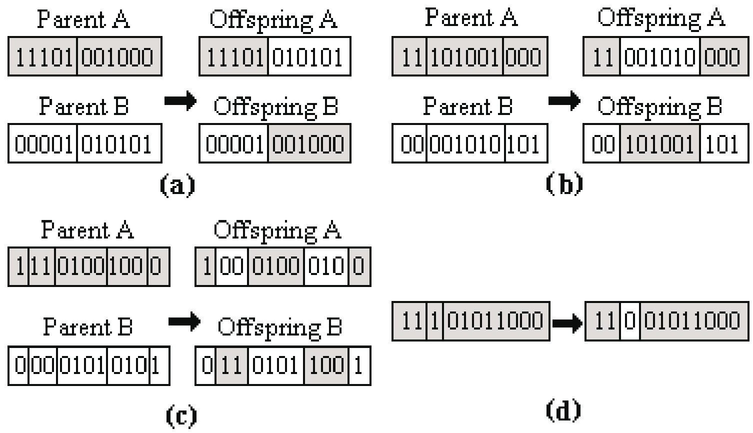


Figure 2. (a) Single-point crossover, (b) Two-point crossover, (c) Uniform crossover, (d) Mutation.

parent A into the second part of offspring B. The two-point crossover operation randomly chooses two bit positions that divide the parents and offspring strings in three parts as shown in Figure 2(b). Offspring A is constructed by copying the first and third parts of parent A into the corresponding positions of offspring A and copying the second part of parent B into the second part of offspring A. The other offspring B is constructed by copying the first and third parts of parent B in the corresponding position of offspring B and the second part of parent A into the middle position of the offspring B. In uniform crossover, each offspring is constructed by randomly choosing a parent for each bit position, as shown in Figure 2(c). 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. These operators are shown in Figure 2(d).

There are four selection schemes commonly used in genetic algorithms (Goldberg and Deb, 1991): 1) proportional reproduction or roulette wheel selection, 2) ranking selection, 3) tournament selection and 4) Genitor (or "steady state") selection. The first selection method (Jong, 1975) chooses individuals for crossover according to their objective function values $f(h_i)$. The probability of selecting an individual is directly proportional to its *fitness* function value (equation 2).

$$p(h_i) = \frac{f(h_i)}{\sum_{j=1}^{p} f(h_j)} \qquad (2)$$

In the second selection method (Baker, 1985), the idea is to sort the population from best to worst, assign the number of copies that each individual should receive according to a non-increasing assignment function and then perform proportionate selection according to that assignment. The third selection method (attributed to unpublished work by Wetzel and later studied by Brindle (1981)) consists of selecting an individual for further genetic processing with the best fitness from a group of them randomly chosen from the population. This operation is repeated as often as desired. The fourth method selects one individual during a generation. This is done by choosing the best individual from the population according to its chance of being selected and choosing the worst individual for being replaced (Whitley, 1989). Commonly, some of the best individuals are copied into the next generation population intact. This operation is known as elitism.

3. Data

Jones (1996) has produced an homogeneous catalog of 48 spectral indices for 684 stars observed at Kitt Peak National Observatory with the coudé feed instrument. The spectral indices were measured from the spectra in the wavelength regions 3820-4500Å and 4780-5450Å by following the definition of the Lick indices (Worthey et al., 1994), the Rose (1994) indices and new Lick-type Balmer indices (Jones and Worthey, 1995). In our application, we used the indices in conjunction with physical atmospheric parameters given in the catalog. The spectral indices are widely used for interpretation of observational and physical properties of not only individual stars, but also galaxies and an automated procedure was developed to measure them from spectra (Gorgas et al., 1998).

There are 32 spectral indices defined in the wavelength region 3820-4500Å (4000Å region), while the region 4780-5460Å (5000Å region) has 16 indices. We list in table I the indices present in Jones catalog. The wavelength region of 4000Å has 7 Lick indices both measured in the form of equivalent width and magnitude units. Also, this region contains 18 Rose indices. Meanwhile, from the 5000Å region, Jones has derived 8 Lick indices both in equivalent width and magnitude units. In this paper we used the indices with the information of atmospheric parameters to develop a machine learning method. There are two sets of values of atmospheric parameters: One is taken directly from the literature and the other is derived after applying a systematic correction; in this paper we use the later one. MK spectral classes and luminosity

Table I. The indices present in Jones Catalog

4000 region	Lick indices (measured in the form of equivalent width and magnitude) Hδ, CN_1, CN_2, Ca4227, G4300, Hγ and Fe4383
	Rose indices Hδ/λ4045, Hδ/λ4063, SrII/λ4045, SrII/λ4063, pGband, Hγ/λ4325, λ4289/λ4271, λ4384/λ4352, p[Fe/H], Ca II, λ3888/λ3859, p4220/p4209, eqw Hγ, eqw Ca I, eqw 4045, eqw Hδ, eqw Ca II K, eqw Ca II H
5000 region	Lick indices (measured in the form of equivalent width and magnitude) Hβ, Fe5015, Mg_1, Mg_2, Mg_b, Fe5270, Fe5335, and Fe5406

classes were obtained from the SIMBAD bright star catalog (Hoffleit and Jaschek, 1991; Cayrel de Strobel et al., 1992).

4. The Methods

Figure 3 shows the structure of the system that was implemented. It uses a GA to find the best binary-valued weight vector that represents the relevant spectral indices and prototypical stars. The fitness function of an individual is

$$fitness = \sqrt{\frac{\sum_{i=1}^{n}(\hat{f}(x_i) - f(x_i))^2}{n}} + C\frac{m' \cdot n'}{m \cdot n}$$

where n and m are the number of stars and spectral indices in the original set; n' and m' are the number of stars and spectral indices in the subset found by the GA; and C 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 spectral indices and/or stars. The goal of this function is to maximize the predictive accuracy of KNN and to reduce the size of the data set.

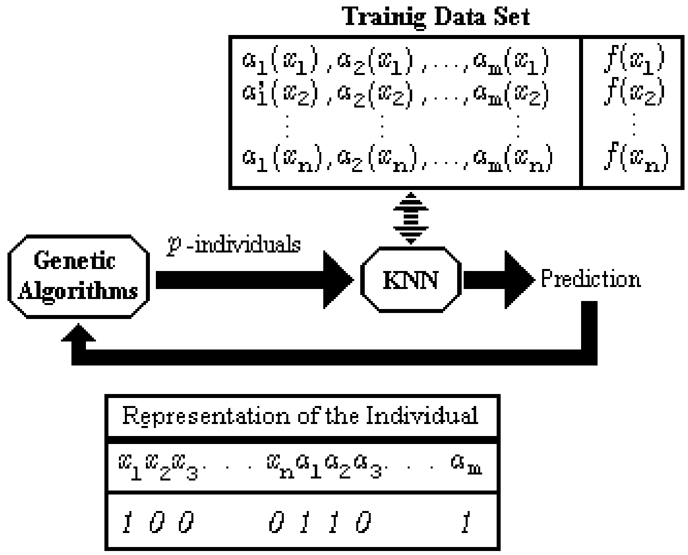


Figure 3. System Structure

Each bit string ($m + n$ bits) in the GA population represents a possible selection of a data subset with n' prototypical stars and m' relevant spectral indices. The first n bit positions represent the stars and the last m bit positions represent the spectral indices. If the bit is one ('1') the star or spectral index in this position is considered in the subset, conversely if it is zero ('0'), it is not considered. This means that KNN only considers the spectral indices chosen by the GA bit string when it computes the distance between two instances and that only those stars chosen by the GA are considered as potential nearest neighbors to the query point.

5. Experimental Results

In our experiments we used Jones's catalog of spectral indices for predicting effective temperature (T_eff), surface gravity, metallicity, luminosity and MK spectral classification.

We implemented this method with 3NN as a predictive classifier. The selection of K=3 was due to experimental tests; using K $>$ 3 did not improve significantly the predictive accuracy, while slightly increasing the running time of the algorithm. Also, we used the following genetic operators: Elitism, retaining the best two individuals in each generation, four-tournament selection, uniform crossover, and mutation operation with 0.01 probability per bit. Figures 4, 5, and 6 plot the catalog versus predicted temperature, surface gravity, and metallicity parameters obtained by the KNN algorithm using the original data sets and after applying the GA. Figure 7 shows the results obtained using the MK spectral classes as the predicted parameter. To use this information with this method we encoded the spectral classes as follow:

$$Spectral\_Code = 10 * a_1 + a_2 + a_3$$

were a_1 varies from 1 to 7 for classes O to M, a_2 corresponds to subclasses from 0 to 9 and a_3 is the fraction of the subclasses. For example, a star with spectral type A9.5 is represented by code 39.5, where a_1=3, a_2=9 and a_3=0.5. Figure 8 shows the confusion matrices of the catalog versus predicted luminosity classes after applying the method.

Figures 9 and 10 show the behavior of the genetic algorithm used for predicting each of the stellar atmospheric parameters. In Figure 9 each plot shows the prediction rms error using 3NN with the stars and spectral indices generated by the GA, and the prediction rms error using 3NN with the original data set. In Figure 10 each plot shows the fitness using the stars and spectral indices generated by the GA, and the fitness using the original data set. The fitness function falls down quickly during the first few generations and it decreases more slowly in later generations. The GA was stopped when no progress was detected during a predefined number of generations. This happened after 285 generations for T_eff, 176 generations for surface gravity, 268 generations for metallicity, 311 generations for luminosity classification, and 322 generations for MK spectral classification.

In our experiment, we used 642 stars from the catalog for the prediction of atmospheric parameters, eliminating 42 stars because some of the parameters were missing in the catalog. For the same reason, we used 603 stars for the prediction of luminosity classes and 614 stars for the prediction of MK classes from the catalog. Figure 11 shows the prototypical stars and relevant spectral indices found by the GA, also it shows the rms prediction error of the 3NN algorithm using the original dataset and the dataset obtained by the GA. Also, it shows the reduction of the running time of KNN, considering that the running time of KNN is Θ(nm). Tables II and III show the spectral indices found by the GA for predicting each atmospheric parameter and classification in the 4000Å and 5000Å region, respectively.

Table II. spectral indices selected in the 4000 region by GA

Spectral indices	T_eff	Surface G.	Metallicity	Luminosity	MK
Hδ width	-	-	-	-	-
CN_1 width	-	-	Y	-	-
CN_2 width	-	-	-	-	-
Ca4227 width	-	Y	Y	-	-
G4300 width	-	-	-	-	-
Hγ width	Y	-	-	-	-
Fe4383 width	-	Y	-	-	-
Hδ mag	-	-	-	-	-
CN_1 mag	-	Y	Y	Y	-
CN_2 mag	-	Y	-	-	-
Ca4227 mag	-	-	Y	-	-
G4300 mag	-	-	-	Y	-
Hγ mag	-	-	-	-	-
Fe4383 mag	-	-	-	-	-
Hδ/λ4045	-	-	Y	-	Y
Hδ/λ4063	-	-	Y	-	Y
SrII/λ4045	-	-	-	Y	-
SrII/λ4063	-	Y	-	Y	-
pGband	-	-	Y	-	-
Hγ/λ4325	-	-	-	-	Y
λ4289/λ4271	-	Y	Y	Y	-
λ4384/λ4352	-	-	Y	-	-
p[Fe/H]	-	-	Y	-	-
Ca II	Y	-	-	-	-
λ3888/λ3859	-	-	-	-	-
p4220/p4209	-	-	-	-	-
eqw Hγ	-	Y	Y	Y	Y
eqw Ca I	-	-	-	Y	-
eqw 4045	-	Y	Y	-	-
eqw Hδ	Y	-	Y	-	-
eqw Ca II K	-	Y	Y	-	Y
eqw Ca II H	-	Y	-	-	-

Table III. spectral indices in the 5000 region selected by GA

Spectral indices	T_eff	Surface G.	Metallicity	Luminosity	MK
Hβ width	-	-	Y	-	-
Fe5015 width	-	-	Y	Y	-
Mg_1 width	Y	-	-	-	-
Mg_2 width	-	-	-	-	Y
Mg_b width	-	-	Y	Y	-
Fe5270 width	-	-	-	Y	-
Fe5335 width	Y	-	Y	-	-
Fe5406 width	-	-	-	-	Y
Hβ mag	Y	Y	-	-	Y
Fe5015 mag	-	-	-	-	-
Mg_1 mag	-	-	-	-	-
Mg_2 mag	-	-	-	-	-
Mg_b mag	Y	Y	-	-	Y
Fe5270 mag	-	-	Y	-	-
Fe5335 mag	-	-	Y	-	-
Fe5406 mag	Y	-	-	-	-

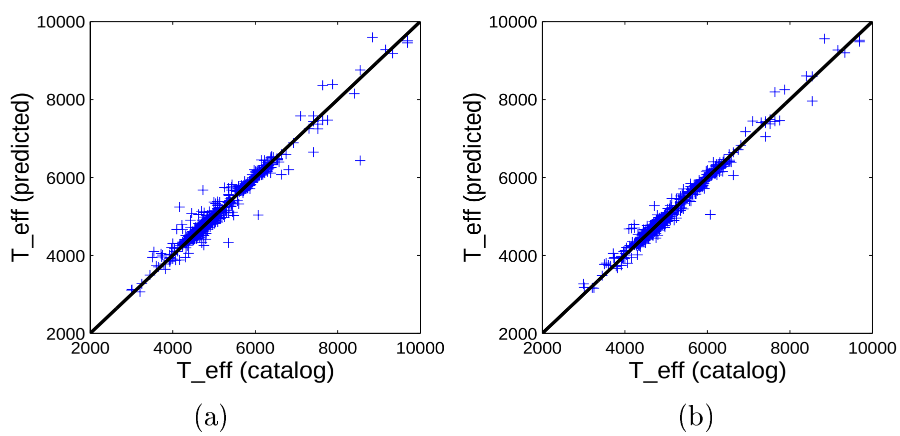
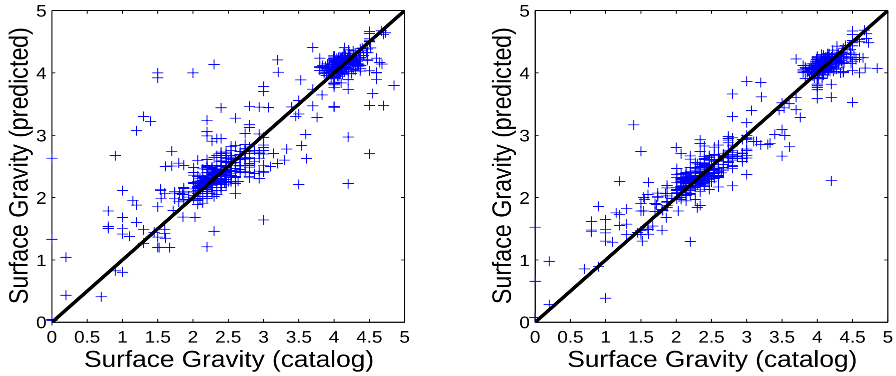
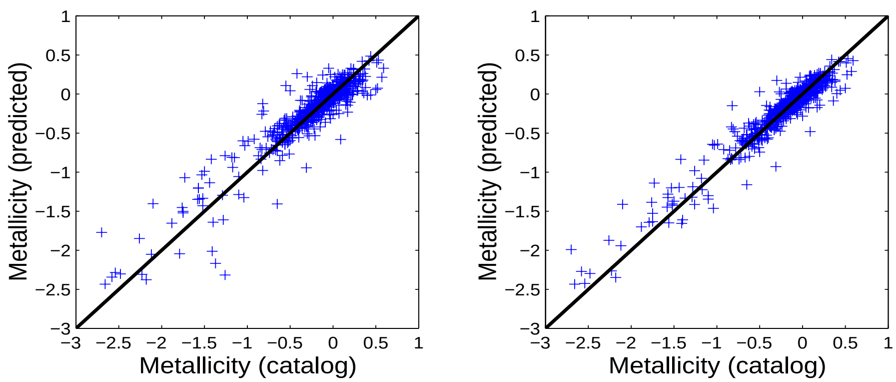


Figure 4. Catalog versus predicted effective temperature parameter using the original (642 stars and 48 spectral index) (a) and GA data set (302 stars and 8 spectral indices) with 3NN predictive classifier (b).



(a) (b)

Figure 5. Catalog versus predicted surface gravity parameter using the original (642 stars and 48 spectral index) (a) and GA data set (310 stars and 12 spectral indices) with 3NN predictive classifier (b).

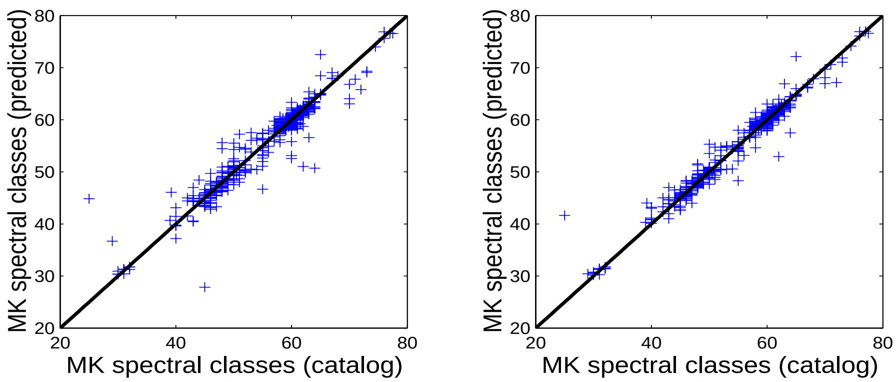


(a) (b)

Figure 6. Catalog versus predicted metallicity parameter using the original (642 stars and 48 spectral index) (a) and GA data set (316 stars and 20 spectral indices) with 3NN predictive classifier (b).

6. Conclusions

In this paper we have presented an approach to improve the prediction accuracy of stellar atmospheric parameters using the k-Nearest Neighbor method by a genetic algorithm. This algorithm generates a vector of binary values for selecting spectral indices and prototypical stars from a dataset. Our experimental results on astronomical data



(a) (b)

Figure 7. Catalog versus predicted MK spectral classes parameter using the original (603 stars and 48 spectral index) (a) and GA data set (243 stars and 11 spectral indices) with 3NN predictive classifier (b).

(a) Luminosity classes (predicted) vs. Luminosity classes (catalog)

	I	II	III	IV	V	VI
I	8	0	0	0	0	0
II	2	5	5	0	0	0
III	0	10	312	12	1	0
IV	0	1	10	38	12	0
V	0	0	3	10	170	4
VI	0	0	0	0	0	0

(b) Luminosity classes (predicted) vs. Luminosity classes (catalog)

	I	II	III	IV	V	VI
I	3	0	0	0	0	0
II	3	4	2	0	0	0
III	2	12	304	16	3	0
IV	0	0	16	28	20	1
V	1	1	6	18	157	3
VI	0	0	0	0	3	0

Figure 8. Confusion matrices for luminosity classes parameter using the original (603 stars and 48 spectral index) (a) and GA data set (284 stars and 11 spectral indices) with 3NN predictive classifier (b).

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

1. The GA reduces the size of the data set so that KNN can classify faster.

2. The GA increases the predictive accuracy, due to the elimination of noisy or irrelevant attributes.

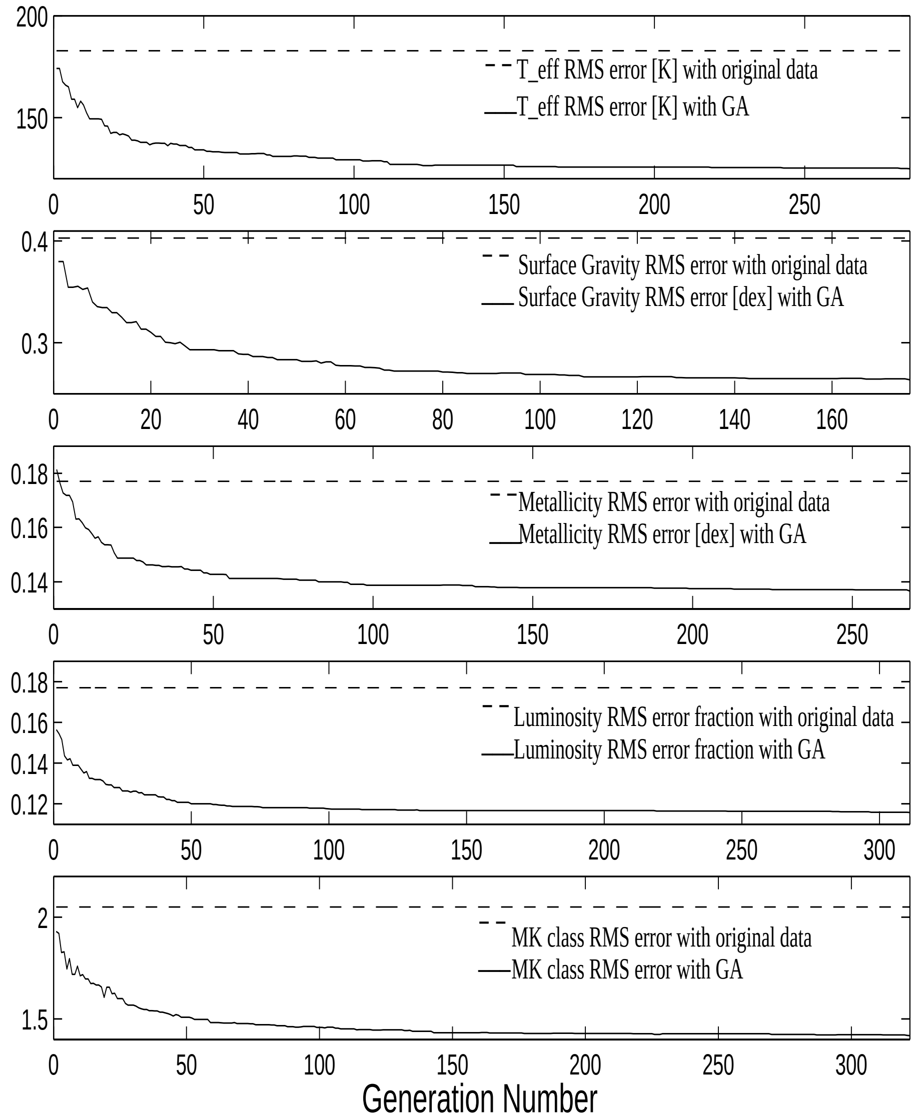


Figure 9. RMS prediction errors.

3. The GA identifies relevant spectral indices, so that the data are easier to understand, which may be useful for other applications.

Future work will attempt to combine evolutionary algorithms with other machine learning algorithms, such as neural networks or decision trees to increase the predictive accuracy. Also, we intend to apply this method to other astronomical problems, including spectral and mor-

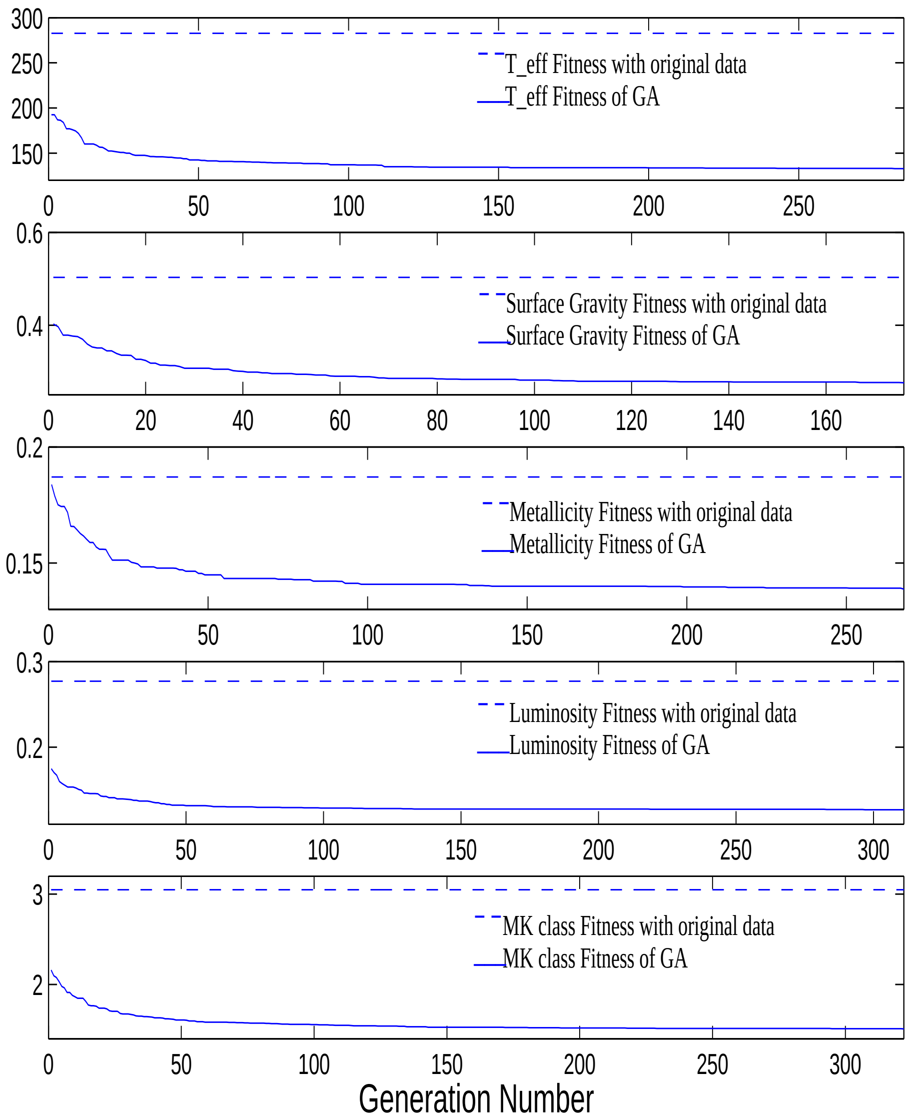


Figure 10. Behavior of GA fitness

phological galaxy classification and determination of radial velocity in stars, and redshift in galaxies.

References

Adams, A. and A. Woolley: 1994, 'Hubble Classification of Galaxies Using Neural Networks'. *Vistas in Astronomy* **38**, 273–280.

	Original Dataset			Dataset provided by G.A.				
	Stars	Spectral Indices	Prediction Error	Stars	Spectral Indices	Prediction Error	Error Reduction (%)	Time Reduction (%)
Efective Temperature	642	48	182.8 K	302	8	124.7 K	32	92
Surface Gravity	642	48	0.403 dex	310	12	0.263 dex	35	87
Metallicity	642	48	0.177 dex	316	20	0.136 dex	23	79
Luminosity Classes	603	48	0.177	284	11	0.116	34	89
MK- spectral Classes	614	48	2.05	243	11	1.41	31	90

Figure 11. Comparison of results of the classification using 3NN method considering as training data the original data sets from catalog and the subsets provided by the genetic algorithm.

Bailer-Jones, C. A. L., M. Irwin, and T. von Hippel: 1997, 'Physical parametrization of stellar spectra: the neural network approach'. *Monthly Notices of the Royal Astronomical Society* **292**(1), 157–166.

Baker, J. E.: 1985, 'Adaptive selection methods for genetic algorithms'. In: J. Grefenstette (ed.): *Proc. of an International Conference on Genetic Algorithms and their Applications.* pp. 101–111, Lawrence Erlbaum Associates.

Brindle, A.: 1981, 'Genetic algorithms for function optimization'. Ph.D. thesis, University of Alberta, Department of Computer Science.

Cayrel de Strobel, G., B. Hauck, P. Franois, F. Thvenin, E. Friel, M. Mermilliod, and S. Borde: 1992, 'A catalogue of [Fe/H] determinations: 1991 edition'. *Astron. Astrophys., Suppl. Ser.* **95**(2), 273–336.

Charbonneau, P.: 1995, 'Genetic Algorithms in Astronomy and Astrophysics'. *Astrophys. Journal Suppl. Ser.* **101**, 309–334.

Fix, E. and J. L. Hodges: 1951, 'Discriminatory analysis—nonparametric discrimination: Consistency properties'. Technical Report 21-49-004, USAF School of Aviation Medicine, Randolph Field, Tex.

Golberg, D.: 1989, *Genetic algorithms in search, optimization and machine learning.* Addison-Wesley Publishing Company.

Goldberg, D. and K. Deb: 1991, 'A comparative analysis of selection schemes used in genetic algorithms'. In: G. Rawlins (ed.): *Foundations of Genetic Algorithms.* Berlin: Morgan Kaufmann, pp. 69–93.

Gorgas, J., N. Cardiel, and S. Pedraz: 1998, 'Towards an Understanding of the $\lambda 4000\AA$ Break Behavior in Old Stellar Populations'. *Astrophysics and Space Science* **263**(1), 167–170.

Gulati, R. K., R. Gupta, P. Gothoskar, and S. Khobragade: 1994, 'Ultraviolet Stellar Spectral Classification Using Multilevel Tree Neural Network'. *Vistas in Astronomy* **38**, 293–299.

Hoffleit, D. and Jaschek: 1991, 'Bright Star Catalog'. Technical report, Yale University Observatory, New Haven. 5th rev.

Holland, J.: 1975, *Adaptation in Natural and Artificial Systems*. MIT Press.

Jones, L. A.: 1996. Ph.D. thesis, University of North Carolina, Chapel Hill, North Carolina.

Jones, L. A. and G. Worthey: 1995, 'New age indicators for old stellar populations'. *Astrophys. J., Lett.* **446**(1), L31–L34.

Jong, K. A. D.: 1975, 'An analysis of the behaviour of a class of genetic adaptive systems'. Ph.D. thesis, University of Michigan. Diss. Abstr. Int. 36(10), 5140B, University Microfilms No. 76–9381.

Naim, A., O. Lahav, L. S. Jr., and M. C. Storrie-Lombardi: 1995, 'Spectral Classification with Principal Component Analysis and Artificial Neural Networks'. *Monthly Notices of the Royal Astronomical Society*.

Odewahn, S. C. and M. L. Nielsen: 1994, 'Star-Galaxy Separation using Neural Networks'. *Vistas in Astronomy* **38**, 281–285.

Quinlan, J.: 1986, 'Induction of Decision Trees'. *Machine Learning* **1**, 81–106.

Rose, J. A.: 1994, 'The integrated spectra of M32 and of 47 Tuc: a comparative study at high spectral resolution'. *Astron. J.* **107**(1), 206–229.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams: 1986, 'Learning Internal Representations by Error Propagation'. In: Rumelhart and McClelland (eds.): *Parallel Distributed Processing, Explorations in the Microstructure of Cognition*, Vol. 1. Cambridge, MA: MIT Press, pp. 318–362.

Sodré, L. and H. Cuevas: 1994, 'Spectral Classification of Galaxies'. *Vistas in Astronomy* **38**, 286–291.

Storrie-Lombardi, M. C., M. J. Irwin, T. von Hippel, and L. J. Storrie-Lombardi: 1994, 'Spectral Classification with Principal Component Analysis and Artificial Neural Networks'. *Vistas in Astronomy* **38**, 331–340.

Storrie-Lombardi, M. C., O. Lahav, L. Sodré, and L. J. Storrie-Lombardi: 1992, 'Morphological Classification of Galaxies by Artificial Neural Networks'. *Monthly Notices of the Royal Astronomical Society* **259**, 8–12.

Szalay, A. S., P. Z. Kunszt, A. Thakar, J. Gray, D. Slutz, and R. J. Brunner: 2000, 'Designing and mining multi-terabyte astronomy archives: the Sloan Digital Sky Survey'. In: *Proc. of SIGMOD*. pp. 451–462.

Whitley, D.: 1989, 'The GENITOR algorithm and selection pressure: why rank-based allocation of reproductive trials is best'. In: J. Schaffer (ed.): *Proceedings of the Third International Conference on Genetic Algorithms*. pp. 116–121, Morgan Kaufmann.

Worthey, G., S. M. Faber, J. J. Gonzlez, and D. Burstein: 1994, 'Old stellar populations. V. Absorption feature indices for the complete Lick/IDS sample of stars'. *Astrophys. J., Suppl. Ser* **94**(2), 687–722.