

Neural Networks and Instance-Based Learning for the Prediction of Stellar Atmospheric Parameters

Olac Fuentes
Instituto Nacional de Astrofísica, Óptica y Electrónica
Luis Enrique Erro # 1
Santa María Tonantzintla, Puebla, 72840, México
fuentes@inaoep.mx

ABSTRACT

In this paper we show how machine learning methods can be effectively applied to the problem of automatically predicting stellar atmospheric parameters from spectral information, a very important problem in stellar astronomy. We apply feedforward neural networks, Kohonen's self-organizing maps and locally-weighted regression to predict the stellar atmospheric parameters effective temperature, surface gravity and metallicity from spectral indices. Our experimental results show that the three methods are capable of predicting the parameters with very good accuracy. Locally weighted regression gives slightly better results than the other methods using the original dataset as input, while self-organizing maps outperform the other methods when significant amounts of noise are added. We also implemented a heterogeneous ensemble of predictors, combining the results given by the three algorithms. This ensemble yields better results than any of the three algorithms alone, using both the original and the noisy data.

KEY WORDS

machine learning, neural networks, instance-based learning, ensembles

1. Introduction

In astronomy and other scientific disciplines, we are currently facing a massive data overflow. With the development of new sophisticated detectors that cover virtually all of the electromagnetic spectrum, and with the rapid development of the Internet, astronomers have at their disposal amounts of information that are vastly larger than what they can analyze by traditional manual or semimanual means.

In recent years astronomers and artificial intelligence researchers have started collaborating towards the goal of automating the task of analyzing astronomical data. By far the most commonly used methodology has been feedforward neural networks, generally applying the backpropagation algorithm for training. In galactic astronomy they have been applied to morphological classification of galaxies [18], star/galaxy separation [16] and spectral classification of galaxies [7, 9]. In stellar astronomy, they have been applied to classification of stars [17, 11, 19] and to physical parameterization of spectra [3].

While artificial neural networks 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 is that the time required to train a network can be very long. Also, since they perform local search in parameter space, they often converge to local minima. A final drawback is that once trained, artificial neural networks are black boxes, in the sense that it is very difficult for humans to interpret the rules learned by the network.

In this paper we compare the performance of feedforward neural networks, the most commonly used method in automated astronomical prediction and classification, to two other methods, Kohonen self-organizing maps [14] and locally-weighted regression [5, 2]. We also implemented a heterogeneous ensemble of predictors, combining the results given by the three algorithms.

The organization of the remainder of this paper is as follows: Section 2 describes the problem of predicting stellar atmospheric parameters from spectral features and the data used in our experiments, Section 3 describes the learning algorithms we applied, Section 4 presents experimental results and Section 5 presents conclusions and directions for future research.

2. Prediction of Stellar Atmospheric Parameters

A stellar spectrum is a plot of flux against wavelength as measured by a spectrograph. Figure 1 shows three sample spectra from the dataset we are using. As can be seen in the plots, the spectra present a series of discontinuities, called absorption lines, which are caused by the presence of certain atoms, molecules and ions in the star's atmosphere that absorb energy only at specific wavelengths. For example, the absorption lines at around 3890, 3930, 3970, 4100, and 4340 angstrom are known to be produced by ionized hydrogen in the star's atmosphere. From the strength of these absorption lines, an expert astronomer can estimate with good accuracy several of the most important properties of the star's atmosphere, including its temperature, surface gravity and metal content. The goal of this research is to develop automated methods for predicting the properties

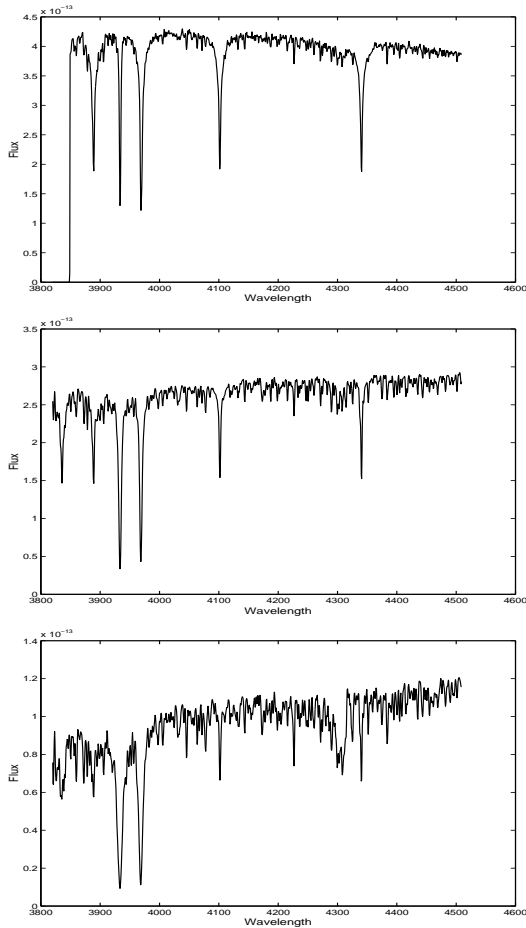


Figure 1. Sample stellar spectra

of a stellar atmosphere given the star’s spectrum with an accuracy comparable to that of a human expert.

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 absorption 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 [13]. This dataset consists of 24 spectral indices for 684 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 author at an anonymous ftp site at the National Optical Astronomy Observatories (NOAO).

3. The Methods

3.1 Feedforward Neural Networks and Self-Organizing Maps

It is beyond the scope of this paper to give a detailed description of feedforward neural networks and self-

organizing maps. Instead, we will briefly describe how they were adapted to our task. For an introduction to neural networks we refer the reader to any of the excellent books that have been published in recent years, such as [12] and [4].

We applied standard feedforward neural networks without any major modification. We used the backpropagation algorithm for training and the tan-sigmoid function as a transfer function. The self-organizing map algorithm was modified to perform in a supervised fashion by combining into each training example and prototype associated with each unit the input and output parameters. That is, each training example (and prototype) is represented by a vector of length 27, where the first 24 elements correspond to the 24 spectral indices and the final 3 elements correspond to the output parameters (the stellar atmospheric parameters). After training, we extract the prototypes in each unit and build a table, where each entry maps the input parameters to the output parameters contained in the same prototype. We then use this table to predict the output parameters of a given star by finding its 3-nearest neighbors in the input space and assigning to it the output parameters given by the weighted average of the output parameters of its neighbors, where the weight is inversely proportional to the Euclidean distance in the input space between that star and its neighbors.

3.2 Locally-Weighted Regression

Locally-weighted regression belongs to the family of instance-based learning algorithms. In contrast to most other learning algorithms, which use their training examples to construct explicit global representations of the target function, instance-based learning algorithms simply store some or all of the training examples and postpone any generalization effort until a new instance must be classified. They can thus build query-specific local models, which attempt to fit the training examples only in a region around the query point. In this work we use a linear model around the query point to approximate the target function.

Given a query point \mathbf{x}_q , to predict its output parameters \mathbf{y}_q , we assign to each example in the training set a weight given by the inverse of the distance from the training point to the query point: $w_i = \frac{1}{|\mathbf{x}_q - \mathbf{x}_i|}$

Let W , the weight matrix, be a diagonal matrix with entries w_1, \dots, w_n . Let X be a matrix whose rows are the vectors $\mathbf{x}_1, \dots, \mathbf{x}_n$, the input parameters of the examples in the training set, with the addition of a “1” in the last column. Let Y be a matrix whose rows are the vectors $\mathbf{y}_1, \dots, \mathbf{y}_n$, the output parameters of the examples in the training set. Then the weighted training data are given by $Z = WX$ and the weighted target function is $V = WY$. Then we use the estimator for the target function $\mathbf{y}_q = \mathbf{x}_q^T (Z^T Z)^{-1} Z^T V$.

3.3 Ensemble Methods

An ensemble of classifiers is a set of classifiers whose individual decisions are combined in some way, normally by voting. Somewhat surprisingly, ensembles often yield better results than the best of the individual classifiers that make them up. Ensembles work only if the individual classifiers are more accurate than random guessing and their errors are not strongly correlated. Several methods have been proposed for constructing ensembles, including bagging [6], boosting [10], error-correcting output-coding [8] and randomization [15, 1]. Normally, ensembles are homogeneous in the sense that the same learning algorithm is implemented by each member of the ensemble, and they are forced to produce non-correlated results by providing slightly different training sets for each member, as in boosting and bagging, by manipulating the outputs, as in ECOC, or by slightly modifying some parameters of the algorithms, as in randomization. Here we propose a heterogeneous ensemble that combines, by means of unweighted average, the results given by the three learning algorithms mentioned previously. It will be shown that this heterogeneous ensemble is more accurate than any of the three algorithms taken individually.

4. Experimental Results

We implemented locally-weighted regression and self-organizing maps in MatlabTM, and used the feedforward neural network implementation that comes with the MatlabTM Neural Network Toolbox. For all the experiments reported we used five-fold cross-validation, that is, we partitioned the original dataset randomly into five equally-sized subsets and performed five experiments, using in each experiment one of the subsets for testing and the other four for training.

Before performing the experiments the indices in the dataset were standardized so as to have a zero mean and unit variance across the dataset. This eliminates the potential problem of the dependence of the results on the particular units in which different indices were measured, but it also implicitly assumes that all the indices are equally important for each prediction task.

Figure 2 plots the catalog versus predicted parameters after applying locally weighted regression to the prediction of the parameters. To build the linear model around the query point we choose the 200 closest points and weighted each point by a factor inversely proportional to its Euclidean distance to the query point. Figure 3 shows the results of applying a self-organizing map. We used a 15 by 15 two-dimensional grid, with a total of 225 units and we trained it for 25 epochs. Figure 4 shows the results obtained applying a feedforward neural network to the prediction of the parameters. The network had a configuration of 24 input units, 8 hidden units and one output unit for each parameter. We trained it for 200 epochs using the backpropagation algorithm. Figure 5 shows the results of applying

Algorithm	Teff [K]	Log g [dex]	[Fe/H] [dex]
LWR	101.25	0.209	0.111
SOM	135.08	0.240	0.134
FFNN	127.28	0.206	0.123
Ensemble	97.80	0.192	0.107

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

Algorithm	Teff [K]	Log g [dex]	[Fe/H] [dex]
LWR	279.81	0.548	0.259
SOM	257.62	0.497	0.276
FFNN	303.67	0.519	0.315
Ensemble	252.56	0.468	0.259

Table 2. Mean absolute errors in the prediction of stellar atmospheric parameters using the three learning algorithms and adding Gaussian noise with zero mean and $\sigma = 1$ to the original data.

the heterogeneous ensemble method, which combines the results of the three other methods by means of a simple (unweighted) average. It can be seen that for all four methods, temperature can be predicted with the highest degree of accuracy, while metallicity is harder and surface gravity is still more difficult. This is consistent with results attained by human experts. Table 1 shows the mean absolute errors for each of the methods and each of the parameters. This was obtained by averaging the results of 3 runs of 5-fold cross-validation for each method. It can be seen that LWR gives the best performance among the individual methods, and that the ensemble method gives the best overall results.

For a second experiment we added noise to the original data to test the noise-sensitivity of each of the methods. Table 2 shows the results obtained after adding Gaussian noise with zero mean and unit standard deviation to each of the attributes of the original data. All methods suffer a degradation in performance, as expected, but surprisingly LWR is not noticeably more affected by noise than the neural network methods. This may be due to the relative large number of neighboring points we choose to build the local approximation, and the fact that the noise, being white, tended to cancel across the neighboring points. Both LWR and SOM still perform slightly better than FFNN, and the ensemble method still yields better results than any of the individual methods.

5. Conclusions and Future Work

We have performed an experimental comparison of locally weighted regression, self-organizing maps and feedforward neural networks, three well-known machine learning methods, applied to the problem of predicting stellar atmospheric parameters from spectral indices. Our experimental results show that all three methods are capable of yielding

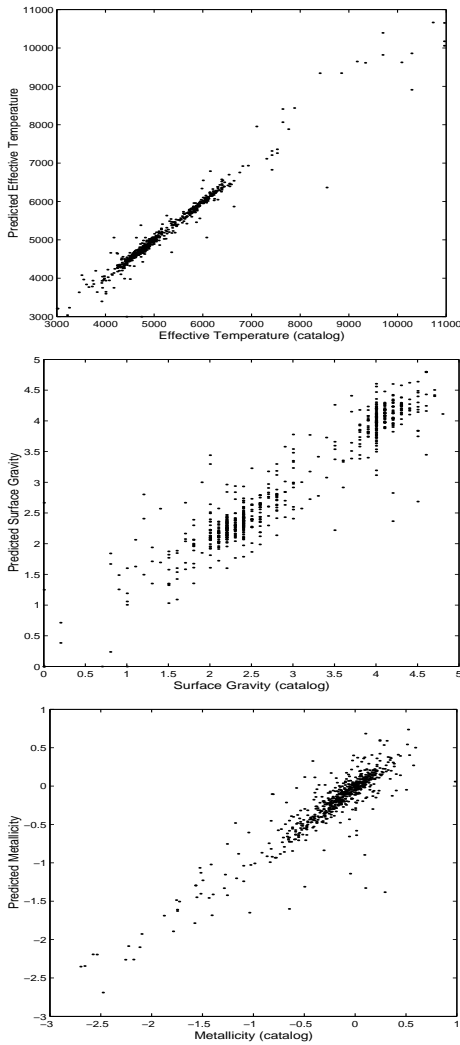


Figure 2. Catalog versus predicted parameters using LWR with the original dataset.

predictions that are as accurate as those made by human experts. While the three methods provide very good results, locally weighted regression provides slightly better accuracy than the other methods. We also introduced a fourth method, a heterogeneous ensemble that combines the results of the three individual methods by means of an unweighted average. This ensemble performs better than any of the individual methods, both with the original data and when a significant amount of Gaussian noise is added to the data.

Future work includes extending the experiments to other datasets to test the generality of the conclusions reached here and work on developing methods for finding the spectral indices that are the most useful for predicting each of the stellar atmospheric parameters.

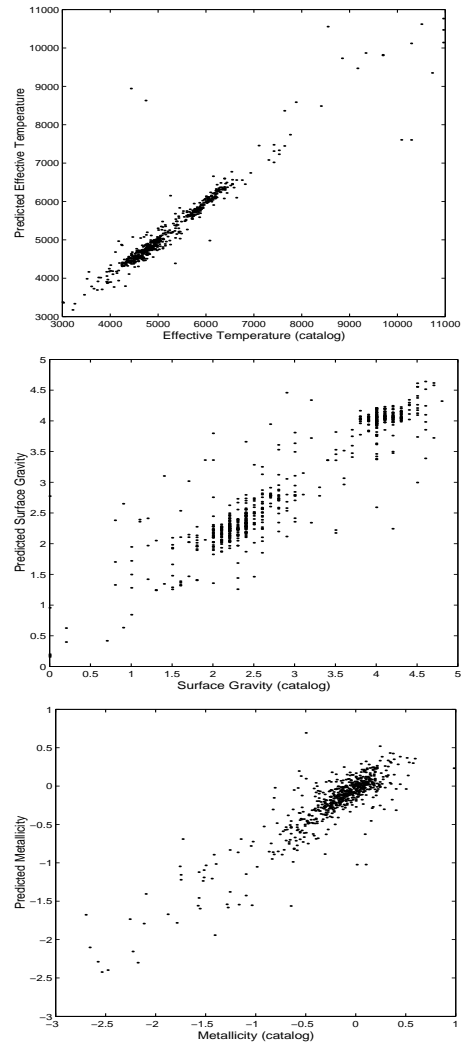


Figure 3. Catalog versus predicted parameters using a SOM with the original dataset.

References

- [1] K. Ali and M. Pazzani. Bagging predictors. *Machine Learning*, 24(3):173–202, 1996.
- [2] C. G. Atkeson, A. W. Moore, and S. Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11:11–73, 1997.
- [3] C. A. L. Bailer-Jones, M. Irwin, and T. von Hippel. Physical parameterization of stellar spectra: The neural network approach. *Monthly Notices of the Royal Astronomical Society*, 292(157), 1997.
- [4] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, England, 1996.
- [5] L. Bouttou and V. Vapnik. Local learning algorithms. *Neural Computation*, 4(6):888–900, 1992.

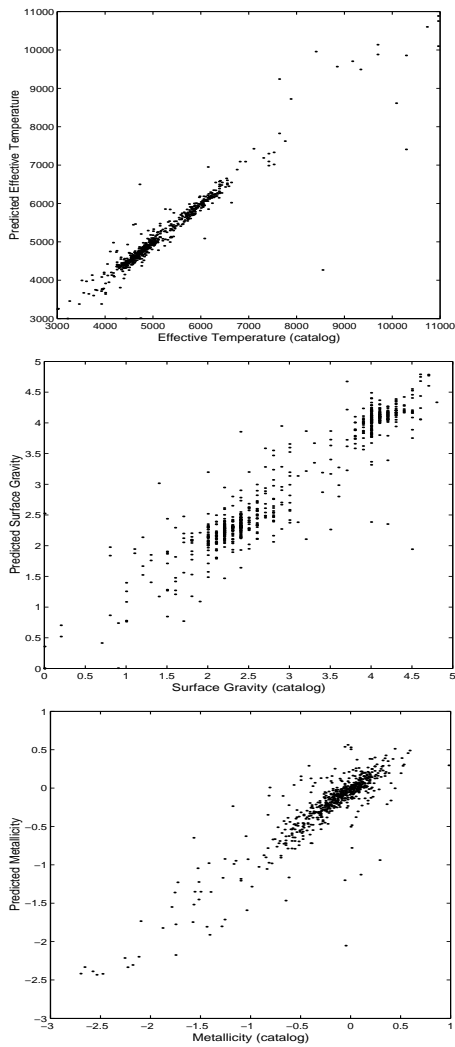


Figure 4. Catalog versus predicted parameters using with a FFNN with the original dataset.

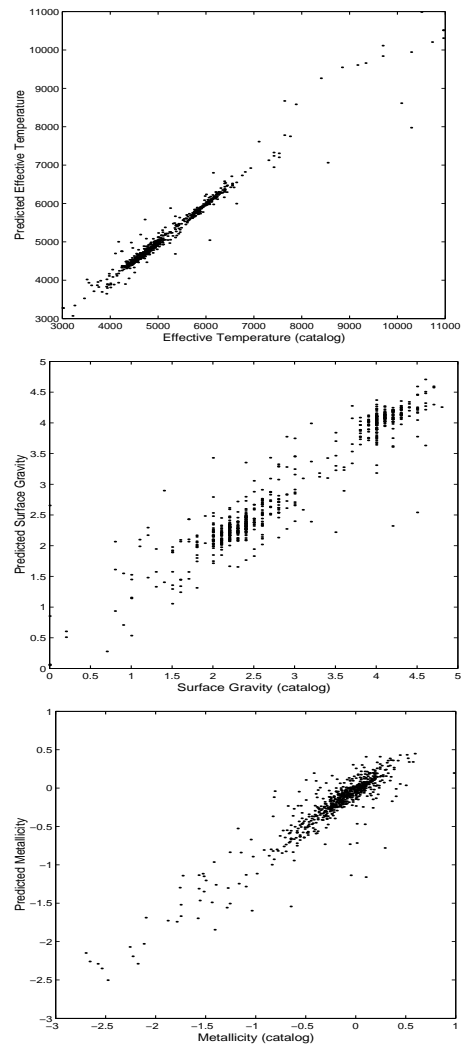


Figure 5. Catalog versus predicted parameters using the heterogeneous ensemble.

[6] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[7] A. J. Connolly, A. S. Szalay, M. A. Bershad, A. L. Kinney, and D. Calzetti. Spectral classification of galaxies: an orthogonal approach. *The Astronomical Journal*, 110(3):1071–1082, 1995.

[8] T. Dietterich and G. Bakiri. Solving multi-class learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.

[9] S. R. Folkes, O. Lahav, and S. J. Maddox. An artificial neural network approach to classification of galaxy spectra. *Monthly Notices of the Royal Astronomical Society*, 283:651–665, 1996.

[10] Y. Freund and R. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the Thir-*

teenth International Conference on Machine Learning, pages 148–156. Morgan Kaufmann, 1996.

[11] R. K. Gulati, R. Gupta, P. Gothoskar, and S. Khobragade. Stellar spectral classification using automated schemes. *The Astrophysical Journal*, 426:340–344, 1994.

[12] R. Hecht-Nielsen. *Neurocomputing*. Addison-Wesley, Reading, MA, 1990.

[13] L. A. Jones. PhD thesis, University of North Carolina, Chapel Hill, North Carolina, 1996.

[14] T. Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, Berlin, 1988. 2nd Edition.

[15] S. W. Kwok and C. Carter. Multiple decision trees. In R. Schachter, T. S. Levitt, L. N. Kamal, and J. F. Lemmer, editors, *Uncertainty in Artificial Intelligence 4*, pages 327–335. Elsevier Science, Amsterdam, 1990.

- [16] S. C. Odewahn and M. L. Nielsen. Star-galaxy separation using neural networks. *Vistas in Astronomy*, 38:281–285, 1994.
- [17] 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.
- [18] M. C. Storrie-Lombardi, O. Lahav, L. Sodre, and L. J. Storrie-Lombardi. Morphological classification of galaxies by artificial neural networks. *Monthly Notices of the Royal Astronomical Society*, 259:8–12, 1992.
- [19] W. B. Weaver and A. V. Torres-Dodgen. Neural network classification of the near-infrared spectra of A-type stars. *The Astrophysical Journal*, 446:300–317, 1995.