Chin. J. Astron. Astrophys. Vol. 5 (2005), No. 2, 203–210
(*http://www.chjaa.org*)

Chinese Journal of
Astronomy and
Astrophysics

Automated Stellar Classification for Large Surveys with EKF and RBF Neural Networks *

Ling Bai[1,2], Ping Guo[1] and Zhan-Yi Hu[2]

[1] Department of Computer Science, Beijing Normal University, Beijing 100875
pguo@bnu.edu.cn
[2] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100080

Received 2004 July 14; accepted 2004 December 21

Abstract An automated classification technique for large size stellar surveys is proposed. It uses the extended Kalman filter as a feature selector and pre-classifier of the data, and the radial basis function neural networks for the classification. Experiments with real data have shown that the correct classification rate can reach as high as 93%, which is quite satisfactory. When different system models are selected for the extended Kalman filter, the classification results are relatively stable. It is shown that for this particular case the result using extended Kalman filter is better than using principal component analysis.

Key words: methods: data analysis — techniques: spectroscopic — stars: general — galaxies: stellar content

1 INTRODUCTION

Stellar spectroscopy is one of the most useful techniques in the study of physical conditions (temperature, pressure, density, etc.) and chemical abundance of stars. At its most general level, the objective of classification is to identify similarities and differences between objects, with the goal of efficiently reducing the number of types of objects one has to deal with.

The huge sky survey telescope now being built in Beijing, named LAMOST, is expected to collect more than 1×10^7 spectra of faint celestial objects. Scientific exploitation of these huge data will require powerful, robust, and automated classification tools. As we know, neural networks have already had many applications in astronomical data classification (Bishop et al. 1994). However, the correct classification rate (CCR) is very low if the spectral data are not

∗ Supported by the National Natural Science Foundation of China (Project No. 60275002) and the National High Technology Research and Development Program of China (863 Program, Project No. 2003AA133060).

pre-processed. Principle Component Analysis (PCA) (Jolliffe 1986) is traditionally used for feature extraction, but as we know, PCA is only efficient when the raw data can be separated linearly. Therefore, we should develop new methods to deal with the issue of pre-processing.

The Kalman filter addresses the general problem of estimating the state $\mathbf{x} \in \Re^n$ of a discrete-time controlled process that is governed by a linear stochastic difference equation (Welch 2004). When the process to be estimated and the relationship between the measurement and the process are non-linear, we consider a Kalman filter that linearizes about the current mean and covariance, known as the extended Kalman filter (EKF). For many systems, EKF has proven to be a useful method of obtaining good estimates of the system state. In this paper, EKF is applied as a pre-processor to stellar spectra recognition.

A radial basis function (RBF) network is trained to perform a mapping from an m-dimensional space to an n-dimensional output space (Todorovic et al. 2002). In our study, recognition is implemented through building a nonlinear relationship between feature spectra and classes.

The MK (Kurtz 1984) classification of stellar spectra classifies stars into seven main spectral types in the decreasing order of temperature, namely: O, B, A, F, G, K and M. Stellar spectra of the seven main types are shown in Fig. 1.

The organization of the present paper is as follows: In Sect. 2, we introduce the data set used in our study; in Sect. 3, the methods of EKF and RBF are described; the experimental results are reported in Sect. 4, together with a discussion; finally concluding remarks are presented in Sect. 5.

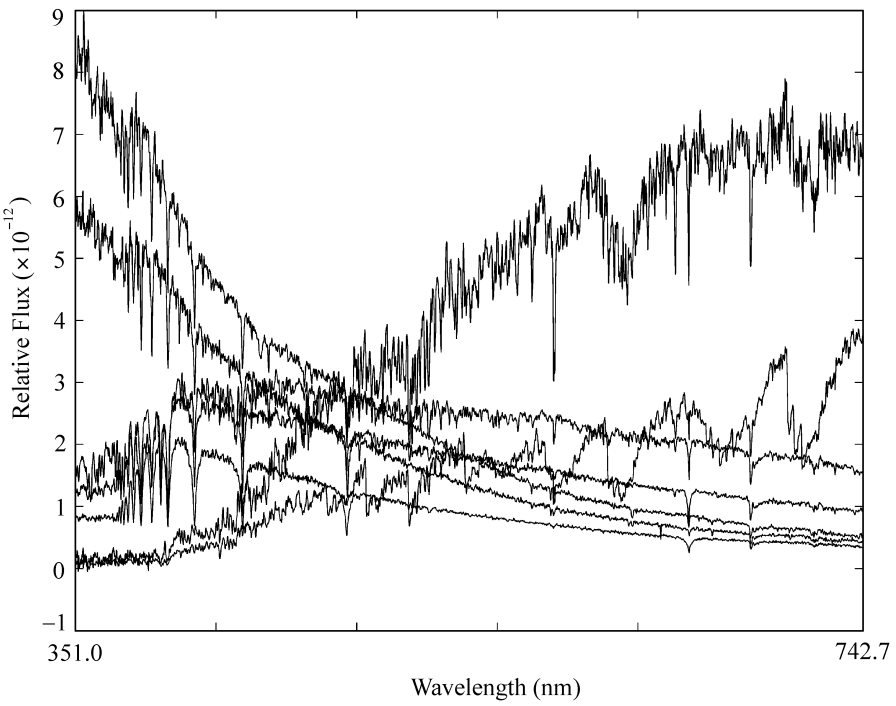

Fig. 1 Seven main types of stellar spectra.

2 THE DATA SET

The stellar spectra used in our experiments are selected from Astronomical Data Center (ADC). These include 161 stellar spectra contributed by Jacoby et al. (1984) and 96 observed by Pickles et al. (1985). The spectra taken from the above libraries have resolutions of 0.14 nm and 0.5 nm, respectively. In order to analyze them on the same scale, all the spectra are digitized and linearly interpolated to the wavelength range of 360–742 nm at steps of 0.5 nm.

The raw data set was selected from two different libraries, it is normalized before classification (Bishop 1995) as:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i,$$

$$\gamma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2, \tag{1}$$

where n is the sample size. The normalized variables are:

$$\ddot{x}_i = \frac{x_i - \bar{x}}{\gamma}. \tag{2}$$

3 BACKGROUND

3.1 Extended Kalman Filter

Here, we give a brief review on the essentials of EKF. As stated above, EKF is an extension of the standard Kalman filter. Assume that the process has a state vector $\mathbf{x} \in \Re^n$, and is governed by the non-linear stochastic difference equation:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_{k-1}), \tag{3}$$

with a measurement $\mathbf{y} \in \Re^l$ that is

$$\mathbf{y}_k = h(\mathbf{x}_k, \mathbf{v}_k), \tag{4}$$

where the random variables $\mathbf{w}_k$ and $\mathbf{v}_k$ are the noise in the process and measurement, respectively.

In practice, we do not know the individual values of the noise $\mathbf{w}_k$ and $\mathbf{v}_k$ at each time step. However, we can approximate the state and measurement vector without them. To estimate a process with non-linear difference and measurement relationships, we begin by writing down new governing equations by linearization of Eqs. (3) and (4),

$$\mathbf{x}_k \approx \tilde{\mathbf{x}}_k + \mathbf{A}(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}) + \mathbf{W}\mathbf{w}_{k-1}, \tag{5}$$

$$\mathbf{y}_k \approx \tilde{\mathbf{y}}_k + \mathbf{H}(\mathbf{x}_k - \tilde{\mathbf{x}}_k) + \mathbf{V}\mathbf{v}_k, \tag{6}$$

where

- $\mathbf{x}_k$ and $\mathbf{y}_k$ are the actual state and measurement vectors.
- $\tilde{\mathbf{x}}_k$ and $\tilde{\mathbf{y}}_k$ are the approximate state and measurement vectors.
- $\hat{\mathbf{x}}_k$ is a posteriori estimate of the state at step k.
- the random variables $\mathbf{w}_k$ and $\mathbf{v}_k$ represent the process and measurement noise.
- $\mathbf{A}$, $\mathbf{W}$, $\mathbf{H}$ and $\mathbf{V}$ are the Jacobian matrices.

Note that the matrix $\mathbf{A}$ is the Jacobian matrix of partial derivatives of f which f is chosen to be a smooth, differential function. So in this case $\mathbf{A}$ can always be calculated. So are the matrices $\mathbf{W}$, $\mathbf{H}$ and $\mathbf{V}$. The complete set of EKF equations are shown in Eqs. (7) and (8), the former is the EKF time update equation, and the latter is the measurement update equation.

$$\begin{aligned} \hat{\mathbf{x}}_k^- &= f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k, 0)\,, \\ \mathbf{P}_k^- &= \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^T + \mathbf{W}_k \mathbf{Q}_{k-1} \mathbf{W}_k^T, \end{aligned} \tag{7}$$

$$\begin{aligned} \mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{V}_k \mathbf{R}_k \mathbf{V}_k^T)^{-1}\,, \\ \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - h(\hat{\mathbf{x}}_k^-, 0))\,, \\ \mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-. \end{aligned} \tag{8}$$

As with the basic discrete Kalman filter, the time update equations in Eq. (7) project the state and covariance estimates from the previous time step $k-1$ to the current time step k, $\mathbf{Q}_k$ is the process noise covariance. The measurement update equations in Eq. (8) correct the state and covariance estimates with the measurement $\mathbf{y}_k$, $\mathbf{R}_k$ is the measurement noise covariance. $\mathbf{K}$ is known as the EKF gain and $\mathbf{P}$ is the estimate error covariance.

Note that in Eq. (8), we have an inverse operation. It is occasionally possible that the square matrix is singular. In that case we use a pseudoinverse operation instead of the inverse operation, and this will not affect our experiment results.

3.2 RBF Neural Network

An RBF neural network is composed of three layers: the input layer, the hidden layer and the output layer. Suppose there are c neurons in the hidden layer and each of the c neurons in the hidden layer applies an activation function. The outputs of the network are sums of weighted hidden layer neurons (Simon et al. 2002).

When the input spectra have been projected to the feature space, an RBF network classifier is invoked for the final classification. The RBF network is built by considering the basis function as a neural activation function and the $\mathbf{W}$ parameters as weights which connect the hidden layer and the output layer.

The k-th component of the RBF neural network output looks like (Bishop 1995):

$$\begin{aligned} \mathbf{z}_k(\mathbf{x}) &= \sum_{j=1}^{C} w_{kj} \varphi_j(\|\mathbf{x} - \mu_j\|) + w_{k0} \\ &= \sum_{j=0}^{C} w_{kj} \varphi_j(\|\mathbf{x} - \mu_j\|)\,, \\ \mathbf{z}_k(\mathbf{x}) &= \mathbf{W}_k \varphi(\mathbf{x}), \end{aligned} \tag{9}$$

where $\mathbf{x}$ is an l-dimensional input vector, $\mathbf{w}_{k0}$ is a set of bias constants, $\varphi_0(\|\mathbf{x} - \mu_j\|) \equiv 1$ and the radial basis functions are chosen to be Gaussians.

$$\varphi_j(\|\mathbf{x} - \mu_j\|) = \exp\Big[-\frac{1}{2\sigma_j^2} \|\mathbf{x} - \mu_j\|^2 \Big]. \tag{10}$$

When training samples $\{\mathbf{x}_i, t_i\}_{i=1}^N$ are given, the weight matrix $\mathbf{W}$ can be initialized as $\mathbf{W} = \mathbf{T}^T \Phi^{-1}$, where $\mathbf{T}$ is the corresponding target matrix and Φ is a matrix about the Gaussian function: $\varphi(\|\mathbf{x} - \mu\|)$.

The network weights are found step by step. Considering $q(\mathbf{W})$ the sum-of-squares error function, the gradient vector $\nabla q = \{\frac{\partial q(\mathbf{W})}{\partial w}\}$ shows the direction of maximum of square mean error. In the discrete time k approximation, at step $k+1$, given the weights at step k, the weights are adjusted to be:

$$\mathbf{W}_{k+1} = \mathbf{W}_k - \eta \nabla q,$$

where $0 \leq \eta < 1$, is named the learning constant and used for tuning the speed and quality of the learning process. In our experiments, η is empirically selected as 0.2. For the weight matrix $\mathbf{W}$ being initialized as $\mathbf{W} = \mathbf{T}^T \Phi^{-1}$, the learning constant η can not be too large, otherwise the RBF neural network could not converge.

The learning process of the RBF neural network is composed of two steps. In the first step, the center μ_j and σ_j of each node are calculated based on all the training samples. In the second step, after selecting the parameters of the hidden layer, the weight matrix $\mathbf{W}$, which connect the hidden and output layers, is estimated.

4 EXPERIMENTS AND DISCUSSION

In our experiments, the data set for the classifier includes 257 stellar spectra as described in Sect. 2. In the experiments the leave-one-out cross validation technique is used to select learning samples. Here $n_j = 16$ training samples are randomly drawn from each class, 15 for training and one for testing. Finally, the average correct classification rate (CCR) of RBF classifier is reported.

First, the data set is pre-processed as described in Sect. 2, using Eqs.(1) and(2), so that the data set is normalized. Secondly, EKF in Eqs.(7) and(8) are used for de-noising and feature extraction. Here $\hat{\mathbf{x}}^-$ stands for the input spectra data, $\hat{\mathbf{x}}$ is its estimate, $\mathbf{y}$ is the target vector.

To calculate iteratively the parameters in the EKF, some assumptions and initial settings are made as follows:

1. The process control parameter u_k is ignored in our experiment.
2. In Eq. (3) f is chosen to be a linear function.
3. The variance of the process and measurement noise $\mathbf{w}_k$, $\mathbf{v}_k$ can reasonably be considered as a very small quantities.
4. In Eq. (4) h is chosen to be a Gaussian function.
5. An identity matrix is chosen as the initial value of $\mathbf{P}$ matrix, and zero is chosen as the initial value of $\hat{\mathbf{x}}_k$.

After iterative computations using Eqs. (7) and (8), we have both the estimate of input $\mathbf{x}$ and matrix $\mathbf{H}$. Figure 4 shows some de-noised results. From the figure we can see that the spectral profiles have become smoother, indicating that EKF can indeed reduce noise to some degree.

The output of EKF can easily be calculated from the estimate matrix $\mathbf{H}$, and it is of seven dimensions. Using the two dimension-showing technique, the two dimension distribution in the feature space of the EKF is shown in Fig. 3a. The figure shows that, when the output of EKF is projected to the feature space, what we call $\mathbf{Y}$ space, some classes are well separated. They will be recognized quite accurately if using the RBF network for the final classification.

Function f in Eq. (3) can also be chosen as a quadratic function. This is of a little complexity, but the result using a quadratic function as a feature selector and pre-classifier is quite similar to that using a linear function. In a way, a quadratic function is slightly better than a linear

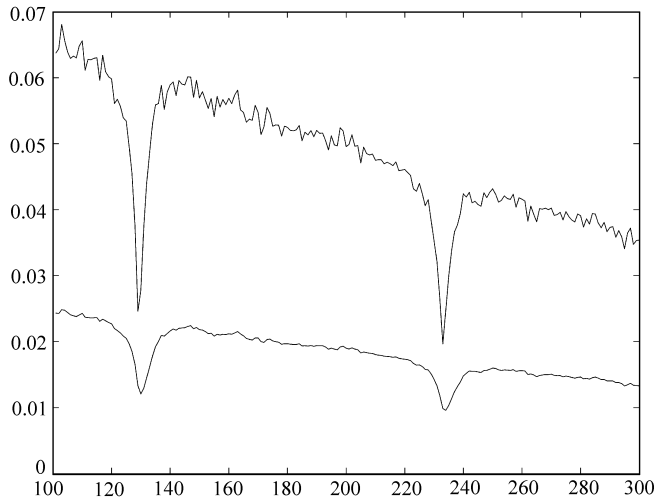

Fig. 2 De-noised result. The upper line is without EKF, bottom line is with EKF.

one. With either a quadratic function or a linear function chosen for f, the CCR can reach as high as 93%.

Then, the feature vectors are used as the input of a RBF network. In the experiments, we design the structure of the RBF network with 35 hidden neurons. For the network output vector $\mathbf{z}$, we use one-of-k encoding method. For the parameter φ_j, it is estimated by the mean value of the 15 training samples, and the parameter σ_j is the corresponding value of φ_j.

In Bai et al. (2003), the Kalman filter is used as the feature selector and pre-classifier. When the structure of RBF is the same, the CCR is 87% when using the Kalman filter, while it can reach 93% when using the EKF. Obviously, EKF is much better than the Kalman filter, being indeed efficient in such non-linear process.

In the experiments described above, we directly use the seven-dimensional feature vectors as the input of an RBF net. As a comparison, we can also use the two-dimensional data shown in Fig. 3a as the input of an RBF net. When the seven-dimensional feature vectors are used, the CCR reaches 93%; When the two-dimensional feature vectors are used, the CCR is 88%. The CCR using two-dimensional data is significantly worsened, but is still a little better than using the Kalman filter. This again demonstrates the superiority of EKF in non-linear process.

The structure of an RBF network is very important for raising the CCR. With different numbers of RBF hidden neurons, the RBF network as a final classifier performs differently in the classification of samples. If a proper neuron number is selected, we can build a better classifier in the process. In this study, we also use a 70-node structure RBF network to classify the stellar data. From experiments we find that the CCR is nearly the same as a 35-node structure RBF network. The RBF network performance becomes noticeably stable when the number of nodes is over 35.

As a comparison, we also use a 7-neuron structure RBF network to classify the data. The center μ_j is estimated by the mean value of 15 samples in each class, and σ_j is the deviation of μ_j. From the experiments we know that the results are much better with 35 neurons than with seven neurons.

Similarly, a 105-neuron structure RBF network is also adopted as a comparison. This is in fact equivalent to exact interpolation. In this case, the classification results are degraded. The RBF net not only shows the characteristic of the feature vectors, but also fits the noise of the feature vectors. The comparative results of different structure RBF network are shown below: for the 7-neuron structure RBF net, the CCR is 82%; for the 35-neuron RBF, 93%; for the 70-neuron RBF, 92%; for the 105-neuron RBF, 78%. These experimental results illustrate that the 35-neuron network structure is a suitable choice.

PCA is a good tool for dimension reduction, data compression and feature extraction. It forms a set of linearly independent basis vectors for describing the data, and can be useful as a classification system by using only the most significant few components. In our study, we can also directly use PCA to extract the principal component for such ill-conditioned problem (Jolliffe 1986).

The distribution of the raw stellar spectra for the first two principal components is shown in Fig. 3b. Some classes are overlapped and can not be separated from each other linearly. Compared with Fig. 3a, we can see that EKF as a data pre-processing technique can separate

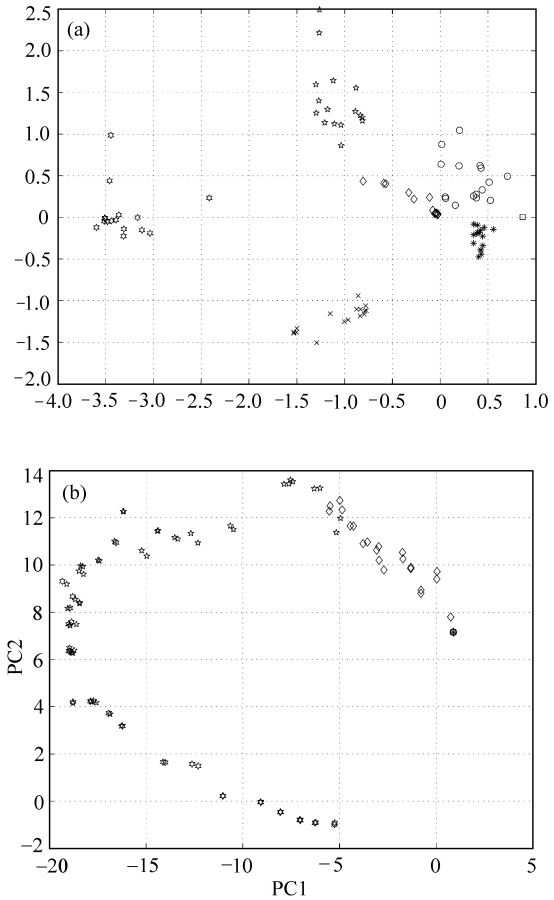

Fig. 3 (a) 2-D distribution of feature data in $\mathbf{Y}$ space; (b) 2-D view of data in the first two principle component space.

classes quite well. If an RBF net is adopted, high CCR can be obtained easily, but PCA is a little worse in this case. PCA is a linear transformation of the data, so any linear classification model which uses the first few principal components is likely to be too simplistic for multiparameter classification. On the other hand, EKF can be made arbitrarily nonlinear, and is convenient for investigating degrees of complexity and nonlinearity (through using different f and h functions). The CCR is only 71% when using PCA as a pre-classifier. It is obvious that using the EKF as a pre-classifier is much better than that using the PCA for this particular case.

Neural networks have many applications in classifications. Bailer-Jones et al. (1998) have constructed a BP neural network to solve the stellar classification problem. However, there are two main drawbacks of it, one is its slow convergence, and the other is its proneness to the local minimum. Xue et al. (2001) have used SOFM for the classification of stellar spectra, and relative good classification results are obtained. In Qin et al. (2003), a combined RBFNN is proposed, and its classification results are quite satisfactory. In this paper, we are mainly concerned with reaching a higher classification accuracy by a combination of EKF and RBF. Our future work will be on the selection and comparison of different neural network models.

5 CONCLUSIONS

Neural networks have already widely been used in the stellar classification, however the non-preprocessed spectra often result in a bad CCR. PCA is a widely used data dimensionality reduction technique, but it is not fit when the process is non-linear.

In our proposed technique, a composite classifier which combines EKF and RBF neural network is applied to stellar spectroscopic classification. First, EKF is employed for de-noising and pre-classifying, then RBF network is used for the final classification. Experiments show that EKF is efficient for this non-linear separating problem. The proposed classifier gives quite good classification results and we believe it is promising in spectrum recognition.

References

Jolliffe I. T., 1986, Principal Component Analysis, New York: Springer-Verlag

Kurtz M. J., 1984, The MK Process and Stellar Classification of Stars, David Dunlap Observatory

Gupta R. A., Gulati R., Gothoskar P. et al., 1994, AJ, 426, 340

Bishop C. M., 1995, Neural Networks for Pattern Recognition, London: Oxford University Press

Simon D., 2002, Neurocomputing, 48, 455

Todorovic B., Stankovic M., Moraga C., 2002, In: J.R. Dorronsoro, ed., ICANN Conf. Vol. 2415, Heidelberg: Springer-Verlag, p.819

Jacoby G. H., Hunter D. A., Christian C. A., 1984, ApJS, 56, 257

Pickles A. J., 1985, ApJS, 59, 33

Bai L., Li Z.B., Guo P., 2003, In: IEEE SMC Conf. Vol. 1, New York: IEEE, p.274

Qin D. M., Guo P., Hu Z. Y. et al., 2003, Chin. J. Astron. Astrophys., 3, 277

Bailer-Jones C., Irwin M., von Hippel T., 1998, MNRAS, 298, 361

Xue J. Q., Li Q. B., Zhao Y. H., 2001, Chin. Astron. Astrophys., 25, 120