Spectral Classification Using Restricted Boltzmann Machine

Chen Fuqiang[1], Wu Yan[1,3], Bu Yude[2] and Zhao Guodong[1]

[1]College of Electronics and Information Engineering, Tongji University, Shanghai 201804, China
[2]School of Mathematics and Statistics, Shandong University, Weihai 264209, China
[3]Email: yanwu@tongji.edu.cn

Abstract

In this study, a novel machine learning algorithm, restricted Boltzmann machine, is introduced. The algorithm is applied for the spectral classification in astronomy. Restricted Boltzmann machine is a bipartite generative graphical model with two separate layers (one visible layer and one hidden layer), which can extract higher level features to represent the original data. Despite generative, restricted Boltzmann machine can be used for classification when modified with a free energy and a soft-max function. Before spectral classification, the original data are binarised according to some rule. Then, we resort to the binary restricted Boltzmann machine to classify cataclysmic variables and non-cataclysmic variables (one half of all the given data for training and the other half for testing). The experiment result shows state-of-the-art accuracy of 100%, which indicates the efficiency of the binary restricted Boltzmann machine algorithm.

Keywords: astronomical instrumentation — methods and techniques — methods: analytical — methods: data analysis — methods: statistical

1 INTRODUCTION

With the rapid development of both the astronomical instruments and various machine learning algorithms, we can apply the spectral characteristics of stars to classify the stars. A great quantity of astronomical observatories have been built to get the spectra, such as the Large Sky Area Multi-Object Fibre Spectroscopic Telescope (LAMOST) in China. A variety of machine learning methods, e.g., principal component analysis (PCA), locally linear embedding (LLE), artificial neural network (ANN), and decision tree etc., have been applied to classify these spectra in an automatic and efficient way. In this study, we apply a novel machine learning method, restricted Boltzmann machine (RBM), to classify the cataclysmic variables (CVs) and non-CVs.

CVs are composed of the close binaries that contain a white dwarf accreting material from its companion (Warner 2003). Generally, they are small with an orbital period of 1–10 h. The white dwarf is often called 'primary' star, whereas the normal star is called the 'companion' or the 'secondary' star. The companion star, which is 'normal' like our Sun, usually loses its material onto the white dwarf via accretion.

The three main types of CVs are novae, dwarf novae, and magnetic CVs. Magnetic CVs (mCVs) are binary star systems with low mass and also with a Roche lobe-filling red dwarf, which 'gives' material to a magnetic white dwarf. Polars (AM Herculis systems) and Intermediate Polars (IPs)

are two major subclasses of mCVs (Wu 2000). More than a dozen of objects have been classified as AM Her systems. Most of the objects were found to be X-ray sources[1] before classified as AM Her's resorting to optical observations.

Besides, Muno et al. (2009) presented a catalogue of 9017 X-ray sources identified in *Chandra* observations of a $2 \times 0.8°$ field around the Galactic center. And they found that the detectable stellar population of external galaxies in X-rays was dominated by accreting black holes and neutron stars, while most of their X-ray sources may be CVs.

1.1 Previous work in spectral classification in astronomy

Singh, Gulati, & Gupta (1998) applied PCA and ANN to stellar spectral classification on O- to M-type stars, where O-type stars are the hottest and the letter sequence (O to M) indicates successively cooler stars up to the coolest M-type stars. They adopted PCA for dimension reduction firstly, in which they reduced the dimension to 20, with the cumulative percentages larger than 99.9%. Then, they used multi-layer back propagation (BP) neural network for classification.

Sarty and Wu (2006) applied two well-known multivariate analysis methods, i.e., PCA and discriminant function analysis, to analyse the spectroscopic emission data collected by

---

[1] http://ttt.astro.su.se/~stefan/amher0.html

1

Williams (1983). By using the PCA method, they found that the source of variation had correlation to the binary orbital period. With the discriminant function analysis, they found that the source of variation was connected with the equivalent width of the $H\beta$ line (Sarty & Wu 2006).

McGurk et al. (2010) applied PCA to analyse the stellar spectra obtained from SDSS (Sloan Digital Sky Survey) DR6. They found that the first four principal components (PCs) could remain enough information of the original data without overpassing the measurement noise. Their work made classifying novel spectra, finding out unusual spectra, and training a variety of spectral classification methods, etc., not as hard as before.

Bazarghan (2012) applied self-organising map (SOM, a kind of unsupervised ANN algorithm) to stellar spectra obtained from the Jacoby, Hunter, and Christian (JHC) library, and the author obtained the accuracy of about 92.4%. In the same year, Navarro, Corradi, & Mampaso (2012) used the ANN method to classify the stellar spectra with low signal-to-noise ratio (S/N) on the samples of field stars, which were along the line of sight towards NGC 6781 and NGC 7027 etc.,. They not only trained but also tested the ANNs with various S/N levels. They found that the ANNs were insensitive to noise and the ANN's error rate was smaller, when there were two hidden layers in the architecture of the ANN in which there were more than 20 hidden units in each hidden layer.

In the above, some applications of PCA for dimension reduction and ANN for spectral classification were reviewed in astronomy. Furthermore, support vector machine (SVM) and decision trees have also been used for spectral classification in astronomy.

Zhang and Zhao (2004) applied single-layer perceptron (SLP) and SVMs etc., for the binary classification problem, i.e., the classification of AGNs (active galactic nucleus) and S & G (stars and normal galaxies), in which they first selected features using the histogram method. They found that SVM's performance was as good as or even better than that of the neural network method when there were more features chosen for classification. Ball et al. (2006) applied decision trees to SDSS DR3. They investigated the classification of 143 million photometric objects and they trained the classifier with 477 068 objects. There were three classes, i.e., galaxy, star, and neither of the former two classes, in their experiment.

From the perspective of feature extraction methods, some researches in spectral classification based on linear dimension reduction technique, e.g., PCA, have been reviewed. Except from linear dimension reduction method, nonlinear dimension reduction technique has also been applied in spectral classification for feature extraction.

Daniel et al. (2011) applied locally linear embedding (LLE, a well known nonlinear dimension reduction technique) to classify the stellar spectra coming from the SDSS DR7. There were 85 564 objects in their experiment. They found that most of the stellar spectra was approximately a one-dimensional (1D) sequence lying in a three-dimensional (3D) space. Based on the LLE method, they proposed a novel hierarchical classification method being free of the feature extraction process.

1.2 Previous application of RBM

In this subsection, we present some representative applications of the RBM algorithm so far.

Salakhutdinov, Mnih, & Hinton (2007) applied RBM for collaborative filtering, which is closely related to recommendation system in machine learning community. Gunawardana & Meek (2008) applied RBM for cold-start recommendations. Taylor & Hinton (2009) applied RBM for modeling motion style. Dahl et al. (2010) applied RBM to phone recognition on the TIMIT data set. Schluter & Osendorfer (2011) applied RBM to estimate music similarity. Tang, Salakhutdinov, & Hinton (2012) applied RBM for recognition and de-noising on some public face databases.

1.3 Our work

In this study, we applied the binary RBM algorithm to classify spectra of CVs and non-CVs obtained from the SDSS.

Generally, before applying a classifier for classification, the researchers always preprocess the original data, for example, normalisation to get better features and thus to get better performance. Thus, firstly, we normalise the spectra with unit norm[2]. Then, to apply binary RBM for spectral classification, we binarise the normalised spectra by some rule, which we will discuss in the experiment. Finally, we use the binary RBM for classification of the data, one-half of all the given data for training and the other half for testing. The experiment result shows that the classification accuracy is 100%, which is state-of-the-art. And RBM outperforms the prevalent classifier, SVM, with accuracy of 99.14% (Bu et al. 2013).

The rest of this paper is organised as follows. In Section 2, we review the prerequisites for training RBM. In Section 3, we introduce the binary RBM and the training algorithm for RBM. In Section 4, we present the experiment result. Finally, in Section 5, we conclude our work in this study and also present the future work.

2 PREREQUISITES

2.1 Markov chain

A Markov chain is a sequence composed of a number of random variables. Each element in the sequence can transit from one state to another one randomly. Indeed, a Markov chain belongs to a stochastic process (Andrieu et al. 2003). In general, the number of possible states for each element or random variable in a Markov chain is finite. And a Markov

---

[2] We say the norm of a vector $\boldsymbol{x} = (x_1, \ldots, x_n)$ is unit, if $\Sigma_i x^2_i = 1$.

chain is a random process without memory. It is the current state rather than the states preceding the current state that can influence the next state of a Markov chain. This is the well-known Markov Property (Xiong, Jiang, & Wang 2012).

Mathematically, a Markov chain is a sequence, X_1, X_2, $X_3, \ldots$, with the following property:

$$P(X_{n+1} = x_{n+1}|X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n)$$
$$= P(X_{n+1} = x_{n+1}|X_n = x_n),$$

where the X_i ($i = 1, 2, \ldots$) is a random variable and it usually can take on finite values for a specific problem in the real world. And all the values as a whole can form a denumerable set S, which is commonly called the state space of the Markov chain (Yang et al. 2009).

Generally, all the probabilities of the transition from one state to another one can be represented as a whole by a transition matrix. And the transition matrix has the following three properties:

- square: both the row number of the matrix and the column number of the matrix equal the total number of the states that the random variable in the Markov chain can take on;
- the value of a specific element is between 0 and 1: it represents the transition probability from one state to another one;
- all the elements in each row sum to 1: the sum of the transition probabilities from any specific one state to all the states equals 1.

If the initial vector, a row vector, is X_0, and the transition matrix is T, then after n steps of inference, we can get the final vector $X_0 \cdot \mathrm{T}^n$.

Then, we introduce the equilibrium of a Markov chain. If there exists an integer $\tilde{N}$, which renders all the elements in the resulting matrix $\mathrm{T}^{\tilde{N}}$ nonzero, or rather, greater than 0, then we say that the transition matrix is a regular transition matrix (Greenwell, Ritchey, & Lial 2003). If the transition matrix T is a regular transition matrix, and there exists one and only one row vector V satisfying the condition that $v \cdot \mathrm{T}^n$ approximately equals V, for any probability vector v and large enough integer n, then we call the vector V as the equilibrium vector of the Markov chain.

2.2 MCMC

Markov chain Monte Carlo (MCMC) is a sampling algorithm from a specific probability distribution. For the detailed information of MCMC, the readers are referred to Andrieu et al. (2003). The sampling process proceeds in the form of a Markov chain and the goal of MCMC is to get a desired distribution, or rather, the equilibrium distribution via running many inference steps. The larger the number of iterations is, the better the performance of the MCMC is. And MCMC can be applied for unsupervised learning with some hidden variables or maximum likelihood estimation (MLE) learning of some unknown parameters (Andrieu et al. 2003).

2.3 Gibbs sampling

Gibbs sampling method can be used to obtain a sequence of approximate samples from a specific probability distribution, in which sampling directly is usually not easy to implement. For the detailed information of the Gibbs sampling, the readers are referred to Gelfand (2000). The sequence obtained via the Gibbs sampling method can be applied to approximate the joint distribution and the marginal distribution with respect to (w.r.t.) one of all the variables etc. In general, the Gibbs sampling method is a method for probabilistic inference.

The Gibbs sampling method can generate a Markov chain of random samples under the condition that each of the sample is correlated with the nearby sample, or rather, the probability of choosing the next sample equals to 1 in the Gibbs sampling (Andrieu et al. 2003).

3 RBM

Considering that RBM is a generalised version of Boltzmann Machine (BM), we first review BM in this section. For the detailed information of BM, the readers are referred to Ackley, Hinton, & Sejnowski (1985).

BM can be regarded as a bipartite graphical generative model composed of two layers in which there are a number of units with both inter-layer and inner-layer connections. One layer is a visible layer v with m binary visible units v_i, i.e., $v_i = 0$ or $v_i = 1$ ($i = 1, 2, \ldots, m$). For each unit in the visible layer, the corresponding value is observable. The other layer is a hidden (latent) layer h with n binary hidden units h_j. As in the visible layer, $h_j = 0$ or $h_j = 1$ ($j = 1, 2, \ldots, n$). For each unit or neuron in the hidden layer, the corresponding value is hidden, latent or unobservable, and it needs to be inferred.

The units coming from the two layers of a BM are connected with weighted edges completely, with the weights w_{ij} ($v_i \leftrightarrow h_j$) ($i = 1, 2, \ldots, m, j = 1, 2, \ldots, n$). For the two layers, the units within each specific layer are also connected with each other, and also with weights.

For a BM, the energy function can be defined as follows:

$$E(v, h) = -\sum_{i,j=1}^{m} v_i a_{ij} v_j - \sum_{i,j=1}^{n} h_i d_{ij} h_j$$
$$- \sum_{i=1}^{m}\sum_{j=1}^{n} v_i w_{ij} h_j - \sum_{i=1}^{m} v_i c_i - \sum_{j=1}^{n} h_j b_j, \quad (1)$$

where a_{ij} is the weight of the edge connecting visible units v_i and v_j, d_{ij} the weight of the edge connecting hidden units h_i and h_j, w_{ij} the weight of the edge connecting visible unit v_i and hidden unit h_j. For a RBM, the b_j is the bias for the hidden unit h_j in the following activation function (Sigmoid

function $f(x) = sigmoid(x) = 1/(1 + e^{-x}))$

$$p(h_j = 1|v) = \frac{1}{1 + e^{-b_j - \sum_{i=1}^{m} w_{ij} v_i}}.$$

And in a RBM, the c_i is the bias for the visible unit v_i in the following formula:

$$p(v_i = 1|h) = \frac{1}{1 + e^{-c_i - \sum_{j=1}^{n} w_{ij} h_j}}.$$

Then, for each pair of a visible vector and a hidden vector (v, h), the probability of this pair can be defined as follows:

$$p(v, h) = \frac{e^{-E(v,h)}}{PF},$$

where the denominator PF in the fraction (a partition function) is:

$$PF = \sum_{\widetilde{v}, \widetilde{h}} p(\widetilde{v}, \widetilde{h}). \tag{2}$$

Besides, a RBM is a graphical model with the units for both layers not connected within a specific layer, i.e., there are only connections between the two layers for the RBM (Hinton & Salakhutdinov 2006). Mathematically, for a RBM, $a_{ij} = 0$ for $i, j = 1, 2, \ldots, m$ and $d_{ij} = 0$ for $i, j = 1, 2, \ldots, n$. Thus, the states of all the hidden units h_j's are independent given a specific visible vector v and so are the visible units v_i's given a specific hidden vector h. Then, we can obtain the following formula:

$$p(h|v) = \prod_j p(h_j|v) \text{ and } p(v|h) = \prod_i p(v_i|h).$$

3.1 Contrastive divergence

Contrastive divergence (CD) is proposed by Hinton and it can be used to train RBM (Hinton, Osindero, & Teh 2006). Initially, we are given v_i ($i = 1, 2, \ldots, m$), then we can obtain h_j ($j = 1, 2, \ldots, n$) by the sigmoid function given in the above. And the value of h_j is determined by comparing a random value r ranging from 0 to 1 with the probability $p(h_j = 1|v)$. Then, we can reconstruct v by $p(v_i = 1|h)$.

We can repeat the above process backward and forward until the reconstruction error is small enough or it has reached the maximum number of iterations, which is set beforehand. To update the weights and biases in a RBM, it is necessary to compute the following partial derivative:

$$\frac{\partial \log p(v, h)}{\partial w_{ij}} = E_{\text{data}}[v_i h_j] - E_{\text{recon}}[v_i h_j], \tag{3}$$

$$\frac{\partial \log p(v, h)}{\partial c_i} = v_i - E_{\text{recon}}[v_i], \tag{4}$$

$$\frac{\partial \log p(v, h)}{\partial b_j} = E_{\text{data}}[h_j] - E_{\text{recon}}[h_j], \tag{5}$$

where $E[\star]$ represents the expectation of $\star$, and the subscript 'data' means that the probability is original-data-driven

whereas the subscript 'recon' means that the probability is reconstructed-data-driven.

Then, the weight can be updated according to the following rule:

$$\Delta w_{ij} = \eta (E_{\text{data}}[v_i h_j] - E_{\text{recon}}[v_i h_j]),$$

where η is a learning rate, which influences the speed of convergence. And the biases can be updated similarly.

In Equations (3)–(5), $E_{\text{data}}[\star]$'s are easy to compute. To compute or inference the latter term $E_{\text{recon}}[\star]$, we can resort to MCMC.

3.2 Free energy and soft-max

To apply RBM for classification, we can resort to the following technique. We can train a RBM for each specific class. And for classification, we need the free energy and the soft-max function for help. For a specific visible input vector v, its free energy equals to the energy that a single configuration must own and it equals the sum of the probabilities of all the configurations containing v. In this study, the free energy (Hinton 2012) for a specific visible input vector v can be computed as follows:

$$F(v) = - \left[\sum_i v_i c_i + \sum_j \log(1 + e^{x_j}) \right], \tag{6}$$

where $x_j = b_j + \Sigma_i v_i w_{ij}$.

For a given specific test vector v, after training the RBM_c on a specific class c, the log probability that RBM_c assigns to v can be computed according to the following formula:

$$\log p(v|c) = -F_c(v) - \log PF_c,$$

here, the PF_c is the partition function of RBM_c. For a specific classification problem, if the total number of classes is small, there will be no difficulty for us to get the unknown log partition function. In this case, given a specific training set, we can just train a 'soft-max' model to predict the label for a visible input vector v resorting to the free energies of all the class-dependent RBM_c's:

$$\log p(label = c|v) = \frac{e^{-F_c(v) - \log \widetilde{PF}_c}}{\sum_d e^{-F_d(v) - \log \widetilde{PF}_d}}. \tag{7}$$

In Equation (7), all the partition functions $\widetilde{PF}s$ can be learned by maximum likelihood (ML) training of the 'soft-max' function, where the maximum likelihood method is a kind of parameter estimation method generally with the help of the log probability. Here, the 'soft-max' function for a specific unit is generally defined in the following form:

$$p_j = \frac{e^{x_j}}{\sum_{i=1}^{k} e^{x_i}},$$

and the parameter k means that there are totally k different states that the unit can take on.

For clarity, we show the complete RBM algorithm in the following. The RBM algorithm as a whole based on the CD method can be summarised as follows:

- Input: a visible input vector v, the size of the hidden layer n_h, the learning rate η, and the maximum epoch M_e;
- Output: a weight matrix W, a biases vector for the hidden layer b, and a biases vector for the visible layer c;
- Training:
  Initialisation: Set the visible state with $v_1 = x$, and set W, b, and c with small (random) values,
  For $t = 1, \ldots, M_e$,
  For $j = 1, \ldots, n_h$,
  Compute the following value:
  $p(h_{1j} = 1|v_1) = sigmoid(b_j + \sum_i v_{1i} W_{ij})$;
  Sample h_{1j} from the conditional distribution $P(h_{1j}|v_1)$ with the Gibbs sampling method;
  End
  For $i = 1, 2, \ldots, n_v$, //Here, the n_v is the size of the visible input vector v
  Compute the following value:
  $p(v_{2j} = 1|h_1) = sigmoid(c_i + \sum_j W_{ij} h_{1j})$;
  Sample v_{2i} from the conditional distribution $P(v_{2i}|h_1)$ with the Gibbs sampling method;
  End
  For $j = 1, \ldots, n_h$,
  Compute the following value:
  $p(h_{2j} = 1|v_2) = sigmoid(b_j + \sum_i v_{2i} W_{ij})$;
  End
  Update the parameters:
  W = W + $\eta[P(h_1 = 1|v_1)v_1 - P(h_2 = 1|v_2)v_2]$;
  $c = c + \eta(v_1 - v_2)$;
 $c = c + \eta[P(h_1 = 1|v_1) - P(h_2 = 1|v_2)]$;
  End

For classification, after training the RBM using the above algorithm, we need to compute the free energy function by Equation (6) and then we can assign a label for the sample v with Equation (7).

4 EXPERIMENT

4.1 Data description

There have been a large amount of surveys in astronomy. SDSS is one of those surveys and it is one of the most not only ambitious but also influential ones (The official website of SDSS is http://www.sdss.org/). The SDSS has begun collecting data since 2000. From 2000 to 2008, the SDSS collected deep and multi-colour images containing no less than a quarter of the sky and it also created 3D maps for over 930 000 galaxies and also for over 120 000 quasars. Data Release 7 (DR7) is the seventh major data release and it provides spectra and redshifts etc., for downloading.
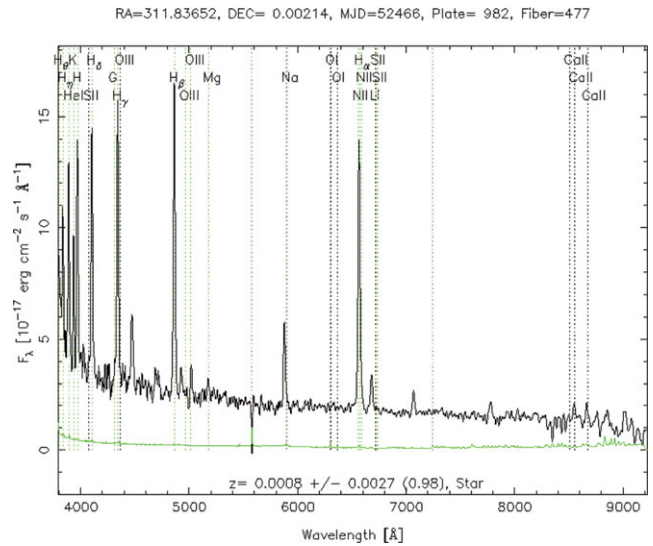
Figure 1. Spectrum of a cataclysmic variable star. The online version is available at: http://cas.sdss.org/dr7/en/tools/explore/obj.asp?id=587730847423725902.

All the data used in our experiment are coming from the SDSS. All the samples in the entire data set are divided into two classes, one class composed of non-CVs whereas the other class composed of CVs. There are totally 6 818 non-CVs and 208 CVs in our data set. Each sample is composed of 3 522 variables, or rather, spectral components. Among the total 6 818 non-CVs, there are 1 559 belonging to Galaxies, 3 981 belonging to Stars, and the remaining 1 278 belonging to QSOs (Quasi-stellar objects)[3].

In the following, we show the CVs in detail in our experiment. It is common that there will be transparent Balmer absorption lines in their spectra when the CVs outburst. A representative spectrum of the CV from the SDSS is shown in Figure 1. Much work has been done on the CVs for ages. Without high-tech, the earlier researches are focused on the optical characteristics of the spectrum. Then, with the help of the high-tech astronomical instruments, the multi-wavelength studies of the spectrum become to be true and the astronomers can obtain much more information about the CVs than before (Bu et al. 2013). From 2001 to 2006, Szkody et al. had been using the SDSS to search for CVs. The CVs in our data set are from their studies (Szkody et al. 2002, 2003, 2004, 2005, 2006, 2007), and we are deeply grateful to their researches. For clarity, we show the number of the CVs they found using the SDSS in Table 1. And the spectrum of a CV in our data set is shown in Figure 2.

In our experiment, we chose randomly half of the whole data for training and the remaining half for testing for both non-CVs and CVs. In detail, for non-CVs, half of the total 6 818 samples (i.e. 3 409) were randomly chosen to train the RBM classifier and the remaining half to test the RBM

---

[3] For detail, the readers are referred to the official website of SDSS DR7: http://www.sdss.org/dr7/

Table 1. The number of the CVs that Szkody et al. searched using the SDSS.

Paper	# of CVs
Szkody et al. (2002)	22
Szkody et al. (2003)	42
Szkody et al. (2004)	36
Szkody et al. (2005)	44
Szkody et al. (2006)	41
Szkody et al. (2007)	28
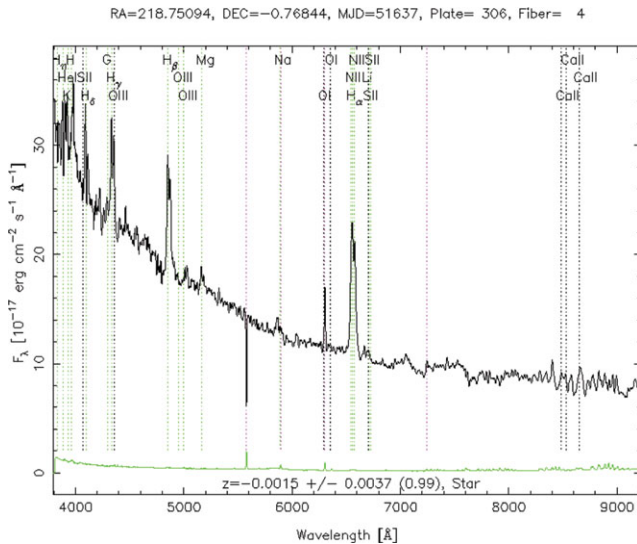
Figure 2. Spectrum of a cataclysmic variable star in our data set.

Table 2. The number of the original data for training and for testing respectively, where the number 3 522 is the dimension of the original data.

CV/non-CV	Train/Test	#×Dim
non-CV	Train	3 409 × 3 522
non-CV	Test	3 409 × 3 522
CV	Train	104 × 3 522
CV	Test	104 × 3 522

classifier. Similarly, for CVs, half of all the 208 samples (i.e. 104) were randomly chosen to train the RBM classifier and the remaining half to test the RBM classifier. To explain it clearly, we showed the data used for training and testing the RBM classifier in Table 2.

4.2 Parameter chosen

In this subsection, we present the parameters in our experiment. We chose all the parameters referring to Hinton (2012). The learning rate in the process of updating was set to be 0.1. The momentum for smoothness and to prevent over-fitting was chosen to be 0.5. The maximum number of epochs was chosen to be 50. The weight decay factor, penalty, was chosen to be 2×10^{-4}. The initial weights were randomly generated

Table 3. The parameters in our experiment.

Parameter	Value
Learning rate	0.1
Momentum	0.5
Maximum epochs	50
Number of hidden units	100
Initial biases vector	*0*

Table 4. The classification accuracy with different $\alpha's$.

α	Accuracy
1/5	97%
2/5	**100%**
3/5	97%
4/5	97%
1/4	97%
1/2	97.2%
3/4	97%
1/3	**100%**
2/3	97%

from the standard normal distribution, while the biases vectors b and c were initialised with 0. For clarity, we present them in Table 3.

4.3 Experiment result

We first normalised the data to make it have unit l_2 norm, i.e. for a specific vector $x = [x_1, x_2, \ldots, x_n]$, the l_2 norm of the vector satisfies $\Sigma_i x^2_i = 1$. Then, we could get two matrixes, one was A = 6818×3522 and the other was B = 208×3522. Then, we found out the maximum element and the minimum element for CVs and non-CVs, respectively. Finally, to apply binary RBM for classification, we found a parameter to assign the value of the variable in our experiment with 0 or 1, or rather, binarisation.

Mathematically, if

$$S(i, j) - \min S(i, j) < \alpha (\max S(i, j) - \min S(i, j)),$$

then we set $S(i, j)$ with 0, otherwise we set $S(i, j)$ with 1. Here, we used $S(i, j)$ (after binarisation) to denote the element of the matrix A and B in the ith row and the jth column. The parameter α satisfied $0 < \alpha < 1$. To investigate the influence of the parameter α on the final performance of the binary RBM algorithm, we first chose it to be 1/2 heuristically. Then, we chose it to be 1/3. The experiment result shows that the classification accuracy is 100%, which is state-of-the-art and it outperforms the prevalent classifier SVM (Bu et al. 2013).

For clarity, we show the result in Table 4, in which we also show the performance the binary RBM algorithm based on other values for the variable α. From Table 4, we can see that

the classification accuracy is 97.2%, when $\alpha = 1/2$. However, almost all of the CVs for testing is labelled as non-CVs.

Table 4 shows the classification accuracy computed by the following formula:

$$Acc = \frac{\sum[\hat{y} == y]}{Card(y)}, \tag{8}$$

where y is a vector, denoting the label of all the test samples. In our experiment, there are 3 413 (3 409 non-CVs + 104 CVs) test samples. And 'Card(y)' represents the number of elements in vector y. In Equation (8), the denominator $\hat{y}$ is the label of all the test samples predicted by Equation (7), in which $c = +1$ or $c = -1$. In this paper, $c = +1$ means that the sample belongs to non-CVs, whereas $c = -1$ means that the sample belongs to CVs[4]. And $\sum[\hat{y} == y]$ means the total number of equal elements in vector y and vector $\hat{y}$.

5 CONCLUSION AND FUTURE WORK

RBM is a bipartite generative graphical model, which can extract features representing the original data well. By introducing free energy and soft-max function, RBM can be used for classification. In this paper, we apply RBM for spectral classification of non-CVs and CVs. And the experiment result shows that the classification accuracy is 100%, which is the state-of-the-art and outperforms the rather prevalent classifier, SVM.

Since RBM is the building block of deep belief nets (DBNs) and deep Boltzmann machine (DBM), then we can infer that DBM (Salakhutdinov & Hinton 2009) and DBN can also perform well on spectral classification, which is our future work.

REFERENCES

Ackley, D. H., Hinton, G. E., & Sejnowski, T. J. 1985, Cogn. Sci., 9, 147

Andrieu, C., De Freitas, N., Doucet, A., et al. 2003, Mach. Learn., 50, 5

Ball, N. M., Brunner, R. J., Myers, A. D., & Tcheng, D. 2006, ApJ, 650, 497

Bazarghan, M. 2012, Ap&SS, 337, 93

Bu, Y., Pan, J., Jiang, B., Chen, F., & Wei, P. 2013, PASA, 30, e24

Dahl, G., Mohomed, A. R., & Hinton, G. E. 2010, Ad. NIPS, 23, 469

Daniel, S. F., Connolly, A., Schneider, J., Vanderplas, J., & Xiong, L. 2011, AJ, 142, 203

Gelfand, A. E. 2000, J. Am. Stat. Assoc., 95, 1300

Greenwell, R. N., Ritchey, N. P., & Lial, M. L. 2003, Calculus with Applications for the Life Sciences (Boston, MA: Addison-Wesley)

Gunawardana, A., & Meek, C. 2008, in Proceedings of the 2008. Vol. 2. no. 3, ed. P. Pu, D. Bridge, B. Mobasher, & F. Ricci. Lausanne, Switzerland: ACM Conference on Recommender Systems, (ACM), 19

Hinton, G. E. 2012, in Neural Networks: Tricks of the Trade, ed. G. Montavon, G. B. Orr, & K.-R. Müller (Berlin Heidelberg: Springer), 599

Hinton, G. E., Osindero, S., & Teh, Y. W. 2006, Neural Computation, 18, 1527

Hinton, G. E., & Salakhutdinov, R. R. 2006, Sci., 313, 504

Muno, M. P. et al. 2009, ApJS, 181, 110

McGurk, R. C., Kimball, A. E., & Ivezić, Ž. 2010, AJ, 139, 1261

Navarro, S. G., Corradi, R. L. M., & Mampaso, A. 2012, A&A, 538, A76

Salakhutdinov, R., & Hinton, G. E. 2009, in AISTATAS (Vol. 5; Cambridge, MA: MIT Press), 448

Salakhutdinov, R., Mnih, A., & Hinton, G. 2007, in Proceedings of the 24th ICML. Vol. 24, no. 100, ed. Z. Ghahramani (Corvallis, Oregon, USA: ACM), 791

Sarty, G., & Wu, K. 2006, PASA, 23, 106

Schluter, J., & Osendorfer, C. 2011, in ICMLA, 2. Vol. 2, ed. X. Chen, T. S. Dillon, H. Ishbuchi, J. Pei, H. Wang, & M. A. Wani (Honolulu, Hawaii, USA: IEEE), 118

Singh, H. P., Gulati, R. K., & Gupta, R. 1998, MNRAS, 295, 312

Szkody, P. et al. 2002, AJ, 123, 430

Szkody, P. et al. 2003 AJ, 583, 430

Szkody, P. et al. 2004 AJ, 128, 1882

Szkody, P. et al. 2005 AJ, 129, 2386

Szkody, P. et al. 2006, AJ, 131, 973

Szkody, P. et al. 2007, AJ, 134, 185

Tang, Y., Salakhutdinov, R., & Hinton, G. 2012, in CVPR. Vol. 25, no. 290 (Providence, RI, USA: IEEE), 2264

Taylor, G. W., & Hinton, G. E. 2009, in ICML. Vol. 26, no. 129, ed. A. P. Danyluk, L. Bottou, & M. L. Littman (Montreal, Quebec, Canada: ACM), 1025

Warner, B. 2003, Cataclysmic Variable Stars (Cambridge: Cambridge University Press)

Williams, G. W. 1983, ApJS, 53, 523

Wu, K. 2000, Sp. Sci. Rev., 93, 611

Xiong, Z., Jiang, W., & Wang, G. 2012, in Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on, 81, 11, ed. G. Min, Y. Wu, L. (Chris) Liu, X. Jin, S. Jarvis, & A. Y. Al-Dubai (Washington, DC: IEEE Computer Society), 640

Yang, N., Tang, C., Wang, Y., Tang, R., Li, C., Zheng, J., & Zhu, J. 2009, in Advances in Data and Web Management, 5446, Lecture Notes in Computer Science ed. Q. Li, L. Feng, J. Pei, S. X. Wang, X. Zhou, & Q.-M. Zhu (Berlin, Heidelberg: Springer), 297

Zhang, Y., & Zhao, Y. 2004, A&A, 422, 1113

---

[4] You can use any two different integers to represent the labels of the samples belonging to non-CVs and CVs, and this does not impact the result of the experiment.