J. Astrophys. Astr. (2016) 37: 12

Stellar Spectral Classification with Locality Preserving Projections and Support Vector Machine

Liu Zhong-bao
School of Computer and Control Engineering, North University of China, Taiyuan 030051, China.
e-mail: liu_zhongbao@hotmail.com

Received 5 January 2016; accepted 25 February 2016
DOI: 10.1007/s12036-016-9387-8

Abstract. With the help of computer tools and algorithms, automatic stellar spectral classification has become an area of current interest. The process of stellar spectral classification mainly includes two steps: dimension reduction and classification. As a popular dimensionality reduction technique, Principal Component Analysis (PCA) is widely used in stellar spectra classification. Another dimensionality reduction technique, Locality Preserving Projections (LPP) has not been widely used in astronomy. The advantage of LPP is that it can preserve the local structure of the data after dimensionality reduction. In view of this, we investigate how to apply LPP+SVM in classifying the stellar spectral subclasses. In the comparative experiment, the performance of LPP is compared with PCA. The stellar spectral classification process is composed of the following steps. Firstly, PCA and LPP are respectively applied to reduce the dimension of spectra data. Then, Support Vector Machine (SVM) is used to classify the 4 subclasses of K-type and 3 subclasses of F-type spectra from Sloan Digital Sky Survey (SDSS). Lastly, the performance of LPP+SVM is compared with that of PCA+SVM in stellar spectral classification, and we found that LPP does better than PCA.

Key words. Stellar spectral classification—locality preserving projections (LPP)—support vector machine (SVM).

1. Introduction

With the development of astronomical observation instruments, a huge amount of spectra are obtained and it has become untenable to classify all of them ‘manually’. With the help of computer tools and algorithms, automatic spectral classification has become an area of current interest.

In recent years, several dimension reduction and classification methods are proposed and widely used in practice. In the perspective of dimension reduction methods, most of the previous works had focused on Principal Component Analysis (PCA) and Locally Linear Embedding (LLE). In the perspective of classification methods, Artificial Neural Network (ANN) and Support Vector Machine (SVM) are quite popular in spectral classification. In order to improve the efficiencies of classification, dimension reduction should be done firstly in many cases. As a widely used technique in astronomy (Deeming 1964; Connolly *et al*. 1995; Yip *et al*. 2004a, b; Singh *et al.* 1998; Ting *et al*. 2012), PCA is applied to extract the first 20 principal components of 575 spectra derived from the objective prism plates of Houk and Artificial Neural Network (ANN) is applied to classify the spectra (Storrie-Lombardi *et al.* 1994). PCA is first used in dimensionality reduction and then SVM is applied to classify the spectra from the RAdial Velocity Experiment (RAVE) and the Sloan Digital Sky Survey (SDSS) (Re Fiorentin *et al.* 2008). LLE as a nonlinear dimensionality reduction method tries to preserve the relationships between the input data. LLE is widely used in spectral classification (Vanderplas & Connolly 2009; Daniel *et al.* 2011). LLE is applied to reduce the dimensionality of SDSS spectra and it can be concluded that LLE performs well in astronomical data processing (Vanderplas & Connolly 2009). The authors describe LLE to be very good at separating stars, galaxies and quasars (Daniel *et al.* 2011). LLE was also applied to the stellar subclass classification in Bu *et al.* (2013). As a typical classification tool, ANN is quite popular in astronomy, especially in the stellar spectral classification A new and robust classification scheme based on ANN was proposed in Navarro *et al.* (2012). The Self Organizing Map (SOM) is used as an unsupervised ANN algorithm for classification of stellar spectra (Mahdi 2012). A probabilistic neural network model is used for automated classification of stellar spectra (Mahdi 2008). SVM is quite suitable to classify the spectral data SVM is applied to classify the spectra to separate quasi-stellar objects from variable stars, non-variable stars and microlensing events (Kim *et al.* 2011). Isomap+SVM was used to classify the stellar spectral data in Bu *et al.* (2014). In addition, PCA, LLE and ISOmetric Feature Mapping (ISOMAP) are applied to reduce the dimensionality of SDSS spectra (Riden 2002); the weighted frequent pattern tree in Cai *et al.* (2013) was used to mine the association rules of a stellar spectrum. We have also tried to deal with the spectral classification problem, and several methods have been proposed, such as Minimum within-class and Maximum between-class scatter Support Vector Machine (MMSVM) (Liu 2016), non-linearly assembling learning machine (Liu *et al.* 2016) and the method based on Fisher criterion and manifold learning (Liu & Song 2015).

To our knowledge, Locality Preserving Projections (LPP), a popular dimensionality reduction technique, has not been widely used in stellar spectra classification. LPP can be seen as an alternative to PCA – a classical linear technique that projects the data along the directions of maximal variance. When the high dimensional data lies on a low dimensional manifold embedded in the ambient space, LPP obtained the optimal linear approximations to the eigenfunctions of the Laplace Beltrami operator on the manifold. As a result, LPP shares many of the data representation properties of nonlinear techniques such as Laplacian eigenmaps or LLE.

In view of the above analysis, this paper describes the application of LPP+SVM in the stellar classification. The paper is organized as follows. The review of LPP and

SVM is provided in section 2. Experiments are provided in section 3 and section 4 concludes our work.

2. Review of LPP and SVM

2.1 *LPP*

LPP is a linear projective map that arises by solving a variational problem that optimally preserves the neighborhood structure of the data set. It builds a graph incorporating neighborhood information of the data set. Using the notion of the Laplacian of the graph, we then compute a transformation matrix which maps the data points to a subspace. This linear transformation optimally preserves local neighborhood information in a certain sense. The representation map generated by the algorithm may be viewed as a linear discrete approximation to a continuous map that naturally arises from the geometry of the manifold (He & Niyogi 2003). The algorithmic procedure is formally stated below:

Given a set $x_1, x_2, \ldots, x_N$ in R^n, find a transformation matrix A that maps these N points to a set of points $y_1, y_2, \ldots, y_N$ in $R^l (l << n)$, such that y_i represents x_i where $y_i = A^T x_i$.

2.1.1 *Constructing the adjacency graph.* Let G denote a graph with N nodes. We put an edge between nodes i and j if x_i and x_j are 'close'. There are two variations:

(a) *ε-Neighborhoods* (*parameter $\varepsilon \in R$*): Nodes i and j are connected by an edge if $\|x_i - x_j\|^2 < \varepsilon$ where the norm is the usual Euclidean norm in R^n.
(b) *k nearest neighbors* (*parameter $k \in N$*): Nodes i and j are connected by an edge if i is among k nearest neighbors of j or j is among k nearest neighbors of i.

The method of constructing an adjacency graph outlined above is correct if the data actually lie on a low dimensional manifold. Once such an adjacency graph is obtained, LPP will try to optimally preserve it in choosing projections.

2.1.2 *Choosing the weights.* Here, as well, we have two variations for weighting the edges. W is a sparse symmetric $N \times N$ matrix with W_{ij} having the weight of the edge joining vertices i and j, and 0 if there is no such edge.

(a) *Heat kernel* (*parameter $t \in R$*): If the nodes i and j are connected, put the weight $W_{ij} = \mathrm{e}^{-\frac{\|x_i - x_j\|^2}{t}}$ on the edge.
(b) *Simple-minded* (*No parameters*): $W_{ij} = 1$ if and only if vertices i and j are connected by an edge.

2.1.3 *Eigenmaps.* Compute the eigenvectors and eigenvalues for the generalized eigenvector problem:

$$XLX^T\mathbf{a} = \lambda XDX^T\mathbf{a}, \tag{1}$$

where D is a diagonal matrix whose entries are column (or row, since W is symmetric) sums of W, $D_{ii} = \sum_j W_{ji}$. $L = D - W$ is the Laplacian matrix. The i-th column of matrix X is x_i.

Let the column vectors $\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_{l-1}$ be the solutions of equation (1), ordered according to their eigenvalues, $\lambda_0 < \lambda_1 < \cdots < \lambda_{l-1}$. Thus, the embedding is as follows:

$$x_i \rightarrow y_i = A^T x_i, A = (\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_{l-1}) \tag{2}$$

where y_i is a l-dimensional vector, and A is a $n \times l$ matrix.

2.2 *SVM*

The geometry explanation of SVM is to find a hyperplane with maximal classification margin to separate two classes (Bovolo *et al.* 2010). Let the hyperplane $w^T x + b = 0$ and the margin be $2/\|w\|$. The optimal problem of SVM can be summarized as follows.

$$\min \; J(w, b, \xi_i) = \frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i, \tag{3}$$

$$\text{subject to: } y_i(w^T x_i + b) \geq 1 - \xi_i, \quad \text{for } i = 1, 2, \ldots, N, \tag{4}$$

$$\xi_i \geq 0, \quad \text{for } i = 1, 2, \ldots, N, \tag{5}$$

where the regularization parameter C is a constant to trade off the two goals. The large C, the more the error term is emphasized. Small C means that large classification margin is encouraged. The parameter ξ_i is the relaxation parameter which allows the existence of classification errors.

The decision function can be defined as follows:

$$f(x) = \text{sign}(w^T x + b_0). \tag{6}$$

The optimal threshold b_0 can be found by computing the following equation:

$$b_0 = \frac{1}{N} \sum_{i \in S} \left(y_i - \frac{1}{2} \sum_{j=1}^{N} \alpha_j y_j x_j^T x_i \right).$$

3. Experimental analysis

In this section, we will investigate the performance of LPP+SVM compared with PCA+SVM. The experimental environments are Intel core i3 CPU, 4G RAM, Windows 7 and Matlab7.0. All the data in our experiment are from Sloan Digital Sky Survey (SDSS), Data Release 8 (Almeida & Prieto 2013). The data sets consist of 4 subclasses of K-type spectra, K1-type, K3-type, K5-type and K7-type, whose signal-to-noise ratios (SNRs) are 10 < SNRs < 20, 3 subclasses of F-type spectra, F2-type, F5-type and F9-type, whose SNRs are 50 < SNRs < 60, as listed in Table 1a, b. In order to avoid the unbalanced problem of F-type stellar spectra, we randomly select

Table 1a. The total number of K start with 10 < SNRs < 20.

Stellar subclass type	K1	K3	K5	K7
Number	5505	6092	4597	4476

Table 1b. The total number of F start with 50 < SNRs < 60.

Stellar subclass type	F2	F5	F9
Number	1416	8156	13785

1200 spectra from each subclass. The following SQL statements are used to select the K-type spectra from the SDSS DR8 database:

```
select dbo.fGetUrlFitsSpectrum(specobjid),mjd,plate,fiberid,class,
   subclass,snmedian
from specojball
where subclass like ‘K%’
```

The SQL statements of selecting the F-type spectra are almost the same with K-type except the third statement changed to ‘where subclass like ‘F%’. These data with the wavelength coverage from 3800 Å to 9000 Å have been shifted to a common rest-frame and normalized to a constant total flux. In LPP, the k nearest neighbors method is used to construct the adjacency graph and we use $k = 5$ in our experiments. The simple-minded method is used in choosing the weight. The performances of PCA+SVM, LPP+SVM are dependent on the hyperparameters of the models, such as the output dimensions (for PCA and LPP) or the softening parameter (for SVM). We use 5-fold cross validation to select the experimental parameters. Parameters are selected by grid search. In SVM, the regularization parameter C is searched in the $\{0.1, 0.5, 1, 5, 10\}$ grid.

The stellar spectral classification process is composed of the following steps:

Step 1: The experimental dataset is divided into two parts, one for training and the other for test.
Step 2: The optimal projection **W** is calculated from the training dataset by PCA and LPP respectively.
Step 3: The test dataset is mapped to the low-dimensional subspace.
Step 4: SVM is applied in the low-dimensional subspace for classification.

In our experiments, 30, 40, 50, 60 and 70% of K-, F-spectral datasets are respectively used for training, and the remaining are used for test. The original data are projected to 5-dimensional space. The optimal values of the parameter C in SVM is shown in Table 2. Tables 3a, b give the comparative experimental results, described as ‘means ± standard deviation’ where ‘means’ denote the average precision on the experimental dataset and ‘standard deviation’ denotes the degree of deviation from the average performance, which respectively reflect the average performance and the stability of the classifiers.

It can be seen from Tables 3a, b that the classification accuracies of PCA+SVM, LPP+SVM rise with the size of training datasets. Compared with PCA+SVM, LPP+SVM performs better except when the training datasets are 30% of F-type

Table 2. The optimal values of the parameter C in SVM.

Parameter C	K starts	F starts
Optimal value	0.5	0.1

Table 3a. The comparative experimental results on the K-type dataset.

Training size	Test size	PCA+SVM	LPP+SVM
30%	70%	0.5685 ± 0.0614	0.6349 ± 0.0499
40%	60%	0.5849 ± 0.0349	0.6300 ± 0.0202
50%	50%	0.7601 ± 0.0595	0.7799 ± 0.0295
60%	40%	0.7753 ± 0.0653	0.8352 ± 0.0351
70%	30%	0.8556 ± 0.0456	0.9149 ± 0.0354
Average classification accuracy		0.7089 ± 0.0533	0.7590 ± 0.0340

Table 3b. The comparative experimental results on the F-type dataset.

Training size	Test size	PCA+SVM	LPP+SVM
30%	70%	0.5754 ± 0.0556	0.5750 ± 0.0321
40%	60%	0.5908 ± 0.0389	0.6908 ± 0.0297
50%	50%	0.6550 ± 0.0550	0.7500 ± 0.0400
60%	40%	0.7223 ± 0.0410	0.7355 ± 0.0445
70%	30%	0.7556 ± 0.0445	0.8611 ± 0.0389
Average classification accuracy		0.6598 ± 0.0470	0.7225 ± 0.0370

Table 4. The computational time of PCA+SVM, LPP+SVM and SVM.

Training size	Test size	PCA+SVM	LPP+SVM	SVM
50%	50%	78.1	127.7	224.3

spectral dataset. In view of the average performance, the classification accuracy of LPP+SVM is obviously higher than that of PCA+SVM.

As we know that partial information is lost after dimension reduction, and thus either PCA+SVM or LPP+SVM would not out perform SVM. But in view of the computational cost, the performances of PCA+SVM and LPP+SVM are much better than SVM. The computational time of PCA+SVM, LPP+SVM and SVM on the K starts is shown in Table 4, in which the unit of time is seconds. 50% of K-spectral datasets are respectively used for training, and the remaining are used for test.

It can be seen from Table 4 that the computational time of SVM is much longer than PCA+SVM and LPP+SVM. Specially, when the training data size gets large, the classification results cannot be obtained by SVM in a short time, but the other methods can. In view of classification accuracies, LPP+SVM performs better than PCA+SVM. Therefore, it can be concluded that LPP+SVM is a compromise of LPP+SVM in view of the computational time and SVM in view of classification accuracies.

4. Conclusions

Dimensionality reduction is quite important to stellar spectral classification. PCA and LLE are widely used in practice, while another popular dimensionality reduction algorithm LPP has not been used in astronomy. Therefore, in this paper, we investigate how to apply LPP in dimensionality reduction and SVM in classifying the stellar spectral subclasses. The experiment process is composed of the following steps: Firstly, LPP is used to reduce the dimension of spectral data. And then, SVM is applied to classify the 4 subclasses of K-type and subclasses of F-type spectra from Sloan Digital Sky Survey (SDSS). Lastly, the comparative experiment with PCA shows that the performance of LPP is better than PCA. The drawback of LPP is quite obvious that its computational cost is higher than PCA. With approximate neighbor searches and sparse decompositions, it may be made faster, but this remains an active area of research even in the Machine Learning community and we attempt to overcome the above problems in our future work.

References

Almeida, J. S., Prieto, C. A. 2013, *AJ*, **763**, 1.

Bovolo, F., Bruzzone, L., Carlin, L. 2010, *IEEE TIP*, **19(11)**, 2983.

Bu, Y. D., Chen, F. Q., Pan, J. C. 2014, *New Astron.*, **28**, 35.

Bu, Y. D., Pan, J. C., Jiang, B. *et al.* 2013, *PASJ*, **65**, 81.

Cai, J. H., Zhao, X. J., Sun, S. W. *et al.* 2013, *Res. Astron. Astrophys.*, **13(3)**, 334.

Connolly, A. J., Szalay, A. S., Bershady, M. A. *et al.* 1995, *AJ*, **110**, 1071.

Daniel, S. F., Connolly, A., Schneider, J. *et al.* 2011, *AJ*, **142**, 203.

Deeming, T. J. 1964, *MNRAS*, **127**, 493.

He, X. F., Niyogi, P. 2003, *Advances in Neural Information Processing Systems (NIPS)*, 153.

Kim, D. W., Protopapas, P., Byun, Y. I. *et al.* 2011, *APJ*, **735(2)**, 68.

Liu, Z. B. 2016, *J. Astrophys. Astron.*, accepted.

Liu, Z. B., Song, L. P. 2015, *PASP*, **127(954)**, 789.

Liu, Z. B., Song, L. P., Zhao, W. J. 2016, *MNRAS*, **455**, 4289.

Mahdi, B. 2008, *Bull. Astr. India*, **36**, 1.

Mahdi, B. 2012, *Astrophys. Space Sci.*, **1(337)**, 93.

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

Re Fiorentin, P., Bailer-Jones, C. A., Beers, T. C. *et al.* 2008, *American Institute of Physics Conference Series*, **1082**, 76.

Riden, J. 2002, Unsupervised learning on galaxy spectra, Thesis, Master of Science.

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

Storrie-Lombardi, M. C., Irwin, M. J., Hippel, T., Storrie-Lombardi, L. J. 1994, *Vistas in Astron*, **38**, 331.

Ting, Y. S., Freeman, K. C., Kobayashi, C. *et al.* 2012, *MNRAS*, **421**, 1231.

Vanderplas, J., Connolly, A. J. 2009, *AJ*, **138**, 1365.

Yip, C. W. *et al.* 2004a, *AJ*, **128**, 585.

Yip, C. W. *et al.* 2004b, *AJ*, **128**, 2603.