New Astronomy 28 (2014) 35–43

Contents lists available at ScienceDirect

New Astronomy

journal homepage: www.elsevier.com/locate/newast

Stellar spectral subclasses classification based on Isomap and SVM

Yude Bu [a,*], Fuqiang Chen [b], Jingchang Pan [c]

[a] School of Mathematics and Statistics, Shandong University, Weihai 264209, China
[b] College of Electronics and Information Engineering, Tongji University, Shanghai 201804, China
[c] School of Mechanical, Electrical & Information Engineering, Shandong University, Weihai 264209, China

HIGHLIGHTS

- We employ Isomap to extract the features within stellar spectra.
- Isomap is more efficient than PCA in extracting spectral classification information.
- We employ Isomap + SVM to the classification of stellar spectral subclasses.
- Isomap + SVM performs better than traditional method PCA + SVM with the default γ in SVM.
- We provide the optimal parameter for the stellar subclasses classification.

ARTICLE INFO

Article history:
Received 27 April 2013
Received in revised form 24 July 2013
Accepted 25 September 2013
Available online 10 October 2013

Keywords:
Principal component analysis
Isometric feature map
Support vector machine
Spectral subclasses classification

ABSTRACT

Isometric feature map (Isomap), a nonlinear dimension reduction technique, can preserve both the local and global structure of the data when embed the original data into much lower dimensional space. In this paper we will investigate the performance of Isomap + SVM in classifying the stellar spectral subclasses. We first reduce the dimension of spectra data by PCA and Isomap respectively. Then we apply support vector machine (SVM) to classify the 4 subclasses of K-type spectra from Sloan Digital Sky Survey (SDSS). The experiment result shows that Isomap-based SVM (IS) perform better than PCA-based SVM (PS) with the default γ in SVM, except on the spectra whose SNRs are between 5 and 10 in our experiment. The performance of PS and IS both change in a larger range with the increase of signal-to-noise ratio of the spectra.

1. Introduction

Since the development of astronomical observation instrument, we can obtain a huge amount of spectra. It clearly requires to extract the astronomical information from the spectra automatically and accurately. Various methods have been introduced in the spectral data processing, which includes feature extraction and spectral classification.

The principal component analysis (PCA) is among the most widely used feature extraction method. However, PCA is a linear dimensionality reduction method. It has been proved that PCA is efficient in extracting the spectral continuum information, but not sensitive to the line information of the spectrum. To overcome this difficult, the nonlinear method has been introduced in the spectral feature extraction procedure. Locally linear embedding and Isomap are two widely used nonlinear dimensional reduction method. In (Vanderplas and Connolly, 2009), the author investigate the application of LLE in the classification of galaxy spectra. In Daniel et al. (2011), the authors investigate the application of LLE in classifying the stellar spectra. They find that the LLE is efficient in classifying the galaxy spectra and stellar spectra.

Comparing with LLE and PCA, the Isomap has not been widely investigated in spectral feature extraction. In this paper we will investigate the performance of Isomap in spectral subclass classification. We will choose the K-type star as a demonstration class to show our conclusion. We will show that Isomap is more efficient than PCA in the feature extraction. Then we will test the performance of Isomap + SVM in the spectral classification. By comparing Isomap + SVM with PCA + SVM, we find that Isomap + SVM is accurate and efficient in spectral subclass classification.

The paper is organized as follows. We will first review the previous works on the spectral classification in Section 2. In Section 3, we will give a brief introduction to PCA. Then we will give a brief review of the Isomap in Section 4. After a short introduction to the support vector machine (SVM) in Section 5, we will compare the performance of PCA-based SVM and Isoamp-based SVM in Section 6. Section 7 concludes our work.

* Corresponding author. Tel./fax: +86 6315688523.
E-mail addresses: buyude001@163.com, 123974934@qq.com (Y. Bu).

1384-1076/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.newast.2013.09.007

2. Previous work

Spectral feature extraction and classification are the key procedure in spectral data procedure. The most widely used dimension reduction technique in this area is PCA. Storrie-Lombardi et al. (1994) applied PCA to extract the first 20 principal components (PCs) of the 575 spectra derived from the objective prism plates of Houk and then they applied artificial neural network (ANN) to classify the spectra (Storrie-Lombardi et al., 1994). Singh et al. (1998) reduced the dimension of the data (a library of optical stellar spectra for O to M type stars) by PCA and then applied multi-layer back propagation (MLBP) artificial neural network to classify them (Singh et al., 1998). Not only ANN was applied to classify the stellar spectra, SVM was also widely used in the spectral classification. Re Fiorentin et al. (2008) applied SVM to classify spectra from the RAdial Velocity Experiment (RAVE) and the Sloan Digital Sky Survey (SDSS/SEGUE) after dimension reduction by PCA (Re Fiorentin et al., 2008). Kim et al. (2011) applied SVM to separate quasi-stellar objects (QSOs) from variable stars, non-variable stars, and microlensing events (Kim et al., 2011).

It is well known that PCA is a linear dimension reduction technique, which may perform poorly when the global structure of the original data is nonlinear. Thus, a nonlinear dimension reduction technique, LLE, was introduced to deal with the nonlinear problem (Roweis and Saul, 2000). Vanderplas and Connolly (2009) applied LLE to reduce the dimension of Sloan Digital Sky Survey spectra, and they found that it performed better than PCA (Vanderplas and Connolly, 2009). Daniel et al. (2011) applied LLE to reduce the dimension of stellar spectra from SDSS and presented a hierarchical classification scheme (Daniel et al., 2011). In addition, Riden (2002) applied PCA, LLE and Isomap to reduce the dimension of spectra from SDSS, in which he only considered galaxies and wavelengths common to all the galaxies (Riden, 2002).

2.1. Our work

In this paper, we will investigate the performance of Isomap in the stellar subtypes classification. We first apply PCA and Isomap to reduce the dimension of stellar spectra, respectively. Then we apply SVM to classify the spectra. When applying PCA, we project the original data into the 1–10 dimension, and find that the first 10 principal components remaining more than 92% of the original information. When applying Isomap, various values of parameter k in Isomap has been used in experiment to investigate it's effect on the performance of Isomap. We then determine the optimal k for Isomap. In our paper, we choose the popular radial basis function (RBF) as the kernel function in SVM, and we investigate the parameter γ's (a parameter in RBF) effect on the performance of SVM.

3. Brief introduction to PCA

In this section, we will introduce PCA briefly. For more detailed information about PCA, we refer the reader to Jolliffe (2002). It has been widely applied in spectral classification (Storrie-Lombardi et al., 1994; Singh et al., 1998; Re Fiorentin et al., 2008; Kim et al., 2011; Roweis and Saul, 2000).

For consistency and simplicity, we suppose that the original data is a set composed of n samples, $X = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)^T$, in which $\overline{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id})^T$ $(i = 1, 2, \ldots, n)$ is a d-dimensional column vector (i.e., there are d features for each sample) and each x_{ij}, $j = 1, 2, \ldots, d$, can be regarded as a random variable (in all of this paper, we refer to this data set unless specified).

Generally, the variances of different features may be different from each other largely. So before applying PCA, we need to normalize the original data with zero mean and unit variance for each feature. For a specific feature, we can substract the mean of all the samples from the original data to get zero mean, then we can compute the sum of the squares of the difference between the original data and the mean. Finally, we can get unit variance by dividing all the data (after equalization) by the square root of the sum of the squares.

Mathematically, we can compute the following matrix firstly:

$$X_s = CXA^{-1}, \tag{1}$$

where

$$C = E - \frac{1}{n}A_O$$

is the "centering matrix". E is the identity matrix[1] and A_O is a matrix with all of its elements equal 1. A is a diagonal matrix

$$A = diag\{\|\overline{HX}_{(1)}\|, \ldots, \|\overline{HX}_{(d)}\|\},$$

i.e., its diagonal element is $\|\overline{HX}_{(i)}\|(i = 1, 2, \ldots, d)$ where

$$\overline{X}_{(i)} = \begin{pmatrix} x_{1i} \\ x_{2i} \\ \vdots \\ x_{ni} \end{pmatrix}$$

and $\overline{H} = (1, \ldots, 1)$. Then we can renew X with X_s.

After renewing X, we need to compute the eigenvalues of matrix X^TX, where X^T is the transpose of the matrix X. Since the matrix X^TX is a real symmetric positive semi-definite matrix,[2] it has d non-negative eigenvalues. We can denote them as $\lambda_1, \lambda_2, \ldots, \lambda_d$ in decreasing order.

All of the eigenvalues have finite number of corresponding eigenvectors. For the jth eigenvalue λ_j $(j = 1, 2, \ldots, d)$, we denote the corresponding eigenvector as $\overline{\alpha}_j$, where $\overline{\alpha}_j$ is a d-dimensional column vector. Then we can get a matrix $\alpha = [\overline{\alpha}_1, \overline{\alpha}_2, \ldots, \overline{\alpha}_d]$. In practice, we need to normalize all the eigenvectors to unit l_2 norm.[3] If we rewrite X as follows

$$X = \begin{pmatrix} \overline{x}_1^T \\ \overline{x}_2^T \\ \vdots \\ \overline{x}_n^T \end{pmatrix}, \tag{2}$$

we can obtain the jth principal component (PC) of the original data by multiplying X with the jth column vector α_j.

Therefore, to obtain the first k principal components, i.e., reducing the original data into a k dimensional subspace (generally, $k \ll d$), we can multiply X with $\widetilde{\alpha} = [\overline{\alpha}_1, \overline{\alpha}_2, \ldots, \overline{\alpha}_k]$

$$X\widetilde{\alpha} = X[\overline{\alpha}_1, \overline{\alpha}_2, \ldots, \overline{\alpha}_k]. \tag{3}$$

Generally, we choose the k which satisfies

$$CR_k = \frac{\sum_{i=1}^{k}\lambda_i}{\sum_{i=1}^{d}\lambda_i} \geqslant 85\%,$$

where CR_k is the accumulative contribution rate of the first k principal components.

[1] We say a matrix is an identity matrix if all of its diagonal elements equal 1.

[2] We say a square matrix A of $n \times n$ is positive semi-definite matrix, if for any column vector $x \neq \mathbf{0}$, the following formula satisfies, $x^TAx \geqslant 0$.

[3] Suppose that a vector $\overline{a} = [a_1, a_2, \ldots, a_n]$, we say a has unit l_2 norm if $\sum_{i=1}^{n} a_i^2 = 1$.

4. Short review of Isomap

The well known Swiss roll is a classical example with nonlinear global structure. Since Euclidean distance can not measure the similarity of the points in Swiss roll well, J.B. Tenenbaum et al. proposed Isometric feature map (Isomap) (Tenenbaum et al., 2000), a nonlinear dimension reduction technique. Isomap can embed the Swiss roll into 2-dimensional space well. It is based on the geodesic distance globally and Euclidean distance locally, which guarantees the samples from different classes or clusters far away. In the following, we present a brief introduction to Isomap.

Generally, Isomap can be divided into the following three steps. For convenience and clarity, the problem mentioned in Section 3 will take as an example when we discuss dimension reduction by Isomap. There are totally n d-dimensional samples $\overline{x}_i$ $(i = 1, 2, \ldots, n)$ (in this section, we denote it as a column vector) in the data set X.

4.1. Create a graph (usually partially connected graph)

Firstly, for the n samples, we need to compute Euclidean distance between each pair of samples. In practice, we only need to compute $(n-1)+(n-2)+\cdots+1 = n(n-1)/2$ values because of the symmetry (i.e. $d(\overline{x}_i, \overline{x}_j) = d(\overline{x}_j, \overline{x}_i)$, for $i \neq j$) and $d(\overline{x}_i, \overline{x}_i) = 0, i, j = 1, 2, \ldots, n$.

Then it is required to connect two samples if the distance between them is smaller than ε (mathematically, i.e., $d(\overline{x}_i, \overline{x}_j) < \varepsilon$) or one point x_i is among the k nearest points of the other point x_j. Mathematically, if we denote

$$k_{near}(\overline{x}_i) = \{\overline{x}_j | \overline{x}_j \text{ is in the set of } k \text{ nearest points of } \overline{x}_i\},$$

we need to connect $\overline{x}_i$ and $\overline{x}_j$ when $\overline{x}_j \in k_{near}(\overline{x}_i)$.

Finally, we assign a path length between any connected two points $\overline{x}_i$ and $\overline{x}_j$ with $d(\overline{x}_i, \overline{x}_j)$.

4.2. Get the shortest paths between any two points

In step 1, if any two points are disconnected, then in step 2 the shortest path between them is initialized with $+\infty$; otherwise, the shortest path between them is initialized with the distance in step 1.

For any two samples, we apply Dijkstra algorithm (cf. Appendix A) to find the shortest path length between them. Finally we can obtain a distance matrix $D_X = (d(\overline{x}_i, \overline{x}_j))$, where $d(\overline{x}_i, \overline{x}_j)$ is the shortest geodesic distance (i.e., the shortest path length) between samples $\overline{x}_i$ and $\overline{x}_j$.

4.3. Embedding

The embedding step is similar to that of metric multidimensional scaling (MDS)(Appendix B). We attempt to find n points $\overline{y}_1, \ldots, \overline{y}_n$ in $\widetilde{d}$-dimensional space such that D_Y is similar to D_X, where $D_y = (d(\overline{y}_i, \overline{y}_j))_{n\times n}$ is the distance matrix. Here $d(\overline{y}_i, \overline{y}_j)$ denotes the Euclidean distance between $\overline{y}_i$ and $\overline{y}_j$. Mathematically, we need to solve the following problem

$$\min_Y \sum_{i=1}^{n}\sum_{j=1}^{n} (d(\overline{x}_i, \overline{x}_j) - d(\overline{y}_i, \overline{y}_j))^2.$$

Obviously, $d(\overline{y}_i, \overline{y}_j) = d(\overline{x}_i, \overline{x}_j)$ leads to the minimum value of above sum of square equation. Since $d(\overline{x}_i, \overline{x}_j)$ has been obtained in step 2, we now know the value of $d(\overline{y}_i, \overline{y}_j)$. Assume $B = (B_{ij}) = (\overline{y}_i^T \overline{y}_j)$, we can prove that $B = -CEC/2$, where $E = (e(i,j))_{n\times n}$ with $e(i,j) = d^2(\overline{y}_i, \overline{y}_j)$. The matrix $C = \{\delta_{ij} - \frac{1}{n}\}_{n\times n}$, where $\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq = j \end{cases}$. Thus, from $d(\overline{y}_i, \overline{y}_j)$ we can obtain the matrix B. We now attempt to construct $\overline{y}_i$ from matrix B.

Because both C and E are symmetric, B is also symmetric and positive semi-definite. Thus B has n nonnegative eigenvalues. Similar to the notation in PCA, we can denote the n eigenvalues as $\lambda_1, \lambda_2, \ldots, \lambda_n$ in decreasing order and denote the eigenvector of λ_i as $\overline{\alpha}_i$. Then, we can prove that $\overline{y}_i = \sqrt{\lambda_i}\overline{\alpha}_i$, which can be regarded as the $\widetilde{d}$ dimension embedding of $\overline{x}_i$.

When using PCA, once we have the eigenvectors, we can project new data onto them without having to re-solve the eigenvalue problem. However, when using Isomap, we have to re-start the Isomap algorithm to obtain the low dimensional representation of the new data points. This is an disadvantage of Isomap compared with PCA.

5. SVM overview

5.1. Two-class SVM

In this section, we introduce two-class supervised[4] SVM (Burges, 1998), and in Section 5.2 we introduce multi-class SVM. For convenience, the data set mentioned in Section 3 will be taken as an example when we introduce SVM. Here we suppose that $y_i \in \{-1, +1\}$ is the label of sample $\overline{x}_i (i = 1, 2, \ldots, n)$.

A hyperplane $y = \overline{\omega} \cdot \overline{x} + b$ is called the separating hyperplane if it has the following properties: if the label of $\overline{x}_i$ is -1, then $\overline{\omega} \cdot \overline{x}_i + b \leqslant -1$, and if the label of $\overline{x}_i$ is $+1$, then $\overline{\omega} \cdot \overline{x}_i + b \geqslant +1$. If this hyperplane can be constructed, we say that the sample data are separable, otherwise, the sample data are non-separable.

For the separable sample data, let $d_+ (d_-)$ be the shortest distance from the separating hyperplane to the closest data with $+1(-1)$ label. $d_+ + d_-$ is defined as the margin of the separating plane. We can prove that $d_+ + d_- = \frac{2}{\|\overline{\omega}\|}$. SVM attempts to find a hyperplane $y = \overline{\omega} \cdot \overline{x} + b$ with maximum margin in sample space. Mathematically, the problem of SVM can be expressed as follows

$$\min \frac{1}{2}\|\overline{\omega}\|^2, \tag{4}$$

$$\text{subject to } y_i(\overline{\omega} \cdot \overline{x}_i + b) \geqslant 1 \quad (i = 1, 2, \ldots, n).$$

By introducing positive Lagrange multipliers α_i, we can obtain Lagrangian

$$L_p = \frac{1}{2}\|\overline{\omega}\|^2 - \sum_{i=1}^{n} \alpha_i y_i (\overline{\omega} \cdot \overline{x}_i + b) + \sum_{i=1}^{n} \alpha_i.$$

Requiring that the gradient of L_p with respect to $\overline{\omega}$ and b vanish give the conditions:

$$\overline{\omega} = \sum_{i=1}^{n} \alpha_i y_i \overline{x}_i, \tag{5}$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0.$$

Considering that $\max_{\alpha_i} L_p \leqslant \frac{1}{2}\|\overline{\omega}\|^2$, we have

$$\max_{\alpha_i} \min_{\overline{\omega}, b} L_p \leqslant \min_{\overline{\omega}} \frac{1}{2}\|\overline{\omega}\|^2. \tag{6}$$

Substituting Eq. (5) into L_p, and considering Eq. (6), we obtain the dual problem of Problem 4

$$\max_{\alpha_i} L_D = \max \left\{ \sum_i \alpha_i - \frac{1}{2} \sum_{ij} \alpha_i \alpha_j y_i y_j \overline{x}_i \cdot \overline{x}_j \right\}. \tag{7}$$

[4] If the labels of the samples are known, we say the problem is supervised.

Using standard quadratic programming techniques and programs, we can obtain α_i $(i = 1, \ldots, n)$ which maximize L_D. Then we can get the corresponding $\overline{\omega}$ by Eq. (5). By equation

$$y_i(\overline{\omega} \cdot \overline{x}_i + b) - 1 \geqslant 0,$$

we can determine the corresponding b. Then we can obtain the separating hyperplane $f(\overline{x}) = \overline{\omega} \cdot \overline{x} + b$. For a new sample $\overline{x}_i$ with unknown label, we can get the label of it by

$$f(\overline{x}_i) = sign(\overline{\omega} \cdot \overline{x}_i + b),$$

where $sign(\cdot)$ is the sign function. If $\overline{\omega} \cdot \overline{x}_i + b > 0$, its label is $f(\overline{x}_i) = +1$; if $\overline{\omega} \cdot \overline{x}_i + b < 0$, its label is $f(\overline{x}_i) = -1$.

However, not all sample data are separable. That is, for a data set with huge amounts of data points, it is impossible for us to construct a hyperplane which can separate the data points with different labels correctly. To deal with this non-separable problem, we introduce a slack variables $\xi_i \geqslant 0$. Then, SVM attempts to find a hyperplane $f(\overline{x}) = \overline{\omega} \cdot \overline{x} + b$ such that

$$\overline{x}_i \cdot \overline{\omega} + b \geqslant 1 - \xi_i \quad y_i = 1,$$

$$\overline{x}_i \cdot \overline{\omega} + b \leqslant -1 + \xi_i \quad y_i = -1.$$

Using a similar technique as in separable case, we can find $\overline{\omega}$ and b satisfying above conditions, and then construct the corresponding separating hyperplane $f(\overline{x})$.

Another technique to deal with the non-separable problem is projecting the sample $\overline{x}$ into high dimensional space by a map $g(\overline{x})$. A non-separable problem in low dimensional space may be a separable problem in high dimensional space. However, it is difficult for us to give an exact form of $g(\overline{x})$. To solve this problem, we introduce kernel function. A kernel function is defined as $k(\overline{x}, \overline{y}) = (g(\overline{x}), g(\overline{y}))$, where $g(\overline{x})$ is a map projecting $\overline{x}$ into higher dimensional space, and (A, B) denotes the inner product of A and B. By using kernel function, we just need to know the inner product of $g(\overline{x})$ and $g(\overline{y})$ instead of knowing the exact form of $g(\overline{x})$ and $g(\overline{y})$. By replacing the inner product $\overline{x}_i \cdot \overline{x}_j$ with $k(\overline{x}_i, \overline{x}_j)$ (which is called kernel trick), the object function L_D can be rewritten as

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{ij} \alpha_i \alpha_j y_i y_j k(\overline{x}_i, \overline{x}_j). \tag{8}$$

Once we obtain solutions, α_i, which maximize L_D, we can construct the separating hyperplane and further determine the label of a new sample $\overline{x}_i$ similar to linear separable case.

The most common kernel function in SVM is the following Gauss radial basis function (and we use this kernel funtion in our experiment):

$$k(\overline{x}, \overline{y}) = e^{-\gamma \|\overline{x} - \overline{y}\|^2},$$

where γ is a parameter deciding the distribution of x around y or y around x.

5.2. Multi-class SVM

In Section 5.1, we introduce SVM for two-class case. For multi-class problem, we apply one-against-one multi-class SVM (Chang and Lin, 2011). We will use the separable problem to show how this methods works.

Table 1
The total number of each subtype of K star with different SNR.

	K1	K3	K5	K7
DATA1	1872	3249	2983	2700
DATA2	2258	3104	2957	3465
DATA3	4101	2684	3262	2795

Table 2
The number of each subtype spectra for training and test.

Subclass	DATA1		DATA2		DATA3	
	Training	Test	Training	Test	Training	Test
K1	300	300	400	400	500	500
K3	550	550	500	500	300	300
K5	500	500	500	500	400	400
K7	450	450	550	550	300	300

For training data from the ith and the jth classes, we solve the following two-class classification problem:

$$\min \frac{1}{2} \|\overline{\omega^{ij}}\|^2, \tag{9}$$

subject to

$$\begin{cases} \overline{\omega^{ij}} \cdot \overline{x}_t + b^{ij} = 1 & y_t = i, \\ \overline{\omega^{ij}} \cdot \overline{x}_t + b^{ij} = -1 & y_t = j. \end{cases} \tag{10}$$

The difference between this problem and Problem 4 is that the label y_t of x now is i and j $(i, j = 1, \ldots, n)$ instead of $+1$ and -1 in Problem 4. By solving the above problem, we can obtain a classifier to separate class i and class j. Using this way, we can obtain a total of $\frac{n(n-1)}{2}$ classifiers. Once these classifiers are constructed, we can use the following voting strategy to determine the label of a new sample $\overline{x}$: if $f(\overline{x}) = sign(\overline{\omega^{ij}} \cdot \overline{x} + b^{ij})$ says $\overline{x}$ is in the ith class, then the vote for the ith class is added by one. Otherwise, the jth is added by one. Then we predict $\overline{x}$ is in the class with the largest vote.

6. Experiment

6.1. Data description

In this section, we will present the performance of Isomap in classifying the stellar spectra. All the data in our experiment are from Sloan Digital Sky Survey (SDSS), Data Release 9.[5] The data consists of four subclasses of K-type spectra, K1-type, K3-type, K5-type and K7-type. 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.

Table 1 shows detailed information of the spectra used in experiment. Each subclass spectra is divided into three sets: the first set consists of the spectra whose signal-to-noise ratios (SNRs) are not greater than 5 (DATA1). The second set consists of the spectra with $5 < \text{SNRs} \leq 10$ (DATA2) and the third set consists of the spectra with SNRs > 10(DATA3).

In Isomap, we need to compute the distances between any two samples and try to find the shortest geodetic distances. Confined to the memory, we perform experiments 4 times to guarantee that all of the samples can be chosen one time in the 4 times for DATA3. For DATA1, we perform experiments 3 times, and for DATA2 also 3 times. Here we suppose that isoprobability of all the samples in any one of the four classes. For the four subtypes of K-star, we randomly choose 1000 K1-stars, 600 K3-stars, 800 K5-stars and 600 K7-stars (cf. Table 2) each time for DATA3, the first half of each subtype for training the SVM classifier, and the remain half for testing.

6.2. How to choose parameter

6.2.1. Parameter choose for PCA

For PCA, we need to determine the number of principal component, k, to investigate the k's effect on the performance of PCA.

[5] http://www.sdss.org/dr9/.

Table 3
The accumulative contribution rate by the first 16 principle components for DATA3.

Dimension	1	2	3	4	5	6	7	8
CR_K	0.842	0.881	0.895	0.907	0.912	0.917	0.920	0.923
	9	10	11	12	13	14	15	16
CR_K	0.926	0.928	0.930	0.932	0.933	0.934	0.935	0.936

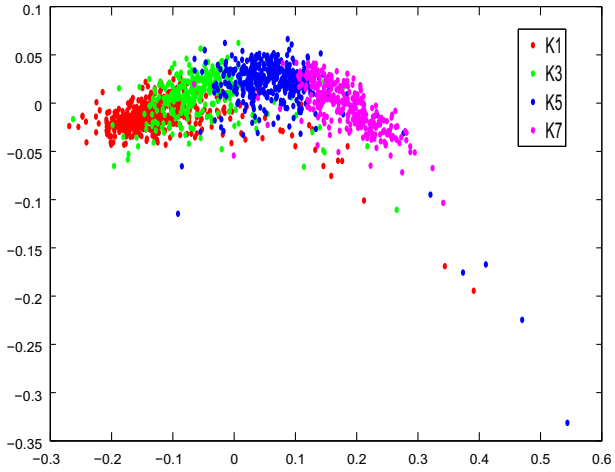

Fig. 1. Two-dimensional representation of original data (SNRs > 10) with dimension 3522 given by PCA. Ki ($i = 1,3,5,7$) represents the subtypes of K-star.

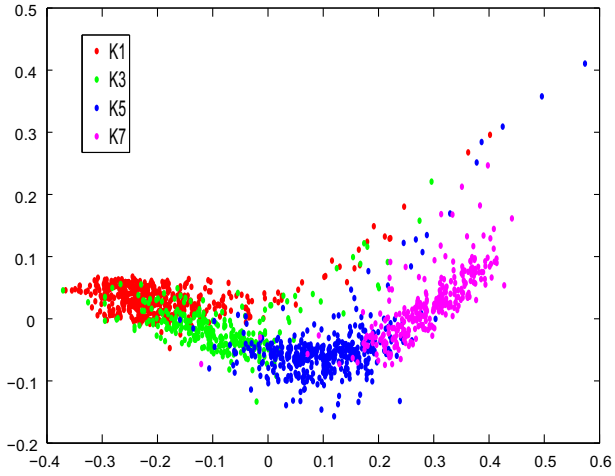

Fig. 2. Two-dimensional representation of the original data (SNRs > 10) with dimension 3522 given by Isomap ($k = 30$). Ki ($i = 1,3,5,7$) represent the subtypes of K-star.

Generally, the criterion to determine k is the accumulative contribution rate, CR_K. The larger the CR_K is, the more information we can get from the first k principal components. The CR_K for DATA3 is shown in Table 3. To ensure $CR_K \geqslant 92\%$, we will set $k = 10$. For consistency and comparison, we also set $k = 10$ for DATA1 and DATA2.

6.2.2. Parameter choose for Isomap

There are two kinds of Isomap, ε-Isomap and k-Isomap (cf. Section 4.1). In our experiment, we choose the latter one (i.e., k-Isomap) to illustrate that Isomap performs better than PCA. For Isomap, we consider different values of k to investigate its effect on the performance of Isomap. Meanwhile, we can determine the optimal k. In our experiment, the following k has been used on all three data set (i.e., DATA1, DATA2, DATA3),

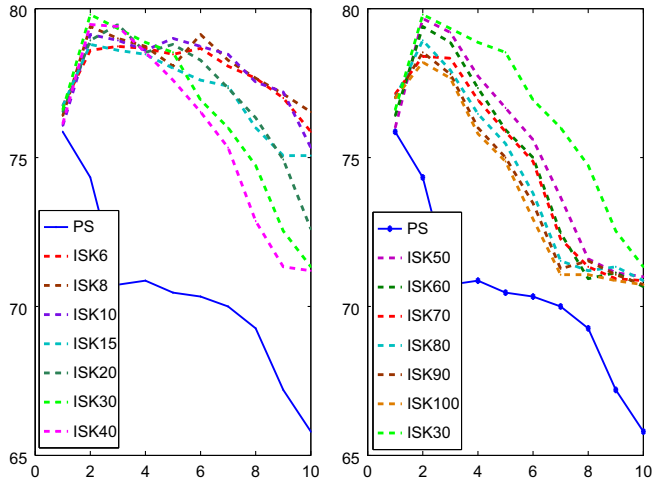

Fig. 3. In this figure, the horizontal ordinate is the dimension of data used in classification, while the vertical coordinate is the accuracy rate. The original data is projected to 1—10-dimensional space by PCA and Isomap respectively. PS means the accuracy of PCA-based SVM and 'ISKi' ($i = 6:2:10:5:20:10:100$) means the accuracy of Isomap-based SVM with $k = i$ in Isomap. ISmax means the best accuracy of IS among all the dimensions and k's. Both the performance of PCA-based SVM and that of Isomap-based SVM degradate with the increase of the reduced dimension. To illustrate that the performance of Isomap-based SVM when $k = 30$, we plot the ISK30 in both subfigures (in this figure, the SNR> 10).

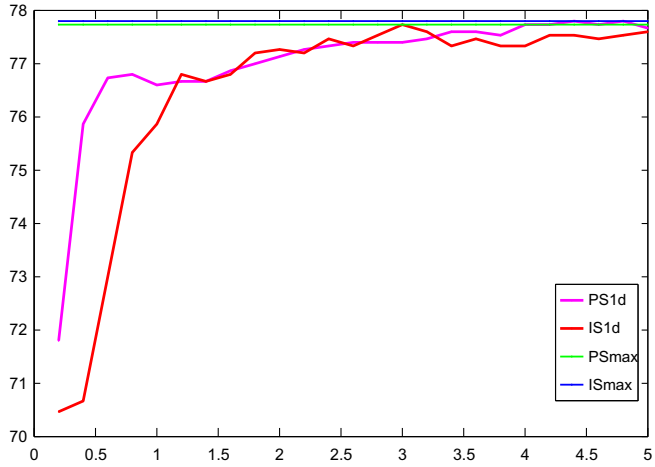

Fig. 4. In this figure, the horizontal ordinate is the value of γ ($\gamma = 0.2:0.2:5$) in SVM, while the vertical coordinate is the accuracy rate (percentage). The original data is projected to 1-dimensional space by PCA (the red line of dashes) and Isomap ($k = 30$) (the green solid line) respectively. The magenta and blue straight line are the best performance of the PCA-based SVM and Isomap-based SVM respectively, and the performance of Isomap-based SVM is slightly better than that of PCA-based SVM (in this figure, the SNR > 10). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$6, 8, 10, 15, 20, 30, 40, 50, 60, 70, 80, 90, 100.$$

Formally, we can also denote the k as $k = 6:2:10:5:20:10:100$, in which the even terms are the intervals.

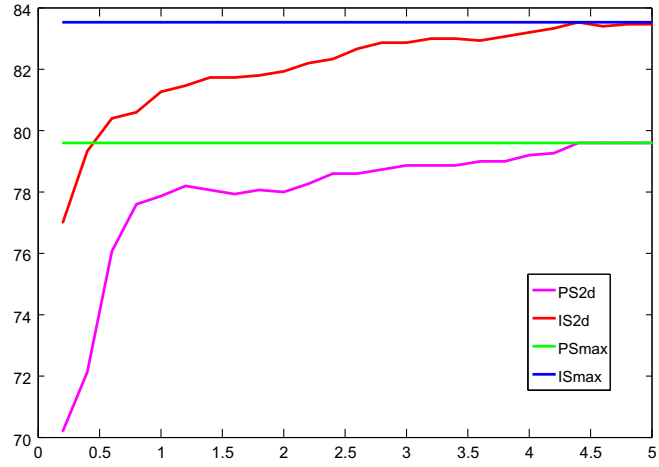

Fig. 5. In this figure, the horizontal ordinate is the value of γ ($\gamma = 0.2 : 0.2 : 5$) in SVM, while the vertical coordinate is the accuracy rate (percentage). The original data is projected to 2-dimensional space by PCA (the red line of dashes) and Isomap ($k = 30$) (the green solid line) respectively. The blue and magenta straight line show the best performance of PCA-based SVM and Isomap-based SVM respectively, and apparently, the performance of Isomap-based SVM is better than that of PCA-based SVM (in this figure, the SNR> 10). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

6.2.3. Parameter choose for SVM

In the experiment, we will first compare the performance of Isomap + SVM with PCA + SVM. The parameter γ have to be determined before applying the SVM. For simplicity, the default value of γ in SVM (Our experiment is implemented on matlab R2011b, 32bit), which is the reciprocal of the dimension, will be used. We find that other values of γ will give the similar conclusion.

Furthermore, to investigate the effect of γ varying on the performance of SVM, the γ from 0.2 to 5 with the step length of 0.2 will be used in experiment.

6.3. Experiment result and analysis

The two-dimensional representation of the original data given by PCA and Isomap are shown in Figs. 1 and 2, respectively. Both

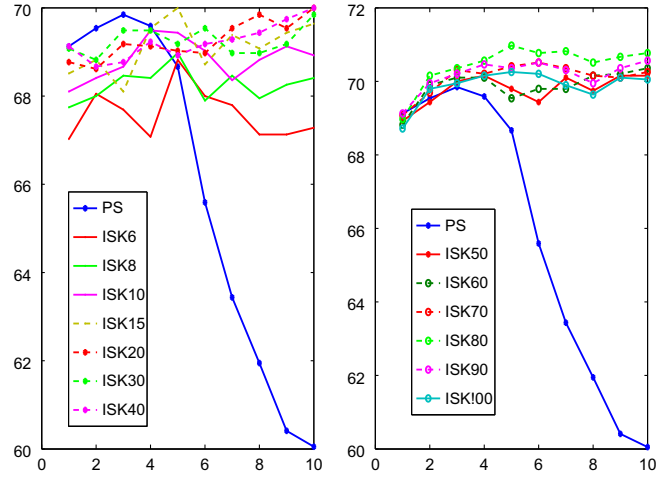

Fig. 6. In this figure, the horizontal ordinate is the dimension of data used in classification, while the vertical coordinate is the accuracy rate. The original data is projected to 1—10-dimensional space by PCA and Isomap respectively. PS means the accuracy of PCA-based SVM and 'ISKi' ($i = 6 : 2 : 10 : 5 : 20 : 10 : 100$) means the accuracy of Isomap-based SVM with $K = i$ in Isomap. The performance of PCA-based SVM degradates with the increase of the reduced dimension while that of Isomap-based SVM increases. The performance of Isomap-based SVM is the best when $k = 80$ (in this figure, the 5 <SNR≤ 10).

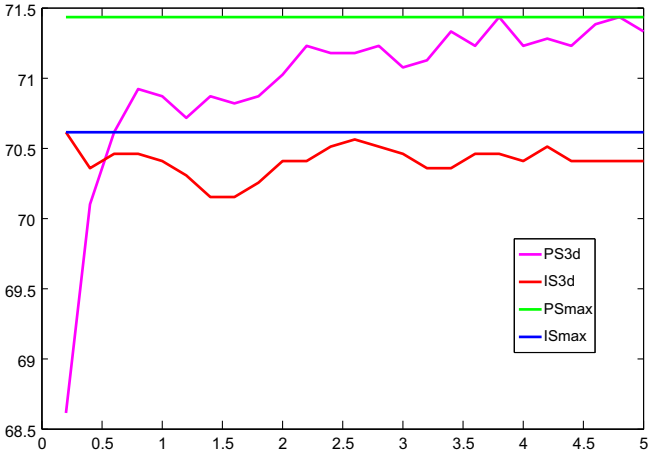

Fig. 7. In this figure, the horizontal ordinate is the value of γ ($\gamma = 0.2 : 0.2 : 5$) in SVM, while the vertical coordinate is the accuracy rate (percentage). The original data is projected to 3-dimensional space by PCA (the magenta solid line) and Isomap ($k = 80$) (the red solid line) respectively. The green and blue straight line are the best performance of the PCA-based SVM and Isomap-based SVM respectively, and the performance of PCA-based SVM is slightly better than that of Isomap-based SVM (in this figure, the 5 <SNR≤ 10). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

PCA and Isomap have mapped the spectra of different subtypes into different regions. It is not easy for us to determine which projection is better than the other. To compare two methods more accurately, we will compare the performance of Isomap + SVM with that of PCA + SVM. The classification results will then be used to determine which method (PCA or Isomap) is more accurate in dimensionality reduction and feature extraction. The classification results are shown in Figs. 3–11. The accuracy can be obtained by the following formula.

$$Accuracy = \frac{A}{N},$$

where $N = 1500$ for DATA3 ($N = 1800$ for DATA1 and $N = 2000$ for DATA2 respectively) is the total number of test sample and A is the

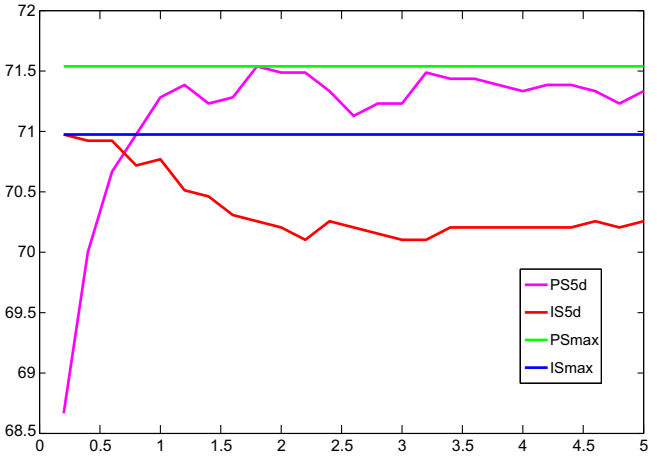

Fig. 8. In this figure, the horizontal ordinate is the value of γ ($\gamma = 0.2 : 0.2 : 5$) in SVM, while the vertical coordinate is the accuracy rate (percentage). The original data is projected to 5-dimensional space by PCA (the magenta solid line) and Isomap ($k = 80$) (the red solid line) respectively. The green and blue straight line are the best performance of the PCA-based SVM and Isomap-based SVM respectively. The performance of PCA-based SVM is slightly better than that of Isomap-based SVM (in this figure, the 5 <SNR≤ 10). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

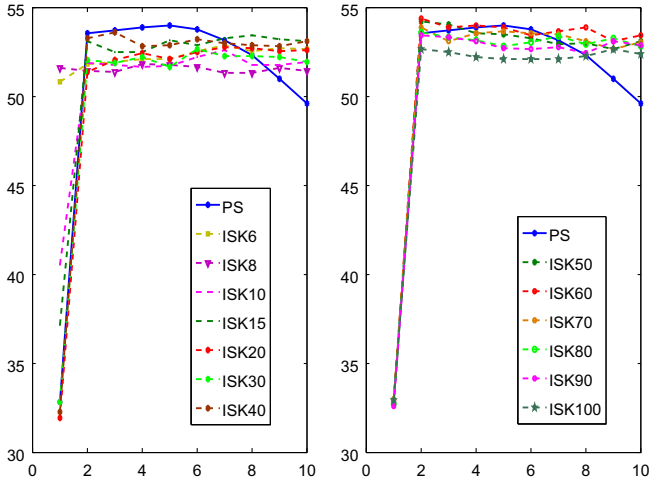

Fig. 9. In this figure, the horizontal ordinate is the dimension of the data used in classification, while the vertical coordinate is the accuracy rate. The original data is projected to 1—10-dimensional space by PCA and Isomap respectively. PS means the accuracy of PCA-based SVM and 'ISKi' ($i = 6:2:10:5:20:10:100$) means the accuracy of Isomap-based SVM with $k = i$ in Isomap. The performance of PCA-based SVM increases and then degradates with the increase of the reduced dimension while that of Isomap-based SVM oscillates slightly with the increase of the reduced dimension. The performance of Isomap-based SVM is the best when $k = 60$ and it is better than PCA-based SVM (in this figure, the SNR$\leq$ 5).

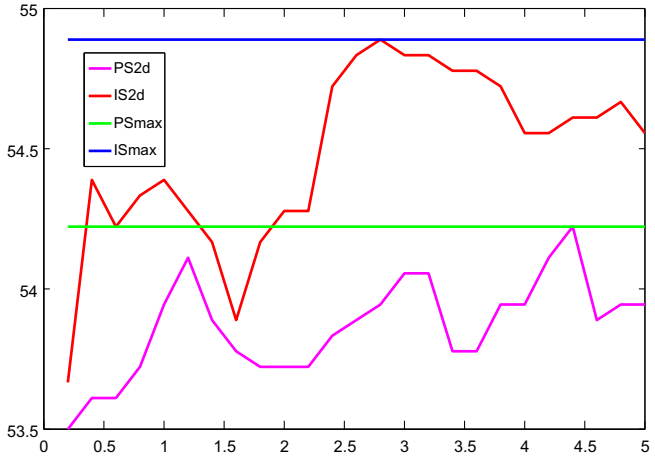

Fig. 10. In this figure, the horizontal ordinate is the value of γ ($\gamma = 0.2:0.2:5$) in SVM, while the vertical coordinate is the accuracy rate (percentage). The original data is projected to 2-dimensional space by PCA (the magenta line of dashes) and Isomap ($k = 30$) (the red solid line) respectively. The green and blue straight line are the best performance of the PCA-based SVM and Isomap-based SVM respectively. The performance of Isomap-based SVM is slightly better than that of PCA-based SVM (in this figure, the SNR$\leq$ 5). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

number of test sample classified correctly by our trained SVM classifier.

In the following, we present the result of DATA3 in detail.

Fig. 3 shows the test accuracy of PCA + SVM (PS) and Isomap + SVM (IS) with the first 10 dimensions. We can see that IS performs better than PS. It ensures the nonlinear intrinsic structure of the original data. Thus, we can apply Isomap to reduce the dimension of the original data. The result shows that the performance of PS and IS both become worse with the increase of the dimension, which suggests that the intrinsic dimension of the original data is smaller than the given dimension (3522).

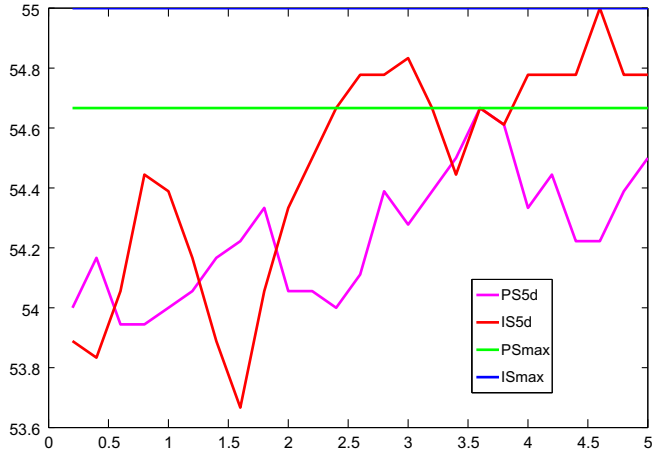

Fig. 11. In this figure, the horizontal ordinate is the value of γ ($\gamma = 0.2:0.2:5$) in SVM, while the vertical coordinate is the accuracy rate (percentage). The original data is projected to 5-dimensional space by PCA (the magenta line of dashes) and Isomap ($k = 30$) (the red solid line) respectively. The green and blue straight line are the best performance of the PCA-based SVM and Isomap-based SVM respectively. The performance of Isomap-based SVM is slightly better than that of PCA-based SVM (in this figure, the SNR$\leq$ 5). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Furthermore, we find that IS performs best on the 3 dimension data, and PS performs better on 4 dimension data. Thus, Isomap uses three dimensions to capture most of classification information in the spectra, while PCA needs four dimensions to capture the classification information in the spectra. It shows that Isomap is more efficient than PCA in capturing the classification information in the spectra.

From Fig. 3, we can see that the best performance of IS is achieved when $k = 30$ and the dimension of Isomap compressed data is 3. Thus, the optimal k for Isomap is 30. It is worth noting that, the performance of IS will not change dramatically with the varying of k. The dimension of the data used in classification will affect the classification results significantly.

To investigate the effect of γ in SVM, the γ ranges from 0.2 to 5 with fixed step length 0.2 is used in experiment. Mathematically, we can denote γ as $\gamma = 0.2:0.2:5$. The results of using different γ are shown in Fig. 4,5. We find the trend that the performance of IS improves with the increase of γ, though the performance of IS oscillates slightly instead of improving when γ is between some interval. For PS, the performance of PS will also increase with the increase of γ. This conclusion provides us with a simple method of determining the optimal γ: in the interval 0.2–5, it is better for us to use the large value of γ when classifying the stellar spectra subclasses.

The results of IS and PS on DATA1 (SNRs$\leqslant$5) are shown in Figs. 6–8, and the results of IS and PS on DATA2 (5 < SNRs$\leqslant$10) are shown in Figs. 9–11. The trend of IS and PS on DATA1 and DATA2 with increasing γ and number of dimensions used in classification is similar to that of IS and PS on DATA3. It is worth noting that the classification result of IS on DATA2 is slightly worse than that of PS on DATA2 (see Figs. 7 and 8). However, since the difference between result of IS and that of PS on DATA2 is not obvious, it will not affect the application of Isomap in spectral subclasses classification.

7. Conclusion

In this paper, we investigate the performance of Isomap + SVM in stellar subtype classification. We apply PCA, a linear dimension reduction technique, and Isomap, a nonlinear dimension reduction

technique, to reduce the dimension of the spectra. Then we apply SVM, a classifier based on the largest margin, convex quadratic programming and dual problem to classify the stars.

In Isomap, we use different k to investigate the effect of the parameter k on Isomap in reducing the dimension of spectra. Finally, we investigate the effect of γ on the performance of SVM. The result shows that the performance of SVM improves with the increase of γ under the condition that the γ increases in a specific range. And the continuation of increase of γ makes the performance of SVM oscillates slightly instead of improving. Our conclusion is as follows:

(1) Isomap-based SVM (IS) performs better than PCA-based SVM (PS) with the default γ in SVM.
(2) The optimal parameter k of Isomap in classifying the stellar subtype will increase with the increasing of the SNR of the spectra.
(3) The performance of PS and IS both change significantly with the increase of SNR.
(4) The varying of k in Isomap will not significantly affect the final classification result of IS. However, the dimension of the data used in IS will significantly affect the classification result.
(5) The variation of parameter γ in SVM will affect the final classification results. The experiment shows the trend that the classification results will increase with the increasing of γ in interval [0.2,5].

Appendix A. Dijkstra algorithm

- Nomenclature for Dijkstra algorithm
  $\mathbb{V}$: set of vertices with size n. $\mathbb{V}_1, \mathbb{V}_2, \ldots, \mathbb{V}_n$, and $\mathbb{V}_1$ is the source vertices;
  l_i: path length from vertices $\mathbb{V}_i$ to $\mathbb{V}_i$ $i = 1, 2, \ldots, n$, and $l_1 = 0$;
  d_{ij}: distance from vertices i to $j, i, j = 1, 2, \ldots, n$;
  A: set of vertices which has been found the shortest path length to the source vertices;
  N: set of vertices which has not been found the shortest path length to the source vertices;

- Step 1
  Initialization, $l_1 = 0, l_i = d_{1i},\ i = 1, 2, \ldots, n$, $\mathrm{A} = \{\mathbb{V}_1\}$, $\mathrm{N} = \{\mathbb{V}_2, \mathbb{V}_3, \ldots, \mathbb{V}_n\}$;
- Step 2
  Find a vertices $\mathbb{V}_k$ in N which satisfies
  $$l_k = \min_{j \in \mathrm{N}} \{l_j\}.$$
  Set $\mathrm{A} = \mathrm{A} \bigcup \{\mathbb{V}_k\}$, $\mathrm{N} = \mathrm{N} - \{\mathbb{V}_k\}$. If N=∅, then stop; otherwise, switch into step 3.
- Step 3
  For all vertices $j \in$ N,
  $$l_j = \min\{l_j, l_k + d_{kj}\},$$

then return to step 1.

Appendix B. Metric multidimensional scaling

For more detailed information about MDS, we refer the reader to Cox and Cox (2010). Suppose that $\bar{x}_i$ $(i = 1, \ldots, n)$ are d-dimensional data. Mathematically, metric multidimensional scaling (MDS) attempts to find n data points $\bar{y}_i$ $(i = 1, \ldots, n)$ with $\widetilde{d}$ dimension which satisfy

$$\min_{Y} \sum_{i=1}^{n} \sum_{i=1}^{n} (d(\bar{x}_i, \bar{x}_j) - d(\bar{y}_i, \bar{y}_j))^2.$$

Here $d(\bar{y}_i, \bar{y}_j))$ is the Euclidean distance between $\bar{y}_i$ and $\bar{y}_j$. In MDS the $d(\bar{x}_i, \bar{x}_j)$ is the Euclidean distance between $\bar{x}_i$ and $\bar{x}_j$, and in Isomap $d(\bar{x}_i, \bar{x}_j)$ is the geodesic distance between $\bar{x}_i$ and $\bar{x}_j$.

A solution of above problem is $d(\bar{y}_i, \bar{y}_j) = d(\bar{x}_i, \bar{x}_j)$. Since $d(\bar{x}_i, \bar{x}_j)$ $(i, j = 1, \ldots, n)$ are known distances, we will know the values of $d(\bar{y}_i, \bar{y}_j)$ for all $i, j = 1, \ldots, n$. The problem is to compute $\bar{y}_i (i = 1, \ldots, n)$ from known distances $d(\bar{y}_i, \bar{y}_j)$ $(i, j = 1, \ldots, n)$. Note that

$$d^2(\bar{y}_i, \bar{y}_j) = (\bar{y}_i - \bar{y}_j)^T (\bar{y}_i - \bar{y}_j).$$

Let $B = (B_{ij}) = (\bar{y}_i^T \bar{y}_j)$ be the inner product matrix. From $d(\bar{y}_i, \bar{y}_j)$ $(i, j = 1, \ldots, n)$ we can obtain B, and from B we can obtain $\bar{y}_i (i = 1, \ldots, n)$.

Now we show how to obtain B from $d(\bar{y}_i, \bar{y}_j)$ $(i, j = 1, \ldots, n)$. To avoid the uncertain of the solution due to arbitrary translation, we assume that $\sum_{k=1}^{d} y_i^k = 0$ $(i = 1, \ldots, n)$. Since

$$d^2(\bar{y}_i, \bar{y}_j) = y_i^T y_i + y_j^T y_j - 2 y_i^T y_j. \tag{11}$$

Then we have

$$\frac{1}{n} \sum_{i=1}^{n} d^2(\bar{y}_i, \bar{y}_j) = \frac{1}{n} \sum_{i=1}^{n} \bar{y}_i^T \bar{y}_i + \bar{y}_j^T \bar{y}_j,$$

$$\frac{1}{n} \sum_{j=1}^{n} d^2(\bar{y}_i, \bar{y}_j) = \frac{1}{n} \sum_{j=1}^{n} \bar{y}_j^T \bar{y}_j + \bar{y}_i^T \bar{y}_i,$$

$$\frac{1}{n^2} \sum_{j=1}^{n} \sum_{i=1}^{n} d^2(\bar{y}_i, \bar{y}_j) = \frac{2}{n} \sum_{j=1}^{n} \bar{y}_j^T \bar{y}_j.$$

Thus

$$B_{ij} = \bar{y}_i^T \bar{y}_j = a_{ij} - a_{i\cdot} - a_{\cdot j} + a_{\cdot\cdot}, \tag{12}$$

where $a_{ij} = -\frac{1}{2} d^2(\bar{y}_i, \bar{y}_j)$ and

$$a_{i\cdot} = \frac{1}{n} \sum_{j=1}^{n} a_{ij}, \quad a_{\cdot j} = \frac{1}{n} \sum_{i=1}^{n} a_{ij}, \quad a_{\cdot\cdot} = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}.$$

Define $A = (a_{ij})$, then

$$B = CAC = -\frac{1}{2} C D_y C. \tag{13}$$

Here $D_y = d^2(\bar{y}_i, \bar{y}_j) = d^2(\bar{x}_i, \bar{x}_j)$ are known squared distance matrix. C is the matrix $C = \{\delta_{ij} - \frac{1}{n}\}_{n \times n}$, where $\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq = j \end{cases}$. From known distances $d(\bar{y}_i, \bar{y}_j)(i, j = 1, \ldots, n)$ we can obtain the squared distance matrix D_y, and then by Eq. (13) we can obtain B.

Now we show how to obtain $\bar{y}_i$ from B. B can be expressed as

$$B = YY^T,$$

where $Y = (\bar{y}_1, \ldots, \bar{y}_n)^T$ is the $n \times \widetilde{d}$ matrix. Thus the rank of B, $r(B)$, is $\widetilde{d}$. Now B is symmetric, positive semi-definite and of rank $\widetilde{d}$, and hence has $\widetilde{d}$ non-negative eigenvalues and $n - \widetilde{d}$ zero eigenvalues. B can be decomposed into the following form

$$B = RSR^T,$$

where $S = diag\{\lambda_1, \ldots, \lambda_{\widetilde{d}}\}$ is the diagonal matrix whose elements in diagonal are the eigenvalues, λ_i, of B. $R = (\bar{v}_1, \ldots, \bar{v}_{\widetilde{d}})$ is the matrix of corresponding eigenvectors with unit norm. Set

$$Y = RS^{\frac{1}{2}}, \tag{14}$$

then

$$B = YY^T.$$

Thus, once we obtain B, we can get the matrix of eigenvalue S and matrix of eigenvectors R. By Eq. (14), we can obtain the low dimensional embedding $Y = (\bar{y}_1, \ldots, \bar{y}_n)^T$.

References

Burges, C.J., 1998. Data Min. Knowl. Discovery 2 (2), 121.
Chang, C.C., Lin, C.J., 2011. ACM Trans. Intell. Syst. Tech. (TIST) 2 (3), 27.
Cox, T.F., Cox, M.A.A., 2010. Multidimensional Scaling[M]. CRC Press.
Daniel, Scott F. et al., 2011. AJ 142, 203–212.
Jolliffe, I.T., 2002. Principal Component Analysis. Springer.
Kim, D.W., Protopapas, P., Byun, Y.I., Alcock, C., Khardon, R., Trichas, M., 2011. ApJ 735 (2), 68.
Re Fiorentin, P., Bailer-Jones, C.A., Beers, T.C., Zwitter, T., 2008. On spectral classification and astrophysical parameter estimation for galactic surveys. American Institute of Physics Conference Series, vol. 1082, pp. 76–82.
Riden, James, 2002. Unsupervised learning on Galaxy spectra. Thesis, Master of Science.
Roweis, Sam, Saul, Lawrence, 2000. Science 290 (5500), 2323.
Singh, H.P., Gulati, R.K., Gupta, R., 1998. MNRAS 295 (2), 312.
Storrie-Lombardi, M.C., Irwin, M.J., von Hippel, T., Storrie-Lombardi, L.J., 1994. Vistas in Astron. 38, 331.
Tenenbaum, J.B., De Silva, V., Langford, J.C., 2000. Science 290 (5500), 2319.
Vanderplas, J., Connolly, A., 2009. AJ 138, 1365.