Vibrational Spectroscopy 51 (2009) 276–282

Contents lists available at ScienceDirect

Vibrational Spectroscopy

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

Comparison of chemometric methods for brand classification of cigarettes by near-infrared spectroscopy

Chao Tan [a,b,*], Xin Qin [c], Menglong Li [d]

[a] Department of Chemistry and Chemical Engineering, Yibin University, Yibin, Sichuan 644007, PR China
[b] Key Laboratory of Computational Physics, Yibin University, Yibin, Sichuan, 644007, PR China
[c] China Tobacco Chuanyu Industrial Corporation, Chengdu, Sichuan 610017, PR China
[d] College of Chemistry, Sichuan University, Chengdu, Sichuan 610064, PR China

ARTICLE INFO

Article history:
Received 4 May 2009
Received in revised form 28 July 2009
Accepted 30 July 2009
Available online 8 August 2009

Keywords:
Near-infrared spectroscopy
Classification
Cigarette
Support vector machine

ABSTRACT

The combinations of NIR spectroscopy and three classification algorithms, i.e., multi-class support vector machine (BSVM), *k*-nearest neighbor (KNN) and soft independent modeling of class analogies (SIMCA), for discriminating different brands of cigarettes, were explored. The influence of the training set size on the relative performance of each algorithm was also investigated. A NIR spectral dataset involving the classification of cigarettes of three brands was used for illustration. Three performance criteria based on "*correctly classified rate* (*CCR*)", i.e., "*Average CCR*", "*95 percentile of CCR*" and "*S.D. of CCR*", were defined to compare different algorithms. It was revealed that BSVM is significantly better than KNN or SIMCA in the statistical sense, especially in cases where the training set is relatively small. The results suggest that NIR spectroscopy together with BSVM could be an alternative to traditional methods for discriminating different brands of cigarettes.

1. Introduction

In a globally competitive world, maintaining the authenticity/consistency of cigarette products is essential for the survival of a tobacco company. In fact, cigarette is a complex mixture consisting of many components, which are responsible for aroma and flavor; it is exactly the special composition that make cigarette of a brand unique among those of other brands [1]. Different brands of cigarettes differ in compositions, aroma and retail prices and as well as in the levels of potentially hazardous substances [2,3]. Even for the same brand, since the materials (tobacco leaves) from different places may have variable relative contents of different constituents, the quality may also differ from batch to batch. To make each brand of cigarette to have a consistent quality and avoid fluctuation, it is very necessary to perform a routine monitoring. Otherwise, the product will not meet specifications and expectations when delivered. Also, a consumer may feel a modified aroma for the same brand. Thus, it is important to have appropriate methods to distinguish different types of cigarettes, and therefore insure their authentication. However, due to the multi-component nature, even today, distinguishing different types of cigarettes mainly depends on human sensory responses, which are time-consuming, laborious, and subjective, and may lead to unreliable results, and so there is a need to develop alternative methods that are faster and more objective.

Ideally, a method used to distinguish cigarettes of different types should perform discrimination without sample pretreatment. In addition, it should accomplish a fast data acquisition and carry out the data treatment accurately with relatively low cost [4]. Nowadays, the combination of near-infrared (NIR) spectrometry and chemometric methods provides a powerful tool for monitoring a variety of processes and, as such, is arousing increasing interest for quality control [5]. As a fast, accurate, easy and non-destructive technique, NIR spectroscopy has become a preferred choice over traditional time and money-consuming techniques in various fields such as food [6,7], pharmaceutical [8–10], medical [11–13] and petrochemical [14–16] industries. However, the NIR spectrum is a result of overtones and combinations of fundamental vibrations of several functional groups such as C–H, N–H and O–H. This is the reason why in applications related to NIR spectroscopy, a key step is the development of a prediction model to reveal specific information [17]. Over the past decade, chemometricians have developed many valuable algorithms intended for NIR applications, among which support vector machine (SVM) is an outstanding representative [18–20]. The standard SVM is originally designed for binary classification. For multi-class SVM classification, there exist two types of solutions.

* Corresponding author at: Department of Chemistry and Chemical Engineering, Yibin University, Yibin, Sichuan 644007, PR China. Tel.: +86 831 3551080.
E-mail addresses: chaotan1112@yahoo.com.cn, chaotan1112@163.com (C. Tan).

0924-2031/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.vibspec.2009.07.004

One is by constructing and combining several binary classifiers while the other is by directly considering all classes in one single optimization formulation. Vojtěch Franc has proposed to modify slightly the original optimization formulation by adding a bias term to its objective function, which is considerably simpler than its original formulation [21]. This is the so-called BSVM. In addition, it is known that one of the very challenging works for spectroscopists is to select the most appropriate algorithm for a given task since the superiority of one algorithm over another for a task can not be generalized to another task [22], Thus, although a wide range of algorithms are available, it is still necessary to perform a careful selection and evaluation of the methods for a given task.

In the present work, for discriminating different types of cigarettes, the combinations of NIR spectroscopy and three classification algorithms, i.e., BSVM, *k*-nearest neighbor (KNN) and soft independent modeling of class analogies (SIMCA), were explored. The modeling strategies of these classification methods are substantially different. BSVM focuses on the dissimilarity between classes, whereas *k*-NN and SIMCA focus on the similarity within a class. In SIMCA modeling, common features of each class are extracted by principle component (PC) model. The SIMCA classification rule determines class membership by the orthogonal projection distance between an unknown sample and the PC model of each class. In *k*-NN modeling, the distance between an unknown sample and each sample in the training set is calculated, and the unknown sample is assigned to the class that the majority of the *k*-nearest neighbors in the training set belongs to. Three performance criteria based on the "*correctly classified rate*" (*CCR*) were used to compare different algorithms. A NIR spectral dataset involving the discrimination of three brands of cigarettes was used for illustration. Also, we investigated the influence of the training set size and several pre-preprocessing techniques on each algorithm.

2. Theory and algorithm

2.1. BSVM

The SVM [23–25] is a supervised method that has been applied to a large range of pattern recognition problems. The aim of SVM is to find an optimal hyperplane (classifier) that correctly separates objects of the different classes as much as possible. This is done by leaving the largest possible fraction of points of the same class on the same side and maximizing the distance of either class from the hyperplane. It is based on structural risk minimum mistake instead of the minimum mistake of the misclassification on the training set that SVM can effectively avoid over-fitting problem. For a two-class problem, this hyperplane is chosen as the one that maximizes the minimum distance from the hyperplane to the closest training points in both classes (margin). These points determining the hyperplane are called support vectors (SVs). The hyperplane is found by defining a specific mapping that transforms the input data into a higher dimensional feature space. A SVM model actually corresponds to solve a quadratic programming problem. However, solving such a problem is difficult due to the high dimensionality of the feature space; often, a so-called kernel function is used to calculate the separating hyperplane. This function implicitly represents the construction of an optimal hyperplane in a high dimensional space and then returns to the original space as a nonlinear decision frontier.

The standard SVM are designed for binary classification. How to effectively extend it for multi-class classification is still on-going issue. Currently, there are two types of approaches for multi-class SVM classification. One is by constructing and combining several binary classifiers such as one-against-all, one-against-one, and complete-code while the other is by directly considering all data in one single optimization formulation.

For introducing BSVM, let us first consider that we are given labeled training patterns $\{(x_i, y_i), i \in L\}$, where each $\mathbf{x}_i$ is from an n-dimensional space R^n and its class-label attains a value from a set $K = \{1, 2, \ldots, k\}$. The goal is to train a multi-class rule $q{:}X \in R^n \to K$. In the linear case, the classification rules, i.e., $f_m(\boldsymbol{x}) = \langle \boldsymbol{w}_m, \boldsymbol{x} \rangle + b_m$, can be found directly by solving the SVM problem

$$(\mathbf{w}^*, b^*, \boldsymbol{\xi}) = \underset{w,b}{\operatorname{argmin}} \underbrace{\frac{1}{2}\sum_{y \in K}(\|\mathbf{w}_y\|^2 + b_y^2) + C\sum_{i \in L}\sum_{y \in K \setminus \{y_i\}}(\xi_i)^p}$$
$$\text{s.t.} < \mathbf{w}_{y_i}, \mathbf{x} > + b_{y_i} - (< \mathbf{w}_y, \mathbf{x} > + b_y) \geq 1 - (\xi_i)^p \quad (1)$$
$$(\xi_i)^p \geq 0, i \in L, y \in K \setminus \{y_i\}$$

The minimization of the sum of norms $\|\mathbf{w}_y\|^2$ leads to maximization of the margin between classes. The slack variables ξ_i relax the constraint inequalities for non-separable cases. The weighted sum using constant C means that the cost function penalizes misclassification of training data. The linear ($p = 1$) or quadratic ($p = 2$) cost functions are often used. To employ kernel functions, one has to formulate a dual form of the multi-class SVM decision (1). However, such a dual problem has $K \times L$ variables, which is too large in practical problems and consequently it is very difficult to solve the dual quadratic problem directly. So, Franc [21] proposed (i) to modify slightly the original problem (1) by adding a bias term $(1/2)b^2$ to the objective function, i.e.,

$$(\mathbf{w}^*, b^*, \boldsymbol{\xi}) = \underset{w,b}{\operatorname{argmin}} \underbrace{\frac{1}{2}\sum_{y \in K}(\|\mathbf{w}_y\|^2 + b_y^2) + C\sum_{i \in L}\sum_{y \in K \setminus \{y_i\}}(\xi_i)^p}$$
$$\text{s.t.} < \mathbf{w}_{y_i}, \mathbf{x} > + b_{y_i} - (< \mathbf{w}_y, \mathbf{x} > + b_y) \geq 1 - (\xi_i)^p \quad (2)$$
$$(\xi_i)^p \geq 0, i \in L, y \in K \setminus \{y_i\}$$

and (ii) to transform the modified problem to the simpler single-class SVM problem, which has highly coincident solution with the original problem.

Actually, the ideas in BSVM and its modeling results are similar to the one-against-all approach; the main difference lies in the way of solving the classification rules. In brief, BSVM solves only one single optimization while one-against-all approach solves as many optimizations as the number of classes. For example, for a problem of K classes, BSVM will construct K two-class rules/functions where each function, i.e., $f_k(\mathbf{x})$, tends to separate samples of a certain class from the other samples. Thus, like the case of one-against-all approach, once obtaining the K rules, one can assign a new sample x to its category by the following strategy:

$$q(\boldsymbol{x}) = \underset{y \in K}{\operatorname{argmax}} f_y(\boldsymbol{x}) \quad (3)$$

It is exactly the introduction of a bias term $(1/2)b^2$ that the multi-class SVM problem (2) is termed as BSVM (B stands for the added bias) [26].

2.2. KNN

The *k*-nearest neighbor (KNN) [27–29] is amongst the simplest of all unsupervised classification algorithms. A sample is classified by a majority vote of its neighbors, with the sample being assigned to the class most common amongst its *k*-nearest neighbors. *k* is a positive integer, typically small. If $k = 1$, then the sample is simply assigned to the class of its nearest neighbor. In binary classification, it is helpful to choose *k* to be an odd number as this avoids tied votes. In KNN, the neighbors are taken from a set of samples for

which the correct classification is known and that can be thought of as the training set for the algorithm, though no explicit training step is required. In order to identify neighbors, all samples are represented by position vectors (i.e., spectra in this case) in an input/feature space. It is usual to use the Euclidean distance, though other distance measures.

The training phase of KNN consists only of storing the feature vectors and class labels of the training samples. In the actual classification phase, the test sample (whose class is not known) is represented as a vector in the feature space. Distances from the new vector to all stored vectors are computed and k closest samples are selected. There are a number of ways to classify the new vector to a particular class; one of the most used techniques is to predict the new vector to the most common class amongst the k-nearest neighbors. The best choice of k is of great importance and depends upon the dataset; generally, larger values of k reduce the effect of noise on the classification, but make boundaries between classes less distinct. The optimum k is selected by the validation set.

2.3. SIMCA

Soft independent modeling of class analogies (SIMCA) is a well-known and widely used supervised classification technique introduced by Wold [30,31]. Its main idea is to build a PCA model for each class belonging to a training set. Each borderline of these models is determined by multiplying the average reference sample deviation from the model with the appropriate F value (corresponding degrees of freedom and selected level of significance). Subsequently, new samples (test samples) can be fitted to these models. By comparing the residuals to the maximum allowed residuals (the borderline of the model), test samples can be classified. In this study, optimum component of PC model is determined for each class by the validation set.

2.4. Sample set partitioning

Given a dataset with fixed number of samples, the selection of a representative training set upon which training the classifiers is performed is very important. Further, a test set is necessary in order to evaluate the performance of such classifiers. In the strictest sense, the evaluation is valid only if the test set has the same distribution as the training set. Moreover, a validation set may be necessary for optimizing certain parameters. For this purpose, the classical Kennard–Stone (KS) algorithm [32,33], are used. The original KS algorithm is designed to select a limited number of training samples from the entire dataset. In this study, our first goal is to partition a dataset into three subsets, i.e., a training set, a validation and a test set, rather than only select a representative training set. So, an especial scheme, i.e., KS algorithm coupled with an alternate re-sampling, is used to realize sample set partitioning. More specifically, the KS algorithm is first used to stepwise choose samples of the same class, i.e., one by one (the expected number of samples is the same as the total number of available samples). According to the chosen order of each sample, it yields a sequence for each class. Then, an alternative re-sampling is used to pick out one sample of every two samples; this procedure is carried out for each class separately. As a result, about 1/2 of the samples are chosen from every class, and put together as the training set. Once the training set has been obtained, similarly, the remaining samples can be subdivided into a validation set and a test set again, each containing about 1/4 samples.

It should be stressed that even if the Euclidean distances may be sub-optimal in the case of spectra, the above-mentioned scheme is still reasonable, at least in terms of two reasons. Firstly, if the test set is beyond the space spanned by the training set, it instead provides an opportunity to evaluate the generalization ability of a model constructed on the training set. Secondly, the criteria used to compare different methods are based on 100 models instead of a model, therefore being statistical. More importantly, how to partition the sample set is merely responsible for a better model and has actually no effect on the comparison due to the same sets are intended for all methods.

2.5. Performance criteria

To verify and compare different classifiers, three criteria based on the *CCR* (*correctly classified rate*) were used. The *CCR* was defined as follows:

$$CCR = \frac{\sum_{i=1}^{K} \text{correctly classified samples in class } i}{\text{total number of samples}} \quad (4)$$

where k is the total number of class. Considering that modeling was repeated m times for each fixed training set size, we first defined "*Average CCR*" as follows:

$$A\,CCR = \frac{1}{m}\sum_{i=1}^{m} CCR_i \quad (5)$$

To further verify the robustness of classifiers, the other two criteria, i.e., "*95 percentile of CCR*" and "*S.D. of CCR*" are used. Hereinto, the *Average CCR* describes the average behavior of a classification algorithm, while *95 percentile CCR* describes the extremely bad behavior of a classification algorithm with 5% chance. For example, if *95 percentile CCR* is 98%, then the classification algorithm has a chance of 95% to produce a classifier with as large as 98% *CCR*. *S.D. of CCR* is the standard deviation of *CCR* able to indicate the dispersancy of an algorithm, i.e., the influence of the training set composition on the *CCR*. The detailed definition on these criteria can be found elsewhere [22].

3. Experimental

3.1. Sampling and spectra collection

All samples were collected from a cigarette factory in China. The NIR spectrum was on-line recorded in the diffuse reflectance model using the Matrix-E system (Matrix-E, Bruker, German), which was suspended exactly over the conveyer belt where shredded tobacco was passing at a speed of 3.5 cm/s. On the conveyer belt, the width, the length, the packing density and the thickness of shredded tobacco are about 1 mm, 1–2 cm, 4.8–5.0 g/cm^3 and 2 cm, respectively. Besides a Fourier transform NIR spectrometer, the system comprises special designed sampling and light-collecting modules. NIR light from the sources is focused on to the conveyor belt through a window. The distance from the window to the conveyor belt is controlled at 20 cm and the measured spot size is approximately 2.5 cm in diameter. Such a system is very suitable for on-line measurement of solid samples since it collects only diffusely scattered light and can avoid the influence of the distance fluctuation between the window and the sample to some extent. Each final NIR spectrum is the average spectrum of 64 scans over the range 4000–12,000 cm^{-1}, with a resolution of 8 cm^{-1}. Spectra were collected in the second half of 2006 and the first three months of 2008. During these two periods, on the same production line, this factory alternately produced three brands of cigarettes, i.e., Jiaozi, Wuniu, and Xiongshi, denoted as *A*, *B*, *C*, respectively, for convenience. Each day, the brand remained unchanged. So, we collected a sample at the same time every day. Since the production quantity varied from brand to brand, the numbers of spectra corresponding to different brands were unequal. As a result, depending on the practical production, a

Table 1
Results of sample set partitioning.

Sample set	Total	*A*	*B*	*C*
Training	130	29	66	35
Validation	64	14	33	17
Test	65	15	33	17

total of 259 spectra were obtained: the class *A* consisted of 58 spectra, the class *B* consisted of 132 spectra and the class *C* consisted of 69 spectra. Each spectrum was then assigned a label from 1 to 3 according to its brand (i.e., *A*: 1, *B*: 2 and *C*: 3) and each spectrum companied by its label becomes a sample.

3.2. Software and calculations

The Matrix-E system was controlled by Bruker Optics OPUS software package. All of the other calculations were performed with Matlab version 7.0 under Windows Xp, based on Pentium IV with 256 RAM. The SVM and KNN algorithms were performed by the Statistical Pattern Recognition toolbox (http://cmp.felk.cvut.cz/cmp/software/). The SIMCA algorithm was performed on LIBRA toolbox for Robust Analysis, which is available on http://wis.kuleuven.be/stat/robust.html.

3.3. Modeling and optimization

First, based on the proposed sample set partitioning, overall 259 samples were divided into a training set of 130 samples, a validation set of 64 samples and a test set of 65 samples; Table 1 summarized the partitioning results. Also, to investigate the influence of the training set size on each kind of classification models, the original training set with 130 samples were further subdivided into a series of training subsets with increasing sizes, at an increment of 5 (for simplicity, still called training set instead of training subset); that is, we considered the calibration set sizes of 5, 10, . . ., 130. Clearly, some of the training sets are larger than the test set while the others are smaller. For each size and each algorithm, random sampling was repeated 100 times to create 100 training sets, on which 100 classifiers were built and evaluated on the test set, in terms of the three performance criteria, i.e., "*Average CCR*", "*95 percentile of CCR*" and "*S.D. of CCR*".

When modeling using BSVM, the popular kernel, i.e., radical basis function (RBF) is chosen, among which, there are two parameters needed to be optimized: (i) the regularization parameter *C*, which determines the tradeoff between minimizing the training error and minimizing model complexity; (ii) parameter of the RBF kernel that implicitly defines the non-linear mapping from input space to feature space. Based on our previous attempts, the regularization constant *C* was found to have little influence on BSVM models and was therefore set as the appropriate value 10, whereas, the kernel parameter was tuned in [0.1, 0.2, 0.4, 0.8, 1.6 and 3.2]. In KNN modeling, the optimal value of *K* was searched in [1,3,5,7,9]. In SIMCA modeling, the best number of PCs was found in the range from 1 to 7. Each time, the parameter value of a classifier (BSVM, KNN or SIMCA) was always selected as the one that led to the optimal performance, i.e., the lowest *CCR*, on the validation set.

4. Results and discussion

4.1. Preliminary analysis

To observe the general spectral features, Fig. 1 shows the mean original spectra of the three classes of cigarettes (spectra corresponding to *A* and *C* are shifted upward 0.1 and 0.2 units against *B*, respectively, to prevent superposition). It is obvious that

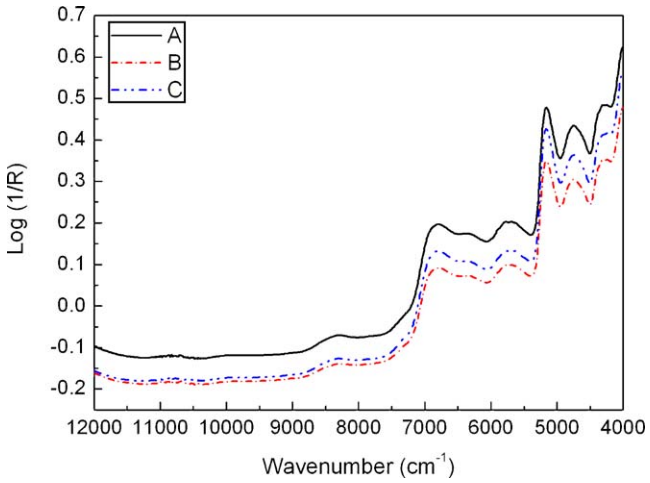

Fig. 1. Mean original spectra of the three brands of cigarettes (to prevent superposition, the spectra corresponding to *A* and *C* are shifted upward 0.1 and 0.2 units against *B*, respectively).

the trends of those spectra are very similar and all of them are subject to large baseline shifts. In general, when working with NIR spectra, an important decision is whether a pre-processing method is necessary. If so, the selection of a suitable pre-processing method is therefore another important step in the method development. Thus, several pre-processing techniques i.e., standard normal variate (SNV), multiplicative scattering correction (MSC), Savitzky–Golay first and second derivative, were employed. However, as can be seen in Fig. 2, even if different pre-processing methods modified the profile of spectra more or less, no significant contributions to the classification of cigarettes can be found, as discussed later. Why several kinds of pre-processing cannot improve the results is maybe due to scattering related to physical properties of the tobacco and need to be further investigated. Using principal component analysis (PCA), Fig. 3 gives the score plots of PCs. As can be seen from Fig. 3, even if the first two/three account for 86%/94% variance in the samples, maybe useful for distinguishing different brands of cigarettes, theses points associated to different brands still remain considerably overlapped. These evidences indicate that the classification task is not easy, meaning that a powerful modeling algorithm is absolutely necessary.

4.2. Comparison of the three algorithms

Figs. 4–6 give the comparison of the *Average CCR*, *95 percentile of CCR*, and *S.D. of CCR* of different algorithms based on 100 runs/classifiers, respectively. As shown in Fig. 4, when the training set size is smaller than 65 (i.e., the test set size), the *Average CCR* curve corresponding to each of the three types of algorithms ascends fast, and then becomes relatively flat with the increase of the training set size, suggesting that hereafter, the *Average CCR* can only be slowly improved by using more training samples. It seems to be difficult to build an accurate classifier on a smaller training set compared to the test set, which can be explained as follows: When the training set is comparatively smaller, the information responsible for constructing a model with an acceptable performance on the test set could be not sufficient. In this case, the test set perhaps spans larger space than the training set, thereby exhibiting low *CCR*. This result has some relations with the scheme of partitioning sample set in situations where the available data is limited. Also, the curve corresponding to BSVM is always above the curves corresponding to both KNN and SIMCA. It is obvious that in each case, the percentile of *CCR* value for BSVM classifier is higher than that for KNN or SIMCA, meaning that for a fixed size, BSVM

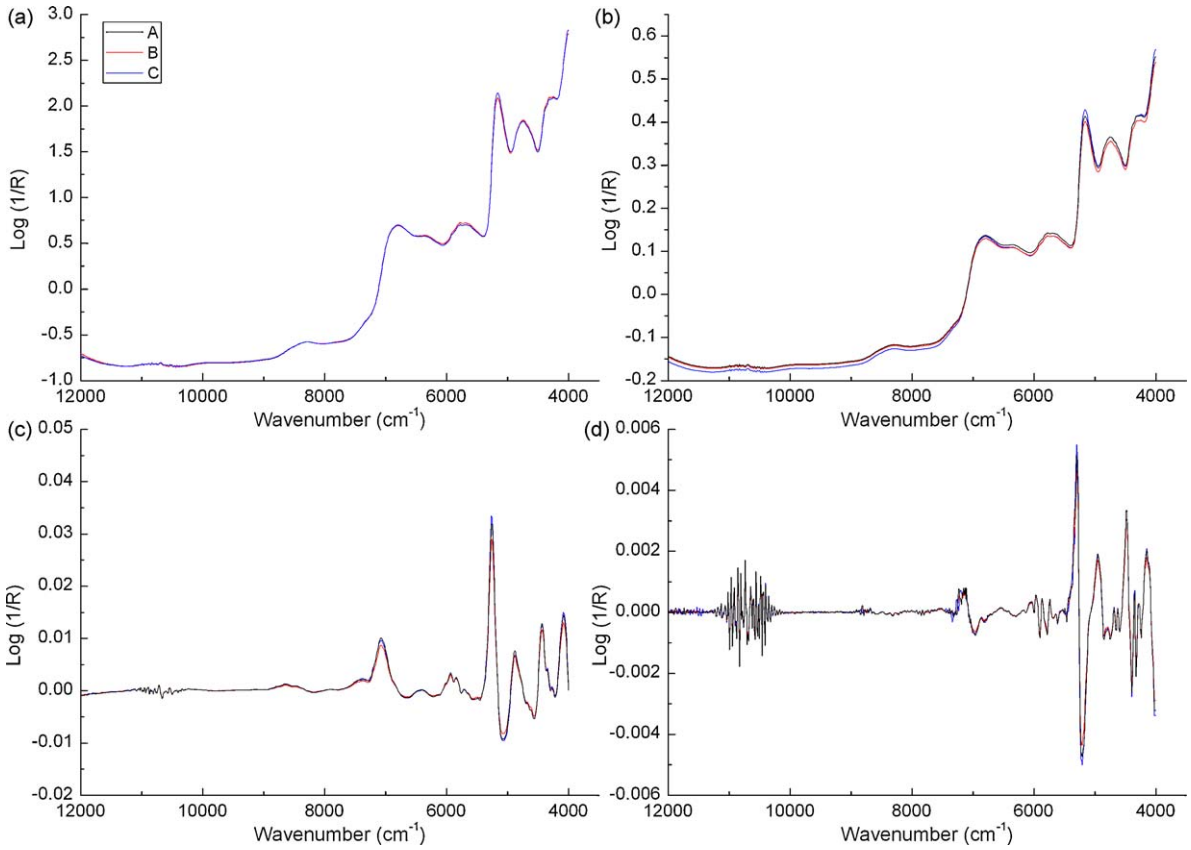

Fig. 2. Mean pre-processed spectra of the three brands of cigarettes based on four techniques; the subplots of a, b, c and d correspond to standard normal variate (SNV), multiplicative scattering correction (MSC), Savitzky–Golay first and second derivative, respectively.

can always construct a better classifier. In other words, to build a classifier with expected accuracy, BSVM need the minimum number of training samples, therefore saving the cost.

In Fig. 5, as a whole, all curves present as the similar trends as Fig. 4. Nevertheless, another interesting phenomenon can be observed; that is, the performance of BSVM classifiers can always be steadily improved while either KNN or SIMCA happen to have an unusual instance. For example, the *95 percentile of CCR* value of either the KNN classifiers with 90 training samples or the SIMCA classifier with 80 training samples achieve a local minimum, implying that increasing training samples does not guarantee the performance improvement of classifiers in both cases. So, it can be concluded that BSVM can produce more robust classifiers in terms of *95 percentile of CCR*. In practice, since one has to take the cost of collecting samples into account, the combination of *Average of CCR* and *95 percentile of CCR* provide a valuable reference on how to make a trade-off between average accuracy and robustness. From Fig. 6, it can be seen that the *S.D. of CCR* curve corresponding to BSVM is at the bottom, indicating that the dispersancy of BSVM classifiers, i.e., the influence of the training set composition on the

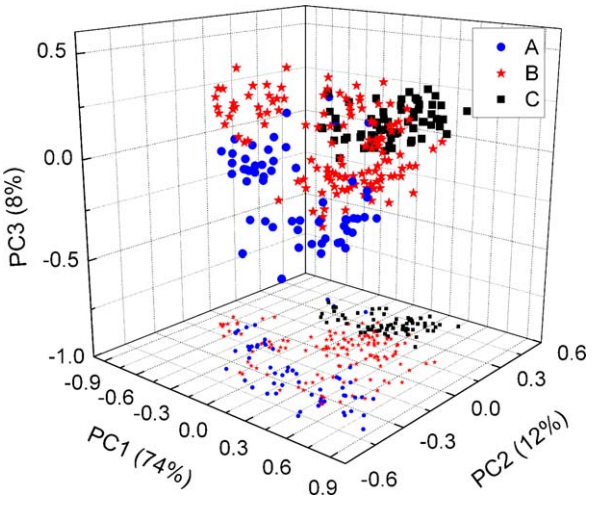

Fig. 3. Score plots of the first three/two components extracted by principal component analysis (PCA).

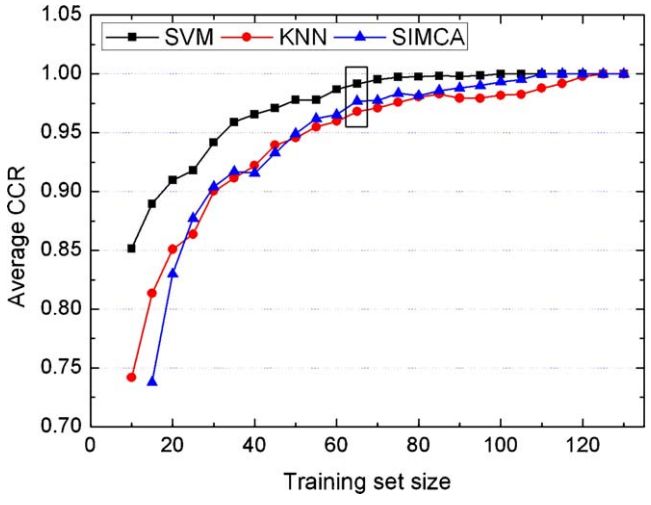

Fig. 4. Comparison of the *Average CCR* values of different classifiers with the changes of the training set size based on 100 runs.

Table 2
Comparison of BSVM, KNN and SIMCA combined with different preprocessing techniques.

Pre-processing	*Average of CCR*			*95 percentile of CCR*			*S.D. of CCR*		
	BSVM	KNN	SIMCA	BSVM	KNN	SIMCA	BSVM	KNN	SIMCA
No	99.1	96.8 (0.042)	97.7 (0.028)	96.3	92.7	94.5	0.014	0.024	0.021
SNV	99.2 (0.506)	95.5 (0.008)	96.5 (0.027)	96.0	93.3	94.1	0.014	0.025	0.022
MSC	98.5 (0.041)	96.1 (0.026)	96.6 (0.182)	95.7	91.5	94.1	0.016	0.020	0.020
1st-derivative	98.0 (0.039)	95.5 (8.1×10^{-3})	97.1 (0.078)	95.7	93.0	94.9	0.019	0.021	0.019
2st-derivative	96.3 (0.0421)	94.8 (2.3×10^{-4})	95.2 (6.9×10^{-4})	94.8	89.3	94.4	0.018	0.027	0.025

Note: The values in brackets are the *p*-values of t-test with respect to the BSVM model without preprocessing of spectra.

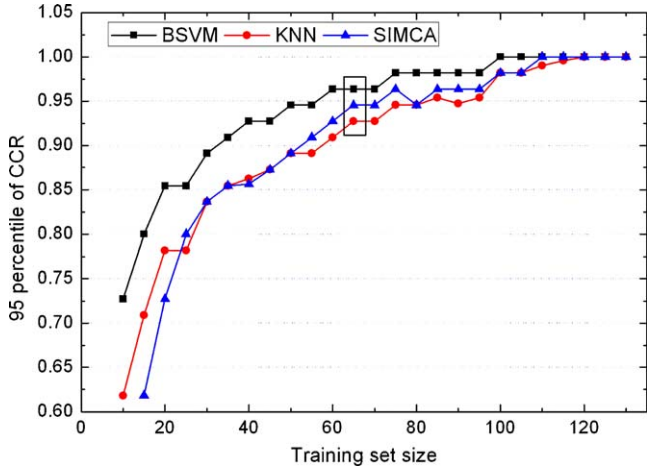

Fig. 5. Comparison of the *95 percentile of CCR* values of different classifiers with the changes of the training set size based on 100 runs.

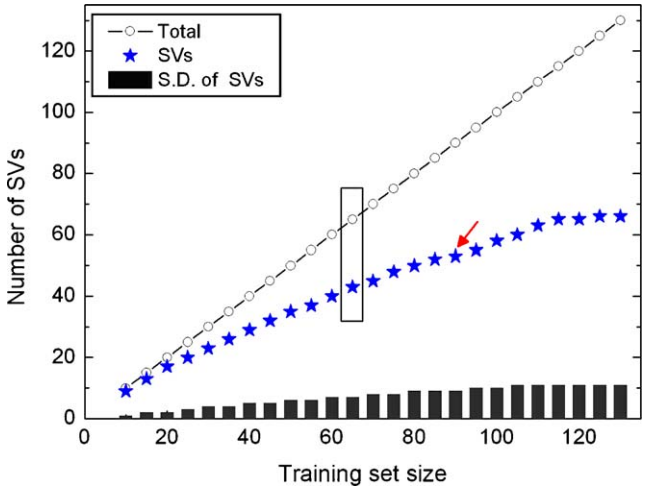

Fig. 7. The number of support vectors (SVs) corresponding to BSVM classifiers for each training set size and its standard deviation (S.D.) based on 100 runs.

CCR, is least, and thereby confirming the robustness of BSVM once again. Fig. 7 shows the number of SVs corresponding to BSVM classifiers for each size and its standard deviation. It can be noticed in Fig. 7 that with the increase of the training set size, the ratio of SVs in the training set declines gradually, while the S.D. of SVs first rises fast, then slows down, and finally remains, meaning that using too many training samples may be not necessary for building a BSVM classifier.

Fixed the training set size as 65, i.e., equal to the test set size, we took into account 15 cases based on the combination of three algorithms (BSVM, KNN and SIMCA) and four kinds pre-processing techniques as described above. Table 2 summarizes the corresponding results. In this case, it seems that the effect of pretreatments is negligible, which means that these pre-processing methods cannot improve the classifier. To further validate if the differences between BSVM and KNN/SIMICA are significant or not, based on 100 runs, *t*-test for *Average of CCR* was conducted. As shown in Table 2, in all cases except for BSVM combined with SNV, the *p* values are smaller than 0.05, the most commonly used level of significance. Even if performing a SNV pre-processing seems to produce comparable results to BSVM without any pre-processing, it is undoubtedly more advisable to build a classifier without any pre-processing.

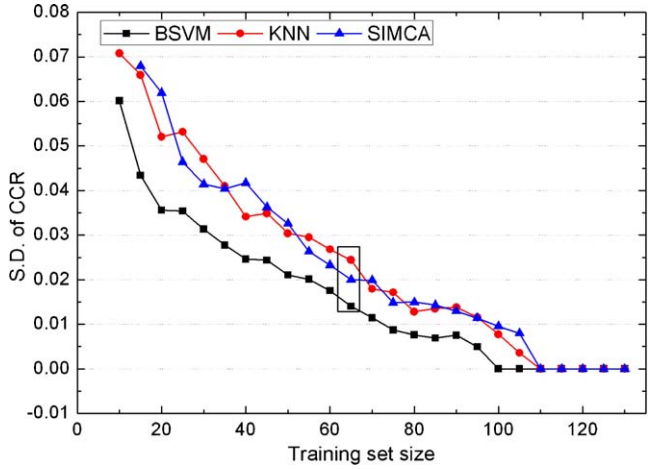

Fig. 6. Comparison of the *S.D. of CCR* values of different classifiers with the changes of the training set size based on 100 runs.

5. Conclusions

This study explored the combination of NIR spectroscopy and chemometric methods for discriminating cigarettes of different brands. The influence of the training set size and four kinds of pre-preprocessing techniques on the relative performance of each algorithm was also investigated. It was found that pre-preprocessing techniques are unnecessary since they have little impact on the classifier of each algorithm, and that the training set should be at least as big as the test set when using the proposed scheme of sample set partitioning. Also, compared to two classic algorithms, i.e., KNN and SIMCA, BSVM shows the best overall performance in all cases, suggesting that the combination of NIR spectroscopy and BSVM is a promising tool of discriminating cigarettes of different brands in tobacco industry.

Acknowledgements

This work was supported by Sichuan Province Science Foundation for Youths (09ZQ026-066) and Scientific Research Startup Fund for Doctor, Yibin University (2008B06).

References

[1] C. Tan, M.L. Li, X. Qin, Anal. Bioanal. Chem. 389 (2007) 667.
[2] J.F. Pankow, J.E. Henningfield, B.E. Garrett, Nicotin Tob. Res. 6 (2004) 199.
[3] S.D. Bolboacă, L. Jäntschi, Int. J. Environ. Res. Public Health 4 (2007) 233.
[4] M.J.C. Pontes, S.R.B. Santos, M.C.U. Araújo, L.F. Almeida, R.A.C. Lima, E.N. Gaião, U.T.C.P. Souto, Food Res. Int. 39 (2006) 182.
[5] M. Cocchi, C. Durante, G. Foca, A. Marchetti, L. Tassi, A. Ulrici, Talanta 68 (2006) 1505.
[6] X.B. Zou, J.W. Zhao, Y.X. Li, Vib. Spectrosc. 44 (2007) 220.
[7] H.Y. Cen, Y. He, Trends Food Sci. Tech. 18 (2007) 72.
[8] Y. Dou, Y. Sun, Y.Q. Ren, P. Ju, Y.L. Ren, J. Pharm. Biomed. Anal. 37 (2005) 543.
[9] J. Luypaert, D.L. Massart, Y. Vander Heyden, Talanta 72 (2007) 865.
[10] K. Awa, T. Okumura, H. Shinzawa, M. Otsuka, Y. Ozaki, Anal. Chim. Acta 691 (2008) 81.
[11] K.Z. Liu, M.H. Shi, A. Man, T.C. Dembinski, R.A. Shaw, Vib. Spectrosc. 38 (2005) 203.
[12] N. Kang, S. Kasemsumran, Y.-A. Woo, H.-J. Kim, Y. Ozaki, Chemom. Intell. Lab. Syst. 82 (2006) 90.
[13] V.R. Kondepati, M. Keese, R. Mueller, B.C. Manegold, J. Backhaus, Vib. Spectrosc. 44 (2007) 236.
[14] M.J. Kim, Y.H. Lee, C.H. Han, Comput. Chem. Eng. 24 (2000) 513.
[15] R.M. Balabin, R.Z. Safieva, Fuel 87 (2008) 1096.
[16] R.M. Balabin, R.Z. Safieva, E.I. Lomakina, Chemom. Intell. Lab. Syst. 93 (2008) 58.
[17] C. Tan, M.L. Li, Anal. Sci. 23 (2007) 201.
[18] U. Thissen, M. Pepers, B. Üstün, W.J. Melssen, L.M.C. Buydens, Chemom. Intell. Lab. Syst. 73 (2004) 169.
[19] T.T. Zou, Y. Dou, H. Mi, J.Y. Zou, Y.L. Ren, Anal. Biochem. 355 (2006) 1.
[20] Y.K. Li, X.G. Shao, W.S. Cai, Talanta 71 (2007) 217.
[21] V. Franc, V. Hlavac, 16th International Conference on Pattern Recognition, vol. 2, 2002, p. 236.
[22] J. Huang, D. Brennan, L. Sattler, J. Alderman, B. Lane, C. O' Mathuna, Chemom. Intell. Lab. Syst. 62 (2002) 25.
[23] V. Vapnik, Statistical Learning Theory, John Wiley and Sons, New York, 1998.
[24] C. Cortes, V. Vapnik, Mach. Learn. 20 (1995) 273.
[25] A.I. Belousov, S.A. Verzakov, J. von Frese, J. Chemometr. 16 (2002) 482.
[26] C.W. Hsu, C.J. Lin, IEEE Trans. Neural Networks 13 (2002) 415.
[27] M.A. Sharaf, D.L. Illman, B.R. Kowalski, Chemometrics, John Wiley and Sons, New York, 1986.
[28] K. Teknomo, http://people.revoledu.com/kardi/tutorial/knn.
[29] R. Maesschalck, D. Jouan-Rimbaud, D.L. Massart, Chemom. Intell. Lab. Syst. 50 (2000) 1.
[30] S. Wold, Pattern Recog. 8 (1976) 127.
[31] G.R. Flaten, B. Grung, O.M. Kvalheim, Chemom. Intell. Lab. Syst. 72 (2004) 101.
[32] R.W. Kennard, L.A. Stone, Technometrics 11 (1969) 137.
[33] K.R. Kanduc, J. Zupan, N. Majcen, Chemom. Intell. Lab. Syst. 65 (2003) 221.