Investigation into the use of PCA with Machine Learning for the Identification of Narcotics based on Raman Spectroscopy

Tom Howley, Michael G. Madden, Marie-Louise O'Connell and Alan G. Ryder

National University of Ireland, Galway,
thowley@vega.it.nuigalway.ie, michael.madden@nuigalway.ie,
ML.OConnell@nuigalway.ie, alan.ryder@nuigalway.ie

Abstract. The identification of narcotics using high dimensional spectral data poses an interesting challenge to machine learning, as the presence of high numbers of redundant or highly correlated attributes can seriously degrade classification accuracy. This paper investigates the use of Principal Component Analysis (PCA) to reduce spectral data and to improve the predictive performance of some well-known machine learning methods. Experiments are carried out on a high dimensional Raman spectral dataset, in which the task is to identify Acetaminophen, a pain-relieving drug, within a mixture. These experiments employ the NIPALS (Non-Linear Iterative Partial Least Squares) PCA method, a method that has been used in the field of chemometrics for spectral classification, and is a more efficient alternative than the widely used eigenvector decomposition approach. The experiments show that the use of this PCA method can improve the performance of machine learning in the classification of high dimensionsal spectral data.

1 Introduction

Automatic identification of illicit materials using Raman spectroscopy is of significant importance for law enforcement agencies. In recent years, Raman spectroscopy has enjoyed a strong resurgence in popularity due to advances in the technology that have resulted in more sensitive and lower cost instruments. The high dimensionality of spectral data can pose problems for machine learning as predictive models based on such data run the risk of overfitting. Furthermore, many of the attributes may be redundant or highly correlated, which can also lead to a degradation of prediction accuracy. Typically, methods from a field of study known as chemometrics have been applied to this particular problem [1], and these methods use PCA to handle the high dimensional spectra. PCA is a classical statistical method for transforming attributes of a dataset into a new set of uncorrelated attributes called principal components (PCs). PCA can be used to reduce the dimensionality of a dataset, while still retaining as much of the *variability* of the dataset as possible. The goal of this research is to determine if PCA can be used to improve the performance of machine learning methods in the identification of a material based on spectral data.

In the first set of experiments presented in this paper, the performance of five competitive and well-known machine learning techniques (Support Vector Machines, k-Nearest Neighbours, C4.5 Decision Tree, RIPPER and Naive Bayes) along with classification by Linear Regression are compared by testing them on a Raman spectral dataset. A number of pre-processing techniques such as normalisation and first derivative are applied to the data to determine if they can improve the classification accuracy of these methods. A second set of experiments is carried out in which PCA and machine learning (and the various pre-processing methods) are used in combination. This set of PCA experiments also facilitates a comparison of machine learning with the popular chemometric technique of Principal Component Regression (PCR), which combines PCA and Linear Regression.

The paper is organised as follows. Section 2 will give a brief description of Raman spectroscopy and outline the characteristics of the data it produces. Section 3 describes PCA and the PCR method that incorporates PCA into it. Section 4 provides a brief description of each machine learning technique used in this investigation. Experimental results along with a discussion are presented in Section 5. Section 6 describes related research and Section 7 presents the conclusion of this study.

2 Raman Spectroscopy

Raman spectroscopy is the measurement of the wavelength and intensity of light that has been scattered inelastically by a sample, known as the Raman effect [2]. This Raman scattering provides information on the vibrational motions of molecules in the sample compound, which in turn provides a chemical fingerprint. Every compound has its own unique Raman spectrum that can be used for sample identification. Each point of a spectrum represents the intensity recorded at a particular wavelength. A Raman dataset therefore has one attribute for each point on its constituent spectra. Raman spectra can be used for the identification of materials such as narcotics [1] and explosives [3].

Raman spectra are a good example of high dimensional data; a Raman spectrum is typically made up of 500-3000 data points, and many datasets may only contain 20-200 samples. However, there are other characteristics of Raman spectra that can be problematic for machine learning:

- *Collinearity:* many of the attributes (spectral data points) are highly correlated to each other which can lead to a degradation of the prediction accuracy.
- *Noise:* particularly prevalent in spectra of complex mixtures. Predictive models that are fitted to noise in a dataset will not perform well on other test datasets.
- *Fluorescence:* the presence of fluorescent materials in a sample can obscure the Raman signal and therefore make classification more difficult [1].
- *Variance of Intensity:* a wide variance in spectral intensity occurs between different sample measurements [4].

3 Principal Component Analysis

In the following description, the dataset is represented by the matrix X, where X is a $N \times p$ matrix. For spectral applications, each row of X, the p-vector x_i contains the intensities at each wavelength of the spectrum sample i. Each column, X_j contains all the observations of one attribute. PCA is used to overcome the previously mentioned problems of high-dimensionality and collinearity by reducing the number of predictor attributes. PCA transforms the set of inputs $X_1, X_2, \ldots, X_N$ into another set of column vectors $T_1, T_2, \ldots, T_N$ where the T's have property that most of the original data's information content (or most of its variance) is stored in the first few T's (the principal component scores). The idea is that this allows reduction of the data to a smaller number of dimensions, with low information loss, simply by discarding some of the principal components (PCs). Each PC is a linear combination of the original inputs and each PC is orthogonal, which therefore eliminates the problem of collinearity. This linear transformation of the matrix X is specified by a $p \times p$ matrix P so that the transformed variables T are given by:

$$T = XP \quad \text{or alternatively } X \text{ is decomposed as follows: } X = TP^T \tag{1}$$

where P is known as the *loadings matrix*. The columns loadings matrix P can be calculated as the eigenvectors of the matrix $X^T X$ [5], a calculation which can be computationally intensive when dealing with datasets of 500-3000 attributes. A much quicker alternative is the NIPALS method [6]. The NIPALS method does not calculate all the PCs at once as is done in the eigenvector approach. Instead, it iteratively calculates the first PC, then the second and continues until the required number of PCs have been generated. See Ryder [1] and O'Connell *et al.* [4] for examples of the use of PCA in the classification of materials from Raman spectra.

3.1 Principal Component Regression

The widely used chemometric technique of PCR is a two-step multivariate regression method, in which PCA of the data is carried out in the first step. In the second step, a multiple linear regression between the PC scores obtained in the PCA step and the predictor variable is carried out. In this regression step, the predictor variable is a value that is chosen to represent the presence or absence of the target in a sample, e.g. 1 for present and -1 for absent. In this way, a classification model can be built using any regression method.

4 Machine Learning

4.1 Support Vector Machine

The SVM [7] is a powerful machine learning tool that is capable of representing non-linear relationships and producing models that generalise well to unseen data. For binary classification, a linear SVM (the simplest form of SVM) finds an optimal linear separator between the two classes of data. This optimal separator is the one that results in the widest margin of separation between the two classes, as a wide margin implies that the classifier is better able to classify unseen spectra. To regulate overfitting, SVMs have a complexity parameter, C, which determines the trade-off between choosing a large-margin classifier and the amount by which misclassified samples are tolerated. A higher value of C means that more importance is attached to minimising the amount of misclassification than to finding a wide margin model. To handle non-linear data, kernels (e.g. Radial Basis Function (RBF), Polynomial or Sigmoid) are introduced to map the original data to a new feature space in which a linear separator can be found. In addition to the C parameter, each kernel may have a number of parameters associated with it. For the experiments reported here, two kernels were used: the RBF kernel, in which the kernel width, σ, can be changed, and the Linear kernel, which has no extra parameter. In general, the SVM is considered useful for handling high dimensional data.

4.2 k-Nearest Neighbours

k-Nearest Neighbours (k-NN) [8] is a learning algorithm which classifies a test sample by firstly obtaining the class of the k samples that are the closest to the test sample. The majority class of these nearest samples (or nearest single sample when k = 1) is returned as the prediction for that test sample. Various measures may be used to determine the distance between a pair of samples. In these experiments, the Euclidean distance measure was used. In practical terms, each Raman spectrum is compared to every other spectrum in the dataset. At each spectral data point, the difference in intensity between the two spectra is measured (distance). The sum of the squared distances for all the data points (full spectrum) gives a numerical measure of how close the spectra are.

4.3 C4.5

The C4.5 decision tree [9] algorithm generates a series of if-then rules that are represented as a tree structure. Each node in the tree corresponds to a test of the intensity at a particular data point of the spectrum. The result of a test at one node determines which node in the tree is checked next until finally, a leaf node is reached. Each leaf specifies the class to be returned if that leaf is reached.

4.4 RIPPER

RIPPER [10] (Repeated Incremental Pruning to Produce Error Reduction) is an inductive rule-based learner that builds a set of prepositional rules that identify classes while minimising the amount of error. The number of training examples misclassified by the rules defines the error. RIPPER was developed with the goal of handling large noisy datasets efficiently whilst also achieving good generalisation performance.

5 Experimental Results

5.1 Dataset

In the following experiments, the task is to identify acetaminophen, a pain-relieving drug that is found in many over-the-counter medications. The acetaminophen dataset comprises the Raman spectra of 217 different samples. Acetaminophen is present in 87 of the samples, the rest of the samples being made up of various pure inorganic materials. Each sample spectrum covers the range 350-2000 cm^{-1} and is made up of 1646 data points. For more details on this dataset, see O'Connell *et al.* [4].

5.2 Comparison of Machine Learning Methods

Table 1 shows the results of six different machine learning classification methods using a 10-fold cross-validation test on the acetaminophen dataset. The first column shows the average classification error achieved on the raw dataset (RD). The three remaining columns show the results of using each machine learning method in tandem with a pre-processing technique:

– ND: dataset with each sample normalised. Each sample is divided across by the maximum intensity that occurs within that sample.
– FD: a Savitzky-Golay first derivative [11], seven-point averaging algorithm is applied to the raw dataset.
– FND: a normalisation step is carried out after applying a first derivative to each sample of the raw dataset.

Table 1. Percentage Error in Identifying Presence of Acetaminophen, using various ML Methods in combination with various Pre-processing Techniques

	Pre-processing Technique			
Method	**RD**	**ND**	**FD**	**FND**
Linear SVM	**6.45** *(C=100)*	**2.76** *(C=1)*	**3.23** *(C=10000)*	**0.92*** *(C=0.1)*
RBF SVM	**5.07** *(C=1000, σ=0.1)*	**2.76** *(C=1000, σ=0.001)*	**1.84** *(C=10000, σ=10)*	**0.92*** *(C=10, σ=0.01)*
k-NN	11.06 *(k=1)*	7.83 *(k=1)*	**4.61** *(k=10)*	**4.15** *(k=1)*
C4.5	10.14	7.83	**1.84**	**1.38**
RIPPER	15.67	11.06	**3.69**	**2.3**
Naive Bayes	25.35	13.82	25.81	**5.53**
Linear Reg.	27.65	16.13	25.35	20.28

Table 1 shows the lowest average error average achieved by each classifier and pre-processing combination. For all these methods, apart from k-NN, the WEKA [8] implementation was used. The default settings were used for C4.5, RIPPER and Naive Bayes. For SVMs, RBF and Polynomial kernels with different parameter settings were tested. The parameter settings that achieved the best results are shown in parentheses. The Linear SVM was tested for the following values of C: $0.1, 1, \ldots, 10000$. The same range of C values were used for RBF SVM, and these were tested in combination with the σ values of: $0.0001, 0.001, \ldots, 10$. For k-NN, the table shows the value for k (number of neighbours) that resulted in the lowest percentage error. The k-NN method was tested for all values of k from 1 to 20. The results of each machine learning and pre-processing technique combination of Table 1 were compared using a paired t-test based on a 5% confidence level and using a corrected variance estimate [12]. The lowest average error over all results in Table 1 of 0.92% (i.e. only two misclassifications, achieved by both Linear and RBF SVM) is highlighted in bold and indicated by an asterisk. Those results which do not differ statistically from the best results (according to the t-test) are also highlighted in bold.

On both the raw (RD) and normalised (ND) dataset, both SVM models perform better than any of the other machine learning methods, as there is no significant difference between the best overall result and the SVM results on RD and ND, whereas a significant difference does exist between the best overall result and all other machine learning methods on RD and ND. This confirms the notion that SVMs are particularly suited to dealing with high dimensional spectral data and it also suggests that SVMs are capable of handling a high degree of collinearity in the data. Linear Regression, on the other hand, performs poorly with all pre-processing techniques. This poor performance can be attributed to its requirement that all the columns of the data matrix are *linearly independent* [5], a condition that is violated in highly correlated spectral data. Similarly, Naive Bayes has recorded a high average error on the RD, ND and FD data. This is presumably because of its assumption of independence of each of the attributes. It is clear from this table that the pre-processing techniques of FD and FND improve the performance of the majority of the classifiers. For SVMs, the error is numerically smaller, but not a significant improvement over the RD and ND results. Note that Linear Regression is the only method that did not achieve a result to compete with the best overall result.

Overall, the SVM appears to exhibit the best results, matching or outperforming all other methods on the raw and pre-processed data. With effective pre-processing, however, the performance of other machine learning methods can be improved so that they are close to that of the SVM.

5.3 Comparison of Machine Learning methods with PCA

As outlined in Section 3, PCA is used to alleviate problems such as high dimensionality and collinearity that are associated with spectral data. For the next set of experiments, the goal was to determine whether machine learning methods could benefit from an initial transformation of the dataset into a smaller set of PCs, as is used in PCR. The same series of cross-validation tests were run, except in this case, during each fold the PC scores of the training data were fed as inputs to the machine learning method. The procedure for the 10-fold cross-validation is as follows:

1. Carry out PCA on the training data to generate a loadings matrix.
2. Transform training data into a set of PC scores using the first P components of the loadings matrix.
3. Build a classification model based on the training PC scores data.
4. Transform the held out test fold data to PC scores using the loadings matrix generated from the training data.
5. Test classification model on the transformed test fold.
6. Repeat steps 1-5 for each iteration of the 10-fold cross-validation.

With each machine learning and pre-processing method combination, the above 10-fold cross-validation test was carried out for P=1 to 20 principal components. Therefore, 20 different 10-fold cross-validation tests were run for Naive Bayes, for example. For those classifiers that require additional parameters to be set, more tests had to be run to test the different combinations of parameters, e.g. C, σ, and P for RBF SVM. The same ranges for C, σ and k were tested as those used for the experiments of Table 1.

Table 2. Percentage Error in Identifying Presence of Acetaminophen, using various ML Methods (and Pre-processing Techniques) with PCA

	Pre-processing Technique			
Method	**RD**	**ND**	**FD**	**FND**
Linear SVM	5.07 *(P=18,C=0.1)*	**1.84** *(P=13,C=0.1)*	**3.23** *(P=14,C=0.01)*	**0.46** *(P=4,C=0.1)*
RBF SVM	6.91 *(P=19,C=100,* σ=0.001)	**2.76** *(P=16,C=10,* σ=0.001)	**2.23** *(P=12, C=10,* σ=0.001)	**0.46** *(P=5,C=10,* σ=0.001)
k-NN	11.06 *(P=17,k=3)*	5.99 *(P=10,k=1)*	**2.3** *(P=14,k=1)*	**0.0*** *(P=4,k=5)*
C4.5	7.83 *(P=20)*	7.37 *(P=19)*	7.37 *(P=5)*	**1.38** *(P=6)*
RIPPER	11.98 *(P=20)*	8.29 *(P=8)*	6.45 *(P=5)*	**2.3** *(P=3)*
Naive Bayes	38.71 *(P=1)*	10.6 *(P=8)*	11.52 *(P=5)*	3.23 *(P=2)*
PCR (PCA+Linear Reg.)	9.22 *(P=16)*	5.53 *(P=20)*	8.29 *(P=11)*	**1.38** *(P=80)*

Table 2 shows the lowest average error achieved by each combination of machine learning and pre-processing method with PCA. The number of PCs used to achieve this lowest average error is shown in parentheses, along with the additional parameter settings for the SVM and k-NN classifiers. As with Table 1, the best result over all the results of Table 2 is highlighted in bold and denoted by an asterisk, with those results that bear no significant difference from the best overall result also highlighted in bold. Again, the pre-processing method of FND improves the performance of the majority of the classifiers, Naive Bayes being the exception in this case. In comparing the best result of Table 1 with the best result of Table 2 for each machine learning method (all in the FND column), it can be seen that the addition of the PCA step results in either the same error (C4.5 and RIPPER) or a numerically smaller error (Linear SVM, RBF SVM, k-NN and Linear Regression). The improvement effected by the inclusion of this

PCA step is particularly evident with the Linear Regression technique. Note that this combination of PCA and Linear Regression is equivalent to PCR.

Despite the fact that for the SVM and k-NN classifiers, there is no significant difference between the best results with or without PCA, it is noteworthy that the SVM and k-NN classifiers with PCA were capable of achieving such low errors with far fewer attributes, only four PCs for the Linear SVM and k-NN and 5 PCs for the RBF SVM. This makes the resulting classification model much more efficient when classifying new data. In contrast, PCR required a much greater number of PCs (80) to achieve its lowest error. (This result was discovered in the experiment detailed in the next section.)

To make an overall assessment of the effect of using PCA in combination with machine learning, a statistical comparison (paired t-test with 5% confidence level) of the 28 results of Table 1 and Table 2 was carried out. This indicates that, overall, a significant improvement in the performance of machine learning methods is gained with this initial PCA step. It can therefore be concluded that the incorporation of PCA into machine learning is useful for the classification of high dimensional data.

5.4 Effect of PCA on Classification Accuracy

To further determine the effect of PCA on the performance of machine learning methods, each machine learning method (using the best parameter setting and pre-processing technique) was tested using larger numbers of PCs. Each method was tested for values of P in the range 1-640. Figure 1 shows the change in error of some of the methods versus the number of PCs retained to build the model.

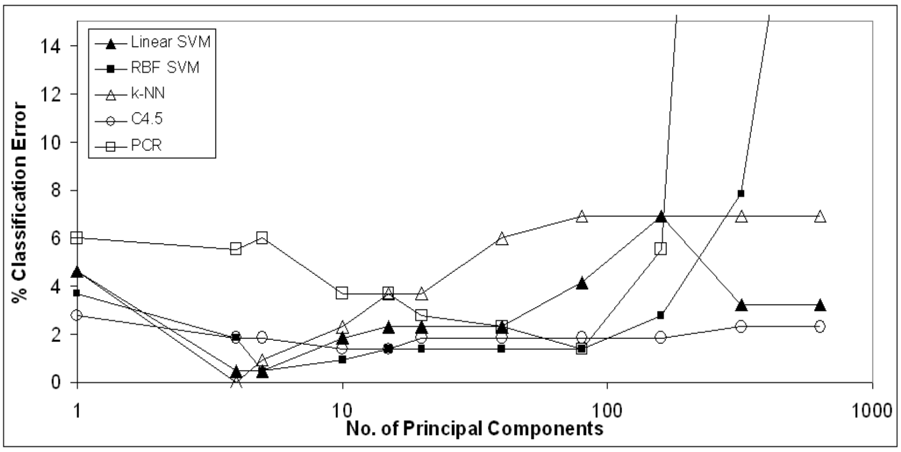

Fig. 1. Effect of changing the number of PCs on Machine Learning Classification Error

It can be seen from this graph that as PCs are added, error is initially reduced for all methods. Most methods require no more than six PCs to achieve the lowest error. After this lowest error point, the behaviour of the methods differ somewhat. Some classifiers suffer drastic increases in error within the range of PCs tested: PCR, RBF SVM, and k-NN (although not to the same extent as the previous examples). In contrast, the error for C4.5 never deviates too much from its lowest error at six PCs. This may be due to its ability to prune irrelevant attributes from the decision tree model. The Linear SVM initially seems to follow the pattern of the majority of classifiers, but then returns to a more acceptable error with the higher number of PCs. Overall, it is evident that all of the classifiers, apart from PCR, will achieve their best accuracy with a relatively small number of PCs; it is probably unnecesary to generate any more than twenty PCs. However, the number of PCs required will depend on the underlying dataset. Further experiments on more spectral data, or other examples of high dimensional data, are required to determine suitable ranges of PCs for these machine learning methods.

6 Related Research

The most closely related research to the work presented here can be found in Sigurdsson *et al.* [13], where they report on the use of neural networks for the detection of skin cancer based on Raman data that has been reduced using PCA. They perform PCA using singular value decomposition (SVD), a method which calculates *all* the eigenvectors of the data matrix, unlike the NIPALS method that was used here. In addition, they do not present any comparison with neural networks on the raw data without the PCA step.

As far as the authors are aware, few studies have been carried out that investigate the effect of using PCA with a number of machine learning algorithms. Popelinsky [14] does analyse the effect of PCA (again, eigenvector decomposition is used) on three different machine learning algorithms (Naive Bayes, C5.0 and an instance-based learner). In this paper, the principal component scores are added to the original attribute data and he has found this to result in a decrease in error rate for all methods on a significant number of the datasets. However, the experiments were not based on particularly high dimensional datasets. It is also worth noting that there does not appear to be any evidence of the use of NIPALS PCA in conjunction with machine learning for the classification of high dimensional data.

7 Conclusions

This paper has proposed the use of an efficient PCA method, NIPALS, to improve the performance of some well-known machine learning methods in the identification of materials based on high dimensional spectral data. Experiments in the classification of Raman spectra have shown that, overall, this PCA method improves the performance of machine learning when dealing with high dimensional data. Furthermore, through the use of PCA, these low errors were achieved despite a major reduction of the data; from the original 1646 attributes to at least six attributes. Additional experiments have shown that it is not necessary to generate more than twenty PCs to find an optimal set for the spectral dataset used, as the performance of the majority of classifiers degrades with increasing numbers of PCs. This fact makes NIPALS PCA particularly suited to the proposed approach, as it does not require the generation of all PCs of a data matrix, unlike the widely used eigenvector decomposition methods. This paper has also shown that the pre-processing technique of first derivative followed by normalisation improves the performance of the majority of these machine learning methods in the classification of the dataset used.

Overall, the use of NIPALS PCA in combination with machine learning and the first derivative with normalisation pre-processing technique appears to be a promising approach for the classification of high dimensional spectral data. Future work will involve using this approach for the identification of other materials based on Raman spectra, with tests also being carried out on other high dimensional datasets. This work will also investigate the automatic selection of parameters for these techniques, such as the number of PCs, kernel parameters for SVM and k for k-NN.

Acknowledgements

This research has been funded by Enterprise Ireland's Basic Research Grant Programme. The authors are also grateful to the High Performance Computing Group at NUI Galway, funded under PRTLI I and III, for providing access to HPC facilities.

References

1. Ryder, A.: Classification of narcotics in solid mixtures using Principal Component Analysis and Raman spectroscopy and chemometric methods. J. Forensic Sci **47** (2002) 275–284
2. Bulkin, B.: The Raman effect: an introduction. New York: John Wiley and Sons, Inc (1991)
3. Cheng, C., Kirkbride, T., Batchelder, D., Lacey, R., Sheldon, T.: In situ detection and identification of trace explosives by Raman microscopy. J. Forensic Sci **40** (1995) 31–37
4. O'Connell, M., Howley, T., Ryder, A., Leger, M., Madden, M.: Classification of a target analyte in solid mixtures using principal component analysis, support vector machines and Raman spectroscopy. In: Proc. SPIE - Int. Soc. Opt. Eng. Volume 5826 (in press). (2005)
5. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer (2001)
6. Geladi, P., Kowalski, B.: Partial Least Squares: A Tutorial. Analytica Chemica Acta **185** (1986) 1–17
7. Scholkopf, B., Smola, A.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press (2002)
8. Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann Publishers (2000)
9. Quinlan, R.: Learning Logical Definitions from Relations. Machine Learning **5** (1990)
10. Cohen, W.: Fast Eeffective Rule Induction. In: Proc. of the 12th Int. Conference on Machine Learning. (2002) 115–123
11. Savitzky, A., Golay, M.: Smoothing and differentiation of data by simplified least squares procedures. Anal. Chem. **36** (1964) 1627–1639
12. Nadeau, C., Bengio, Y.: Inference for generalisation error. In: Advances in Neural Information Processing 12. MIT Press (2000)
13. Sigurdsson, S., Philipsen, P., Hansen, L., Larsen, J., Gniadecka, M., Wulf, H.: Detection of Skin Cancer by Classification of Raman Spectra. IEEE Transactions on Biomedical Engineering **51** (2004)
14. Popelinsky, L.: Combining the Principal Components Method with Different Learning Algorithms. In: Proc. of ECML/PKDD IDDM Workshop (Integrating Aspects of Data Mining, Decision Support and Meta-Learning. (2001)