Available online at www.sciencedirect.com
SCIENCE DIRECT
Knowledge-Based Systems 18 (2005) 217–224
Knowledge-Based SYSTEMS
www.elsevier.com/locate/knosys

An improved genetic programming technique for the classification of Raman spectra

Kenneth Hennessy*, Michael G. Madden, Jennifer Conroy, Alan G. Ryder

Department of Information Technology and Department of Chemistry, National University of Ireland, Galway, Ireland

Received 26 October 2004; accepted 30 October 2004
Available online 11 April 2005

Abstract

The aim of this study is to evaluate the effectiveness of genetic programming relative to that of more commonly-used methods for the identification of components within mixtures of materials using Raman spectroscopy. A key contribution of the genetic programming technique proposed in this research is that it explicitly aims to optimise the certainty levels associated with discovered rules, so as to minimize the chance of misclassification of future samples.

Keywords: Machine learning; Genetic programming; Neural networks; Spectroscopy; Raman

1. Introduction

Raman spectroscopy may be described as the measurement of the intensity and wavelength of inelastically scattered light from molecules when they are excited by a monochromatic light source. The Raman scattered light occurs at wavelengths that are shifted from the incident light by the energies of molecular vibrations. The analytical applications of Raman spectroscopy continue to grow; typical applications are in structure determination [1], multi-component qualitative analysis and quantitative analysis [2].

Traditionally, multivariate data analysis techniques such as partial least squares (PLS) and principal component regression (PCR) have been used to identify the presence of specific compounds in mixtures from their Raman spectra [2]. However, Raman spectral elucidation suffers from several problems. The presence of fluorescent compounds, impurities, complex mixtures and other environmental and instrumental factors can greatly add to the difficulty in identifying compounds from their spectra [3]. Increasingly, machine learning techniques are being investigated as a possible solution to these problems, as they have been shown to be successful in conjunction with other spectroscopic techniques, such as the use of neural networks to identify bacteria from their infra-red spectra [4] and the application of neural networks to quantification of Fourier transform infra-red (FTIR) spectroscopy data [5]. Schultz et al. [6] used a neural network and PLS to identify individual components in biological mixtures from their Raman spectra, and Benjathapanum et al. [7] used PCR and neural networks to classify ultraviolet–visible spectroscopic data.

In this paper, neural networks, PLS and PCR are compared with the evolutionary technique of genetic programming for predicting which of four solvents are present in a range of mixtures. Genetic programming offers an advantage over neural networks and chemometric methods in this area as the rules generated are interpretable and may be used in isolation or in conjunction with expert opinion to classify spectra.

In combination with the environmental and instrumental problems outlined above, a significant challenge that also arises in other machine learning problems, is in the high sample dimensionality and low sample number commonly found in this area. In many real laboratory applications, it is required to identify materials based on a small number of reference spectra. While commercial spectral databases

* Corresponding author. Address: Department of Information Technology, National University of Ireland, Galway, Ireland. Tel.: +353 91 524411x2041.
E-mail addresses: hennessy@vega.it.nuigalway.ie (K. Hennessy), michael.madden@nuigalway.ie (M.G. Madden), jennifer.conroy@nuigalway.ie (J. Conroy), alan.ryder@nuigalway.ie (A.G. Ryder).

0950-7051/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2004.10.001

typically contain spectra for some thousands of materials, they are organised into categories and for individual groups of materials such as the solvents considered here, spectra would be provided for only a small number of mixtures, if any. Machine learning models exhibiting poor generalisation and overfitting to the training data are a consequence of this problem.

In response to this, rather than aiming simply to evolve equations that classify the training data correctly, our approach aims to optimise selection of equations so as to minimize the chance of misclassification of future predicted samples and thereby minimize the problems associated with low sample numbers.

Not many research groups have published applications of genetic programming for the interpretation of spectra. Goodacre [8] discusses the application genetic programming to FTIR spectroscopy image analysis. Using the same genetic programming software, Ellis et al. [9] have quantified the spoilage of meat from its FTIR spectra and Taylor et al. [10] have classified *Eubacterium* species based on their pyrolysis mass spectra.

2. Description of task

Raman spectra were recorded on a Labram Infinity (J-Y Horiba) equipped with a liquid nitrogen cooled CCD detector and a 488 nm excitation source. All spectra were recorded at a set interval of $\sim$400–3340 cm^{-1} with a resolution of $\sim$11 cm^{-1}. The liquid samples were held in 1 cm pathlength quartz cuvettes and mounted in a macro sample holder (J-Y Horiba). The macro lens has a focal length of 40 mm, which focuses through the cuvette to the centre of the liquid. The spectral data was not corrected for instrument response. Three spectra were taken for each sample, the raw data for each sample were then averaged and analysed using the Unscrambler chemometrics software package. The solvents (all spectroscopic grade), acetone, acetonitrile, cyclohexane, and toluene were obtained from Sigma-Aldrich and used as received. Solutions of different concentrations (Table 1) were made up by mixing known volumes of each solvent.

The objective is to be able to predict accurately whether or not a specific solvent is present in a mixture of other solvents. The 24 samples contain differing combinations of four solvents, acetone (A), cyclohexane (C), acetonitrile (Acn) and toluene (T), with compositions as listed in Table 1. Identification of each solvent is treated as a separate classification task. For each solvent, the dataset was divided into a training/testing set of 14 samples and a validation set of 10. The validation set in each case contained five positive and five negative samples.

There are two challenging aspects to the dataset. Firstly, as mentioned earlier, the dimensionality of the data is very high, with 1024 points per sample and the number of samples is low. Secondly, for all four solvents, the most intense peaks occur in the same region of the spectra. This may be seen in Figs. 2–4 (Section 4.1), which plot the Raman spectra of each pure solvent. The solvent mixtures detailed in Table 1 are a preliminary dataset produced for this study; the authors are currently collecting a more extensive and diverse dataset for future research.

Table 1
Chemical composition of samples used in this study (acetone (A), cyclohexane (C), acetonitrile (A_{cn}) and toulene (T))

No.	A (%)	C (%)	Acn (%)	T (%)
1	0	100	0	0
2	0	0	0	100
3	100	0	0	0
4	0	0	100	0
5	50	50	0	0
6	50	0	0	50
7	50	0	50	0
8	0	50	0	50
9	0	0	50	50
10	75	25	0	0
11	75	0	0	25
12	75	0	25	0
13	0	75	0	25
14	25	75	0	0
15	25	0	0	75
16	0	25	0	75
17	0	0	25	75
18	25	0	75	0
19	0	0	75	25
20	33	0	33	33
21	33	33	33	0
22	33	33	0	33
23	0	33	33	33
24	25	25	25	25

3. Analysis techniques

3.1. Overview

This section outlines the use of standard chemometric techniques and neural networks to identify components in mixtures from their Raman spectra. It then goes on to describe an alternative technique based on genetic programming.

As mentioned in Section 1, chemometric techniques are widely used for analysing spectra. While there are many such techniques, the two chosen in this study are PCR and PLS, as they are particularly well established for the classification of spectroscopic data [11–13]. Neural networks have been used successfully in conjunction with spectroscopic data in past research for classification purposes [4,6]. Conventional feed-forward neural networks, however, can be hard to configure for a given problem and the means by which they form predictions are not particularly easy to interpret. Hence, they are often viewed as a ‘black box’ technique.

Genetic programming is a well-known and well-documented technique in machine learning [14]. In the approach taken in this paper, we attempt to evolve

a mathematical formula through which data may be classified. A particular benefit of this technique is that it develops classification rules that are quite easily interpretable, in so far as it can be seen which wavelengths in the spectrum are used as the basis for decisions and how they are combined.

3.2. Chemometric methods

PLS and PCR were carried out on the data using the Unscrambler software (CAMO A/S). PLS and PCR are extensions of the multiple linear regression approach and they are both well suited for estimating linear regression models when the predictor variables are highly collinear [15]. Both PCR and PLS extract successive linear combinations of the predictors, called factors, so that there is no correlation between the factor score variables used in the predictive regression model. The techniques differ, however, in the methods used in extracting factor scores. PCR produces factors reflecting the covariance structure between the predictor variables, while PLS regression produces factors reflecting the covariance structure between the predictor and response variables [12].

One of the main advantages of PLS over PCR is that the factors created by PLS are directly related to the constituents of interest and may be informative. In PCR, the factors are created solely by the spectral data and represent the most common variation in the data, ignoring their relation to the constituents of interest until the final regression step [16]. In the chemometric analyses, positive values above or equal to 0.5 classified a solvent as present.

3.3. Neural networks

For this analysis, conventional feed-forward neural network structures [17] are used, and are trained using the backpropagation with momentum algorithm. Such networks consist of layers of neurons, with outputs from neurons in one layer connected to inputs in the next. Each neuron sums a set of weighted inputs and then applies a non-linear activation function (tanh, in this work) to this sum to derive an output. A separate neural network was trained for identification of each solvent. The inputs corresponded to Raman spectra of mixtures and the single output corresponded to a prediction whether or not the solvent was present in the mixture. The neural network configurations for each solvent are detailed in Table 2. These settings were found through experimentation.

Table 2
Neural network configurations

Neural network setting	Cyclohexane	Acetonitrile	Acetone	Toluene
Number of hidden nodes	15	23	20	23
Input-hidden learning rate	0.02	0.03	0.06	0.03
Hidden-output learning rate	0.001	0.008	0.001	0.008
Momentum	0.002	0.001	0.001	0.001
Epochs	1000	1000	1000	1000

3.4. Genetic programming

Genetic programming is a learning technique based on evolution. It views learning as a competition between individuals in an initially random population of potential solutions to a problem [14]. The approach attempts to find an optimal solution by breeding individuals in the population, chosen based on their fitness in partially or completely solving the problem, over a number of generations.

In this research, each individual in the population represents a mathematical formula, composed of functions and variables. The functions used are the simple mathematical operators + and −. (Others could have been used, but sufficiently good performance was achieved with just these.) The variables correspond to wavelengths in each spectrum. The population was initialised using random combinations of functions and variables to create trees with a maximum depth of five nodes. Together, the functions and wavelengths selected by an individual i form an equation E_i, which, when evaluated for a specific spectrum S_j, produces a value $E_i(S_j)$. We interpret this value as indicating the presence ($E_i(S_j) \geq 0$) or absence ($E_i(S_j) < 0$) of the corresponding solvent. Fitness is calculated based on performance in classifying the training data, and also on minimising future misclassifications, as discussed in Section 3.5.

For this analysis, we use a population of 2000 individuals. Once the fitness of each equation in the population is ascertained, the fittest ones are chosen to breed in order to create a new population. Our breeding strategy consists of elitism, crossover and mutation as illustrated schematically in (Fig. 1). Elitism involves selecting the top two fittest individuals from each population, and copying them without mutation into the next generation. Crossover is used to generate the rest of the next generation's population. This involves randomly selecting two individuals (according to a uniform distribution) from the top 1.5% of the preceding population and producing a new individual that combines features from both. The tree depth was set at a maximum of five nodes, and trees were prevented from becoming too large by addition of only the smaller of the two progeny to the population after crossover. Mutation involves random changes to an individual in the newly-generated population and is pre-configured to a fixed probability. The mutation rate in this experiment was set at 10%, i.e. one mutation in every 10 individuals. The algorithm was run until convergence, which took 50 generations.

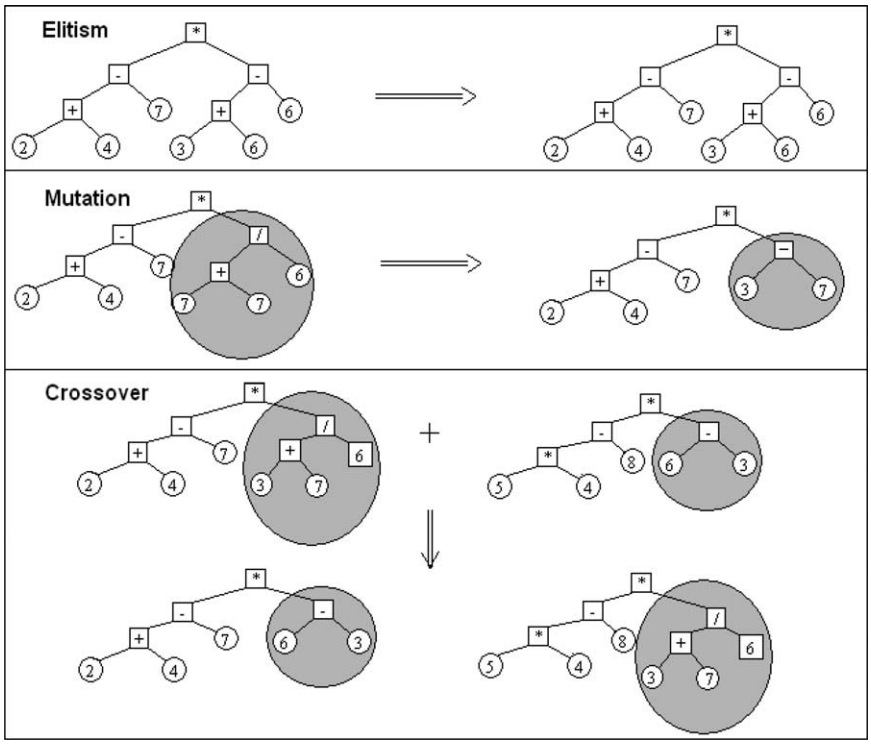

Fig. 1. Schematic representation of genetic programming operations.

3.5. Optimisation of certainty of equations

The primary goal of the individuals being evolved is, of course, to be able to classify all training data correctly. However, a secondary goal is to minimise the risk of future misclassifications. To achieve the primary goal, fitness is required to be defined in terms of classification performance on the training data. To achieve the secondary goal requires a mechanism whereby, if two individuals are equally good at classifying the data, the one with the greatest certainty is preferred. To this end, a two-level fitness function is proposed. To encourage achievement of the first goal, the first-level fitness of an individual i is calculated very simply as

$$F_1(i) = \mathrm{Acc}(i) \times N \tag{1}$$

where $\mathrm{Acc}(i)$ is the classification accuracy of the individual and N is the number of training cases. Thus, a score of 1 is given for each correctly-classified individual, and a score of 0 is given for each incorrectly-classified one. To search for individuals with high levels of certainty, a measure of certainty is first required. We define the certainty factor, $C_i(S_j)$ of an individual i relative to a spectrum S_j as:

$$C_i(S_j) = \begin{cases} 0 & \text{if } S_j \text{ is misclassified by } i \\ |E_i(S_j)| & \text{if } S_j \text{ is classified correctly by } i \end{cases} \tag{2}$$

To encourage achievement of the second goal, the second-level fitness of an individual i is calculated as:

$$F_2(i) = \min_j C_i(S_j) \tag{3}$$

In other words, it is equal to the lowest certainty factor for all spectra in the training set. Thus, if an individual does not classify all spectra correctly, its F_2 value will be 0. The overall fitness of an individual is the sum of these:

$$F(i) = F_1(i) + F_2(i) \tag{4}$$

The result of this is that F_2 does not affect fitness unless an equation correctly identifies all training set samples. Accordingly, F_2 does not affect the fitness rankings of equations that do not classify all spectra correctly, but acts as a tie-breaker for those that do, as their fitness is increased by the minimum certainty factor relative to all spectra. This encourages the evolution of equations with increasing levels of certainty, thereby reducing risk of misclassification on future spectra. Since the F_2 term arises from evaluating the equations, its order of magnitude may be completely different from that of the F_1 term. However, since individuals are selected for crossover according to a uniform distribution from the top 1.5% of the population (see Section 3.4), the fitness function is required simply to give a ranking of individuals rather than an estimate of their relative performance.

Table 3
Results of analysis techniques

Technique	Prediction (number incorrect out of 10)				
	Cyclo-hexane	Aceto-nitrile	Acetone	Toluene	Overall error rate (%)
PCR	2	0	1	0	7.5
PLS	1	0	1	0	5.0
Neural network	3	0	0	0	7.5
Naïve Bayes	0	2	1	4	17.5
Ripper	1	3	4	2	25
C4.5	0	2	4	4	25
Genetic programming	0	0	0	0	0

4. Experimental results

4.1. Comparison of techniques

The PLS, PCR, neural network and genetic programming techniques that were discussed in Section 3 have been applied to the task of predicting the presence/absence of each solvent (described in Section 2). For comparison purposes, the authors have also included results using three other popular general machine learning techniques, Naïve Bayes, Ripper and C4.5, as implemented in WEKA [18], using the default settings. For all algorithms, the same sub-divisions of the data were used for training, parameter tuning and final validation testing.

Table 3 provides details of the numbers of incorrect predictions made by each technique on the 10 validation samples for each solvent. As it indicates, PLS performed slightly better than PCR, with no incorrect classifications for cyclohexane. The neural network performed relatively badly on the cyclohexane test set but performed well on the others. In spite of large amounts of time spent exploring different configurations and parameter settings for the neural network to classify this solvent, the best result achieved was three incorrect. Naïve Bayes performed somewhat better than Ripper and C4.5 with an overall error rate of 17.5%; however, it did not perform as well as the other techniques. In contrast with this, our genetic programming technique classified all validation samples correctly with little configuration.

Figs. 2–5 show the equations chosen by the genetic programming algorithm for identification of each of the four solvents, the chemical structure of the solvents and also illustrate the position of the wavelengths chosen in each of the equations. The genetic programming equations for toluene and cyclohexane tend to focus principally on the most intense peaks of the spectra. This corresponds to the C–H region of the spectrum. Acetone has six C–H bonds in its structure and, therefore, its main peak (at 2925 cm^{-1}) coincides with significant peaks of toluene and acetonitrile. Presumably, this is why the genetic programming equations do not identify this peak as useful for discriminating between the solvents, and instead focus on points around 1700 cm^{-1}. In fact, the peak around 1700 cm^{-1} in acetone corresponds to the presence of a C=O functional group, which the other solvents do not have. Similarly, acetonitrile was classified using two points around a peak at 2255 cm^{-1}. This corresponds to the presence of a C≡N bond in acetonitrile, which is not present in any of the other solvents. This correlation between points chosen and chemical structure demonstrates the practicality of this method for use by chemists.

4.2. Effect of optimising certainty of equations

As was outlined in Section 3.5, a key aspect of our approach is that it seeks to promote the evolution of equations so as to maximise their certainty, with the intention that this should in turn minimise errors in subsequent prediction. Naturally, this assumes that increased certainty in the training set will correlate with increased certainty in the validation set. Such a correlation should be present if the two datasets are independently and identically distributed (IID). To verify this assumption empirically, Fig. 6 presents plots for each solvent separately, comparing F_2 values (minimum certainty factor for all spectra) as calculated on training data with corresponding values calculated on testing data. Each point in a graph

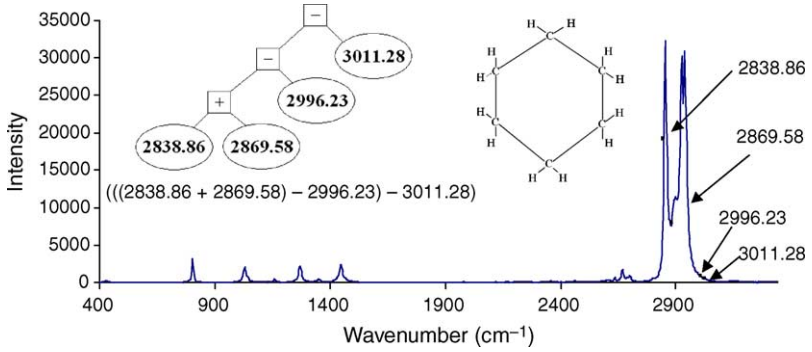

Fig. 2. Raman spectrum of pure cyclohexane sample showing data points used.

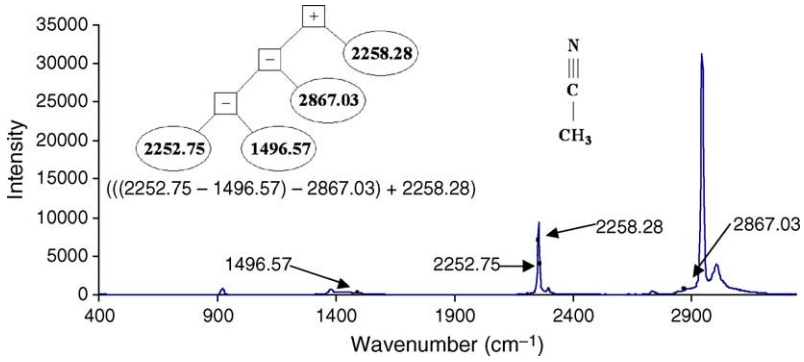

Fig. 3. Raman spectrum of pure acetonitrile sample showing data points used.

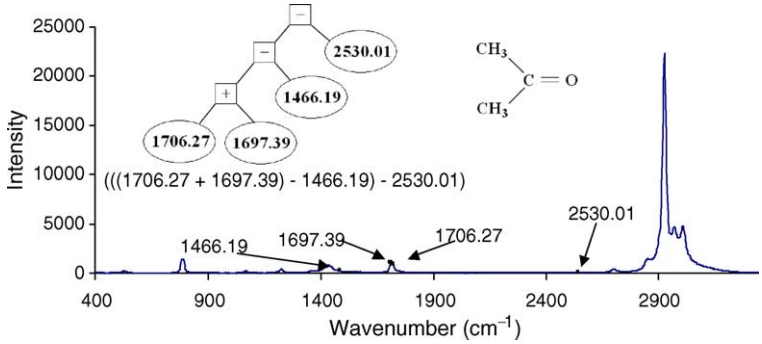

Fig. 4. Raman spectrum of pure acetone sample showing data points used.

corresponds to a single equation evolved for a solvent, all unique equations generated for each solvent are shown. The figure shows that there is reasonably good correlation between the F_2 values for the training and testing data.

It is worthwhile to note that, in Fig. 6, the equations plotted all have F_2 values for the training data that are greater than 0; in other words, they all classify all training samples correctly. On the other hand, several have F_2 values for the testing data that are equal to 0, indicating that they do not classify all test samples correctly. This highlights the need to select equations that not only classify the training data correctly, but that do so with as high certainty as possible. The graphs indicate that the strategy of maximising certainty on the training data generally results in improving performance on unseen data, though there are some exceptions.

It should be noted that the axes have different scales for the different solvents. The equations predicting acetone have significantly lower certainty levels than those of the other solvents, with the best acetone figure having a certainty level of just 208, compared to values of over 2000 for the other solvents. Accordingly, we expect the acetone predictions to be the most vulnerable to misclassification of future test samples.

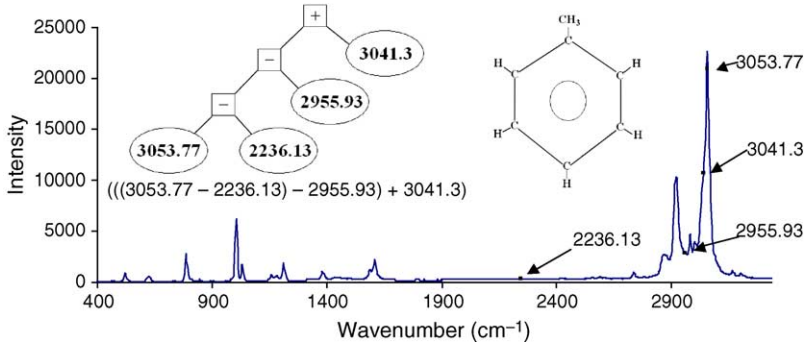

Fig. 5. Raman spectrum of pure toluene sample showing the data points used.

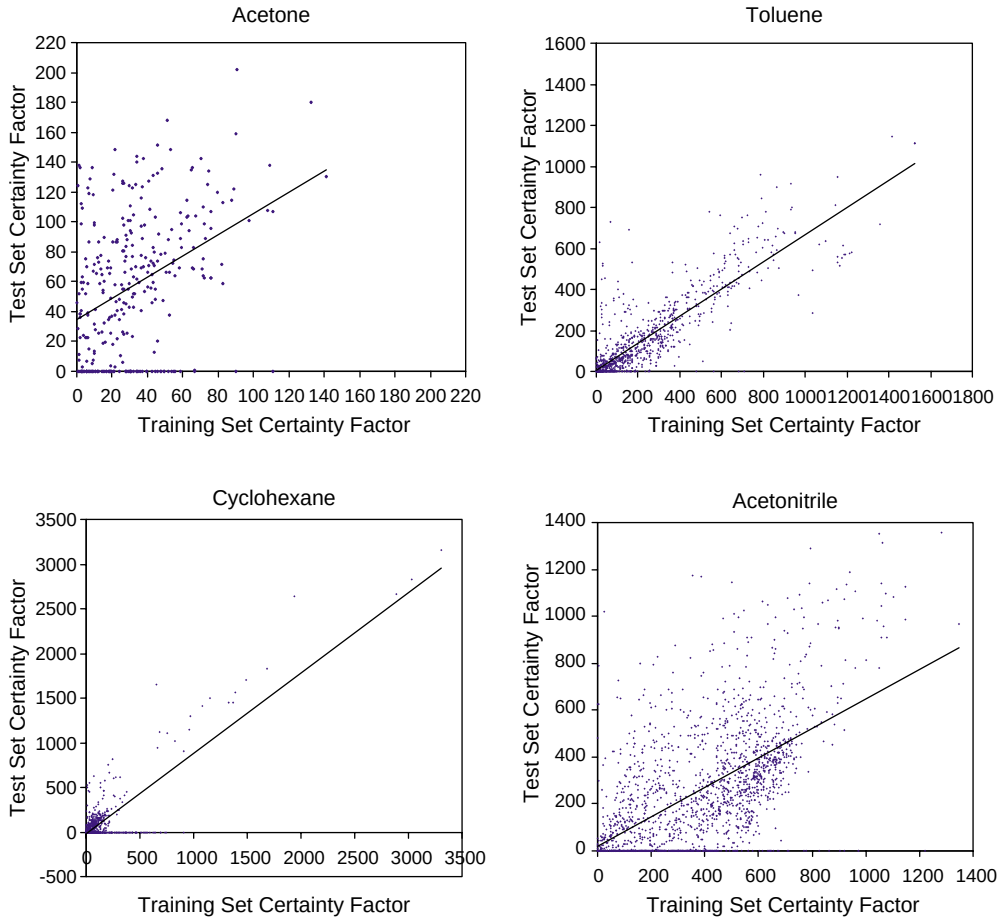

Fig. 6. Comparisons of minimum certainty factors on training data relative to those of validation testing data.

5. Conclusions and future work

This paper has described the value of genetic programming for Raman spectral classification and has introduced an improved fitness function to reduce the risk of misclassification of future samples. Genetic programming identified all solvent samples correctly with little configuration and the equations generated provide an insight into how decisions are made which offers an advantage over other techniques such as PLS, PCR and neural networks. This is very important in 'real world' practical applications of machine learning techniques, as troubleshooting misclassifications by 'black box' techniques is difficult. The evolved rules allow for decisions to be made which take both human and machine opinion into account and are informative when viewed in conjunction with the chemical structure of the compound whose presence is being investigated.

The genetic programming technique developed in this research optimises the assurance levels associated with discovered rules, so as to reduce the likelihood of misclassification of future samples. This is useful in areas where the number of training samples is low. Future investigations in this area include the use of genetic programming in the prediction of chemical concentration from Raman spectra. We also propose in the future, to refine our fitness equations so that they can incorporate different misclassification costs for false positives and false negatives, as in some forensic applications one type of error is more serious than the other.

Acknowledgements

This work was supported by funding from Enterprise Ireland's Commercialisation Fund Technology Development Programme (TD/03/212) and by the National Centre for Biomedical Engineering Science as part of the Higher Education Authority Programme for Research in Third Level Institutions.

References

[1] J.R. Ferraro, K. Nakamoto, C.W. Brown, Introductory Raman Spectroscopy, second ed., Academic Press, San Diego, 2003.
[2] J.M. Shaver, Chemometrics for Raman Spectroscopy, Handbook of Raman Spectroscopy: Practical Spectroscopy Series, vol. 28, Marcel Dekker, New York, 2001.

[3] B.J. Bulkin, The Raman Effect: An Introduction, Analytical Raman Spectroscopy, Chemical Analysis, vol. 114, Wiley, New York, 1991. pp. 1–19.
[4] R. Goodacre, E.M. Timmins, P.J. Rooney, et al., Rapid identification of *Streptococcus* and *Enterococcus* species using diffuse reflectance-absorbance Fourier transform infrared spectroscopy and artificial neural networks, FEMS Microbiol. Lett. 140 (1996) 233–239.
[5] H. Yang, P.R. Griffiths, J.D. Tate, Comparison of partial least squares regression and multi-layer neural networks for quantification of nonlinear systems and application to gas phase Fourier transform infrared spectra, Anal. Chim. Acta 489 (2003) 125–136.
[6] H.G. Schulze, L.S. Greek, B.B. Gorzalka, et al., Artificial neural network and classical least squares method for neurotransmitter mixture analysis, J. Neurosci. Meth. 56 (1995) 155–167.
[7] N. Benjathapanun, W.J.O. Boyle, K.T.V. Grattan, Classification of UV–Vis spectroscopic data using principal component analysis and neural network techniques, Measurement 24 (1998) 1–7.
[8] R. Goodacre, Explanatory analysis of Spectroscopic data using machine learning of simple, interpretable rules, Vib. Spectrosc. 32 (2003) 33–45.
[9] D.I. Ellis, D. Broadhurst, D.B. Kell, et al., Rapid and quantitative detection of the microbial spoilage of meat by Fourier transform infrared spectroscopy and machine learning, Appl. Environ. Microbiol. 68 (2002) 2822–2828.
[10] J. Taylor, R. Goodacre, W.G. Wade, J.J. Rowland, D.B. Kell, The deconvolution of pyrolysis mass spectra using genetic programming: application to identification of some *Eubacterium* species, FEMS Microbiol. Lett. 160 (1998) 237–246.
[11] A.G. Ryder, G.M. O'Connor, T.J. Glynn, Quantitative analysis of cocaine in solid mixtures using Raman spectroscopy and chemometric methods, J. Raman Spectrosc. 31 (2000) 221–227.
[12] F. Estienne, D.L. Massart, Multivariate calibration with Raman data using fast principal component regression and partial least squares methods, Anal. Chim. Acta 450 (2001) 123–129.
[13] M.G. Madden, A.G. Ryder, Machine learning methods for quantitative analysis of Raman spectroscopy data, Proceedings of SPIE, the International Society for Optical Engineering 4876 (2003) 1130–1139.
[14] J.R. Koza, F.H. Bennett III, D. Andre, et al., Genetic Programming III, Morgan Kaufmann, San Francisco, 1999.
[15] StatSoft, Inc. Electronic Statistics Textbook, http://www.statsoft.com/textbook/stathome.html (2004).
[16] M. Berthold, D.J. Hand, Intelligent Data Analysis, Springer, Berlin, 2003.
[17] P. Brierley, Visual Neural Data Mining, 2004http://www.philbrierley.com.
[18] I.H. Witten, F. Eibe, Data Mining: Practical Machine Learning Tools with Java Implementations, Morgan Kaufmann, San Francisco, 2000.