FULL PAPER
THE ANALYST
www.rsc.org/analyst

Total luminescence spectroscopy with pattern recognition for classification of edible oils

Simon M. Scott,[a] David James,[a] Zulfiqur Ali,*[a] William T. O'Hare[a] and Fred. J. Rowell[b]

[a] *School of Science and Technology, University of Teesside, Middlesbrough, UK TS1 3BA. E-mail: z.ali@tees.ac.uk*

[b] *School of Health, Natural and Social Sciences, University of Sunderland, Sunderland, UK SR1 3SD*

Received 17th March 2003, Accepted 23rd April 2003
First published as an Advance Article on the web 22nd May 2003

Total luminescence spectroscopy combined with pattern recognition has been used to discriminate between four different types of edible oils, extra virgin olive (EVO), non-virgin olive (NVO), sunflower (SF) and rapeseed (RS) oils. Simplified fuzzy adaptive resonance theory mapping (SFAM), traditional back propagation (BP) and radial basis function (RBF) neural networks provided 100% classification for 120 samples, SFAM was found to be the most efficient. The investigation was extended to the adulteration of percentage v/v SF or RS in EVO at levels from 5% to 90% creating a total of 480 samples. SFAM was found to be more accurate than RBF and BP for classification of adulterant level. All misclassifications for SFAM occurred at the 5% v/v level resulting in a total of 99.375% correctly classified oil samples. The percentage of adulteration may be described by either RBF network (2.435% RMSE) or a simple Euclidean distance relationship of the principal component analysis (PCA) scores (2.977% RMSE) for v/v RS in EVO adulteration.

Introduction

Total luminescence spectroscopy (TLS), commonly used as excitation–emission matrices (EEM), measures fluorescence intensity as a function of both excitation and emission. The excitation wavelength is varied in this method, producing more spectral information than conventional fluorescence spectroscopy. A specific compound, will have a unique matrix (dependent upon external factors), therefore this intensity matrix may be used like a fingerprint to identify fluorescent compounds. Pattern recognition applied to these 'fingerprints' allows for unknown samples to be identified with a specific degree of confidence.

Olive oil adulteration detection has been reported using mass spectrometry (MS),[1,2] nuclear mass resonance (NMR),[3] gas chromatography (GC)[4] and high performance liquid chromatography (HPLC).[5,6] These methods require skilled operators and (or) chemical modification of the samples.

In a previous investigation discrimination between fresh edible oils using a piezoelectric quartz crystal sensor array was possible.[7] However, this technique involved heating of the samples in order to generate sufficient headspace volatiles to be evaluated. Fluorescence spectroscopy is an attractive technique as it provides high dimensional information and is also non-invasive.

TLS has been successfully utilised for many varied applications. Shimoyama and co-workers[8] determined the plant dyestuffs used in traditional Japanese woodblock prints by means of TLS using a bifurcated quartz fibre optic cable to focus and receive light onto and collect from the sample. In that application the non-invasive nature of the technique was important to avoid damage of the ancient coloured cloth. Alexander *et al.*[9] successfully investigated discrimination of different gasolines by fluorescence. Patra and Mishra[10,11] carried out discrimination of petroleum fuels and the effect of adulteration to the fuel samples. Baker[12] monitored water quality using TLS for the detection of farm wastes.

Artificial neural networks (ANN) have been applied to synchronous fluorescence spectra. Li *et al.*[13] applied both BP and RBF networks for spectrofluorimetry of multicomponent mixtures. For their work they reduced the dimensionality of the data by PCA, using the first seven components as input into a BP network and concluded ANNs were a feasible strategy for working with fluorescence data.

In this study TLS has been applied to the classification of olive and other seed oils, including, rapeseed and sunflower. PCA, BP, RBF and SFAM networks were used for pattern recognition. It is shown that SFAM is an effective pattern recognition algorithm for total luminescence data and that the Euclidean distance of the PC scores may be used to accurately quantify adulteration levels.

Computational analysis techniques

Principal component analysis

PCA is a commonly used multivariate technique which acts unsupervised.[14] PCA finds an alternative set of axes about which a data set may be represented. It indicates along which axis there is the most variation; axes are orthogonal to one another. PCA is designed to provide the best possible view of variability in the independent variables of a multivariate data set. When the principal component scores are plotted they may reveal natural patterns and clustering in the data samples.

Artificial neural networks

Artificial neural networks are becoming increasingly popular due to their capability in statistical analysis and data modelling. The three network types discussed here are all supervised, that is, the input pattern is presented to the net and the response is compared to the target output, corrective action on the network

DOI: 10.1039/b303009a

This journal is © The Royal Society of Chemistry 2003

to enforce the correct output is taken if necessary. In this way the network is trained to learn the information it is given. The back propagation algorithm is the most common, the radial basis function network is a good alternative in classification and regression problems. The ART family of networks have been shown to be very efficient for mapping multi-dimensional input to output data.

Back propagation network

The back propagation algorithm is perhaps the most widely used supervised training algorithm for multilayered feed forward networks. In training an iterative gradient algorithm designed to minimize the mean square error between the actual output of a multilayer feed forward perceptron and the desired output is used. A feed forward phase is first performed on an input pattern to calculate the net error, then, the algorithm uses this computed output error to change the weight values in the backward direction. The error is slowly propagated backwards through the hidden layers. The actual derivations for the different formulae used in the back propagation algorithm come from the generalized delta rule. The delta rule is based on the idea of the error surface. The error surface represents cumulative error over a data set as a function of the network weights. Each possible network weight configuration is represented by a point on this error surface. The partial derivative of the network error with respect to each weight gives information about the direction the error of the network is moving. If the negative of this derivative is taken (*i.e.* the rate change of the error as the value of the weight increases) and then added to the weight, the error will decrease until it reaches a local minimum. The taking of these partial derivatives and then applying them to each of the weights takes place, starting from the output layer to hidden layer weights, then, from the hidden layer to input layer weights, backwards through the network.

Radial basis function network

Radial basis neural networks were popularised by Broomhead and Lowe[15] in the late 1980's, they are quick to train and conceptually elegant. The standard back propagation networks suffer from some serious drawbacks such as slow convergence in the learning phase, the potential convergence to a local minimum, common chaotic behaviour, and the inability to detect over-fitting. Radial basis function networks are a different type of multilayer network, the output units form a linear combination of the basis functions in the hidden or kernel layer. The basis functions produce a localised response to the input. A basis function may be viewed as an activation function that produces a localised response to the input vector. RBF networks may overcome some of the limitations of back propagation by relying on a rapid training phase, avoiding chaotic behaviour, having a simpler architecture whilst keeping a complicated mapping capability. Such characteristics coupled with an intrinsic simplicity make the RBF network an interesting alternative for pattern recognition.

RBF networks are also well suited to function approximation. Each hidden node is trained to contribute a Gaussian based measurement that is then weighted with the others to produce the output. It is possible to train the centre, width and weighting of each Gaussian to smoothly cover the region of pattern space populated by the training data. If small widths are used then it may take a large number of nodes to adequately cover the region, conversely if few nodes with large widths are used there is a risk that not enough detail will be supplied in the structure of the decision region. A balance of the number of nodes and the accuracy of fit (without over fitting and losing the ability to generalise) needs to be met.

Simplified fuzzy adaptive resonance theory mapping

Carpenter and Grossberg[16] developed the adaptive resonance theory (ART) family of neural networks to solve some of the problems that other neural networks suffer from. The aim was to have a stable memory structure even with fast on-line learning that was capable of adapting to new data input, even forming totally new category distinctions. Fuzzy ARTMAP is a specialisation of the general ART case, developed for supervised slow learning, unlike parametric probability estimators fuzzy ARTMAP does not depend on *a priori* assumptions about the underlying data. Online computation is able to achieve probability estimates and compression by partitioning the input space into categories. Recognition categories large or small are produced to output best predictions. The network has a small number of parameters and does not require guesswork to determine the initial configuration since the network is self-organising. In a standard back propagation network used for pattern classification an output node is assigned to every class of object that the network is expected to learn. In fuzzy ARTMAP the assignment of output nodes to categories is dynamically assessed by the network. Unlike traditional back propagation neural networks the architecture of fuzzy ARTMAP is self organising, Carpenter and Grossberg refer to this phenomenon as the plasticity–stability dilemma, how a network may retain learned patterns (stable) while remaining able to learn new ones (plastic). Kohonen's self-organising network uses a gradually reducing learning rate; this however simply limits the plastic period of the net, the Kohonen network acts unsupervised and is more appropriate for exploratory data analysis and visualisation. A further problem in neural network computing is one of generalisation; to fix the number of nodes required to describe the pattern space. If a large number of nodes are used then a finely graded solution will be obtained but the possibility of over-training will occur and computation times will increase, too few nodes and the granularity will be too coarse resulting in imprecise calculation. In fuzzy ARTMAP the network is allowed to organise itself in this respect so that the number of nodes produced results in the appropriate accuracy required according to the 'vigilance' parameter. Simplified fuzzy adaptive resonance theory mapping is a simplified form of fuzzy ARTMAP developed by Kasuba.[17] A block diagram of the SFAM network showing the main architecture is shown in Fig. 1.

Input into SFAM

Input into the network must be normalised to a value from 0 to 1, hence a suitable normalisation value must be chosen so that no input will fall outside of the valid range. A compliment coder normalises the input and also provides the fuzzy compliment for each value. This expanded input (*I*) is then passed to the input layer. Weights (*w*) from each output node sample the input

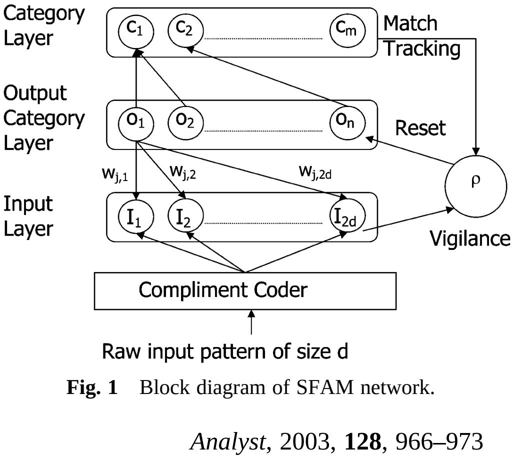

Fig. 1 Block diagram of SFAM network.

layer, making the weighting top-down. The category layer merely holds the names of the (m) categories that the network is expected to classify.

Vigilance

The vigilance parameter (ρ) is used in the learning phase of the network; its range is 0 to 1 and is used to control the granularity of the output nodes. In general, higher vigilance values result in a greater number of output category nodes to form. The network is able to self adjust its vigilance during learning from some base value (user defined) in response to errors found in classification. It is through this "match tracking" that the network is able to adjust its own learning parameters to enable the production of a new output node or to reshape the decision regions. The vigilance should not be initially set too high or the network will not generalise, becoming overtrained, a large number of output category nodes will be formed, in the worst case one for each vector input, the network will simply become a look-up table.

Compliment coding

Compliment coding ensures that the presence or lack of presence of a particular feature in the input is visible. For a given input vector **a** of d features the compliment vector $\bar{\mathbf{a}}$ represents the absence of each feature.

$$\bar{\mathbf{a}}_i = 1 - \bar{\mathbf{a}}_i \tag{1}$$

The internal compliment coded input vector **I** is then of dimension $2d$.

$$\mathbf{I} = (\mathbf{a},\bar{\mathbf{a}}) = (\bar{\mathbf{a}}_1,\ldots\mathbf{a}_d,\bar{\mathbf{a}}_i\ldots\bar{\mathbf{a}}_d) \tag{2}$$

The normalisation of a fuzzy vector is the sum of all of its points, if a fuzzy vector **x** contains n points, its norm $|\mathbf{x}|$ is

$$|\boldsymbol{x}| = \sum_{i=1}^{n} \boldsymbol{x}_i \tag{3}$$

Output node activation

If a new category is detected then a new output node is created with weights set to:

$$w_j^{\text{new}} = \mathbf{I} \tag{4}$$

When an SFAM network receives a compliment coded input pattern **I**, all of the output nodes are activated to some extent. If the activation level of a node is T, then the activation of the jth output node with weights w_j is T_j.

The function

$$T_j(\boldsymbol{I}) = \frac{|\boldsymbol{I} \cap w_j|}{\alpha + |w_j|} \tag{5}$$

where α is a small number, typically 0.0000001.

The winning node is then the node that has the highest activation value.

$$T_{\text{win}} = \max(T_j) \tag{6}$$

If two or more output nodes share the winning value then the node with the lowest index j is arbitrarily chosen to win. The category associated to this node becomes the networks classification for that input pattern.

A match function compares the compliment coded input features and the weights in the winning, selected output node to determine if learning should occur.

$$M = \frac{|\boldsymbol{I} \cap w_j|}{|\boldsymbol{I}|} \tag{7}$$

This equation may be simplified due to the fact that the norm of any compliment-coded vector is equal to the dimension d of the original input vector.

$$M = \frac{|\boldsymbol{I} \cap w_j|}{d} \tag{8}$$

Resonance and mismatch

If M is greater or equal to the vigilance parameter ρ then the selected jth output node is capable of encoding the input **I** (if node j represents the same category C as the input vector **I**) and the network is said to be in a state of resonance. The output node may then update its weights. Only one output node is allowed to alter its weights for any given training input vector.

Resonance if

$$\frac{|\boldsymbol{I} \cap w_j|}{d} \geq \rho \tag{9}$$

If the output encodes a different category from the input vector there is a 'category mismatch' condition. The node activation is suppressed and the weights for that node are not updated. If the match function value is less than the vigilance a 'mismatch reset' condition applies, the current output node does not meet the granularity represented by the vigilance, its activation is suppressed and its weights are not updated. This prevents the category from becoming increasingly non-specific (low vigilance). The vigilance value is set to match the value of the winning node plus a small value (α), eqn. (10). A new output node must be formed with its initial weights set to match the input vector, eqn. (4).

$$\rho_{\text{new}} = M + \alpha \tag{10}$$

The selected output node has its weight vector w_j updated according to the rule

$$w_j^{\text{new}} = \beta(\mathbf{I} \bigcap w_j^{\text{old}}) + (1 - \beta)w_j^{\text{old}} \; 0 \leq \beta \leq 1 \tag{11}$$

Learning rate

The learning rate β may be set to 1 for 'fast learning'. If this is the case then eqn. (11) reduces to a simple fuzzy AND of the input vector and the top-down weights of the selected output node O_j.

$$w_j^{\text{new}} = (\boldsymbol{I} \bigcap w_j^{\text{old}}) \tag{12}$$

Classification

Once SFAM has been trained a 'feed-forward' pass through the compliment-coder and into the input layer classifies an unknown pattern. The output node activation function is evaluated for each output node in the network. The category of the input vector is found by assigning it the category of the most highly activated node T_{win}.

Experimental procedure

Apparatus

A Hitachi F-2000 spectrofluorimeter with a quartz cell (Merck, UK) of 10 mm path length were used to make all total luminescence measurements. The excitation and emission slit width was set at 10 nm and the PMT voltage 700 V. The machine was operated remotely using F-3D software (Hitachi) on a 486 33 MHz PC connected *via* an RS232 port.

Four different types of edible oil were used in this study, extra virgin olive oil, non virgin olive oil, sunflower oil and rapeseed oil, all purchased from a local retailer. Hexane and acetone (Sigma-Aldrich, UK) were both of spectroscopic grade. The oil samples were kept in a cool dark cupboard, simulating a kitchen environment.

A plotting program for the total luminescence data was written using Matlab 6.5 (Mathworks,UK) that allowed visual analysis of fluorescence peaks in the EEM data. Customised versions of PCA and SFAM were written using C++, specifically for the total luminescence data. Neuroshell 2 (Ward Systems, Group Inc.,USA) was used to create, train and test both BP and RBF networks. All analysis was carried out using an Intel Celeron 433 MHz based PC.

Sampling procedure

Oil samples (3 mL) were directly pipetted from an oil bottle into a fluorescence cell and then placed in the fluorescence spectrofluorimeter and an EEM cycle carried out. One cycle reading consisted of a sweep of the λ_{ex} from 350 to 450 nm with 10 nm intervals and the λ_{em} set from 400 to 720 nm with 5 nm intervals. A complete EEM was collected in 6 min and consisted of 715 data points (11×65 matrix). There were 40 separate readings taken for each oil class, the cell being cleaned by rinsing in hexane followed by acetone before the next sample reading was taken.

Results and discussion

Excitation emission matrices of edible oils

The EEM spectra for the four unadulterated oils are shown in Fig. 2, the distinct shapes could be visually analysed for discrimination. However, with small amounts of adulteration, the changes are less obvious. Therefore analytical methods must be used for discrimination.

Principal component analysis

Fig. 3 shows a PCA scores plot, the four unadulterated oil groups occupy four separate regions within the variance space. RS oil is found in the bottom left (−2.25, −0.75) NVO central left (−1.5, 0.1), EVO central right (1.0, −0.1) and SF oil the top left (−1.25, 1.25). The adulteration shows clusters that move towards the adulterant, specific to the concentration added. At low adulteration levels there is difficulty in distinguishing between samples, therefore visual discrimination is only useful between different unadulterated oils, or oils that have been adulterated to a level above 10% v/v.

Back propagation network

Training the BP network. Training of a back propagation network involves feeding the chosen training samples as input vectors through the neural network, calculating the error of the output layer, and then adjusting the weights of the network to

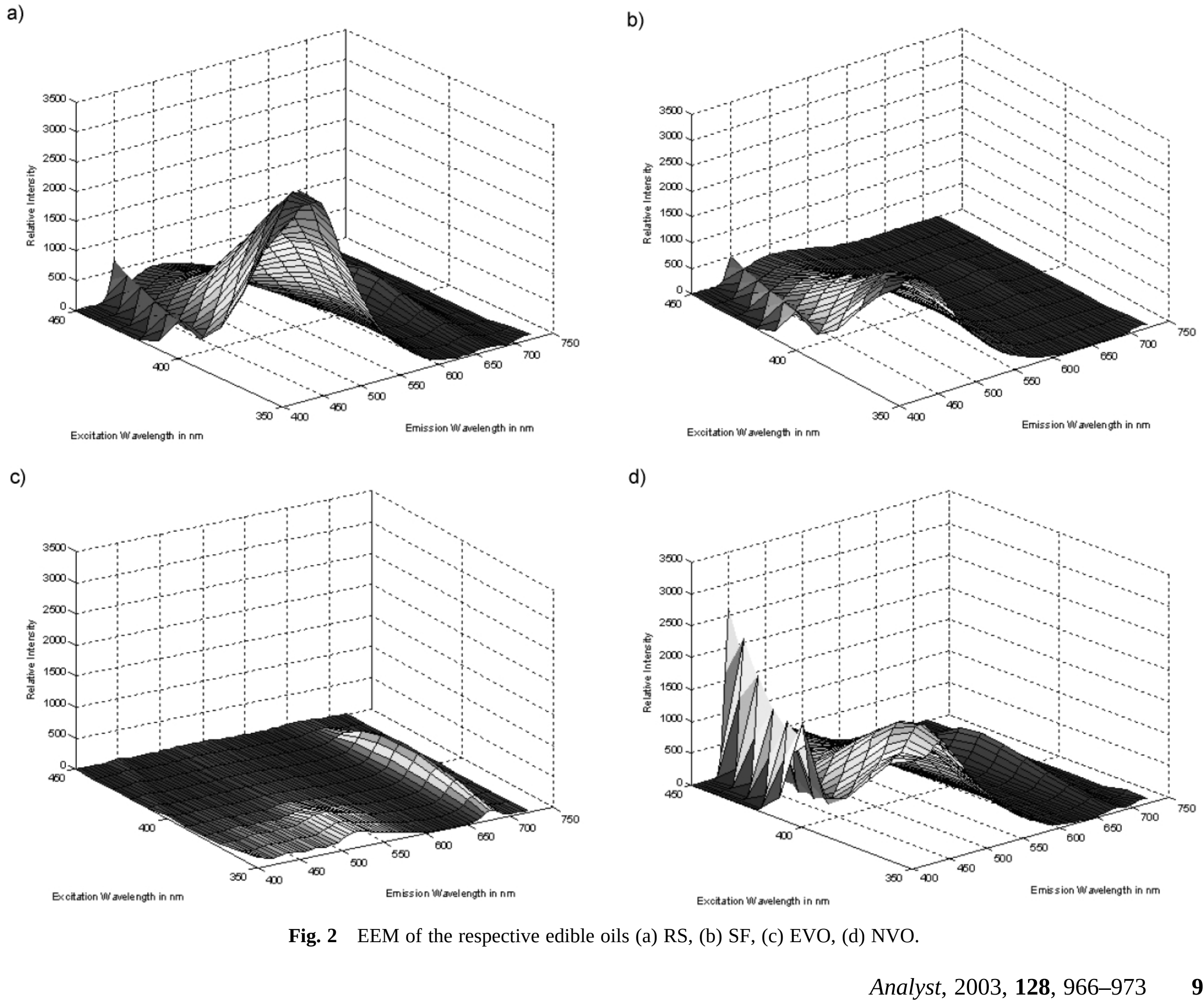

Fig. 2 EEM of the respective edible oils (a) RS, (b) SF, (c) EVO, (d) NVO.

minimize the error. Each "training epoch" involves one exposure of the network to a training sample from the training set, and adjustment of each of the weights of the network once, layer by layer. Selection of training samples from the training set may be random, or selection may simply involve going through each training sample in order.

One method of stopping training is when the network error dips below a particular error threshold. However, it is found that excessive training can have damaging results in such problems as pattern recognition. The network may become too adapted in learning the samples from the training set, and thus may be unable to accurately classify samples outside of the training set. When this happens samples from outside the original training set are either included in a revised training set and the network retrained, or a more lenient error threshold is set, or training is stopped after a pre-determined number of epochs after a minimum is detected. The error on a test set will typically start to increase so training is stopped when a sufficient number of training epochs have elapsed after a minimum to ensure that it is a global and not a local minimum.

BP results for total luminescence data. It is not feasible to set a back propagation network to have 715 input nodes as the complexity of the network would require a large amount of data and the training times would increase to an unacceptable level; therefore the data was pre-processed using PCA. The 715 dimension data was reduced to 3 dimensions, (the first three principal components, covering 97.56% of the total variance). The back propagation network was set with an architecture of 3 input, 35 hidden and 16 output nodes; the data was randomly split into 336 training points and 144 test points. The network was set to train for a maximum of 20000 epochs after the best test set configuration to avoid training to a local minimum, this took approximately 1 h 30 min to accomplish. When tested with the entire data set after training the BP network performed 100% discrimination between the unadulterated edible oils. However it did not perform well with low levels of adulteration. There were a total of ten incorrect classifications giving 97.27% correct. Eight incorrect classifications were due to 5% v/v RS in EVO, samples were classified as either 5% v/v SF in EVO or 10% v/v SF in EVO. One incorrect classification was due to 5% v/v SF in EVO being classified as 5% v/v RS in EVO, and one incorrect classification was due to 10% v/v SF in EVO being classified as 5% v/v SF in EVO.

Radial basis function network

RBF results for fluorescence data. The RBF network architecture was set to 3 input, 48 hidden and 16 output nodes; the same PCA pre-processed data as the back propagation network was used. The network was set to train for a maximum of 20000 epochs after the best test set configuration to avoid training to a local minimum, this took approximately 1 h 20 min to accomplish. The relatively long training time was due to the larger number of hidden nodes compared to the back propagation network.

The RBF network was applied to the entire data set after training had taken place. The RBF network performed 100% discrimination between the unadulterated edible oils. It also outperformed the BP network in classifications of adulterated samples, only making four incorrect predictions. There were three incorrect predictions for 5% v/v RS in EVO and one incorrect prediction 10% v/v RS in EVO groups, predicted as 5% v/v SF in EVO and 33% v/v SF in EVO respectively, giving 99.17% correct.

Simplified fuzzy adaptive resonance theory mapping

SFAM suitability for total luminescence data. Fuzzy ARTMAP has already proven itself as a supervised incremental learning system for pattern recognition with M to N dimensional mappings. SFAM reduces the computational overhead and architectural reduncancy of fuzzy ARTMAP with no loss of pattern recognition capability. The ART$_b$ and mapping modules of fuzzy ARTMAP are replaced with a category layer, the output category layer forming simple links to the appropriate category node. The output for the fluorescence data is a one dimensional vector of labels—the oil and its adulteration level. The complex mapping capabilities of fuzzy ARTMAP is not needed, SFAM with its vector of nodes, known as the output category layer is ideal for this data. SFAM is also easily capable of accepting an input vector of 715 data points, normalising and compliment coding it. The input vector then simply becomes the input layer from which the output category layer nodes are updated. So long as the length of the input vector remains constant SFAM will easily cope with that input vector, whatever the length may be.

SFAM results for total luminescence data. No pre-processing of the data was used so each sample consisted of 715

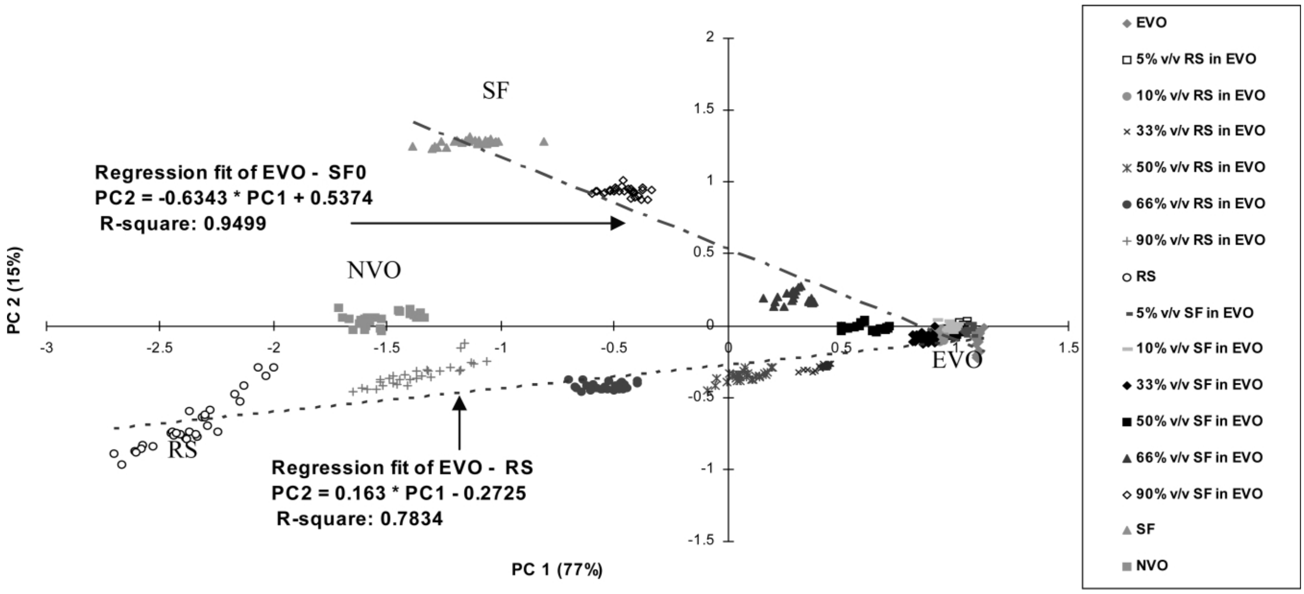

Fig. 3 PCA scores plot for total luminescence of edible oils including adulteration.

data points, resulting in 343200 pieces of data in total. The data was split into 320 readings for training and 160 for testing, a total of 480 samples. The network was set with a vigilance of 0.5 and took approximately 8 s to train, forming 30 output nodes. The network when tested on the entire data set after training had taken place performed 100% discrimination between the unadulterated edible oils. High discrimination rates for the adulterated edible oils were obtained; only misclassifying three 5% v/v SF in EVO as 5% v/v RS in EVO giving 99.375% successful classifications. Table 1 shows a summary of the neural network classification results. A limitation of the SFAM technique is that it cannot predict unknown concentrations, it is a mapping technique; it can only match data to predetermined groups. It is possible to sample data at small intervals and train on these, nonetheless SFAM is a discrete technique. For a continuous measurement on the adulteration level another method must be used. SFAM has however proved to be a both fast and useful technique for the validity of unadulterated oils when dealing with TLS data.

Calculation of adulteration

NVO was the only oil to be classified by the third principal component (5.7% of the total variation) of the PCA scores, therefore, for the adulteration of oils, neither NVO nor PC3 were used. EVO was adulterated by either SFO or RS oils. All three of these oils lie on the same PC1-PC2 plane and detecting the adulteration level of EVO with these oils is therefore the strictest test available.

RBF network curve fit. RBF networks were trained to calculate the adulteration level of both types of adulterant of EVO using the first two principal components as inputs. The architecture was set at two input nodes, 18 hidden nodes and one output node.

Principal component distance measure. The PCA scores plot in Fig. 3 shows that increasing the adulteration level of an oil follows a trend line. The adulteration percentage plotted against distance along these trend lines shows that a logarithmic relationship exists. All data was fitted using the Matlab Curve Fitting Toolbox (Mathworks, UK)

$$\text{Adult\%} = a(1 - e^{-bx}) \tag{13}$$

where a and b are constants, x is the Euclidean distance of an adulterated sample from the unadulterated sample cluster centre.

Table 1 Network classifications of adulterated oils

	BP	RBF	SFAM
Correct	470	476	477
Incorrect	10	4	3
Percentage correct	97.92	99.17	99.38

Fig. 4a and b show the Euclidean distance between the PC scores (based on cluster centres calculated using the Fuzzy c-means algorithm of Bezdek[18]) against the actual concentration of adulteration for RS in EVO and SF in EVO. Fig. 5a the actual data set for RS in EVO, Fig. 5b shows a RBF calculated adulteration level fitted. Fig. 5c shows the actual data set for SF in EVO and Fig. 5d shows a RBF calculated adulteration level fitted. Table 2 shows the calculated coefficients and degree of fit for both adulterants for the RBF calculated adulteration and the actual adulteration level.

Table 2 shows that SF in EVO using cluster centres produced a RMSE of 4.61. The RBF network was able to fit the data accurately for SF in EVO using all the data points. A RMSE of 11.19 was obtained. The simple exponential based equation performed almost as well. For SF in EVO, using all the data points a RMSE of 11.4 was obtained. Both of these methods need PCA scores values for input so the equation produces results more rapidly.

Speed of analysis

It was found that for classification training BP and RBF networks were time consuming in comparison with SFAM. Both BP and RBF networks require pre-processing of the data to allow a reasonable network architecture to be achieved. SFAM is capable of using the complete data set and is both more accurate in classification and quicker in training when used with total luminescence data.

Conclusions

SFAM is a technique that is both quick and easy to train and produces well formed predictions on TLS data for unadulterated or adulterated edible oils. A mathematical relationship between the Euclidean distances of an adulterated oil to its unadulterated base PCA score is both a quick and accurate method to determine the level of adulteration of the oil. RBF networks were found to be slightly more accurate in fitting the data but

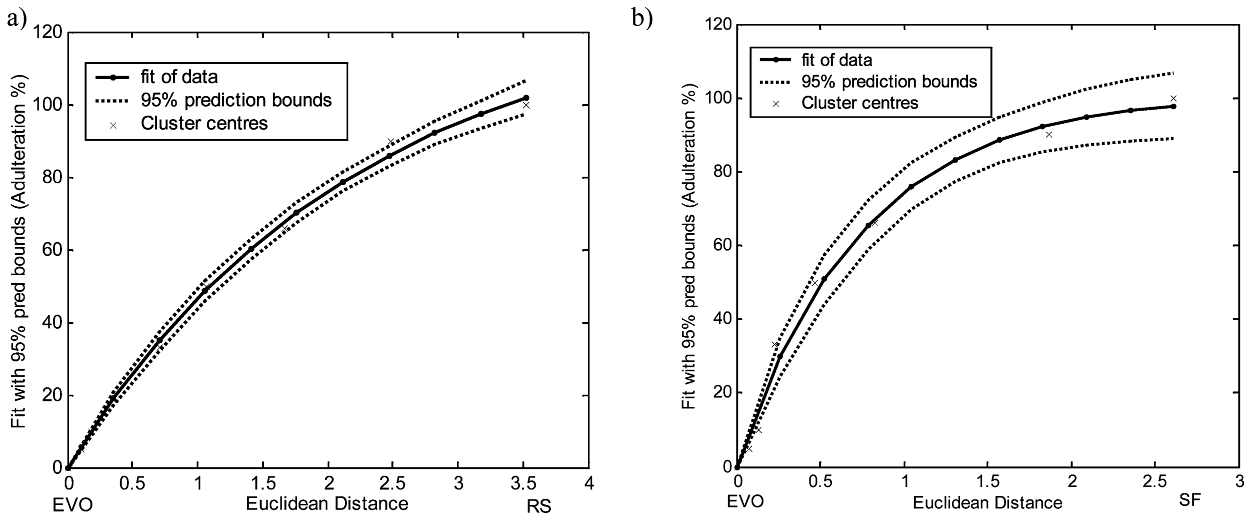

Fig. 4 Adulteration *vs.* Euclidean distance of cluster centres (a) RS in EVO, (b) SF in EVO.

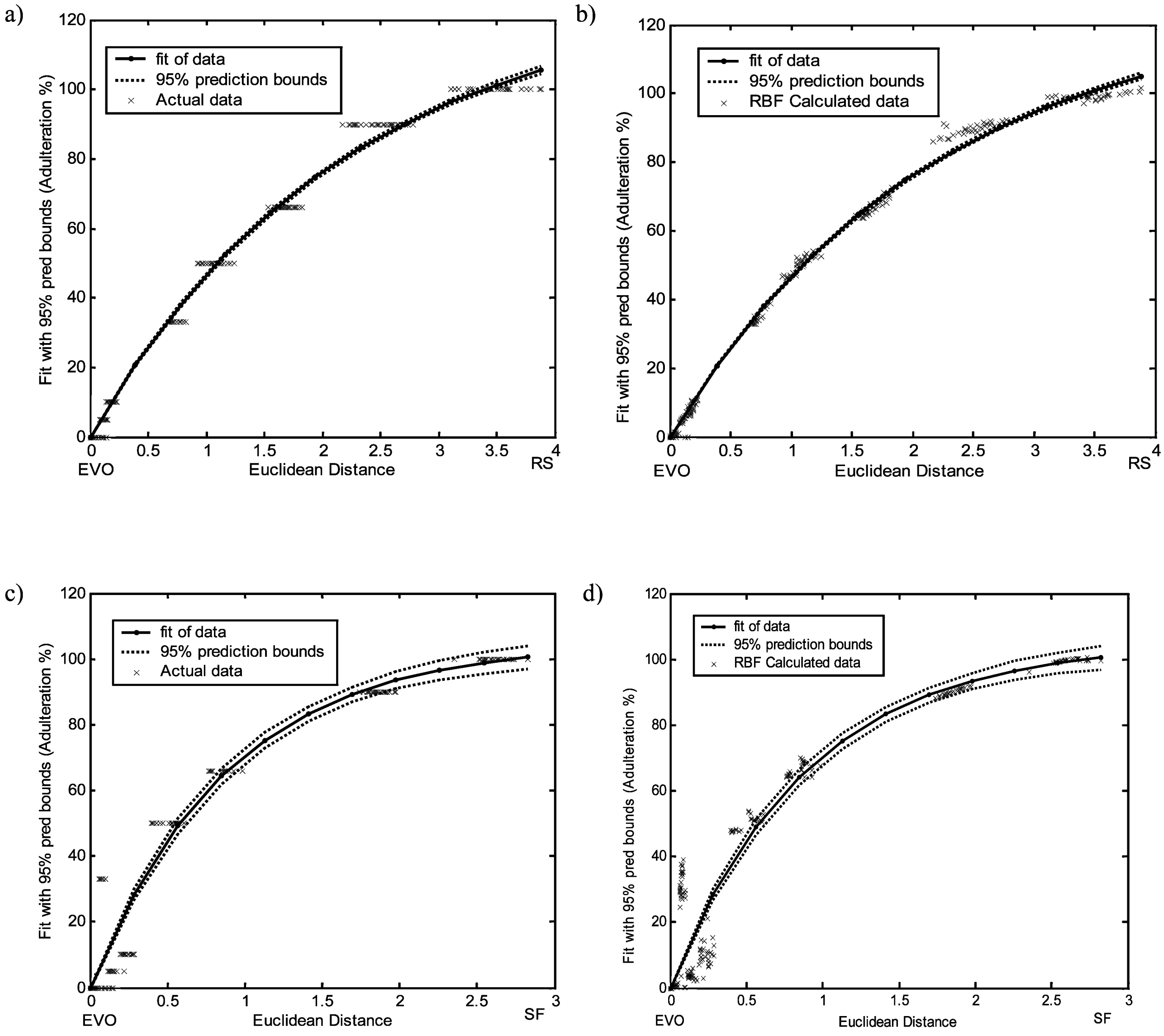

Fig. 5 RS in EVO (a) Actual data, (b) RBF fit SF in EVO, (c) Actual data, (d) RBF fit.

Table 2 Fit and coefficients for adulterated oil level calculation

Fit	*a*	*b*	R^2	SSE	RMSE
SF in EVO cluster	100.9	1.338	0.9879	127.5	4.61
SF in EVO actual	105.0	1.118	0.902	31070	11.4
SF in EVO RBF	105.0	1.112	0.9058	29950	11.19
RS in EVO cluster	127.9	0.4513	0.9975	26.73	2.111
RS in EVO actual	127.1	0.4563	0.9933	2118	2.977
RS in EVO RBF	125.5	0.4656	0.9955	1417	2.435

are, however, more time consuming to train. A combination of SFAM and PCA score curve fitting could be used to make both qualitative and quantitative determinations of unknown adulterated and unadulterated fresh edible oils. SFAM will quickly determine if the oil is adulterated and to the approximate level. The Euclidean distance equation may then be used to calculate the level of adulteration.

References

1 R. Goodacre, D. B. Kell and G. Bianchi, *J. Sci. Food Agric.*, 1993, **63**, 297–307.

2 I. M. Lorenzo, J. L. P. Pavon, E. F. Laespada, C. G. Pinto and B. M. Cordero, *J. Chromatogr., A*, 2002, **945**, 221–230.

3 A. D. Shaw, A. di Camillo, G. Vlahov, A. Jones, G. Bianchi, J. Rowland and D. B. Kell, *Anal. Chim. Acta*, 1997, **348**, 357–374.

4 L. Webster, P. Simpson, A. M. Shanks and C. F. Moffat, *Analyst*, 2000, **125**, 97–104.

5 D.-S. Lee, E.-S. Lee, H.-J. Kim, S.-O. Kim and K. Kim, *Anal. Chim. Acta*, 2001, **429**, 321–330.

6 A. H. El-Hamdy and N. K. El-Fizga, *J. Chromatogr., A*, 1995, **708**, 351–355.

7 Z. Ali, D. James, W. T. O'Hare, F. J. Rowell and S. M. Scott, *J. Therm. Anal. Calorim.*, 2003, **71**, 147–154.

8 S. Shimoyama, Y. Noda and S. Katsuhara, *Dyes History Archaeol.*, 1996, **15**, 27–42.

9 J. Alexander, G. Mashak, N. Kapitan and J. A. Siegel, *J. Forensic. Sci.*, 1987, **32**, 72–86.

10 D. Patra and A. K. Mishra, *Anal. Chim. Acta*, 2002, **454**, 209–215.

11 D. Patra and A. K. Mishra, *Appl. Spectrosc.*, 2001, **55**, 338–342.

12 A. Baker, *Water Res.*, 2002, **36**, 189–195.

13 Q. Li, X. Yao, X. Chen, M. Liu, R. Zhang, X. Zhang and Z. Hu, *Analyst*, 2000, **125**, 2049–2053.

14 K. Pearson, *London Edinburgh Philos. Mag. J. Sci.*, 1901, **2**, 559–572.

15 D. S. Broomhead and D. Lowe, *Complex Systems*, 1988, **2**, 321–355.

16 G. A. Carpenter and S. Grossberg., *Comput. Vision Graph. Image Process.*, 1995, **37**, 116–165.

17 T. Kasuba, *AI Expert*, 1993, **8**, 18–25.

18 J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum, NY, 1981.