Wavelet Transform Preprocessing for Temperature Constrained Cascade Correlation Neural Networks

Chunsheng Cai[†] and Peter de B. Harrington*

Center for Intelligent Chemical Instrumentation, Department of Chemistry and Biochemistry,
Clippinger Laboratories, Ohio University, Athens, Ohio 45701-2979

Received April 22, 1999

Wavelet transform (WT) preprocessing offers two advantages: data compression and noise reduction. Wavelet compression increases the training rate of a neural network and allows neural network models to be obtained from data that otherwise would be prohibitively large. Two types of WT compressions have been studied: linear and nonlinear. Linear wavelet compression in which data are compressed by frequency usually provides better compression efficiency. Nonlinear wavelet compression in which data are compressed by amplitude is useful when the information cannot be easily represented by low-frequency components. The reduction of noise is important in the multivariate analysis because many methods overfit the data if care is not taken. Ion mobility spectrometry (IMS) is a sensing technique that can generate large amounts of data in short-time monitoring events. The temperature constrained cascade correlation neural networks (TCCCN) are a powerful chemometric method for pattern recognition. The IMS data of some volatile organic compounds are used to evaluate the WT-TCCCN method, and the results indicate that WT-TCCCN works well.

INTRODUCTION

Compression of analytical data is reemerging as an important research area because miniaturized sensors are much smaller than the data storage devices. The storage or transmission of large amounts of data may be a problem especially for smaller instruments, so data compression becomes a useful tool. Alternatively, the direct processing of large data sets is typically time consuming. Sometimes direct processing of data may exceed a personal computer's processing capability if the data set is large enough or the computer is not equipped with sufficient memory. Therefore, the purpose of data compression is not only for storage but also for preprocessing, in which the compressed data are analyzed without prior decompression. There are two types of compression approaches. The first type uses coordinate systems that are defined by the data. A typical example is principal component analysis (PCA) for which the coordinate system maximizes the variance of the data set. The second type uses a fixed coordinate system, such as Fourier and wavelet compression.

PCA is an efficient technique for compression[1,2] and can be a useful preprocessing tool.[3,4] However, when the data size is overwhelmingly large (e.g., over 10000 × 1000), the PCA calculation may not be possible. Because the PCA scores are based on the variable loadings from a specific data set, the observation scores will not be comparable for different data sets if the loadings vary.

The Fourier transform (FT) can be a preprocessing means of reducing the dimensionality of data, and it can be performed very fast.[5,6] The wavelet transform (WT) has been an efficient method for data compression, fast computation, and noise reduction.[7] If a suitable wavelet is selected, the WT compression can offer better compression efficiency than FT compression. Both Fourier and wavelet compressions are lossy compression methods. Lossy compression is a term used in the engineering literature and indicates that some information is lost during the data compression. For some types of data the information that is lost may be of little use or even detrimental to pattern recognition. The WT is potentially more robust in terms of compression and denoising than FT. These attributes make WT an ideal preprocessing method for modeling large sets of data. The use of WT as a preprocessing tool is increasing in popularity.[8−11]

The advantages of WT over FT include versatility and efficiency. There are many wavelet types. However, selecting the correct wavelet type poses a disadvantage, as demonstrated in wavelet denoising.[12] Bos and Vrielink used WT as a preprocessing tool for classifying IR spectra.[8] They did not use a criterion for selecting the wavelet type and the transform level. For the WT denoising issue, only one spectrum has to be processed at a time, so some criteria can be proposed to choose the wavelet type and other parameters.[12]

For WT compression, the same criteria may be used to select a wavelet type. However, two aspects must be considered. First, the spectra in the training and test sets should be processed uniformly because an optimized set of wavelet coefficients for a spectrum may not be optimal for the other spectra. If each spectrum is processed with different wavelet transforms, the network may model artifacts or minute differences in the spectra. Second, the major advantage of fast training may disappear due to the additional computational time for selecting the wavelet type. Fortunately, there is an empirical rule to guide the selection of a suitable wavelet type. For example, if the signal contains sharp peaks or discontinuities, Haar or other compact wavelets are applicable. If the signal comprises broad peaks,

* To whom correspondence should be addressed. Fax: (740) 593-0148. E-mail: harring@helios.phy.ohiou.edu.

[†] Present address: Hoechst Marion Roussel, C1-M0336, 10236 Marion Park Drive, Kansas City, MO 64137-1405.

10.1021/ci9903253 CCC: $18.00
Published on Web 09/09/1999

a smoother wavelet such as Daublet 20 or larger may be employed. For the signals in between such as IMS spectra, medium wavelets such as Daublets from 6 to 18 may give good results.

Two characteristics of lossy compression are compression efficiency and efficacy. Efficiency measures the extent of compression. Compression efficiency may be measured simply by the ratio of the compressed data to the uncompressed data. Efficacy measures the loss of information that occurs during the compression process. Compression efficacy usually cannot be quantified because the underlying true signal is unknown for real data. A tradeoff usually exists between these two features. More compression efficiency means less compression efficacy. A better compression method should give larger compression efficiency with suitable efficacy.

Data compression is helpful when data are transferred wirelessly from field sensors to a computer. Data acquisition is very fast (25 ms/spectrum) in ion mobility spectrometry (IMS). Portable IMS sensors tend to have limited peak capacities and intricate spectra that are caused by the formation of cluster ions. IMS spectra usually have a reactant ion peak (RIP) which is a varying background component. The atmospheric pressure chemical ionization (APCI) charge transfer reactions between the RIP and the analytes make the instrumental response nonlinear. Chemometric methods can help simplify the data and thereby facilitate interpretation. Therefore, efficient compression and processing methods are needed for IMS data.

PCA is not suitable for this purpose because it requires groups of spectra to calculate the principal components (PCs). The entire set of PCs should be recalculated with any additional spectra. Besides, the loadings need to be stored also. Fixed-ordinate transforms such as WT offer two advantages. First, the spectra are transformed individually, and a very high efficiency can be achieved. Second, the compressed spectra can be input to other processing methods without decompressing the data, and the results are comparable. Meanwhile, data processing methods are made more efficient by the use of compressed data.

Neural networks have been developed and applied to every scientific field. Back-propagation neural networks (BNNs) are popular among chemists.[13,14] The BNNs are feedforward networks trained by error back-propagation. The network size is fixed before training. The disadvantages for BNNs include the extra step of configuring the network and longer training times.

Cascade correlation neural network (CCN) was developed to alleviate these problems.[15] The CCN can build its own topology during training. It starts with a minimal network (i.e. input neurons and output neurons). If the error does not decrease below the desired value, then a hidden neuron is added. Each added hidden unit is connected to input neurons as well as previously installed neurons. Several hidden units can be trained in parallel, and the one with the largest covariance is used. Once trained, the hidden unit no longer changes; thus only one unit is trained at a time, eliminating the chaos of simultaneously adjusting all processing units as in the BNN training. The number of hidden units of the CNN keeps increasing until the desired error is reached or the error converges above the threshold.

However, the CCN still has an overfitting problem. One way to reduce the overfitting is to introduce a parameter, called computational temperature, to the sigmoid transfer function. Temperature constrained CCNs (TCCCNs) have been developed as pattern recognition methods.[16] TCCCNs are powerful chemometric methods for pattern recognition and quantitation. However, the training usually is very time-consuming for large data sets. Numerous methods have been proposed to speed the training step by using different weight optimization algorithms. The other way to speed up the overall training occurs outside the training loop by reducing the input size without much loss of information.

The TCCCN is directly performed on WT compressed data, eliminating the need to restore the data to their original domain. Two additional advantages are demonstrated: faster computation due to the reduced dimensionalities and better predictive results due to the removal of noise. For large amounts of data, such as tens of thousands of IMS spectra, direct processing of the data sometimes is impossible due to its large memory requirement. The WT compressed methods make it possible. In this paper, TCCCN classification of IMS data coupled with WT preprocessing is studied.

THEORY

There are two approaches for performing wavelet compression, linear and nonlinear. The linear compression keeps low-frequency wavelet coefficients, also called compression by frequency. If a 50% compression is desired, only one transform is needed. The detail part is removed, and only the first smooth part is retained. Similarly, two transforms are needed for a 75% compression, three transforms for an 87.5% compression, and four transforms for a 93.75% compression. The extent of compression depends on the data and the wavelet type. High compression is always expected, so a suitable wavelet type should give as much information in the low-frequency region as possible. It should be emphasized that the purpose is not reconstruction fidelity but discrimination ability. If there are some large coefficients (relatively, compared to noise) in the high-frequency region that are not characteristic for a specific class, they can be safely eliminated to achieve a higher compression. The 87.5% compression is not uncommon for most chemical data that are usually continuous. The selection of wavelet type is important for high compression. For example, for the same 87.5% compression, a certain wavelet type may result in more information in the low 12.5% region than another type. As described above, an empirical rule may be applied, and usually several wavelet types are suitable.

Figure 1 demonstrates the linear wavelet compression of an IMS spectrum. A spectrum is plotted in panel A (top, left) and the complete wavelet spectrum by Daublet 14 transform is given in panel B (top, right). After using an 8:1 linear compression, i.e., only using the first 64 variables in panel B, the spectrum is regenerated and given in panel C (bottom, left). Panel D (bottom, right) gives the difference of the spectra in A and C. The difference indicates that noise is mostly removed and little signal is lost.

Nonlinear compression keeps all wavelet coefficients whose amplitudes are larger than a threshold, also called compression by amplitude. In this way, an index number

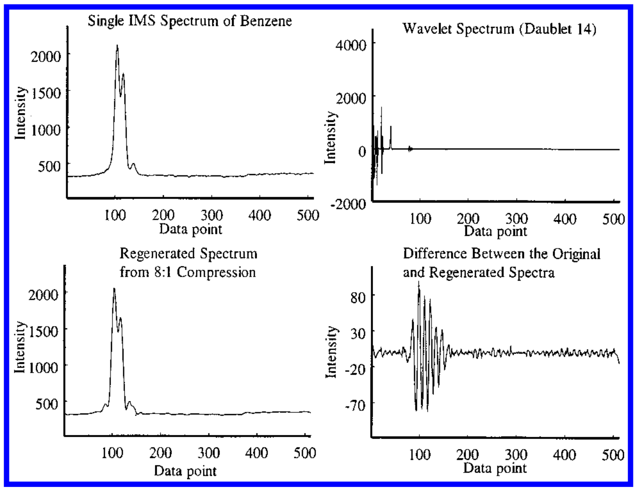

Figure 1. Demonstration of linear wavelet compression with positive ion mobility spectra. Intensities have units of mV and data points span a range of 3.0−15.0 ms.

that preserves the position information must be stored also,[17] similar to the computer storage of mass spectra for which the mass to charge ratios and intensities are stored. A global cutoff is simple and can give good compression even though some other thresholding methods are available as described for wavelet denoising.[12] The selection of the universal threshold is

$$t = \sigma[2 \ln(N)]^{1/2} \tag{1}$$

for which N is the length of data array and σ is the standard deviation of the noise.[18] For most real data, σ is unknown but can be estimated as s. The first detail part of wavelet coefficients x_i can be used to estimate the noise by

$$s \approx \text{median}(|x_i|)/0.6745 \tag{2}$$

for which s is the noise estimate.

For a single spectrum, the operation of nonlinear wavelet compression has no difference with wavelet denoising. Wavelet transforms can achieve very high compression ratios and are used mostly for data storage and for noise removal for a single spectrum. However, it is possible to select the feature by using a variable mapping routine, like the preprocessing of mass spectra,[16] because the compressed data resemble low-resolution mass spectra. The index axis is built from the training data set. For the prediction sets, if the spectra contain index numbers that are missing from the index axis, they are omitted from the calculation, and if the positions are missing in the prediction set, they are assigned intensity values of zero.

For both linear and nonlinear wavelet compressions, the retained wavelet coefficients are input into the neural networks. The temperature is adjusted so that it maximizes the magnitude of the first derivative of the covariance between the output and the residual error with respect to temperature. This objective function is advantageous, because it causes the error surface to remain steep, which facilitates gradient training. In addition, outputs are continuously distributed throughout their range when the derivative is maximized, which ensures fuzzy interpolation of the hidden unit.

The TCCCN is a modified version of the one described by Harrington.[16] The transfer function is a temperature constrained sigmoid function for hidden layer training and a sigmoid function for output layer training. The training is performed by the conjugate-gradient method instead of quickprop. Quickprop was devised with the CCN to speed up the back-prop training.[19] Practically, conjugate-gradient algorithm is efficient for a large number of weights in the training.[20]

For classification, the TCCCN may not always converge with the quickprop algorithm. In the output layer, there are no temperature constraints and the weight vectors were computed by the conjugate gradient minimization of the back-propagated errors. An improvement was achieved by combining the second-order Levenberg−Marquardt algorithm with the conjugate-gradient algorithm. This approach was only applied to nonlinear units in the output layer and will be presented in more detail. Each output unit weight is trained by using singular value decomposition to regress the residual errors between the target and network outputs onto the Jacobian matrix of the inputs. This weight vector of errors is used to replace the gradient in the conjugate gradient algorithm. The steps are given as follows.

The Jacobian matrix (**J**) is calculated by multiplying the column vector of the derivative of the nonlinear transfer function by the inputs feeding into the output unit.

$$\mathbf{J} = f'(\text{net})\mathbf{I} \tag{3}$$

for which $f'(\text{net})$ is the derivative of the transfer function. In this case, the input matrix **I** is composed of the h outputs from the temperature constrained hidden units and a column that is set to unity for bias. For sigmoid units, the derivative is simply the output value for the object multiplied by unity minus the output value. The pseudoinverse is calculated from this Jacobian matrix by using singular value decomposition.

$$\mathbf{J}^{+} = \mathbf{U}\Lambda^{-1}\mathbf{V}^{\mathrm{T}} \tag{4}$$

The matrix **J** tends to be well-conditioned, because the hidden units are trained to maximize the covariance of the residual error and their output values. Also, because only the hidden units connect to the output units, the number of objects will in most cases exceed the number of columns. Nevertheless, only eigenvectors whose eigenvalue is greater than a part per million of the eigenvalue sum are used to calculate the pseudoinverse $\mathbf{J}^{+}$. The number of hidden units is h, and the eigenvalues are λ_i. Finally, the gradient is replaced by the new vector of $\Delta\mathbf{w}_j$, but instead of adding this value as the LM algorithm does, $\Delta\mathbf{w}_j$ replaces the gradient in the conjugate gradient optimization. This method decreased training time and improved convergence.

$$\Delta\mathbf{w}_j = \mathbf{J}^{+}(\mathbf{y}_j - \hat{\mathbf{y}}_j) \tag{5}$$

The vector of weight updates for output unit j is computed by multiplying $\mathbf{J}^{+}$ by the residual error obtained by subtracting the target outputs from the network outputs for each property j.

The training for the output layer undergoes the same procedures, except that it uses the outputs from the hidden layer as inputs and no temperature training for an output layer. After the training, the weights and the temperatures for both hidden layers and the output layer are stored. For the testing, the input data are processed by the hidden layer to generate hidden layer outputs, and then these outputs are input into the output layer for training, generating the predicted values.

To evaluate WT-TCCCN for different wavelets and non-preprocessing TCCCN methods, the prediction accuracy, i.e. percentage of the correctly classified, is used. Root mean square error (RMSE) for evaluating classification models is unduly influenced by outliers. However, RMSE is still an effective evaluation parameter because training uses RMSE as the convergence criterion. RMSE is also useful in evaluating the selectivity of the classifier and providing extra information when the methods have equivalent prediction accuracy. For the classes, unordered coding with target variables, i.e. (1, 0, 0, ...) for class A, (0, 1, 0, ...) for class B, etc., is used. In this way, RMSE can be calculated by the following equation,

$$\mathrm{RMSE} = \left[\frac{\sum_{i}^{N_s}\sum_{j}^{N_c}(\hat{y}_{ij} - y_{ij})^2}{N_s N_c}\right]^{1/2} \qquad (6)$$

for which $\hat{y}$ and y are the predicted and target output values, respectively, and N_s and N_c are the number of spectra and outputs, respectively.

EXPERIMENTAL SECTION

The IMS data were used to evaluate the proposed WT-TCCCN classification method. The IMS spectrometer was a handheld device, the Chemical Agent Monitor, CAM 482-301N (Graseby Ionics Ltd., Watford, Herts, U.K.) and was used with a single modification. The reagent chemistry was based on water rather than acetone. The acetone reservoir was removed from the instrument, and the instrument was placed in a vacuum oven to remove residual acetone. The setup is described elsewhere.[5] Positive ion mode was used for all data collection. The data acquisition rate was 80 kHz.

The IMS spectra from six organic compounds, i.e., ethanol, 1-propanol, 1-butanol, acetaldehyde, propionaldehyde, and *n*-butyraldehyde, were collected. The sample was exposed to the inlet of the instrument for several seconds then removed. The concentration profiles had a rapid increase and slow decay, which is typical with instruments that sample through a membrane interface. For each compound, several hundred spectra were collected. Therefore, the data were from a variety of concentrations of each compound, spanning from high concentration to very low concentration.

The data were partitioned into two sets in two ways. First, the 800 spectra from high and low concentrations were used to train the neural networks, and the 1300 spectra from medium concentration were used for testing. The second method partitioned the data randomly into two subsets, half of the data in the training set and the other half in the testing set. Three randomly partitioned training−testing subset pairs were used.

The IMS spectra had 1300 variables between drift times of 1.0 and 17.0 ms. Because an analyte signal rarely occurs outside the range of 3.0−15.0 ms under these experimental conditions, a filter routine is used to reduce the variables to 961. Then each spectrum was extended to a length of 1024 using a linear interpolation of the first point at (3.0 ms) and the last point (15.0 ms). This step was necessary because the WT requires that the spectrum have a length that is a power of 2, and the linear interpolation prevents introduction of discontinuities into the data that may affect the WT.

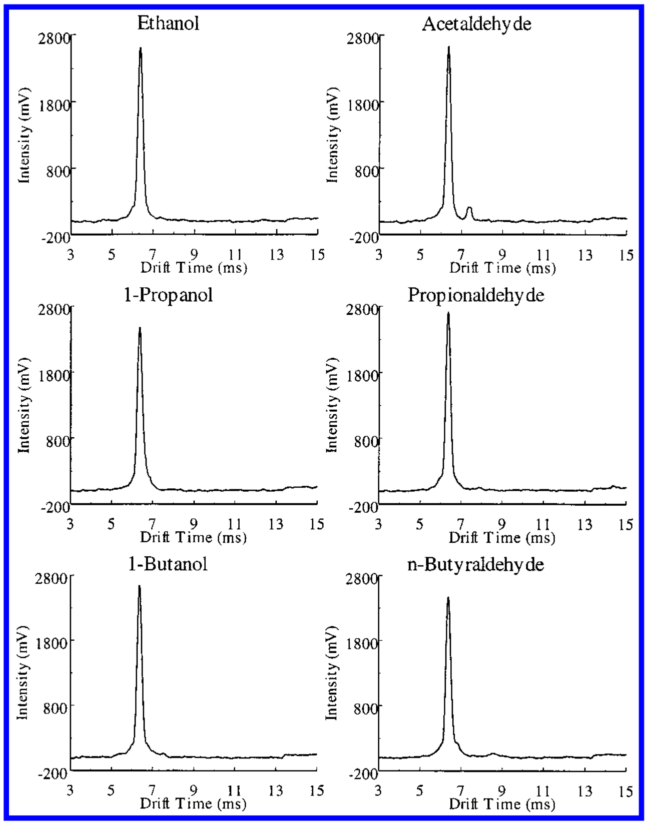

Figure 2. Positive mode IMS spectra for each compound at the lowest concentration.

The data processing programs were written in C++, compiled within a Borland C++ compiler (Version 5.02), and run on a Pentium Pro 200 MHz computer equipped with 64 MB of RAM, which was operated by MS-Windows NT 4.0. The one-dimensional wavelet transform routine was adapted from DAUBWAVE,[21] and wavelet types were extended to 15 Daublets, 5 Coiflets, and 7 Symmlets.

RESULTS AND DISCUSSION

WT-PCA. For the low concentration of each compound, there are no obvious analyte peaks present in the IMS spectra and the differences among the spectra were not obvious. The signal appeared as a small peak after the large RIP peak, as given in Figure 2. PCA is a well-known display method for characterizing multidimensional data. For IMS data that contained 2100 spectra with 1300 variables, direct calculation of PCA is time-consuming for a personal computer. It took 34 min for the direct PCA calculation using singular value decomposition.[22] PCA scores can be easily obtained from Fourier compression.[6] Linear wavelet compression may be more efficient than Fourier compression if a suitable wavelet is chosen. For the IMS spectra, several wavelets can give a satisfying compression ratio ranging from Daublet 6 through Daublet 12. The compression ratio of 87.5% can remove most noise without throwing away much information. For Daublet 6 compressed data, the PCA calculation only took 1 min. The WT-PCA score plot, given in Figure 3, gives no visual difference from the direct PCA score plot.

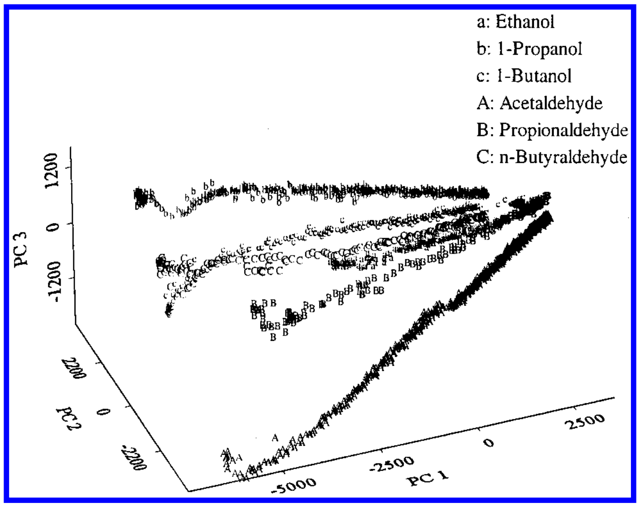

Figure 3. Plot of the first three principal components (PC). The principal component analysis was performed on the wavelet compressed IMS data. Daublet 6 was used.

From Figure 3, it can be seen for all compounds that the spectra converge to the same position in the data space. This position is the blank spectrum. The scores diverge in different space direction with increasing concentration. The low-concentration spectra had very subtle differences that indicated classification would be difficult for these spectra.

Linear Wavelet Compression. For the data sets that were partitioned by concentration, the results for noncompressed and different wavelet compressions are given in Table 1. The noncompressed data are designated by the 1:1 ratio. For the Daublet 2 (namely, Haar) transform, the wavelet probably is too compact for IMS data and some information cannot be moved to the low-frequency region, which means that the WT-TCCCN may not give better results than the TCCCN. For the Daublet 14 transform, the wavelet is wider compared to the peaks of the IMS spectra, and the WT-TCCCN results are not better than the noncompressed TCCCN results. For wavelets in this range, a 40% lower RMSE can be obtained by WT-TCCCN from the 50% compression (2:1) to 96.88% (32:1). The prediction accuracy of 100% can be achieved for some cases compared to the 98.38% for the noncompression.

The results in Table 1 are from the compression with partial wavelet transforms. For a given compression, for example 8:1, retaining one-eighth of the spectrum data can be further transformed. This could change the feature of this portion of the spectrum. However, the results are not significantly different. Therefore, for the linear compression, only partial transforms are used instead of complete transforms.

For the three randomly partitioned data sets, five runs of WT-TCCCN models were built using different random initial conditions for each training−testing set pair. Therefore, a total of 15 runs were accomplished for a given wavelet at a specific compression ratio. Figure 4 gives the results of prediction error plots. Each subplot gives the results from a wavelet type from Daublet 2 through Daublet 14. The data sets from this random partitioning should be more difficult than the data from the first partitioning, because they contain many very low concentration spectra for each compound. As shown in Figure 2, no visual difference between the low-concentration spectra for some compounds can be observed, and if they are included in the prediction set, it simply increases the difficulty of classification. Generally, WT-TCCCN gives lower RMSE prediction values than TCCCN for the compression ratio between 2:1 and 32:1. For the prediction accuracy, similar results are observed and WT-TCCCN gives better results than TCCCN.

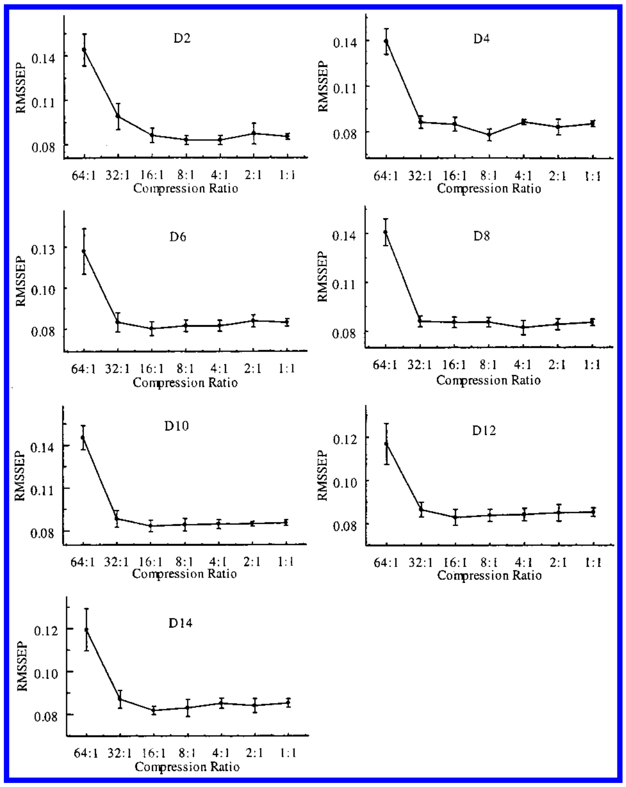

Figure 4. Prediction RMS for the testing data randomly partitioned. Subplots are results for Daublets 2−14 (D2−D14). Error bars stand for 95% confidence intervals.

Nonlinear Wavelet Compression. For the training data from the first partitioning way, a single spectrum, which has 1024 points, can be compressed to less than 50 points. After union of the variables for the 800 spectra, the variables are between 750 and 800 points for different types of wavelet compression. The overall compression is slightly above 20%, and the faster computation is not a significant advantage,

Table 1. Prediction RMSE and Percent Accuracy (A%) for WT-TCCCN

	D2		D4		D6		D8		D10		D12		D14	
ratio	RMSE	A%	RMSE	A%	RMSE	A%	RMSE	A%	RMSE	A%	RMSE	A%	RMSE	A%
1:1	0.1157	98.38	0.1157	98.38	0.1157	98.38	0.1157	98.38	0.1157	98.38	0.1157	98.38	0.1157	98.38
2:1	0.1420	87.92	0.0606	100.0	0.1077	100.0	0.0617	99.85	0.0749	100.0	0.0756	99.85	0.1268	96.08
4:1	0.1373	98.69	0.0641	99.85	0.1099	99.46	0.0753	99.92	0.0713	100.0	0.0700	100.0	0.1077	100.0
8:1	0.1907	92.23	0.0654	100.0	0.0723	100.0	0.0703	100.0	0.0776	100.0	0.0799	99.00	0.1128	99.62
16:1	0.1458	99.31	0.0814	100.0	0.0679	100.0	0.0757	100.0	0.0723	99.46	0.0524	100.0	0.1272	97.23
32:1	0.2127	84.92	0.0720	100.0	0.0870	99.85	0.0983	99.92	0.0628	100.0	0.0703	100.0	0.1306	95.15
64:1	0.3016	43.69	0.2350	63.15	0.2031	73.38	0.1446	91.62	0.2224	66.38	0.1173	96.23	0.1943	86.15

Table 2. Comparison of Results for Nonlinear WT-TCCCNs and Noncompression TCCCN

noncompression	
RMSE	0.1157
A (%)	98.38
WT (D12) compression	
RMSE	0.0712
A (%)	100.0
WT (D14) compression	
RMSE	0.0875
A (%)	100.0

Table 3. Prediction Results for the Randomly Partitioned Data Set by TCCCN, TCCCN with Decimation (D-TCCCN), and TCCCN with Wavelet Compression

	RMSE (std dev)	*A* (%) (std dev)
TCCCN	0.0845 (0.0037)	98.7 (0.52)
D-TCCCN	0.0837 (0.0142)	97.9 (1.31)
WT-TCCCN	0.0819 (0.0052)	98.9 (0.47)

although the data matrix becomes very sparse after compression and the computation speed can be improved. The prediction results, given in Table 2, indicate an improvement for nonlinear compression, but not better than some linear compression results. Therefore, in terms of preprocessing, linear wavelet compression performs better than nonlinear compression. That is the reason why the linear compression is advocated for preprocessing. However, nonlinear compression is useful in case the information cannot easily transfer to the low-frequency region for some data.

TCCCN with Decimation. By visually examining the IMS spectra, one may notice that the data are oversampled. The IMS spectra have relatively wide peaks, and 40 data points would characterize a 0.5 ms peak width. Decimation is an alternative and simple means of compression, which simulates collecting the data at a slower acquisition rate.

For the IMS spectra that have 961 variables, decimation by eight was used to mimic the 10 kHz sampling rate. The spectra were decimated by a factor of eight (961 compressed to 120 variables) to compare the results of the 8:1 wavelet compression. The decimated data were input into the TCCCN.

For the randomly partitioned data with 1050 spectra in the training set and 1050 spectra in the test data set, Table 3 gives the results from direct TCCCN, TCCCN with decimated data, and WT-TCCCN. The TCCCN with decimated data gives an equivalent RMSE but has worse predictive ability. Decimation by eight does not improve the predictive ability of TCCCN, while WT-TCCCN improves the performance of TCCCN. Both decimation and wavelet methods compressed the spectra at the ratio of 8:1.

Comparison of Computational Efficiency. The times consumed by directly performing a TCCCN and WT-TCCCN of six data sets are listed in Table 4. Note the time varies largely for different data sets because different numbers of hidden layers were built during the training. For the WT-TCCCN, the time to perform WT is included. Actually, the time for WT can be ignored compared to the training time. For example, the typical time to transform 1000 spectra is less than 10 s. The training time improvement may not be significant. However, in two cases, the compression method must be considered. First, if the training data are too large for the computer to handle, the compression may make the data processing possible. Second, if a method is to be applied online and it is not fast enough to give the result immediately, the compression should be a better choice than simply reducing the sampling frequency to reduce the numbers of variables.

Table 4. Comparison of Time Consumed in Minutes

data size			
no. of spectra	no. of variables	TCCCN	WT-TCCCN
800	481	4.5	1.9
	961	16.2	2.6
	1300	22.0	2.4
2100	481	13.7	5.4
	961	72.7	4.5
	1300	26.5	6.8

CONCLUSION

The wavelet transform is an effective compression method for both data storage and further data processing. The wavelet compression can significantly improve the training rate of neural networks and achieve better results. By using WT preprocessed data, a 40% lower prediction error may be obtained for IMS data. Both linear and nonlinear wavelet compression can be used to speed the processing and improve the results due to the reduction of data size and removal of noise. However, linear compression performs better than nonlinear compression in terms of training speed and prediction ability for IMS data. For linear wavelet compression, only several layers of transformation are needed, because no change in classification performance was observed by further transforming the spectra, even though further transformation could change the features of the spectra.

ACKNOWLEDGMENT

This work was presented in part at the 50th Pittsburgh Conference in Orlando, FL, 1999. The Center for Intelligent Chemical Instrumentation is thanked for supporting the Pittcon trip. U.S. Army ERDEC is thanked for the use of their instruments. Tricia Buxton is thanked for her suggestions.

REFERENCES AND NOTES

(1) Jolliffe, I. T. *Principal Component Analysis*; Springer-Verlag: New York, 1986.

(2) Brown, S. D. Chemical Systems under Indirect Observation: Latent Properties and Chemometrics. *Appl. Spectrosc.* **1995**, *49* (12), 14A−31A.

(3) Brown, C. W.; Lo, S.-C. Chemical Information Based on Neural Network Processing of Near-IR Spectra. *Anal. Chem.* **1998**, *70*, 2983−2990.

(4) Poppi, R. J.; Massart, D. L. The Optimal Brain Surgeon for Pruning Neural Network Architecture Applied to Multivariate Calibration. *Anal. Chim. Acta* **1998**, *375*, 187−195.

(5) Cai, C.; Harrington, P. B.; Davis, D. M. Two-Dimensional Fourier Transform Compression. *Anal. Chem.* **1997**, *69*, 4249−4255.

(6) Harrington, P. B.; Hu, L. Recovery of Variable Loadings and Eigenvalues Directly from Fourier Compressed Ion Mobility Spectra. *Appl. Spectrosc.* **1998**, *52*, 1328−1338.

(7) Bruce, A.; Donoho, D.; Gao, H. Wavelet Analysis. *IEEE Spectrum* **1996**, *33* (10), 26−35.

(8) Bos, M.; Vrielink, J. A. M. The Wavelet Transform for Pre-Processing IR Spectra in the Idendification of Mono- and Di-substituted Benzenes. *Chemom. Intell. Lab. Syst.* **1994**, *23*, 115−122.

(9) Mallet, Y.; Coomans, D.; Vel, O. V. Recent Developments in Discriminant Analysis on High Dimensional Spectral Data. *Chemom. Intell. Lab. Syst.* **1996**, *35*, 157−173.

(10) Mallet, Y.; Coomans, D.; Kautsky, J.; Vel, O. V. Classification Using Adaptive Wavelets for Feature Extraction. *IEEE Trans. Pattern Anal. Mach. Intell.* **1997**, *19*, 1058−1066.

(11) Szu, H.; Telfer, B.; Garcia, J. Wavelet Transforms and Neural Networks for Compression and Recognition. *Neural Networks* **1996**, *9*, 695−708.

(12) Cai, C.; Harrington, P. B. Different Discrete Wavelet Transforms Applied to Denoising Analytical Data. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 1161−1170.

(13) Zupan, J.; Gasteiger, J. Neural Networks-A New Method for Solving Chemical Problems or Just a Passing Phase. *Anal. Chim. Acta* **1991**, *248*, 1−30.

(14) Wythoff, B. J. Backpropagation Neural Networks-A Tutorial. *Chem. Intell. Lab. Syst.* **1993**, *18*, 115−155.

(15) Fahlman, S. E.; Lebiere, C. *The Cascade-Correction Learning Architecture*; Carnegie Mellon University Technique Report CMU-CS- 90−100; Carnegie Mellon University: Pittsburgh, PA, August 1991.

(16) Harrington, P. B. Temperature-Constrained Cascade Correlation Networks. *Anal. Chem.* **1998**, *70*, 1297−1306.

(17) Barclay, V. J.; Bonner, R. F.; Hamiliton, I. P. Application of Wavelet Transforms to Experimental Spectra: Smoothing, Denoising, and Data Set Compression. *Anal. Chem.* **1997**, *69*, 78−90.

(18) Donoho, D. L. Nonlinear Wavelet Methods for Recovery of Signals, Densities, and Spectra from Indirect and Noisy Data. In *Proceedings of Symposia in Applied Mathematics: Different Perspectives on Wavelets*; Daubechies, I., Ed.; American Mathematical Society: Providence, RI, 1993; pp 173−205.

(19) Fahlman, S. E. Faster-Learning Variations on Back-Propagation: An Emprical Study. In *Proceedings of the 1988 Connectionist Models Summer School*; Touretzky, D., Hinton, G., Sejnowski, T., Eds.; Morgan Kaufmann: San Mateo, CA, 1989; pp 38−51.

(20) Sarle, W. S. Neural Network FAQ, Part 2 of 7: Learning. ftp://ftp.sas.com/pub/neural/FAQ.html (accessed March 1999).

(21) Steven Gollmer, DAUBWAVE package, http://www.cedarville.edu/dept/sm/smg/daubwave.zip (last accessed March 1999).

(22) Press, W. H.; Teukolsky, S. A.; Vettering, W. T.; Flannery, B. P. *Numerical Recipes in C*; Cambridge University Press: Cambridge, U.K., 1992.

CI9903253