Least-squares support vector machines for chemometrics: an introduction and evaluation

R.P. Cogdill[a,*] and P. Dardenne[b]
[a]*Duquesne University Center for Pharmaceutical Technology, 410 Mellon Hall, 600 Forbes Ave., Pittsburgh, PA 15282, USA. E-mail: cogdillr@duq.edu*

[b]*Centre de Recherches Agronomiques de Gembloux, Chaussée de Namur, B-5030 Gembloux, Belgium*

Support vector machines (SVM) are a relatively new technique for modelling multivariate, non-linear systems, which is rapidly gaining acceptance in many fields. There has been very little application or understanding of SVM methodology in chemometrics. The objectives of this paper are to introduce and explain SVM regression in a manner that will be familiar to the NIR and chemometrics community, and provide some practical comparisons between least-squares SVM regression and more traditional methods of multivariate data analysis. Least squares support vector machines (LS-SVM) were compared to partial least squares (PLS), LOCAL and artificial neural networks (ANN) for regression and classification using four, diverse datasets. LS-SVM was shown to be the most effective algorithm, and required the lowest number of calibration samples to achieve superior predictive performance.

Keywords: chemometrics, support vector machines, artificial neural networks, linear regression, non-parametric regression, radial basis function

Introduction

Linear methods of least-squares regression, partial least squares (PLS)[1] being the dominant method, are often used for chemometric modelling of near infrared (NIR) spectroscopic data. While PLS can be used to derive a satisfactory solution in most cases, in some situations a non-linear model is clearly required. Furthermore, experience has shown that even though a linear model may be adequate, the performance of some calibrations may be significantly improved with the use of a non-linear model.[2]

When faced with the task of creating a non-linear model, chemometricians basically have three options:

1) Transform the independent or dependent variables to linearise the problem, or fit a polynomial to the data.[1,2]
2) Develop local linear approximations to the solution,[3-8] or implement some other type of non-parametric strategy.[9,10]
3) Derive a global parametric model using a method of universal approximation, such as artificial neural networks (ANN).[7,8,11]

Although there are many useful methods within this set of options, an algorithm with the power of local and ANN methods, which retains the rationality of PLS is still being sought.

Support vector machines (SVM),[12–14] a semi-parametric, non-linear modelling technique, is rapidly gaining application in a number of fields. Only recently have SVM been applied to chemometrics[15,16] as a non-linear classification scheme. Support vector machine theory has been extended beyond classification tasks into the realm of multivariate function estimation, or non-linear regression. Little is known about the practical application of SVM regression to chemometrics; SVM methodology has generally been presented in ways quite foreign to most chemometricians. Furthermore, with a lack of comparative studies to illustrate its capabilities, there has been little motivation for experimentation with support vector machines within the NIR community.

The intent of this paper is to present SVM regression in a way that is more familiar to the fields of chemometrics and NIR spectroscopy. In doing so, wherever possible, the traditional terminology of support vector machines has been modified to better agree with the terminology of chemometrics. More esoteric details of SVM theory will be purposefully avoided in an effort to facilitate general understanding of the basic concepts. For a more detailed explanation of SVM theory, those interested should consult some of the many references available on kernel methods[14,17] and support vector machines.

Theory

Kernel regression

In the linear least-squares model, a vector of prediction coefficients (**b**), is derived to describe the relationship between a matrix of independent variables (**X**), and a vector dependent variables (**y**), with the minimum of squared prediction error ($\mathbf{e}^2$). Thus, the objective, or primal, function of least squares regression is simply:

$$\min(\mathbf{e}^2) = \min[\Sigma(\mathbf{y} - \hat{\mathbf{y}})^2] \tag{1}$$

The solution to this problem, the ordinary least squares equation, follows directly from calculus:

$$\mathbf{b} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{2}$$

This formula is more commonly referred to as multiple linear regression (MLR).[1] For the case of NIR spectroscopy, **X** would be a $[m \times n]$ matrix of spectral responses (spectra), with m samples and n measured responses (wavelengths) per sample:

$$\mathbf{X} = \begin{bmatrix} \chi_{1,1} & \chi_{1,2} & \chi_{1,3} & \cdots & \chi_{1,n} \\ \chi_{2,1} & \chi_{2,2} & \chi_{2,3} & \cdots & \chi_{2,n} \\ \chi_{3,1} & \chi_{3,2} & \chi_{3,3} & \cdots & \chi_{3,n} \\ \vdots & \vdots & \vdots & & \vdots \\ \chi_{m,1} & \chi_{m,2} & \chi_{m,3} & \cdots & \chi_{m,n} \end{bmatrix}$$

Thus, predictions ($\hat{\mathbf{y}}$) would be derived from a new $[1 \times n]$ spectrum ($\mathbf{x}_t$) by taking the inner (dot) product of the new spectrum, and the $[n \times 1]$ vector of coefficients:

$$\hat{\mathbf{y}} = \mathbf{x}_t\mathbf{b} \tag{3}$$

For NIR spectroscopy, the applicability of MLR is often limited by the co-linearity and rank deficiency within the calibration matrix, **X**. The problems were overcome with the addition of latent variable projection methods like principal components regression (PCR)[1] or PLS.

The traditional implementation of ordinary least squares regression is variable-centric; the solution is obtained by evaluating the distribution of data for each variable (wavelength). An equivalent solution could be obtained by taking a sample-centric view, where the relationship between every sample is characterised, and a hyperspace is defined using kernel substitution.[17–20] The sample–sample relationship is characterised (e.g. distance) by applying a kernel function. Thus, during ordinary least squares derivation [Equation (1)], the $[m \times n]$ matrix of spectral responses, **X**, is substituted by a $[m \times m]$ kernel matrix, **K**, where the i,j^{th} element of the matrix is the kernel function between samples i and j:

$$\mathbf{K} = \begin{bmatrix} k_{1,1} & k_{1,2} & k_{1,3} & \cdots & k_{1,m} \\ k_{2,1} & k_{2,2} & k_{2,3} & \cdots & k_{1,m} \\ k_{3,1} & k_{3,2} & k_{3,3} & \cdots & k_{3,m} \\ \vdots & \vdots & \vdots & & \vdots \\ k_{m,1} & k_{m,2} & k_{m,3} & \cdots & k_{m,m} \end{bmatrix}$$

A natural choice of kernel to model a linear relationship would then be the inner (dot) product, since it is computationally efficient and scales linearly with distance:

$$k_{i,j} = \mathbf{x}_i \cdot \mathbf{x}_j, \tag{4}$$

For nearly every case, though, Equation (1) cannot be solved after the substitution of **K** for **X,** since the substitution will result in an over-determined solution and $(\mathbf{K}^T\mathbf{K})^{-1}$ will be nearly singular. Just as in the case of any other ill-conditioned regression problem, however, PLS (or some other regularisation technique) can be used to successfully derive the solution, which takes the form of a $[m \times 1]$ vector of coefficients.

Predictions would be derived from a test $[1 \times n]$ spectrum ($\mathbf{x}_t$) by first calculating the kernel vector for the new spectrum ($\mathbf{k}_t$) as the kernel function between the test spectrum and the spectrum of each calibration sample, resulting in a new $[1 \times m]$ vector. The predicted value is the inner product of the kernel vector and the $[m \times 1]$ vector of coefficients:

$$\hat{\mathbf{y}} = \mathbf{k}_t\mathbf{b} \tag{5}$$

While this may be an interesting exercise, nonetheless, the solution derived would be the same as if PLS regression were simply performed on the original matrix of spectra, **X**. Indeed, the ideal number of latent variables is the same whether **X** or **K** is used, and it would appear as if nothing is to be gained by using kernel substitution. This is generally true when a linear (inner product) kernel is used, but the technique becomes quite useful when the linear kernel is replaced with a non-linear function. Depending on the choice of kernel function, it is possible to model non-linear processes, while retaining the benefits of linear least-squares regression. The form of the non-linearity that can be modelled depends on the shape of the kernel function. The Gaussian radial basis function (RBF) is an ideal kernel choice since it can be used to model functions of arbitrary nonlinearity:[19]

$$k_{ij} = e^{\left(\frac{-\left\|x_i - x_j\right\|_2^2}{\sigma^2}\right)} \tag{6}$$

While the inner product kernel is a linear measure of similarity between two vectors, the RBF kernel is conceptually a non-linear measure of similarity. The adjustment of the variance parameter (σ^2) changes the width of the kernel, and the degree of non-linearity that can be modelled. Vectors that are very similar produce an RBF output near 1, and as the vectors become progressively dissimilar, the RBF output asymptotically approaches zero (Figure 1). As σ^2 is increased, the kernel becomes wider, forcing the model towards a linear solution. Walczak and Massart[20] coupled PLS regression with RBF kernel substitution to successfully model non-linear processes using NIR spectroscopy.

Support vector machine regression

When they initially began to develop support vector machines, Vapnik and Chervonenkis focused their work on

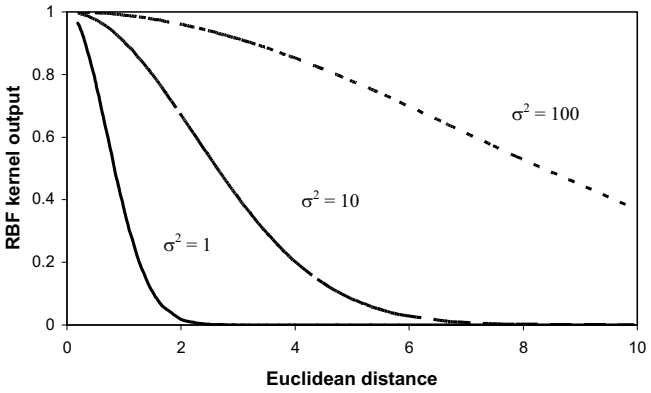

Figure 1. Graphical illustration of the relationship between Euclidean distance and RBF kernel output. As the Euclidean distance between vectors decreases, RBF similarity approaches unity. Increasing the variance (σ^2) is roughly analogous to increasing the neighbourhood size during kNN calculation.

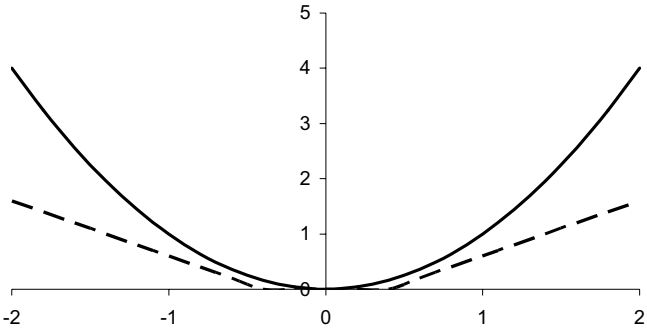

Figure 2. Visual comparison of least-squares loss function (solid line) and the e-insensitive loss function (dashed line, $\varepsilon = 0.4$).

creating a robust algorithm for classification and discrimination.[12] It was not until 1997 that Vapnik extended the theory to non-linear regression using the SVM framework.[13] While the basics of SVM regression are shared with kernel regression, as described above, SVM theory proposes some changes in the method of optimising the coefficient vector **b** (Lagrangian multipliers in the case of SVM and LS-SVM). There are basically two major differences between kernel regression and the Vapnik–Chervonenkis SVM.

First, rather than seeking to minimise prediction error only, the SVM objective function has been augmented with terms to minimise the rms magnitude of the coefficient vector, **b**, referred to as model complexity. The proportional influence of prediction error and model complexity on the objective function optimisation is controlled by a regularisation constant (γ). With these changes, the original objective function [Equation (1)] is replaced by the primal-dual form:

$$\min[2^{-1}\Sigma(\mathbf{y}-\hat{\mathbf{y}})^2]+\gamma\Sigma(2^{-1}\mathbf{b}^{\mathrm{T}}\mathbf{b})] \quad (7)$$

Thus, as γ is increased, more emphasis is placed on reducing the rms magnitude of the model coefficients. In this case, **b** indicates the vector of model coefficients in the variable-centric space. Following Lagrangian substitution (during SVM training), the **b**-coefficients will be in the sample-centric space. This is characteristically similar to the concept of regularisation in ridge regression and neural network training.[8,11]

The second major difference is in the form of error function. In an effort to increase the robustness of training, the least-squares error criterion [Equation (1)] is replaced with the so-called e-insensitive error of generalisation:

$$\xi_i = \begin{cases} 0, & \text{if } |y_i - y_i| \le \varepsilon \\ |y_i - y_i| - \varepsilon & \text{otherwise} \end{cases} \quad (8)$$

The objective function now becomes:

$$\min[2^{-1}\Sigma(\xi)+\gamma\Sigma(2^{-1}\mathbf{b}^{\mathrm{T}}\mathbf{b})] \quad (9)$$

The e-insensitive error function has some important consequences for model training. Primarily, during model training, any residual error of magnitude less than ε is ignored as zero (Figure 2). More simply, e-insensitive error implies that calibration error cannot be significantly better than the error in the training data, and that any solution with lower error has likely over-fit the training data. Also, for residual errors larger than ε, its absolute value is summed, rather than its square, which tends to limit the influence of outliers on training. Finally, because of the inequality constraints imposed on the optimisation, calibration samples with residual error less than ε will be given a coefficient (**b**) of zero, which means it has no influence on prediction, and can be omitted from the calibration. The samples whose **b**-coefficients are non-zero are referred to as *support vectors*. This process is conceptually analogous to thresholding the coefficients of a PLS model for automatic variable selection.

Though the use of e-insensitive loss (and the inequality constraints it imposes) precludes deriving **b** by solving a linear system, the model can be optimised in the space of Lagrangian multipliers by using quadratic programming. While optimisation by quadratic programming is slower than least squares methods, it is still a convex, deterministic process, guaranteed to converge to a global minimum. Unlike ANN training, which generally uses error backpropagation, for example, there is no danger of training terminating in a local minimum, and the same solution will always be achieved.

Least-squares support vector machine regression

To simplify SVM regression, Suykens *et al.* proposed an alternate formulation of the SVM strategy called the least-squares support vector machine (LS-SVM).[14,21] While the primal-dual form of the objective function was retained [Equations (7 and 9)], a squared loss function replaced the e-insensitive loss function, from which equality constraints follow (instead of inequality constraints). While foregoing the benefits of automatic sparseness (all LS-SVM **b**-coefficients will be non-zero), and perhaps some insensitivity to outliers, the LS-SVM can be trained much more efficiently. After constructing the Lagrangian, setting equality constraints,

and simplifying, a linear Karush–Kuhn–Tucker (KKT) system results:

$$\left[\begin{array}{c|c} 0 & \mathbf{1}_m^{\mathbf{T}} \\ \hline \mathbf{1}_m & \mathbf{K}+\mathbf{I}/\gamma \end{array}\right]\left[\begin{array}{c} b_0 \\ \hline \mathbf{b} \end{array}\right]=\left[\begin{array}{c} 0 \\ \hline \mathbf{y} \end{array}\right] \qquad (10)$$

Where, $\mathbf{I}$ refers to an $[m \times m]$ identity matrix, γ is the regularisation constant, $\mathbf{1}_m$ is a $[m \times 1]$ vector of ones, $\mathbf{y}$ is the vector of reference values, $\mathbf{b}$ is the vector of model coefficients, b_0 is the model bias term and $\mathbf{K}$ is the $[m \times m]$ kernel matrix.

Following transformation into a positive definite form, the LS-SVM KKT system can subsequently be solved using many methods for solving large sets of linear equations,[22] such as conjugate gradient descent. The LS-SVM solution, $\mathbf{b}$, follows from the solution of a system based on $\mathbf{K}$, since $\mathbf{K}$ can be thought of as a sample–sample correlation matrix. Furthermore, along the same lines, the matrix quantity $(\mathbf{K}+\mathbf{I}/\gamma)$ bears striking resemblance to the defining operation of linear ridge regression.[23] Indeed, one might loosely consider LS-SVM regression as ridge regression in the sample space (as opposed to variable space). Also, the relationship between LS-SVM/SVM and local methods such as LOCAL,[3–5] LWR,[6–8] kNN[9] and CARNAC,[10] should be noted. When considering the case of deriving predictions for a new sample, by taking the inner product of the test sample's kernel vector and the regression vector, a sort of non-linearly-weighted average of the reference values is performed. Thus, support vector machine methodology can be seen as formalising the local relationship between data points and reference values by defining a set of global, fixed coefficients.

The LS-SVM derivation has not been shown to exhibit any significant loss in performance relative to the Vapnik–Chervonenkis SVM,[14] yet it is much more efficient in optimisation. This is an important result since the time of optimisation for both the SVM, and LS-SVM increases with the square of the number of training samples, and linearly with the dimension of the training samples (number of wavelengths). To implement the Vapnik–Chervonenkis SVM algorithm, after choosing suitable pre-processing, the user must specify three parameters: σ^2, ε and γ; the user is only required to specify two parameters (σ^2 and γ) to implement the LS-SVM algorithm.

Experimental

Despite the theoretical underpinnings and optimistic derivations that accompany any new algorithm, the matters of defining the guidelines and scope of its application, and illustrating that the algorithm works in practice always remain. While it is rarely feasible to prove the general validity, much less superiority, of a method; side-by-side comparisons (with other techniques) are often useful in allowing others to decide for themselves whether or not the method warrants further investigation with their own applications. The remaining portions of this paper are devoted to providing examples of SVM methodology in practice, with comparisons to other techniques.

Datasets

For the performance comparison, datasets of NIR spectra and reference values were compiled for four, diverse products: apples, meat, corn and animal feed. Three of the datasets (apples, meat and corn) were used for regression analysis. Each consisted of spectra from a typical NIR analyser, and each had multiple analytes. The fourth dataset (animal feed) was a discriminant analysis problem with the objective of detecting meat and bone meal contamination in ruminant feed; the spectra for this dataset were collected using an imaging spectrometer.[24]

Each dataset consisted of calibration and test subsets. For the apple dataset, the subsets were randomly split from the same pool of samples. For the meat dataset, a portion of the dataset was randomly selected for parameter optimisation; then, after the optimal parameters were set for each algorithm, the entire dataset was broken into seven subsets according to the year in which the sample was drawn. Predictions were derived for each subset by calibrating on the remaining subsets, using the previously-determined parameter settings. While the process is essentially cross-validation, the results are reported as standard error of prediction (*SEP*) since there is some degree of independence between each subset. The corn test set was drawn from independent sources, including a different harvest and a different analyser. For the feed analysis dataset, the test set was drawn from separate batches and data collection sessions. For all datasets, no predictions were drawn from the test set until model optimisation was complete; model performance was compared on the basis of *SEP*. In all, 12 comparisons would be performed. Details for each dataset are shown in Table 1.

Software

Four regression methods were tested during the comparison: MPLS, LOCAL and ANN (ISI II v1.50, Infrasoft International, LLC, Port Matilda, PA, USA), and the LS-SVM[14] toolbox for MATLAB. The LS-SVM calculations were carried out in MATLAB 6.5 (The Mathworks, Inc., Natick, MA, USA). While it would be desirable to include the Vapnik–Chervonenkis SVM, freely-available regression software had not been found with suitable speed to be feasible for inclusion in the comparison. Moreover, informal tests with the SVM regression code that was available did not suggest any advantage over LS-SVM.

Calibration optimisation

The optimisation of MPLS, LOCAL and ANN calibrations were carried out by Dr Pierre Dardenne at CRAGx, Gembloux, Belgium. Selection of pre-processing, removal of training outliers and parameter optimisation were completed for each algorithm and dataset. Depending on the capabilities of the individual algorithms, optimisation was

Table 1. Product and constituent datasets used for performance comparison.

		Samples (*n*)		Instrument description		Wavelength range		
Dataset	Product	Cal.	Test	Make	Model	MIN (nm)	MAX (nm)	inc (nm)
Apples	Sucrose	601	139	NIR Systems	6500	796	2416	2
	pH	514	118			1104	2300	2
	Acidity	519	120			800	2498	2
	Firmness	506	114			800	1998	2
Meat†	Moisture	691		NIR Systems	5000	1300	2398	2
	Protein	658						
	Fat	651						
	Collagen	472						
Feed		3603	654	Spectral Dimensions	MatrixNIR	1200	1700	10
Corn	Moisture	891	429	Foss-Tecator	Infratec 1229, 1241	850	1048	2
	Protein	907	429					
	Oil	899	429					

†Meat dataset was tested using group-vs-group cross-validation

performed with cross-validation, or by splitting the calibration into a training and validation set.

To reduce the likelihood of experimental bias, the optimisation of LS-SVM was carried out independently by R.P. Cogdill, using the same datasets. Because the LS-SVM toolbox was not designed with chemometrics in mind, and because little was understood about applying LS-SVM at the beginning of this study, the selection of pre-processing, spectral window (Table 1) and removal of training outliers was performed using PLS regression and the PLS_Toolbox 3.0 (Eigenvector Research, Inc., Manson, WA, USA). Thus, the performance of LS-SVM may have been limited, since it isn't known if the algorithm is affected by pre-processing in the same manner as PLS. Once the pre-processing had been applied, and the calibration set was prepared, the LS-SVM γ and σ^2 parameters were optimised using cross-validation. The LS-SVM cross-validation was performed using MATLAB functions custom-written for this paper, since the cross-validation procedure supplied with the LS-SVM toolbox often suggested overly optimistic γ and σ^2 settings, leading to over-fitting of the training data.

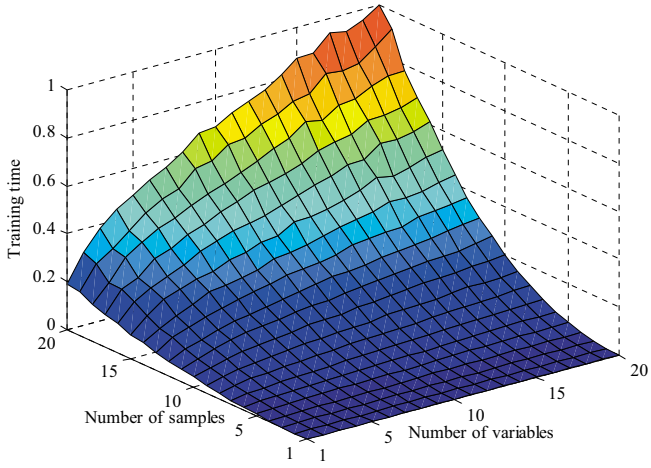

Figure 3. Graphical illustration of the importance of dataset size, and number of variables, in determining the time required to calculate the LS-SVM solution. Training time increases with the square of dataset size, and linearly with the number of variables (note: time has been scaled to fit the interval [0 1]).

Results and discussion

The apples dataset was the first to be analysed using LS-SVM. Though the spectra had been truncated, training with many hundred wavelengths and data points was prohibitively slow for feasible parameter optimisation. As was stated earlier, it was found that training time increases with the square of the number of training samples, and linearly with the number of independent variables (Figure 3), when γ and σ^2 are held constant. With this in mind, the decision was made to replace the spectra with (an excess of) PLS factors when there were too many wavelengths to feasibly include the entire spectrum. PLS factors were used with the apples and meat datasets (Table 2); for the corn dataset, whose spectra consisted of only 100 variables, using the entire spectrum was feasible.

Indeed, using the complete spectrum was superior to PLS compression during both optimisation and testing of the corn LS-SVM calibration models. This result was not surprising since the application of the RBF kernel [Equation (6)] is not

Table 2. Performance comparison results and LS-SVM training parameters.

		SEP				LS-SVM parameters		
Dataset	Product	MPLS	LOCAL	ANN	LS-SVM	Input	γ	σ_2
Apples	Sucrose	0.37	0.34	0.33	**0.32**	19 lv	3000	3000
	pH	0.12	0.11	0.13	**0.10**	20 lv	8000	4000
	Acidity	1.47	1.47	1.36	**1.28**	20 lv	4000	3000
	Firmness	1.00	1.03	1.02	**0.87**	20 lv	6250	4500
Meat	Moisture	0.62	0.85	**0.61**	0.69	15 lv	5000	1000
	Protein	0.88	1.00	1.05	**0.81**	20 lv	3500	7000
	Fat	0.52	0.56	0.91	**0.47**	10 lv	4000	1000
	Collagen	0.87	1.94	1.08	**0.71**	16 lv	6500	4500
Corn	Moisture	0.73	0.75	0.59	**0.57**	100 λ	5000	2000
	Protein	0.41	0.45	0.43	**0.36**	100 λ	3000	3000
	Oil	0.41	0.40	0.47	**0.40**	100	3000	4000

unlike a basis function which consolidates the multivariate relationship between two vectors into a single scalar, *k*. Though extraneous information may be carried along, the signal-to-noise ratio (SNR) of *k* is likely to remain quite high. Since the LS-SVM function, as implemented, included no means of explicitly weighting variables, using latent variable compression may actually reduce the SNR of the kernel substitution. Equal weight is applied to higher factors, which may contain little relevant information. Moreover, latent variable compression may constrain the data to a subspace that is less effectively modelled by kernel substitution. Perhaps a better solution would be to use stepwise variable selection, or some method of weighting individual variables during kernel substitution, known as *automatic relevance determination* (ARD) among practitioners of kernel methods.[14]

Optimisation of the LS-SVM γ and σ^2 parameters was a manageable task, similar to the process of selecting the number of factors for a PLS model, but was complicated by the two-dimensional nature of the problem. An example optimisation surface is shown in Figure 4. While it may seem that automatic optimisation is possible, by way of response-surface methodology, the chemometrician must understand the relationship influence of γ and σ^2 on model accuracy and robustness. Conceptually, an increase in γ is analogous to an increase in the number of factors in a PLS model. Just as the rms magnitude of PLS **b**-coefficients, as well as the likelihood of over-fitting the training data, increase with added latent variables, the rms magnitude of LS-SVM model coefficients, and risk of over-fit, increase along with γ (Figure 5). Adjustment of σ^2, on the other hand, is similar to adjusting the neighbourhood size of a local/kNN model. As σ^2 is increased, the resulting LS-SVM becomes confined to a linear model; as σ^2 is decreased, differences between similar training samples are increasingly resolved, resulting in a tighter

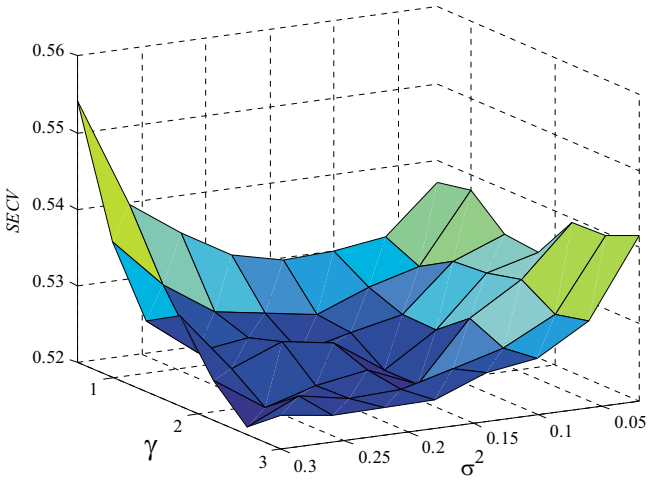

Figure 4. Example parameter optimisation response surface. This figure would be analogous to a plot of *SECV*, or *SEP* versus factors during PLS optimisation.

fit of the training data. Practically, the optimum level of γ seems to be related to the density of the available training data, while σ^2 seems to be related to the SNR and non-linearity of the calibration problem at hand. During optimisation with cross-validation, any increase in γ or decrease in σ^2 should be weighed cautiously against the significance of any improvement in perceived performance.

Even though RBF-PLS was not included in the comparative study for this paper, some tests were done in preliminary phases of the work to compare its performance to LS-SVM. Though it was simpler and more familiar to set the complexity of the RBF-PLS model, which entailed selecting a reasonable number of PLS factors, rather than tuning a γ parameter;

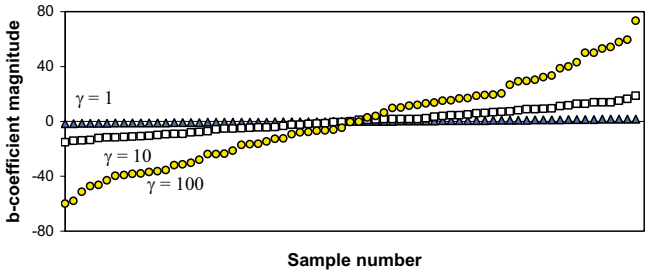

Figure 5. **b**-coefficient vector versus level of γ. As γ increases, so does the rms magnitude of the **b**-coefficient vector; leading to a "tighter" fit of the training data, and increasing the likelihood of over-fit. The level of γ which best exploits the trade-off between fitness and robustness must be determined by the chemometrician.

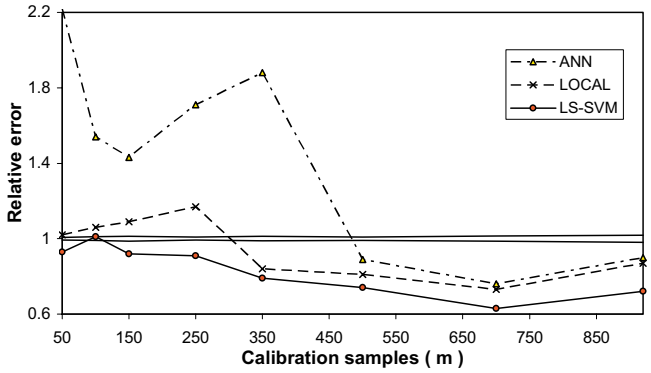

Figure 6. Calibration dataset size versus error (relative to PLS) for ANN, LOCAL and SVM; relative errors greater than one indicate poorer performance, while relative errors less than one indicate improved performance. The confidence interval for PLS is shown as solid lines immediately above and below one. In this case, LS-SVM requires far fewer samples than ANN and LOCAL to achieve superior performance.

the process of selecting σ^2 was the same for both algorithms. The performance of RBF-PLS was generally on-par with, or better than PLS regression, which is in line with what was found by Walczak and Massart.[20] However, LS-SVM always performed better than RBF-PLS in prediction with independent samples. While both algorithms use the same basic model formulation, the LS-SVM **b**-coefficients were orders of magnitude less than the same coefficients derived using RBF-PLS. It can be surmised that the LS-SVM solution is more able to generalise to future datasets.

The results of the regression tests (apples, meat and corn) are shown in Table 2. While LS-SVM was superior to MPLS, LOCAL and ANN, in all but two tests, it is more surprising that LS-SVM performed best even for calibrations that are generally considered to be linear, such as protein in corn. For the ruminant feed discriminant analysis problem, LS-SVM misclassified 6 out of 654 samples, while MPLS and ANN misclassified 9 and 15 samples, respectively; LOCAL was not found to be applicable to the problem. For the discriminant analysis problem, LS-SVM was trained using 12 PLS factors, with γ and σ^2 levels of 6000 and 1000, respectively.

Because of its large number of calibration samples, the ruminant feed dataset presented a difficult challenge for optimisation of the LS-SVM parameters. Even with the data compressed to 12 PLS factors, cross-validation with the entire dataset was too slow to be feasible (given the available CPU time). Instead, the cross-validation was reversed such that, when the calibration data was broken up into subsets, models were built using only one subset, and predictions were made using the remaining subsets. After the parameter values were selected, using the reversed cross-validation, the final calibration model was trained using the entire calibration set. Though the resulting model performance seems satisfactory, it is yet unknown whether more appropriate parameter settings could have been found using cross-validation in the conventional manner. This issue is likely more important for SVM methods, since the omission of training samples is equivalent to omitting an independent variable in the model. Cross-validation may not be relevant for LS-SVM training; perhaps some other method of determining the correct parameter settings could be more effective.

While it is apparent that (without some method of reducing training time) application of LS-SVM may not be feasible for extremely large-scale spectroscopic modelling problems, questions remain as to how much data is required for (comparatively) good performance, and is there limit to database size beyond which LS-SVM performance will fail to improve? While the latter question is part of a series of questions concerning the application of large-databases to chemometrics and NIR spectroscopy, the former question is more easily addressed. Using the corn protein dataset, a test was devised whereby progressively smaller subsets were randomly drawn from the original set of 920 samples. For each subset, PLS, LOCAL, ANN and LS-SVM calibrations were derived (with update of the parameter settings) and tested using the same set of 429 independent test samples.

The results are shown in Figure 6, with calibration database size on the horizontal axis, and predictive performance on the vertical axis. Predictive performance is shown as a ratio, relative to the PLS results; a result above 1 indicates poorer predictive ability than PLS (using the same calibration set), and a result below 1 indicates better performance. Upper and lower confidence limits are included for the PLS results. The performances of LOCAL and ANN, relative to PLS, were much as expected; both algorithms required somewhere between 250 and 500 samples before their performance began to significantly improve on that of PLS. Surprisingly, regardless of the size of calibration dataset, LS-SVM always performed better than LOCAL and ANN, and in only one case was not significantly better than PLS.

Conclusions

Given the results of this work, some conclusions may be made regarding the application of least-squares support vector machines to chemometrics:

- LS-SVM was generally superior to MPLS, LOCAL and ANN in predictive performance.
- A large sample database is not required for calibration development using LS-SVM regression.
- Since model training time increases rapidly with the addition of calibration data, without additional modification, SVM methods may be precluded from extremely large-scale calibration problems.
- While the effect of various pre-processing methods on LS-SVM performance was not tested; latent variable compression was not always necessary during LS-SVM calibration. The form of the RBF kernel [Equation (6)] suggests that some form of scatter correction, or baseline removal will usually be necessary for successful implementation of RBF methods using full spectra.
- The proper selection of tuning parameters (γ and σ^2) is critical to avoid over-fitting during LS-SVM training.
- The form of the LS-SVM model solution (**b**-coefficient vector) does not lend itself well to interpretation of the model. Further efforts will be needed to create useful methods of model interpretation and validation.

Though it is unlikely the optimism of these results will apply to every situation that may arise in the course of NIR calibration development, they certainly indicate that SVM methodology has a place in NIR spectroscopy and chemometrics. Just as for any traditional chemometric technique, proper use of kernel methods, including LS-SVM, requires some understanding and experience; with the power to model virtually any non-linear function, the ever-present danger of over-fitting, and subsequent poor predictive performance, places an even greater demand on the skills of the chemometrician.

References

1. H. Martens and T. Næs, *Multivariate Calibration*. Wiley, New York (1993).
2. S. Wold, N. Kettaneh-Wold and B. Skagerberg, *Chemometr. Intell. Lab. Sys.* **7,** 53 (1989).
3. J.S. Shenk and M.O. Westerhaus, *Crop Sci.* **31,** 469 (1991).
4. J.S. Shenk, P. Berzaghi and M.O. Westerhaus, *J. Near Infrared Spectrosc*. **5,** 223 (1997).
5. P. Berzaghi, P.C. Flinn, P. Dardenne, M. Lagerholm, J.S. Shenk, M.O. Westerhaus and I.A. Cowe, in *Near Infrared Spectroscopy: Proceedings of the 10th International Conference*, Ed by A.M.C. Davies and R.K. Cho. NIR Publications, Chichester, UK, p. 107 (2002).
6. T. Næs, T. Isaksson and B. Kowalski, *Anal. Chem.* **62,** 664 (1990).
7. Z. Wang, T. Isaksson and B. Kowalski, *Anal. Chem.* **66,** 249 (1994).
8. S. Roussel, C.L. Hardy, C.R. Hurburgh, Jr and G.R. Rippke, *Appl. Spec*. **55(10),** (2001).
9. B.G.M. Vandeginste, D.L. Massart, L.M.C Buydens, S. de Jong, P.J. Lewi and J. Smeyers-Verbeke, in *Handbook of Chemometrics and Qualimetrics: Part B*. Elsevier Science, Amsterdam, p. 223 (1998).
10. A.M.C. Davies, H.V. Britcher, J.G. Franklin, S.M. Ring, A. Grant and W.F. McClure, *Mikrochim. Acta (Wien)* **I**, 61 (1988).
11. F. Girosi, M. Jones and T. Poggio, *Neural Computation* **7,** 219 (1995).
12. V. Vapnik and A. Chervonenkis, *Automation and Remote Control*. **24,** 774 (1963).
13. V. Vapnik, in *Nonlinear Modelling: Advanced Black-box Techniques*, Ed by J.A.K. Suykens and J. Vandewalle. Kluwer Academic Publishers, Boston, p. 55 (1998).
14. J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor and J. Vandewalle, *Least-Squares Support Vector Machines*. World Scientific, Singapore (2002).
15. A.I. Belousov, S.A. Verzakov and J. von Frese, *J. Chemometrics*. **16,** 482 (2002).
16. A.I. Belousov, S.A. Verzakov and J. von Frese, *Chemom. Intell. Lab. Syst.* **64(1),** 15 (2002).
17. www.kernel-machines.org, April 30, 2003.
18. W. Wu, D.L. Massart and S. de Jong, *Chemometr. Intell. Lab. Syst.* **36,** 165 (1997).
19. T. Poggio and F. Girosi, *Science*. **247,** 978 (1990).
20. B. Walczak and D.L. Massart, *Anal Chim. Acta.* **331,** 177 (1996).
21. J.A.K. Suykens, Neural Network World, *Special Issue on Fundamental Issues in Control*. **7(2–3),** 311 (2001).
22. A. Greenbaum, *Iterative Methods for Solving Linear Systems*. SIAM, Philadelphia (1997).
23. H.R. Draper and H. Smith, *Applied Regression Analysis*, Second Edition. Wiley, New York (1981).
24. A.M. Renier, V. Baeten, G. Sinnaeve and P. Dardenne, in *Near Infrared Spectroscopy: Proceedings of the 11th International Conference*, Ed by A.M.C. Davies and A. Garrido-Varo. NIR Publications, Chichester, UK (in press).

Received: 3 July 2003
Revised: 23 January 2004
Accepted: 9 March 2004
Web Publication: 5 July 2004