Research Article

JOURNAL OF CHEMOMETRICS

Received: 26 November 2012, Revised: 9 January 2013, Accepted: 11 January 2013, Published online in Wiley Online Library: 20 February 2013

(wileyonlinelibrary.com) DOI: 10.1002/cem.2492

Sparse models by iteratively reweighted feature scaling: a framework for wavelength and sample selection

Erik Andries[a,b*]

In the past decade, there has been an increase in the use of sparse multivariate calibration methods in chemometrics. Sparsity describes a parsimonious state of model complexity and can be defined in terms of a subset of samples or covariates (e.g., wavelengths) that are used to define the calibration model. With respect to their classical counterparts such as principal component regression or partial least squares, sparse models are more easily interpretable and have been shown to exhibit non-inferior prediction performance. However, sparse methods are still not as fast as the classical methods in spite of recent numerical advances. In addition, for many chemometricians, sparse methods are still "black-box" algorithms whose internal workings are not well understood. In this paper, we describe a simple framework whereby classical multivariate calibration methods can be iteratively used to generate sparse models. Moreover, this approach allows for either wavelength or sample sparsity. We demonstrate the effectiveness of this approach on two spectroscopic data sets. Copyright © 2013 John Wiley & Sons, Ltd.

Keywords: multivariate calibration; sparsity; wavelength selection; sample selection; least absolute shrinkage and selection operator (LASSO); Tikhonov regularization (TR); support vector regression (SVR)

1. INTRODUCTION

In applications such as near-infrared spectroscopy, wavelength selection (sometimes known as feature or variable selection) attempts to find only those relevant variables useful for prediction in a multivariate calibration (MC) model. There are many wavelength selection methods associated with regression-based MC models [1,2].

Classical MC methods—ridge regression (RR), partial least squares (PLS), and principal component regression (PCR)—shrink the size of the regression vector, but they do not totally suppress any of the regression coefficients to zero, and as a result, all wavelengths are used in the prediction of unknown samples. Recently, sparse methods have become increasingly widespread in practice because they simultaneously build a regression (or model) vector and perform wavelength selection by shrinking many regression coefficients to zero [3–11]. Algorithmic advances have also made sparse methods computationally tractable for medium-to-large-sized data sets [12–18]. Despite these advances, sparse methods are still not as fast and efficient as classical MC methods, especially in high-dimensional, low-sample-size settings. The intent of this paper is to provide a unified framework whereby many wavelength selection methods can be recast as an iterative procedure where off-the-shelf, classical MC methods are performed per iteration.

The paper is organized as follows. Section 2 discusses wavelength selection via feature scaling. Section 3 details how feature scaling can be re-appropriated for sample selection. Section 4 outlines algorithmic implementation and model selection. Section 5 discusses the results, and Section 6 is the conclusion.

We now describe our notation. Lowercase and uppercase symbols that are not boldface represent scalars (x or P). Lowercase and uppercase boldface symbols represent column vectors ($\mathbf{x}$) and matrices ($\mathbf{X}$), respectively. The superscripted symbols ^{T}, $^{-1}$, and $^+$ indicate the transpose, inverse, and pseudoinverse, respectively, of a vector or matrix. A vector of n ones or zeros is indicated by $\mathbf{1}_n$ and $\mathbf{0}_n$, respectively, whereas $\mathbf{I}_n$ represents the identity matrix of dimension n. A $n \times d$ matrix $\mathbf{A}$ can be formed by concatenating its d column vectors $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_d]$, whereas a diagonal matrix is indicated via the "diag" notation, for example, $\mathbf{I}_n = \mathrm{diag}(\mathbf{1}_n) = \mathrm{diag}(1, 1, \ldots, 1)$. The element associated with the ith row and jth column of the matrix $\mathbf{A}$ will be denoted in two ways: a_{ij} or $(A)_{ij}$. In this paper, the $n \times d$ matrix $\mathbf{X}$ represents calibration spectra of d absorbance measurements across n samples such that $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^T$ where $\mathbf{x}_j = [x_{j1}, x_{j2}, \ldots, x_{jd}]^T$. The vector $\mathbf{y} = [y_1, y_2, \ldots, y_n]^T$ represents the response variables (e.g., analyte concentrations) across samples. Given $\mathbf{X}$ and $\mathbf{y}$, one attempts to infer the model or regression vector $\mathbf{b} = [b_1, b_2, \ldots, b_d]^T$ that relates $\mathbf{X}$ to $\mathbf{y}$. In this paper, we

* *Correspondence to: E. Andries, Center for Advanced Research Computing, University of New Mexico, Albuquerque, NM 87106, USA.*
E-mail: erik.andries@gmail.com

a *E. Andries*
Center for Advanced Research Computing, University of New Mexico, Albuquerque, NM 87106, USA

b *E. Andries*
Department of Mathematics, Central New Mexico Community College, Albuquerque NM 87106, USA

are interested in the L_1 and L_2 vector norms that measure the size or length of a vector $\mathbf{x} = [x_1, \ldots, x_d]^T$. The L_1 and L_2 vector norms are designated by the symbols $||\mathbf{x}||_1$ and $||\mathbf{x}||_2$, respectively, and are defined as

$$||\mathbf{x}||_1 = |x_1| + \cdots + |x_d| \quad \text{and} \quad ||\mathbf{x}||_2 = \sqrt{x_1^2 + \cdots + x_d^2}$$

We will refer to these norms as the one-norm and two-norm.

It is standard procedure in chemometrics to mean center the calibration data. If $\bar{\mathbf{x}} = 1/n\,(\mathbf{X}^T\mathbf{1}_n)$ and $\bar{y} = 1/n\,(\mathbf{y}^T\mathbf{1}_n)$ denote the mean spectrum and mean response, respectively, of the calibration samples, then mean-centering is accomplished by

$$\mathbf{X} \leftarrow \mathbf{X} - \mathbf{1}_n\bar{\mathbf{x}}^T \quad \text{and} \quad \mathbf{y} \leftarrow \mathbf{y} - \mathbf{1}_n\bar{y}$$

(The left arrow symbol $\leftarrow$ indicates that the uncentered data is being reassigned with its mean-centered counterparts.) Prediction on an unseen spectrum $\mathbf{z}$ is then given by $f(\mathbf{z}) = (\mathbf{z} - \bar{\mathbf{x}})^T\,\mathbf{b} + \bar{y}$.

2. WAVELENGTH SELECTION BY FEATURE SCALING

In spectroscopy, the number of wavelengths is often on the order of hundreds or thousands, and as a result, there is a need to distinguish between meaningful and spurious wavelengths. The search for a predictive set of wavelengths is a computationally intensive undertaking.

2.1. Least absolute shrinkage and selection operator methods

Tikhonov regularization (TR) refers to a general class of methods that shrink regression coefficients by adding a penalty term to the minimization problem associated with ordinary least squares regression

$$\text{minimize } h(\mathbf{b}), \quad h(\mathbf{b}) = \frac{1}{2}||\mathbf{Xb} - \mathbf{y}||_2^2 + P_q(\mathbf{b}) \tag{1}$$

where the penalty term $P_q(\mathbf{b})$ generally takes two forms:

$$P_q(\mathbf{b}) = \begin{cases} \lambda||\mathbf{Lb} - \mathbf{g}||_1, & q = 1 \\ \frac{\lambda^2}{2}||\mathbf{Lb} - \mathbf{g}||_2^2, & q = 2 \end{cases} \tag{2}$$

Here, $\mathbf{L}$ and $\mathbf{g}$ are a $p \times d$ matrix and p-dimension vector, respectively. When $\mathbf{L} = \mathbf{I}_d$, $\mathbf{g} = \mathbf{0}_d$ and $q = 2$ (the two-norm penalty formulation), Equation (1) corresponds to what is traditionally known as RR [19,20].

Recently, one-norm formulations ($q = 1$) have gained traction in chemometrics. When $\mathbf{L} = \mathbf{I}_d$, $\mathbf{g} = \mathbf{0}_d$, and $q = 1$, Equation (1) corresponds to what is commonly known as "the LASSO" [7]. Although both $q = 1$ and $q = 2$ result in small-norm regression vectors, sparse regression vectors arise only when $q = 1$, $\mathbf{L}$ is diagonal, and $\mathbf{g} = \mathbf{0}_d$. The shooting algorithm of Fu gives a simple and insightful explanation of how the LASSO "zeros-out" regression coefficients [12]. The general idea of using one-norm penalty methods for feature selection dates back to the early 1970s in the geophysics literature [3–6]. Two decades later in statistics, the idea of using one-norm penalty methods for regression was popularized by Robert Tibshirani who coined the acronym LASSO that stands for "least absolute shrinkage and selection operator" [7]. Although wavelength selection is the primary motivation for using TR when $q = 1$, variants of the TR formulation earlier for $q = 1$ and/or $q = 2$ have recently been shown to be effective for calibration maintenance and transfer [21–24].

The original LASSO problem was solved by quadratic programming, a nonlinear optimization procedure whose computational cost is prohibitive for medium-to-large-sized data sets [7]. To circumvent this computational bottleneck, algorithmic advances—coordinate descent methods [12,14], least angle regression (LAR)[13], projection and sub-gradient methods [17], and multiplicative updates [18]—have made the LASSO and their variants feasible for larger problems. Despite these advances, LASSO methods still do not scale as well as classical MC methods in high-dimensional settings. Hence, we seek to approximate or mimic the sparsity-inducing properties of the LASSO by using classical MC methods within a simple iterative framework.

2.2. Iteratively reweighted feature scaling

Multivariate calibration involves relating $\mathbf{y}$ (analyte concentrations) to $\mathbf{X}$ (spectroscopic measurements) by

$$\mathbf{Xb} = \mathbf{y} + \mathbf{e} \tag{3}$$

via the model vector $\mathbf{b}$. The $n \times 1$ vector $\mathbf{e}$ symbolizes normally distributed errors with zero mean and covariance matrix $\sigma\mathbf{I}_n$. Solving Equation (3) (ignoring the $\mathbf{e}$ term) is equivalent to solving

$$\mathbf{\Phi}\boldsymbol{\beta} = \mathbf{y} \quad \text{where} \quad \mathbf{\Phi} = \mathbf{XF} \quad \text{and} \quad \mathbf{F}\boldsymbol{\beta} = \mathbf{b} \tag{4}$$

such that $\mathbf{F}$ is invertible. For our purposes, we will restrict $\mathbf{F}$ to be a diagonal matrix $\mathbf{F} = \text{diag}(f_1, \ldots, f_d)$ where $f_i \neq 0$. After first solving $\mathbf{\Phi}\boldsymbol{\beta} = \mathbf{y}$ for $\boldsymbol{\beta}$ in Equation (4), we recover the original regression vector $\mathbf{b}$ via back-substitution: $\mathbf{b} = \mathbf{F}\boldsymbol{\beta}$. The matrix $\mathbf{\Phi} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_d]$ is a rescaled version of $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_d]$ because $\boldsymbol{\phi}_i = \mathbf{x}_i f_i$, $i = 1, \ldots, d$. Likewise, $\mathbf{b}$ is a rescaled version of $\boldsymbol{\beta}$ because $b_i = f_i\beta_i$, $i = 1, \ldots, d$. The ith diagonal entry of $\mathbf{F}$ reflects the importance of the ith wavelength. In short, if $|f_i| \approx 0$, then the ith wavelength of the spectra can effectively be ignored. If k diagonal entries of $\mathbf{F}$ are sufficiently small (approximately zero), then $\mathbf{\Phi} = \mathbf{XF}$ effectively contains k columns that are all approximately zero, and they can be removed from consideration such that $\mathbf{\Phi}$ is of dimension $n \times (d - k)$.

Equation (4) can be generalized into an iterative procedure summarized in Table I. The superscripted k in brackets indicates the current iteration. The ith diagonal element of $\mathbf{F}^{[k]}$, or $f_i^{[k]}$, is defined as a function of $b_i^{[k-1]}$, the regression coefficient from the previous iteratively reweighted feature scaling (IRFS) iteration. In Sections 2.2.1– 2.2.4, different mathematical expressions for $f_i^{[k]}$ will be defined, and each mathematical expression will correspond to a distinct wavelength selection algorithm.

The iterative procedure in Table I is initialized by a "guess," the regression vector $\mathbf{b}^{[0]} = \left[b_1^{[0]}, \ldots, b_d^{[0]}\right]^T$. (For example, $\mathbf{b}^{[0]}$ in Table I can be determined using PLS, PCR, RR, or any other MC method.) The IRFS scheme in Table I then generates a sequence of regression vectors $\mathbf{b}^{[0]}, \mathbf{b}^{[1]}, \mathbf{b}^{[2]}, \ldots$ such that the current iterate $\mathbf{b}^{[k]}$ is more sparse than the previous one $\mathbf{b}^{[k-1]}$. In Section 2.2.2, we will briefly explain, from the maximum likelihood approximation perspective, why the current update is a smaller-norm (and sparser) version of the previous update. In Sections 2.2.1– 2.2.4, we will also examine many seemingly

Table I. IRFS scheme

Step 0: Solve $\mathbf{X}\mathbf{b}^{[0]} = \mathbf{y}$ by PLS, RR, or PCR for $\mathbf{b}^{[0]}$; set $k = 1$

Step 1: Form scaling matrix $\mathbf{F}^{[k]} = \text{diag}\left(f_1^{[k]}, \ldots, f_n^{[k]}\right)$

Step 2: Solve $\mathbf{\Phi}^{[k]}\boldsymbol{\beta}^{[k]} = \mathbf{y}$ by PLS, RR, or PCR for $\boldsymbol{\beta}^{[k]}$ where $\mathbf{\Phi}^{[k]} = \mathbf{X}\mathbf{F}^{[k]}$

Step 3: Recover $\mathbf{b}^{[k]}$ using back-substitution $\mathbf{b}^{[k]} = \mathbf{F}^{[k]}\boldsymbol{\beta}^{[k]}$

Step 4: Set k = k + 1 and go to Step 1

IRFS, iteratively reweighted feature scaling; PLS, partial least squares; RR, ridge regression; PCR, principal component regression.

disparate iterative techniques for wavelength selection and see how they can be unified under the framework of IRFS.

2.2.1. *Focal underdetermined system solver*

In the signal processing literature, the focal underdetermined system solver (FOCUSS) algorithm was developed to find the "best" basis columns of $\mathbf{X}$ [25,26]. With respect to this goal, the following problem was posed:

$$\min \|\mathbf{b}\|_p, \ (0 \leq p < 1) \quad \text{subject to} \quad \mathbf{X}\mathbf{b} = \mathbf{y} + \mathbf{e} \tag{5}$$

When $p = 0$, the zero-norm $\|\mathbf{b}\|_p = \|\mathbf{b}\|_0$ corresponds to the count of nonzero components in $\mathbf{b}$. In short, Equation (5) with $p = 0$ seeks to solve $\mathbf{X}\mathbf{b} = \mathbf{y} + \mathbf{e}$ with the fewest number of nonzero regression coefficients—a laudable goal in the context of wavelength selection. However, minimizing $\|\mathbf{b}\|_0$ is generally intractable because it requires an enumerative search across all possible combinations of regression coefficients. Note that if the norm of $\mathbf{b}$ was extended to the interval $1 \leq p \leq 2$, then the FOCUSS problem would have a global minimum [27]. However, for $0 \leq p < 1$, the FOCUSS problem can have many local minima [25,26]. Instead of a guaranteed unique solution—located at the global minimum—having less sparsity when $1 \leq p \leq 2$, FOCUSS chooses to have a potentially suboptimal solution with greater sparsity when $0 \leq p < 1$. FOCUSS makes no attempt to search for global minima in a minimization landscape with potentially many local minima.

The solution to Equation (5)—using the notation defined in this paper—can be shown to have the following diagonal element

$$f_i^{[k]} = \left|b_i^{[k-1]}\right|^{1-\frac{p}{2}}$$

in the scaling matrix $\mathbf{F}^{[k]}$ in step 1 of Table I [26]. Since the original FOCUSS papers [25,26] were published, extensions of FOCUSS have been developed that allow for slightly different reweighting rules [28,29]. Our interest, however, is in comparing the base FOCUSS algorithm with other reweighting strategies employed in chemometrics.

2.2.2. *Maximum likelihood approximation*

The ordinary RR problem

$$\text{minimize } h(\mathbf{b}), \quad h(\mathbf{b}) = \frac{1}{2}\|\mathbf{X}\mathbf{b} - \mathbf{y}\|_2^2 + \frac{\lambda^2}{2}\|\mathbf{b}\|_2^2 \tag{6}$$

penalizes regression vectors with large norm such that each element of $\mathbf{b}$ is equally weighted by λ. However, suppose we have *a priori* information regarding unequal weighting for each element in $\mathbf{b}$. For example, assume that the regression coefficients $b_1, \ldots, b_d$ in $\mathbf{b}$ are random variables that follow a normal distribution with probability density function $\rho(\mathbf{b}) = \text{const} \times \exp\left(-1/2\mathbf{b}^T\mathbf{C}_b^{-1}\mathbf{b}\right)$ and are independent and identically distributed, with $\mathbf{C}_b$ the corresponding covariance matrix. Instead of minimizing $h(\mathbf{b})$ in Equation (6), the maximum likelihood estimate (MLE) is found by solving

$$\text{minimize } h(\mathbf{b}), \quad h(\mathbf{b}) = \frac{1}{2}\|\mathbf{X}\mathbf{b} - \mathbf{y}\|_2^2 + \frac{1}{2}\mathbf{b}^T\mathbf{C}_b^{-1}\mathbf{b} \tag{7}$$

or, equivalently, by solving the least squares problem $\left(\mathbf{X}^T\mathbf{X} + \mathbf{C}_b^{-1}\right)\mathbf{b} = \mathbf{X}^T\mathbf{y}$.

The MLE approach requires *a priori* estimates of $\mathbf{C}_b$, which are rarely known in practice. To make the estimation easier, we assume that the covariance matrix is a diagonal matrix of variances, that is, $\mathbf{C}_b = \text{diag}\left(\sigma_1^2, \ldots, \sigma_d^2\right)$ with σ_i representing the spectral noise associated with the ith wavelength. In the absence of measuring the same spectra repeatedly and calculating the sample standard deviation, we can instead use the magnitude of the regression coefficient b_i, derived from an initial least squares estimate, as a proxy for the variance σ_i^2. The spectroscopic premise is that wavelengths with a poor signal-to-noise ratio will be down-weighted to zero. Let $\mathbf{b}^{[0]} = \left[b_1^{[0]}, \ldots, b_d^{[0]}\right]^T$ be an initial least squares estimate derived from a standard MC method such as PLS, RR, or PCR. If the inverse of the diagonal covariance matrix is expressed as

$$\mathbf{C}_b^{-1} = \lambda^2\mathbf{L}^2 \quad \text{where} \quad \mathbf{L} = \text{diag}\left(\frac{1}{\left|b_1^{[0]}\right|}, \ldots, \frac{1}{\left|b_d^{[0]}\right|}\right) \tag{8}$$

then the MLE problem in Equation (7) can be recast in terms of TR as expressed in Equation (1):

$$\text{minimize } h(\mathbf{b}), \quad h(\mathbf{b}) = \frac{1}{2}\|\mathbf{X}\mathbf{b} - \mathbf{y}\|_2^2 + \frac{\lambda^2}{2}\|\mathbf{L}\mathbf{b}\|_2^2 \tag{9}$$

Such an approach was successfully used for wavelength selection [30].

Note that the solution $\mathbf{b}$ in Equation (9) is a smaller-norm version of the initial estimate $\mathbf{b}^{[0]}$. When the magnitude of the coefficient $b_i^{[0]}$ in Equation (8) is small, the corresponding diagonal element of $\mathbf{L}$, that is, $\mathbf{L}_{ii} = 1/\left|b_i^{[0]}\right|$, is large. As a result, a large magnitude for the updated coefficient b_i in Equation (9) will be heavily penalized by $\mathbf{L}_{ii}$ in the term $\lambda^2/2\,\|\mathbf{L}\mathbf{b}\|_2^2$.

Using the variable transformations $\boldsymbol{\Phi} = \mathbf{X}\mathbf{L}^{-1}$ and $\boldsymbol{\beta} = \mathbf{L}\mathbf{b}$, we transform Equation (9) into the standard TR problem [31]:

$$\text{minimize } h(\boldsymbol{\beta}), \quad h(\boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{\Phi}\boldsymbol{\beta} - \mathbf{y}\|_2^2 + \frac{\lambda^2}{2}\|\boldsymbol{\beta}\|_2^2 \qquad (10)$$

Equation (10) corresponds to RR. However, there is nothing particularly special about RR as the baseline least squares solver—one could also use PLS or PCR. As a result, the MLE approach of Equation (9) can easily be recast as an IRFS scheme in Table I where

$$f_i^{[k]} = \left|b_i^{[k-1]}\right|.$$

Moreover, the MLE scheme can be seen as special case of the FOCUSS algorithm where $p = 0$ because $\left|b_i^{[k-1]}\right|^{1-p/2} = \left|b_i^{[k-1]}\right|$.

2.2.3. *Adaptively preconditioned partial least squares*

Recently, adaptively preconditioned PLS (APPLS) was used for wavelength selection [32]. In an iterative fashion, APPLS generates a sequence of regression vectors that become sparser per iteration and can be written as an IRFS scheme of Table I where

$$f_i^{[k]} = \tau \sqrt{\frac{\left(b_{i,j}^{[k-1]}\right)^2}{\sum_{l=1}^{d}\left(b_{l,j}^{[k-1]}\right)^2}} \qquad (11)$$

The subscript j in $b_{i,j}^{[k]}$ in Equation (11) denotes the jth PLS latent vector (or component or factor) for the ith regression coefficient at the kth APPLS iteration. The parameter τ in Equation (11) is a scalar bounded in the interval $(0, 1]$. The values for both j and τ are determined by cross-validation [32]. In this paper, we set $\tau = 1$ because the diagonal elements $f_i^{[k]}$ then have a probabilistic interpretation: $f_i^{[k]} > 0$ and $\sum_{i=1}^{d} f_i^{[k]} = 1$.

2.2.4. *Binary weighting schemes*

The previously mentioned feature scaling schemes— FOCUSS, MLE, and APPLS—all have diagonal weights $\mathbf{F}^{[k]} = \text{diag}\left(f_1^{[k]}, \ldots, f_d^{[k]}\right)$ that are continuously valued and nonnegative. However, the value of the weights can be arbitrary. In particular, if the weights $f_i^{[k]}$ are binary in value (0 or 1), then the transformation $\boldsymbol{\Phi} = \mathbf{X}\mathbf{F}^{[k]}$ leaves the columns associated with the unit-valued weights untouched (there is no rescaling for these columns, whereas the columns associated with the zero-valued weights are discarded).

Suppose we use a regression technique (such as PLS) to build an initial regression vector $\mathbf{b}^{[0]}$ in which all of the wavelengths of $\mathbf{X}$ are used. We then sort the regression coefficient magnitudes $\left|b_i^{[0]}\right|$ in ascending order and remove a proportion of the wavelengths associated with the smallest values of $\left|b_i^{[0]}\right|$. In effect, the weights associated with the smallest magnitudes are zero, whereas the surviving wavelengths have unit weight. Calibration is again performed using the trimmed feature set, the magnitudes $\left|b_i^{[1]}\right|$ of the next iterate are sorted, and a proportion of the remaining wavelengths is removed. This process is repeated until there are no more wavelengths left. In the early bioinformatics literature, this recursive procedure was often applied to microarray data. (Microarray data sets associated with cancer studies often consist of a n (patient samples) $\times$ d (genes) matrix of gene expression values.) In particular, linear support vector machines were used to perform "gene selection" on cancer-related microarray data sets [33]. In the chemometrics literature, similar schemes have been invoked using PLS to perform wavelength selection such as competitive adaptive reweighted sampling [34]. In these recursive schemes, the proportion of wavelengths that are removed per iteration has to be fairly large; otherwise, there would be too many iterations to perform.

For our approach, which we denote as binary weighting scheme (BWS), we follow a similar convention to competitive adaptive reweighted sampling whereby the number of wavelengths kept decays exponentially per iteration:

$$N(i) = N_0 e^{-ki} \quad \text{where} \quad k = -\frac{\ln(q)}{i_{\max}}$$

Here, $N(i)$ is the number of wavelengths kept for the ith IRFS iteration, $N_0 = d$ is the number of wavelengths of $\mathbf{X}$, $i_{\max}$ is the maximum number of IRFS iterations, and q is the desired proportion of surviving wavelengths at $i_{\max}$ iterations. The proportion $q = 0.01$ (or 1% of the original number of wavelengths) is the default value we use.

3. REPURPOSING ITERATIVELY REWEIGHTED FEATURE SCALING SCHEMES FOR SAMPLE SELECTION

In this section, the IRFS schemes from Section 2 will be repurposed for sample selection, that is, finding a predictive subset of samples. Traditionally, weighted least squares procedures have been used for this purpose, but the IRFS schemes, coupled with least squares generalizations of support vector regression (SVR), can effectively perform sample selection as well.

3.1. Weighted least squares

Instead of minimizing $1/2\|\mathbf{X}\mathbf{b} - \mathbf{y}\|_2^2 = 12\sum_{i=1}^{n} r_i^2$ where $r_i = \mathbf{x}_i^T\mathbf{b} - y_i$ is the residual associated with the ith sample, weighted least squares problems instead solve

$$\text{minimize } h(\mathbf{b}), \quad h(\mathbf{b}) = \|\mathbf{W}^{1/2}(\mathbf{X}\mathbf{b} - \mathbf{y})\|_2^2 = \frac{1}{2}\sum_{i=1}^{n} w_i r_i^2 \qquad (12)$$

where $\mathbf{W} = \text{diag}(w_1, \ldots, w_n)$. Generally, when one cares about certain samples more than others (e.g., samples that we expect to see again or samples whose misfit is costlier), then $w_i > 1$. Weighted least squares problems perform "sample scaling" instead of feature scaling because Equation (12) can be rewritten as the solution to the linear system $\tilde{\mathbf{X}}\mathbf{b} = \tilde{\mathbf{y}} + \mathbf{e}$ where $\tilde{\mathbf{X}} = \mathbf{S}\mathbf{X}$ and $\tilde{\mathbf{y}} = \mathbf{S}\mathbf{y}$ with $\mathbf{S} = \mathbf{W}^{1/2} = \text{diag}\left(\sqrt{w_1}, \ldots, \sqrt{w_n}\right)$ being the sample-scaling matrix that premultiplies $\mathbf{X}$ (instead of postmultiplying $\mathbf{X}$ as in the IRFS schemes). However, to reuse IRFS schemes for sample selection, we will want to postmultiply a kernel matrix by a scaling matrix $\mathbf{F}$—this will be discussed in the next section in the context of SVR.

3.2. Support vector regression

Support vector regression, like any other regression technique, strives to find the optimal line/plane-of-fit $g(\mathbf{x}) = \mathbf{x}^T\mathbf{b} + b_0$ by minimizing residuals $r_i = g(\mathbf{x}_i) - y_i$ where $|r_i|$ is the vertical distance between the coordinate $(\mathbf{x}_i, y_i)$ and the line-of-fit $g(\mathbf{x})$. Unlike other regression techniques, however, SVR seeks the best hyper-slab or corridor that contains most of the data with the line or plane in the middle of the corridor being deemed the best line-of-fit (Figure 1). In Figure 1, ϵ is the vertical distance between the middle black line and the red or blue lines above and below. The ϵ-value associated with the red lines is smaller than the ϵ-value associated with the blue line. If ϵ is sufficiently large, then the corridor (or blue ϵ-tube) will contain all of the data points, that is, $-\epsilon \le r_i \le \epsilon, i = 1, \ldots, n$. The simplest (and most naive) SVR is mathematically described by the following [35]:

$$\text{minimize } \frac{1}{2}\|\mathbf{b}\|_2^2 \quad \text{subject to} \quad -\epsilon \le r_i \le \epsilon, \ \ i = 1, \ldots, n \tag{13}$$

Equation (13) describes the scenario when all points are within the ϵ-tube, and such a scenario does not guarantee a good line-of-fit. The width γ between the corridor walls in the ϵ-tube can be expressed as $\gamma = 2/\|\mathbf{b}\|_2$ [36,37]. To maximize the width γ, we minimize the two-norm of $\mathbf{b}$. (An excellent discussion and derivation of the optimization problem associated with SVR, as well as the support vector machine, can be found in Max Welling's Classnotes in Machine Learning [36,37].)

To find a tolerable line-of-fit, we must decrease ϵ in value. However, as we shrink ϵ, the likelihood that data points will lie outside the ϵ-tube will increase. As a consequence, one or more of the constraints $-\epsilon \le r_i \le \epsilon$ will have to be violated, and provisions allowing for constraint violation will have to be defined. In the SVR and support vector machine literature, the slack variables $\boldsymbol{\xi}^{(1)} = \left[\xi_1^{(1)}, \ldots, \xi_n^{(1)}\right]$ and $\boldsymbol{\xi}^{(2)} = \left[\xi_1^{(2)}, \ldots, \xi_n^{(2)}\right]$ are used for this purpose. If the ith sample $(\mathbf{x}_i, y_i)$ is within the tube, then the corresponding slack variables $\xi_i^{(1)}$ and $\xi_1^{(2)}$ are both zero—there is no need for slack. If $(\mathbf{x}_i, y_i)$ lies above the tube, then slack is only needed above the corridor: $\xi_i^{(1)} > 0$ is the vertical distance between $(\mathbf{x}_i, y_i)$ and the upper corridor wall, whereas $\xi_i^{(2)} = 0$. Similarly, if $(\mathbf{x}_i, y_i)$ is below the tube, then $\xi_i^{(2)} > 0$ and $\xi_i^{(1)} = 0$.

Mathematically, the SVR problem allowing for slack is given by

$$\begin{aligned} &\text{minimize } \tfrac{1}{2}\|\mathbf{b}\|_2^2 + P(\boldsymbol{\xi}^{(1)}, \boldsymbol{\xi}^{(2)}) \\ &\text{subject to } -\epsilon - \xi_i^{(1)} \le r_i \le \epsilon + \xi_i^{(2)}, \\ &\qquad \xi_i^{(1)} \ge 0, \ \xi_i^{(2)} \ge 0, \ i = 1, \ldots, n \end{aligned} \tag{14}$$

The penalty term $P(\boldsymbol{\xi}^{(1)}, \boldsymbol{\xi}^{(2)})$ in Equation (14) stipulates that we want to minimize the amount of "corridor violation."

Different mathematical expressions for $P(\boldsymbol{\xi}^{(1)}, \boldsymbol{\xi}^{(2)})$ generate different SVR formulations [38,39]. We use the following two-norm penalty formulation

$$P(\boldsymbol{\xi}^{(1)}, \boldsymbol{\xi}^{(2)}) = C^2 \sum_{i=1}^{n} \left[\left(\xi_i^{(1)}\right)^2 + \left(\xi_i^{(2)}\right)^2 \right] \tag{15}$$

because it generates an SVR that is easy to solve. (Mathematically, the appeal of this penalty is that it is easy to differentiate [35,38,40,41].) The larger the constant $C > 0$ is in Equation (15), the larger the amount of ϵ-tube violation one is willing to tolerate. To further simplify the SVR formulation, we drop the offset variable b_0 in the line/plane-of-fit, that is, we write $g(\mathbf{x}) = \mathbf{x}^T\mathbf{b}$ instead of $g(\mathbf{x}) = \mathbf{x}^T\mathbf{b} + b_0$. In the SVR literature, it is generally assumed that the offset variable b_0 is not known in advance. However, in most chemometric applications, the offset variable b_0 is known. If the data has been mean centered, as we assume here in this paper, then b_0 does not have to be explicitly solved for because $b_0 = \bar{\mathbf{y}}$.

In optimization theory, one typically transforms a problem with inequality constraints (the primal problem) into another one (the dual problem) with simpler constraints. In the case of Equation (14), the transformation of the primal minimization problem becomes the maximization of the following Lagrangian function $\mathcal{L}$ [36,42]:

$$\begin{aligned} \mathcal{L}(\mathbf{b}, \mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \boldsymbol{\xi}^{(1)}, \boldsymbol{\xi}^{(2)}) = &\frac{1}{2}\|\mathbf{b}\|_2^2 + P(\boldsymbol{\xi}^{(1)}, \boldsymbol{\xi}^{(2)}) \\ &+ \sum_{i=1}^{n} a_i^{(1)} \left(-r_i - \epsilon - \xi_i^{(1)}\right) \\ &+ \sum_{i=1}^{n} a_i^{(2)} \left(r_i - \epsilon - \xi_i^{(2)}\right) \end{aligned} \tag{16}$$

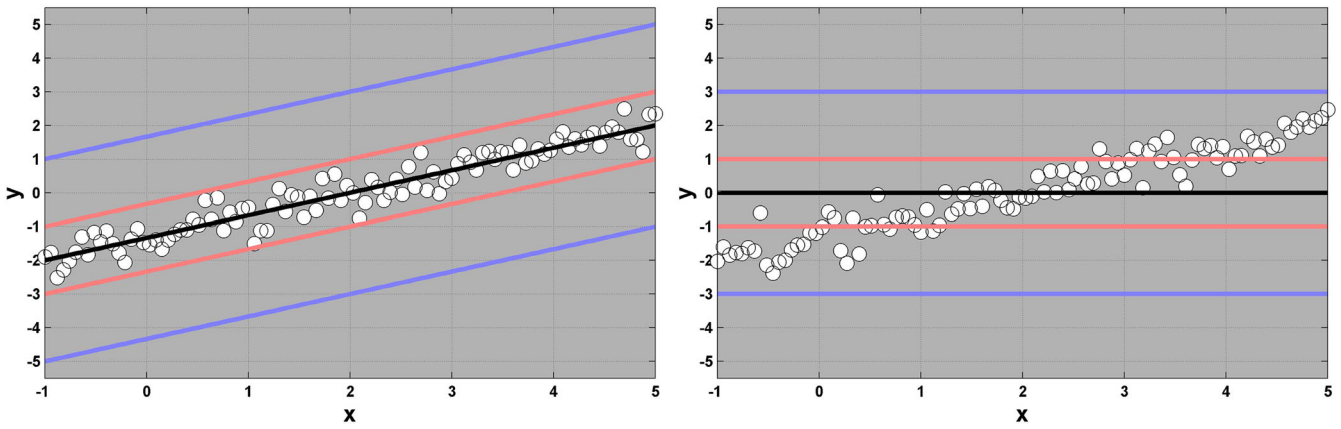

Figure 1. Both subplots contain the same data, but the line-of-fit in each subplot is different. The blue ϵ-tube is wide enough to contain all of the data points in both subplots. However, the narrower ϵ-tube (defined by the red lines) better fits the data. A wide ϵ-tube does not guarantee a good line-of-fit.

The elements in the vectors $\mathbf{a}^{(1)} = \left[a_1^{(1)}, \ldots, a_n^{(1)}\right]$ and $\mathbf{a}^{(2)} = \left[a_1^{(2)}, \ldots, a_n^{(2)}\right]$ are called the "Lagrange multipliers," and they are often referred to as the dual variables. Instead of solving for the primal variables $\mathbf{b}$ (one component for each wavelength), we solve for the dual variables $\mathbf{a}^{(1)}$ and $\mathbf{a}^{(2)}$ (one component for each sample). In practice, the dual variables $\mathbf{a}^{(1)}$ and $\mathbf{a}^{(2)}$ are collapsed into a single vector $\mathbf{a} = \mathbf{a}^{(1)} - \mathbf{a}^{(2)}$ [36,38]. Moreover, the slack and primal variables are linearly related to the dual variables [36]:

$$\xi_i^{(1)} = \frac{1}{C} a_i^{(1)} \quad \text{and} \quad \xi_i^{(2)} = \frac{1}{C} a_i^{(2)} \tag{17}$$

$$\mathbf{b} = \mathbf{X}^T(\mathbf{a}^{(1)} - \mathbf{a}^{(2)}) = \mathbf{X}^T\mathbf{a} = \sum_{i=1}^{n} a_i \mathbf{x}_i \tag{18}$$

Plugging Equations (17) and (18) into Equation (16), we can simplify the Lagrangian function into the following unconstrained minimization problem involving only the dual variables $\mathbf{a}$ [36]:

$$\text{minimize } \mathcal{L}(\mathbf{a}), \quad \mathcal{L}(\mathbf{a}) = \frac{1}{2}\mathbf{a}^T(\mathbf{K} + \lambda^2 \mathbf{I}_n)\mathbf{a} - \mathbf{a}^T\mathbf{y} + \epsilon\|\mathbf{a}\|_1 \tag{19}$$

where $\mathbf{K} = \mathbf{X}\mathbf{X}^T$ is the kernel matrix and $\lambda = 1/C$.

According to Equation (18), the primal regression vector $\mathbf{b}$ can be written as a linear combination of spectral samples $\mathbf{x}_i$. Thus, the samples associated with $a_i = 0$ have no impact in the construction of $\mathbf{b}$. The samples that do matter, the ones associated with nonzero values of a_i, are referred to as the "support vectors." Recall that in the primal problem of Equation (14), samples outside the ϵ-tube (far away from the predicted line/plane-of fit) are punished via the penalty term $P(\boldsymbol{\xi}^{(1)}, \boldsymbol{\xi}^{(2)})$: either $\xi_i^{(1)}$ or $\xi_i^{(2)}$ is nonzero. For the samples within the ϵ-tube, there is no penalization: both $\xi_i^{(1)}$ and $\xi_i^{(2)}$ are zero. Since the slack variables and dual variables are linearly related via Equation (17), if both $\xi_i^{(1)}$ and $\xi_i^{(2)}$ are zero, then $a_i = a_i^{(1)} - a_i^{(2)} = 0$. Hence, the support vectors are precisely those spectral samples that lie "outside" the ϵ-tube.

The aforementioned SVR formulation can easily be generalized to nonlinear regression via nonlinear kernels [35,36,38]. However, we restrict our attention to the linear regression setting where $\mathbf{K} = \mathbf{X}\mathbf{X}^T$. We now show how the IRFS schemes in Section 2 can be applied to SVR for purposes of sample selection.

3.2.1. *Kernel ridge regression*

If we set $\epsilon = 0$ in Equation (19), then the ϵ-tube shrinks to zero, and all the samples become support vectors. The SVR problem of Equation (19) then simplifies to

$$\text{minimize } h(\mathbf{a}), \quad h(\mathbf{a}) = \frac{1}{2}\mathbf{a}^T(\mathbf{K} + \lambda^2 \mathbf{I}_n)\mathbf{a} - \mathbf{a}^T\mathbf{y} \tag{20}$$

In the optimization setting, if we set the gradient of $h(\mathbf{a})$ equal to zero and solve for $\mathbf{a}$, then we arrive at the simple linear system

$$(\mathbf{K} + \lambda^2 \mathbf{I}_n)\mathbf{a} = \mathbf{y} \tag{21}$$

which is often referred to as kernel RR [38].

To see how IRFS schemes can be repurposed for sample selection, we will make precise the connection between Equation (21) and ordinary RR. Note that the kernel matrix $\mathbf{K} = \mathbf{X}\mathbf{X}^T$ is symmetric positive semidefinite (all of the eigenvalues of $\mathbf{K}$ are nonnegative). This property allows us to take fractional powers of $\mathbf{K}$ via the singular value decomposition: $\mathbf{K}^p = \mathbf{U}\boldsymbol{\Sigma}^p\mathbf{U}^T$ where $\mathbf{U}$ is a matrix of singular vectors and $\boldsymbol{\Sigma}$ the diagonal matrix of singular values. As a result, we can exploit the following variable transformation of Franklin [43]

$$\mathbf{K}^* = \mathbf{K}^{1/2} \quad \text{and} \quad \mathbf{y}^* = \mathbf{K}^{-1/2}\mathbf{y} \tag{22}$$

and rewrite Equation (21) as the linear system

$$\mathbf{K}_\lambda \mathbf{a} = \mathbf{y}_\lambda \quad \text{where} \quad \mathbf{K}_\lambda = \begin{bmatrix} \mathbf{K}^* \\ \lambda \mathbf{I}_n \end{bmatrix} \quad \text{and} \quad \mathbf{y}_\lambda = \begin{bmatrix} \mathbf{y}^* \\ \mathbf{0}_n \end{bmatrix} \tag{23}$$

Hence, the solutions to Equations (20)–23) are the same as the solution to $\mathbf{K}^*\mathbf{a} = \mathbf{y}^*$ when solved by RR. However, one is not restricted to using RR—one can also use PLS or PCR. Moreover, any of the IRFS schemes used in Section 2 can now be applied to $\mathbf{K}^*\mathbf{a} = \mathbf{y}^*$ (Table II). The IRFS schemes then generate a sequence of increasingly sparse "support-vector-like" solutions. When we back-substitute $\mathbf{b}^{[k]} = \mathbf{X}^T\mathbf{a}^{[k]}$ in Table II, we likewise generate a sequence of primal regression vectors $\mathbf{b}^{[0]}, \mathbf{b}^{[1]}, \ldots$. However, the regression vector $\mathbf{b}^{[k]}$ is not guaranteed to be sparse in the wavelength sense where many of coefficients $b_i^{[k]}$ are zero. Sparsity is only achieved in the sample sense where many of the coefficients $a_i^{[k]}$ are zero.

3.2.2. *Support vector regression reformulated as a least absolute shrinkage and selection operator problem*

The variable transformation of Franklin in Equation (22) also allows us to rewrite the SVR problem in Equation (19) as

$$\text{minimize } h(\mathbf{a}), \quad h(\mathbf{a}) = \frac{1}{2}\|\mathbf{K}_\lambda \mathbf{a} - \mathbf{y}_\lambda\|_2^2 + \epsilon\|\mathbf{a}\|_1 \tag{24}$$

Table II. IRFS scheme for sample selection

Step 0: Solve $\mathbf{K}^*\mathbf{a}^{[0]} = \mathbf{y}^*$ by RR, PLS, or PCR for $\mathbf{a}^{[0]}$; set $k = 1$

Step 1: Form scaling matrix $\mathbf{F}^{[k]} = \text{diag}\left(f_1^{[k]}, \ldots, f_m^{[k]}\right)$

Step 2: Solve $\boldsymbol{\Phi}^{[k]}\boldsymbol{\alpha}^{[k]} = \bar{\mathbf{y}}^{[k]}$ by RR (or PLS or PCR) for $\boldsymbol{\alpha}^{[k]}$ where $\boldsymbol{\Phi}^{[k]} = \mathbf{K}\mathbf{F}^{[k]}$

Step 3: Recover $\mathbf{a}^{[k]}$ using the back-substitution $\mathbf{a}^{[k]} = \mathbf{F}^{[k]}\boldsymbol{\alpha}^{[k]}$

Step 4: Set k = k + 1 and go to step 1

IRFS, iteratively reweighted feature scaling; PLS, partial least squares; RR, ridge regression; PCR, principal component regression.

Functionally, Equation (24) is the same as the LASSO problem in Equation (2) with $q = 1$. From this perspective, one can see that the sparsity that SVR achieves is really through the sparsity mechanism enabled by the LASSO algorithm. In Section 2, we wanted to replace the computationally burdensome LASSO algorithm with IRFS schemes. Here, in Section 3, we are doing the same thing—replacing the SVR with IRFS schemes—but in the dual space of $\mathbf{a} = [a_1, \ldots, a_n]^T$ instead of the primal space of $\mathbf{b} = [b_1, \ldots, b_d]^T$.

4. IMPLEMENTATION AND MODEL SELECTION

4.1. Least absolute shrinkage and selection operator via least angle regression

The particular LASSO algorithm we use in this paper is known as LAR [13]. If there are d wavelengths in the data set, then LAR builds $d + 1$ regression vectors $\mathbf{b}_0, \mathbf{b}_1, \ldots, \mathbf{b}_d$ where $\mathbf{b}_i$ contains i nonzero regression coefficients. In effect, LAR starts with the most sparse solution ($\mathbf{b}_0$, a regression vector of all zeros) and ends up with a regression vector $\mathbf{b}_d$ containing all nonzero coefficients. When $n \geq d$ (and if $\mathbf{X}$ has full numerical rank), then the final LAR iterate $\mathbf{b}_d$ corresponds to the ordinary least squares solution. Note that LAR can operate in two modes: with or without the "LASSO modification." Ordinary LAR is LAR without the LASSO modification—it adds one feature at a time. Ordinary LAR is a bottom-up, greedy algorithm: the set of i coefficients in $\mathbf{b}_i$ is always a subset of coefficients contained in $\mathbf{b}_{i+1}$. LAR with the LASSO modification either adds or deletes one feature at a time, and it stops when the final iterate contains all of the features. We use ordinary LAR.

4.1.1. *The tuning parameter associated with least angle regression*

It is important to note that the vast majority of LASSO algorithms require λ as an input. These algorithms then solve Equation (2) with $q = 1$, and their output is some λ-dependent regression vector. Only then can one determine the number of nonzero coefficients in the regression vector. Aside from the fact that a larger λ value results in greater sparsity, the LASSO practitioner does not know—ahead of time before the calibration is performed—the number of nonzero coefficients associated with a given λ value. LAR, on the other hand, dispenses with the need to know λ. As LAR creates the regression vectors $\{\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_d\}$ in a forward, stepwise fashion, it computes, as a side effect, the λ values $\{\lambda_0, \lambda_1, \lambda_2, \ldots, \lambda_d\}$ associated with these regression vectors such that $\lambda_0 > \lambda_1 > \lambda_2 > \cdots > \lambda_d$. As a result, LAR is much more intuitive to use than other LASSO algorithms because the tuning parameter of interest is the number of wavelengths used.

4.1.2. *Computational considerations of least angle regression*

There are many implementations of the LAR algorithm. We choose the MATLAB (Natwick, MA, USA) implementation that we denote as DTU-LAR [44]. (DTU is the Danish acronym for the Technical University of Denmark from where it was developed.) The primary computational overhead associated with DTU-LAR is that, at the ith iteration, an $i \times i$ linear system must be solved. There are many numerical techniques that can be used to accelerate the solution of this linear system. The technique employed in DTU-LAR involves the QR factorization of $\mathbf{X}_i$ ($\mathbf{X}$ using only i LAR-selected features), which can be efficiently updated or downdated whenever a feature enters or leaves the active set of features. DTU-LAR, in its default setting, will continue until all N LAR iterations have been exhausted—one iteration for each added feature such that $N = d$.

The computational burden of DTU-LAR is minimal when i ($1 \leq i \leq d$) is small. However, when i is "large," for example, hundreds or thousands, the solution of an $i \times i$ linear system will become prohibitive—regardless of which numerical technique is used to accelerate the linear inversion. Hence, DTU-LAR does not "scale" very well to large numbers of wavelengths. It is at this stage that we expect the IRFS schemes to pay dividends in terms of speedup. Whereas LAR performs d linear inversions where d could be in the hundreds or thousands, IRFS requires only M iterations of PLS (or PCR) where M is typically in the single digits. At this point, there are two remarks worth mentioning regarding iteration number. First, DTU-LAR allows one to prematurely stop at a desired number of features. Instead of performing $N = d$ iterations, one can perform $N = d_0$ iterations where $d_0 << d$. Second, the computational bottleneck associated with classical MC methods such as PLS or PCR is the number of latent vectors K that one wants to use. Hence, when comparing the computational efficiency between DTU-LAR and IRFS, one must keep in mind the number of LAR iterations N versus the number of IRFS iterations M coupled with the number of latent vectors K. An example of this trade-off between N, M, and K is given in Section 5.2.2.

4.2. Support vector regression via least angle regression

In SVR, the penalty parameter ϵ in Equation (19) is the tuning parameter that must be chosen. However, because we can transform the SVR problem in Equation (19) into the LASSO problem of Equation (24), we can use any LASSO method to solve the SVR problem. We will denote the solution of the SVR problem via the LASSO problem in Equation (24) as SVR-LASSO. In particular, we use the DTU-LAR implementation from Section 4.1.2 to solve the SVR problem. In this approach, we construct a sequence of dual solution vectors $\{\mathbf{a}_0, \mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\}$ such that $\mathbf{a}_i$ contains i nonzero coefficients. (Recall that the samples associated with these nonzero coefficients are the support vectors.) The sequence of primal regression vectors $\mathbf{b}_i$ is easily recovered via the relation $\mathbf{b}_i = \mathbf{X}^T\mathbf{a}_i$. Although the support vectors are selected from the entire calibration pool of samples, a chemometrician, in hindsight, may want to go back and see which spectra were used as support vectors. This could be a useful information to explore, especially for subsequent calibrations.

4.3. Iteratively reweighted feature scaling implementation

For IRFS, we will examine its use for both wavelength and sample selection. As opposed to LAR, IRFS is a top-down algorithm—we start with all the features/samples and then winnow down the features/samples using the diagonal scaling matrices of Tables I and II. In Tables I and II, we will use PLS as the baseline MC method. For wavelength selection, we will compare the IRFS schemes (FOCUSS, MLE, APPLS, and BWS) against ordinary PLS. In the case of sample selection, we also compare the IRFS schemes against ordinary PLS. Note that in the sample selection case, we are solving Equation (23) with $\lambda = 0$. Hence, PLS applied to this equation will be referred to as kernel PLS.

To illustrate the simplicity of the IRFS implementations for both wavelength and sample selection, MATLAB scripts for IRFS are available at www.hpc.unm.edu/~andriese

4.4. Model selection

We want to select the appropriate tuning parameter (number of wavelengths in the case of LASSO, the number of support vectors in the case of SVR-LASSO, and the number of components or factors in the case of PLS) that models as much of the complexity of the system without over-fitting. To accomplish this goal, we use a five-fold cross-validation. A tuning parameter that is too small tends to under-fit the data, whereas a tuning parameter that is too large tends to over-fit. *N*-fold cross-validation builds a model on $N-1$ disjoint sample blocks of the calibration spectra. A prediction for the analyte concentrations is then made for the samples in the withheld sample block. This process is repeated for a total of N times until each sample block has been left out once. The analyte concentration predicted for a sample is then compared with the known concentration of its reference sample. We use the root-mean-square error of cross-validation (RMSECV)

$$\rho_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_{ij} - y_i\right)^2}$$

as a measure of how well a particular calibration fits the analyte concentration data. The value $\hat{y}_{ij}$ is the prediction of the ith calibration sample using the jth tuning parameter.

4.4.1. M_α-test for choosing a tuning parameter

A naive choice for the optimal calibration tuning parameter would be the tuning parameter that minimizes the RMSECV curve ρ_j, $j = 1, \ldots, k$. Using the k_*th tuning parameter ($1 \leq k_* \leq k$) that minimizes the RMSECV generally leads to over-fitting. Alternatives to the minimum RMSECV essentially strive to find an optimal model from a collection of models using fewer than k_* tuning parameters. One common alternative used in chemometrics is the *F*-statistic test [45]. The *F*-test generally yields a model that under-fits. However, the models selected by the *F*-test, in our experience, are too parsimonious. We propose a simple model selection method called the M_α-test that works reasonably well, is easy to implement, and results in a model in between the extremes of the *F*-test and the minimum RMSECV [46]. Let k_0 be the index that corresponds to the logarithm of the RMSECV value ρ_{k_0} that first goes below the threshold:

$$a + \alpha(b - a) \quad \text{where} \quad a = \min_j(\log_{10}(\rho_j)) \quad \text{and} \quad b = \max_j(\log_{10}(\rho_j)).$$

The value of α is between 0 and 1 and is set to 0.05. If one wants a more or less parsimonious model, then one can increase or decrease, respectively, the value of α.

Note that the M_α-test depends on the scale of the elements in $\mathbf{y}$. Hence, that is why we take the logarithm of the RMSECV values—in the event that the discrepancy between $\min_j(\rho_j)$ and $\max_j(\rho_j)$ is very large. Although this simple approach works reasonably well with our data in Section 5.1 (and with other spectral data sets that we have worked with), it may not generalize to all data sets. A useful discussion on scale-free model selection performance criteria (such as the one-standard-error rule) can be found in the literature [47].

Once the tuning parameter has been selected using some model selection performance criterion (M_α-test in our case) on the calibration data, we then build a calibration model **b** using the selected tuning parameter on the entire calibration set. This calibration model **b** is then applied to the validation spectra, and a prediction is made on the analyte concentrations for these samples. The root-mean-square error of validation (RMSEV) is then computed.

4.4.2. M_α-test applied to least angle regression and iteratively reweighted feature scaling

For LAR, the tuning parameter is the number of wavelengths, and the M_α-test is used for determining this parameter. For IRFS using PLS, model selection is as follows. For the 0th iterate (ordinary PLS), we use the M_α-test to select the PLS component or factor. For the second, fourth, and sixth iterations, we do the exact same thing. (We examine only a few IRFS iterations to make more manageable the presentation of subsequent results). We are intentionally making no attempt to simultaneously optimize both the number of IRFS iterations and the number of PLS factors. Hence, per IRFS iteration, the number of PLS factors is the lone parameter that we will tune. We are just reporting what would happen if we were to fix the number of IRFS iterations to be 2, 4, or 6. Does sparsity increase as a function of IRFS iteration? Does RMSEV decrease as a function of IRFS iteration?

5. RESULTS AND DISCUSSION

5.1. Data sets

To facilitate a comparison between LASSO-based methods and IRFS, we examine the following spectroscopic data sets: corn [48] and wheat [49].

The corn data set consists of 80 samples of corn with $d = 700$ absorbances measured from 1000 to 2498 nm at 2-nm intervals on three near-infrared (NIR) spectrometers designated m5, mp5, and mp6. Reference values are provided for oil, protein, starch, and moisture content. Protein content is the prediction property studied in this paper, and the spectra measured on instrument m5 serve as the primary calibration set.

The wheat data set consists of 884 spectral samples (777 calibration and 107 validation samples) of whole-grain Canadian wheat measured by diffuse reflectance spectroscopy. There are $d = 1038$ wavelengths from 400 to 2498 nm at 2-nm intervals. The calibration samples represent samples grown in years 1998 and 2000–2005. The validation samples were grown in 1999 and are quite separate from the calibration samples. There are many references associated with this data set, but we are only interested in percentage protein content for each sample. This data set was featured in the "NIR data shoot-out" of the 2008 International Diffuse Reflectance Conference in Chalmersburg, PA, USA.

For the corn data set, the calibration and validation data sets are not specified. In this paper, the first 60% of the data (samples 1 through 48) will be used as the calibration set, that is, $n = 48$ is the number of calibration samples. The remaining 40% (samples 49 through 80) will be used as the validation set. Table III shows the dimensions of each data set—the number of wavelengths and the number of samples in the calibration and validation sets.

For both data sets, no modifications or preprocessing treatments (other than the initial mean centering) were made. The data was not scaled to have unit variance across wavelengths.

Table III. Data set dimensions

Data set	Number of wavelengths	Number of samples		
		Calibration	Validation	Total
Corn	700	48	32	80
Wheat	1050	777	107	884

Moreover, in the results to follow, we were completely blind to the validation spectra during calibration. If a validation spectrum did not necessarily correspond to the trend of the most homogeneous calibration spectra, then the prediction error may be larger than expected. No attempt was made to exclude validation spectra using some exclusion criteria or measure of "outlyingness." For example, in the wheat data set, the validation samples come from a completely different year than that of the calibration samples. In this case, wavelength selection may result in a finely tuned calibration model that may not generalize to the different set of samples in the validation set. (If the calibration set consisted of similar samples as those in the validation set, then a different set of wavelengths may have been chosen.) Hence, for the wheat data set, it will be interesting to see how PLS performs (using all wavelengths) compared with LAR and IRFS (using wavelengths specific to the calibration set).

5.2. An example of wavelength selection using the corn data

We first want to give an example comparison between LASSO and IRFS for wavelength selection. For simplicity, we restrict ourselves (for the time being) to the corn data set and to the MLE IRFS scheme.

5.2.1. Performance comparison between least absolute shrinkage and selection operator and iteratively reweighted feature scaling

In Figure 2, LASSO is compared with the MLE scheme. The upper left subplot displays the RMSEV curve for the LASSO. The large white dot on the RMSEV curve corresponds to the model chosen by the M_α-test. The upper right subplot shows the RMSEV curves across multiple IRFS iterations. Here, the RMSEV for the initial IRFS iterate (the black curve) corresponds to ordinary PLS regression. The red, green, and blue RMSEV curves correspond to IRFS performance after two, four, and six IRFS iterations, respectively. The large dots similarly correspond to the model chosen by the M_α-test. The upper left and upper right subplots highlight the need for wavelength selection. Without wavelength selection, the RMSEV is slightly below 0.01. With wavelength selection, the RMSEV is reduced significantly by both LASSO and IRFS.

The lower-left subplot shows the value of the regression coefficients for the models chosen by the M_α-test for LASSO and IRFS. In addition, the number of nonzero coefficients used in the regression vectors is shown. For example, at the fourth IRFS iteration (the green curve), only 160 out of 700 regression coefficients are nonzero. Not easily displayed in this subplot is the fact that many of the 160 nonzero regression coefficients are quite small in magnitude but not sufficiently small enough (below some threshold) to be set to zero. Hence, sparsity is effectively achieved quite rapidly after a few IRFS iterations. The convergence of IRFS to a sparse model vector can more easily be seen in the lower-right subplot—it is the same as the lower-left subplot, but the y-axis range is restricted to $[-8, 8]$. With enough IRFS iterations, IRFS is able to perform as well as LASSO with respect to RMSEV and parsimony (the number of wavelengths used).

5.2.2. Computational speedup

The subplots in Figure 2 suggest a natural question: How many IRFS iterations are needed to achieve an adequate level of prediction? In the case of the corn data set, four IRFS iteration appear to suffice. Other data sets may require fewer, the same, or more IRFS iterations. Using too few IRFS iterations would not likely drive down the RMSEV value to a reasonable level (a value commensurate with the LASSO RMSEV result). Using too many IRFS iterations might well yield a sparse regression vector with a low RMSEV, but the overall CPU time might be longer than that of the LASSO, diminishing any speedup advantage.

For a given number M of IRFS iterations, the computational speedup achieved by IRFS over LASSO will depend upon two criteria: the number of PLS factors K and the number of LAR iterations N. In the case of LAR, if a data set has d wavelengths, then LAR (by default) will perform $d + 1$ iterations, with the $(i + 1)$th iteration generating a regression vector $\mathbf{b}^{[i]}$ containing i nonzero regression coefficients. However, as seen in the left subplot of Figure 2, one need not use that many wavelengths to achieve a desired result. Because the DTU-LAR implementation of the LASSO allows one to set the maximum number of LAR iterations, N can be set to be less than d. The computational speedup of IRFS over LASSO will therefore be defined as the following ratio:

$$\theta = \frac{t_{\text{LASSO}}(N)}{t_{\text{IRFS}}(M, K)} \tag{25}$$

where $t_{\text{IRFS}}(M, K)$ is the CPU time required to perform M IRFS iterations across K PLS factors and $t_{\text{LASSO}}(N)$ is the CPU time required to perform N LAR iterations. For example, if $\theta = 2$ for a given set of values for N, M, and K, then IRFS is twice as fast as LASSO. In short, if $\theta > 1$ or $\theta < 1$, then IRFS is faster or slower, respectively, than LASSO. In Figure 3, the speedup θ values are shown on a $\log_{10}$-scale for $M = 0, 2, 4, 6$. (Note that $M = 0$ corresponds to the initial IRFS iterate, i.e., ordinary PLS regression.) The x-axis and y-axis tick marks correspond to PLS factors K and LASSO iterations N, respectively. The "hot" colors (red, orange, and yellow) and "cool" colors (blue, indigo, and violet) correspond to the parameter regions where IRFS is faster and slower than LASSO, respectively. For example, an orange pixel value of $\log_{10}(\theta) = 2$ corresponds to a parameter setting of K, N, and M values that results in IRFS being $10^2 = 100$ times faster than the DTU-LAR LASSO implementation. As expected, whenever the maximum number of LASSO iterations is sufficiently high, or when the number of PLS factors is small, IRFS will outperform LASSO in terms of speedup.

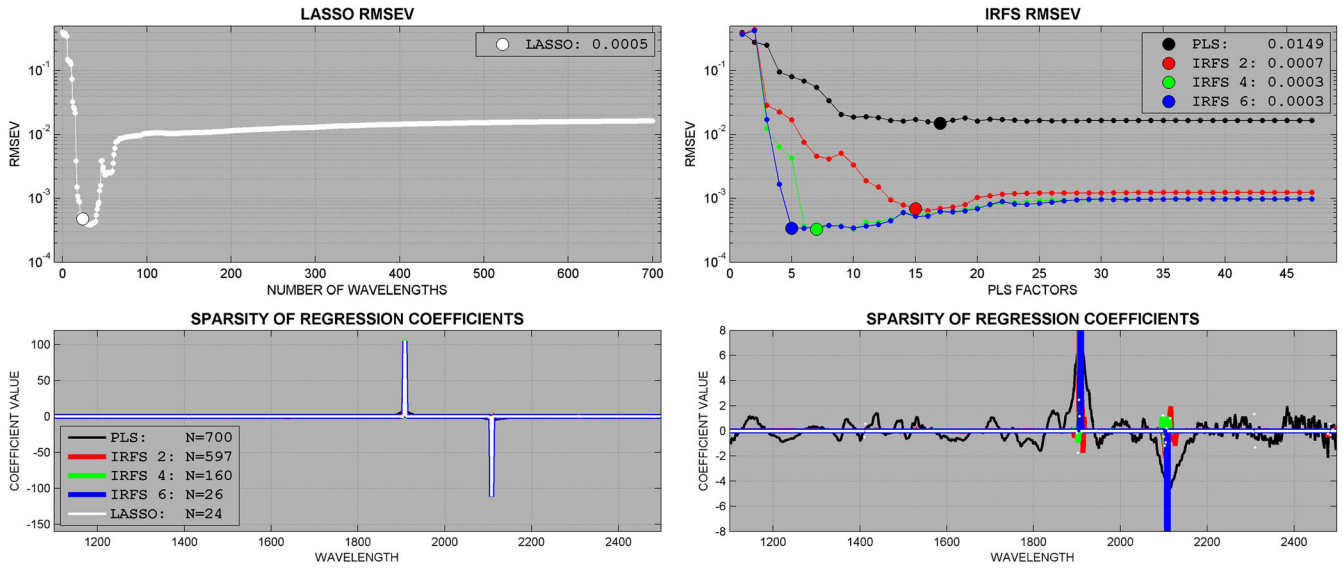

Figure 2. Primal sparsity performance on a corn data set using maximum likelihood estimate scheme. The upper left subplot shows root-mean-square error of validation (RMSEV) as a function of the number of wavelengths used in the calibration model. The large white dot corresponds to the RMSEV value associated with the calibration model chosen by the M_α-test. The upper-right subplot similarly shows RMSEV performance as a function of partial least squares (PLS) factor, for a fixed number of iteratively reweighted feature scaling (IRFS) iterations. The large black, red, green, and blue dots similarly correspond to the RMSEV values associated with the calibration models chosen by the M_α-test. The lower-left subplot displays, for each wavelength, the regression vector coefficient associated with least absolute shrinkage and selection operator (LASSO) and IRFS. Also shown is the sparsity, that is, the number of nonzero coefficients associated with each regression vector. The lower-right subplot is the same as the lower-left subplot except that the y-axis is restricted to the interval [–8, 8]. LASSO, least absolute shrinkage and selection operator.

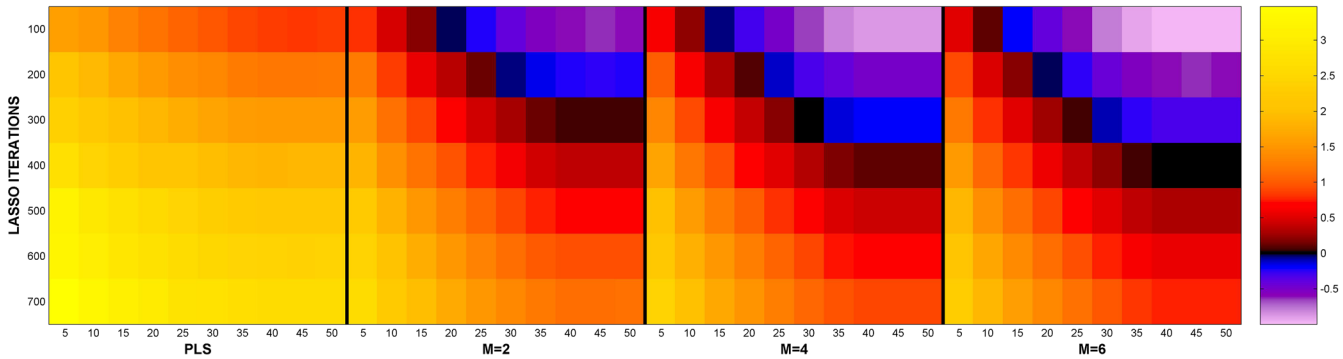

Figure 3. Speedup of iteratively reweighted feature scaling (IRFS) over least absolute shrinkage and selection operator (LASSO) across IRFS iterations. The colors indicate the value of θ in Equation (25) on a $\log_{10}$-scale. The x-axis indicates the number of partial least squares factors, whereas the y-axis indicates LASSO iterations (the number of wavelengths used in the calibration model.)

5.3. Sample selection example using the wheat data set

We now give an example for the comparison between SVR-LASSO and IRFS for sample selection. In this case, we consider the wheat data set and apply the APPLS scheme. Because there are 777 samples in the calibration set, one can ask if all samples are needed for an effective calibration model, that is, can SVR-LASSO and/or IRFS build a well-performing primal regression vector **b** from a parsimonious dual solution vector **a** containing relatively few support vectors?

In the upper-left subplot of Figure 4, one does not need all 777 samples to build a model that outperforms a model using all wavelengths. In the upper-right subplot, an argument can be made that no sample selection is needed because ordinary PLS (the black curve) performs just as well as IRFS. However, as the lower-left subplot shows, as the IRFS iterations increase, one needs fewer and fewer samples to build a model vector that performs just as well (or slightly better) as PLS. After four and six IRFS iterations, only 2.45% (32 samples out of 777) and 1.67% (19 samples out of 777) of the calibration data are needed for adequate model construction.

5.4. Summary root-mean-square error of validation and sparsity results

Figure 5 displays the RMSEV results (associated with the calibration models chosen by the M_α-test) across IRFS schemes

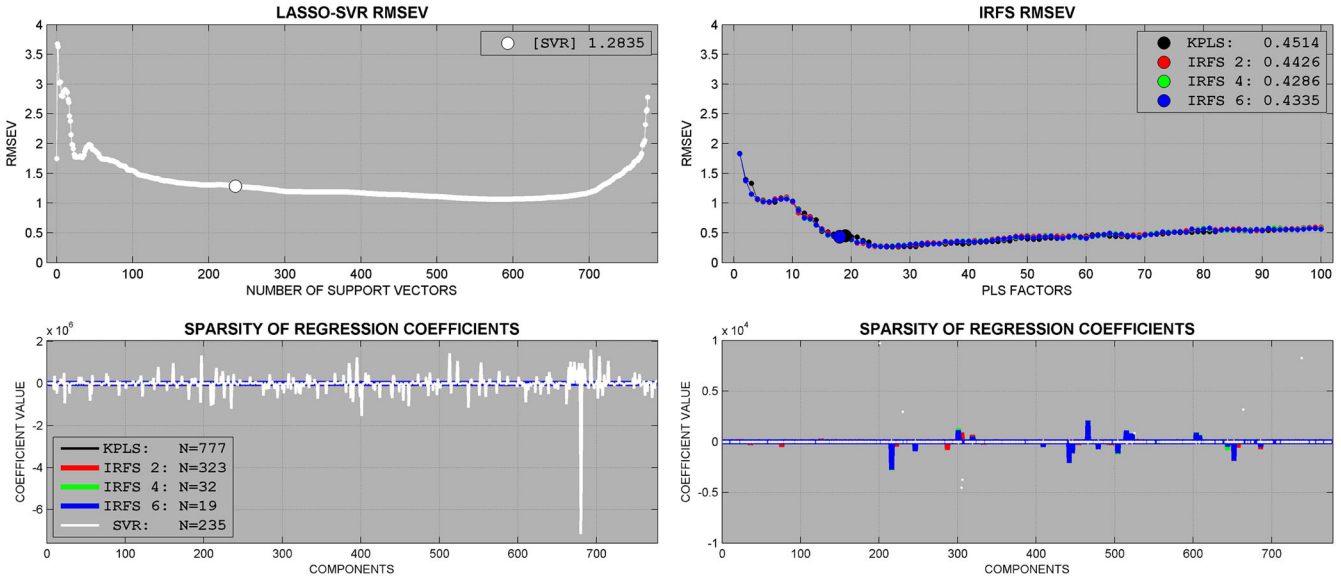

Figure 4. Dual sparsity performance on wheat data set using adaptively preconditioned partial least squares (PLS) scheme. The upper-left subplot shows root-mean-square error of validation (RMSEV) as a function of the number of support vectors used in the calibration model. The large white dot corresponds to the RMSEV value associated with the calibration model chosen by the M_α-test. The upper-right subplot similarly shows RMSEV performance as a function of PLS factor, for a fixed number of iteratively reweighted feature scaling (IRFS) iterations. The large black, red, green, and blue dots similarly correspond to the RMSEV values associated with the calibration models chosen by the M_α-test. The lower-left subplot displays, for each sample, the regression vector coefficient associated with least absolute shrinkage and selection operator (LASSO) and IRFS. Also shown is the number of support vectors (the number of nonzero coefficients) associated with each regression vector. The lower-right subplot is the same as the lower-left except the *y*-axis has been restricted to the interval [–10000, 10000]. SVR, support vector regression; KPLS, kernel PLS.

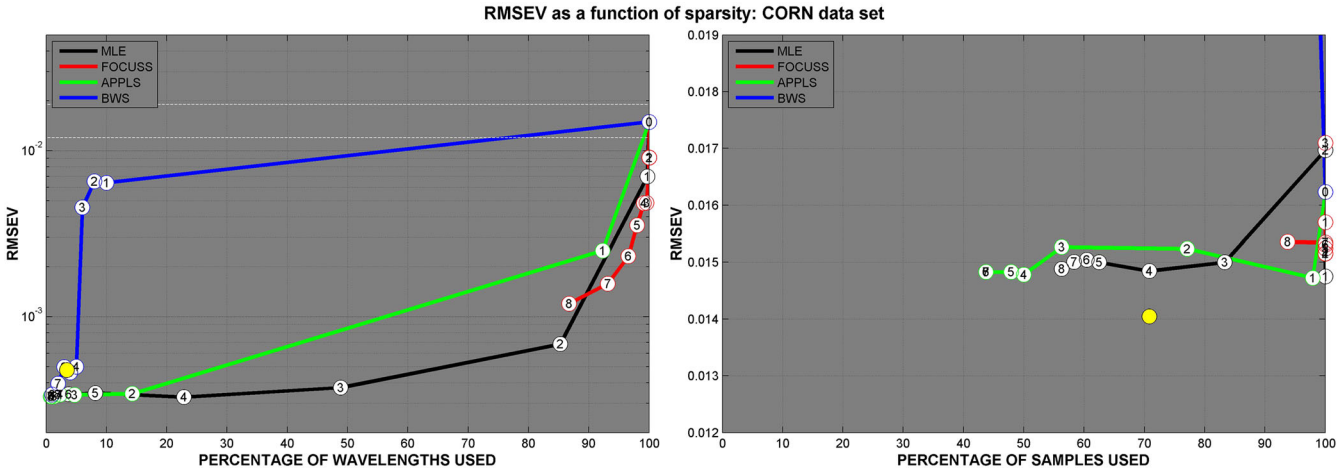

Figure 5. For the iteratively reweighted feature scaling (IRFS) schemes , the root-mean-square error of validation (RMSEV) values (associated with the calibration models chosen by the M_α-test) are plotted as a function of the percentage of wavelengths used (left subplot) and samples used (right subplot). The number inside the white dots corresponds to the IRFS iteration. The yellow dot in each subplot indicates the least absolute shrinkage and selection operator performance associated with the calibration model chosen by the M_α-test. The two white horizontal lines on the left subplot correspond to the *y*-axis limits of the right subplot. MLE, maximum likelihood estimate; FOCUSS, focal underdetermined system solver; APPLS, adaptively preconditioned partial least squares; BWS, binary weighting scheme.

for the corn data set. The *x*-axes on the left and right subplots correspond to the percentage of wavelengths and samples, respectively, used in the calibration. The number inside each white dot corresponds to the IRFS iteration associated with the particular (percentage,RMSEV)-ordered pair. The white dot numbered with a zero on the left subplot corresponds to PLS—it achieved an RMSEV of 0.0149 using all 700 wavelengths. Similarly, the white dot numbered with a zero on the right subplot corresponds to kernel PLS—it achieved an RMSEV of 0.0162 using all 48 samples. The performance of LASSO on the corn data set is shown in two places—as the yellow dot in Figure 5 and in Table IV—it achieved an RMSEV of 0.0005 and 0.0140 using 25 (out of 700) wavelengths and 35 (out of 48) samples, respectively. For a sense of scale, the two white horizontal

Table IV. RMSEV results for wavelength and sample selection using the LASSO

	Wavelength Selection		Sample Selection	
	RMSEV	Percentage	RMSEV	Percentage
Corn	0.0005	3.57% (25)	0.0140	72.91% (35)
Wheat	0.516	4.57% (48)	1.284	30.37% (236)

RMSEV, root-mean-square error of validation; LASSO, least absolute shrinkage and selection operator.
The number in parentheses indicate the number of wavelengths and samples used.

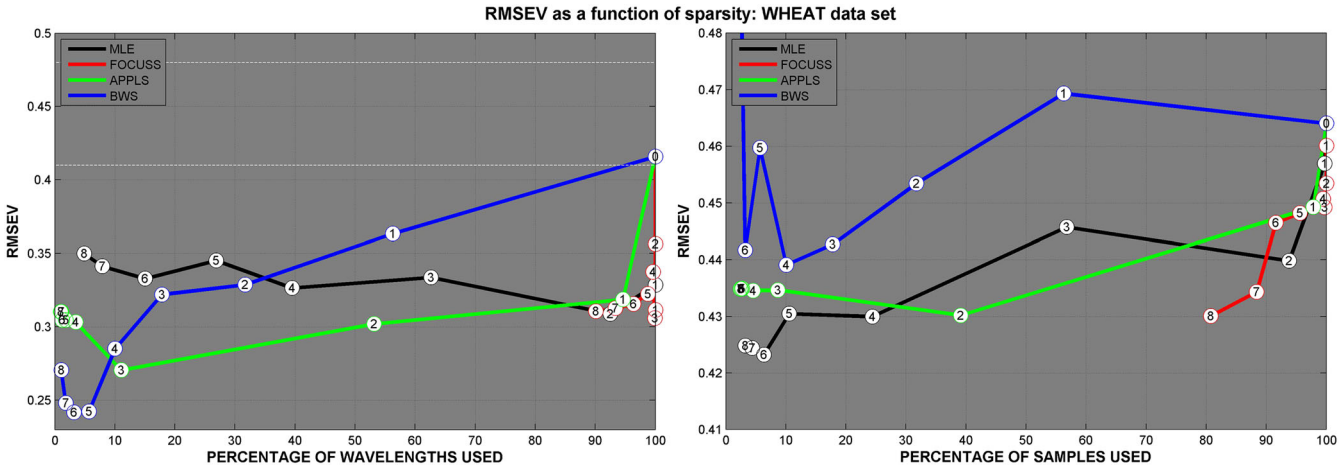

Figure 6. Root-mean-square error of validation (RMSEV) performance for the wheat data set as a function of the percentage of wavelengths and samples used. The same description convention is used as in Figure 5. MLE, maximum likelihood estimate; FOCUSS, focal underdetermined system solver; APPLS, adaptively preconditioned partial least squares; BWS, binary weighting scheme.

lines on the left subplot correspond to the *y*-axis limits of the right subplot.

Wavelength selection of any kind (with the exception of FOCUSS and early-iteration BWS) results in superior performance relative to ordinary PLS with DTU-LASSO being slightly inferior to the MLE and APPLS schemes. The APPLS scheme is quite striking considering that, after two iterations, only 15% of the wavelengths are being used. With respect to sample selection, the improvement over kernel PLS in RMSEV using SVR-LASSO or any of the IRFS schemes (except BWS) is more modest. The BWS scheme performs miserably because the number of samples kept per iteration (in a data set that has relatively few samples) decreases exponentially. The APPLS scheme requires the fewest number of samples (or support vectors), whereas the FOCUSS scheme (with $p = 1$) is ineffective in driving down both the number of samples and wavelengths.

Figure 6 corresponds to the wheat data set, and it follows the same description convention as Figure 5. Here, it is always preferential to perform some form of wavelength or sample selection. Again, the APPLS scheme consistently shows improvement in performance when using just a few IRFS iterations. After just four iterations, APPLS requires just 3.3% and 4.1% of the wavelengths and samples, respectively. As in the corn data set, the BWS scheme also performs admirably in wavelength selection, provided the number of IRFS iteration is relatively high. Even though wavelength selection was performed on wheat samples grown in different years than that of the validation samples, the RMSEV performance was superior to ordinary PLS that used all samples. Unlike the corn data set, sample selection using any of the IRFS schemes (excluding BWS) results in substantial improvement over kernel PLS and the DTU-LAR SVR-LASSO implementation (the yellow dot corresponding to SVR-LASSO on the right subplot is not shown—it is above the displayed *y*-axis limits).

6. CONCLUSION AND FUTURE WORK

In this paper, we propose simpler alternatives to the LASSO and SVR for the purposes of wavelength and sample selection, respectively. These alternatives—the IRFS schemes—require nothing more than the iterative recycling of regression coefficients from widely adopted regression techniques such as PLS, PCR, or RR. The boost in performance, whether it be RMSEV or the small number of wavelengths or samples used in the calibration model, can be substantial. Moreover, only a few IRFS iterations are required to see improvement in performance, making these IRFS alternatives competitive with respect to computational efficiency.

With respect to future work, one could perform both wavelength and sample selection in an alternating fashion to create a minimal set of samples (rows) and wavelengths (columns). Although coupling wavelength/sample selection with regression was the task of interest in this paper, one could easily apply the IRFS schemes to the binary classification scenario. Instead of working with linear kernels, the IRFS versions of SVR could also be extended to nonlinear kernels. In short, IRFS provides a versatile and pragmatic framework for continued research in spectroscopic applications.

REFERENCES

1. Anderson CR, Bro R. Variable selection in regression—a tutorial. *J. Chemom.* 2010; **24**(11–12): 728–737.
2. Balabin RM, Sergey SV. Variable selection in near-infrared spectroscopy: benchmarking of feature selection methods on biodiesel data. *Anal. Chim. Acta* 2011; **692**: 63–72.
3. Claerbout J, Muir F. Robust modeling of erratic data. *Geophysics* 1973; **38**: 826–844.
4. Taylor HL, Banks SC, McCoy JF. Deconvolution with the l_1-norm. *Geophysics* 1979; **44**: 39–52.
5. Levy S, Fullagar P. Reconstruction of a sparse spike train from a portion of its spectrum and application to high-resolution deconvolution. *Geophysics* 1981; **46**: 1235–1243.
6. Santosa F, Symes WW. Linear inversion of band limited reflection seismograms. *SIAM J. Sci. Stat. Comp.* 1986; **7**: 1250–1254.
7. Tibshirani RJ. Regression shrinkage and selection via the LASSO. *J. R. Stat. Soc. Ser. B* 1996; **58**: 267–288.
8. Tibshirani RJ, Saunders M, Rosset S, Zhu J, Knight K. Sparsity and smoothness via the fused LASSO. *J. R. Stat. Soc. Ser. B* 2005; **67**: 91–108.
9. Yuan M, Lin Y. Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. Ser. B* 2005; **68**(1): 49–67.
10. Zou H. The adaptive lasso and its oracle properties. *J. Am. Stat. Assoc.* 2006; **101**(476): 1418–1429.
11. Hastie T, Taylor J, Tibshirani RJ, Walther G. Forward stagewise regression and the monotone LASSO. *Electron. J. Stat.* 2007; **1**: 1–29.
12. Fu W. Penalized regressions: the bridge versus the lasso. *J. Comput. Graph. Stat.* 1998; **7**(3): 397–416.
13. Efron B, Johnstone I, Hastie T, Tibshirani RJ. Least angle regression. *Ann. Stat.* 2004; **32**(2): 407–499.
14. Friedman J, Hastie T, Hofling H, Tibshirani RJ. Pathwise coordinate optimization. *Ann. Appl. Stat.* 2007; **1**(2): 302–332.
15. Sparselab: seeking sparse solutions to linear systems of equations, Standford University. http://sparselab.stanford.edu/ [Accessed on February 12, 2013].
16. Liu J, Ji S, Ye J. SLEP: a sparse learning package, Arizona State University. http://www.public.asu.edu/~jye02/Software/SLEP [Accessed on February 12, 2013].
17. Schmidt M, Fung G, Rosales R. Fast optimization methods for L1 regularization: a comparative study and 2 new approaches. Proceedings of 2007 European Conference on Machine Learning: Warsaw, Poland, 2007. http://www.di.ens.fr/~mschmidt [Accessed on February 12, 2013].
18. Mørup M, Clemmensen L. Multiplicative updates for the lasso. Proceedings of 2007 IEEE International Workshop on Machine Learning for Signal Processing: Thessaloniki, Greece, 2007. http://www.mortenmorup.dk/ [Accessed on February 12, 2013].
19. Tikhonov A. Solution of incorrectly formulated problems and the regularization method. English translation of *Dokl. Akad. Nauk. SSSR* 1963; **151**: 501–504.
20. Hoerl A. Application of ridge analysis to regression problem. *Chem. Eng. Prog.* 1962; **58**: 54–59.
21. Kalivas JH, Siano GS, Andries E, Goicoechea HC. Tikhonov regularization approaches for calibration maintenance and transfer. *Appl. Spectrosc.* 2009; **63**(7): 800–809.
22. Kunz MR, Ottaway J, Kalivas JH, Andries E. Impact of standardization sample design on Tikhonov regularization variants for spectroscopic calibration maintenance and transfer. *J. Chemometr.* 2010; **24**: 218–229.
23. Kunz MR, Kalivas JH, Andries E. Model updating for spectral calibration maintenance and transfer using 1-norm variants of Tikhonov regularization. *Anal. Chem.* 2010; **82**(9): 3642–3649.
24. Kalivas JH. Application of 2-norm (L2) and 1-norm (L1) regularization variants for full wavelength and sparse spectral multivariate calibration models and maintenance. *J. Chemometr.* 2012; **26**: 218–230.
25. Gorodnitsky I, Rao B. Sparse signal reconstruction from limited data using FOCUSS: a re-weighted minimum norm algorithm. *IEEE T. Signal Proces.* 1997; **45**(3): 600–615.
26. Rao B, Kreutz-Delgado K. An affine scaling methodology for best basis selection. *IEEE T. Signal Proces.* 1999; **47**(1): 187–200.
27. Hastie T, Tibshirani RJ, Friedman JH. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer-Verlag: New York City, NY, 2001.
28. Candès EJ, Wakin M, Boyd S. Enhancing sparsity by reweighted l_1 minimization. *J. Fourier Anal. Appl.* 2007; **14**: 877–905.
29. Chartrand R, Yin W. Iteratively reweighted algorithms for compressive sensing. 2008 Proceedings of International Conference on Acoustics, Speech, Signal Processing (ICASSP), 3869–3872: Las Vegas, 2008.
30. Ottaway J, Kalivas JH, Andries E. Spectral multivariate calibration with wavelength selection using variants of Tikhonov regularization. *Appl. Spectrosc.* 2009; **64**(12): 1388–1395.
31. Hansen PC. Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion. SIAM Press: Philadelphia, PA, 1998.
32. Kondylis A, Whittaker J. Adaptively preconditioned Krylov spaces to identify irrelevant predictors. *Chemometr. Intell. Lab. Syst.* 2010; **104**: 205–213.
33. Guyon I, Weston J, Barnhill S, Vapnik V. Gene selection for cancer classification using support vector machines. *Mach. Learn.* 2002; **46**: 389–422.
34. Li H, Liang Y, Xu Q, Cao D. Key wavelengths screening using competitive adaptive reweighted sampling method for multivariate calibration. *Anal. Chim. Acta* 2009; **648**: 77–84.
35. Suykens J, Van Gestel T, De Brabanter J, De Moor B, Vandewalle J. Least Squares Support Vector Machines. World Scientific: Singapore, 2002.
36. Welling M. Kernel support vector regression. Max Welling's Classnotes in Machine Learning. http://www.ics.uci.edu/~welling/classnotes/classnotes.html [Accessed on February 12, 2013].
37. Welling M. Kernel support vector machines. Max Welling's Classnotes in Machine Learning. http://www.ics.uci.edu/~welling/classnotes/classnotes.html [Accessed on February 12, 2013].
38. Cristianini N, Shawe-Taylor J. An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press: Oxford, UK, 1999.
39. Lin CJ, Chang CC. LIBSVM: a library for support vector machines. *ACM TIST* 2011; **2**(3): 27:1–27:27, Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm [Accessed on February 12, 2013].
40. Fung G, Mangasarian O. Proximal support vector machine classifiers. In In *Proceedings KDD-2001: Knowledge Discovery and Data Mining, August 26-29, 2001, San Francisco, CA*, Provost F, Srikant R (eds). Association for Computing Machinery: New York, 2001; 77–86.
41. Peng X. TSVR: an efficient twin support vector machine for regression. *Neural Networks* 2009; **23**: 356–372.
42. Bertsekas B. Constrained Optimization and Lagrange Multiplier Methods. Athena Scientic: Belmonta, MA, 1982.
43. Franklin JN. Minimum principles for ill-posed problems. *SIAM J. Math. Anal.* 1978; **9**(4): 638–650.
44. SpaSM: a Matlab toolbox for performing sparse regression. *J. Stat. Softw.* (submitted), http://www2.imm.dtu.dk/projects/spasm/ [Accessed on February 12, 2013].
45. Haaland DM, Thomas E. Partial least-squares methods for spectral analyses. 1. Relation to other quantitative calibration methods and the extraction of qualitative information. *Anal. Chem.* 1988; **1988**: 1193–2002.
46. Andries E, Kalivas JH. Multivariate calibration leverages and spectral F-ratios via a filter factor representation. *J. Chemometr.* 2010; **24**(5): 249–260.
47. Varmuza K, Filzmoser P. Introduction to Multivariate Statistical Analysis in Chemometrics. CRC Press: Boca Raton, FL, 2009.
48. NIR of corn samples for standardization benchmarking. Eigenvector Research. http://www.eigenvector.com/data/Corn/index.html [Accessed on February 12, 2013].
49. Wheat functionality as measured by diffuse reflectance of whole grain Canadian wheat. 2008 International Diffuse Reflectance Conference: Chambersburg, PA, USA, 2008. http://www.idrc-chambersburg.org/ss20082012.html [Accessed on February 12, 2013].

