Chemometrics and Intelligent Laboratory Systems 143 (2015) 111–121

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

journal homepage: www.elsevier.com/locate/chemolab

Varying-coefficient single-index signal regression

Brian D. Marx

Department of Experimental Statistics, Louisiana State University, Baton Rouge, LA 70803, United States

ARTICLE INFO

Article history:
Received 26 September 2014
Accepted 6 February 2015
Available online 28 February 2015

Keywords:
Multivariate calibration
Tensor P-spline
Signal regression
Single-index
Varying-coefficient models
Unknown link function

ABSTRACT

The penalized signal regression (PSR) approach to multivariate calibration (MVC) assumes a smooth vector of coefficients for weighting a signal or spectrum to predict the unknown concentration of a chemical component. P-splines (i.e. B-splines and roughness penalties, based on differences) are used to estimate the coefficients. In this paper we allow the PSR coefficient vector to vary smoothly along a covariate (e.g. temperature), which results in a smooth surface on the wavelength–temperature domain. Estimation is performed using two-dimensional tensor product P-splines. As such, a slice of this surface effectively estimates the vector of coefficients at any arbitrary temperature. As an added generalization, we further relax the implicit assumption of an identity link function by allowing an unknown, but explicit, link function between the linear predictor and the response. Again, we allow the signal's link function to vary smoothly along a covariate, which produces a two-dimensional *link surface*. The unknown link surface is also estimated using two-dimensional P-splines, which is sliced at the same arbitrary temperature to bend prediction. Typically we use a common covariate (e.g. temperature) to vary the associated link function, as with the signal coefficients, but nothing prohibits the use of two different ones. We term our method: varying single-index signal regression (VSISR). The methods presented are grounded in penalized regression, where difference penalties are placed on the rows and columns of the tensor product coefficients. Each row and column of each surface has its own tuning parameter. An application to ternary mixture data illustrates that both the varying-coefficient and varying-nonlinearity (due to the link) are present. External prediction performance comparisons are made for both the identity link varying-coefficient penalized signal regression (VPSR) and partial least squares (PLS).

1. Introduction

In this paper, we take yet another approach to the multivariate calibration problem, in particular where the signal (spectra) regressors appear to have two-dimensional structure. Although we generally use the term *signal* throughout the paper, our application considers NIR spectra (taken over several temperatures). Through simultaneous estimation, we identify and estimate two separate modeling components, both of which are surfaces: (a) a single *smooth regression coefficient vector*, which effectively ensembles a smooth surface while varying along the temperature covariate [1], and (b) an unknown and nonlinear link function, which also varies along the temperature covariate, yielding a *link surface* and thus extends the work of Eilers, Li and Marx [2] and [3]. Although the first component is linear, the second component explicitly models the nonlinearity, allowing us to learn something about features of the transformed mean, which in some cases enhances insight into the process. We choose to use a common covariate (e.g. temperature) to vary the associated link function, as with the signal coefficients, but the interacting covariate could differ. We will see that the combination of these components can lead to a systematic and tractable modeling approach, that is statistical in nature, while in some cases having improved external prediction performance when compared to identity link model variants and partial least squares.

2. Motivating example

We revisit data used in [3], with permission from Zhenyu Wang and Age Smilde, where the response y comes from the composition (mole fraction) of a mixture, here consisting of three components (water, 1,2-ethanediol, 3-amino-1-propanol). These data are an expanded version of the data used in [4,5], and [6]. The ternary plot for the $m = 34$ mixtures is provided in Fig. 1. The center data point in the triangle represents equal concentrations of the three components, the edge points are mixtures containing only two components, and the corners are pure. Note that there are 3 pure, 12 edge, and 19 interior (1 center) mixtures. The components are modeled one at a time, and not jointly.

Corresponding to each ternary mixture, there exists an extremely rich spectroscopy regressor information, taken under $\breve{p} = 12$ temperature conditions: (30, 35, 37.5, 40, 45, 47.5, 50, 55, 60, 62.5, 65, 70 °C). Fig. 2 displays signal regressors (at only two different temperatures) for each of $m = 34$ observations. Each "signal" actually consists of

E-mail address: bmarx@lsu.edu.

http://dx.doi.org/10.1016/j.chemolab.2015.02.005
0169-7439/

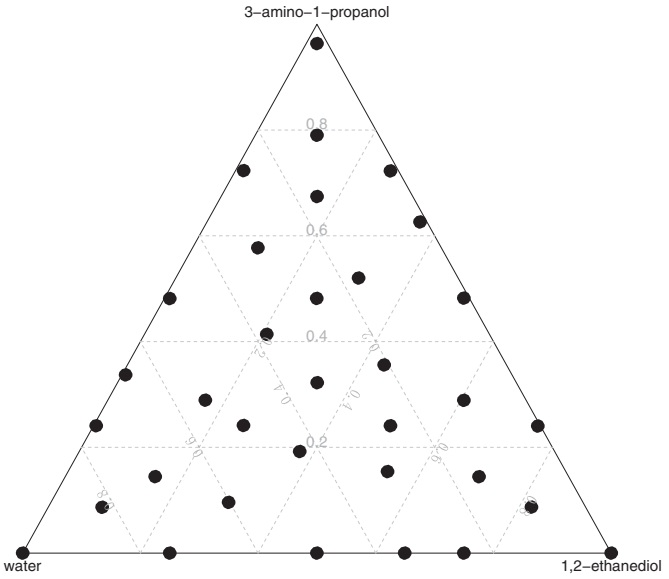

Fig. 1. Ternary plot for mixtures, with $m = 34$: 3 pure, 12 edge, 19 interior.

numerous digitizations ($p = 401$) along the wavelength axis v (700 to 1100, equally-spaced by 1 nm). The top (bottom) panels present the raw (first differenced) spectra. The latter will be our choice, which is in part attractive since constant shifts across spectra are removed.

Notice that the left and right panels of Fig. 2 present signals at the extreme temperature levels of 30° and 70 °C, respectively. One could imagine many more given (or interpolated) temperatures, resulting in a sequence of several "extremely narrow images" to build out a two-dimensional regressor surface.

2.1. Motivation for this paper

Thus a natural question to ask is: what is the true, or more importantly, the most useful regressor structure to predict y?

The primary goal is reliable future (external) prediction. The data set brings some unique structure and several challenges: (a) for all practical purposes, the response is measured *exactly* at the molar level, and only at several dozen concentrations. (b) The rich covariate information has dimension far greater (at least an order of magnitude greater) than the number of observations. (c) Internal prediction is *not* of interest, as it could be perfectly done, if desired, in infinitely many ways. (d) Oddly, it is the signal regressors themselves, and *not* the responses, that change with changes in the covariate t.

The data structure considered by Marx and Eilers [7] and Marx, Eilers, and Li [3] is rethought, where in the latter each of the $m = 34$ mixtures had one image regressor (400×12). As such, the composite of signal regressors was then viewed as fully two-dimensional, where spatial information was taken into account in both (the wavelength and temperature) directions, and this information was related to the response (component concentration). The problem was viewed in the light of a multivariate calibration with multi-dimensional spectra, where, e.g., the second dimension was temperature. Fig. 3 illustrates such a two-dimensional spectra structure with 4800 regressors, summarized in a 400×12 matrix (using first differences), for the center mixture unit, with corresponding scalar responses (water, 1,2-

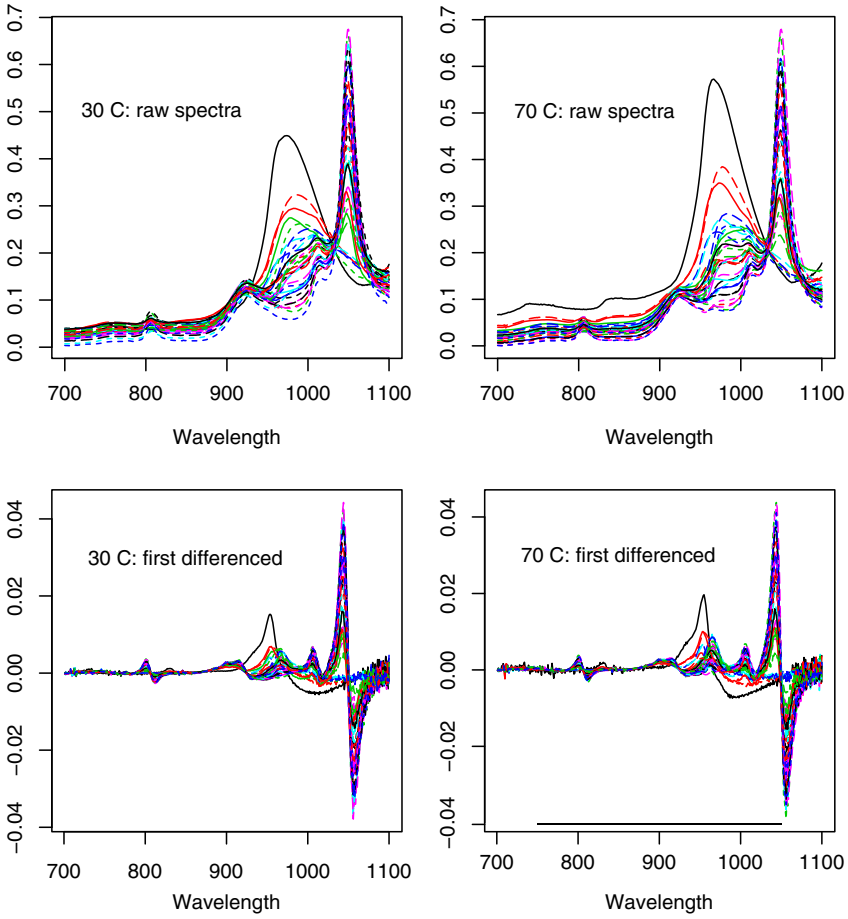

Fig. 2. Signal regressors (raw and first differenced) for mixture experiment, at two different temperatures.

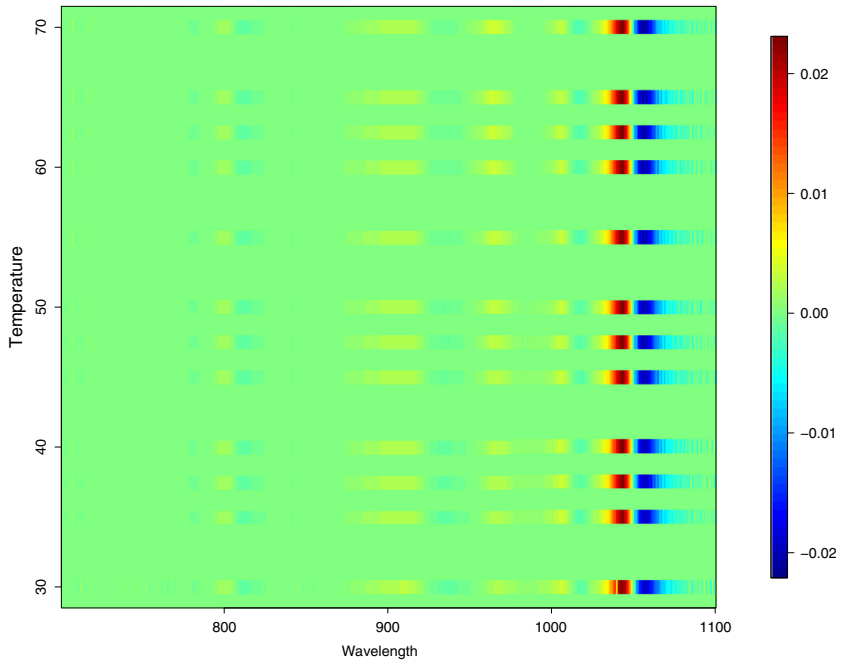

Fig. 3. Two-dimensional (first differenced) signal regressor image for center mixture.

ethanediol, 3-amino-1-propanol, each at $\frac{1}{3}$). We will come back to this data structure below in Section 10.

2.2. Realistic data structure

Perhaps it is more realistic to think of the additional covariate t as simply being recorded, while the signal regressors are generated. In this way the classical MVC approach could be more appropriately modified to allow interactive effects of t with the signal regressors. Such an approach is related to the work of Eilers and Marx [1]. Visually, one can think of unfolding the image in Fig. 3 into twelve separate univariate signals (each of dimension 400), each with a separately measured covariate. This structure plays right into the hands of an interesting varying-coefficient model: as temperature affects the signal regressor information, it stands to reason that their corresponding coefficients should also vary with t. With a smoothness constraints within the estimation of both the signal coefficients and the varying-index t, a coefficient surface gels into shape. In fact, cutting through the surface returns us to classical MVC regression vector, providing the coefficients with which to weigh the signal information. In this paper, we will focus on modifications and extensions of exactly this varying-coefficient signal regression model.

2.3. Some unique modeling contributions

A unique contribution of this work is the presence of a varying-link function associated with an already varying-coefficient signal regression model. In essence, the single-index signal regression model of Eilers, Li, and Marx [2] is generalized on two fronts: (i) the unknown, but explicit, link function is estimated within the context of a varying-coefficient signal model, and (ii) the link function is further allowed to vary itself, producing a link surface. Details of estimation and regularization of this *doubly*-varying model will follow below.

This paper additionally takes the opportunity to clearly delineate between two competing regressor structures, each of which produces a variant of an estimated two-dimensional coefficient surface. Despite similarities in appearance, distinctions quickly manifest: one model uses a one-dimensional signal regressor with measured covariate, whereas the other model uses a full two-dimension image regressor. This paper broadly unifies and bridges notation across the two penalized signal regression approaches of [1] and [7], clearly detailing specific similarities and differences between the modeling approaches. Although motivated through a real data mixture example, the presented notation is rich enough to generalize applications.

2.4. General data structure: triplet {y, x(v), t}

The specifics of this particular data structure are as follows: each observation consists of the data triplet: $\{y_i,\ x_i(v),\ t_i\}$, where $i = 1, \ldots, N$. The response y_i is again a scalar, and independence among the responses is assumed, with common variance $\mathrm{var}(y) = \sigma^2$. However, the signal is now one-dimensional, consisting of a $p \times 1$ vector of ordered regressors, e.g. a digitized signal, along the indexing (wavelength) axis v. In fact, we have $x_i(v)$. Completing the triplet is the covariate t_i, e.g. temperature, which is measured along with each (signal) x_i. The indexing axes v for x_i that define the support coordinates of x_i are very often equally-spaced, but again the only requirement for our method is that of a common support for all i or $v_i = v$. As suggested by Fig. 3, x_i could be viewed as a signal at each temperature slice, again with discrete digitized support (wavelength) with $p = 400$ channels (v takes values of 701 to 1100 nm, by 1 nm). In this example, the companion covariate t_i takes on only $\breve{p} = 12$ levels (30, 35, 37.5, 40, 45, 47.5, 50, 55, 60, 62.5, 65, 70 °C), but prediction could be made at any arbitrarily recorded t. An important distinction needs to be made regarding the response dimension: y is now $N \times 1$, with $N = m\breve{p}$, i.e. each recorded value of y is replicated $\breve{p}$ times. In the example above, y now has length $(34 \cdot 12) \times 1$, as the response is modeled using the signal information at each level of t.

3. Organization of this paper

In Section 4, we recap a general two-dimensional smoothing tool: tensor product B-splines. Such an initial presentation allows us to not only define central notation, but also frames smoothing in a regression setting. Section 5 then applies this two-dimensional smoother to construct a varying-coefficient signal regression paradigm. As we suggest

a rich tensor product basis, regularization is implemented in Section 6, defining the tensor product P-spline approach to estimation. The second modeling component is presented in Section 7, where now an additional two-dimensional link surface is defined and also estimated, along with the model fitting algorithm that simultaneously estimates both the varying-coefficient and varying-link surfaces. Suggestions for optimizing the tuning parameters (associated with the surfaces) are outlined in Section 8, and Section 9 follows with an illustrative example using the motivating ternary mixture data. In Section 10, we elaborate on the connections between alternative views of the data structure (i.e. data triplet vs. data pair with image). Lastly, we close with a Discussion.

4. Jump start notation: tensor product B-spline smoothing

Tensor product B-spline smoothing is a general tool used in two (or more) dimensions. Such a choice is often attractive because of its simplicity and ability to produce very general surfaces that are based in regression. As an added value, with the use of tensor products, derivatives of the smooth surface are easily computed. Among others, Eilers and Marx [1] (Section 4) provided a basic presentation *Tensor product B-splines in a nutshell*, giving essential details to building smooth surfaces, while more complete coverage can be found in various texts, e.g. Dierckx [8] (Chapters 1 and 2).

The essential building block is a bicubic basis function, which is the *tensor product* of the two univariate (cubic) B-splines, say B and $\breve{B}$ (placed on the margins). Despite more general possibilities, we only consider $n(\breve{n})$ equally-spaced B-spline basis functions that are placed along the $v(\breve{v})$ axis. Such a construction requires $n+2q$ $(\breve{n}+2\breve{q})$ knots, where q denotes the degree of the B-splines basis and is typically set to three (cubic). The extra $2q$ (external) knots ensure that there is enough support for the boundary basis functions. Ultimately, the $v\times\breve{v}$ plane is carved out into regularly-spaced subrectangles that support the $n\breve{n}$ bicubic basis functions. More specifically, the rth–sth single tensor product $B_r(v)\breve{B}_s(\breve{v})$ is positive in the rectangular region defined by the knots $R=\left[\varphi_r,\varphi_{r+q+2}\right]\times\left[\breve{\varphi}_s,\breve{\varphi}_{s+\breve{q}+2}\right]$ or on a support of spanned by $(q+2)\times(\breve{q}+2)$ knots. Similar to univariate B-splines, it is convenient to index each tensor product by one of the $n\times\breve{n}$ knot pairs and

$$\begin{aligned} B_r(v)\breve{B}_s(\breve{v})>0 \;\text{ for all }\; v,\;\breve{v}\in R \\ =0 \;\text{ for all }\; v,\;\breve{v}\notin R, \end{aligned} \tag{1}$$

$r=1,\ \ldots,\ n$ and $s=1,\ \ldots,\breve{n}$.

Fig. 4 sparsely displays nine (scaled) tensor product B-splines, which represents only a portion of a full basis. A complete tensor product B-splines basis thus has an unknown coefficient matrix, denoted by $\Gamma_{n\times\breve{n}}=[\gamma_{rs}]$. For given knot grid, a very flexible surface can be approximated, e.g. at the discrete digitized coordinates $(v,\breve{v})$. For $j=1,\cdots,\ p$ and $k=1,\cdots,\ \breve{p}$,

$$\alpha\left(v_j,\breve{v}_k\right)\ =\ \sum_{r=1}^{n}\sum_{s=1}^{\breve{n}}B_r\left(v_j\right)\breve{B}_s(\breve{v}_k)\gamma_{rs}. \tag{2}$$

The surface is in fact driven by relatively few parameters $(n\breve{n})$, such that changing Γ leads to changes in the surface.

4.1. Unfolding Γ and notation

It is computationally efficient to re-express the surface in "unfolded" notation. Before doing so, some further notation is needed. Denote the discrete digitized (support) coordinate matrix $C=\left(v\otimes\mathbf{1}_{\breve{p}},\ \mathbf{1}_p\otimes\breve{v}\right)$ of dimension $p\breve{p}\times 2$. Let the matrix B and $\breve{B}$ (with respective dimensions $p\breve{p}\times n$ and of $p\breve{p}\times\breve{n}$) be the univariate B-spline basis matrix evaluated at each entry of the first and second column of C, respectively. The unfolded expression at the support coordinates then has the standard multiple regression form

$$\mathrm{vec}\{\alpha(v,\breve{v})\}\ =\ \mathbf{T}\boldsymbol{\gamma}, \tag{3}$$

where $\boldsymbol{\gamma}\ =\ \mathrm{vec}(\Gamma)$. Define the matrix

$$\mathbf{T}=B\square\breve{B}=(B\otimes 1'_{\breve{n}})\odot\left(1'_n\otimes\breve{B}\right) \tag{4}$$

of dimension $p\breve{p}\times n\breve{n}$. The symbols $\otimes$ and $\odot$ denote Kronecker product and elementwise multiplication of matrices, respectively. In the applications to follow, the second indexing axis is set to temperature, i.e. $\breve{v}=t$.

5. First component: varying-coefficients for signal

We first consider modeling rich signal regressors, with coefficients that are allowed to interact with another covariate. We refer to this model as *varying-coefficient penalized signal regression* or VPSR. Recall our mixture example with y of dimension $N\times 1$ (each component is measured $\breve{p}$ times). Given the signal matrix $X=[x_{ij}]$ $(i=1,\ \ldots,\ N;\ j=1,\ldots,\ p)$ and the (unknown) coefficient surface $\alpha(v,\breve{v})$, the mean of the ith response can be denoted as

$$\mu_i=\sum_{j=1}^{p}x_{ij}\alpha\left(v_j,t_i\right). \tag{5}$$

The above expression explicitly shows the varying-coefficient structure that depends on t. Note that with the setting $\breve{v}=t_i$, the coefficient

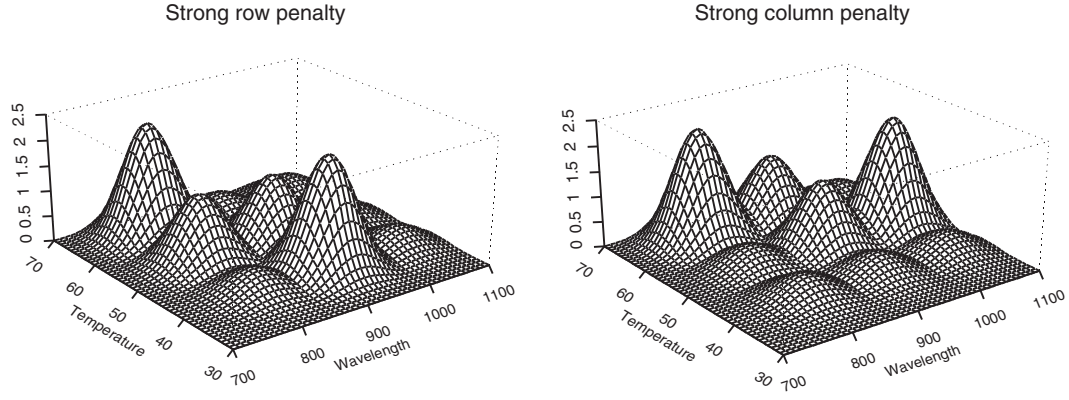

Fig. 4. Nine scaled cubic B-spline tensor products, with a strong linear row penalty (left panel) and a strong linear column penalty (right panel).

surface is evaluated only at one specific slice of the covariate. The single sum over j further highlights the weighting of a one-dimensional signal by varying coefficients, for given t. Approximating the coefficient surface by tensor product B-splines, it follows that

$$\begin{aligned}\mu_i &= \sum_{j=1}^{p} x_{ij} \overbrace{\sum_{r=1}^{n}\sum_{s=1}^{\breve{n}} B_r\left(v_j\right)\breve{B}_s(t_i)\gamma_{rs}}^{\alpha(v_j,t_i)} \\ &= \sum_{j=1}^{p}\sum_{r=1}^{n}\sum_{s=1}^{\breve{n}} x_{ij} b_{jr} \breve{b}_{is}\gamma_{rs} \\ &= \sum_{r=1}^{n}\sum_{s=1}^{\breve{n}}\left(\sum_{j=1}^{p} x_{ij} b_{jr}\right)\breve{b}_{is}\gamma_{rs} \\ &= \sum_{r=1}^{n}\sum_{s=1}^{\breve{n}} u_{ir}\breve{b}_{is}\gamma_{rs},\end{aligned} \tag{6}$$

which again is a tensor product expression, but now using a modified basis $U = XB$. Borrowing notation from Eq. (4), we find that Eq. (6) can be expressed in somewhat simplified matrix notation as $\mu = \mathbf{U}\gamma$, where

$$\mathbf{U} = U\square\breve{B} = (U\otimes 1_{\breve{n}}')\odot\left(1_n'\otimes\breve{B}\right), \tag{7}$$

is of dimension $N \times (n\breve{n})$. An important distinction between Eqs. (4) and (7) follows: now B is a $(p \times n)$ B-spline bases matrix (where rows represent evaluations at the p digitizations of the signal), whereas $\breve{B}$ is a $(m\breve{p} \times \breve{n})$ B-spline matrix (where rows represent evaluations at the $\breve{p}$ digitizations of the covariate t, each replicated m times). From this point forward, it is useful to think of $\mathbf{U}$ as the *effective* regressor matrix for the lower dimensional representation of the smooth coefficient surface.

6. P-spline objective for the VPSR coefficient surface

The P-spline approach to estimation of the coefficient surface requires two basic steps: (a) the $n \times \breve{n}$ knot pairs, associated with Eq. (2), are richly chosen to allow the surface more flexibility than it actually needs, i.e. it is purposely overfit. And the knots within each axis (v or $\breve{v}$) are equally-spaced. This step coursely enforces our choice to regularize the ill-posed regression problem. Said differently, smoothness across the two-dimensional coefficient structure is assumed to be either reasonable or non-detrimental toward future prediction. (b) Penalties are placed on γ vector, leading to regularized estimation. Specifically, a separate difference penalty is assigned to each row and to each column of the array of coefficients in Γ, which ensures that adjacent coefficients within the same row (or same column) do not differ too much from each other. Some further care is taken to assure that the linkage is broken in the penalty from one row to the next row and from one column to the next column. Associated with each penalty are nonnegative tuning parameters λ and $\breve{\lambda}$.

The penalized objective function to be minimized is defined as

$$\begin{aligned}Q_P(\gamma) &= \text{Residual SS} + \text{Row Penalty} + \text{Column Penalty} \\ &= \sum_{i=1}^{m}\left(y_i - \mathbf{u}_i'\gamma\right)^2 + \lambda\sum_{r=1}^{n}\gamma_{r\bullet}D_d'D_d\gamma_{r\bullet}' + \breve{\lambda}\sum_{s=1}^{\breve{n}}\gamma_{\bullet s}'D_{\breve{d}}'D_{\breve{d}}\gamma_{\bullet s} \\ &= ||y - \mathbf{U}\gamma||^2 + \lambda||P\gamma||^2 + \breve{\lambda}||\breve{P}\gamma||^2,\end{aligned} \tag{8}$$

where $\mathbf{u}_i'$ is the ith row of $\mathbf{U}$ and $\gamma_{r\bullet}$ ($\gamma_{\bullet s}$) denotes the rth row (the sth column) of Γ.

Fortunately, the penalties can be conveniently and compactly represented, again using Kronecker products and matrix notation: $P = (D_d'D_d)\otimes I_{\breve{n}}$ and $\breve{P} = I_n\otimes\left(D_{\breve{d}}'D_{\breve{d}}\right)$, where I denotes the identity matrix and d denotes the order of the difference penalty. Both P and $\breve{P}$ are square with dimension $n\breve{n}$. The matrices D_d and $D_{\breve{d}}$ are banded in structure, each have rows that consist of polynomial contrasts (see Marx and Eilers [9]), and have dimension $(n-d) \times n$ and $\left(\breve{n}-\breve{d}\right) \times \breve{n}$, respectively. The coefficients in γ are in vector form, yet P and $\breve{P}$ are cleverly constructed to align a difference penalty within each row (column) of the Γ array.

In practice, the order of the penalties (d, $\breve{d}$) are usually fixed by the user. The non-negative tuning parameters essentially allow continuous control over smoothness, but are chosen in a greedy way to minimize external prediction error. With only two tuning parameters, we choose to apply the same penalty weight for each row (column) of coefficients, however the row and column tuning parameters are independent of each other. Fig. 4 illustrates strong smoothing in action, i.e. large λ and $\breve{\lambda}$, using a second order penalty on each row and column. We find that the limiting behavior for each row and column is linear in this case, with reversals of slopes possible across rows (or columns).

For given tuning parameters ($\lambda, \breve{\lambda}$), the VPSR solution for objective (8) is

$$\hat{\gamma} = \left(\mathbf{U}'\mathbf{U} + \lambda P'P + \breve{\lambda}\breve{P}'\breve{P}\right)^{-1}\mathbf{U}'y. \tag{9}$$

The system of equations is of modest dimension $n\breve{n}$. The effective "hat" matrix can be of use and is defined as

$$H = [h_{i\breve{i}'}] = \mathbf{U}\left(\mathbf{U}'\mathbf{U} + \lambda P'P + \breve{\lambda}\breve{P}'\breve{P}\right)^{-1}\mathbf{U}'. \tag{10}$$

A future (external) predicted value is estimated as $\hat{y}_{new} = \mathbf{u}_{new}'\hat{\gamma}$, where $\mathbf{u}_{new}'$ is constructed as in Eq. (7) using the *new* digitized signal regressors and covariate t.

Of course, many other bases and penalty choices exist that lead to alternative regularizations. In Section 11, we will argue that our choice is a sensible one, while presenting some advantages relative to other options.

6.1. Modification for an intercept term

The model can also include an intercept term β_0 which results in the modified P-spline solution

$$\left(\hat{\beta}_0, \hat{\gamma}'\right)' = \left(\mathbf{U}_1'\mathbf{U}_1 + \lambda P_1'P_1 + \breve{\lambda}\breve{P}_1'\breve{P}_1\right)^{-1}\mathbf{U}_1'y,$$

with $\mathbf{U}_1 = \left(1_{m\breve{p}}|\mathbf{U}\right)$, $P_1 = (0|P)$, and $\breve{P}_1 = \left(0|\breve{P}\right)$. The zero vector in P_1 and $\breve{P}_1$ ensures an unpenalized intercept.

7. Defining and implementing a two-dimensional link surface

Although statistical models often offer meaningful scientific interpretation, they can have difficulties competing with the predictive ability of a broader class of approaches, e.g., support vector machines, genetic algorithms, and neural networks, among others. Yet, such "machine learning" approaches offer very little, if any, interpretability. There appears to be a general and unsatisfying trade-off between competitive prediction and scientific interpretability; gains in one come at a compromise of the other. For all they provide, one possible deficiency of MVC statistical approaches (including PSR) is that they are rather narrow in scope, i.e.: their prediction quality is limited to estimated coefficients that are linear in the signal regressors. To allow additional model flexibility, we capture the nonlinear features of the response process through a general (unknown) link function, that itself can vary across the level of a covariate, yielding a *varying-link surface*.

7.1. The second component: varying link function

We consider the extended model of the form

$$\mu = f(\mathbf{U}\gamma, t), \tag{11}$$

where f is a two-dimensional function of both the linear predictor ($\eta = \mathbf{U}\gamma$) and the covariate t. We refer to this model as *varying-coefficient single-index signal regression* or VSISR. In fact, due to the construction of $\mathbf{U}$ in Eq. (7), each row $\mathbf{u_i}'$ requires the exact level of the covariate t; thus one can think of the second argument in $f(\cdot,\cdot)$ as companion information for the first argument. This is an important point for calculations of derivatives for the surface f: since η specifies the level of t, we will find that we only need partial derivatives of f with respect to η.

As presented in Eq. (11), we have a two-dimensional surface imbedded within another two-dimensional surface. We have: (i) the varying-coefficient surface for the signal regressors, which is estimated with tensor product P-splines coefficients γ, and (ii) now the varying-link surface, which is again estimated with separate tensor product P-spline coefficients θ. The objective in Eq. (8) can be extended as

$$Q_P^\star(\gamma,\theta) = ||y-f(\mathbf{U}\gamma,t)||^2 + \lambda||P\gamma||^2 + \breve{\lambda}||\breve{P}\gamma||^2 + \lambda_f||P_f\theta||^2 + \breve{\lambda}_f||\breve{P}_f\theta||^2. \tag{12}$$

For given tensor B-spline coefficient vector γ, the linear predictor η is known and the estimation of function $f(\eta, t)$ becomes a straight-forward two-dimensional smoothing problem. We can thus apply the tensor product approach outlined in Section 4.1, where $v = \eta$, $\breve{v} = t$, and the unfolded tensor matrix $\mathbf{T}$ has support on unique values of η and t. The number of equally-spaced basis functions $(n_f, \breve{n}_f)$, as well as the order of the penalties $(d_f, \breve{d}_f)$, used to estimate θ can be chosen independently of γ. With tensor product P-splines, smooth θ leads to smooth $f(\eta, t)$, where the $\lambda_f, \breve{\lambda}_f > 0$ control roughness.

P-splines are a natural choice for several reasons: (a) such smoothers are regression based and are easy to use and optimize. (b) Heavy smoothing along one variable, using a dth order penalty, leads to polynomial structure of order d-1. Such polynomial structure could be linear, while allowing interactive effects across another variable. (c) The *partial* first derivative of f with respect to one variable (denoted as $\dot{f}_\partial(\eta,t) = \frac{\partial f(\eta,t)}{\partial \eta}$) can be easily computed (and is needed in our algorithm).

The derivative of a smooth function that is constructed with equally-spaced B-splines has the useful mathematical expression $B_{(q-1)}(\Delta\theta)/b$, where q is the degree, Δ denotes the first difference operator, and b is the step length on the equally-spaced knots. It stands to reason that partial derivatives for tensor product B-spline surfaces can be found: apply the above derivative calculation using adjacent bicubic functions within each row for all rows (or within each column for all columns).

For simplicity in notation, denote $\Upsilon\left\{\mathbf{V}, W, \left(\lambda_f, \breve{\lambda}_f\right), \left(d_f, \breve{d}_f\right), \left(n_f, \breve{n}_f\right)\right\}$ as the operation of fitting a tensor product cubic P-spline scatter smoother on $\mathbf{V} = \{v, \breve{v}\}$ (the input variables) and W (the response) using the penalty tuning parameters (λ_f, λ_f) and difference orders $(d_f, \breve{d}_f)$ on the $n_f \times \breve{n}_f$ knot grid.

7.2. The VSISR model fitting algorithm

For fixed estimate of the surface f (i.e. given θ), and given covariate t, the coefficient vector γ can be estimated using a (first-order) Taylor series approximation of the function f (about the current estimate, γ_0). Specifically, if γ_0 is the current estimate for γ, then the current estimate of $\mu = f(\mathbf{U}\gamma, t)$ can be approximated by

$$f(\mathbf{U}\gamma,t) \approx f(\mathbf{U}\gamma_0,t) + \mathrm{diag}\left\{\dot{f}_\partial(\mathbf{U}\gamma_0,t)\right\}\mathbf{U}(\gamma-\gamma_0). \tag{13}$$

Using Eq. (13), with fixed f and given t, we have an approximation of $Q_P^\star$

$$\begin{aligned} Q_P^\star &\approx ||y-f(\mathbf{U}\gamma_0,t)-\mathrm{diag}\left\{\dot{f}_\partial(\mathbf{U}\gamma_0,t)\right\}\mathbf{U}(\gamma-\gamma_0||^2 + \lambda||P\gamma||^2 + \breve{\lambda}||\breve{P}\gamma||^2 \\ &= ||\left[y-f(\mathbf{U}\gamma_0,t) + \mathrm{diag}\left\{\dot{f}_\partial(\mathbf{U}\gamma_0,t)\right\}\mathbf{U}\gamma_0\right] - \mathrm{diag}\left\{\dot{f}_\partial(\mathbf{U}\gamma_0,t)\right\}\mathbf{U}\gamma||^2 \\ &\quad + \lambda||P\gamma||^2 + \breve{\lambda}||\breve{P}\gamma||^2 \\ &= ||y^\star - \mathbf{U}^\star\gamma||^2 + \lambda||P\gamma||^2 + \breve{\lambda}||\breve{P}\gamma||^2, \end{aligned} \tag{14}$$

where $y^\star = y - f(\mathbf{U}\gamma_0, t) + \mathrm{diag}\{\dot{f}_\partial(\mathbf{U}\gamma_0, t)\}\mathbf{U}\gamma_0$ and $\mathbf{U}^\star = \mathrm{diag}\{\dot{f}_\partial(\mathbf{U}\gamma_0, t)\}\mathbf{U}$. Note that Eq. (14) implies that given f (or θ), the optimal γ that minimizes the right-hand side of Eq. (14) can be obtained through the varying-coefficient penalized signal regression solution provided in Eq. (9), VPSR$\left\{\mathbf{U}^\star, y^\star, \left(\lambda, \breve{\lambda}\right), \left(d, \breve{d}\right), \left(n, \breve{n}\right)\right\}$. Hence, in our algorithm, we first initialize the linear predictor (equivalently γ) using (an identity link) VPSR($\mathbf{U}$, y) (Step 1). Then, given γ, an estimate of f (equivalently θ) is obtained (Step 2). The two steps, estimation of γ and θ, are iterated until convergence. For simplicity of presentation, the intercept term is suppressed ($\beta_0 = 0$) in the algorithm.

Algorithm VSISR.

1. Initializations:
- Fix the tuning parameter values $(\lambda, \breve{\lambda}, \lambda_f, \breve{\lambda}_f)$ for Steps 1 and 2
- Fix number of knots $(n, \breve{n}, n_f, \breve{n}_f)$ and penalty orders $(d, \breve{d}, d_f, \breve{d}_f)$
- Create $\mathbf{U} = U \square \breve{B}$
- Initialize $\hat{\gamma} = \mathrm{VPSR}\left\{\mathbf{U}, y, \left(\lambda, \breve{\lambda}\right), \left(d, \breve{d}\right), \left(n, \breve{n}\right)\right\}$
- Initialize $\hat{\eta} = \mathbf{U}\hat{\gamma}$

2. Cycle until convergence of $\hat{\gamma}$
- Estimate $\hat{f}(\hat{\eta}, t)$ and the estimate of the derivative $\dot{f}_\partial(\hat{\eta}, t)$ from $\Upsilon\left\{\{\eta, t\}, y, \left(\lambda_f, \breve{\lambda}_f\right), \left(d_f, \breve{d}_f\right), \left(n_f, \breve{n}_f\right)\right\}$
- Obtain $y^\star$ and $\mathbf{U}^\star$
- Update $\hat{\gamma} = \mathrm{VPSR}\left(\mathbf{U}^\star, y^\star, \left(\lambda, \breve{\lambda}\right), \left(d, \breve{d}\right), (n, \breve{n})\right)$
- Update $\hat{\eta} = \mathbf{U}\hat{\gamma}$
- Constrain $\hat{\gamma}/||\hat{\gamma}||$

3. Prediction: $\hat{y}_{new} = \hat{f}(\mathbf{u}'_{new}\hat{\gamma}, t_{new})$

end algorithm

As an identifiability constraint, the vector γ is set to have a unit L_2 norm, i.e. $\gamma^{cur}/||\gamma^{cur}||$. To define the convergence criterion, denote $||\gamma||^2 = \sum_{k=1}^{n}\gamma_k^2$. The algorithm terminates when

$$\frac{(\gamma_k^{cur}/||\gamma^{cur}||) - (\gamma_k^{pre}/||\gamma^{pre}||)}{\gamma_k^{cur}/||\gamma^{cur}||} < \epsilon$$

for $k = 1, \ldots, n\breve{n}$, where γ^{cur} (γ^{pre}) is the γ vector for the current (previous) iteration, and ϵ is a prespecified convergence tolerance (default value is 10^{-3}). We will use the final iteration for some fitting summaries.

8. Optimization of the penalty tuning parameters

We suggest an approach to optimize P-spline tuning parameters that is consistent with the one found in [3].

First, we provide some practical guidelines and default values for the VSISR design parameters. We only consider bicubic tensor basis functions ($q = \breve{q} = 3$). As stated, the number of tensor product B-spline basis functions (for either the coefficient surface or the link surface) should be generous enough to allow more flexibility than needed, but at the same time consider computational efficiency; we suggest keeping both $n\breve{n} < 1000$ and $n_f\breve{n}_f < 1000$. We further suggest using a second or third penalty order for $(d, \breve{d}, d_f, \breve{d}_f)$. When searching for optimal $(\lambda, \breve{\lambda}, \lambda_f, \breve{\lambda}_f)$, we recommend varying each tuning parameter on a logarithmic (base 10) grid, while monitoring a performance criterion. Details on how to optimally regulate the amount of smoothness for the coefficient and link surfaces follow next.

Reliable external prediction is the primary goal. There are numerous approaches that "optimize" the four non-negative tuning parameters $\left(\lambda, \breve{\lambda}, \lambda_f, \breve{\lambda}_f\right)$. Alternative optimization strategies are certainly possible. We propose the following scheme (which is also consistent with other analyses of these same mixture data): the data are split into three disjoint and exhaustive groups, denoted as the: *training* set, *validation* set, and *test* set, with $m = m^{train} + m^{valid} + m^{test}$. To start, apply VSISR to the training set and choose "optimal" $(\lambda, \breve{\lambda}, \lambda_f, \breve{\lambda}_f)$ to minimize error on the validation set,

$$\text{RMSEV} = \sqrt{\frac{1}{N^{valid}}\sum_{i=1}^{N^{valid}}(y_i - \hat{y}_{vi})^2}, \tag{15}$$

where $N^{valid} = \breve{p}\, m^{valid}$ is the number of observations in the validation set across all $\breve{p}$ levels of the covariate t, and $\hat{y}_{vi}$ is the predicted response for the ith subject in the validation set, using the parameter estimates from the training model.

Given a chosen "optimal" model, evaluation of external predictive performance can be calculated using the root-mean-square error of prediction (RMSEP) on the independent test set:

$$\text{RMSEP} = \sqrt{\frac{1}{N^{test}}\sum_{i=1}^{N^{test}}(y_i - \hat{y}_i)^2}, \tag{16}$$

where $N^{test} = \breve{p}\, m^{test}$ is the number of observations on the test set across all $\breve{p}$ levels of the covariate t, and $\hat{y}_i$ is the predicted response for the ith subject in the external test set, using the parameter estimates from the combined (training, validation) sets with the "optimal" $\left(\lambda, \breve{\lambda}, \lambda_f, \breve{\lambda}_f\right)$.

Generally, we perform a full or a clever (four dimensional) linear grid search, where each tuning parameter element is varied on a logarithm scale. It is true that there are a variety of approaches to potentially "automatically" select the tuning parameters, but we prefer a grid search that is driven by external prediction performance, which is the ultimate objective and utility of the model.

9. VSISR illustration using ternary mixture data

Recall the motivating example presented in Section 2. Fig. 1 depicted the $m = 34$ levels the molar concentrations of each mixture component: water, 1,2-ethanediol, and 3-amino-1-propanol. We only consider one component at a time, i.e. in turn, each component will be individually modeled. Note that there are 3 pure, 12 edge, and 19 interior (1 center) mixtures. A more detailed description of the experimental setting can be found in the appendix of [3].

Despite that there are $m = 34$ unique mixtures, we have a total of $N = m\breve{p} = 34 \cdot 12 = 408$ observations. Under our scenario each mixture is placed under $\breve{p} = 12$ different temperature conditions (covariate: t): $\breve{v} = \{30, 35, 37.5, 40, 45, 47.5, 50, 55, 60, 62.5, 65, 70\ ^\circ\text{C}\}$. Optical information (regressor: x) is also obtained for each mixture–temperature combination, producing a (first-differenced) spectra at $p = 400$ wavelengths: $v = \{701$ to 1100 nm, by 1 nm$\}$. Thus for given mixture component in units of proportion (response: y), our data triplet $\{y_i,\ x_i(v),\ t_i\}$ is complete (Section 2.4), for $i = 1, \ldots, N$. The data were not preprocessed in any other way.

We focus on an external prediction performance study that directly compares the proposed VSISR method to the standard VPSR method [1] and to partial least squares (PLS). A further goal, and added value of VSISR, is to exploit and explore the nonlinear effect through the estimated link surface.

9.1. Optimal tuning for mixture data

As with other studies, we divided the $m = 34$ observation into three subsets as follows. The training set consisted of $m^{train} = 16$ observations using the 3 pure, 12 edge, and 1 center mixtures. The remaining 18 interior observations (apart from the center) were divided into a validation set (to optimize tuning parameters) and a test set (to quantify quality of external prediction): (i) they were first sorted on the response (either on water or 1,2-ethanediol or 3-amino-1-propanol) in increasing order. (ii) The validation (test) set was constructed using the $m^{valid} = 9$ ($m^{test} = 9$) even (odd) rank of observations. Such an approach was taken in an attempt to have a fair and reasonable range of mixture levels for both the estimation of f and the evaluation of external prediction. Additionally, there is no extrapolation for model optimization or model testing.

Optimal tuning parameters were determined by minimizing RMSEV in the trained model. Given these optimal tuning parameters, external prediction was evaluated on the test data using RMSEP using the newly trained model that combined both the training and validation data. We stress that in the determination of every optimal model, all hyper-parameters (λs or number of PLS components) are optimized using a validation set which is independent of the external test set.

For VSISR, we performed a combination of four-dimensional clever and grid searches, where each $log(\lambda), log\left(\breve{\lambda}\right), log(\lambda_f), log\left(\breve{\lambda}_f\right)$ was varied from -11 to 3 in fifteen steps, at minimum. Once a region was identified, a refined search was then further investigated. Specifically the same search approach was performed but with narrower focus, usually plus or minus a power of ten from the initial optimum, in five steps. A similar search strategy was taken for VPSR, but only on two tuning parameters: $log(\lambda), log\left(\breve{\lambda}\right)$. For PLS, we varied the number of components from 1 to 70, by 1. The number of equally-spaced knots were set to $n \times \breve{n} = 40 \times 20$ (wavelength, temperature) for the coefficient surface and $n_f \times \breve{n}_f = 10 \times 10$ (η, temperature). Second order difference penalties ($d = \breve{d} = d_f = \breve{d}_f = 2$) were used. All reported prediction performance refers to external prediction on the test data.

9.2. External prediction comparison: VSISR, VPSR, PLS

Table 1 presents the root mean square error of prediction (RMSEP) for the external prediction set, using optimal VSISR, VPSR, and PLS

Table 1
VSISR, VPSR, PLS external prediction RMSEP using optimal models.

Response	VSISR	VPSR	PLS
Water	**0.0087**	0.0129	0.0367
1,2-Ethanediol	**0.0094**	0.0104	0.0134
3-Amino-1-propanol	0.0146	**0.0063**	0.0099

Table 2
Optimal tuning parameters for VSISR, VPSR, and PLS models.

	VSISR	VPSR	PLS
Response	$(\lambda,\ \breve{\lambda},\ \lambda_f,\ \breve{\lambda}_f)$	$(\lambda,\ \breve{\lambda})$	Components
Water	$(2\ 10^{-1},\ 3\ 10^{-9},\ 5\ 10^{-3},\ 10^{4})$	$(2\ 10^{-10},\ 6\ 10^{-3})$	8
1,2-Ethanediol	$(2,\ 2\ 10^{-10},\ 2\ 10^{-1},\ 10)$	$(2\ 10^{-10},\ 4\ 10^{-1})$	58
3-Amino-1-propanol	$(10^{-3},\ 10^{-11}$[a], $0.1,\ 10)$	$(10^{-11}$[a], $10)$	50

[a] Light (boundary) smoothing required.

models. Table 2 provides the optimal tuning parameters for each method and mixture response.

For responses water and 1,2-ethanediol, we find an impressive improvement in external prediction for VSISR when compared to either VPSR or PLS. RMSEP reductions ranged from 33% to 77% (for water) and from 10% to 30% (for ethanediol). For VSISR, the external RMSEP values were in fact both less than 0.001, which when multiplied by 100 gives units of percent mixture (in terms of standard error). The tuning parameters were chosen based on minimizing RMSEV values, which achieved values of 0.0089, 0.0146, 0.0324 for water and 0.0091, 0.0132, 0.0287 for 1,2-ethanediol, for VSISR, VPSR and PLS, respectively.

For the response 3-amino-1-propanol, the standard VPSR performed best of all three methods (with RMSEP of 0.0063), suggesting that an identity link was sufficient. VPSR had a 37% reduction in RMSEP when compared to PLS. However, for this component, PLS did out-perform VSISR. VSISR and VPSR required light smoothing for one axis of the estimated two-dimensional coefficient, i.e. along wavelength and temperature, respectively. For tuning, the optimal RMSEV values were 0.0094, 0.0067, 0.0170 for VSISR, VPSR and PLS, respectively.

9.3. Comments on findings

Associated with each mixture component, Figs. 5, 6, and 7 each provides image plots for the varying-coefficient surface (upper), the varying-link surface (lower). In practice, temperature t is measured. One can think of slicing these two surfaces (on left) at various t to get the signal coefficients (for the linear predictor) and the link function (to bend the mean), respectively. The right panels display various slices of each surface. The three figures have some similarities, e.g. each of the upper, right panels clearly demonstrates how the signal coefficients vary, rather dramatically, across temperature. We note that, for each mixture component, light smoothing (small $\breve{\lambda}$ in Table 2) was required for VSISR coefficient surfaces in the temperature direction, indicating a strongly varying signal coefficient vector. Further, each link surface (lower, left panels) displays some torsion suggesting a varying link across temperature. The slices of the link surfaces (lower, right panels) suggest a different insight for each component: (a) for water, a clear nonlinearity is present, but there is some suggestion that a common (non-varying) link would be sufficient; (b) for Ethanol, again we find a clear nonlinearity link (inverted relative to the water link), with the apparent need for a varying-link; (c) for Isopropanol, we observe that a simple identity link may be sufficient, which is consistent with the fact that VPSR (identity link) performed strongly in RMSEP. The dashed line in the lower, right panels displays the identity link for reference.

We could additionally vary the order of the penalty and evaluate RMSEV and RMSEP, e.g. changing second order to a first or third order. In this way, penalty order would be an additional hyper-parameter. Nonetheless, the above results make a reasonably strong case for VSISR, and we do not explore this further.

9.4. Further prediction performance comparison: VSISR, VPSR, PLS

For each mixture component, we additionally compared the *external* prediction performance of VSISR, VPSR, and PLS against each other through RMSEP. To investigate how sensitive each method is to the choice of the data splitting, fifty (50) random independent splits were made for each method and component combination, such that again there were: training (16), validation (9) and testing (9) observations (using all temperatures). Fig. 8 displays the results, which further supports and is consistent with the findings in Table 1, as well as provides evidence that VSISR can achieve highly competitive external prediction performance when compared to VPSR or PLS, especially for Water. For 1,2-ethanediol, VSISR and VPSR appear to be equally competitive,

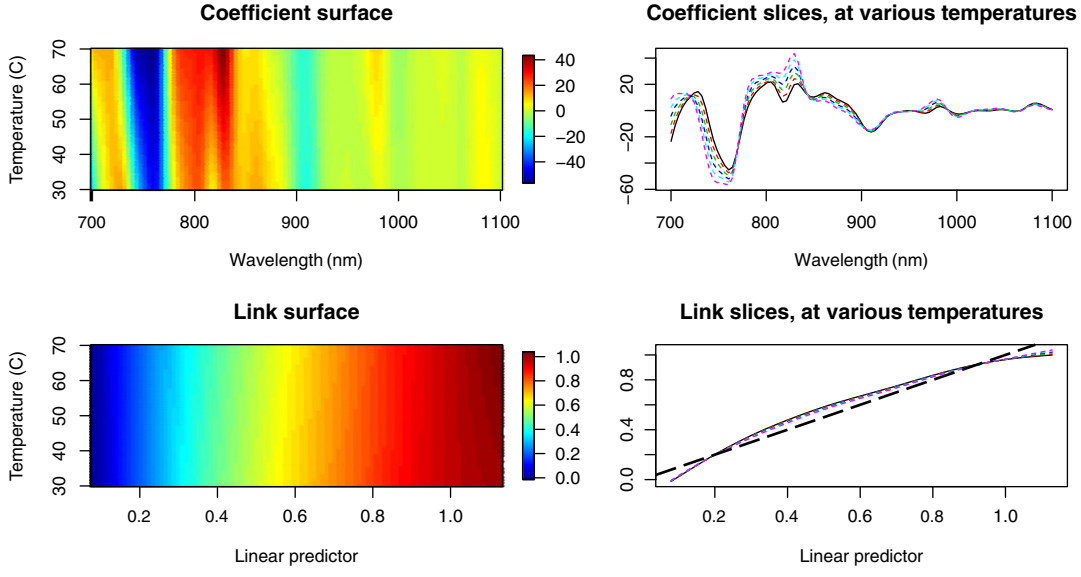

Fig. 5. Water component: optimal varying signal coefficient surface (top panels); optimal varying link surface (bottom panels). Dashed line displays identity link as reference.

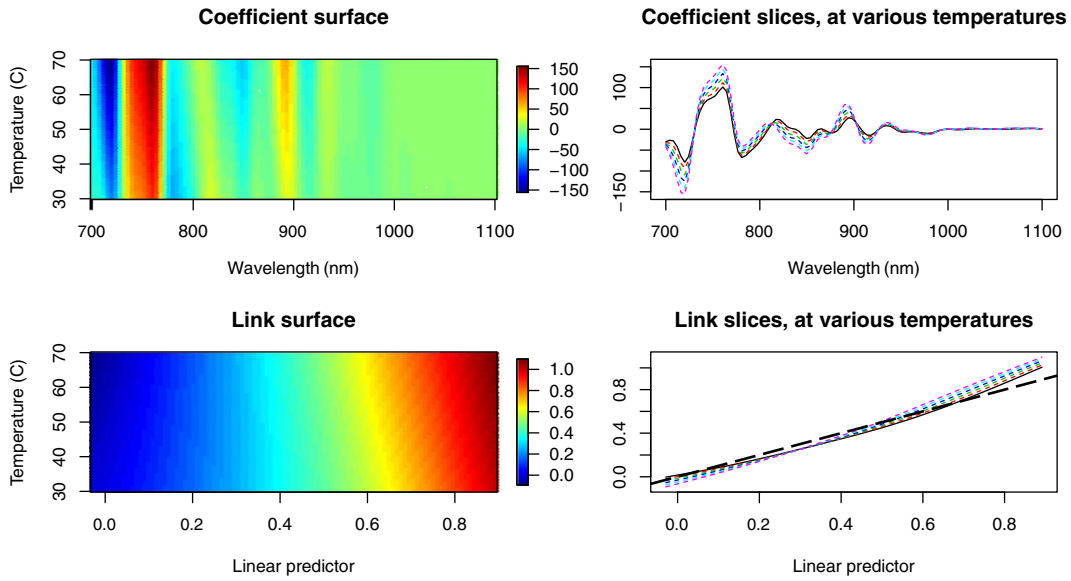

Fig. 6. 1,2-Ethanediol component: optimal varying signal coefficient surface (top panels); optimal varying link surface (bottom panels). Dashed line displays identity link as reference.

while outperforming PLS. While VPSR generally outperformed both VSISR and PLS for 3-amino-1-propanol.

For each random split, the tuning parameters for each method were optimized using RMSEV with respect to its validation set. As previously, the 18 non-training data were ranked with the evens (odds) used for validation (test) set. For VSISR and VPSR, the number of knots and order of the penalty were set as above, and a greedy search was performed on $\left(\lambda, \breve{\lambda}, \lambda_f\right)$, from -11 to 2 (in steps of 1) on a logarithmic scale, to optimize these tuning parameters. Note that $\breve{\lambda}_f$ was the most insensitive and was thus pre-set to 10. For PLS, a search for the optimal number of components again ranged from 1 to 70.

10. Alternative view of data structure: 2D data pair (y, X)

As pointed out in Section 2.3, one goal of this paper is to clearly delineate between two competing data structures, specifically distinguishing between the VSISR/VPSR approaches and their image regressor analogues. Marx, Eilers and Li [3,7] viewed the regressors as fully two-dimensional, along the lines of the image in Fig. 3. They termed their methods as MSISR and MPSR, respectively, which led to the multivariate calibration with "multi-dimensional" signals. The two-dimensional signal consists of (often thousands of) digitized regressors, arranged in a $p \times \breve{p}$ array, which can be generally viewed as an *image*. A common smooth coefficient surface is estimated that creates a linear predictor. Rather than a slice, the entire coefficient surface is used to predict the response. For the mixture data, each image regressor has dimension (400×12). In its unfolded form X has dimension (34×4800). The indexing axes, i.e. v and $\breve{v}$, that define the support coordinates of the image regressors are usually on a regular grid, but the only requirement for our method is that the scatter of digitizations are common across observations.

Table 3 displays the various modeling components for both the varying and multidimensional approaches.

One may ask to make a performance comparison between the varying methods and the multidimensional method. If we look at

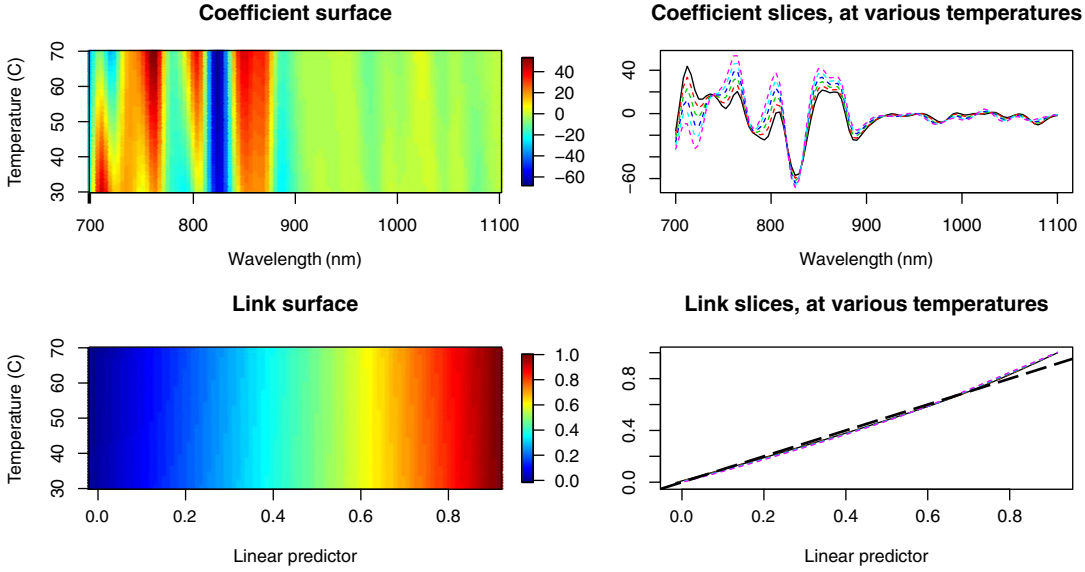

Fig. 7. 3-Amino-1-propanol component: optimal varying signal coefficient surface (top panels); optimal varying link surface (bottom panels). Dashed line displays identity link as reference.

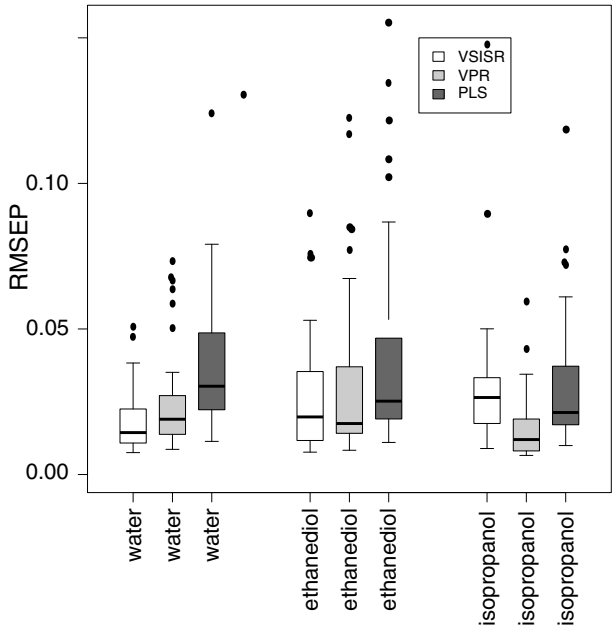

Fig. 8. Boxplot of comparative test errors for VSISR, VPSR, and PLS, based on 50 random splits of the data.

Table 1 of [3], we find that the RMSEP external prediction for MSISR/MPSR suffers when compared to that of VSISR/VPSR, for the same data and splitting. Specifically, with MSISR/MPSR, we had RMSEP in the approximate range of 0.02 to 0.03, which is about 200% to 300% greater than VSISR/VPSR. PLS in the multidimensional setting performed even worse. Admittedly, the numbers stated above are not "directly comparable" to VSISR/VPSR in that the MSISR/MPSR approaches used a third order (not second order) penalties, and the MSISR approach only used a one-dimensional link function. Regarding comparability, perhaps the most salient point is that the number of observations differs by a factor of $\breve{p}$, across data structures. Yet the message is relatively clear that VSISR/VPSR are models of choice for these data. Moreover, Section 2.2 provided a strong argument that the more natural data structure is the measured signal with companion covariate.

In either data structure case, the number of regressors far exceeds the number of observations and thus an ill-conditioned regression problem is posed. If that were not enough, the signal regressors are typically severely collinear, causing yet another level of difficulties to classical modeling approaches. Ironically, more precise information in the data leads to restrictions in the model: parameter estimation requires some form of regularization, and our choice is smoothness. As an added note: each of the varying or multidimensional approaches additionally takes advantage of the ordered or array structure among the regressors. Further each of these approaches welcomes increased precision on the signals, covariate, or images, as the system of equations remains of dimension $n\breve{n}$ (or $n_f\breve{n}_f$).

Table 3
Model specifics comparing varying signal model to 2D image model.

	VPSR/VSISR	MPSR/MSISR
μ_i	$\sum_{j=1}^{p} x_{ij}\alpha(v_j, t_i)$	$\sum_{j=1}^{p}\sum_{k=1}^{\breve{p}} x_{ijk}\alpha(v_j, t_k)$
Data	(y, X, t)	(y, X)
Length (y)	$m\breve{p}$	m
Regressor type	signal	image
Dimension X	$m\breve{p} \times p$	$m \times p\breve{p}$
Dimension B	$p \times n$	$p\breve{p} \times n$
Dimension $\breve{B}$	$m\breve{p} \times \breve{n}$	$p\breve{p} \times \breve{n}$
η	$(XB)\square\breve{B}\gamma$	$X(B\square\breve{B})\gamma$

11. Discussion

We have referred to the combination of VPSR and SISR as *varying single-index signal regression* or VSISR. Indeed, varying-coefficient models have come a long way from the seminal work of Hastie and Tibshirani [10]. The VSISR model is roughly related to "projection pursuit" [11], with additional smoothness constraints, while a rich two-dimensional "missing link" function is estimated in the spirit of Cox [12] and Muggeo and Ferrara [13]. The basic appeal of VSISR includes the following:

- Its simplicity, with doubly-varying coefficient and link surfaces.
- The indexing associated with the surfaces can identify potentially "important" regions.
- The nonlinear structure is targeted, providing potential process insight.
- Each smooth surface can have a very general (non-additive) structure.
- We have highly competitive external prediction ability.
- There is no "black box" algorithm.
- No data preprocessing: the entire signals are used with companion covariates.
- Unlike many competitors, VSISR takes advantage of the ordered structure in the signal.
- Heavy penalization defaults to polynomial structure.
- The regularization manages the severe ill-conditioned model (and collinear data).
- As signal precision increases, the system of equations remains $n\breve{n}$.

Future research for VSISR could include:

- Models that constrain the sum of mixture concentrations to be one.
- Model extensions that also allow other (smooth) covariates or factors, similar to the work found in [14].
- An exploration of how the link function affects prediction stability during calibration transfer, i.e. investigating the robustness of prediction quality as an additional covariate changes, e.g. temperature [15].
- Take a generalized linear model approach to VSISR, e.g. binary classification or Poisson counts [7] (Section 6) and [16].

Conflict of interest

There is no conflict of interest.

References

[1] P.H.C. Eilers, B.D. Marx, Multivariate calibration with temperature interaction using two-dimensional penalized signal regession, Chemom. Intell. Lab. Syst. 66 (2003) 159–174.
[2] P.H.C. Eilers, B. Li, B.D. Marx, Multivariate calibration with single-index signal regression, Chemom. Intell. Lab. Syst. 96 (2009) 196–202.
[3] B.D. Marx, P.H.C. Eilers, B. Li, Mulidimensional single-index signal regression, Chemom. Intell. Lab. Syst. 109 (2011) 120–130.
[4] F. Wülfert, W. Kok, A. Smilde, Influence of temperature on vibrational spectra and consequences for the predictive ability of multivariate models, Anal. Chem. 70 (1998) 1761–1767.
[5] F. Wülfert, W. Kok, O. de Noord, A. Smilde, Linear techniques to correct for temperature induced spectra variation in multivariate calibration, Chemom. Intell. Lab. Syst. 51 (2000) 189–200.
[6] F. Wülfert, W. Kok, O. de Noord, A. Smilde, Correction of temperature-induced spectral variation by continuous piecewise standardization, Anal. Chem. 72 (2000) 1639–1644.
[7] B.D. Marx, P.H.C. Eilers, Multidimensional penalized signal regression, Technometrics 47 (2005) 13–22.
[8] P. Dierckx, Curve and Surface Fitting with Splines, Clarendon Press, Oxford, 1995.
[9] P.H.C. Eilers, B.D. Marx, Flexible smoothing with B-splines and penalties (with comments and rejoinder), Stat. Sci. 11 (1996) 89–121.

[10] T. Hastie, R. Tibshirani, Varying-coefficient models (with discussion), J. R. Stat. Soc.Ser. B 55 (1993) 757–796.
[11] J.H. Friedman, W. Stuetzle, Projection pursuit regression, J. Am. Stat. Assoc. 76 (1981) 817–823.
[12] C. Cox, Generalized linear models — the missing link, J. R. Stat. Soc. Ser. C 33 (1984) 18–24.
[13] V. Muggeo, G. Ferrara, Fitting generalized linear models with unspecified link function: a P-spline approach, Comput. Stat. Data Anal. 52 (2008) 2529–2537.
[14] P.H.C. Eilers, B.D. Marx, Generalized linear additive smooth structures, J. Comput. Graph. Stat. 11 (2002) 758–783.
[15] B.D. Marx, P.H.C. Eilers, Mulivariate calibration transfer: a comparison of methods, J. Chemom. 16 (2002) 1–12.
[16] B.D. Marx, P.H.C. Eilers, B. Li, Generalized single-index signal regression, Technical Report, Department of Experimental Statistics, Louisiana State University, 2010–1, 2010.