Latent Variable Method Demonstrator — software for understanding multivariate data analytics algorithms

Joachim Schaeffer [a], Richard D. Braatz [b],*

[a] *Control and Cyber-Physical Systems Laboratory, Technical University of Darmstadt, Karolinenpl. 5, Darmstadt, 64289, Germany*
[b] *Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, 02139, MA, USA*

ARTICLE INFO

ABSTRACT

The ever-increasing quantity of multivariate process data is driving a need for skilled engineers to analyze, interpret, and build models from such data. Multivariate data analytics relies heavily on linear algebra, optimization, and statistics and can be challenging for students to understand given that most curricula do not have strong coverage in the latter three topics. This article describes interactive software – the Latent Variable Demonstrator (LAVADE) – for teaching, learning, and understanding latent variable methods. In this software, users can interactively compare latent variable methods such as Partial Least Squares (PLS), and Principal Component Regression (PCR) with other regression methods such as Least Absolute Shrinkage and Selection Operator (lasso), Ridge Regression (RR), and Elastic Net (EN). LAVADE helps to build intuition on choosing appropriate methods, hyperparameter tuning, and model coefficient interpretation, fostering a conceptual understanding of the algorithms' differences. The software contains a data generation method and three chemical process datasets, allowing for comparing results of datasets with different levels of complexity. LAVADE is released as open-source software so that others can apply and advance the tool for use in teaching or research.

1. Introduction

Latent variable methods such as partial least squares are among the most widely applied data analytics tools for applications to chemical, physical, and biological processes. It can be challenging to teach these multivariable statistical methods in a way that the students are able to consistently apply these methods effectively in practice, especially considering that many undergraduate programs do not require that their students take courses in linear algebra or applied statistics. The result is that many practitioners use alternative methods instead, such as correlating a peak absorbance to each concentration when calibrating a spectral measurement, which can produce models of much lower accuracy. Alternatively, the latent variable methods are often used in a black-box manner, i.e., without understanding when a situation occurs in which the methods should not be applied, or how to revise the way that data are fed to the method to work around some imperfections in the data such as sensor bias. After all, many software packages are available for applying latent variable methods, and it might seem that a deep understanding is not needed for problem-solving.

A lack of understanding, however, makes it challenging to explain and interpret the models created by these methods, particularly in relating the models generated with the chemistry/physics/biology occurring in the process, to reconcile the data analytics results with domain knowledge. The synthesis of domain knowledge with data analytics is the added value of a well-trained applied data scientist/engineer and is likely to result in the best solutions for the particular problem at hand. Furthermore, preliminary black-box results can lead to choosing overly complicated models that overfit the data. Advancing the understanding and intuition on latent variable methods are needed to avoid overfitting for some types of biased data and ultimately assure model interpretability, leading to higher value, acceptance, and applicability.

This article describes software and examples developed to train students to achieve a deep understanding of latent variable methods (e.g., Chiang et al., 2000; Mardia, 2003) and the related machine learning methods of lasso (Tibshirani, 1996), ridge regression (Hoerl and Kennard, 1970), and elastic net (Zou and Hastie, 2005). The graphical user interface is designed for the explicit purpose of teaching undergraduate and graduate students, which is a distinguishing feature from the graphical user interfaces in existing chemometrics software packages that are focused on just applying a method to a dataset. The software takes the perspective of the optimization being solved to help the understanding of the relationship between the latent variable method selected and the results produced.

---

* Corresponding author.
  E-mail address: braatz@mit.edu (R.D. Braatz).

This tool, referred to as the Latent Variable Demonstrator (LAV-ADE), compares a wide range of latent variable regression and other techniques for four different datasets. The examples are designed to be easy to understand, and various options to customize the problem are available to learn precisely how the different algorithms approach the model construction.

This article is organized as follows. Section 2 describes the latent variable and other regression techniques in the software. Section 3 describes the software design and architecture from a high-level perspective, and Section 4 describes the datasets. Section 5 shows illustrative examples of the software and is followed by the conclusion.

2. Regression techniques

The regression models considered in this article are linear, static models with the general form

$$y = \beta_0 x_0 + \beta_1 x_1 + \cdots + \beta_{n-1} x_{n-1} + \epsilon \tag{1}$$

where the output y, inputs x_i, and regression coefficients β_i are real scalars; $x_0 = 1$; n is the number of inputs and regression coefficients; and ϵ denotes the error term. Without loss of generality, this model can be rewritten in matrix form as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \epsilon \tag{2}$$

where the input data matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, the vector of regression coefficients $\boldsymbol{\beta} \in \mathbb{R}^{n \times 1}$, $\mathbf{y} \in \mathbb{R}^{m \times 1}$, and m is the number of observations. The elements of $\mathbf{X}$ can be any mix of raw input data and transformations of the raw data (aka features). The errors ϵ are assumed to be homoscedastic, have zero mean, and are uncorrelated.

Model building involves determining the vector $\boldsymbol{\beta}$ from the data $\mathbf{X}$ and $\mathbf{y}$ that minimizes the error ϵ concerning a defined measure of the error. The goal of multivariate regression is to find a good solution for $\boldsymbol{\beta}$ when there is a relatively low number of observations relative to the number of regression coefficients. For some specific applications, such as building a model to relate spectra to concentrations, (2) may not even have a unique solution.

Some special cases of (2) of interest are:

(a) The case where the input data matrix $\mathbf{X}$ has more rows than columns ($m > n$) and has full column rank, i.e., rank($\mathbf{X}$) = n. This case arises when many samples are available for building a model for predicting the output $\mathbf{y}$ from a small number of inputs (or calculated features). It is thus very unlikely that any vector $\mathbf{y} \in \mathbb{R}^{m \times 1}$ is in the column space of $\mathbf{X}$, due to the fact that the columns of $\mathbf{X}$ span only a small subspace of $\mathbb{R}^m$.

(b) The case where the input data matrix $\mathbf{X}$ has fewer rows than columns ($m < n$) and $\mathbf{X}$ has full row rank, i.e., rank($\mathbf{X}$) = m. This case can arise, for example, when spectral data are available for a large number of frequencies for building a model for predicting concentration from a relatively small number of samples of known concentration. In this case, the vector $\mathbf{y}$ lies in the column space of $\mathbf{X}$, and there exists no unique solution to (2). This case can be interpreted from the perspective of the right nullspace of $\mathbf{X}$, $N(\mathbf{X})$, which is the set of nonzero vectors $\mathbf{v}$ that satisfy $\mathbf{X}\mathbf{v} = \mathbf{0}$. In this case, this nullspace is not empty, that is, the nullspace consists of more vectors than the zero vector ($\mathbf{0}$). The vectors can be used to define a basis for the nullspace, and subsequently used to parameterize the infinite set of vectors of regression coefficients $\boldsymbol{\beta}$ that satisfy $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}$.

(c) The case where $m = n$ and rank($\mathbf{X}$) = $m = n$, for which (2) has a unique solution for $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}$. This case does not occur when working with larger datasets – due to biophysicochemical constraints such as associated with conservation equations and reaction networks – and is not considered further in this article.

Matrices with redundant rows and/or columns can be reduced to cases (a) or (b) by removing redundant rows and/or columns.

Different approaches find good solutions for cases (a) and (b) by solving

$$\mathbf{X}_{\text{train}} \hat{\boldsymbol{\beta}} = \hat{\mathbf{y}} \tag{3}$$

where $\hat{\boldsymbol{\beta}}$ is the vector of parameters estimated from the data $\mathbf{X}_{\text{train}}$ leading to the response $\hat{\mathbf{y}}$. The latter is related to the measured model outputs by additive noise ϵ.

$$\hat{\mathbf{y}} + \epsilon = \mathbf{y}. \tag{4}$$

The intent of the model is to be capable of making predictions of unseen data by taking a weighted combination of columns based on the estimated weights $\hat{\boldsymbol{\beta}}$. A fundamental question in statistics is how to find a good (or best) vector of model parameters $\hat{\boldsymbol{\beta}}$.[1]

This section describes different approaches that have been developed for finding $\hat{\boldsymbol{\beta}}$. To simplify the notation, $\boldsymbol{\beta}$ and $\mathbf{X}$ are used instead of $\hat{\boldsymbol{\beta}}$ and $\mathbf{X}_{\text{train}}$; in all cases, $\boldsymbol{\beta}$ refers to parameters estimated from training data and $\mathbf{X}$ refers to the training data. The first part of this section is a compact introduction to Ordinary Least Squares (OLS), Least Absolute Shrinkage and Selection Operator (lasso), Ridge Regression (RR), and elastic net (EN) regression. The second part is an introduction to Principal Component Analysis (PCA), Principal Component Regression (PCR), and Partial Least Squares (PLS), which are methods for dimensionality reduction and regression is performed subsequently in the lower dimensional space.

2.1. Ordinary least squares regression

The key idea of OLS regression is to find a solution $\boldsymbol{\beta}$ that minimizes the L_2-norm of the model error ϵ (Strang, 2016),

$$\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2. \tag{5}$$

Practically, this optimization can be solved by multiplying (3) on the left by $\mathbf{X}^\top$,

$$\mathbf{X}^\top \mathbf{X} \boldsymbol{\beta} = \mathbf{X}^\top \mathbf{y}, \tag{6}$$

which is known as the normal equation. For Case a, $\mathbf{X}^\top X$ is invertible, which leads to the analytical solution

$$\boldsymbol{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}. \tag{7}$$

This optimization formulation and solution are known as OLS. The Gauss–Markov theorem states that the OLS solution is optimal with the assumed error structure (4).

In Case b, $\mathbf{X}^\top \mathbf{X}$ is singular. A solution for $\boldsymbol{\beta}$ in this case is

$$\boldsymbol{\beta} = \mathbf{X}^\top (\mathbf{X}\mathbf{X}^\top)^{-1} \mathbf{y} \tag{8}$$

which is known as the minimum norm solution (Monticelli, 1999).

2.2. Lasso

OLS estimates may have low bias but have very large variance for many real-world data analytics problems, resulting in low prediction accuracy on unseen data. Lasso is a strategy for addressing this problem by adding the L_1-norm of the weights as a penalty to the least-squares objective (Tibshirani, 1996),

$$\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1. \tag{9}$$

---

[1] Multiple terms are used in the literature to refer to $\hat{\boldsymbol{\beta}}$, including *model parameters* and *regression coefficients*. From (3), it is clear that $\hat{\boldsymbol{\beta}}$ can be interpreted as being a vector of weights, that is, the model output is a weighted combination of the model inputs, with the weights being the elements of $\hat{\boldsymbol{\beta}}$.

For positive values for λ, lasso builds a model in which some weights are zero, $\beta_i = 0$, due to the structure of the optimization. The larger the value of λ, the more weights are forced to zero.

While there exists no closed-form solution for (9), common solvers are available. The L_1-norm penalty on β in (9) can be rewritten as linear constraints and the resulting optimization can be written as a convex quadratic program which can be solved efficiently using convex programming. A visual explanation for how the shape of the constraints and shape of the objective function in the optimization are responsible for the sparse solution can be found in Tibshirani (1996).

Models that use only a subset of available columns of the data matrix $\mathbf{X}$ are called *sparse models*. lasso selects a subset of columns to be used in the regression and can potentially lead to increased interpretability of results by removing measurements that are not needed in the model prediction.

A drawback of lasso is that, when some columns in $\mathbf{X}$ have a high pairwise correlation, lasso tends to select only one column and the choice of column can be sensitive to arbitrarily small perturbation in the data (Zou and Hastie, 2005). This randomness reduces the suitability of such models for interpreting the system in terms of which model inputs appear in the model.

2.3. Ridge regression

The underlying motivation for Ridge Regression (RR) is identical to lasso. The difference between the methods is that Ridge Regression (RR) adds a L_2-norm penalty to the objective function,

$$\min_{\beta} \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda\|\beta\|_2^2. \tag{10}$$

Setting the derivative of the objective function of (10) to zero leads to the closed-form solution (Hoerl and Kennard, 1970)

$$\beta = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}. \tag{11}$$

As in lasso, the L_2-norm penalty on β in (10) can be rewritten as a constraint. Due to the shape of the constraint in the resulting optimization, the weights β_i in RR will never reach zero although they can be arbitrarily small (James et al., 2021). Similarly to lasso, RR improves the predictive accuracy of the model by introducing a bias that reduces variance in the estimated parameters (Zou and Hastie, 2005). However, models produced by RR can be challenging to interpret, since all model inputs are retained in the model even for high values of λ.

2.4. Elastic Net

The Elastic Net (EN) is a combination of lasso and RR,

$$\min_{\beta} \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda P(\beta) \tag{12}$$

$$P(\beta) = \frac{1-\alpha}{2}\|\beta\|_2^2 + \alpha\|\beta\|_1 \tag{13}$$

for $\alpha \in (0,1)$ and non-negative values of λ. The L_1-norm provides the ability to set unimportant parameters to zero whereas the L_2-norm improves robustness in the selection of which parameters to retain in the model. Lasso and RR are limiting cases of the EN, for α equal to 1 and 0, respectively.

2.5. PCA and PCR

Principal Component Analysis (PCA) is an approach to manipulating the data matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$. The idea of PCA is to find a lower dimensional representation of $\mathbf{X}$ while conserving the variations in the data. From an optimization perspective, PCA solves

$$\max_{\|\mathbf{w}\|_2=1} \|\mathbf{X}\mathbf{w}\|_2^2 \tag{14}$$

for each principal component, with subsequent principal components also required to be orthogonal to the previous principal components.

PCA finds a weighted combination of the mean subtracted columns of $\mathbf{X}$ retaining maximal variance. For the first principal component, the constrained optimization (14) is equivalent to the unconstrained optimization

$$\mathbf{w_1} = \arg\max_{\mathbf{w}} \frac{\mathbf{w}^\top\mathbf{X}^\top\mathbf{X}\mathbf{w}}{\mathbf{w}^\top\mathbf{w}}. \tag{15}$$

The optimization (15) is solved by $\mathbf{w}$ being the eigenvector corresponding to the largest eigenvalue of the positive semidefinite matrix $\mathbf{X}^\top\mathbf{X}$.

More broadly, performing PCA involves the steps:

1. Center the columns of $\mathbf{X}$ to have zero mean.
2. Find the singular values and corresponding singular vectors. The singular values and right singular vectors of $\mathbf{X}$ correspond to the eigenvalues and eigenvectors of the covariance matrix $\mathbf{S} = \frac{1}{m-1}\mathbf{X}^\top\mathbf{X}$.
3. Project $\mathbf{X}$ onto the hyperplane spanned by the ℓ largest eigenvectors of the covariance matrix $\mathbf{S}$. The symmetry of the covariance matrix $\mathbf{S}$ implies that all of its eigenvectors are orthogonal.

The complete projection step of PCA can be written as

$$\mathbf{T} = \mathbf{X}\mathbf{W} \tag{16}$$

where $\mathbf{T} \in \mathbb{R}^{m\times n}$ is the PCA score matrix, $\mathbf{X} \in \mathbb{R}^{m\times n}$ is the data matrix, and $\mathbf{W} \in \mathbb{R}^{n\times n}$ is a coefficient matrix. The columns of $\mathbf{W}$ are called loadings and correspond to the eigenvectors of $\mathbf{X}^\top\mathbf{X}$ in descending order.

Eq. (16) describes a linear transformation. The loadings form an orthogonal basis, which results in the columns of $\mathbf{T}$ being decorrelated (Bengio et al., 2013). The first $\ell < n$ components of $\mathbf{T}$ form the ℓ-dimensional representation of $\mathbf{X}$ preserving most variance and are denoted by $\mathbf{T}_\ell$.

Applied data analytics problems nearly always have $\ell < m$, in which case OLS can be used for regressing the lower dimensional representation $\mathbf{T}$ versus $\mathbf{y}$ in an additional step. The combination of these two methods is known as Principal Component Regression (PCR). Combining (7) and (16) leads to

$$\beta_\ell = (\mathbf{T}_\ell{}^\top\mathbf{T}_\ell)^{-1}\mathbf{T}_\ell{}^\top\mathbf{y} \tag{17}$$

where ℓ is the number of principal components. The regression coefficients γ_ℓ correspond to the lower dimensional space based on ℓ principal components. Consequently, the regression coefficients in the original space are constructed by

$$\beta = \mathbf{W}_\ell\beta_\ell \tag{18}$$

where $\mathbf{W}_\ell$ denotes the first ℓ columns of the matrix $\mathbf{W}$.

2.6. PLS

Partial Least Squares (PLS) aims to find lower dimensional representations of $\mathbf{X}$ and $\mathbf{Y}$ and is not restricted to scalar objectives. While PCA performs dimensionality reduction in an unsupervised way, PLS incorporates information about the target $\mathbf{Y}$ in the dimensionality reduction scheme. In general, the governing equations for PLS are

$$\mathbf{X} = \mathbf{T}\mathbf{P}^\top + \mathbf{E} \tag{19}$$

$$\mathbf{Y} = \mathbf{U}\mathbf{Q}^\top + \mathbf{F} \tag{20}$$

where the matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$ is the data matrix, and $\mathbf{Y} \in \mathbb{R}^{m\times p}$ is the matrix of responses, $\mathbf{T} \in \mathbb{R}^{m\times l}$ is the score matrix, $\mathbf{P} \in \mathbb{R}^{n\times l}$ is the loading matrix, and $\mathbf{E} \in \mathbb{R}^{m\times n}$ is the residual matrix corresponding to $\mathbf{X}$. Similarly for $\mathbf{Y}$, the matrix $\mathbf{U} \in \mathbb{R}^{m\times l}$ is its score matrix, $\mathbf{Q} \in \mathbb{R}^{p\times l}$ is its loading matrix, and $\mathbf{F} \in \mathbb{R}^{m\times p}$ is its residual matrix.

In data analytics applications, the case $p = 1$ occurs frequently and is known as PLS1. The next paragraph introduces PLS1 with $\mathbf{y}$ being an single column vector. Readers interested in more details on other
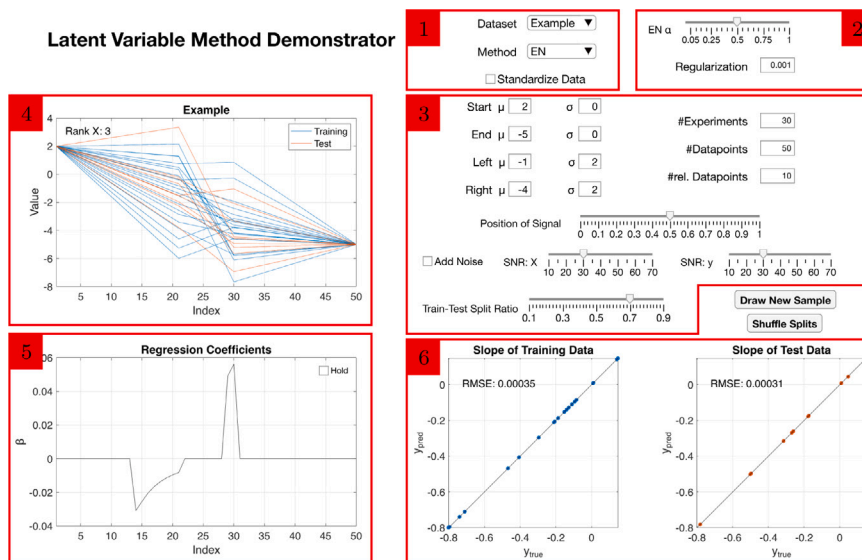
Fig. 1. Layout of Latent Variable Demonstrator (LAVADE).

variants of PLS are directed to other Refs. Chiang et al. (2000), Wold et al. (2001), MacGregor et al. (2005) and Boulesteix and Strimmer (2006).

Similarly to PCA, PLS performs a linear transformation on the input $\mathbf{X}$, with

$$\mathbf{T} = \mathbf{XW}. \tag{21}$$

From an optimization perspective, PLS maximizes the sample covariance between the $\mathbf{X}$ scores and the responses (Wold et al., 2001; Boulesteix and Strimmer, 2006). The calculation of the first component can be written in the form of the unconstrained optimization

$$\mathbf{w_1} = \arg\max_{\mathbf{W}} \frac{\mathbf{w}^{\top}\mathbf{X}^{\top}\mathbf{y}\mathbf{y}^{\top}\mathbf{Xw}}{\mathbf{w}^{\top}\mathbf{w}}. \tag{22}$$

The regression part of PLS is identical to (17), but using $\mathbf{T}_{\ell}$ estimated via the PLS algorithm.

3. Demonstrator design

The Latent Variable Demonstrator (LAVADE) software is designed to evaluate and compare the regression techniques described in Section 2 on datasets of different complexity. The Graphical User Interface (GUI) of the software is a single window, with drop-down menus, sliders, buttons, and axis objects (Fig. 1). Depending on the selection of the dataset and the regression technique, the relevant elements on the window become visible or invisible, reducing user distraction.

The six elements in the GUI are

1. Dataset and Method selection drop-down menu, and check box, to select whether inputs shall be standardized[2]
2. Hyperparameters for the respective regression method
3. Modification of data: Data generation parameters ("Example case"), addition of white Gaussian noise
4. Visualization of the data, training, and test data, color-coded
5. Visualization of regression coefficients
6. Regression results on the training and test data, respectively

On the backend, the software consists of two MATLAB classes. The "lavade" class inherits from the "AppBase" class, a generic MATLAB app class and contains all the elements of the GUI and general variables

---

[2] Standardization refers to centering the columns and scaling to have unit variance.

associated with the functionality of the software. The "data" class is a superclass from which all the individual dataset classes inherit. The constructor of the individual dataset classes loads or generates the data. Furthermore, the data class contains methods to perturb the signal by noise, and to split the data. This structure helps to implement new datasets quickly.

4. Datasets

The LAVADE software includes an illustrative numerical example data generator and three real-world datasets. The datasets can also be downloaded separately from the corresponding GitHub repository.

Example. The motivation of the example dataset is to generate a dataset similar in character to process measurements, in which only a subset of the measurements have predictive power. The dataset is constructed by starting by defining four data points: start, final, and two intermediate data points (Fig. 2). The four data points are drawn from a Gaussian distribution with defined means and standard deviations. The output responses $\mathbf{y}$ are the (average) slopes in the relevant section.

Fig. 2 shows the default case in which the parameters of the four Gaussian distributions are $\mu_{start} = 2$, $\mu_{left} = -1$, $\mu_{right} = -4$, $\mu_{end} = -5$, and $\sigma_{start} = \sigma_{end} = 0$, $\sigma_{left} = \sigma_{right} = 2$. The data are within three sections. The slope in the relevant section can also be inferred from data outside of the relevant section due to $\sigma_{start} = \sigma_{end} = 0$.

This dataset illustrates the common situation in which the data are highly correlated while not having any datapoints repeated, while being simple enough to easily understand the dataset and the model being constructed, and be able to interpret the results.

Fig. 3 shows the result of standardizing the data in Fig. 2. For each data vector, all datapoints left of the relevant section have the same value and carry the same information, except for the first and the last datapoints. The same holds true for the datapoints right of the relevant section.

A noisy dataset adds white Gaussian noise with defined Signal-to-Noise Ratio (SNR) to the matrix $\mathbf{X}$ and vector of responses $\mathbf{y}$ (Fig. 4). By varying the Signal-to-Noise Ratio (SNR), the user can assess how noise affects the model training and model predictions. This example dataset encourages the user to develop a general understanding of model types, hyperparameters, noise, training, and prediction accuracy on a simple dataset inspired by common process data analytics problems.
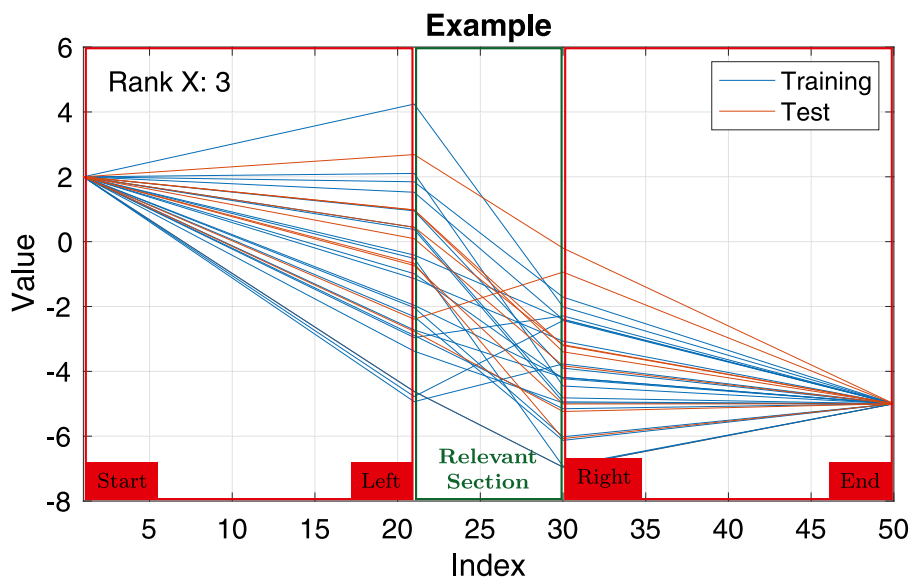
Fig. 2. Default case: $\sigma_{\text{start}} = \sigma_{\text{end}} = 0$, $\sigma_{\text{left}} = \sigma_{\text{right}} = 2$. "Relevant section" refers to the middle section among the three sections.
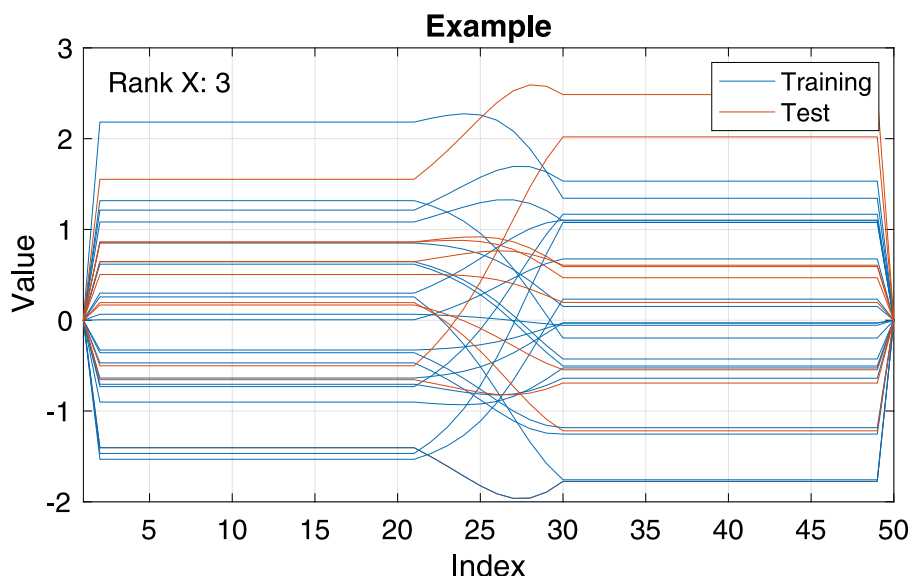

Fig. 3. Default standardized case: $\sigma_{\text{start}} = \sigma_{\text{end}} = 0$, $\sigma_{\text{left}} = \sigma_{\text{right}} = 2$, standardized columns.

ATR-FTIR spectra. This dataset consists of Attenuated Total Reflectance-Fourier Transform Infrared (ATR-FTIR) spectra that were recorded and analyzed in Fujiwara et al. (2002). The model objective is to predict the paracetamol concentration C associated with each spectrum. The dataset consists of spectra collected for six values of the concentration (six "groups"), with varying temperature. For simplicity, we omit the temperature readings and use only the ATR-FTIR spectra as model input.

Each spectrum is highly correlated with the other spectra, and the absorbances within each spectrum are highly correlated. Each group of spectra is evenly spaced from adjacent groups of spectra except for the group of spectra at $C = 0.014\,\text{g/g}$, which is the second group from the bottom in Fig. 5. Because the rest of the data appears to have roughly a linear relationship, it is expected that much of the $C = 0.014\,\text{g/g}$ data are biased, most likely by some experimental error. As such, the data at $C = 0.014\,\text{g/g}$ are placed in the test dataset as the default. This data assignment leads to more conservative test errors and prevents the model from learning an aphysical correlation due to this bias. The strategy of splitting of this dataset can be chosen in

the GUI to be either fully at random or such that groups of spectra roughly corresponding to the same concentration are always part of only the training or the test dataset. This latter respects the fact that, for spectral data, the variation of data between groups is typically larger than the variation of data within a group (Sun and Braatz, 2021). Given that error structure, placing the data from all groups into the training dataset, for example, would result in overfitting and overly confident/optimistic prediction errors for the test data (Escobar et al., 2014). Usually, data are standardized before applying ridge regression, lasso, and elastic net, to account for widely different ranges of values for each data type (e.g., a temperature could be more than 100 degrees for a system in which absorbance spans 0.01 absorbance units). However, this particular dataset consists only of spectra, and all readings have the same order of magnitude. In this case, skipping standardization can result in better models. Typically many absorbances are not associated with the molecule of interest and contain nearly all noise; standardization normalizes those absorbances to have the same magnitude of importance in the matrix $\mathbf{X}$ in the optimization objective. Thereby standardization feeds noise into the parameter estimation
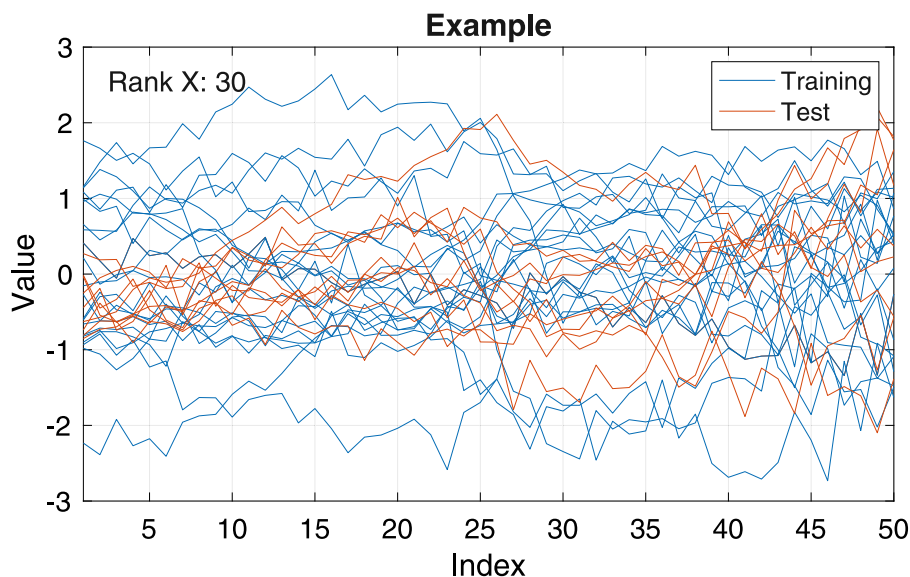
Fig. 4. Noisy case: $\sigma_{\text{start}} = \sigma_{\text{end}} = 2$, $\sigma_{\text{left}} = \sigma_{\text{right}} = 2$, SNR = 20 for all sections.
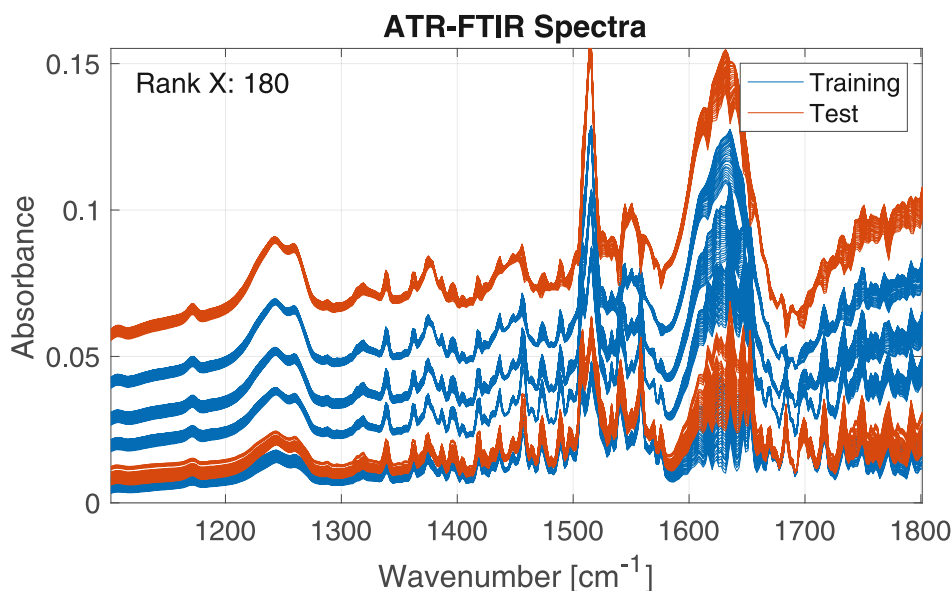

Fig. 5. Visualization of ATR-FTIR spectra for paracetamol in aqueous solution.

and degrades the quality of the model. Also, the potential benefits of standardization are lower for spectral data, as most absorbance values are already of a similar order of magnitude.

Raman spectra. Raman scattering refers to the inelastic scattering of photons by matter. Raman spectra have high molecular specificity and are used to build models to predict molecular concentrations in solutions (Das and Agrawal, 2011; Pandey et al., 2017). A promising use case for this type of model is noninvasive glucose monitoring (Mehdizadeh et al., 2015), which uses Raman spectra from different bioreactor runs to learn mappings to glucose concentrations from Nova FlexII cell culture measurements. This task is challenging due to noise and background signals in the Raman spectra and noise in the Nova FlexII readings. Furthermore, the biochemical reactions taking place in the bioreactor lead to correlations in the concentration changes of various metabolites. We removed outliers from the cell culture measurements of Bozinovski et al. (2022) by applying a moving median filter having a window size of three days and removing all values that

were above three standard deviations. The Raman spectra are available in the software (1) without background removal, (2) background removal by the method of Lieber and Mahadevan-Jansen (2003) using the Biodata MATLAB Toolbox implementation (De Gussem et al., 2009), and (3) background removal with the MATLAB algorithm `msbackadj`. The range of Raman shifts was restricted to the range 400–1800 cm^{-1} as suggested by Whelan et al. (2012) and Rowland-Jones and Jaques (2019).

Fig. 6 shows 327 Raman spectra, each corresponding to a cell culture measurement. The cell culture measurements were collected at different times than the Raman spectra. The relatively slow concentration changes enabled the concentrations corresponding to the Raman spectra to be linearly interpolated from the cell culture measurements.

Lithium-iron-phosphate batteries. The Lithium-Iron-Phosphate (LFP) battery dataset contains cycling data for 124 batteries that have underwent widely varying charging protocols and discharged at a uniform current (Severson et al., 2019). The model objective is the cycle life,
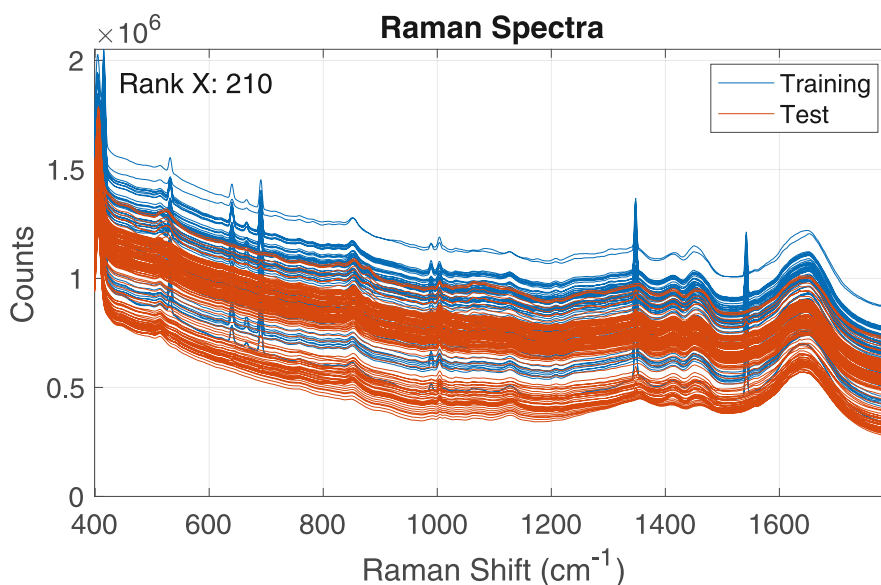
Fig. 6. Visualization of Raman spectra without background removal from a bioreactor.
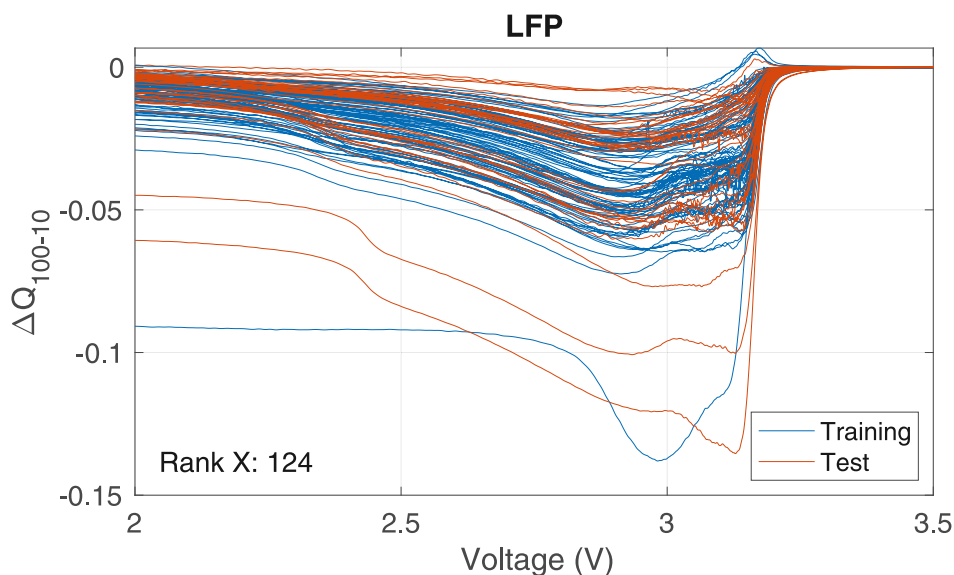

Fig. 7. Visualization of the LFP dataset in the LAVADE software.

i.e., the number of cycles until the battery's capacity drops below 80% of its nominal capacity.

Severson et al. (2019) showed that features based on the difference between the discharge capacity of voltage curves for two cycles, subsequently called ΔQ_{a-b}, linearly correlate well with the logarithm of the cycle life. The cycle pair $a = 100$ and $b = 10$ was used in Severson et al. (2019). While other cycle combinations also work well, the implemented input data $\mathbf{X}$ is ΔQ_{100-10} for this demonstrator (see Fig. 7). More information about the dataset and reasoning about the modeling objective can be found in Severson et al. (2019).

Some charging protocols were applied to only on a single cell, whereas others were used on up to nine different cells, leading to a grouping structure. In the LAVADE software, the train-test split can be chose to be fully at random or randomly assigning groups such that cells with the same charging protocol are only part of either the training or the test dataset. The cycle life distribution of the dataset is non-Gaussian, with a heavy concentration of cells having a relatively low cycle life and only a few having longer cycle life. Depending on the

split, models will tend to struggle to extrapolate to the longer-lived cells, leading to high variability in prediction errors calculated for the test set.

Furthermore, the batteries were cycled in three batches with different starting dates. Thus the cells were stored over different durations between manufacturing and cycling. The first batch was cycled approximately a year earlier than the last batch leading to prolonged calendar aging for some cells. This additional information is crucial when building robust data analytics models but is not elaborated on further in this context.

5. Illustrative examples

This section illustrates the LAVADE software by applying the various regression methods to the datasets. While small differences can arise when reproducing the results, due to the randomized splitting of the datasets, the observations made in this section are general. The regularization parameters in the software must be chosen by hand. For
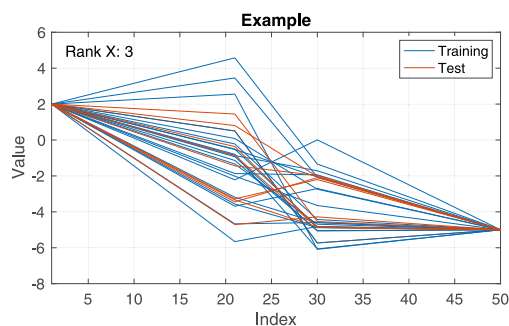
(a) Noise-free example dataset
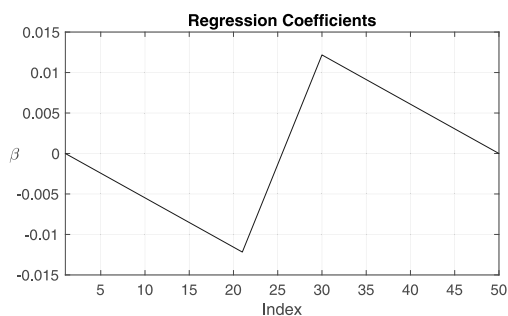
(b) PLS and RR regression coefficients
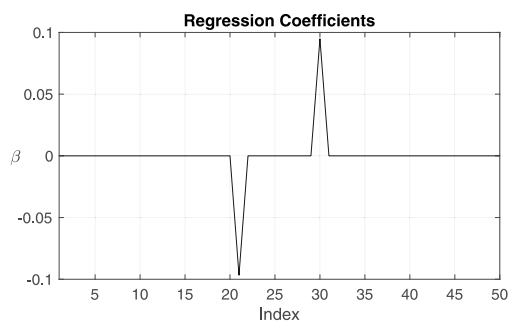
(c) lasso regression coefficients

Fig. 8. Noise-free example dataset and regression coefficients for PLS, RR, and lasso, with hyperparameters and RMSEs reported in Table 1.

Table 1
Regression methods, hyperparameters, regression coefficients, and RMSE for the training and test datasets for the noise-free example.

Regression method	Regression coefficients	RMSE training	RMSE test
PLS (2 components)	Fig. 8b	1.4×10^{-16}	7.3×10^{-17}
RR ($\lambda = 1 \times 10^{-6}$)	Fig. 8b	4.6×10^{-10}	4.6×10^{-10}
Lasso ($\lambda = 0.015$)	Fig. 8c	0.011	0.011

Table 2
PLS regression coefficients, numbers of components, and RMSE for the training and test ATR-FTIR spectral datasets.

Regression coefficients	Number of components	RMSE train	RMSE test
Fig. 9a	7	1.4×10^{-5}	3.5×10^{-5}
Fig. 9b	7	1.4×10^{-5}	8×10^{-4}
Fig. 9c	8	1.5×10^{-5}	1.7×10^{-5}

the following examples, we chose the hyperparameters to showcase interesting regression coefficients and prediction results.

5.1. Example dataset

The regression results for the noise-free example dataset illustrate several important properties of the implemented methods. PLS and RR result in nearly identical regression coefficients in this ideal dataset (Fig. 8b). Due to the mathematical similarities between PCR, PLS, and RR (de Jong and Farebrother, 1994; Sundberg, 1993; Stone and Brooks, 1990) regression coefficients are expected to have certain similarities given appropriate parameters. Lasso leads to sparse regression coefficients that are intuitive; an observant human would also recognize that all of the data points in Fig. 8a can be determined from the difference in values between the 30th and 21st measurements.[3] Table 1 shows that the three methods investigated here lead to regression coefficients that result in accurate model predictions, with PLS and lasso being the most and least accurate, respectively, given the chosen regularization parameters. The regression coefficients, however, have very different shapes (cf. Fig. 8bc). Because the methods result in very similar predictions, the differences between the regression coefficients must be within or very close to the nullspace of the data, $N(\mathbf{X})$.

---
[3] Or, equivalently, by the slope of the line in the middle region computed from the maximum and minimum values.

5.2. FTIR spectral data

The results of PLS regression for the ATR-FTIR spectral dataset shown in Fig. 9 and Table 2 highlight implications of different ways of splitting data into training and test datasets in the presence of groups. In Fig. 9ab, groups of spectra belonging to the same concentration are randomly assigned to either the training or test dataset. In all cases, the root-mean-squared errors for the test data are higher than for the training data, due to the biased spectra corresponding to $C = 0.014\,\text{g/g}$ that was chosen to be in the test dataset for all three splits. Fig. 9b shows a split where the test dataset contains groups with concentrations outside the training data. The root-mean-squared errors of prediction of the training and test datasets differ by one order of magnitude and show clearly that the model fails to extrapolate to the concentrations in the test dataset. It is recommended to assign the groups with spectra corresponding to the lowest and highest concentration as part of the training dataset, as is the case in Fig. 9a, so that the model interpolates rather than extrapolates.

Fig. 9c shows prediction results with randomized splitting of the training and test datasets. This violation of the grouping structure when splitting the dataset leads to *what appears to be* more accurate predictions for the test dataset (Table 2). It is important to take into account that the prediction error for the test dataset obtained in case c approximates the expected generalization error only for new data *that are within the groups that are already part of the training data*. The reason for this restriction is that the between-group variance is higher

(a) Interpolation data split (b) Extrapolation data split (c) Data split ignoring grouping
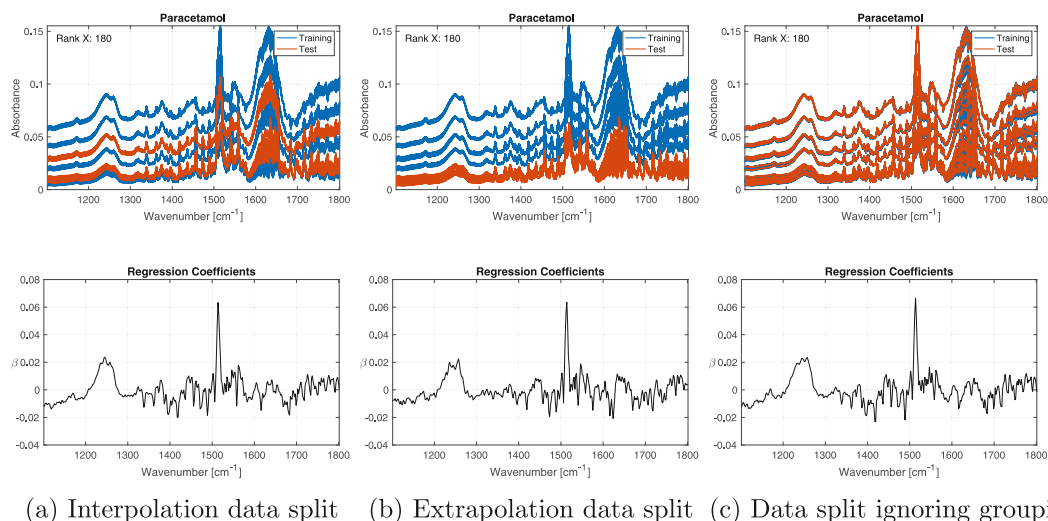
Fig. 9. Three ways of splitting the ATR-FTIR spectral dataset (top) and the corresponding regression coefficients obtained by PLS (bottom), with number of components and RMSEs reported in Table 2.

than the within-group variance. Such an estimated prediction error will under-predict the prediction error when the model is applied to data that are not within the groups in the original dataset. In practice, the model is intended for use at *any* paracetamol concentration within the range of concentration that was in the training data, and the groupings constraint must be respected to obtain a model accuracy from the test set that is a reasonably accurate approximation of the generalization error.

5.3. Raman spectral data

Fig. 10 shows the results for a PLS model trained on data from bioreactor run 1, with the test set being data from bioreactor run 2 (Fig. 10d). The regression coefficients show that the Raman shifts in the region around 1000–1200 cm^{-1} are most important for predicting glucose concentration. The model predicts the glucose concentration in the test dataset fairly well, however, the model accuracy is significantly higher on the training dataset than on the test dataset.

Fig. 10 shows the results for a PLS model trained on data from bioreactor run 2, with the test set being data from bioreactor run 1 (Fig. 11d). Most of the glucose concentrations predicted by the model are biased and the prediction errors are much larger than estimated by the model that was trained on bioreactor run 1 and predicted on bioreactor run 2 (cf. Fig. 10d). However, the regression coefficients determined from bioreactor run 2 are very similar to the regression coefficients determined from bioreactor run 1. Ensuring a strict split of the data, in which data from each bioreactor are only in the training or test datasets, is advised to obtain a more realistic generalization error of the model. As seen in the ATR-FTIR spectral dataset example, grouping should be respected when building such models, due to the between-group variation being higher than the within-group variation. The large test errors in Fig. 11 illustrate the challenges that occur when trying to build models that will predict glucose concentration well for new bioreactor runs. The intrinsically very low photon efficiency of Raman spectroscopy and batch-to-batch and time-varying concentrations of other species makes it challenging to build highly accurate predictive models of glucose in bioreactors.

Fig. 12 shows the results of building a PLS model for Raman spectra from multiple bioreactor runs. The data are split randomly, leading to data from all of the bioreactors being present in both the training and test datasets. The model accurately predicts the glucose concentration in the test dataset. The number of PLS components is significantly higher than in the cases of Figs. 10 and 11, but overall the regression coefficients are similar. However the regression coefficients in Fig. 12

have a higher level of noise, suggesting that the model overfits to specific conditions being present in the respective bioreactors, i.e., that too many components were used. The split leads to *what appears to be* more accurate predictions for the test dataset similar to the ATR-FTIR spectral dataset. It is important to take into account that the prediction error for the test dataset obtained approximates the expected generalization error only for new data *that are within the reactor runs that are already part of the training data*. In addition, the random split also does not respect the time series structure of the dataset and is thus not a fair estimate of the model accuracy for reactor conditions further in the future (i.e., not in between conditions present in the training dataset as it is the case here for most data in the test dataset).

5.4. LFP battery dataset

The distribution of cycle life of batteries in the LFP dataset is non-Gaussian, with a high amount of cells with medium cycle life, three cells with a significantly shorter cycle life, and a few cells with significantly longer cycle life. This distribution results in the root-mean-squared-errors varying depending on precisely which data are in the training and test datasets, even with well-tuned hyperparamaters. For the split in Fig. 13, the RMSE for the test data is lower than for the training data, which is because the test dataset contains only medium- to short-lived cells that are well captured by the model. The RMSE for the training dataset for medium- to short-lived cells in Fig. 13 is similar to the RMSE for the test dataset. The RMSE for the training data is strongly influenced by the higher RMSE of the longest lived cells. This example illustrates the significant effect that a small number of data points at extreme values of the data distribution, although reasonably consistent with the rest of the data, can have on the estimated prediction errors.

Fig. 14 shows the regression coefficients for models that were trained on ten training-test splits of the dataset. The resulting coefficients vary significantly, particularly in the region around 3 V marked by a red box. This high variability suggests that PLS finds it challenging to extract the information contained in this voltage region. Reasons for this behavior could be nonlinear effects or inherent biases due to battery physics that manifest in this voltage region. For more insights, we refer to the supplementary information of Attia et al. (2020) that contains more details about the measuring campaign and corresponding differences in calendar aging due to different storage times of batteries before the start of the measurements.
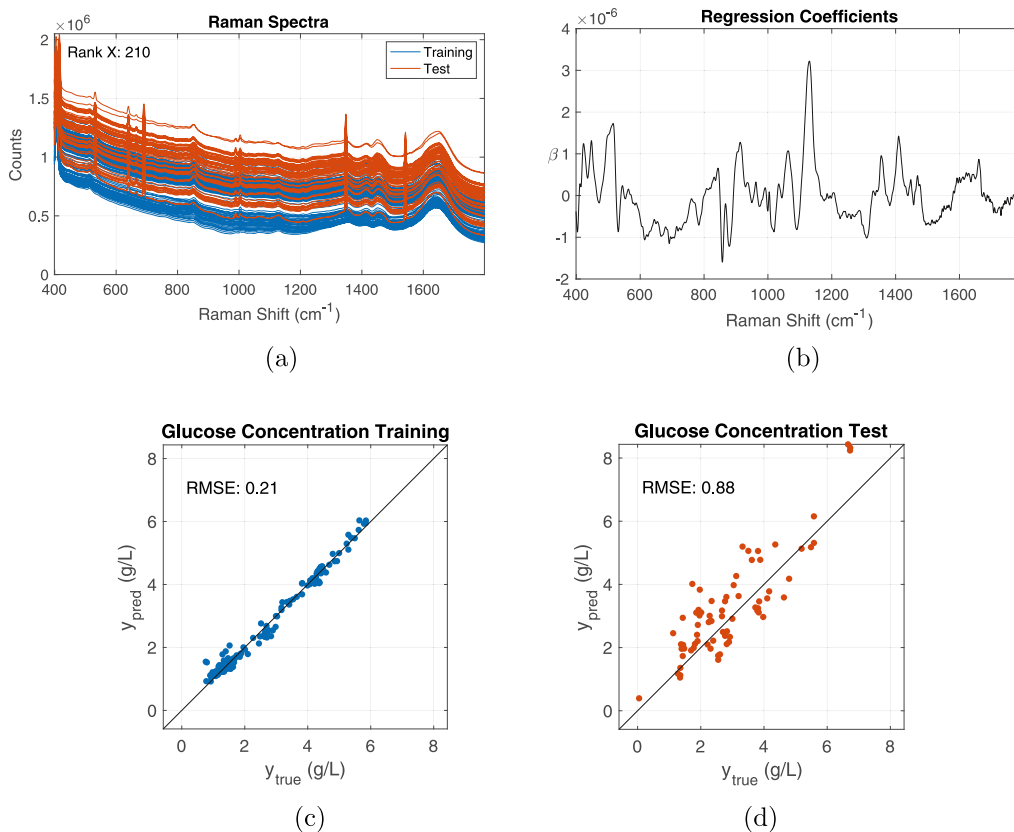
Fig. 10. (a) Raman spectral dataset without background subtraction with the training data (bioreactor run 1) in blue and the test data (bioreactor run 2) in red, (b) the regression coefficients for a PLS model with 7 components, (c,d) the regression results for the training and test data.
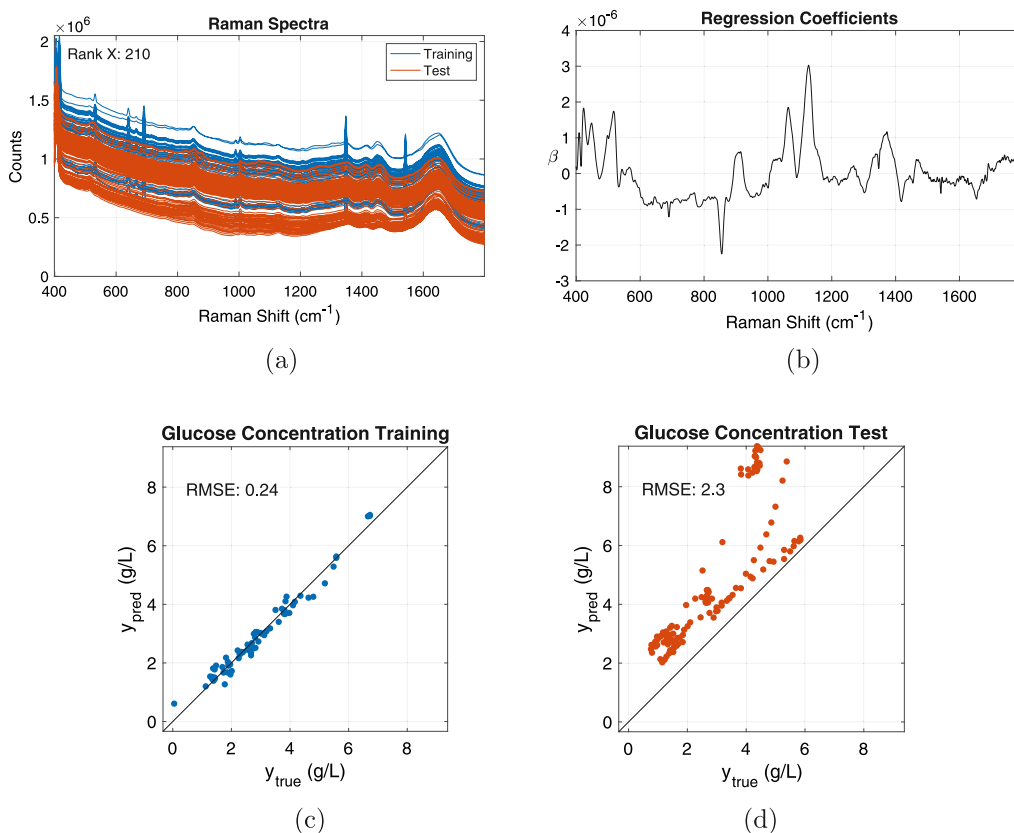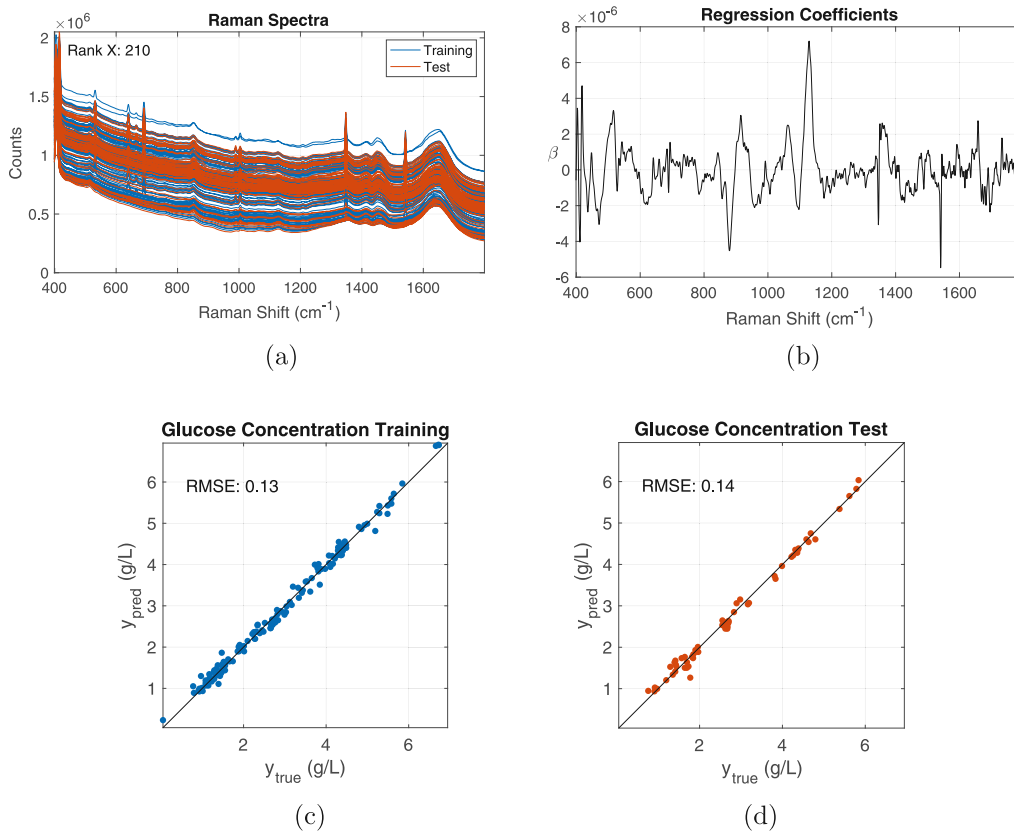

Fig. 11. (a) Raman spectral dataset without background subtraction with the training data (bioreactor run 2) in blue and the test data (bioreactor run 1) in red, (b) the regression coefficients for a PLS model with 6 components, (c,d) the regression results for the training and test data.

Fig. 12. (a) Raman spectral dataset without background subtraction with the training data in blue and the test data in red using a random split, (b) the regression coefficients for a PLS model with 12 components, (c,d) the regression results for the training and test data.
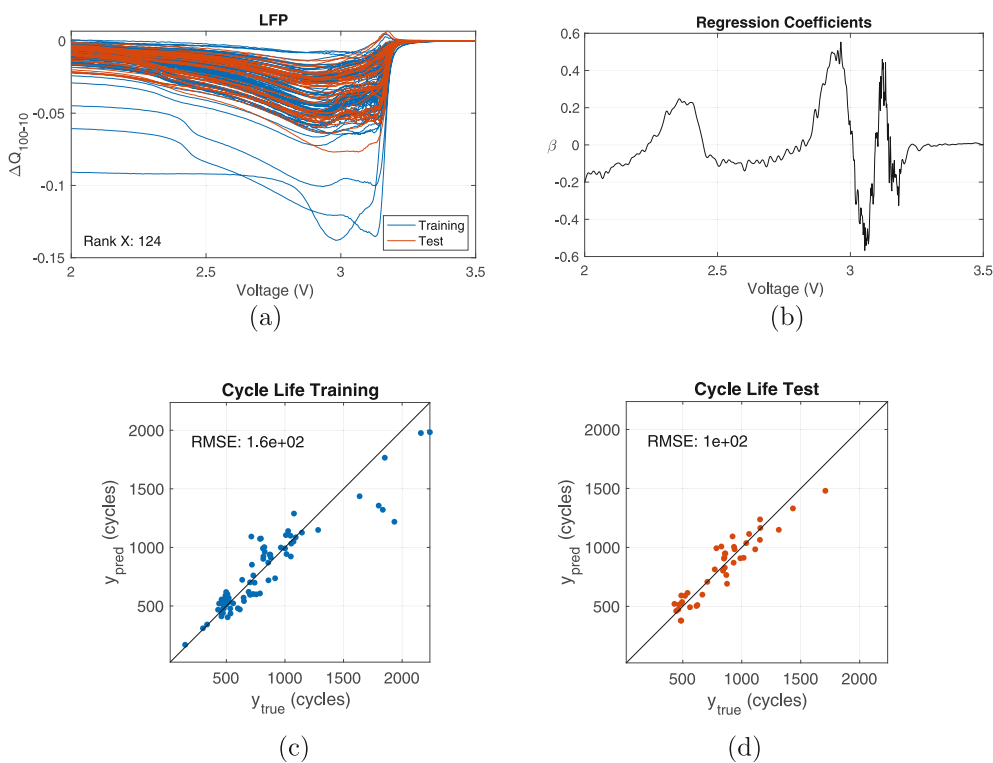

Fig. 13. (a) LFP dataset with the training data in blue and the test data in red, (b) the regression coefficients for a PLS model with four components, (c,d) the regression results for the training and test data.
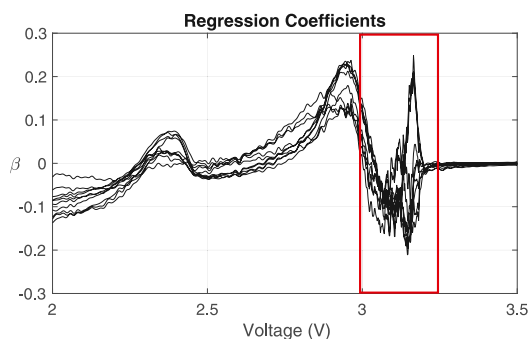
Fig. 14. Varying regression coefficients for different train-test splits.

6. Conclusion

The LAVADE software enables users to easily compare different latent variable regression and alternative regression methods when applied to four datasets of different complexity ranging from a synthetic dataset to real-world datasets. The visualizations of the data, train-test splits, regression coefficients, and predictions are useful to understanding in these regression methods. These visualizations reveal important details about the data and implications of the model building strategy. The interactive environment of the GUI fosters a deeper understanding of how models interact with data. Comparing regularization parameters of lasso, RR, EN, and the number of components for PLS and PCR helps to understand how the different approaches of regularization affect the resulting regression coefficients.

The open-source LAVADE software intends to be a starting point to gain intuition on latent variable and other multivariate regression methods. Ultimately we hope to inspire teachers to use this tool to advance their teaching methods and encourage extensions of the software and examples.

CRediT authorship contribution statement

Joachim Schaeffer: Conceptualization, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization, Funding acquisition. **Richard D. Braatz:** Conceptualization, Investigation, Resources, Writing – original draft, Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Code and data availability

The code is available at https://github.com/JoachimSchaeffer/LAVADE. The repository contains the datasets that are described in this article. The paracetamol dataset was recorded and an analysis of the dataset was published with Fujiwara et al. (2002). The repository contains only a small subset of the LFP dataset that was published with Severson et al. (2019). The entire LFP dataset is available at https://data.matr.io/1/.

References

Attia, P.M., Grover, A., Jin, N., Severson, K.A., Markov, T.M., Liao, Y.-H., Chen, M.H., Cheong, B., Perkins, N., Yang, Z., Herring, P.K., Aykol, M., Harris, S.J., Braatz, R.D., Ermon, S., Chueh, W.C., 2020. Closed-loop optimization of fast-charging protocols for batteries with machine learning. Nature 578 (7795), 397–402. http://dx.doi.org/10.1038/s41586-020-1994-5.

Bengio, Y., Courville, A., Vincent, P., 2013. Representation learning: A review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. 35 (8), 1798–1828. http://dx.doi.org/10.1109/TPAMI.2013.50.

Boulesteix, A.-L., Strimmer, K., 2006. Partial least squares: A versatile tool for the analysis of high-dimensional genomic data. Brief. Bioinform. (ISSN: 1467-5463) 8 (1), 32–44. http://dx.doi.org/10.1093/bib/bbl016.

Bozinovski, D., Cummings Bende, E., Maloney, A., Sangerman, J., Dubs, A., Lu, A., Hong, M.S., Persits, N., Artamonova, A., Ou, R.W., Sun, W., Wolfrum, J., Barone, P., Springs, S., Braatz, R., Sinskey, A., 2022. Biomanufacturing and testbed development for the continuous production of monoclonal antibodies. In: ACS Spring Meeting. San Diego, California, Abstract 3661909, associated journal manuscript in preparation.

Chiang, L.H., Russell, E.L., Braatz, R.D., 2000. Fault Detection and Diagnosis in Industrial Systems. Springer Verlag, London, UK, http://dx.doi.org/10.1007/978-1-4471-0347-9.

Das, R.S., Agrawal, Y., 2011. Raman spectroscopy: Recent advancements, techniques and applications. Vib. Spectrosc. 57 (2), 163–176. http://dx.doi.org/10.1016/j.vibspec.2011.08.003, URL: https://www.sciencedirect.com/science/article/pii/S0924203111001111.

De Gussem, K., De Gelder, J., Vandenabeele, P., Moens, L., 2009. The biodata toolbox for MATLAB. Chemometr. Intell. Lab. Syst. 95 (1), 49–52. http://dx.doi.org/10.1016/j.chemolab.2008.08.003, URL: https://www.sciencedirect.com/science/article/pii/S0169743908001561.

de Jong, S., Farebrother, R.W., 1994. Extending the relationship between ridge regression and continuum regression. Chemometr. Intell. Lab. Syst. 25 (2), 179–181. http://dx.doi.org/10.1016/0169-7439(94)85041-0.

Escobar, M.S., Kaneko, H., Funatsu, K., 2014. Flour concentration prediction using GAPLS and GAWLS focused on data sampling issues and applicability domain. Chemometr. Intell. Lab. Syst. 137, 33–46. http://dx.doi.org/10.1016/j.chemolab.2014.06.005.

Fujiwara, M., Chow, P.S., Ma, D.L., Braatz, R.D., 2002. Paracetamol crystallization using laser backscattering and ATR-FTIR spectroscopy: Metastability, agglomeration, and control. Cryst. Growth Des. 2 (5), 363–370. http://dx.doi.org/10.1021/cg0200098.

Hoerl, A.E., Kennard, R.W., 1970. Ridge regression: Biased estimation for nonorthogonal problems. Technometrics 12 (1), 55–67. http://dx.doi.org/10.1080/00401706.1970.10488634.

James, G., Witten, D., Hastie, T., Tibshirani, R., 2021. Linear model selection and regularization. An Introduction to Statistical Learning Book Subtitle with Applications in R. Springer, New York, pp. 225–288. http://dx.doi.org/10.1007/978-1-0716-1418-1.

Lieber, C.A., Mahadevan-Jansen, A., 2003. Automated method for subtraction of fluorescence from biological Raman spectra. Appl. Spectrosc. 57 (11), 1363–1367, URL: http://opg.optica.org/as/abstract.cfm?URI=as-57-11-1363.

MacGregor, J.F., Yu, H., García Muñoz, S., Flores-Cerrillo, J., 2005. Data-based latent variable methods for process analysis, monitoring and control. Comput. Chem. Eng. 29 (6), 1217–1223. http://dx.doi.org/10.1016/j.compchemeng.2005.02.007.

Mardia, K., 2003. Multivariate Analysis. Academic Press, London, UK.

Mehdizadeh, H., Lauri, D., Karry, K.M., Moshgbar, M., Procopio-Melino, R., Drapeau, D., 2015. Generic Raman-based calibration models enabling real-time monitoring of cell culture bioreactors. Biotechnol. Prog. 31 (4), 1004–1013. http://dx.doi.org/10.1002/btpr.2079, arXiv:https://aiche.onlinelibrary.wiley.com/doi/pdf/10.1002/btpr.2079 URL: https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/btpr.2079.

Monticelli, A., 1999. State Estimation in Electric Power Systems: A Generalized Approach. Springer, Boston, pp. 15–37.

Pandey, R., Paidi, S.K., Valdez, T.A., Zhang, C., Spegazzini, N., Dasari, R.R., Barman, I., 2017. Noninvasive monitoring of blood glucose with Raman spectroscopy. Acc. Chem. Res. 50 (2), 264–272. http://dx.doi.org/10.1021/acs.accounts.6b00472.

Rowland-Jones, R.C., Jaques, C., 2019. At-line raman spectroscopy and design of experiments for robust monitoring and control of miniature bioreactor cultures. Biotechnol. Prog. 35 (2), e2740. http://dx.doi.org/10.1002/btpr.2740, arXiv:https://aiche.onlinelibrary.wiley.com/doi/pdf/10.1002/btpr.2740 URL: https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/btpr.2740.

Severson, K.A., Attia, P.M., Jin, N., Perkins, N., Jiang, B., Yang, Z., Chen, M.H., Aykol, M., Herring, P.K., Fraggedakis, D., Bazant, M.Z., Harris, S.J., Chueh, W.C., Braatz, R.D., 2019. Data-driven prediction of battery cycle life before capacity degradation. Nat. Energy 4 (5), 383–391. http://dx.doi.org/10.1038/s41560-019-0356-8.

Stone, M., Brooks, R.J., 1990. Continuum regression: Cross-validated sequentially constructed prediction embracing ordinary least squares, partial least squares and principal components regression. J. R. Stat. Soc. Ser. B Stat. Methodol. 52 (2), 237–258. http://dx.doi.org/10.1111/j.2517-6161.1990.tb01786.x.

Strang, G., 2016. Introduction to Linear Algebra, fifth ed. Cambridge Press, Wellesley, Massachusetts.

Sun, W., Braatz, R.D., 2021. Smart process analytics for predictive modeling. Comput. Chem. Eng. 144, 107134. http://dx.doi.org/10.1016/j.compchemeng.2020.107134.

Sundberg, R., 1993. Continuum regression and ridge regression. J. R. Stat. Soc. Ser. B Stat. Methodol. 55 (3), 653–659. http://dx.doi.org/10.1111/j.2517-6161.1993.tb01930.x.

Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. J. R. Stat. Soc. Ser. B Stat. Methodol. 58 (1), 267–288. http://dx.doi.org/10.1111/j.2517-6161.1996.tb02080.x.

Whelan, J., Craven, S., Glennon, B., 2012. In situ Raman spectroscopy for simultaneous monitoring of multiple process parameters in mammalian cell culture bioreactors. Biotechnol. Prog. 28 (5), 1355–1362. http://dx.doi.org/10.1002/btpr.1590, arXiv:https://aiche.onlinelibrary.wiley.com/doi/pdf/10.1002/btpr.1590 URL: https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/btpr.1590.

Wold, S., Sjöström, M., Eriksson, L., 2001. PLS-regression: A basic tool of chemometrics. Chemometr. Intell. Lab. Syst. 58 (2), 109–130. http://dx.doi.org/10.1016/S0169-7439(01)00155-1.

Zou, H., Hastie, T., 2005. Regularization and variable selection via the elastic net. J. R. Stat. Soc. Ser. B Stat. Methodol. 67 (2), 301–320. http://dx.doi.org/10.1111/j.1467-9868.2005.00503.x.