Simultaneous variable selection for the classification of near infrared spectra

Leila Belmerhnia, El-Hadi Djermoune, Cédric Carteret, David Brie

▶ **To cite this version:**

Leila Belmerhnia, El-Hadi Djermoune, Cédric Carteret, David Brie. Simultaneous variable selection for the classification of near infrared spectra. Chemometrics and Intelligent Laboratory Systems, 2021, 211, pp.104268. 10.1016/j.chemolab.2021.104268 . hal-03152234v2

HAL Id: hal-03152234
https://hal.science/hal-03152234v2

Submitted on 26 Feb 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Simultaneous Variable Selection for the Classification of Near Infrared Spectra

Leila Belmerhnia[a,b], El-Hadi Djermoune[a,*], Cédric Carteret[c], David Brie[a]

[a] *Université de Lorraine, CNRS, CRAN, F-54000 Nancy, France*
[b] *TVT Innovation, Maison du numérique et de l'innovation, Place Georges Pompidou, F-83000, Toulon, France*
[c] *Université de Lorraine, CNRS, LCPME, F-54000 Nancy, France*

Abstract

This paper addresses the problem of wood wastes recycling automation. We propose variable selection methods based on near infrared spectroscopic data to select a set of wavebands that captures the main spectral peaks of wood materials to improve the sorting performances. The spectra are first jointly modeled as linear combinations of explanatory variables drawn from a collection of Gaussian-shaped functions. The aim is to select a common subset of wavebands shared by several spectra. The variable selection is then formulated as an unconstrained simultaneous sparse approximation problem in which the coefficients related to different spectra are encouraged to be piecewise constant, *i.e.* the coefficients associated to successive spectra should have comparable magnitudes. We also investigate the case where the coefficients are constrained to be nonnegative. These problems are solved using the fast iterative shrinkage-thresholding algorithm. The proposed approaches are illustrated on a dataset of 290 spectra of wood wastes; each spectrum is composed of 1647 wavelengths. We show that the selected variables lead to better classification performances as compared to standard approaches.

Keywords: Variable selection; Simultaneous sparse approximation; NIR

[☆] This work is supported by the French FUI AAP15 Trispirabois project funded by BPI France and Région Lorraine.

[*] Corresponding author

Email addresses: belmerhnia@tvt.fr (Leila Belmerhnia), el-hadi.djermoune@univ-lorraine.fr (El-Hadi Djermoune), cedric.carteret@univ-lorraine.fr (Cédric Carteret), david.brie@univ-lorraine.fr (David Brie)

Preprint submitted to Chemolab

spectroscopy; Classification

1. Introduction

Near infrared (NIR) spectroscopy is a vibrational spectroscopy which provides information about the molecular composition and interactions within the studied material sample [1, 2]. As the sample spectrum is a kind of signature characterizing the material, NIR spectroscopy is used in a wide range of applications, including material identification, characterization and non destructive evaluation [3, 4]. In this work the targeted application is material (wood wastes) sorting, which is envisaged as a supervised binary classification problem. The main issue in this type of application is that existing optical recycling systems can only operate using a few spectral bands to ensure a good trade-off between sorting quality and processing time which includes acquisition and classification times. To face the curse of dimensionality and avoid overfitting problems, feature selection has been recognized as a key step, especially when the variables are highly correlated. The problem is to find a small subset of variables describing the main characteristics of the different classes. Popular features selection approaches include random forest [5], Bayesian variable selection (see [6] and references therein), best subset selection [7, 8], forward and backward stepwise regression [9, 10], forward stagewise regression and sparse linear regression known as the Lasso (*Least absolute shrinkage and selection operator*) [11, 12].

Sparse representations have been widely studied over the last decade, and applied to different problems such as data compression [13, 14], pattern recognition [15], classification and clustering [16, 17], and hyperspectral image unmixing and classification [18, 19, 20]. It is based on the assumption that the essential characteristics of the data can be approximated by a linear combination of a few atoms drawn from an overcomplete dictionary of features. Feature selection may also be viewed as dimensionality reduction problem that can be tackled using a sparse approximation. The idea is simply to select the set of atoms corresponding to nonzero coefficients resulting from the approximation. Yuan and Lin [21] introduced a group sparsity criterion allowing the selection of grouped variables. An important instance of group sparsity is the simultaneous sparsity in which we seek to approximate several input signals at once using different linear combinations of the same elementary signals [22]. As it involves a counting "norm", achieving the exact simultaneous sparse decomposition is an NP-hard problem for which the greedy methods provide a good compromise between reconstruction accu-

racy and computational cost [22, 23, 24]. Convex relaxations of the simultaneous sparse approximation was also proposed by Tropp in [25].

As mentioned before, the critical point in industrial wood sorting is the selection of spectral bands (their number and their positions) to be used to guarantee a good classification rate together with a sufficient throughput (in the order of tons per hour). Nowadays, to achieve fast sorting, the number of spectral bands used in industrial NIR sorting systems is around 16. This number is sufficient for most of plastic materials but not for wood wastes. By exploiting the power of current processors, it is possible to increase this number to 30 or even 40. In any case, performing variable selection for wood waste sorting is of major importance. In this paper, we propose a simultaneous variable selection strategy for NIR spectra based on sparse decomposition. Given a set of training spectra, the core idea in this work consists in finding a small subset of wavebands/variables that captures the main spectral components shared by several measurements. The wavebands are picked from a dictionary containing Gaussian features of various centers and widths. The regression coefficients associated to the selected variables may then be used to perform classification of candidate spectra. Some similar approaches have been already proposed in the literature. For example, Turlach *et al.* [26] presented a simultaneous variable selection algorithm and applied it to NIR spectra. In contrast, the sparse decomposition problem considered in this work incorporates a regularization term enforcing the rows of the coefficient matrix to be piecewise constant. The intuition behind the proposed approach is quite simple. Consider the situation where the samples can be divided into classes which in turn include different subclasses. Rather than randomly gathering the samples into the data matrix, we propose to order them according of their subclass labels. Figure 1 illustrates this ordering for a two class problem with 11 subclasses. Consecutive samples belong to the same subclass and are expected to share common features. This will be captured by enforcing piecewise constant coefficients and group sparsity. Malli and Natschläger [12] also proposed a waveband selection algorithm for spectroscopy based on fused Lasso [27]. The fused penalty encourages the selection of connected wavelengths resulting in the so-called "wavebands". On the contrary, the method presented here consists in modeling the spectra with Gaussian-shaped functions. By doing so, not only the algorithm is structurally able to select wavebands rather than individual wavelengths but it also allows to reduce the number of spectral features. These properties are particularly suitable for high speed industrial classification because the computational cost of the regression coefficients, associated to a small number of variables, is pretty low.

The paper is organized as follows. In Section 2, we present the regularized si-

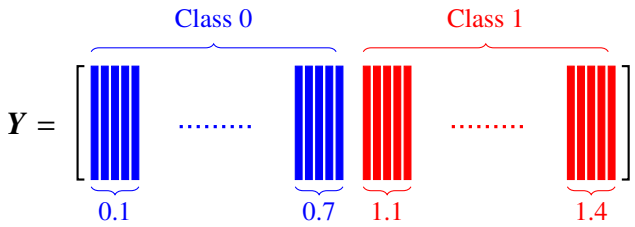


Figure 1: Illustration of spectra ordering in data matrix $\boldsymbol{Y}$. The spectra are ordered according to their class and subclass labels.

multaneous sparse approximation problem involving the mixed $L_{2,0}$ pseudo-norm. The details of the convex relaxation approaches are described in Section 3. Specifically, we first propose the $L_{1,1}$ relaxation and then the $L_{2,1}$ surrogate of the $L_{2,0}$ norm. Both relaxations lead to algorithms that enjoy a decomposition property allowing one to compute an efficient solution even for large scale problems. We also propose a nonnegative version of these algorithms. An application to wood wastes sorting based on NIR measurements is provided in Section 4. Finally, conclusions and drawn in Section 5.

Notation. Scalars are denoted by regular letters (N, s, λ), column vectors by lower-case bold-face letters $(\boldsymbol{x}, \boldsymbol{\phi})$, and matrices as bold-face capitals $(\boldsymbol{X}, \boldsymbol{\Phi})$. $\boldsymbol{x}_i$ is i-th column of $\boldsymbol{X}$ and $\boldsymbol{x}^i$ denotes the transpose of the i-th row. Notation $(\cdot)^\top$ stands for matrix or vector transposition. $\|\boldsymbol{A}\|_{p,q} = (\sum_i \|\boldsymbol{x}_i\|_p^q)^{1/q}$ is the mixed $L_{p,q}$-norm and $\|\boldsymbol{A}\|_F$ is the Frobenius (or $L_{2,2}$) norm of matrix $\boldsymbol{A}$. The symbols "$\otimes$", "$*$", and "$\circ$" denote the Kronecker product, the Hadamard (entrywise) product, and the composition operator, respectively.

2. Problem formulation

Suppose that K response variables (spectra) are collected and stacked in the columns of a data matrix $\boldsymbol{Y} \in \mathbb{R}^{M \times K}$ where M is the number of observations in each spectrum. The matrix $\boldsymbol{Y}$ is assumed to be ordered as illustrated in fig. 1. We seek to decompose the matrix $\boldsymbol{Y}$ such that:

$$\boldsymbol{Y} \approx \boldsymbol{\Phi}\boldsymbol{X}, \tag{1}$$

where $\boldsymbol{X} \in \mathbb{R}^{N \times K}$ is a *sparse* coefficient matrix meaning that only a small subset of its rows is nonzero. The columns of the redundant dictionary $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_N] \in \mathbb{R}^{M \times N}$ represents the explanatory variables (also called *atoms*). This dictionary is

designed to concentrate the energy of the signals in $\boldsymbol{Y}$ over a set of a few atoms. Its choice depends essentially on the application at hand. As in NIR spectra, the observed peaks are typically very broad, we assume in the present work that the $\boldsymbol{\phi}_n$'s are Gaussiand-shaped functions whose locations (central wavelengths) and widths cover all the NIR range. In other words, each atom models a relevant spectral band in the available data.

The simultaneous sparse approximation [28, 29] consists in finding a solution $\boldsymbol{X}$ having a limited number of active rows. The problem can be formulated as

$$\underset{\boldsymbol{X}}{\text{minimize}} \quad \tfrac{1}{2}\|\boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{X}\|_F^2, \tag{2a}$$

$$\text{subject to} \quad \|\boldsymbol{X}^\top\|_{2,0} \leq s, \tag{2b}$$

where $\|\boldsymbol{X}^\top\|_{2,0}$ is the mixed $L_{2,0}$ pseudo-norm of $\boldsymbol{X}^\top$ (*i.e.* the number of rows with nonzero ℓ_2-norm) and $s \ll N$ is the sparsity parameter which is related to the support of $\boldsymbol{X}$: $\text{supp}(\boldsymbol{X}) = \{1 \leq n \leq N \,|\, \boldsymbol{x}^n \neq \boldsymbol{0}\}$. The rationale behind simultaneous reconstruction for variable selection is to find predictors for all input signals in $\boldsymbol{Y}$ from a *common subset* of active variables [26] which are indexed by the support of the solution $\boldsymbol{X}$.

The regularized simultaneous sparse approximation aims at reconstructing piecewise constant rows. In that respect, as in the study by Malli and Natschläger [12], we propose to include a regularization term leading to the following problem

$$\underset{\boldsymbol{X}}{\text{minimize}} \quad \tfrac{1}{2}\|\boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{X}\|_F^2 + \lambda_2\|\boldsymbol{D}\boldsymbol{X}^\top\|_{1,1}, \tag{3a}$$

$$\text{subject to} \quad \|\boldsymbol{X}^\top\|_{2,0} \leq s, \tag{3b}$$

where $\boldsymbol{D} \in \mathbb{R}^{(K-1)\times K}$ is a matrix of finite differences of order 1:

$$\boldsymbol{D} = \begin{bmatrix} -1 & 1 & & \boldsymbol{0} \\ & \ddots & \ddots & \\ \boldsymbol{0} & & -1 & 1 \end{bmatrix}. \tag{4}$$

Criterion (3) includes an additional total variation-like penalty enforcing sparsity on the difference between successive columns of $\boldsymbol{X}$: $\|\boldsymbol{D}\boldsymbol{X}^\top\|_{1,1} = \sum_{i=1}^{K-1} \|\boldsymbol{x}_{i+1} - \boldsymbol{x}_i\|_1$. This, in fact, promotes the reconstruction of piecewise constant rows. Again, it is important to note that this penalty makes sense only if the signals in $\boldsymbol{Y}$ have a meaningful ordering. This is for example the case when the signals are ordered

according to their subclass labels in the training phase. In hyperspectral image classification, the signals are naturally ordered according to their spatial position. Due to the ℓ_1 penalty, consecutive columns of $\boldsymbol{X}$ tend to be similar which helps to decrease intraclass variance. Following the terminology of the fused Lasso, this term will be referred to as the fusion penalty. The formulation in (3) involving the mixed $L_{2,0}$ norm will lead to an NP-hard problem, thus making the resolution not easy. In the next section, we propose a convex relaxation of the $L_{2,0}$ norm and the resulting problem is solved using fast and effective algorithms.

3. Convex relaxations

3.1. Fused Sparse Lasso

The first relaxation of the constrained problem (3) consists in minimizing the following penalized criterion:

$$J_{FSL}(\boldsymbol{X}) = \tfrac{1}{2}\|\boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{X}\|_F^2 + \lambda_1\|\boldsymbol{X}^\top\|_{1,1} + \lambda_2\|\boldsymbol{D}\boldsymbol{X}^\top\|_{1,1} \tag{5}$$

where $\|\boldsymbol{X}^\top\|_{1,1} = \sum_{i=1}^{K}\|\boldsymbol{x}_i\|_1 = \sum_{n=1}^{N}\|\boldsymbol{x}^n\|_1$. The parameters $\lambda_1, \lambda_2 \geq 0$ are controlling the tradeoff between data fitting, sparsity $\|\boldsymbol{X}^\top\|_{1,1}$, and fusion penalty $\|\boldsymbol{D}\boldsymbol{X}^\top\|_{1,1}$. This criterion is in fact an extension to the multiple measurement vector setting of the sparse fused Lasso which was studied Tibshirani, Saunders *et al.* [27]. Friedman *et al.* [30] proposed to solve the generalized sparse fused Lasso problem (including both sparsity and fusion terms) in the special case where the dictionary $\boldsymbol{\Phi}$ is an identity matrix. This work was then extended by Xin *et al.* [31] to general dictionary. Here, we propose to solve this problem in the special case where the fusion term only acts on the rows of $\boldsymbol{X}$. According to this specific structure it is possible to obtain a computationally efficient implementation of the minimization problem. However, before going further, let us give a few comments on criterion (5). In fact, the sparsity term $\|\boldsymbol{X}^\top\|_{1,1}$ does not correspond to a proper convex relaxation of $\|\boldsymbol{X}^\top\|_{2,0}$. As will be explained in section 3.2, the mixed norm $\|\boldsymbol{X}^\top\|_{2,1}$ is more appropriate. But combining $\|\boldsymbol{X}^\top\|_{1,1}$ to $\|\boldsymbol{D}\boldsymbol{X}^\top\|_{1,1}$ yields to a kind of simultaneous sparse approximation: the simultaneity is actually enforced by the row regularization term $\|\boldsymbol{D}\boldsymbol{X}^\top\|_{1,1}$, but there is no direct control on the number of active rows.

Let vec$(\cdot)$ be the vectorization operator that converts a matrix into a vector by stacking the columns of the matrix on top of one another. We set $\boldsymbol{x} = \text{vec}(\boldsymbol{X}^\top)$ and $\boldsymbol{y} = \text{vec}(\boldsymbol{Y}^\top)$. Then, criterion (5) can be rewritten as:

$$J_{FSL}(\boldsymbol{x}) = \tfrac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda_1\|\boldsymbol{x}\|_1 + \lambda_2\|\boldsymbol{F}\boldsymbol{x}\|_1 \tag{6}$$

with $\boldsymbol{A} = \boldsymbol{\Phi} \otimes \boldsymbol{I}_K \in \mathbb{R}^{NK \times MK}$, $\boldsymbol{F} = \boldsymbol{I}_N \otimes \boldsymbol{D} \in \mathbb{R}^{N(K-1) \times NK}$, and $\boldsymbol{I}_N$ denotes the $N \times N$ identity matrix. Note that (6) is also a generalization to the multiple measurement vector setting of the fused Lasso criterion already proposed for variable selection in spectroscopy by Malli and Natschläger [12]. To minimize (6), we use FISTA (Fast Iterative Shrinkage-Thresholding Algorithm) [32] designed to solve convex optimization problems including both smooth and non-smooth terms. First, let

$$f(\boldsymbol{x}) = \tfrac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2, \tag{7}$$

$$g(\boldsymbol{x}) = \lambda_1\|\boldsymbol{x}\|_1 + \lambda_2\|\boldsymbol{F}\boldsymbol{x}\|_1. \tag{8}$$

Then, following Xin *et al.* [31], the update of vector $\boldsymbol{x}$ at iteration $k+1$ is:

$$\boldsymbol{x}_{(k+1)} = \arg\min_{\boldsymbol{x} \in \mathbb{R}^{NK}} \left(g(\boldsymbol{x}) + \tfrac{L}{2}\|\boldsymbol{x} - \boldsymbol{v}_{(k)}\|_2^2 \right) \tag{9}$$

where

$$\boldsymbol{v}_{(k)} = \boldsymbol{x}_{(k)} - \tfrac{1}{L}\nabla f(\boldsymbol{x}_{(k)}) \tag{10}$$

and $\nabla f(\boldsymbol{x}) = \boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y})$ is the gradient of $f(\boldsymbol{x})$. L is the Lipschitz constant of $\nabla f(\boldsymbol{x})$. Note that the update of $\boldsymbol{v}_{(k)}$ according to (10) involves the calculation and storage of $\boldsymbol{A}^\top\boldsymbol{A}$ and $\boldsymbol{A}^\top\boldsymbol{y}$. Instead, to save on computational costs, we can update the matrix $\boldsymbol{V}_{(k)}$ according to:

$$\boldsymbol{V}_{(k)} = \boldsymbol{X}_{(k)} - \tfrac{1}{L}\boldsymbol{\Phi}^\top(\boldsymbol{\Phi}\boldsymbol{X}_{(k)} - \boldsymbol{Y}), \tag{11}$$

where $\boldsymbol{V}_{(k)}$ is the matrix satisfying $\boldsymbol{v}_{(k)} = \mathrm{vec}(\boldsymbol{V}_{(k)}^\top)$. As a consequence, only lower dimension matrices, $\boldsymbol{\Phi}^\top\boldsymbol{\Phi}$ and $\boldsymbol{\Phi}^\top\boldsymbol{Y}$, need to be computed and stored. Solving (9) is similar to the 1D fused Lasso signal approximator (FLSA) [33]. Moreover, due to the block diagonal structure of $\boldsymbol{F}$, it is obvious that $\|\boldsymbol{F}\boldsymbol{x}\|_1 = \sum_{n=1}^{N}\|\boldsymbol{D}\boldsymbol{x}^n\|_1$. Therefore, problem (9) can be solved separately for each row $\boldsymbol{x}^n$ of $\boldsymbol{X}$:

$$\boldsymbol{x}^n_{(k+1)} = \arg\min_{\boldsymbol{x} \in \mathbb{R}^K} \tfrac{1}{2}\|\boldsymbol{x} - \boldsymbol{v}^n_{(k)}\|_2^2 + \tfrac{\lambda_1}{L}\|\boldsymbol{x}\|_1 + \tfrac{\lambda_2}{L}\|\boldsymbol{D}\boldsymbol{x}\|_1. \tag{12}$$

The solution to (12) is obtained by using subgradient technique. Indeed, any solution corresponding to (λ_1, λ_2) is obtained by a soft thresholding of the solution obtained for $(\lambda_1 = 0, \lambda_2)$. This is stated by the following theorem.

Theorem 1 (Friedman *et al.* [30], Liu *et al.* [34]). *Let*

$$\boldsymbol{x}(\lambda_1, \lambda_2) = \arg\min_{\boldsymbol{x}} \tfrac{1}{2}\|\boldsymbol{x} - \boldsymbol{v}\|_2^2 + \lambda_1\|\boldsymbol{x}\|_1 + \lambda_2\|\boldsymbol{D}\boldsymbol{x}\|_1. \tag{13}$$

Algorithm 1: Fused Sparse Lasso (FSL)

Input : $\boldsymbol{Y} \in \mathbb{C}^{M\times K}$, $\boldsymbol{\Phi} \in \mathbb{C}^{M\times N}$, λ_1, λ_2, *maxiter*

Initialization: $\boldsymbol{X}_{(0)} = \boldsymbol{0}$, $\boldsymbol{Z}_{(1)} = \boldsymbol{0}$, $t_{(1)} = 1$;
for $k \leftarrow 1$ **to** *maxiter* **do**
  $\boldsymbol{V}_{(k)} \leftarrow \boldsymbol{Z}_{(k)} - \frac{1}{L}\boldsymbol{\Phi}^\top(\boldsymbol{\Phi}\boldsymbol{Z}_{(k)} - \boldsymbol{Y})$;
  for $n \leftarrow 1$ **to** N **do**
    $\mathbf{x}^n_{(k)} \leftarrow \arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{v}^i_{(k)}\|_2^2 + \frac{\lambda_1}{L}\|\boldsymbol{x}\|_1 + \frac{\lambda_2}{L}\|\boldsymbol{D}\boldsymbol{x}\|_1$;
  end
  $t_{(k+1)} \leftarrow \frac{1+\sqrt{1+4t^2_{(k)}}}{2}$;
  $\boldsymbol{Z}_{(k+1)} \leftarrow \boldsymbol{X}_{(k)} + \frac{t_{(k)}-1}{t_{(k+1)}}(\boldsymbol{X}_{(k)} - \boldsymbol{X}_{(k-1)})$;
end

Output: $\boldsymbol{X} \in \mathbb{R}^{N\times K}$

For all λ_1, $\lambda_2 \geq 0$, we have:

$$\boldsymbol{x}(\lambda_1, \lambda_2) = \text{sign}(\boldsymbol{x}(0, \lambda_2)) * \max(|\boldsymbol{x}(0, \lambda_2)| - \lambda_1, 0). \tag{14}$$

where $*$ denotes the element-wise product operator. □

In this paper, each problem in (12) is solved using the FLSA routine implemented in SLEP package[1]. Finally, the main steps of the Fused Sparse Lasso (FSL) algorithm are presented in Algorithm 1, where $\boldsymbol{Z}$ is a linear combination of two consecutive estimates of $\boldsymbol{X}$; it is updated at each FISTA iteration.

3.2. Fused Sparse Group Lasso

The fused sparse Lasso is not a proper relaxation of the problem in (3). Indeed, the term $\|\boldsymbol{X}^\top\|_{1,1}$ does not allow to control the number of active rows. Here, we propose to relax the $L_{2,0}$ pseudo-norm into the $L_{2,1}$ mixed norm defined by: $\|\boldsymbol{X}^\top\|_{2,1} = \sum_{n=1}^{N} \|\boldsymbol{x}^n\|_2$, which is a particular instance of the group Lasso penalty. So we propose the following criterion:

$$J_{FSGL}(\boldsymbol{x}) = \tfrac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda_1\|\boldsymbol{x}\|_1 + \lambda_2\|\boldsymbol{F}\boldsymbol{x}\|_1 + \lambda_3 \sum_{n=1}^{N} \|\boldsymbol{x}^n\|_2, \tag{15}$$

[1] `http://yelab.net/software/SLEP/`

as a convex relaxation of problem (3). Note that the $\|\boldsymbol{x}\|_1$ penalty is maintained to eventually control the global sparsity of the solution. The proximal operator associated with the composite of non-smooth penalties in the fused sparse group Lasso (FSGL) is defined as:

$$\begin{aligned}\text{prox}_{FSGL}(\boldsymbol{v}) = \arg\min_{\boldsymbol{x}} &\tfrac{1}{2}\|\boldsymbol{x}-\boldsymbol{v}\|_2^2 \\ &+ \tfrac{\lambda_1}{L}\|\boldsymbol{x}\|_1 + \tfrac{\lambda_2}{L}\|\boldsymbol{F}\boldsymbol{x}\|_1 + \tfrac{\lambda_3}{L}\sum_{n=1}^{N}\|\boldsymbol{x}^n\|_2. \end{aligned} \tag{16}$$

Here again, it is clear that each row of $\boldsymbol{X}$ is decoupled in (16). So we only need to solve the following optimization problem for each row $n = 1, \ldots, N$:

$$\begin{aligned}\text{prox}_{FSGL}(\boldsymbol{v}^n) = \arg\min_{\boldsymbol{x}^n} &\tfrac{1}{2}\|\boldsymbol{x}^n-\boldsymbol{v}^n\|_2^2 \\ &+ \tfrac{\lambda_1}{L}\|\boldsymbol{x}^n\|_1 + \tfrac{\lambda_2}{L}\|\boldsymbol{D}\boldsymbol{x}^n\|_1 + \tfrac{\lambda_3}{L}\|\boldsymbol{x}^n\|_2. \end{aligned} \tag{17}$$

Now, with the three non-smooth terms in the objective function, the proximal operator may be computed as suggested in [35]. In fact, the proximal operator in (17) has a decomposition property that allows to compute it in two steps based on the following theorem.

Theorem 2 (Zhou *et al.* [35]). *Define*

$$\text{prox}_{FSL}(\boldsymbol{v}) = \arg\min_{\boldsymbol{x}} \tfrac{1}{2}\|\boldsymbol{x}-\boldsymbol{v}\|_2^2 + \lambda_1\|\boldsymbol{x}\|_1 + \lambda_2\|\boldsymbol{D}\boldsymbol{x}\|_1 \tag{18}$$

$$\text{prox}_{GL}(\boldsymbol{v}) = \arg\min_{\boldsymbol{x}} \tfrac{1}{2}\|\boldsymbol{x}-\boldsymbol{v}\|_2^2 + \lambda_3\|\boldsymbol{x}\|_2. \tag{19}$$

Then, the following holds for all $\lambda_1, \lambda_2, \lambda_3 \geq 0$:

$$\text{prox}_{FSGL}(\boldsymbol{v}) = (prox_{GL} \circ \text{prox}_{FSL})(\boldsymbol{v}). \tag{20}$$

where $\circ$ is the composition operator. □

This result implies that we can first compute the proximal operator associated to the fused sparse Lasso as in the previous section. The solution is then plugged in the proximal operator associated to the group Lasso. The latter is finally computed using the ALTRA routine also available in the SLEP package. The resulting algorithm (FSGL) is summarized in Algorithm 2.

Algorithm 2: Fused Sparse Group Lasso (FSGL)

Input : $\boldsymbol{Y} \in \mathbb{C}^{M\times K}$, $\boldsymbol{\Phi} \in \mathbb{C}^{M\times N}$, λ_1, λ_2, λ_3, *maxiter*

Initialization: $\boldsymbol{X}_{(0)} = \boldsymbol{0}$, $\boldsymbol{Z}_{(1)} = \boldsymbol{0}$, $t_{(1)} = 1$;
for $k \leftarrow 1$ **to** *maxiter* **do**
  $\boldsymbol{V}_{(k)} \leftarrow \boldsymbol{Z}_{(k)} - \frac{1}{L}\boldsymbol{\Phi}^\top(\boldsymbol{\Phi}\boldsymbol{Z}_{(k)} - \boldsymbol{Y})$;
  for $n \leftarrow 1$ **to** N **do**
    $\mathbf{w}^n_{(k)} \leftarrow \arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{v}^n_{(k)}\|_2^2 + \frac{\lambda_1}{L}\|\boldsymbol{x}\|_1 + \frac{\lambda_2}{L}\|\boldsymbol{D}\boldsymbol{x}\|_1$;
    $\boldsymbol{x}^n_{(k)} \leftarrow \arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{x} - \mathbf{w}^n_{(k)}\|_2^2 + \frac{\lambda_3}{L}\|\boldsymbol{x}\|_2$;
  end
  $t_{(k+1)} \leftarrow \frac{1+\sqrt{1+4t^2_{(k)}}}{2}$;
  $\boldsymbol{Z}_{(k+1)} \leftarrow \boldsymbol{X}_{(k)} + \frac{t_{(k)}-1}{t_{(k+1)}}(\boldsymbol{X}_{(k)} - \boldsymbol{X}_{(k-1)})$;
end

Output: $\boldsymbol{X} \in \mathbb{R}^{N\times K}$

3.3. Nonnegative Fused Sparse Group Lasso

As we deal with positive data, it is suitable to impose a nonnegativity constraint on the solution. Indeed, the solution proposed above may induce artifacts due to bad conditioning of matrices, causing the appearance of negative values. From a physical point of view, such a solution is unacceptable and a rigorous recovery process must take into account this additional constraint. So, we propose here to minimize the nonnegative version of the fused sparse group Lasso algorithm. The constrained problem expresses as:

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad J_{FSGL}(\boldsymbol{x}), \tag{21a}$$

$$\text{subject to} \quad \boldsymbol{x} \geq 0. \tag{21b}$$

First, we include a slack variable $\boldsymbol{u} \in \mathbb{R}^{NK}$ to the objective function which leads to:

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad J_{FSGL}(\boldsymbol{x}), \tag{22a}$$

$$\text{subject to} \quad \boldsymbol{x} - \boldsymbol{u} = 0, \boldsymbol{u} \geq 0. \tag{22b}$$

The equality constraint in (22b) can be handled by using the quadratic penalty method [36]. The new objective in then:

$$J_{NN-FSGL}(\boldsymbol{x},\boldsymbol{u}) = \tfrac{1}{2}\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2 + \tfrac{\xi}{2}\|\boldsymbol{x}-\boldsymbol{u}\|_2^2 + \lambda_1\|\boldsymbol{x}\|_1 + \lambda_2\|\boldsymbol{F}\boldsymbol{x}\|_1 + \lambda_3\sum_{n=1}^{N}\|\boldsymbol{x}^n\|_2, \quad \boldsymbol{u}\geq 0 \tag{23}$$

where ξ is the parameter that penalizes the constraint violations in the sense that, when $\xi\to\infty$, the entries of the vector $\boldsymbol{x}$ tend toward those of the vector $\boldsymbol{u}$ making the inequality constraint $\boldsymbol{x}\geq 0$ satisfied asymptotically. The surrogate problem (23) is unconstrained with respect to $\boldsymbol{x}$. Hence, by stacking the two quadratic terms of the objective $J_3(\cdot)$ we obtain:

$$J_{NN-FSGL}(\boldsymbol{x},\boldsymbol{u}) = \tfrac{1}{2}\|\boldsymbol{y}'(\boldsymbol{u})-\boldsymbol{B}\boldsymbol{x}\|_2^2 + g'(\boldsymbol{x}), \quad \boldsymbol{u}\geq 0 \tag{24}$$

where $\boldsymbol{B} = [\boldsymbol{A}^\top,\ \sqrt{\xi}\boldsymbol{I}^\top]^\top$, $\boldsymbol{y}'(\boldsymbol{u}) = [\boldsymbol{y}^\top,\ \sqrt{\xi}\boldsymbol{u}^\top]^\top$, $\boldsymbol{I}$ is an identity matrix of the same size as $\boldsymbol{A}$, and $g'(\boldsymbol{x})$ is defined by:

$$g'(\boldsymbol{x}) = \lambda_1\|\boldsymbol{x}\|_1 + \lambda_2\|\boldsymbol{F}\boldsymbol{x}\|_1 + \lambda_3\sum_{i=1}^{N}\|\boldsymbol{x}^i\|_2. \tag{25}$$

We should now minimize the cost function $J_{NN-FSGL}(\boldsymbol{x},\boldsymbol{u})$ with respect to $\boldsymbol{x}$ (without constraint) and $\boldsymbol{u}$ (with the nonnegativity constraint). The minimization with respect to $\boldsymbol{x}$ leads to an iteration similar to that of FSGL:

$$\begin{cases} \boldsymbol{v}_{(k)} = \boldsymbol{x}_{(k)} - \frac{1}{L'}\nabla f'(\boldsymbol{x}_{(k)}) \\ \boldsymbol{x}_{(k+1)} = \arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{v}_{(k)}\|_2^2 + \frac{1}{L'}g'(\boldsymbol{x}), \end{cases} \tag{26}$$

where $f'(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{y}'(\boldsymbol{u})-\boldsymbol{B}\boldsymbol{x}\|_2^2$, $\nabla f'(\boldsymbol{x}) = \boldsymbol{B}^\top(\boldsymbol{B}\boldsymbol{x}-\boldsymbol{y}'(\boldsymbol{u}))$ and $L' = L+\xi$ is the Lipschitz constant of $\nabla f'(\boldsymbol{x})$. Once again, the optimization is separable for each row $\boldsymbol{x}^n$. Define matrix $\boldsymbol{U}$ such that $\boldsymbol{u} = \mathrm{vec}(\boldsymbol{U}^\top)$. Replacing $\boldsymbol{B}$ and $\boldsymbol{y}'(\boldsymbol{u})$ by their expressions yields:

$$\begin{cases} \boldsymbol{V}_{(k)} & = \boldsymbol{X}_{(k)} - \frac{1}{L'}\boldsymbol{\Phi}^\top(\boldsymbol{\Phi}\boldsymbol{X}_{(k)}-\boldsymbol{Y}) - \frac{\xi}{L'}(\boldsymbol{X}_{(k)}-\mathbf{U}_{(\ell)}), \\ \boldsymbol{x}^n_{(k+1)} & = \arg\min_{\boldsymbol{x}\in\mathbb{R}^K} \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{v}^n_{(k)}\|_2^2 + \frac{\lambda_1}{L'}\|\boldsymbol{x}\|_1 + \frac{\lambda_2}{L'}\|\boldsymbol{D}\boldsymbol{x}\|_1 + \frac{\lambda_3}{L'}\|\boldsymbol{x}\|_2, \\ & \text{for } n = 1,\ldots,N. \end{cases} \tag{27}$$

Algorithm 3: Nonnegative Fused Sparse Group Lasso (NN-FSGL)

Input : $\boldsymbol{Y} \in \mathbb{C}^{M \times K}$, $\boldsymbol{\Phi} \in \mathbb{C}^{M \times N}$, λ_1, λ_2, λ_3, β, *maxiter*, *nniter*

Initialization: $\boldsymbol{X}_{(0)} = \boldsymbol{0}$, $\boldsymbol{Z}_{(1)} = \boldsymbol{0}$, $t_{(1)} = 1$, $\boldsymbol{u} = 0$, $\xi_{(1)} = 1$;
for $\ell \leftarrow 1$ **to** *nniter* **do**
  $L' \leftarrow L + \xi_{(\ell)}$;
  for $k \leftarrow 1$ **to** *maxiter* **do**
    $\boldsymbol{V}_{(k)} \leftarrow \boldsymbol{Z}_{(k)} - \frac{1}{L'}\boldsymbol{\Phi}^\top(\boldsymbol{\Phi}\boldsymbol{Z}_{(k)} - \boldsymbol{Y}) - \frac{\xi_{(\ell)}}{L'}(\boldsymbol{Z} - \mathbf{U}_{(\ell)})$;
    for $n \leftarrow 1$ **to** N **do**
      $\mathbf{w}^n_{(k)} \leftarrow \arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{v}^n_{(k)}\|_2^2 + \frac{\lambda_1}{L'}\|\boldsymbol{x}\|_1 + \frac{\lambda_2}{L'}\|\boldsymbol{D}\boldsymbol{x}\|_1$;
      $\boldsymbol{x}^n_{(k)} \leftarrow \arg\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{x} - \mathbf{w}^n_{(k)}\|_2^2 + \frac{\lambda_3}{L'}\|\boldsymbol{x}\|_2$;
    end
    $t_{(k+1)} \leftarrow \frac{1+\sqrt{1+4t^2_{(k)}}}{2}$;
    $\boldsymbol{Z}_{(k+1)} \leftarrow \boldsymbol{X}_{(k)} + \frac{t_{(k)}-1}{t_{(k+1)}}(\boldsymbol{X}_{(k)} - \boldsymbol{X}_{(k-1)})$;
  end
  $\boldsymbol{u}_{(\ell+1)} \leftarrow \max(0, \boldsymbol{x}_{(maxiter)})$;
  $\xi_{(\ell+1)} \leftarrow \beta\xi_{(\ell)}$;
end

Output: $\boldsymbol{X} \in \mathbb{R}^{N \times K}$

Hence, an external loop (ℓ) is added to update the variable $\boldsymbol{u}$. The minimization of $J_{NN-FSGL}(\boldsymbol{x}, \boldsymbol{u})$ with respect to the slack variable $\boldsymbol{u}$ is simply a hard thresholding operation:

$$\boldsymbol{u}_{(\ell+1)} = \max(0, \boldsymbol{x}^*), \tag{28}$$

where $\boldsymbol{x}^*$ is the value of $\boldsymbol{x}_{(k)}$ when the final iteration on k is completed. The tuning parameter ξ is updated in the loop with the classical linear rule: $\xi_{(\ell+1)} = \beta\xi_{(\ell)}$, with $\beta > 1$ and $\xi_1 = 1$. The complete NN-FSGL algorithm is summarized in Algorithm 3.

3.4. Sofware

An open source Matlab implementation of FSL, FSGL and NN-FSGL can be downloaded from `http://w3.cran.univ-lorraine.fr/el-hadi.djermoune/?q=content/publications`. The software also contains a test program and the experimental NIR data used in the next section.

Table 1: Composition of the two classes of wood wastes

(a) Class 0: recyclable

Subclass	Type	Samples
0.1	raw wood	32
0.2	painted solid wood	36
0.3	varnished solid wood	35
0.4	raw plywood	16
0.5	varnished plywood	16
0.6	raw particle board	28
0.7	painted particle board	6

(b) Class 1: non-recyclable

Subclass	Type	Samples
1.1	raw wood metallic salts	35
1.2	MDF-HDF	28
1.3	painted MDF-HDF	50
1.4	raw fiber board	8

4. Wood wastes sorting

4.1. Motivations

We are interested in sorting wood wastes which have to be separated into two broad classes: recyclable and non recyclable. Each class includes a number of wood wastes types called "subclasses" as given in Table 1. The wood wastes sorting is addressed as a binary classification of NIR spectra. A single spectrum is acquired for each wood sample and the classifier has to decide whether it is recyclable or not recyclable.

The goal of this section is to show the effectiveness of the algorithms presented before in variable selection and classification. These algorithms are primarily intended at selecting the explanatory variables used in classifiers. Here we restrict our attention to the kernel SVM classifier which proved to be among the most effective for the considered problem, and the question at hand is: is it possible to improve the classification rates and decrease the computational burden by performing a proper variable selection?

4.2. Data acquisition and pre-processing

We collected several hundred samples of wood in a waste park amongst which 290 were gathered by experts into 11 labeled subclasses as shown in Table 1. The data acquisition was carried in reflectance mode on a Nicolet 8700 FTIR spectrometer equipped with a MCT detector and a CaF2 beam splitter. Near infrared reflectance spectra cover the spectral range $[3562, 10000]$ cm^{-1} (corresponding to $[1000, 2800]$ nm). The spectral resolution is 16 cm^{-1}. The spectral sampling step is 4 cm^{-1} yielding 1647 spectral bands. Each spectrum is obtained by averaging 100 scans. The data pre-processing includes baseline removal using the method proposed in [37], offset correction ensuring zero lower bound, and unit energy normalization. Some spectra from these different subclasses are shown in Figure 2. It appears that the discriminant features cannot be determined by a simple visual examination.

The data are then gathered in matrix $\boldsymbol{Y} \in \mathbb{R}^{1647\times 290}$. Note that the spectra are ordered according to the subclass they belong to. The spectra from class 0 are put in the first columns of $\boldsymbol{Y}$ starting from subclass 0.1 through subclass 0.7. In the same manner, the spectra from class 1 are put in the last columns. This is a very important point since it is this ordering which enables the rows of the coefficient matrix $\boldsymbol{X}$ to be piecewise constant when $\lambda_2 > 0$.

4.3. Dictionary

The dictionary $\boldsymbol{\Phi}$ is composed of normalized Gaussian-shaped functions whose means $m_{i,j} \in [3660, 10000]$ cm^{-1} and widths $\sigma_j \in [30, 600]$ cm^{-1} are covering uniformly their respective intervals. The discretization leads to 20 different values for σ_j. For each σ_j, the interval $[3660, 10000]$ cm^{-1} is discretized such that two adjacent $m_{i,j}$'s corresponding to the same σ_j are separated by σ_j. Specifically, for $j = 1, \ldots, 20$:

$$\sigma_j = 30j, \tag{29}$$

$$m_{i,j} = 3660 + (i-1)\sigma_j, \quad i = 1, \ldots, \left\lfloor \tfrac{10000-3660}{\sigma_j} \right\rfloor \tag{30}$$

where $\lfloor \cdot \rfloor$ is the floor function. As a consequence, the dictionary is composed of 773 atoms.

4.4. Variable selection

Here we compare FSL and FSGL to the simultaneous variable selection algorithm (SVS) proposed by Turlach *et al.* [26]. This algorithm is an extension of

Lasso strategy and corresponds to the following optimization problem:

$$\min_{\boldsymbol{X}} \tfrac{1}{2}\|\boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{X}\|_F^2 \quad \text{s. t.} \quad \sum_{n=1}^{N} \|\boldsymbol{x}^n\|_\infty \leq t, \tag{31}$$

where t is a user parameter controlling the sparsity of the solution. The problem is solved using an interior point method and the C implementation is kindly provided by the author (Berwin A. Turlach).

Figure 3 displays the selected variables obtained by SVS, FSL and FSGL for some values of t, $1/\lambda_1$ and $1/\lambda_3$, respectively. Note that the number of active variables returned by SVS increases when t increases whereas, for FSL and FSGL, it decreases when λ_1 or λ_3 increases. The horizontal lines connect two adjacent values of the parameters when the coefficient associated to a selected variable does not vanish. For small values of t, $1/\lambda_1$ and $1/\lambda_3$, the variables are mainly picked in the range $[6600, 6700]$ cm^{-1} where broad and intense spectral peaks are observed (see Fig. 2). By increasing the value of these parameters, more and more variables are selected. In a given application, the practitioner can stop the selection when the desired number of variables is reached. In our case, out of about fourty variables (with $t = 2$, $\lambda_1 = 0.045$ and $\lambda_3 = 0.35$), SVS shares 28 common variables with FSL and FSGL. The latter algorithms share 38 common variables. It can also be seen that some wavenumbers actually have a chemical interpretation. For instance, the variables located in the ranges 4000-4500 cm^{-1} and 5800-8200 cm^{-1} are related to the main components of wood including cellulose, hemicellulose and and lignin [38, 39].

The computational time required by each algorithm to perform variable selection is reported in Table 2. The results are obtained using a 2.4 Ghz Intel Core i5 processor with 8 Gigabytes of RAM. We note that FSL is generally faster than all other approaches. FSGL algorithm is a bit slower. The additional loop with hard thresholding operator makes NN-FSGL about ten times time demanding than its unconstrained version. For SVS we did not try all the configurations because we found that this algorithm is much more slower and needs about four hours to select 32 variables. Finally, FSL and FSGL algorithms are not only numerically efficient but also provide good classification rates as will be shown in the next paragraph.

4.5. Classification of wood wastes using NIR spectra

Here we perform classification of recyclable and non-recyclable wood samples using SVM with quadratic kernel function. The variable selection algorithms

Table 2: Computational time (in seconds) of the different approaches for variable selection

# Variables	FSL	FSGL	NN-FSGL	G-SVM
25	9	17	110	27
32	9	17	101	27
40	10	15	112	29
50	9	12	105	31

SVS, FSL, FSGL and NN-FSGL are tuned to produce 32 spectral bands. Classification is then performed using matrix $\boldsymbol{X}$ corresponding to the unconstrained least-squares solution of equation (2a) where dictionary $\boldsymbol{\Phi}$ is restricted to the 32 active atoms. The results are compared to those obtained with the following algorithms:

- *SVM*: the classification is performed on the original data in $\boldsymbol{Y}$ without variable selection[2].
- *G-SVM [40]*: it consists in solving an augmented SVM criterion where the sparsity constraint is imposed on the support vectors. The sparsity of the support vectors is controlled by parameter[3] C which is set to $C = 150$ making the decision rule made on 32 entries of the support vectors.
- *Wavelet-based Bayesian variable selection (W-BVS) [6, 41]*: the wavelet transform is carried out using Daubechies wavelets with 4 vanishing moments. The Bayesian selection method[4] requires the use of Markov chain Monte Carlo (MCMC) algorithms that have to be tuned to select 32 variables. Following the notations in Sha *et al.* [41], these parameters include, among others, h, c, and w.
- *Random forest [5]*: the algorithm is applied on the original data $\boldsymbol{Y}$ by setting the number of decision splits to $T = 289$ and the number of predictors (features) to select at random for each split to 32.

[2]As the number of variables (wavenumbers) in SVM is much greater than the number of spectra, the results obtained on our database may not be generalizable. They are given here only as an indicative basis.

[3]`http://remi.flamary.com/soft/soft-gsvm.html`

[4]The Matlab code for Bayesian variable selection is downloaded from `http://www.stat.rice.edu/~marina/matlab/bvsprob.tar`

Table 3: Wood wastes classification accuracy

Algorithm	Number of variables	Parameters			Accuracy		
		λ_1	λ_2	λ_3	Success	TPR	TNR
SVM	1647	–			77.7 ± 1.4%	77.6 ± 0.9%	77.8 ± 1.0%
G-SVM	32	$C = 150$			76.9 ± 1.5%	81.4 ± 1.8%	71.3 ± 1.7%
W-BVS	32	$h = 10^6$, $c = 10$, $w = 32/M$			81.5 ± 1.5%	84.8 ± 1.4%	77.1 ± 2.0%
Random forest	32	$T = 289$			82.6 ± 1.5%	83.9 ± 1.2%	80.7 ± 2.3%
SVS	32	$t = 1.75$			84.1 ± 1.2%	83.1 ± 1.3%	85.6 ± 1.9%
FSL	32	0.075	0.5	–	85.9 ± 0.8%	85.6 ± 1.2%	86.3 ± 1.3%
FSGL	32	0	0.5	0.625	**87.8 ± 0.8%**	**86.1 ± 1.3%**	**90.3 ± 1.8%**
	32	0	0	0.8	86.6 ± 1.2%	84.7 ± 1.5%	89.3 ± 1.6%
NN-FSGL	32	0	0.5	0.6	86.9 ± 1.3%	85.5 ± 1.9%	88.8 ± 1.7%

The classification results for 10 cross validation runs are shown in Table 3. For each method we report the overall rate of success, the true positive rate TPR (rate of recyclable samples correctly identified), and the true negative rate TNR (rate of non-recyclable samples correctly rejected). In terms of total accuracy, FSL, FSGL ad NN-FSGL outperform their competitors. The best result is about 88% obtained with FSGL. As in our application it is also important to reject the maximum number of polluted samples from the recycling process, the performances of SVM, G-SVM and W-BVS are clearly not satisfactory. Here, the best parameters for FSGL are $\lambda_1 = 0$, $\lambda_2 = 0.5$ and $\lambda_3 = 0.625$ (see also section 4.6). We note that, while NN-FSGL is slightly outperformed by FSGL in termes of classification rate, the nonnegativity constraint is relevant if a further physical interpretation is required. Figure 4 shows an example of error rates resulting in each subclass of wood wastes when the spectra are randomly split into training samples (203 spectra) and test ones (87 spectra). The results obtained using the variables selected by FSGL are: TPR = 87.3%, TNR = 94.4%, and an overall rate of success of 89.7%. One can see that the two pieces of painted particle boards are misclassified. This is mainly due to *(i)* the small number of samples in the corresponding subclass: only 4 samples are used for training and 2 for the test; *(ii)* the presence of painted wood samples in both classes.

4.6. Parameter adjustment

The performances of all the algorithms considered here depend on the choice of tuning parameters. For instance, the set of selected variables (and thus, the overall classification performance) with FSGL depend on λ_1, λ_2 and λ_3. Our aim now is to evaluate the impact of each parameter on the number of selected variables and classification rates using 10-fold cross validation. For $\lambda_2 = 0.2$ and without the group penalty ($\lambda_3 = 0$), the results in terms of total classification error and cardinality of the support are reported in Figure 5(a), for several values of λ_1. We note that the classification error rate decreases from 35% ($\lambda_1 = 10^{-2}$) to 14% ($\lambda_1 \in [0.18, 0.28]$). Obviously, the performances degrade drastically for values of λ_1 beyond 0.3 which correspond to less than 20 variables. For $\lambda_2 = 0.5$ and without the sparsity term ($\lambda_1 = 0$), the results are shown on Figure 5(b), for different values of the grouping parameter λ_3. We observe that the total classification error rate is under 15% for $\lambda_3 \in [0.1, 1.5]$. In particular the value of λ_3 yielding the lowest error rate (12.2%) is shown in Table 3 with 32 spectral bands. It is worth to notice from these two experiments that both sparsity and grouping parameters act directly on the number of selected variables but not with the same intensity: the sparsity parameter has stronger influence than the grouping parameter. For

instance, to obtain less than 80 variables, the grouping parameter should be set to 0.2 (and $\lambda_1 = 0$) while the same number of variables is obtained for $\lambda_1 \approx 0.06$ (and $\lambda_3 = 0$). To analyse the effect of the fusion parameter on the general classification performances, we set the sparsity parameter λ_1 to 0 and vary both the grouping and the fusion parameters such that 40 variables are retained. The results are reported on Figure 5(c). The error rate is less than 15% in the range $\lambda_2 \in [0.1, 0.6]$. The minimum value of the classification error rate is 11.6%; it corresponds to $\lambda_2 = 0.55$.

5. Discussion and perspectives

In this work we proposed a variable selection technique based on simultaneous and regularized sparse approximation inspired by the fused Lasso methodology. Given several NIR spectra properly ordered in a data matrix, variable selection is formulated as an optimization problem allowing to achieve a trade-off between data approximation and model parsimony including group (simultaneous) and structural (fused) sparsities. The data are modeled as linear combinations of a family of Gaussian functions and a sparse coefficient matrix in order to achieve a good reconstruction of the NIR spectra using only a few elementary functions. Thanks to the simultaneous sparsity, the selected functions are those that show maximum correlation with all measurements. To properly account for data ordering, the fused penalty enforces successive regression coefficients associated to the same selected variable and corresponding to consecutive spectra to be similar. Overall, the goal is to achieve dimensionality reduction that takes into account the global shape of the measurements in order to control the computational cost and avoid overfitting.

Using a FISTA iteration, we have shown that the optimization problem may be solved efficiently thanks to the fused Lasso signal approximator (FLSA) applied on each row of the coefficient matrix. We also present a non-negative version of the algorithm. Application to real NIR spectra has shown that the proposed algorithms are able to select wavebands leading to better classification rates as compared to competitors such as random forest [5], wavelet-based Bayesian variable selection method [6], and simultaneous variable selection [26].

The proposed methodology provides a comprehensive way to encode feature selection in the objective function. With the exception of non-negative FSGL, the resulting algorithms have a low computational cost suitable for large-scale problems. Using FSGL in practice requires the specification of three parameters, namely λ_1, λ_2 and λ_3. Parameters λ_1 and λ_3 almost play the same role. Both act

directly on the sparsity of the regression matrix and thus on the number of selected variables. According to our experience, if the objective is to select a fixed number of variables, it is more convenient to set λ_1 to 0 and tune λ_3 because the latter enforces row sparsity which implies the selection of only one additional variable at a time when λ_3 is decreased continuously. Our strategy here was to choose $\lambda_1 = \lambda_2 = 0$ and to tune λ_3 to reach 32 variables. Thereafter, the parameter λ_2 is progressively increased while λ_3 is decreased to maintain the same number of variables. Through this trial and error approach, we selected the set of parameters leading to the maximum accuracy with $\lambda_1 = 0$. Using cross-validation to tune λ_2 and λ_3 is obviously more efficient than the traditional trial and error. To decrease the computational burden, one can define a grid for λ_3 around the value yielding the desired number of variables. It should be pointed out that the proposed approaches remain relatively easy to parameterize as compared to other Bayesian-based methods such as [6, 41], while ensuring great flexibility according to application requirements (number of variables, reconstruction accuracy or discrimination level).

This work can be extended in several directions. From an application point of view, we are considering the implementation of an optical sorting system on industrial hyperspectral imagers. We also plan to conduct further analysis of FSGL with comparisons to state-of-the-art methods using other datasets. To progress in this direction, we just acquired a new NIR spectrometer to study the performances of the algorithm as a function of noise level, spectral resolution, sorting speed, and so on. From a methodological perspective, we plan to develop dictionary learning techniques to avoid specifying *a priori* the Gaussian-shaped functions over a grid of discrete values.

References

[1] B. Stuart, Infrared spectroscopy, Wiley Online Library, New York, USA, 2005.

[2] H. W. Siesler, Y. Ozaki, S. Kawata, H. M. Heise, Near-infrared spectroscopy: Principles, instruments, applications, Wiley-VCH, Weinheim, Germany, 2002.

[3] M. J. Adams, Chemometrics in analytical spectroscopy, Royal Society of Chemistry, Cambridge, UK, 1995.

[4] P. Jonsson, S. J. Bruce, T. Moritz, J. Trygg, M. Sjöström, R. Plumb, J. Granger, E. Maibaum, J. K. Nicholson, E. Holmes, H. Antti, Extraction, interpretation and validation of information for comparing samples in metabolic LC/MS data sets, Analyst 130 (5) (2005) 701–707.

[5] L. Breiman, Random forests, Machine Learning 45 (1) (2001) 5–32.

[6] M. Vannucci, N. Sha, P. Brown, NIR and mass spectra classification: Bayesian methods for wavelet-based feature selection, Chemometrics and Intelligent Laboratory Systems 77 (1-2) (2005) 139–148.

[7] G. M. Furnival, R. W. Wilson, Regressions by leaps and bounds, Technometrics 42 (1) (2000) 69–79.

[8] A. J. Miller, Subset selection in regression, Chapman and Hall, London, London, England, 1990.

[9] T. Marill, D. Green, On the effectiveness of receptors in recognition systems, IEEE transactions on Information Theory 9 (1) (1963) 11–17.

[10] Z. Xiaobo, Z. Jiewen, M. J. Povey, M. Holmes, M. Hanpin, Variables selection methods in near-infrared spectroscopy, Analytica Chimica Acta 667 (1) (2010) 14–32.

[11] R. Tibshirani, Regression shrinkage and selection via the Lasso, Journal of the Royal Statistical Society. Series B (Methodological) (1996) 267–288.

[12] B. Malli, T. Natschläger, Fused stagewise regression – A waveband selection algorithm for spectroscopy, Chemometrics and Intelligent Laboratory Systems 149 (2015) 53–65.

[13] D. Donoho, Compressed sensing, IEEE Transactions on Information Theory 52 (2006) 1289–1306.

[14] E. J. Candès, J. Romberg, T. Tao, Stable signal recovery for incomplete and inaccurate measurements, Communication on Pure and Applied Mathematics 59 (2006) 1207–1223.

[15] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2) (2009) 210–227.

[16] K. Huang, S. Aviyente, Sparse representation for signal classification, in: Advances in Neural Information Processing Systems, 2006, pp. 609–616.

[17] J. Kim, H. Park, Sparse nonnegative matrix factorization for clustering, Tech. rep., Georgia Institute of Technology (2008).

[18] M. D. Iordache, J. Bioucas-Dias, A. Plaza, Sparse unmixing of hyperspectral data, IEEE Transactions on Geoscience and Remote Sensing 49 (2011) 2014–2039.

[19] Y. Chen, N. M. Nasrabadi, T. D. Tran, Hyperspectral image classification using dictionary-based sparse representation, IEEE Transactions on Geoscience and Remote Sensing 49 (10) (2011) 3973–3985.

[20] Z. Wang, R. Zhu, K. Fukui, J.-H. Xue, Cone-based joint sparse modelling for hyperspectral image classification, Signal Processing 144 (2018) 417–429.

[21] M. Yuan, Y. Lin, Model selection and estimation in regression with grouped variables, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68 (1) (2006) 49–67.

[22] J. A. Tropp, A. C. Gilbert, M. J. Strauss, Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit, Signal Processing 86 (2006) 572–588.

[23] L. Belmerhnia, E.-H. Djermoune, D. Brie, Greedy methods for simultaneous sparse approximation, in: 22nd European Signal Processing Conference (EUSIPCO), 2014, pp. 1851–1855.

[24] D. Kim, J. P. Haldar, Greedy algorithms for nonnegativity-constrained simultaneous sparse recovery, Signal Processing 125 (2016) 274–289.

[25] J. A. Tropp, Algorithms for simultaneous sparse approximation. Part II: Convex relaxation, Signal Processing 86 (2006) 589–602.

[26] B. A. Turlach, W. N. Venables, S. J. Wright, Simultaneous variable selection, Technometrics 47 (3) (2005) 349–363.

[27] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, K. Knight, Sparsity and smoothness via the fused lasso, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67 (1) (2005) 91–108.

[28] S. F. Cotter, B. D. Rao, K. Engan, K. Kreutz-Delgado, Sparse solutions to linear inverse problems with multiple measurement vectors, IEEE Transactions on Signal Processing 53 (7) (2005) 2477–2488.

[29] J. Chen, X. Huo, Theoretical results on sparse representations of multiple-measurement vectors, IEEE Transactions on Signal Processing 54 (12) (2006) 4634–4643.

[30] J. Friedman, T. Hastie, H. Höfling, R. Tibshirani, Pathwise coordinate optimization, The Annals of Applied Statistics 1 (2) (2007) 302–332.

[31] B. Xin, Y. Kawahara, Y. Wang, W. Gao, Efficient generalized fused lasso and its application to the diagnosis of Alzheimer's disease, in: Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014, pp. 2163–2169.

[32] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM Journal on Imaging Sciences 2 (1) (2009) 183–202.

[33] H. Hoefling, A path algorithm for the fused lasso signal approximator, Journal of Computational and Graphical Statistics 19 (4) (2010) 984–1006.

[34] J. Liu, L. Yuan, J. Ye, An efficient algorithm for a class of fused lasso problems, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2010, pp. 323–332.

[35] J. Zhou, J. Liu, V. A. Narayan, J. Ye, Modeling disease progression via fused sparse group lasso, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2012, pp. 1095–1103.

[36] J. Nocedal, S. J. Wright, Numerical optimization, 2nd Edition, Springer Series on Operation Research and Financial Engineering, New York, USA, 2006.

[37] V. Mazet, C. Carteret, D. Brie, J. Idier, B. Humbert, Background removal from spectra by designing and minimising a non-quadratic cost function, Chemometrics and Intelligent Laboratory Systems 76 (2) (2005) 121–133.

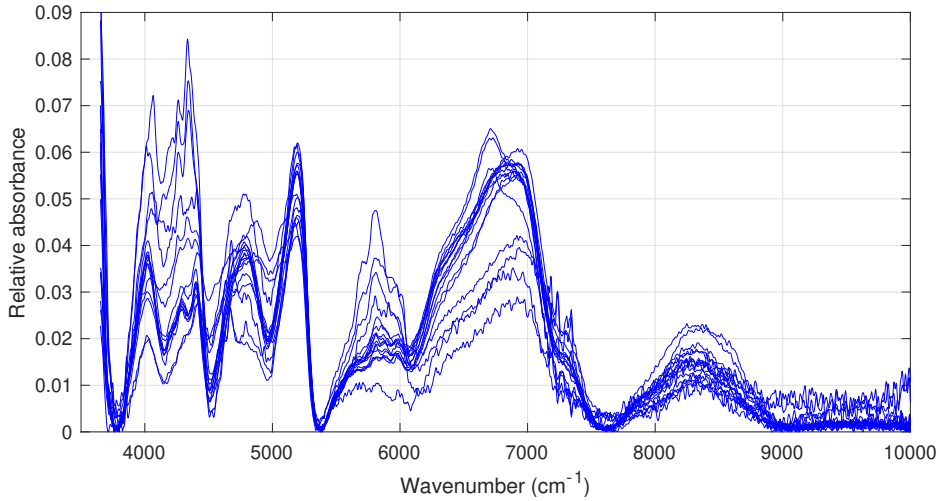
[38] C. Krongtaew, K. Messner, T. Ters, K. Fackler, Characterization of key parameters for biotechnological lignocellulose conversion assessed by FT-NIR

spectroscopy. Part I: Qualitative analysis of pretreated straw, BioResources 5 (4) (2010) 2063–2080.

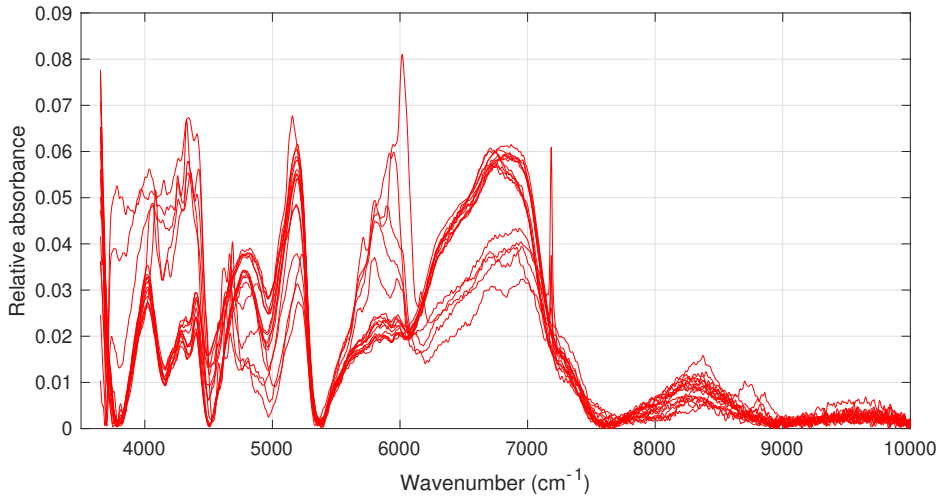
[39] M. Schwanninger, J. Rodrigues, K. Fackler, A review of band assignments in near infrared spectra of wood and wood components, Journal of Near Infrared Spectroscopy 19 (2011) 287–308.

[40] R. Flamary, N. Jrad, R. Phlypo, M. Congedo, A. Rakotomamonjy, Mixed-norm regularization for brain decoding, Computational and Mathematical Methods in Medicine 2014 (2014) ID 317056.

[41] N. Sha, M. Vannucci, M. Tadesse, P. Brown, I. Dragoni, N. Davies, T. Roberts, A. Contestabile, M. Salmon, C. Buckley, F. Falciani, Bayesian variable selection in multinomial probit models to identify molecular signatures of disease stage, Biometrics 60 (3) (2004) 812–819.

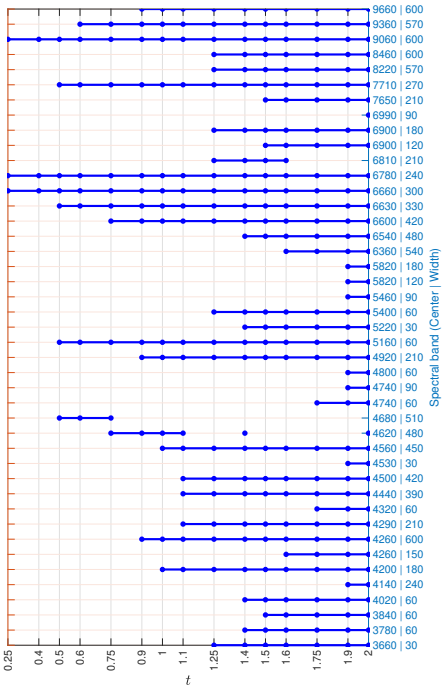


(a) Class 0: recyclable

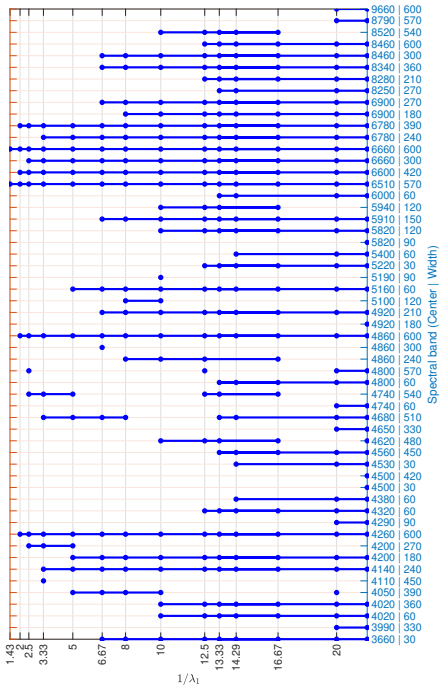


(b) Class 1: non-recyclable

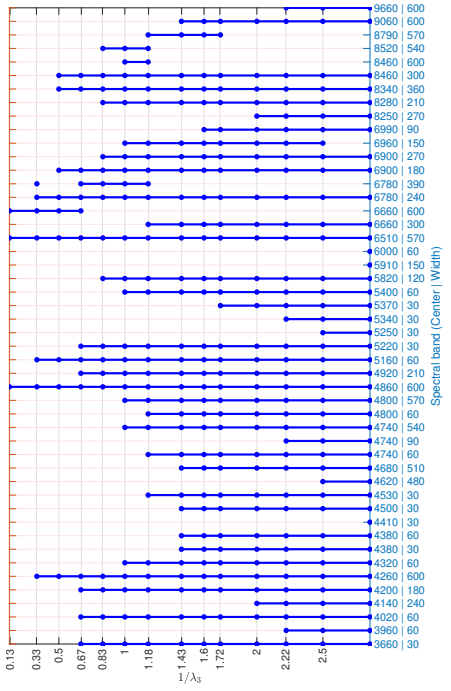
Figure 2: Some (pre-processed) NIR spectra from the two classes of wood wastes.



(a) SVS [26]



(b) FSL ($\lambda_2 = 0.5$)



(c) FSGL ($\lambda_1 = 0$, $\lambda_2 = 0.5$)

Figure 3: Selected variables *versus* tuning parameters.

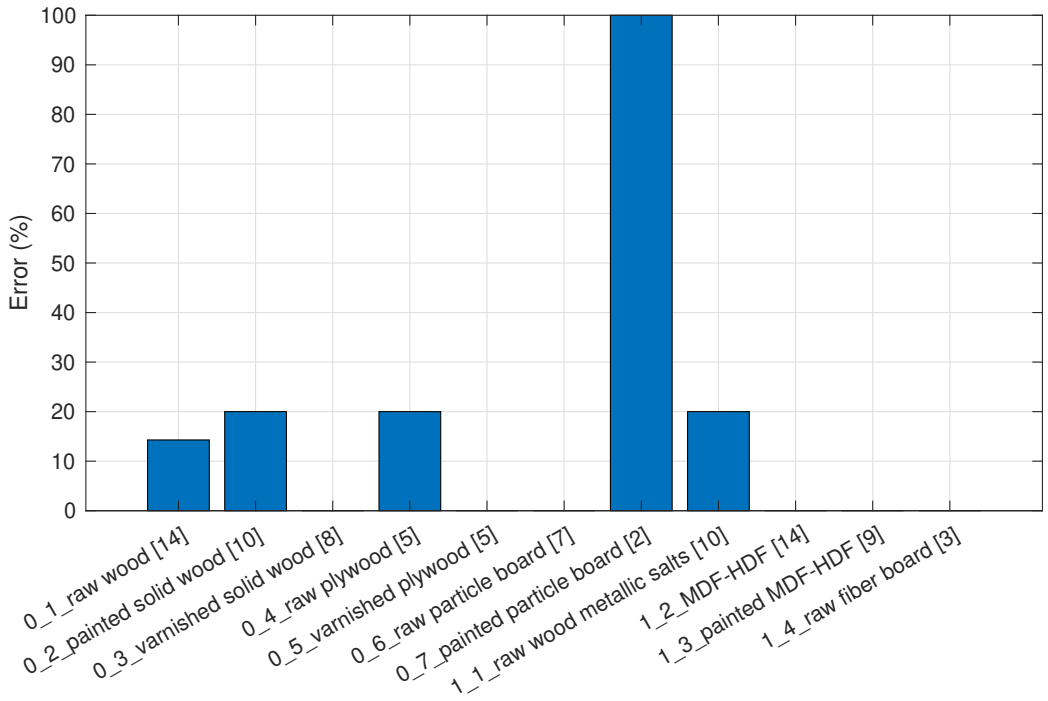
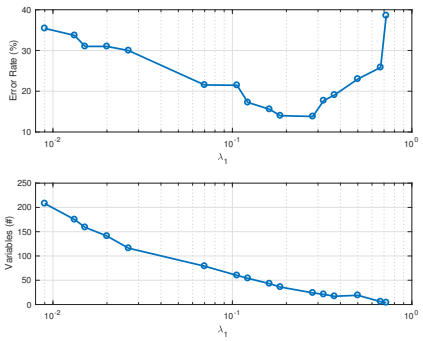
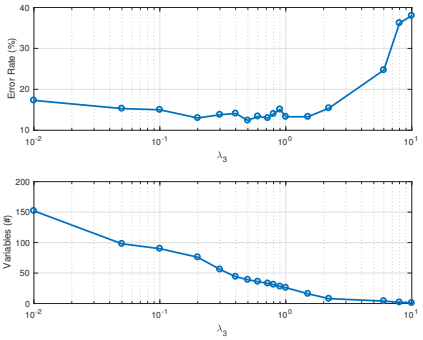


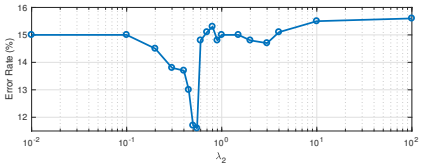
Figure 4: An example of classification error rate in each subclass. The dataset is randomly split into training samples (70%) and test samples (30%).



(a) Sparsity parameter ($\lambda_2 = 0.2, \lambda_3 = 0$)



(b) Grouping parameter ($\lambda_1 = 0, \lambda_2 = 0.5$)



(c) Fusion parameter ($\lambda_1 = 0$ and λ_3 is adjusted to yield 40 variables)

Figure 5: Evolution of the total classification error rate as a function of the regularization parameters.