



Contents lists available at [ScienceDirect](#)

Journal of Computational Mathematics and Data Science

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



Bayesian sparsity and class sparsity priors for dictionary learning and coding



A. Bocchinfuso¹, D. Calvetti, E. Somersalo *

Department of Mathematics, Applied Mathematics, and Statistics Case Western Reserve University, 10900 Euclid Avenue, Cleveland, OH 44106, United States of America

ARTICLE INFO

Keywords:

Bayesian hierarchical models
Sparsity
Krylov subspace methods
Hyperspectral imaging

ABOUT

Dictionary learning methods continue to gain popularity for the solution of challenging inverse problems. In the dictionary learning approach, the computational forward model is replaced by a large dictionary of possible outcomes, and the problem is to identify the dictionary entries that best match the data, akin to traditional query matching in search engines. Sparse coding techniques are used to guarantee that the dictionary matching identifies only few of the dictionary entries, and dictionary compression methods are used to reduce the complexity of the matching problem. In this article, we propose a work flow to facilitate the dictionary matching process. First, the full dictionary is divided into subdictionaries that are separately compressed. The error introduced by the dictionary compression is handled in the Bayesian framework as a modeling error. Furthermore, we propose a new Bayesian data-driven group sparsity coding method to help identify subdictionaries that are not relevant for the dictionary matching. After discarding irrelevant subdictionaries, the dictionary matching is addressed as a deflated problem using sparse coding. The compression and deflation steps can lead to substantial decreases of the computational complexity. The effectiveness of compensating for the dictionary compression error and using the novel group sparsity promotion to deflate the original dictionary are illustrated by applying the methodology to real world problems, the glitch detection in the LIGO experiment and hyperspectral remote sensing.

1. Introduction

Model-based interpretation of data is a core task in computational sciences, encompassing various goals like model selection, validation or falsification, estimation of unknown parameters in a model, or solving an inverse problem arising, e.g., in medical imaging. The increase in computational power, and the current paradigm shift towards data-driven methods have given rise to novel ways of explaining data by relating them to precomputed templates, or previously observed controlled data. The range of applications where this process arises, referred to as *dictionary learning* and *dictionary matching*, continues to widen well beyond classical applications to signal analysis [1], image denoising [2,3], medical imaging [4–6], remote sensing [7,8], and astrophysics [9,10]. Dictionary learning and dictionary matching are important steps in artificial intelligence engines [11,12]. The logistic problems with

* Corresponding author.

E-mail address: ej49@case.edu (E. Somersalo).

¹ Current address: Oak Ridge National Laboratory.

pre-trained generative chat robots, which makes use of Natural Language Processing and the ability to organize and process huge collections of language items using the Large Language Models, include effective and computationally efficient ways to organize large dictionaries as well as the ability to encode new items as a clever combination of dictionary items. In this data intensive era, many dictionaries consist of large collections of items, referred to as atoms. In some cases the dictionary entries are known to belong to different clusters, and often several of the atoms share common features, thus making the dictionary somehow redundant.

Most dictionary learning applications can be thought of as either a classification or a coding tasks, and in both cases, sparsity may be a desirable feature of the solution. Sparse dictionary coding has a remarkable similarity with searching for a sparse solution of a linear inverse problem, with the unknown of interest being the coefficient vector of the linear combination of the dictionary atoms. In some applications, it is believed a priori that the signals to be encoded can be explained in terms of a few of the dictionary atoms. This belief may be supported by the nature of the signal and by specific choices of the dictionary. In many signal and image processing applications, a dictionary consisting of discrete cosine or wavelet transform bases may be employed to obtain a sparse representation [11,13], in which case the dictionary learning task is akin to a generalized change of representation basis. In other applications, where the dictionary is a collection of atoms which may share some traits with the signal to be explained in terms of the dictionary, sparse coding may be desirable to simplify the interpretation. A canonical motivational example of sparse dictionary learning is compressed sensing [14], where high dimensional information can be efficiently recovered and retrieved from few linear observations. Whatever the motivation, algorithms for computing sparse solutions of inverse problems are the engine behind sparse dictionary encoding.

The literature on sparse solutions of inverse problems is rather extensive. One popular way to promote sparse solutions is by means of the basis pursuit [15], which penalizes the ℓ_1 -norm of the unknown of interest, a technique referred to as LASSO in the statistics community [16]. Another way to promote sparse solutions is by recasting the inverse problem within a Bayesian framework, and using a hierarchical prior model to express the sparsity belief. It has been shown that for some choices of the hierarchical models, the Maximum A Posteriori solution computed by the Bayesian approach provides a computationally efficient approximation of the ℓ_1 regularized solution [17–19]. A hybrid Bayesian sparsity promoting approach, which eventually switches from a globally convergent sparsity promoting prior to a greedier sparsity promoting model has been recently proposed [20,21].

2. Outline and novelty

In this work, we consider the class of dictionary learning/matching problems where the computational complexity is a challenge. Let the matrix $D \in \mathbb{R}^{n \times p}$ represent an original dictionary consisting of p atoms in \mathbb{R}^n arranged in its columns. The atoms may be earlier observations of a specific quantity, or they may be synthetic outputs of a computational model. In the case where there is large data set of possible outcomes, we have $n \ll p$, that is, the number of atoms generously exceeds the dimensionality of the data. In other cases we may have few atoms of large dimensionality in which case $n > p$. Consider now a new datum $b \in \mathbb{R}^n$ that we want to explain in terms of few of the dictionary atoms. This is equivalent to seeking to solve

$$b = Dx + \varepsilon, \quad (1)$$

where $x \in \mathbb{R}^p$ is sparse with non-negative entries, and ε represents the discrepancy between the datum and what is representable in terms of the dictionary. In other words, the coefficient vector x identifies the few atoms needed to explain the data, and the discrepancy is ideally insignificantly small. If the atoms are outputs of a parametrized computational model, such sparse representation can be seen as a way to identify model parameters that best fit to the new observation. Ideally, only one atom is identified, but more realistically, several candidate model outputs may be singled out.

If the full dataset D is the dictionary, one way to reduce the complexity of the problem and simultaneously avoid overfitting is to compute a compressed dictionary by computing a low rank approximation of D ,

$$D = WH + E, \quad W \in \mathbb{R}^{n \times k}, \quad H \in \mathbb{R}^{k \times p}, \quad k \ll p. \quad (2)$$

It may be desirable to require the entries of the coefficient matrix H to be non-negative, making it easier to interpret them as weights. The columns of the matrix W , referred to as the *code book* in [22], may be required to retain some properties of the original atoms, e.g., non-negativity or sparsity. Several algorithms for the model reduction have been proposed, [22], and often this step is referred to as dictionary learning in the literature. The matrix E accounts for the discrepancy between the original dictionary D and its low-rank approximation WH in terms of the reduced dictionary. After replacing the dictionary with its low rank approximation, expressing the new datum b in terms of the compressed dictionary requires the solution of the linear system

$$b = Wh + \varepsilon', \quad (3)$$

where $h \in \mathbb{R}^k$ is non-negative and possibly sparse, and ε' accounts for the difference between the datum and its approximation in terms of the feature vectors. If the dictionary reduction is performed so that the code book entries are interpretable, the reduced representation may also allow an interpretation for the data b .

In some dictionary applications the atoms are partitioned into disjoint classes and the task is to classify a new datum into one of the classes. A number of algorithms for reducing the computational complexity of dictionary classification have been proposed in the literature, see, e.g., [23,24]. The algorithmic approach that we propose here is naturally suitable for dictionary classification also, which may be seen as an intermediate step to reduce the computational complexity of dictionary matching.

In this article, we propose a multi-phase process of the dictionary learning algorithm, summarized in the following steps.

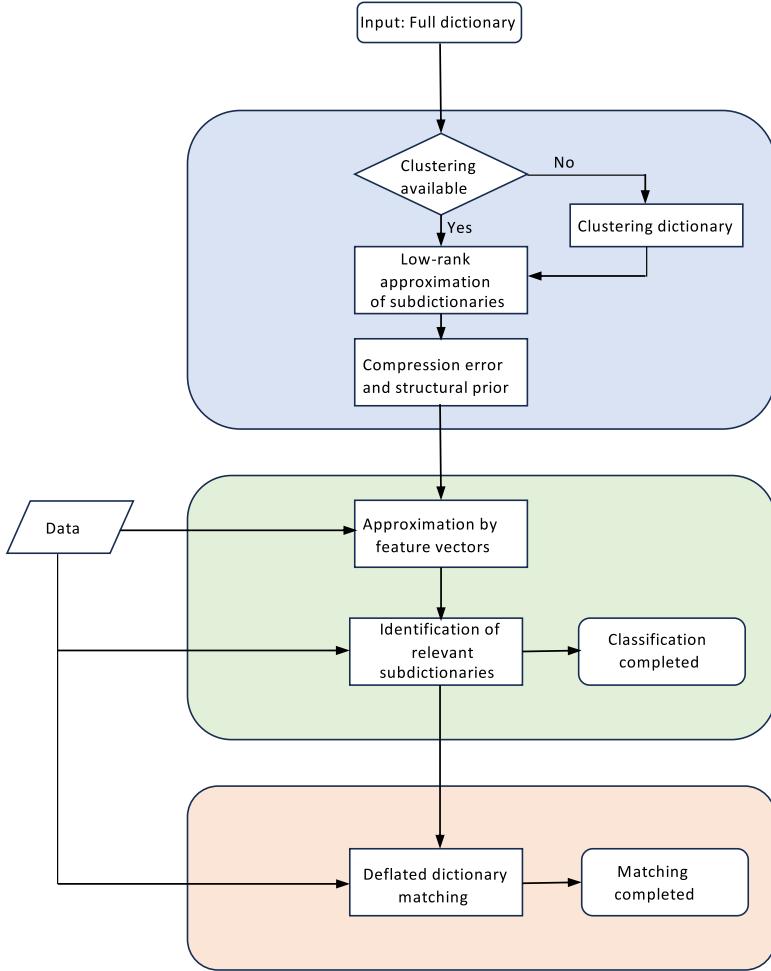


Fig. 1. Overview of the layered dictionary matching algorithm. The lavender box contains the preprocessing of the dictionary, including the (optional) clustering, reduction of the subdictionaries by low-rank matrix factorization, and computation of the compression error mean and covariance as well as the structural prior covariance matrices. The sage box contains the dictionary matching of the data in terms of the reduced dictionaries, including the (optional) classification in terms of the subdictionary classes. Finally, the apricot box contains the dictionary matching of the data in terms of the deflated dictionary, comprising only the relevant subdictionaries identified in the green box.

1. **Clustering:** Partition the original full dictionary into subdictionaries $D^{(j)}$, $1 \leq j \leq K$ using supervised or unsupervised clustering.
2. **Reduction:** Compress each subdictionary into a low rank code book, $D^{(j)} = W^{(j)}H^{(j)} + E^{(j)}$, $1 \leq j \leq K$.
3. **Cluster identification:** Given a new datum $b \in \mathbb{R}^n$, identify reduced subdictionaries that are needed to explain b ,

$$b = W^{(j_1)}h^{(j_1)} + \dots + W^{(j_r)}h^{(j_r)} + \epsilon',$$

4. **Deflation:** Represent b in terms of original subdictionaries $D^{(j)}$ that correspond to the selected compressed dictionaries,

$$b = D^{(j_1)}x^{(j_1)} + \dots + D^{(j_r)}x^{(j_r)} + \epsilon.$$

Depending on the objective, one or more of the phases may be skipped. The flowchart in Fig. 1 provides an overview of how the four different phases can be connected.

The process is carried out in a Bayesian framework. In particular, all steps involving sparse coding are performed by using sparsity promoting Bayesian hypermodels and priorconditioning techniques that have been shown earlier to provide a computationally efficient way to find compressible solutions to linear inverse problems. As a novelty, in the cluster identification problem, we introduce a new, data-informed way to implement group sparsity in order to identify more clearly those clusters that do not contribute to explaining the data. Moreover, our approach borrows ideas from the previous works on Bayesian modeling error analysis to quantify the modeling error $E^{(j)}$ when passing from the full dictionary cluster to the reduced one. We analyze the approximation errors ϵ and ϵ' using the Bayesian modeling error methods, and demonstrate that in particular the covariance

structure of the error induced by the dictionary reduction may have an impact on the quality of the representation. The algorithm is tested with several computed examples.

3. Dictionary learning/matching: A layered approach

In this section, we discuss the four steps outlined in the previous section, leaving some of the details for later discussion. The main computational workhorse is a Bayesian hierarchical solver for linear equations with sparsity and possibly positivity constraints. The solver, referred to as the Iterative Alternating Sequential (IAS) algorithm, will be described in detail in a later section.

3.1. Clustering

In many applications, the atoms of the dictionary D may originate from different sources, or be associated to different models, thus suggesting a natural way of dividing them into subgroups. Even when there are no obvious reasons to partitions the atoms into subgroups, there may be a latent clustered structure in the dictionary which can be highlighted with the help of unsupervised clustering techniques, e.g., k -means or k -medoids [25]. In the following, we assume that the dictionary atoms can be partitioned into K clusters, and that the dictionary atoms are ordered so that atoms in the same cluster are stored in adjacent columns. This yields a block partitioning of the matrix representation of the dictionary of the form

$$D = [\ D^{(1)} \ | \ \dots \ | \ D^{(K)} \],$$

where $D^{(j)} \in \mathbb{R}^{n \times p_j}$, $p_1 + \dots + p_K = p$. We remark that, while for some classification problems the annotation of the atoms is provided, in dictionary matching the original annotation of the atoms may not reflect objective similarity between atoms within a class, but may be based on non data-driven criteria. In general, unsupervised clustering may be advisable from the point of view of dividing the dictionary in intrinsically similar batches even if no particular reason for a cluster structure in the data were known. We will not address here how to determine the number K of classes in unsupervised clustering, referring instead to the pertinent literature [25].

3.2. Low rank approximation

A natural way to reduce the complexity of dictionary learning/matching tasks is to replace each subdictionary $D^{(j)}$ by a low rank approximation of the form

$$D^{(j)} \approx W^{(j)}H^{(j)}, \quad W^{(j)} \in \mathbb{R}^{n \times k_j}, \quad H^{(j)} \in \mathbb{R}^{k_j \times p_j}, \quad k_j < p_j.$$

The columns of $W^{(j)}$ are often called the *feature vectors* of the subdictionary $D^{(j)}$, because the atoms in $D^{(j)}$ can be approximated by their linear combinations and often the entries of $H^{(j)}$ are all nonnegative.

Computing a low rank approximation of a matrix is often an intermediate step in a complex task. Among several possible way to compute a low rank approximation of a matrix, the most popular approach, known a principal component approximation (PCA), is based on a truncated singular value decomposition. Given the singular value decomposition

$$D^{(j)} = U^{(j)}\Sigma^{(j)}[V^{(j)}]^\top,$$

of $D^{(j)}$, let $W^{(j)}$ be the matrix consisting of the first k_j columns of $U^{(j)}$, and $H^{(j)}$ comprise the first k_j rows of $[V^{(j)}]^\top$ scaled by the first k_j singular values,

$$W^{(j)} = U_{k_j}^{(j)} = U^{(j)}(:, 1:k_j), \quad H^{(j)} = \Sigma_{k_j}^{(j)}[V_{k_j}^{(j)}]^\top = \Sigma^{(j)}(1:k_j, 1:k_j)[V^{(j)}(:, 1:k_j)]^\top.$$

It follows from the well known Eckart–Young–Mirsky theorem that the PCA is the optimal low rank approximation in the sense that it minimizes the Frobenius norm as well as the nuclear norm of the discrepancy. Denoting the singular values by σ_ℓ , $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \geq 0$ with $r = \min\{n, p_j\}$, an upper bound for the approximation error in terms of the Frobenius norm is given by

$$\|D^{(j)} - W^{(j)}H^{(j)}\|_F^2 = \sum_{\ell=k_j+1}^r \sigma_\ell^2 \leq \sigma_{k_j+1}^2(r - k_j).$$

Besides the optimality, the low rank approximation by PCA is attractive because of the availability of standard robust algorithms. On the other hand the optimality leaves no room for requiring further properties such as non-negativity of the entries of $H^{(j)}$.

When a matrix has all nonnegative entries, it is possible to compute its approximate factorization into the product of two low rank matrices with nonnegative entries, known as nonnegative matrix factorization (NMF) [26]. The nonnegativity of the feature vectors may simplify their interpretation, while the nonnegativity of the weights makes it easier to assess the importance of each feature vector for explaining the columns of the original matrix. The most popular algorithms for computing a rank- k nonnegative matrix factorization of a nonnegative matrix proceed in an iterative fashion, updating one of the factors at a time, while keeping the other fixed. More specifically, a non-negative matrix factorization of a matrix $D^{(j)} \geq 0$ seeks to determine $W^{(j)} \geq 0$, $H^{(j)} \geq 0$ that minimize the approximation error,

$$d(D^{(j)}, W^{(j)}, H^{(j)}) = \|D^{(j)} - W^{(j)}H^{(j)}\|_F, \quad W^{(j)} \in \mathbb{R}^{n \times k_j}, \quad H^{(j)} \in \mathbb{R}^{k_j \times p_j}, \quad (4)$$

or some other divergence measure. Commonly used algorithms include alternating multiplicative updating schemes, as well as non-negative least squares (NNLS) schemes: If $W^{(j)}$ is given, the entries of the ℓ th column of the matrix $H^{(j)}$ are the coefficients of the expansions of the ℓ th column of $D^{(j)}$ in terms of feature vectors, stored as columns of $W^{(j)}$, thus $H^{(j)}$ can be updated column-wise by solving

$$h^{(\ell)} = \underset{h \in \mathbb{R}^{k_j}, h \geq 0}{\operatorname{argmin}} \|d^{(\ell)} - W^{(j)}h\|_2.$$

Conversely, given $H^{(j)} \in \mathbb{R}^{k_j \times p_j}$, from the observation that

$$\|D^{(j)} - W^{(j)}H^{(j)}\|_F = \left\| (D^{(j)})^T - (H^{(j)})^T (W^{(j)})^T \right\|_F,$$

it follows that we can update $W^{(j)}$ row-wise by similarly solving a least squares problem, see [25,26] for details. Compared to the PCA approach, convergence criteria and approximation properties of the factorizations are less well developed, however, the algorithms allow the introduction of extra conditions such as sparsity, and the interpretability of the results is usually better.

Whichever low rank approximation scheme is chosen, we may write

$$D^{(j)} = W^{(j)}H^{(j)} + E^{(j)}, \quad 1 \leq j \leq K,$$

where the matrices $E^{(j)}$ represent the approximation errors. Replacing each subdictionary of D by its low rank approximation yields

$$\begin{aligned} [D^{(1)} \mid \dots \mid D^{(K)}] &= [W^{(1)}H^{(1)} + E^{(1)} \mid \dots \mid W^{(K)}H^{(K)} + E^{(K)}] \\ &= [W^{(1)} \mid \dots \mid W^{(K)}] \begin{bmatrix} H^{(1)} & 0 & \dots & 0 \\ 0 & H^{(2)} & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \dots & 0 & H^{(K)} \end{bmatrix} + [E^{(1)} \mid \dots \mid E^{(K)}] \\ &= WH + E, \end{aligned}$$

E representing the total approximation error. The product WH is the induced low rank approximation of D , whose feature vectors are inherited from the feature vectors of the subdictionaries. In the applications of interest in this paper, the matrix W can be viewed as a compression of the dictionary D , thus we will refer to it as *compressed dictionary*. The number of atoms in the compressed dictionary is the sum of the number of feature vectors of the subdictionaries, which may be chosen to account for the effective dimensionality of the subdictionaries, as will be illustrated in the computed examples section. We remark that, in general, the low rank approximation induced by the partitioning of the dictionary may differ from what the low rank approximation algorithm applied to the whole dictionary would yield. The matrix E accounts for the error introduced by the reduction of the dimensionality of the dictionary, effectively quantifying how much discrepancy can be expected between the full dictionary and its reduced form.

When representing a given datum b in terms of the feature vectors constituting the columns of W ,

$$b = Wh + \epsilon', \quad (5)$$

the approximation error ϵ' can be thought of as a realization of the random variable with a distribution that can be approximated by considering the sample

$$E = D - WH.$$

Before moving ahead with the algorithm, we consider the effect of the reduction error through a Bayesian perspective.

3.2.1. Dictionary compression error

In the following, let $D \in \mathbb{R}^{n \times p}$ denote either the full dictionary, a single subdictionary $D^{(j)}$, or, a subset of i subdictionaries combined, $[D^{(j_1)}, \dots, D^{(j_i)}]$. Furthermore, we denote by W and H the corresponding rank- k approximate matrix factors. The problem considered here is to approximate a given vector b in terms of W , that is, to find a vector h so that the approximation error in (5) is as small as possible.

Consider the columns $d^{(\ell)} \in \mathbb{R}^n$ of D that allows a perfect representation in terms of the dictionary D , and denote by $h^{(\ell)} \in \mathbb{R}^k$ the corresponding coefficient vector in the matrix H . We have

$$\begin{aligned} d^{(\ell)} &= De_\ell = Wh^{(\ell)} + (De_\ell - Wh^{(\ell)}) \\ &= Wh^{(\ell)} + m^{(\ell)}, \quad 1 \leq \ell \leq p, \end{aligned}$$

where $e_\ell \in \mathbb{R}^p$ is the canonical ℓ th unit vector, and $m^{(\ell)}$ is the error due to compression, referred to as dictionary compression error (DCE). Adhering to a probabilistic interpretation of $d^{(\ell)}$ as a realization of a random vector arising from some underlying probability density, the distribution of the compression error can be interpreted as a push-forward of this distribution by the mapping $d^{(\ell)} \mapsto m^{(\ell)}$. Observe that while the vectors $h^{(\ell)}$ may be computed by minimizing the unweighted least squares error between $d^{(\ell)}$ and $Wh^{(\ell)}$, there is no guarantee that the discrepancies $m^{(\ell)}$ represent scaled white noise.

Having generated the compression error sample $\{m^{(\ell)}\}_{\ell=1}^p$, we compute the mean and covariance,

$$\mu_{DCE} = \frac{1}{p} \sum_{\ell=1}^p m^{(\ell)},$$

$$C_{DCE} = \frac{1}{p} \sum_{\ell=1}^p (m^{(\ell)} - \mu_{DCE})(m^{(\ell)} - \mu_{DCE})^T + \epsilon I_n,$$

where the term ϵI_n , with ϵ a small positive constant, is added to ensure that the covariance matrix is positive definite. To take the covariance structure into account in dictionary learning, we whiten the noise ϵ' in (5), by writing

$$C_{DCE}^{-1/2}(b - \mu_{DCE}) = C_{DCE}^{-1/2}W\mathbf{h} + w,$$

so that w can be assumed to be mean zero and have covariance close to the identity. This equation gives rise to the *dictionary compression error-enhanced least squares problem*,

$$h = \operatorname{argmin} \left\| C_{DCE}^{-1/2}(b - \mu_{DCE} - W\mathbf{h}) \right\|_2^2, \quad (6)$$

that can be augmented by appropriate sparsity requirements as specified later. We point out the similarity of this approach with the techniques previously used to account for the error due, e.g., to coarse discretization or poorly known model parameters in computational inverse problems, see, [27–29].

3.3. Subdictionary identification

Consider the minimization problem (6), assuming that the compressed dictionary W consists of all subdictionaries $W^{(1)}, \dots, W^{(K)}$. Let a datum b is given, and assume that b is such that it is expressible as a linear combination of few dictionary entries of D . As a first task, we seek to identify few subdictionaries that explains the datum, by seeking to represent b in the form

$$b \approx W^{(1)}h^{(1)} + \dots + W^{(K)}h^{(K)}.$$

Consequently, if the dictionary compression reflects well the structure of the original full dictionary, one could expect that only few of the compressed subdictionaries $W^{(j)}$ should be needed to explain the datum b with reasonable accuracy. To find out which subdictionaries are necessary, we adopt a *group sparsity* prior for the vectors $h^{(j)}$. Assume that for each j , $1 \leq j \leq K$, an appropriate norm $\|\cdot\|_{(j)}$ in \mathbb{R}^{k_j} is defined. The choice of these norms is discussed below. To promote group sparsity, most of the norms $\|h^{(j)}\|_{(j)}$ should vanish or be negligibly small, therefore we want to find h such that

$$h = \operatorname{argmin} \left\{ \left\| C_{DCE}^{-1/2}(b - \mu_{DCE} - W\mathbf{h}) \right\|_2^2 + \lambda \left\| \left(\|h^{(1)}\|_{(1)}, \dots, \|h^{(K)}\|_{(K)} \right) \right\|_0 \right\}, \quad (7)$$

where $\|\cdot\|_0$ indicates the support of the vector and $\lambda > 0$. Equivalently, we seek a solution of (6) such that the cardinality of the vector of the norms of $h^{(j)}$ is as small as possible. The minimization problem (7) is complicated by the fact that the objective function is not convex, and a common practice is to replace it by a convex proxy. In this work, the minimization is performed by using an appropriately defined Bayesian group sparsity promoting algorithm, the details being given later. We discuss next the norms $\|\cdot\|_{(j)}$, which are constructed by using structural information about the reduced subdictionary representations of the dictionary entries.

3.3.1. Structural prior for subcluster coefficients

When solving (6), we expect $h^{(j)}$ to be structurally similar to the column vectors of the matrix $H^{(j)}$ corresponding to the subdictionary $D^{(j)}$. Observe, however, that we cannot expect $h^{(j)}$ to be close to the group mean of the columns of $H^{(j)}$, as we believe a priori that most of the groups $W^{(j)}$ contribute little or not at all to explaining b . Therefore, we postulate a priori that $h^{(j)}$ should point in the general direction of the columns in $H^{(j)}$, with its amplitude controlled by the group sparsity promoting prior. To encode that kind of belief, we consider the singular value decomposition

$$H^{(j)} = \hat{U}^{(j)} \hat{\Sigma}^{(j)} [\hat{V}^{(j)}]^T, \quad \hat{\Sigma}^{(j)} = \operatorname{diag}([\sigma_1^{(j)}, \dots, \sigma_{r_j}^{(j)}, 0, \dots, 0]),$$

and observe that the first columns of $\hat{U}^{(j)}$ identify the directions along which the columns have the largest spread [25], which is also where we would expect $h^{(j)}$ to have the largest orthogonal projection, regardless of its norm. In light of this observation, we consider the symmetric positive definite matrix

$$\begin{aligned} C^{(j)} &= \left(\frac{1}{\sigma_1^{(j)}} \right)^2 \hat{U}^{(j)} \hat{\Sigma}^{(j)} [\hat{\Sigma}^{(j)}]^T [\hat{U}^{(j)}]^T + \epsilon^{(j)} I \\ &= u_1^{(j)} [u_1^{(j)}]^T + \sum_{\ell=2}^{r_j} \left(\frac{\sigma_{\ell}^{(j)}}{\sigma_1^{(j)}} \right)^2 u_{\ell}^{(j)} [u_{\ell}^{(j)}]^T + \epsilon^{(j)} I, \end{aligned}$$

where r_j is the rank of $H^{(j)}$ and $\epsilon^{(j)} > 0$ is a small constant ensuring the positive definiteness of the matrix. We assume a priori that the *direction vector* of unknown $h^{(j)}$, denoted by $\hat{h}^{(j)}$ follows the angular central Gaussian measure with covariance $C^{(j)}$, defined as the pullback of the Gaussian measure $\mathcal{N}(0, C^{(j)})$ with respect to the radial projection $\mathbb{R}^{k_j} \rightarrow \mathbb{S}^{k_j-1}$. Alternatively, we may express

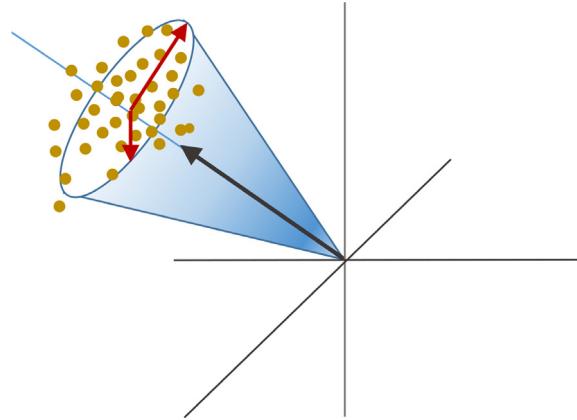


Fig. 2. Schematic picture of the structural prior. The dots indicate the column vectors of the matrix $H^{(j)}$. The black arrow is the direction $u_1^{(j)}$ of maximum spread, which in this case is close to the mean of the column vectors, while the red arrows indicate vectors $u_\ell^{(j)}$, $\ell \geq 2$, scaled by the ratios $\sigma_\ell^{(j)}/\sigma_1^{(j)}$. The prior expresses the belief that the vector $h^{(j)}$ has a direction close to $u_1^{(j)}$, and the width and shape of the cone of high probability is determined by the sizes of the singular values relative to the largest one.

this prior in terms of the conditional distribution, given the variance of the length of the vector, as

$$\pi^{(j)}(h^{(j)} | \theta_j) \propto \exp\left(-\frac{\|h^{(j)}\|_{(j)}^2}{2\theta_j}\right),$$

where the structural norm is defined as

$$\|h^{(j)}\|_{(j)}^2 = [h^{(j)}]^\top [C^{(j)}]^{-1} h^{(j)}. \quad (8)$$

This is equivalent to expressing the belief that $h^{(j)}$ has a component along $u_1^{(j)}$ with length regulated by the variance parameter θ_j , and it should remain within the cone defined by the columns of $H^{(j)}$ with high probability, see Fig. 2. The selection of the parameter θ_j is part of the group sparsity problem and will be addressed in the sequel.

3.3.2. Thresholding

When solving the dictionary representation problem with group sparsity promoting prior, one cannot expect that the norms $\|h^{(j)}\|_{(j)}$ defined by (8) for the irrelevant subdictionaries to vanish, but rather we expect some of them to be under a threshold value, similarly to compressibility of a signal in compressed sensing applications. To establish whether a coefficient vector $h^{(j)}$ is significantly large, consider the dictionary compression error covariance $C_{DCE}^{(j)}$ computed by using the dictionary entries of the j th subdictionary $D^{(j)}$ alone. We say that the coefficient vector $h^{(j)} \in \mathbb{R}^{k_j}$ is significant if and only if its contribution to explain the datum b is above the standard deviation of the compression error, that is

$$\|W^{(j)}h^{(j)}\|^2 > \text{trace}(C_{DCE}^{(j)}).$$

Denote by $J = \{j_1, \dots, j_i\} \subset \{1, 2, \dots, K\}$ the set of the indices of the relevant subdictionaries.

3.4. Solving the deflated problem

The final step in the suggested algorithm is to identify the atoms in the deflated dictionary $D^{(J)} = [D^{(j_1)}, \dots, D^{(j_i)}]$. To this end, we compute first the dictionary compression error mean and covariance corresponding to the subdictionary $D^{(J)}$, denoted by $\mu_{DCE}^{(J)}$ and $C_{DCE}^{(J)}$. The final solution is the DCE-enhanced solution, augmented with a sparsity promotion.

3.5. Bayesian hierarchical linear solvers

Underdetermined linear systems arise in many important inverse problems applications, requiring to address how to handle contributions from the null space. However, the presence of a significant null space adds flexibility to the solution, making it possible to add additional constraints and objectives such as sparsity and positivity. In this section, we briefly review a sparsity promoting Bayesian hierarchical model that will be used to solve the subproblems 2, 3 and 4 of the proposed layered approach. For further details, we refer to [19].

Consider the linear system of equations

$$b = Ax + e, \quad A \in \mathbb{R}^{n \times p} \quad (9)$$

where the additive error e is assumed to be zero mean Gaussian distribution with covariance matrix Γ , thus yielding the likelihood

$$\pi_{B|X}(b | x) \propto \exp\left(-\frac{1}{2}(b - Ax)^T \Gamma^{-1}(b - Ax)\right).$$

We encode our a priori belief that the solution x will have only very few nonzero entries, or entries above a small threshold value, by modeling x as zero mean Gaussian random variable X with a diagonal covariance matrix,

$$X \sim \mathcal{N}(0, D_\theta), \quad D_\theta = \text{diag}(\theta_1, \dots, \theta_p), \quad \theta_j > 0.$$

In the hierarchical approach, we assume further that the variance vector θ is unknown, and model it as a random variable Θ . Hence, the conditional prior for X is

$$\pi_{X|\Theta}(x | \theta) \propto \frac{1}{\sqrt{\det(D_\theta)}} \exp\left(-\frac{1}{2} \|D_\theta^{-\frac{1}{2}} x\|^2\right) = \exp\left(-\frac{1}{2} \sum_{j=1}^p \log \theta_j - \frac{1}{2} \|D_\theta^{-\frac{1}{2}} x\|^2\right).$$

To model the variance vector θ , observe first that the smaller the variance, the more likely it is that the variable X takes on a value close to zero, therefore if the support of the solution were known, it would suffice to make the corresponding diagonal entries large, while setting all others very close to zero. In line with the Bayesian philosophy, since this information is not known a priori, the variances are modeled as random variables. Furthermore, the variances should be independent of each other, positive and, to promote sparsity, most of them should be very small while a few of them could be large. To achieve such outcome, we assume that the variables θ_j are independent and distributed according to a fat-tailed distribution such as the gamma distribution,

$$\pi_\Theta(\theta | \beta, \vartheta) = \prod_{j=1}^p \frac{1}{\Gamma(\beta)} \theta_j^{\beta-1} \exp\left(-\frac{\theta_j}{\vartheta_j}\right),$$

where the value of the shape parameter β is related to the level of sparsity promotion and the values of the scale parameters ϑ_j are related the sensitivity of b to the entries of x ; see [19] for details. Without loss of generality, we may assume that the additive error in the right hand side of (9) is zero mean white Gaussian. Combining the likelihood and the joint prior according to Bayes' formula yields the posterior density for the pair (X, Θ)

$$\pi_{X,\Theta|B}(x, \theta | b) \propto \exp\left(-\frac{1}{2} \|b - Ax\|^2 - \frac{1}{2} \|D_\theta^{-\frac{1}{2}} x\|^2 + \left(\beta - \frac{3}{2}\right) \sum_{j=1}^p \log \theta_j - \sum_{j=1}^p \frac{\theta_j}{\vartheta_j}\right),$$

which in the Bayesian framework is the solution of the inverse problem. While getting insight in the properties of the posterior distribution requires methods like Markov chain Monte Carlo (MCMC) sampling [30], we restrict ourselves here to numerically approximate the Maximum a Posterior (MAP) estimate, which is also the minimizer of the Gibbs energy,

$$\mathcal{E}(x, \theta) = \frac{1}{2} \|b - Ax\|^2 + \frac{1}{2} \sum_{j=1}^p \frac{x_j^2}{\theta_j} - \left(\beta - \frac{3}{2}\right) \sum_{j=1}^p \log \theta_j + \sum_{j=1}^p \frac{\theta_j}{\vartheta_j}.$$

A computationally efficient way of approximating the MAP estimate, in particular for large matrices A , is provided by the Iterative Alternative Sequential (IAS) algorithms, discussed in detail in [19]. The algorithm proceeds by updating sequentially x and θ , taking advantage of the fact that the update of x amounts to the solution of a linear least squares problem, and the update of $\theta_1, \dots, \theta_p$ can be done independently via a simple formula evaluation. Moreover, the computationally efficient version of the IAS uses the technique of priorconditioning together with Krylov subspace iterative schemes with early stopping. The outline of the algorithm is summarized below, with the details omitted. We point out that the algorithm can be made even more efficient by combining gamma distributed hyperpriors with generalized gamma distributions in a hybrid manner, see [20,30,31]. The hybrid algorithm will be used in the last of the computed examples to enhance the sparsity promotion. Rather than reviewing details that are not central for this article, we refer interested readers to the cited articles.

IAS algorithm

Given: matrix $A \in \mathbb{R}^{n \times p}$, $b \in \mathbb{R}^n$, β, ϑ_j , $1 \leq j \leq p$,

Initialize: $\theta_j^{(0)} = \vartheta_j$, $1 \leq j \leq p$, set the counter $k = 0$.

Iterate until stopping criterion is met:

1. Update x : Set $\theta_j = \theta_j^{(k)}$, $1 \leq j \leq p$ and set

$$x^{(k+1)} = \arg \min \left\{ \|b - Ax\|_2^2 + \|D_{\theta^{(k)}}^{-\frac{1}{2}} x\|_2^2 \right\}.$$

2. Update θ_j , $1 \leq j \leq p$ by solving

$$\frac{\partial \mathcal{E}}{\partial \theta_j}(x^{(k+1)}, \theta) = 0,$$

yielding

$$\theta_j^{(k+1)} = \frac{\vartheta_j}{2} \left(\eta + \sqrt{\eta^2 + \frac{[x_j^{(k+1)}]^2}{4\vartheta_j}} \right), \quad \eta = \beta - \frac{3}{2}.$$

3. If $\|\theta^{(k)} - \theta^{(k+1)}\| \leq \delta \|\theta^{(k)}\|$ for a given tolerance $0 < \delta < 1$, stop, otherwise advance the counter $k \rightarrow k + 1$ and continue.

The properties of the IAS algorithm, and its generalizations using generalized gamma distributions have been reported in the literature. When using the gamma distribution for the variances, the functional minimized by the IAS algorithm is convex with a unique global minimum, and the IAS iterates converge to the unique minimizer. It has been shown that, in the limit as $\eta = \beta - 3/2 \rightarrow 0^+$, the solution computed with the IAS algorithm converges to a scaled ℓ_1 -regularized solution,

$$x_{\ell_1} = \operatorname{argmin} \left\{ \frac{1}{2} \|b - Ax\|^2 + \sqrt{2} \sum_{j=1}^p \frac{|x_j|}{\sqrt{\vartheta_j}} \right\},$$

suggesting that the selection of the value of the parameter β depends on how sparse the solution is believed to be. In general, the value of $\eta = \beta - \frac{3}{2} > 0$ should be very close to zero if the solution x is believed to be very sparse, while with larger values of η the solution computed by the IAS will be closer to an ℓ_2 -regularized solution.

Moreover, the above formula indicates that the variables ϑ_j provide a component-wise scaling, and it was demonstrated that under the hypothesis that all components x_j must have the same chance to explain the data within a given signal-to-noise ratio, the proper choice of the scaling parameters is related to the sensitivity of the data on the components x_j , see [18,19] for details. Moreover, if the entries of the solution are subject to bound constraints, e.g., $x_j \geq 0$, the IAS algorithm can be modified so as to ensure that the constraints are respected by concatenating the IAS iterations with a projection onto the feasible set to the update of x , see [21].

In the present dictionary learning context, we employ the IAS algorithm in the NNLS-NMF algorithm to update the columns of H and rows of W , with the projection on the nonnegative cone guaranteeing the nonnegativity of the entries of H , see [24] for details. If each dictionary entry is to be explained in terms of few feature vectors, a very small value of $\eta > 0$ is used, while a larger values of η is used for the computation of feature vectors that are not expected to be sparse. Observe that the positivity constraint for W can also be applied if the feature vectors are required to be non-negative to allow interpretation in terms of the dictionary. Furthermore, the IAS algorithm will be employed also in the phase of solving the deflated problem with $A = D^{(J)}$, the group reduced full dictionary, where the sparsity of the coefficient vector h is crucial.

3.6. IAS with group sparsity

We consider now the step of identifying relevant clusters by using the reduced subdictionaries, and in particular, the minimization problem (7) by using the IAS algorithm.

The goal here is to find an approximate representation

$$b = Wh + \epsilon',$$

where

$$W = [W^{(1)} \mid \dots \mid W^{(K)}], \quad h = \begin{bmatrix} h^{(1)} \\ \vdots \\ h^{(K)} \end{bmatrix}, \quad h^{(j)} \in \mathbb{R}^{k_j},$$

with the prior belief that most of the vectors $h^{(j)}$ are insignificant.

Define a group sparsity promoting structural prior for h ,

$$\begin{aligned} \pi_{H|\theta}(h \mid \theta) &\propto \prod_{j=1}^K \frac{1}{\sqrt{\det(\theta_j C^{(j)})}} \exp \left(-\sum_{j=1}^K \frac{[h^{(j)}]^T [C^{(j)}]^{-1} h^{(j)}}{2\theta_j} \right) \\ &\propto \exp \left(-\frac{1}{2} \sum_{j=1}^K p_j \log \theta_j - \frac{1}{2} \sum_{j=1}^K \frac{\|h^{(j)}\|_{(j)}^2}{\theta_j} \right) \\ &\propto \exp \left(-\frac{1}{2} \sum_{j=1}^K p_j \log \theta_j - \frac{1}{2} \|D_\theta^{-1/2} h\|_2^2 \right), \end{aligned}$$

where now D_θ is the block diagonal matrix

$$D_\theta = \begin{bmatrix} \theta_1 C_1 & 0 & \dots & 0 \\ 0 & \theta_2 C_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \theta_K C_K \end{bmatrix}$$

and $C^{(j)} \in \mathbb{R}^{k_j \times k_j}$ are the structural covariance matrices constructed on the basis of the cone condition of Section 3.3.1.

To promote group sparsity, most θ_j 's should be close to zero, with a few large outliers. As in the basic form of the IAS algorithm, we achieve this by postulating that the variables θ_j are independent and distributed according to a fat-tailed distribution such as the gamma distribution. Combining the likelihood, enhanced by the dictionary compression error, we find that the posterior density in this case assumes the form

$$\pi_{H,\Theta|B}(h, \theta | b) \propto \exp \left(-\frac{1}{2} \|C_{\text{DCE}}^{-12}(b - \mu_{\text{DCE}} - Wh)\|_2^2 - \frac{1}{2} \sum_{j=1}^K \frac{\|h^{(j)}\|_{(j)}^2}{\theta_j} + 2 \sum_{j=1}^K \eta_j \log \theta_j - \sum_{j=1}^K \frac{\theta_j}{\vartheta_j} \right),$$

where

$$\eta_j = \beta_j - \frac{k_j + 2}{2}, \quad 1 \leq j \leq K.$$

The MAP estimate corresponding to this posterior density can be computed by the Group Sparsity IAS (GS-IAS), a version of the IAS algorithm that promotes group sparsity. As the standard IAS, the IAS algorithm for group sparsity proceeds by a sequential iteration that alternates between the update of h with fixed θ , and the update of θ with fixed h .

GS-IAS algorithm

Given: matrix W , b , β_j , ϑ_j , $1 \leq j \leq K$,

Initialize: $\theta_j^{(0)} = \vartheta_j$, $1 \leq j \leq K$, set the counter $k = 0$.

Iterate until stopping criterion is met:

1. Update h : Fix $\theta_j = \theta_j^{(0)}$, $1 \leq j \leq K$ and set

$$h^{(k+1)} = \arg \min \left\{ \|C_{\text{DCE}}^{-12}(b - \mu_{\text{DCE}} - Wh)\|_2^2 + \|D_{\theta}^{-\frac{1}{2}} h\|_2^2 \right\}.$$

2. Update θ_j , $1 \leq j \leq K$ by setting

$$\theta_j^{(k+1)} = \frac{\vartheta_j}{2} \left(\eta_j + \sqrt{\eta_j^2 + \frac{\|h^{(j)}\|_{(j)}^2}{4\vartheta_j}} \right),$$

where

$$\eta_j = \beta_j - \frac{k_j + 2}{2}.$$

3. If $\|\theta^{(k)} - \theta^{(k+1)}\| \leq \delta \|\theta^{(k)}\|$ for a given tolerance $0 < \delta < 1$, stop, otherwise advance the counter $k \rightarrow k + 1$ and continue.

4. Computed experiments

We demonstrate the viability of the novel ideas of this article with three computed examples. The first one is a toy problem using the MNIST handwritten digits, demonstrating the performance of the full proposed algorithm from dictionary clustering to dictionary matching with the deflated dictionary. The second example addresses a more realistic problem discussed previously in the literature, related to the identification of glitches in the LIGO/VIRGO experiment for detecting gravitational waves. The third example is a remote sensing application based on hyperspectral imaging and identification of the terrain characteristics. The last two examples highlight the importance of the central new features of this article, the dictionary compression error and the structural prior in group sparsity pursuit.

4.1. Sparse dictionary coding applied to handwritten digits

To demonstrate the main features of the proposed algorithm, we consider first a simple example that allows easily interpretable visualizations. Starting with the standard MNIST database of 16×16 pixel images of handwritten digits stored as vectors in \mathbb{R}^{256} , we generate a dictionary matrix D consisting of $p = 834$ samples of the four digits 0, 1, 3, and 9, hence $D \in \mathbb{R}^{256 \times 834}$. We then run the k -medoids algorithm with an ℓ^2 -distance matrix (see [25]) to cluster the data in four clusters. Observe that the clustering is done in an unsupervised manner, that is, the known annotation is not used to guide the clustering. The clustered vectors are then arranged in four subdictionaries, $D^{(1)}, \dots, D^{(4)}$. The clustering result is visualized on the left in Fig. 3 by plotting the three LDA components [25].

The images are non-negative, and we use a non-negative matrix factorization, computed by using the NNLS-IAS algorithm [24] with non-negativity constraint to extract three feature vectors per each cluster. The feature vectors for each class are shown on the

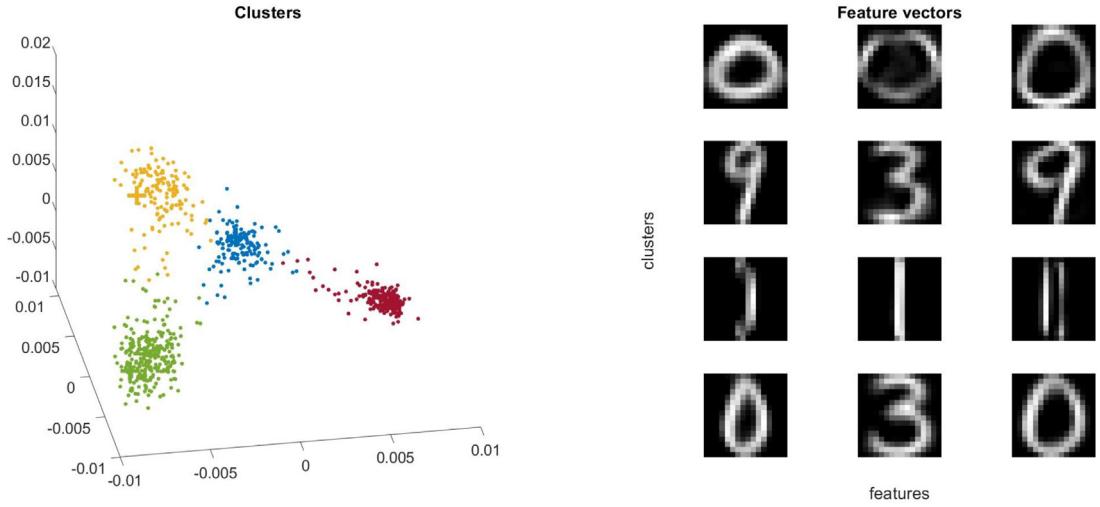


Fig. 3. Left: Scatter plot of the four clusters $D^{(j)}$, visualized by projecting the data on the first three (non-orthogonal) direction vectors obtained by linear discriminant analysis (LDA). The unsupervised clustering does not fully correspond to the classification corresponding to the annotation. Right: The three feature vectors in the reduced dictionary W for each of the four clusters, found by the k -medoids algorithm. Observe that the clustering is not following the annotation, as some of the clusters contain mixtures of features belonging to different annotated groups.

Table 1

The frequencies of occurrence of dissimilarity index values. The total sample size is 4000, and $I_\delta(x, \hat{x}) = 0$ means perfect identification of the atoms, while $I_\delta(x, \hat{x}) = k > 0$ indicates that k atoms were either not identified or an incorrect identification occurred.

Frequency	3904 (97.6%)	34 (0.85%)	1 (0.02%)	1 (0.02%)	1 (0.02%)	59 (1.48%)
$I_\delta(x, \hat{x})$	0	1	2	3	4	≥ 5

right in [Fig. 3](#), plotted as black-and-white images. The figure indicates that the clusters do not correspond to the annotated classes, as the feature vectors of some clusters do not represent the same digit. For the present algorithm, this is not a problem since correct clustering in terms of the annotation information is of no consequence here.

The test data consist of 4000 vectors, each one being linear combinations of one, two, or three digits drawn randomly from different subdictionaries of D with positive coefficients. In particular, the data is within-sample, i.e., drawn from the data set constituting the dictionary. The performance of the algorithm is measured by its capability of correctly identifying the classes constituting each data vector. To define a quantifier for the success, let $x \in \mathbb{R}^{834}$ denote the coefficient vector defining the data vector, $b = Dx$, and let $\hat{x} \in \mathbb{R}^{834}$ be the coefficient vector found by the algorithm. We define the thresholded supports of the vectors,

$$S_\delta = \{j \mid x_j > \delta\}, \quad \hat{S}_\delta = \{j \mid \hat{x}_j > \delta\}, \quad \delta > 0,$$

and define the dissimilarity index as the cardinality of the symmetric difference of them,

$$I_\delta(x, \hat{x}) = \text{card}(S_\delta \Delta \hat{S}_\delta) = \text{card}((S_\delta \cup \hat{S}_\delta) \setminus (S_\delta \cap \hat{S}_\delta)).$$

When $I_\delta(x, \hat{x}) = 0$, the algorithm has correctly identified the atoms that constitute the data. [Table 1](#) shows that in over 95% of cases, the dissimilarity index vanishes, indicating that the algorithm is capable of identifying perfectly the constituting atoms.

The failure of the algorithm can be defined in terms of the relative error in the reconstruction of the coefficient vector,

$$r_x(x, \hat{x}) = \frac{\|x - \hat{x}\|_2}{\|x\|_2}, \quad (10)$$

a quantity that cannot be computed without the knowledge of the true x . However, it turns out that the relative error in b , defined as

$$r_b(x, \hat{x}) = \frac{\|b - D\hat{x}\|_2}{\|b\|_2} = \frac{\|D(x - \hat{x})\|_2}{\|Dx\|_2}, \quad (11)$$

which is a quantity that can be computed directly from the data, provides a reliable proxy for (10). [Fig. 4](#) shows the scatter plot of the two relative errors in logarithmic scale, revealing a strong correlation between the two quantities. Consequently, serious failures of the algorithm can be directly detected by thresholding the relative error in the data.

Finally, we consider the effect of the dictionary compression error (DCE), which is one of the main novelties of this article. We consider a case in which the data corresponds to a single atom out of a single class, and we perturb the data by Gaussian white noise with standard deviation 0.03. We run the algorithm using 5000 simulations, with and without the inclusion of the DCE. Including the DCE leads to a correct cluster identification in 4835 cases, or 97% of the cases. Leaving out the DCE, the correct class

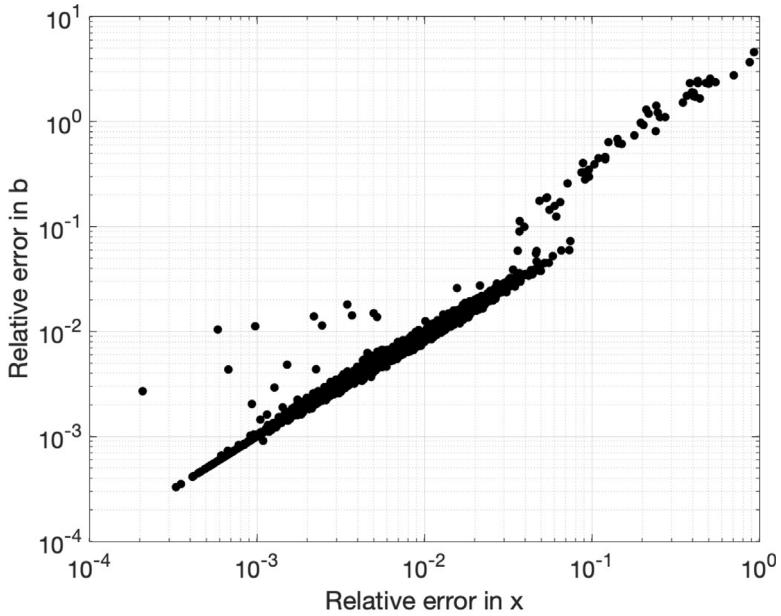


Fig. 4. Relative errors in the reconstruction versus relative error in the data.

is identified 4705 times, or 94% of the cases. Not surprisingly, once the correct class is identified, the correct atom is identified in both simulations in approximately 98% of the cases. The effect of the DCE is observable in this case, but not as crucial as in the following example.

4.2. Dictionary classification for gravitational wave glitch identification

In the second example, we consider the dictionary learning and matching problem related to the LIGO/VIRGO experiment in which the observed gravitational wave, or a coincident event of any unknown origin registered by the devices, is to be identified based on a precomputed dictionary, as finding a universal predictive parametric model is not feasible. Due to the extremely high sensitivity of the interferometer, the data stream is contaminated by frequent artifacts of non-gravitational origin, known as glitches, that due to their high occurrence rate may lead to false coincidence detections. The glitches are transients with particular morphologies that allow them to be classified and identified by using precomputed glitch libraries. The examples considered in this work follow the simplified glitch morphology described in [32]. In the referred article, the authors describe three morphologically different glitch classes, sine Gaussian (SG), Gaussian (G), and ring-down (RD) signals, parametrized as

$$\begin{aligned} h_{\text{SG}}(t) &= h_0 \sin(2\pi f_0(t - t_0)) e^{-(t-t_0)^2/2\tau^2}, \\ h_{\text{G}}(t) &= h_0 e^{-(t-t_0)^2/2\tau^2}, \\ h_{\text{RD}}(t) &= h_0 \sin(2\pi f_0(t - t_0)) e^{-(t-t_0)/\tau} \theta(t - t_0), \end{aligned}$$

where h_0 is the amplitude, f_0 is the frequency, t_0 is the center time and τ is the characteristic time of the glitch, and θ is the Heaviside function. The morphology of real LIGO glitches is wider than this, see, e.g., [33,34] for typical spectrograms.

We begin by generating a model-based library of signals. The duration of each signal is one second, and the sampling frequency is 16 384 Hz as in the LIGO experiment, leading to a dictionary with $n = 16 384$. We generate 1000 atoms per class. The model parameters are drawn randomly as in [32]: For SG and RD waveforms, f_0 (Hz) is drawn from log-uniform distribution over the interval [40, 1500] Hertz, and consequently, τ is set as $\tau = Q/\sqrt{2\pi}f_0$, where the quality factor Q is drawn from log-uniform distribution over the interval [2, 20]. For the Gaussian waveform, τ is drawn from log-uniform distribution over [0.001, 0.01] seconds. The amplitudes are chosen so that the maximum of the signal is one, and the signals are centered by setting $t_0 = 0.5$ s.

We partition the atoms into three subdictionaries, using the natural clustering by the morphology of the glitches rather than resorting to clustering by k -medoids. The subdictionaries are subsequently compressed with a low-rank matrix factorization. Since the sine Gaussian and the ring-down signals admit negative values while for the Gaussian glitches are nonnegative, in the low-rank approximation we require non-negativity and sparsity of the coefficient matrix $H^{(j)}$ for each class, but non-negativity of $W^{(j)}$ only for the feature vectors of the Gaussian class. The number of feature vectors in each low-rank approximation is set for each subdictionary based on an estimation of its effective rank. More specifically, let K be the upper bound for the number of feature vectors in each subdictionary, chosen, e.g., based on computational considerations. If $\sigma_k^{(j)}$ is the k th singular value of the j th subdictionary matrix

$D^{(j)}$, the rank k_j of low-rank approximation of $D^{(j)}$ is set as

$$k_j = \min \left\{ K, \min \left\{ k \mid \frac{\sigma_k^{(j)}}{\sigma_1^{(j)}} < \delta \right\} \right\}$$

for some fixed δ , $0 < \delta < 1$. Here we use the values $\delta = 0.001$ and $K = 50$. We remark that while computing the full SVD of the subdictionaries is an expensive task, estimating the first few singular values is not a demanding task and often the signals in each subdictionary are quite similar, thus only a few singular values are required. The dimensions of the reduced dictionaries in this case are $k_{SG} = 50$, $k_G = 11$ and $k_{RD} = 50$. Using the training set, we compute the dictionary reduction error mean and covariance.

To demonstrate that including the modeling error in the identification problem makes a significant difference in the performance of the task, which in this case is to identify the morphology of a given glitch using the reduced subdictionaries, we generate 20 000 noisy signals by drawing randomly the parameters of the generative model and add scaled Gaussian white noise with standard deviation $\sigma = 0.05 h_0$, after which we normalize the signals so that their maximum value is 1. Hence, each glitch is corrupted with slightly different levels of scaled white noise and, unlike in the previous example, the test signals are out-of-sample data.

To classify b , we find the least squares solution of the linear systems

$$b \approx W^{(j)} h^{(j)} = \hat{b}^{(j)}, \quad j \in \{SG, G, RD\},$$

whitened to compensate for the DCE correction, using the IAS algorithm with positivity and sparsity promotion. We compute the approximations $\hat{b}^{(j)}$ and, as in [32], compare them to b using the structural dissimilarity index proposed in [35],

$$DSSIM(b, \hat{b}^{(j)}) = \frac{1}{2} (1 - SSIM(b, \hat{b}^{(j)})).$$

The datum b is assigned to the class with smallest dissimilarity index.

To acknowledge the fact that the noise level is not known exactly, we overestimate it slightly, using

$$\tilde{\sigma} = 1.05\sigma. \quad (12)$$

Below, the confusion matrices with noise levels $\sigma = 0.05 h_0$, $\sigma = 0.02 h_0$ and $\sigma = 0.01 h_0$ are given, the rows corresponding to the classification results, and columns to the true classes:

	SG	G	RD		SG	G	RD		SG	G	RD
SG	6775	0	4	SG	6749	0	2	SG	6755	0	19
G	57	6535	0	G	0	6598	0	G	0	6599	0
RD	129	1	6499	RD	37	0	6499	RD	29	0	6598

In particular, the success rates are 99.0%, 99.2%, and 99.8%, respectively.

Interestingly, the compensation of the DCE in this example is crucial for the success of the algorithm. In fact, when omitting the DCE correction the classification goes egregiously astray even for $\sigma = 0.01 h_0$, as illustrated by the following confusion matrix

	SG	G	RD
SG	2	6797	3
G	6583	0	0
RD	6071	337	207

4.3. Group sparsity promotion for hyperspectral data classification

In this computed experiment we illustrate how to use the group structural prior to filter out from a partitioned dictionary those subdictionaries that play no role in the dictionary representation of a new datum. The dictionary is the Pavia University dataset, a hyperspectral image dataset collected in 2002. A sensor, known as the Reflective Optics System Imaging Spectrometer (ROSIS-3) [36], during a flight over the city of Pavia, Italy, collected 115 spectral bands covering wavelengths from 430 to 860 nm at locations above the University of Pavia. Of the spectral bands, 12 of them corresponding to noisy channels were removed, leaving 103 spectral bands per each pixel. The image area consists of 610×340 pixels with spatial resolution of 1.3 m. Of all the hyperspectral samples corresponding to the pixels, 40,776 of them, or approximately 20%, were labeled according to the ground truth into 9 urban classes, including asphalt, meadows, gravel, trees, metal sheet, bare soil, bitumen, brick, and shadow.

While testing whether the hyperspectral data has enough resolution to resolve the ground truth classes by comparing the annotation with the results of unsupervised clustering with different number of classes, it was observed that several of the 9 urban classes corresponding to the ground truth were confounded, suggesting that effectively it would be more reasonable to reduce the partitioning to 6 classes, not necessarily in line with the ground truth. The ground truth clustering into 9 classes, together with the pseudoimage of the partitioning into 6 classes obtained by unsupervised clustering (k -medoids with ℓ_1 -distance matrix) are displayed in the left and right panels of Fig. 5. A comparison of the two pseudocolor images with an RGB image of the same area indicates that the partitioning via unsupervised clustering is closer to the RGB image than the expert annotation.

We select half of the atoms in each of the six subdictionary for training, reserving the remaining for testing the effect of structural cone condition on the classification of the remaining data. Denoting the j th training subdictionary by $D^{(j)}$, we compute its rank- k PCA approximation,

$$D^{(j)} \approx U_k^{(j)} \Sigma_k^{(j)} (V_k^{(j)})^\top = W^{(j)} H^{(j)},$$

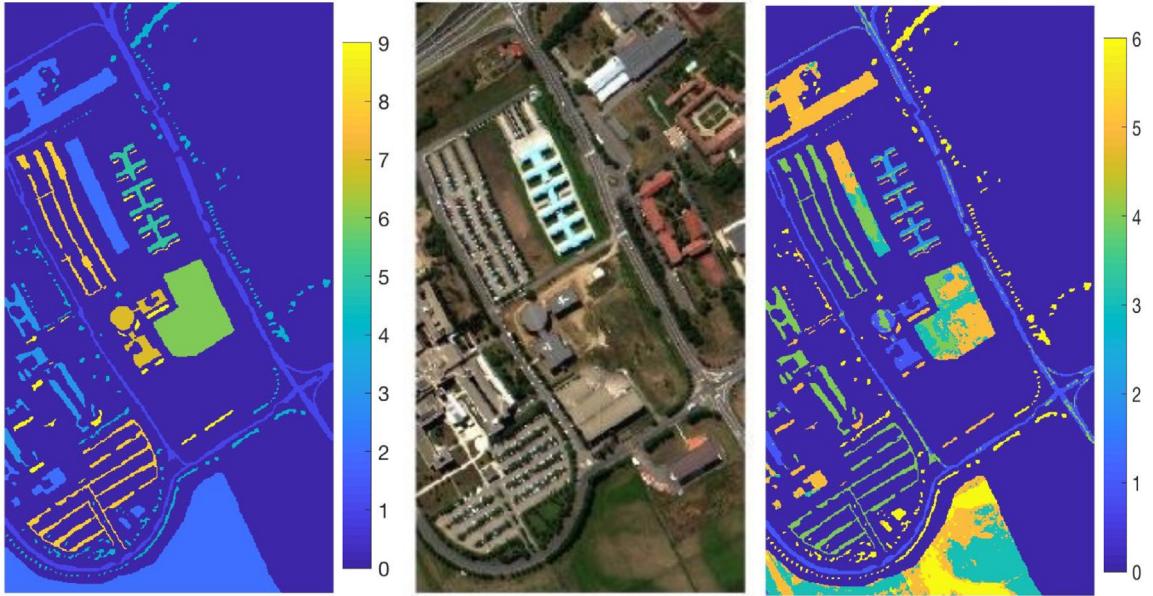


Fig. 5. The partitioning of the labeled pixels of the University of Pavia areal view according to ground truth labeling (left) and unsupervised clustering (right). The ground truth classes correspond to Asphalt (1), Meadows (2), Gravel (3), Trees (4), Metal Sheet (5), Bare Soil (6), Bitumen (7), Brick (8), Shadows (9). Pixels in class 0 are those for which no ground truth based label is available.

where the truncation index k is chosen based on the singular values $\sigma_\ell^{(j)}$ such that

$$\sigma_k^{(j)} \geq \delta \sigma_1^{(j)} > \sigma_{k+1}^{(j)}, \quad \delta = 5 \times 10^{-3}.$$

By construction, the SVD of the coefficient matrix $H^{(j)}$ is simply given by

$$H^{(j)} = \Sigma_k^{(j)} \left(V_k^{(j)} \right)^T,$$

therefore the cone condition defines the structural prior covariance corresponding to the j th class as

$$C^{(j)} = \text{diag} \left((1, \sigma_2^{(j)} / \sigma_1^{(j)}, \dots, \sigma_k^{(j)} / \sigma_1^{(j)}, 0, \dots, 0) \right) + \epsilon I.$$

To test the group promotion power of the structural cone prior, we randomly select 100 hyperspectra from the testing data from each subdictionary and solve the compressed dictionary coding problem using the group sparsity promoting IAS with and without the structural cone prior. We use the value of the hyperparameter θ_j , $1 \leq j \leq 6$ as a proxy for the relevance of the j th class, scaling the vector θ of each draw by its largest component, so as to have an infinity norm 1, and look at how many classes have large enough θ to be deemed as relevant.

Fig. 6 shows the normalized vectors θ for the six classes computed by the IAS algorithm with the structural cone prior (top row) and without (bottom row) when the 100 data vectors are random draws of atoms from the testing set subdictionaries 1–3. The analogous results for subdictionaries 4–6 are shown in Fig. 7. The yellower the segment corresponding to the class, the higher the probability that the class plays a significant role, while the bluer the segment, the more likely the corresponding class is irrelevant for the datum on hand.

A popular way to quantify the accuracy of a classification algorithm are the *precision* or *positive predictive value*, and *precision* or *sensitivity*, defined as

$$\text{Precision} = P = \frac{\# \text{ relevant retrieved classes}}{\# \text{ retrieved classes}}, \quad \text{Recall} = R = \frac{\# \text{ relevant retrieved classes}}{\# \text{ relevant classes}}.$$

Hence, the precision indicates how effectively the algorithm deflates the dictionary without missing the correct subdictionary, while recall indicates how often the algorithm misses the true subdictionary, regardless of its ability to deflate. Ideally, both indices should be close to one. The precision and recall are often summarized by the F_1 -score, defined as a harmonic mean of the two,

$$\frac{1}{F_1} = \frac{1}{2} \left(\frac{1}{P} + \frac{1}{R} \right).$$

Fig. 8 shows the precision, recall and F_1 -score of the group sparsity algorithm with and without the structural cone condition, as a function of the cut-off value θ^* . The plots show that with a typical value $\theta^* = 0.3$, the algorithm with the cone condition reduces the problem to approximately two subdictionaries (precision ≈ 0.5) while without the cone condition, roughly 5 subdictionaries are retrieved (precision ≈ 0.2). On the other hand, the algorithm without the cone condition is likely to retrieve the true class (recall

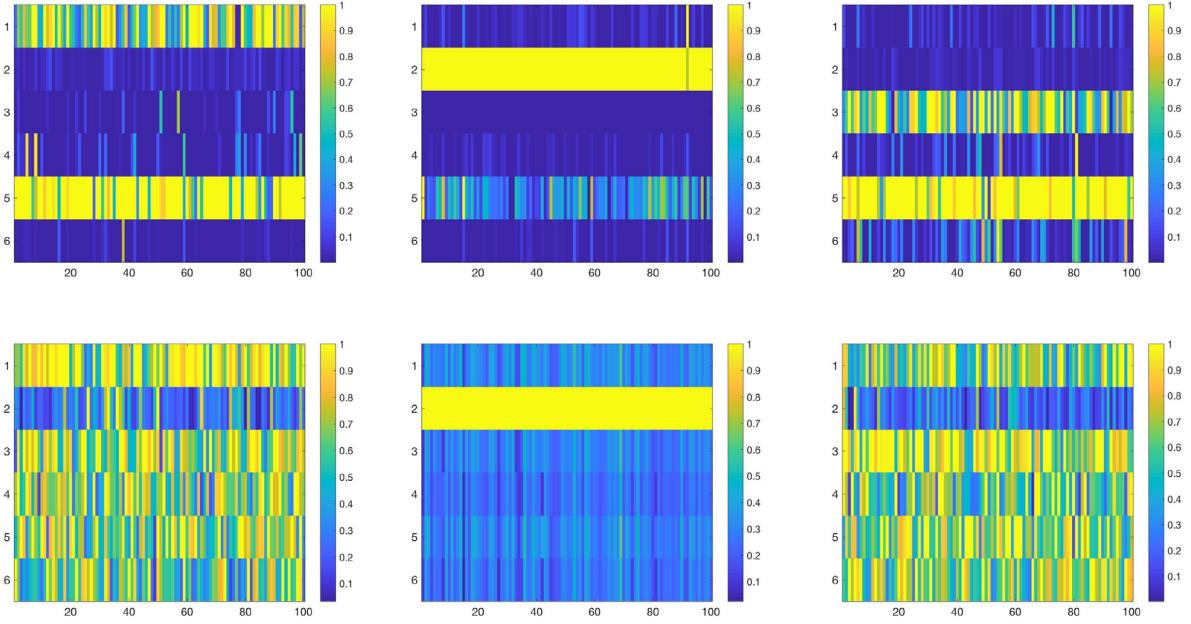


Fig. 6. Each column shows the hyperparameters for the six classes scaled to have infinity unit norm. The panels in each columns correspond to data from subdictionaries 1, 2 and 3, respectively. The top rows reports the result for IAS with the structural code prior, while the results in the bottom row are obtained with IAS without structural code prior.

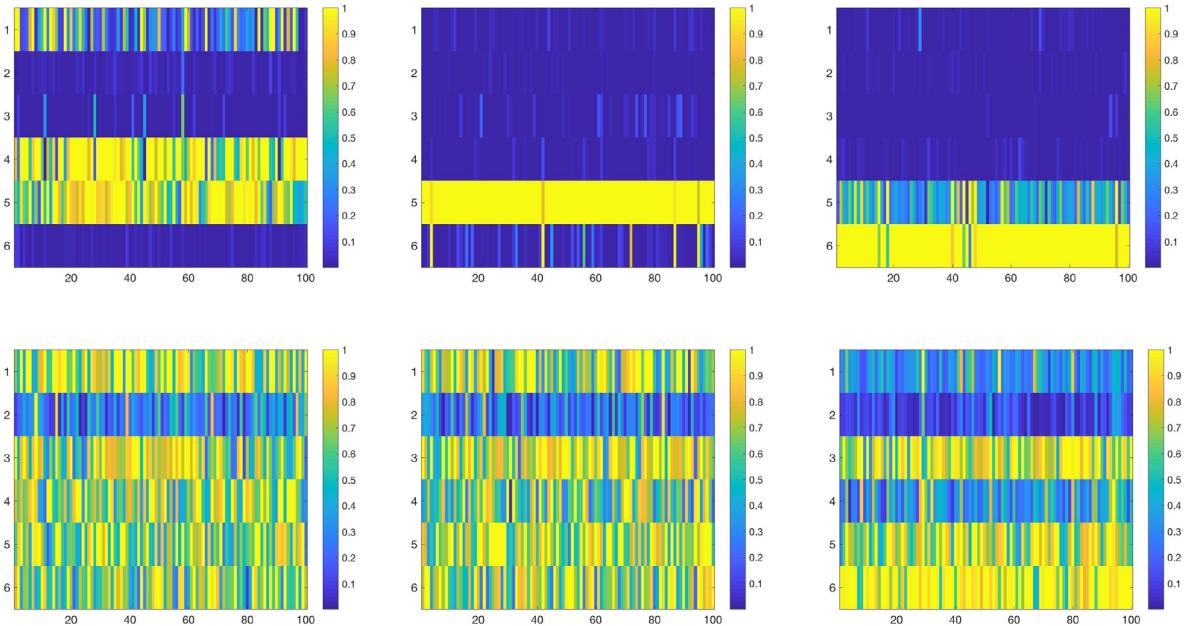


Fig. 7. Each column shows the hyperparameters for the six classes, scaled to have infinity unit norm. The panels in each columns correspond to data from subdictionaries 4, 5 and 6, respectively. The top rows reports the result for IAS with the structural code prior, while the results in the bottom row are obtained with IAS without structural code prior.

≈ 1) as it does a poor job in excluding classes, while the algorithm with the cone condition discards the correct class in roughly one case out of 20 (recall ≈ 0.95 .)

To complete this example, we run the full dictionary identification problem as follows: First, we draw randomly ten hyperspectra from the first test dictionary sample, and reduce the dictionary by using the group sparsity algorithm including the structural prior covariance. After discarding the subdictionaries with $\theta_j/\theta_{\max} < 0.3$, we run the deflated dictionary identification algorithm using the hybrid IAS algorithm combining the gamma and inverse gamma hyperpriors to guarantee a strong sparsification of the coefficient

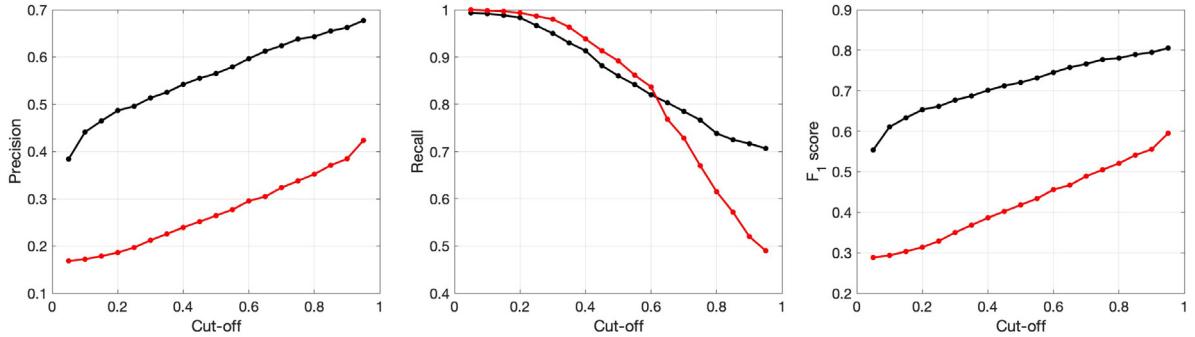


Fig. 8. Precision and recall of the two group sparsity algorithms as a function of the cut-off value θ^* . The black curves correspond to the group sparsity when the structural prior based on the cone condition is used, while the red ones correspond to the case when the structural prior covariances are replaced by an identity. We observe that including the cone condition strongly improves the discarding of irrelevant classes (higher precision), without significantly deteriorating the recall.

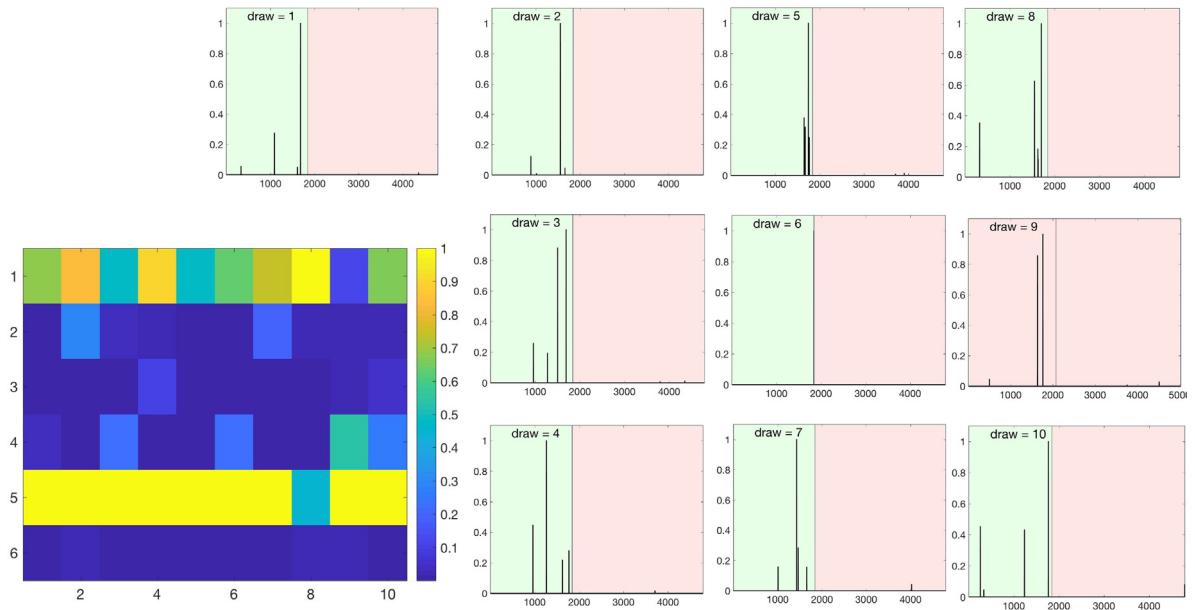


Fig. 9. Ten randomly drawn hyperspectra from the class 1 test subdictionary. The result of the relevant class identification shows that the first class is routinely confused with the fifth class. Nonetheless, when running the dictionary identification test using the hybrid IAS, the algorithm systematically proposes non-zero coefficients from the correct class (green background) rather than from the incorrect one (red background), except in the case of draw number 9 when the correct class was classified as irrelevant and discarded. The cut-off value for relevant classes in this example is $\theta^* = 0.3$.

vectors. In Fig. 9, we show both the scaled group sparsity vectors as well as the outcomes of the deflated dictionary identification problems. In the plot, the coefficients corresponding to the correct subdictionary are plotted on a green background, while the coefficients in an incorrect subdictionary are on a red background. Interestingly, when the correct subdictionary is identified by the group sparsity phase as relevant for explaining the datum, the deflated solver seems to recognize the correct class: The non-zero coefficients are primarily chosen to correspond to the correct class rather than the incorrect ones even when the group sparsity phase indicates a stronger preference for a wrong class. The only case when coefficients from an incorrect class are proposed is when the correct class was deemed irrelevant.

5. Conclusions

Bayesian methods provide a versatile and rich environment for computational inverse problems, making it possible to integrate prior information about the unknowns, and to compensate for shortcomings in the models. Interpreting data science problems such as dictionary learning and dictionary matching as inverse problems opens new ways to introduce well developed Bayesian methods into this active and rapidly growing field. This article demonstrates how the Bayesian modeling error approach and structural prior information can be translated into effective tools to improve classification in the sense of subdictionary identification, and to

solve efficiently large scale dictionary matching problems with sparsity promoting priors. The Bayesian approach opens also a way of addressing an important and novel area of research, the uncertainty quantification in data science, crucial for the reliability of data-driven methods in areas like medical imaging and structural health research where the assessment of the reliability is of central importance.

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.

Acknowledgments

This work made use of the High Performance Computing Resource in the Core Facility for Advanced Research Computing at Case Western Reserve University. The work of DC was supported in part by US National Science Foundation Award DMS 1951446. The work of ES was supported in part by US National Science Foundation Award DMS 2204618.

References

- [1] Tošić I, Frossard P. Dictionary learning. *IEEE Signal Process Mag* 2011;28:27–38.
- [2] Dong W, Li X, Zhang L, Shi G. Sparsity-based image denoising via dictionary learning and structural clustering. In: *CVPR*. IEEE; 2011, p. 457–64, (June).
- [3] Elad M, Aharon M. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans Image Process* 2006;15:3736–45.
- [4] Xu Q, Yu H, Mou X, Zhang L, Hsieh J, Wang G. Low-dose X-ray CT reconstruction via dictionary learning. *IEEE Trans on Med Imaging* 2012;31:1682–97.
- [5] Ma D, Gulani V, Seiberlich N, Liu K, Sunshine JL, Duerk JL, et al. Magnetic resonance fingerprinting. *Nature* 2013;495:187–92.
- [6] Vu TH, Mousavi HS, Monga V, Rao U, Rao G. DFDL: Discriminative feature-oriented dictionary learning for histopathological image classification. In: *Proc IEEE international symposium on biomedical imaging*, vol. 35, 2015, p. 990–4.
- [7] Bioucas-Dias JM, Plaza A, Dobigeon N, Parente M, Du Q, Gader P, et al. Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches. *IEEE J Sel Top Appl Earth Observ Remote Sens* 2012;5:354–79.
- [8] Sun X, Qu Q, Nasrabadi NM, Tran TD. Structured priors for sparse-representation-based hyperspectral image classification. *IEEE Geosci Remote Sens Lett* 2014;11:1235–1239.
- [9] Saiz-Pérez A, Torres-Forné A, Font JA. Classification of the core-collapse supernova explosion mechanism with learned dictionaries. 2021, arXiv preprint [arXiv:2110.12941](https://arxiv.org/abs/2110.12941).
- [10] Torres-Forné A, Marquina A, Font JA, Ibanez JM. Denoising of gravitational wave signals via dictionary learning algorithms. *Phys Rev D* 2016;94:124040.
- [11] Wright J, Yang AY, Ganesh A, Sastry SS, Ma Y. Robust face recognition via sparse representation. *IEEE Trans Pattern Anal Mach Intell* 2009;31:210–27.
- [12] Van Nguyen H, Patel VM, Nasrabadi NM, Chellappa R. Design of non-linear kernel dictionaries for object recognition. *IEEE Trans Image Process* 2013;22:5123–35.
- [13] Candes E, Tao T. Decoding by linear programming. *IEEE Trans Inform Theory* 2005;12:4203–15.
- [14] Donoho DL. Compressed sensing. *IEEE Trans Inform Theory* 2006;52:1289–306.
- [15] Chen S, Donoho D, Saunders M. Atomic decomposition by basis pursuit. *SIAM J Sci Comput* 1998;20:33–61.
- [16] Tibshirani R. Regression shrinkage and selection via the lasso. *J R Stat Soc Ser B Stat Methodol* 1996;58:267–88.
- [17] Calvetti D, Pascarella A, Pitelli F, Somersalo E, Vantaggi B. A hierarchical Krylov-Bayes iterative inverse solver for MEG with physiological preconditioning. *Inverse Problems* 2015;31:125005.
- [18] Calvetti D, Somersalo E, Strang A. Hierarchical Bayesian models and sparsity: ℓ_2 -magic. *Inverse Problems* 2019;35:035003.
- [19] Calvetti D, Somersalo E. Bayesian scientific computing. New York: Springer Verlag; 2023.
- [20] Calvetti D, Pragliola M, Somersalo E, Strang A. Sparse reconstructions from few noisy data: Analysis of hierarchical Bayesian models with generalized gamma hyperpriors. *Inverse Problems* 2020;36:025010.
- [21] Pragliola M, D. Calvetti, E. Somersalo. Overcomplete representation in a hierarchical Bayesian framework. *Inverse Probl Imaging* 2022;16:19–38.
- [22] Aharon M, Elad M, Bruckstein A. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans Signal Process* 2006;54:4311–22.
- [23] Yang M, Zhang L, Yang J, Zhang D. Metaface learning for sparse representation based face recognition. In: *17th IEEE international conference on image processing*. 2010, p. 1601–4.
- [24] Waniorek N, Calvetti D, Somersalo E. Bayesian hierarchical dictionary learning. *Inverse Problems* 2023;39:024006.
- [25] Calvetti D, Somersalo E. Mathematics of data science: A computational approach to clustering and classification. Philadelphia: SIAM; 2020.
- [26] Gillis N. Nonnegative matrix factorization. Philadelphia: Society for Industrial and Applied Mathematics; 2020.
- [27] Kaipio J, Somersalo E. Statistical inverse problems: Discretization, model reduction and inverse crimes. *J Comp Appl Math* 2007;198:493–504.
- [28] Arridge SR, Kaipio JP, Kolehmainen V, Schweiger M, Somersalo E, Tarvainen T, et al. Approximation errors and model reduction with an application in optical diffusion tomography. *Inverse Probl* 2006;22:175.
- [29] Calvetti D, Dunlop M, Somersalo E, Stuart A. Iterative updating of model error for Bayesian inversion. *Inverse Problems* 2018;34:025008.
- [30] Calvetti D, E. Somersalo. Computationally efficient sampling methods for sparsity promoting hierarchical Bayesian models. 2023, arXiv preprint [arXiv:2303.16988](https://arxiv.org/abs/2303.16988).
- [31] Calvetti D, Pragliola M, Somersalo E. Sparsity promoting hybrid solvers for hierarchical Bayesian inverse problems. *SIAM J Sci Comput* 2020;42:A3761–84.
- [32] Llorens-Monteagudo M, Torres-Forné A, Font JA, Marquina A. Classification of gravitational-wave glitches via dictionary learning. *Classical Quantum Gravity* 2019;36:075005.
- [33] Powell J, Trifirò D, Cuoco E, Heng IS, Cavaglia M. Classification methods for noise transients in advanced gravitational-wave detectors. *Classical Quantum Gravity* 2015;32(21):215012.
- [34] Powell J, Torres-Forné A, Lynch R, Trifirò D, Cuoco E, Cavagliá M, et al. Classification methods for noise transients in advanced gravitational-wave detectors II: Performance tests on advanced LIGO data. *Classical Quantum Gravity* 2017;34(3):034002.
- [35] Wang L, Zhao L, Bi G, Wan C. Hierarchical sparse signal recovery by variational Bayesian inference. *IEEE Signal Process Lett* 2014;21:110–3.
- [36] Kunkel B, Blechinger F, Lutz R, Doerffer R, Van der Piepen H, Schroder M. ROSIS (reflective optics system imaging spectrometer)-A candidate instrument for polar platform missions. In: *Optoelectronic technologies for remote sensing from space* 1988 Apr 13, vol. 868, SPIE; p. 134–41.