Signal Processing 137 (2017) 199–212

Contents lists available at ScienceDirect

Signal Processing

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

Online sparse and low-rank subspace learning from incomplete data: A Bayesian view

Paris V. Giampouras[a,b,*,**], Athanasios A. Rontogiannis[a], Konstantinos E. Themelis[a], Konstantinos D. Koutroumbas[a]

[a] Institute for Astronomy, Astrophysics, Space Applications and Remote Sensing (IAASARS), National Observatory of Athens, 15236, Penteli, Greece
[b] Department of Informatics and Telecommunications, University of Athens, 15784 Athens, Greece

ARTICLE INFO

Article history:
Received 14 October 2016
Revised 25 January 2017
Accepted 5 February 2017
Available online 10 February 2017

Keywords:
Subspace tracking
Online matrix completion
Online variational Bayes
Incomplete data
Sparse subspace learning
Low-rank

ABSTRACT

Extracting the underlying low-dimensional space where high-dimensional signals often reside has been at the center of numerous algorithms in the signal processing and machine learning literature during the past few decades. Moreover, working with incomplete large scale datasets has recently been commonplace for diverse reasons. This so called *big data era* we are currently living calls for devising online subspace learning algorithms that can suitably handle incomplete data. Their anticipated goal is to *recursively* estimate the unknown subspace by processing streaming data sequentially, thus reducing computational complexity. In this paper, an online variational Bayes subspace learning algorithm from partial observations is presented. To account for the unawareness of the true rank of the subspace, commonly met in practice, low-rankness is explicitly imposed on the sought subspace data matrix by exploiting sparse Bayesian learning principles. Sparsity, *simultaneously* to low-rankness, is favored on the subspace matrix by the sophisticated hierarchical Bayesian scheme that is adopted. The proposed algorithm is thus adept in dealing with applications whereby the underlying subspace may be also sparse. The new subspace tracking scheme outperforms its state-of-the-art counterparts in terms of estimation accuracy, in a variety of experiments conducted on both simulated and real data.

1. Introduction

Recent years are by all means characterized by the vast amounts of data, commonly named with the blanket term *big data*, generated by a wealth of sources such as social media, environmental monitoring sensors, medical application devices, e-commerce sites etc. to mention just a few. In first place, having at hand a lot of data seems to be fairly advantageous. However, enjoying the merits emerging from this so called data deluge raises a number of issues needed to be properly addressed. Among other things, computational complexity and memory storage requirements are undoubtedly two basic aspects needed to be carefully taken into consideration in the challenging task of devising appropriate processing tools for extracting useful information from big data.

Detecting the underlying low-dimensional space (subspace) where high-dimensional data reside, is at the heart of several signal processing and machine learning tasks, such as network anomalies detection, [1], image denoising, [2,3], direction of arrival (DOA) estimation, [4], etc. Batch methods such as the celebrated PCA, which indubitably holds a prominent position in the family of this kind of algorithms, face considerable difficulties since a) their computational complexity scales with the size of the available measurement data and b) they require the storage of the whole bunch of data in memory. Therefore, its application is becoming practically prohibitive in the big data scenario under study. In light of this, *online* subspace estimation (tracking) algorithms, that first came into the scene in the 1970s, [5,6], have nowadays regain their popularity, [4,7,8]. These tools build upon the hypothesis that datums are sequentially arriving and thus the unknown subspace is *adaptively* estimated each time a new data sample becomes available. Interestingly, this premise, besides reducing the computational complexity, leads to schemes with no need of storing data in memory. Moreover, in a variety of applications dealing with large scale datasets, datums to be processed are partly observed i.e., a fraction of them might be missing. Depending on the case, incomplete datasets may result either from applying compressed sensing ideas in an effort to facilitate or account for failures in the data acquisition process, [9,10] or from the inherent nature of signals met in disparate applications, e.g. collaborative filtering, [11],

* Corresponding author.
** This work was supported by the PHySIS Project under Contract 640174 within the H2020 Framework Program of the European Commission.
E-mail address: parisg@noa.gr (P.V. Giampouras).

http://dx.doi.org/10.1016/j.sigpro.2017.02.003
0165-1684/

image reconstruction [12], etc. Consequently, algorithms that perform subspace tracking from (possibly highly) incomplete data have flourished notably in the last few years.

1.1. Related work

Along those lines, the GROUSE algorithm, which brings forth an approach based on stochastic gradient descent on the Grassmanian manifold of subspaces, has been presented in [7]. Since stochastic approximation is at the core of GROUSE, its computational complexity classifies it to the low-complexity subspace tracking algorithms, [13]. Local and global convergence of GROUSE to the global minimum has been recently theoretically proved in [14] and [15], respectively. In [4], a second order subspace tracking algorithm, of similar computational complexity to GROUSE, dubbed PETRELS, has been presented. PETRELS is an *unconstrained* alternating minimization recursive least squares (RLS)-type algorithm, building upon the seminal PASTd subspace tracking algorithm, [16], and extending it for handling missing data. A common characteristic of both the aforementioned algorithms is the rather strong assumption that the true rank of the sought subspace is known in advance. This shortcoming, which makes PETRELS exhibit an unstable behavior in case the assumption does not hold, is addressed in [17], where two different algorithms are described. Therein, an upper bound of the nuclear norm is favorably employed for imposing low-rankness on the unknown subspace matrix, thus robustifying the algorithms in the challenging yet realistic scenario of lacking the knowledge of the subspace rank. In that vein, Algorithm 1 of Mardani et al. [17] is introduced, deriving from an alternating minimization strategy on an exponentially weighted *regularized* cost function. In addition, a more efficient in terms of computational complexity Algorithm 2 is presented, based on a stochastic gradient descend approach.

In a Bayesian framework, low-rank subspace estimation from incomplete data has been recently dealt with in [18]. Through an elegant joint column sparsity promoting mechanism, originally proposed in [19] in the context of nonnegative matrix factorization, the initially selected subspace rank is progressively reduced, tending to the true rank of the unknown subspace. In [18], group sparsity promoting Student-t type priors are employed and the variational Bayes method [20] is used for inference. In a similar vein, in [21], a Bayesian approach based on generalized approximate message passing for addressing the bilinear inference problem was presented. However, the subspace estimation algorithms developed in [18,21] are of a *batch* type and thus are incapable to process in the end high volumes of incomplete streaming data.

1.2. Contribution

Capitalizing on our previous work on online (group) sparse linear regression [22,23] and leveraging the low-rank promotion idea presented in [19], we devise a new online sparse and low-rank subspace estimation algorithm, termed OVBSL, that learns from incomplete data. After the statement of the problem (Section 2), the followed methodology consists of a) the definition of an appropriate Bayesian model for the problem at hand (Section 3), b) the use of the variational Bayes inference method to solve the problem via a batch iterative scheme Section 4) and c) the derivation of an *online* algorithm with a suitable extension and modification of the batch algorithm (Section 5). It is worth emphasizing that in this paper the proposed Bayesian model incorporates exponentially weighted data and parameter priors, which facilitate online inference in a time-varying environment. Moreover, departing from the Bayesian subspace estimation scheme of Babacan et al. [18], multiple sparsity constraints are imposed on the subspace matrix in the form of appropriate Gaussian scale mixture priors, in order for the proposed scheme to be capable of addressing the sparse dictionary learning problem, [8,24].

In Section 6, the relevance of OVBSL to the deterministic PETRELS and Algorithm 1 of [17] is brought to light within a *maximum a posteriori* (MAP) framework arising from our adopted hierarchical model. It is favorably illuminated that from a deterministic point of view OVBSL departs from the *unconstrained* RLS-type PETRELS algorithm and likewise the algorithms of Mardani et al. [17] can be deemed as an algorithm closely associated with the minimization of an exponentially weighted *regularized* least squares cost function. However, the key difference to the algorithms of Mardani et al. [17] is that instead of utilizing the upper-bound of the nuclear norm introduced in [25], low-rankness on the subspace matrix is now aptly provoked by the group-sparsity inducing ℓ_2/ℓ_1 norm. As far as the computational complexity of OVBSL is concerned, by virtue of the statistical independence imposed on the elements of the subspace matrix, it is similar to that of GROUSE, PETRELS and the stochastic approximation type Algorithm 2 of Mardani et al. [17] and lower than that of the second-order Algorithm 1 of Mardani et al. [17].

OVBSL shares the compelling characteristic of all Bayesian approaches, that is, it is fully automated. Thus, contrary to its deterministic state-of-the-art rivals, herein no tuning parameter is required. Moreover, since all parameters are treated as variables, OVBSL instead of point estimates, provides the sufficient statistics of the probability distributions of all the involved parameters, thus offering more valuable supplementary information, compared to its deterministic counterparts. As demonstrated on simulated and real data experiments in Section 7, it presents superior estimation performance than the three most related state-of-the-art algorithms described earlier. To validate this, online matrix completion and (either sparse or non sparse) subspace recovery from missing data are simulated as case studies. Finally, the hyperspectral image reconstruction and the eigenface learning problems are examined, corroborating the effectiveness and higher reconstruction performance of OVBSL on real data.[1]

1.3. Notation

Column vectors are represented as boldface lowercase letters, e.g. $\mathbf{x}$, and matrices as boldface uppercase letters, e.g. $\mathbf{X}$, while, unless otherwise explicitly stated, x_i is the ith element of $\mathbf{x}$ and x_{ij} the ijth element of $\mathbf{X}$. In particular, small boldface calligraphic letters are used to denote columns of a matrix $\mathbf{X}$ (i.e. $\boldsymbol{x}_i$) and regular boldface letters to denote rows, that is $\mathbf{x}_j^T$ and $(\cdot)^T$ denotes transposition. Moreover $\mathbf{I}_k$ is the $k \times k$ identity matrix, $\|\cdot\|_2$ is the standard ℓ_2-norm, $\|\cdot\|_F$ stands for the Frobenius norm, $\|\cdot\|_*$ is the nuclear norm, $\odot$ denotes Hadamard entrywise product, $\langle\cdot\rangle$ is the expectation operator, diag($\mathbf{x}$) denotes a diagonal matrix whose diagonal entries are the elements of $\mathbf{x}$, diag($\mathbf{X}$) is a column vector whose entries are the diagonal elements of the square matrix $\mathbf{X}$, Trace($\mathbf{X}$) is the trace of the square matrix $\mathbf{X}$, $|\mathbf{X}|$ its determinant and span($\mathbf{X}$) is the range (column space) of matrix $\mathbf{X}$. Finally, $\mathcal{N}(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\Sigma})$ denotes the Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. $\mathcal{GIG}(x;p,a,b)$ is the one-dimensional generalized inverse Gaussian distribution defined as

$$\mathcal{GIG}(x;p,a,b)=\frac{(a/b)^{p/2}\exp\left[(p-1)\log x-(ax+\frac{b}{x})/2\right]}{2\mathcal{K}_p(\sqrt{ab})},$$

where $x>0$, $a>0$, $b>0$, p is real, and $\mathcal{K}_p(\cdot)$ denotes the modified Bessel function of second kind with p degrees of freedom. The pdf of the Gamma distribution is

$$\mathcal{G}(x;\zeta,\tau)=\exp[(\zeta-1)\log x-x\tau-\log\Gamma(\zeta)+\zeta\log\tau],$$

[1] A preliminary version of a part of this work was presented in [26].

where $\Gamma(\cdot)$ is the gamma function, while

$$\mathcal{IG}(x;\zeta,\tau)=\exp\left[-(\zeta+1)\log x-\frac{\tau}{x}-\log\Gamma(\zeta)+\zeta\log\tau\right]$$

is the inverse Gamma distribution.

2. Problem statement

Let n be the time-index and $\mathbf{y}(n)$ a sequence of high-dimensional $K\times 1$ vectors of observations that lie in a linear low-dimensional subspace of rank $r(n)$ with $r(n)\ll K$. Both the linear subspace and its rank may be time-varying. Accordingly, the observations at time n can be expressed as,

$$\mathbf{y}(n)=\mathbf{U}(n)\mathbf{c}(n), \tag{1}$$

where $\mathbf{U}(n)$ is a $K\times r(n)$ matrix whose columns span the underlying data subspace and vector $\mathbf{c}(n)$ contains the coefficients of $\mathbf{y}(n)$ in this subspace. Since, in general, the true rank $r(n)$ of $\mathbf{U}(n)$ is unknown and in order to account for noisy observations, we may assume that our data are produced based on the following linear regression model

$$\mathbf{y}(n)=\mathbf{W}(n)\mathbf{x}(n)+\mathbf{e}(n), \tag{2}$$

where $\mathbf{W}(n)$ is a $K\times L$ subspace matrix with $K\gg L\geq r(n)$ and $\text{span}(\mathbf{U}(n))\subseteq\text{span}(\mathbf{W}(n))$. Moreover, in (2), the $L\times 1$ vector $\mathbf{x}(n)$ is the low-dimensional representation of $\mathbf{y}(n)$ in the subspace spanned by the columns of $\mathbf{W}(n)$ and $\mathbf{e}(n)$ is additive Gaussian noise. In other words, besides the noise, a reasonable *overestimate* of the true rank of the unknown data subspace is considered in our data generation model.

To generalize our model, we further assume that a) the unknown subspace matrix $\mathbf{W}(n)$ may be sparse, a condition appearing in several applications and b) part of the entries of $\mathbf{y}(n)$ are missing. The latter means that what we actually have is not $\mathbf{y}(n)$, but $\mathbf{z}(n)$, where

$$\mathbf{z}(n)=\boldsymbol{\phi}(n)\odot\mathbf{y}(n)=\boldsymbol{\Phi}_n\mathbf{y}(n). \tag{3}$$

In (3), $\boldsymbol{\phi}(n)$ is a $\{0,1\}$-binary $K\times 1$ vector having 0's at the positions where $\mathbf{y}(n)$ has missing entries and 1's elsewhere and $\boldsymbol{\Phi}_n=\text{diag}(\boldsymbol{\phi}(n))$. If we now stack together observation vectors (with possible missing elements) up to time n, as *rows* in a $n\times K$ matrix $\mathbf{Z}(n)$, yields

$$\mathbf{Z}(n)=\boldsymbol{\Phi}(n)\odot\mathbf{Y}(n)=\boldsymbol{\Phi}(n)\odot\left(\mathbf{X}(n)\mathbf{W}^T(n)+\mathbf{E}(n)\right), \tag{4}$$

where

$$\mathbf{Z}(n)=[\mathbf{z}(1),\mathbf{z}(2),\ldots,\mathbf{z}(n)]^T=[\boldsymbol{z}_1(n),\boldsymbol{z}_2(n),\ldots,\boldsymbol{z}_K(n)], \tag{5}$$

$$\mathbf{Y}(n)=[\mathbf{y}(1),\mathbf{y}(2),\ldots,\mathbf{y}(n)]^T=[\boldsymbol{y}_1(n),\boldsymbol{y}_2(n),\ldots,\boldsymbol{y}_K(n)], \tag{6}$$

$$\boldsymbol{\Phi}(n)=[\boldsymbol{\phi}(1),\boldsymbol{\phi}(2),\ldots,\boldsymbol{\phi}(n)]^T=[\boldsymbol{\varphi}_1(n),\boldsymbol{\varphi}_2(n),\ldots,\boldsymbol{\varphi}_K(n)], \tag{7}$$

$$\mathbf{X}(n)=[\mathbf{x}(1),\mathbf{x}(2),\ldots,\mathbf{x}(n)]^T=[\boldsymbol{x}_1(n),\boldsymbol{x}_2(n),\ldots,\boldsymbol{x}_L(n)] \tag{8}$$

and $\mathbf{E}(n)=[\mathbf{e}(1),\mathbf{e}(2),\ldots,\mathbf{e}(n)]^T$. In addition, we define the subspace matrix $\mathbf{W}(n)$ row- and columnwise as[2]

$$\mathbf{W}(n)=[\mathbf{w}_1(n),\mathbf{w}_2(n),\ldots,\mathbf{w}_K(n)]^T=[\boldsymbol{w}_1(n),\boldsymbol{w}_2(n),\ldots,\boldsymbol{w}_L(n)]. \tag{9}$$

It can be noticed from Eqs. (5)–(9) that the row size of matrices $\mathbf{Z}(n)$, $\mathbf{Y}(n)$, $\boldsymbol{\Phi}(n)$ and $\mathbf{X}(n)$ increases with time, while $\mathbf{W}(n)$ is a time-varying fixed size $K\times L$ matrix.

The goals of this work are a) the estimation and tracking of the underlying low-dimensional subspace where measurement data reside, b) the estimation of the low-rank representation of data in this subspace in time and, as a by-product, c) the recovery of the complete measurement data matrix $\mathbf{Y}(n)$ via online matrix completion. In this context, given the batch of incomplete data $\mathbf{Z}(n)$, we aim at estimating the unknown low-rank subspace matrix $\mathbf{W}(n)$ and the latent matrix of projections $\mathbf{X}(n)$ in this subspace. However, in case of streamingly received data, the use of a batch iterative solver entails the processing of the whole bunch of data that are available up to every time instant, rendering the whole procedure computationally prohibitive and thus practically infeasible. A way to alleviate this impediment is by employing online data handling, whereby incomplete observation vectors $\mathbf{z}(n)$ are acquired and processed sequentially to learn and track $\mathbf{W}(n)$ and provide estimates of the vectors of coefficients $\mathbf{x}(n)$.

In the following, we tackle the aforementioned problem using a Bayesian approach. First, an appropriate Bayesian model is defined that effectively promotes the low-rankness of the sought subspace through column sparsity inducing Laplace priors. As it will become clear below, the adopted modeling aims at revealing the true data subspace (spanned by the columns of $\mathbf{U}(n)$) and its true rank $r(n)$, starting from an overestimate L of it. Based on the proposed Bayesian model, a variational Bayes batch iterative subspace estimation algorithm is developed, which after suitable adjustments leads to an efficient online subspace learning scheme.

3. The proposed Bayesian model

To develop a Bayesian inference method, first a Bayesian model must be defined consisting of a) the likelihood function of the data and b) suitable priors assigned to the parameters of the model. The likelihood function of the observed data depends on the statistical properties of the additive noise, which is commonly taken to be Gaussian with zero mean and constant variance. In this work, in order to place more importance on recent data and downgrade older measurements which is meaningful under time-varying conditions, we employ a so-called forgetting factor λ with $0\ll\lambda<1$ and define the noise distribution as,[3]

$$\mathbf{E}(n)=\prod_{i=1}^{n}\mathcal{N}(\mathbf{e}(i)|\mathbf{0},\beta^{-1}\lambda^{i-n}\mathbf{I}_K), \tag{10}$$

where β is a noise precision parameter, while we define

$$\boldsymbol{\Lambda}(n)=\text{diag}\left(\left[\lambda^{n-1},\lambda^{n-2},\ldots,\lambda,1\right]^T\right). \tag{11}$$

In the following, whenever not necessary, the time index n is omitted to simplify derivations. The time index is reestablished in Section 5, where the new online subspace estimation algorithm is presented. In this context, based on (4) and the noise distribution given in (10), the likelihood function of the measurement data is expressed as

$$\begin{aligned}p(\mathbf{Z}\mid\mathbf{X},\mathbf{W},\beta)&=\prod_{i=1}^{n}p(\mathbf{z}(i)\mid\mathbf{x}(i),\mathbf{W},\beta)\\&=\prod_{i=1}^{n}\prod_{k\in\mathcal{I}_{\boldsymbol{\phi}(i)}}\mathcal{N}(z_k(i)\mid\mathbf{w}_k^T\mathbf{x}(i),\beta^{-1}\lambda^{i-n}),\end{aligned} \tag{12}$$

where $\mathcal{I}_{\boldsymbol{\phi}(i)}$ is the set of indices for which the corresponding entries of $\boldsymbol{\phi}(i)$ are 1.

[2] Recall that in (5)–(9), small boldface calligraphic letters have been used to denote columns of matrices and regular boldface letters to denote rows.

[3] From (10), more recent error vectors have smaller variance compared to older ones, which is equivalent to giving more reliability to recent measurements than to older ones.

Now that the likelihood function has been defined, we proceed by presenting the prior distributions imposed on the subspace matrix $\mathbf{W}$ and the coefficients matrix $\mathbf{X}$. These priors aim at simultaneously decreasing the rank and imposing sparsity on the unknown subspace matrix $\mathbf{W}$. Recall that the matrix product $\mathbf{X}\mathbf{W}^T$ in (4) is equivalently written as the sum of the outer products between the columns of $\mathbf{X}$ and $\mathbf{W}$ i.e.,

$$\mathbf{X}\mathbf{W}^T = \sum_{l=1}^{L} \boldsymbol{x}_l \boldsymbol{w}_l^T. \tag{13}$$

From (13) it is readily seen that the rank of the matrix $\mathbf{X}\mathbf{W}^T$ equals to the number L of the rank-one terms existing into this summation. Hence, a natural approach to reduce the rank L of the sought subspace is to somehow eliminate some of the rank-one contributing terms in (13). A relevant scheme, [18], reduces the rank by imposing column sparsity *jointly* on $\mathbf{X}$ and $\mathbf{W}$. Herein, as in [18], this sparsity constraint is integrated in the modeling of the prior distributions of $\boldsymbol{x}_l$ and $\mathbf{w}_l$, as explained below. At the same time, as stated earlier, in several applications (e.g. [27,28]) the subspace matrix $\mathbf{W}$ is required to be sparse. That said, joint sparsity on $\boldsymbol{x}_l$ and $\mathbf{w}_l$ and the sparse structure on subspace matrix $\mathbf{W}$ are simultaneously incorporated in the modeling process of the corresponding prior distributions. In light of this, three-level hierarchical priors[4] are assigned to the columns of $\mathbf{X}$ and $\mathbf{W}$. At the first level of hierarchy the following Gaussian priors are defined,

$$p(\mathbf{X} \mid \mathbf{s}, \beta) = \prod_{l=1}^{L} \mathcal{N}(\boldsymbol{x}_l \mid \mathbf{0}, \beta^{-1} s_l^{-1} \boldsymbol{\Lambda}^{-1}), \tag{14}$$

$$p(\mathbf{W} \mid \mathbf{s}, \boldsymbol{\Gamma}, \beta) = \prod_{l=1}^{L} \mathcal{N}(\boldsymbol{w}_l \mid \mathbf{0}, \beta^{-1} s_l^{-1} \boldsymbol{\Gamma}_l^{-1}), \tag{15}$$

where $\mathbf{s} = [s_1, s_2, \ldots, s_L]^T$, $\boldsymbol{\Gamma} = [\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \ldots, \boldsymbol{\gamma}_L]$, $\boldsymbol{\gamma}_l = [\gamma_{1l}, \gamma_{2l}, \ldots, \gamma_{Kl}]^T$ and $\boldsymbol{\Gamma}_l = \mathrm{diag}(\boldsymbol{\gamma}_l)$ for $l = 1, 2, \ldots, L$. It can be observed from (14) and (15) that the lth columns of $\mathbf{X}$ and $\mathbf{W}$ share the same *joint sparsity promoting parameters* s_l's. At the same time, the diagonal matrix $\boldsymbol{\Gamma}_l$ which appears in the prior distribution of $\mathbf{W}$ is responsible for *independently* imposing sparsity on the entries of the lth column of the subspace matrix.[5] In particular, some of the s_l's take large values when Bayesian inference is performed and as a result, both the lth columns of $\mathbf{X}$ and $\mathbf{W}$ are driven to zero. Notably, in cases where a parameter s_l does not enforce joint sparsity, the kth element of the lth column of $\mathbf{W}$ may be independently led to zero by the corresponding *subspace sparsity promoting* parameter γ_{kl} of $\boldsymbol{\Gamma}_l$. It should be also noted that the exponentially weighting matrix $\boldsymbol{\Lambda}$ appears in the prior of $\mathbf{X}$, but not in that of $\mathbf{W}$. This is so because in a streaming data environment the size of $\mathbf{X}$ is time-increasing while the fixed-size data subspace matrix $\mathbf{W}$ is estimated based not only on the most recent row $\mathbf{x}(n)$, but also on the previous rows of $\mathbf{X}$ with appropriate weighting. On the other hand, in such an online scenario, the current projection coefficients vector $\mathbf{x}(n)$ shall be estimated only from the more recent estimate of $\mathbf{W}$, which, being fixed-size, does not have to be exponentially weighted.

The prior distribution of $\mathbf{X}$ in (14) can be written in an equivalent form with respect to the rows of $\mathbf{X}$ as follows

$$p(\mathbf{X} \mid \mathbf{s}, \beta) = \prod_{i=1}^{n} \mathcal{N}(\mathbf{x}(i) \mid \mathbf{0}, \beta^{-1} \lambda^{i-n} \mathbf{S}^{-1}), \tag{16}$$

[4] Hierarchical priors are required in order to ensure *conjugacy* with respect to the likelihood as well as among them, which is a prerequisite for deriving a tractable posterior inference procedure, [3,22].

[5] In case $\mathbf{W}$ is not sparse, we set $\boldsymbol{\Gamma}_l = \mathbf{I}_K$ in (15) and no prior applies to $\boldsymbol{\Gamma}$, i.e. Eqs (18) and (20) are needless.

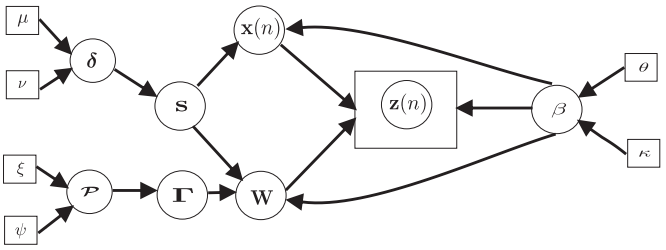


Fig. 1. Directed acyclic graph of the proposed Bayesian model.

where $\mathbf{S} = \mathrm{diag}(\mathbf{s})$. Note that it is the form of the prior in (16) that is mainly used in the analysis of the next sections, although (14) serves in this section to show how the rank is reduced by the proposed model. At the second level of the hierarchy we define the following conjugate inverse Gamma distributions for $\mathbf{s}$ and $\boldsymbol{\Gamma}$,

$$p(\mathbf{s} \mid \boldsymbol{\delta}) = \prod_{l=1}^{L} \mathcal{IG}(s_l \mid \frac{K+n+1}{2}, \frac{\delta_l}{2}), \tag{17}$$

$$p(\boldsymbol{\Gamma} \mid \mathcal{P}) = \prod_{k=1}^{K} \prod_{l=1}^{L} \mathcal{IG}(\gamma_{kl} \mid 1, \frac{\rho_{kl}}{2}). \tag{18}$$

where $\boldsymbol{\delta} = [\delta_1, \delta_2, \ldots, \delta_L]^T$ and $\mathcal{P}$ is the $K \times L$ matrix whose entries are the ρ_{kl}'s. Finally, at the third level of the hierarchy, conjugate Gamma distributions are defined for the scale parameters δ_l's and ρ_{kl}'s, i.e.

$$p(\delta_l) = \mathcal{G}(\delta_l; \mu, \nu) \tag{19}$$

$$p(\rho_{kl}) = \mathcal{G}(\rho_{kl}; \psi, \xi). \tag{20}$$

By integrating out $\mathbf{s}$ from (14) and (15) using (17) with $\boldsymbol{\Gamma}$ kept fixed, we are led to a heavy-tailed multiparameter Laplace-type distribution for the joint prior of $\mathbf{X}$ and $\mathbf{W}$ that promotes joint column sparsity, as is shown in Appendix B. Similarly, by fixing $\mathbf{s}$, from (15) and (18) we get a multiparameter Laplace prior that imposes sparsity on $\mathbf{W}$.

The proposed Bayesian model is concluded by assigning a conjugate to the likelihood Gamma prior to the precision of the noise β as follows,

$$p(\beta) = \mathcal{G}(\beta; \kappa, \theta). \tag{21}$$

It should be noted that the proposed Bayesian model, which is built upon the likelihood (12) and the priors (14)–(21), differs considerably and improves over the relevant model reported in [18]. The novelties of the new model come from a) the promotion of sparsity on $\mathbf{W}$ aside from low-rank through the use of the parameter matrix $\boldsymbol{\Gamma}$, b) the (necessary for online processing) exponential weighting of the data by incorporating a forgetting factor in the likelihood and the prior of $\mathbf{X}$ and c) the adoption of Laplace-type marginal priors for $\mathbf{X}$ and $\mathbf{W}$, instead of Student-t used in [18], in order to promote sparsity and low-rankness. In the next section, based on the multi-hierarchical model described in this section and presented graphically in Fig. 1, an approximate Bayesian inference scheme is derived for low-rank sparse subspace learning from partial observations.

4. Batch variational Bayes inference

Inferring the joint posterior distribution of multiple variables given the data boils down to an intractable process when it comes to composite Bayesian models, such as those springing from hierarchical dependences of the involved variables, which are modeled

by suitable priors. This is also the case for the Bayesian model described in the previous section, and graphically depicted in Fig. 1. Following the Bayes' theorem, the exact joint posterior of our variables given the observations is obtained by

$$p(\mathbf{X},\mathbf{W},\mathbf{s},\mathbf{\Gamma},\boldsymbol{\delta},\mathcal{P},\beta \mid \mathbf{Z}) = \frac{p(\mathbf{Z},\mathbf{X},\mathbf{W},\mathbf{s},\mathbf{\Gamma},\boldsymbol{\delta},\mathcal{P},\beta)}{\int p(\mathbf{Z},\mathbf{X},\mathbf{W},\mathbf{s},\mathbf{\Gamma},\boldsymbol{\delta},\mathcal{P},\beta)d\mathbf{X}d\mathbf{W}d\mathbf{s}d\mathbf{\Gamma}d\boldsymbol{\delta}d\mathcal{P}d\beta}. \tag{22}$$

Apparently, getting a closed form expression for the posterior given in (22) involves the daunting task of estimating the integral at the denominator. To obviate obstacles of this type, plentiful approximate inference schemes have come to light in literature, [29,30]. Herein, the ubiquitous variational Bayes inference approach is adopted, [20]. The basic premise of this approach inspired from the field of statistical physics is the assumption that the posterior distribution can be approximately expressed in a factorized form. Based on this particular hypothesis, the exact joint posterior $p(\mathbf{X},\mathbf{W},\mathbf{s},\mathbf{\Gamma},\boldsymbol{\delta},\mathcal{P},\beta \mid \mathbf{Z})$ is approximated by $q(\mathbf{X},\mathbf{W},\mathbf{s},\mathbf{\Gamma},\boldsymbol{\delta},\mathcal{P},\beta)$, defined as

$$q(\mathbf{X},\mathbf{W},\mathbf{s},\mathbf{\Gamma},\boldsymbol{\delta},\mathcal{P},\beta) = q(\beta)\prod_{i=1}^{n} q(\mathbf{x}(i))\prod_{k=1}^{K}\prod_{l=1}^{L} q(w_{kl})\prod_{l=1}^{L} q(s_l) \times \prod_{l=1}^{L} q(\delta_l)\prod_{k=1}^{K}\prod_{l=1}^{L} q(\gamma_{kl})q(\rho_{kl}). \tag{23}$$

From (23) it is easily noticed that there has been considered full statistical *a posteriori* independence among the rows of $\mathbf{X}$, as well as among all the elements of the subspace matrix $\mathbf{W}$. As far as $\mathbf{x}(i)$'s are concerned, being statistical independent is something that is naturally brought up due to the presumed independence among the corresponding observation vectors $\mathbf{z}(i)$'s. On the other hand and in contrast to previous related works (e.g. [18]), posterior independence is imposed on the entries of $\mathbf{W}$ in (23). This gives rise to coordinate-descent recursions for retrieving w_{kl}'s, which, as shown later, reduces significantly the computational complexity of the online subspace estimation task. Notably, as implied by (23), those explicit assumptions on the independence among the rows of $\mathbf{X}$ and the elements of $\mathbf{W}$ dictate relevant statistical independence on the variables of our model belonging to the second and the third level of hierarchy, namely $\mathbf{s}$, $\boldsymbol{\delta}$, $\mathbf{\Gamma}$ and $\mathcal{P}$.

In an attempt to bring to light the particular way that the posterior distributions $q(\cdot)$'s of all variables in (23) are recovered according to the variational Bayes scheme, we define the cell array $\boldsymbol{\theta} = \{\mathbf{x}(1),\cdots,\mathbf{x}(n), w_{11},\cdots,w_{nK}, s_1,\ldots,s_L,\gamma_{11},\ldots,\gamma_{KL},\delta_1,\ldots,\delta_L,p_{11},\ldots,p_{KL}\}$.[6] The posterior distribution $q(\theta_i)$ of each component θ_i is then obtained by minimizing the Kullback–Leibler distance between the posterior i.e. $p(\boldsymbol{\theta}|\mathbf{Z})$, and the approximate one $q(\boldsymbol{\theta})$ leading to the following closed-form expressions [20]

$$q(\boldsymbol{\theta}_i) = \frac{\exp\left(\langle \ln p(\mathbf{Z},\boldsymbol{\theta})\rangle_{i\neq j}\right)}{\int \exp\left(\langle \ln p(\mathbf{Z},\boldsymbol{\theta})\rangle_{i\neq j}\right)d\boldsymbol{\theta}_i}. \tag{24}$$

In the last equation $\langle\cdot\rangle_{i\neq j}$ denotes expectation taken with respect to all $q(\boldsymbol{\theta}_j)$'s but $q(\theta_i)$. Interestingly, through (24) the parameters of each posterior $q(\theta_i)$ are expressed in terms of the parameters of the other distributions $q(\boldsymbol{\theta}_j)$'s, for $j \neq i$. Thus, the minimization of the Kullback–Leibler distance gives birth to a cyclic iterative scheme, whereby the parameters of each $q(\theta_i)$ are computed based on the most recent estimates of the parameters of the rest $q(\boldsymbol{\theta}_j)$'s, as it will also become more clear below. This procedure is applied for our three-level hierarchical Bayesian model and the whole derivation is provided in Appendix A.

[6] Note that for notational convenience, the entries of $\boldsymbol{\theta}$ i.e. the $\boldsymbol{\theta}_i$'s may represent either vectors or scalars.

Note that due to the novelty of the proposed Bayesian model and the assumed posterior independence of the entries of $\mathbf{W}$, Eqs. (A.5)–(A.23) derived in Appendix A are new. The mutual dependence among the moments of all the model parameters, that can be easily observed in the respective expressions, paves the way for an iterative scheme over the involved quantities. It should be emphasized though that since we aim at handling a massive amount of streaming data, the utilization of those expressions ends up to be a prohibitive task. More specifically, as the number n of the observations increases, calculations that involve quantities such as $\mathbf{Z}$, $\mathbf{X}$, become increasingly demanding in terms of the memory storage as well as the computational effort needed. In light of this, an online scheme is presented in the next section, that favorably adjusts the above defined expressions to the streaming processing scenario.

5. Online variational Bayes subspace estimation

In this section we derive a new online variational Bayes algorithm for sparse and low-rank subspace estimation from incomplete data. As shown below, moving from the batch to the online scenario is not a trivial task. It requires the definition of appropriate fixed-size quantities that can be recursively updated and their combination with other formulas coming from the batch algorithm in a cohesive scheme. According to the online scenario, incomplete high dimensional datums $\mathbf{z}(n)$'s are streamingly received at each time instance n. Then, the proposed algorithm proceeds by a) computing an estimate $\hat{\mathbf{x}}(n)$ of the coefficients vector of the observations on the subspace acquired in the previous iteration (i.e. $\hat{\mathbf{W}}(n-1)$) and next b) updating *elementwise* the subspace matrix $\hat{\mathbf{W}}(n-1)$ to $\hat{\mathbf{W}}(n)$. In the sequel, for notational convenience, we disregard the expectation operator $\langle\cdot\rangle$. Then, with a slight but straightforward abuse of notation and by handling the time index appropriately, we get from (A.2) to (A.4),

$$\hat{\mathbf{x}}(n) = \beta(n-1)\mathbf{\Sigma}_{\hat{\mathbf{x}}}(n)\hat{\mathbf{W}}^T(n-1)\mathbf{z}(n), \tag{25}$$

$$\mathbf{\Sigma}_{\hat{\mathbf{x}}}(n) = \beta^{-1}(n-1)\left(\hat{\mathbf{W}}^T(n-1)\mathbf{\Phi}_n\hat{\mathbf{W}}(n-1) + \sum_{k=1}^{K}\phi_k(n)\mathbf{\Sigma}_{\hat{\mathbf{w}}_k}(n-1) + \mathbf{S}(n-1)\right)^{-1}. \tag{26}$$

Next, we define the following *fixed-size with respect to time* quantities,

$$\mathbf{T}(n) = \hat{\mathbf{X}}^T(n)\mathbf{\Lambda}(n)\mathbf{Z}(n), \tag{27}$$

$$\mathbf{Q}(n) = \hat{\mathbf{X}}^T(n)\mathbf{\Lambda}(n)\hat{\mathbf{X}}(n) + \sum_{i=1}^{n}\lambda^{n-i}\mathbf{\Sigma}_{\hat{\mathbf{x}}}(i), \tag{28}$$

and for $k = 1, 2, \ldots, K$,

$$\mathbf{P}_k(n) = \hat{\mathbf{X}}^T(n)\mathbf{\Lambda}(n)\mathbf{\Phi}_k(n)\hat{\mathbf{X}}(n) + \sum_{i=1}^{n}\lambda^{n-i}\phi_k(i)\mathbf{\Sigma}_{\hat{\mathbf{x}}}(i), \tag{29}$$

$$d_k(n) = \boldsymbol{z}_k^T(n)\mathbf{\Lambda}(n)\boldsymbol{z}_k(n). \tag{30}$$

The basic idea in any online scheme is the formulation of the various quantities that carry the past knowledge of the relevant process in a time-recursive manner. Interestingly, Eqs. (27)–(30) can easily be written in time-recursive forms i.e.,

$$\mathbf{T}(n) = \lambda\mathbf{T}(n-1) + \hat{\mathbf{x}}(n)\mathbf{z}^T(n), \tag{31}$$

$$\mathbf{Q}(n) = \lambda\mathbf{Q}(n-1) + \mathbf{\Sigma}_{\hat{\mathbf{x}}}(n) + \hat{\mathbf{x}}(n)\hat{\mathbf{x}}^T(n), \tag{32}$$

$$\mathbf{P}_k(n) = \lambda \mathbf{P}_k(n-1) + \phi_k(n)\left(\mathbf{\Sigma}_{\hat{\mathbf{x}}}(n) + \hat{\mathbf{x}}(n)\hat{\mathbf{x}}^T(n)\right), \tag{33}$$

$$d_k(n) = \lambda d_k(n-1) + z_k^2(n). \tag{34}$$

Moreover, for $k = 1, 2, \ldots, K$, we define the following matrices that stem from $\mathbf{P}_k(n)$'s with the addition of appropriate diagonal terms,

$$\mathbf{R}_k(n) = \mathbf{P}_k(n) + \mathbf{\Gamma}_k(n-1)\mathbf{S}(n-1). \tag{35}$$

Having aptly obtained the above computationally efficient formulas, we can now head for online processing. Towards this, the equations derived for the batch case are suitably modified by incorporating the previously defined recursively computed quantities. More specifically, by substituting (A.8), (A.9) in (A.6), (A.7) respectively and using (27), (29) and (35) we get the following time update expressions for the entries of the subspace matrix estimate $\hat{\mathbf{W}}$ at time n,

$$\hat{w}_{kl}(n) = \beta(n-1)\sigma^2_{\hat{w}_{kl}}(n-1)\left(t_{lk}(n) - \mathbf{r}^T_{k\neg l}(n)\hat{\mathbf{w}}_{k\neg l}(n)\right), \tag{36}$$

$$\sigma^2_{\hat{w}_{kl}}(n) = \beta^{-1}(n-1)r^{-1}_{k,ll}(n), \tag{37}$$

where $t_{lk}(n)$ is the lkth entry of the $L \times K$ matrix $\mathbf{T}(n)$, $\mathbf{r}^T_{k\neg l}(n)$ is the lth row of $L \times L$ autocorrelation matrix $\mathbf{R}_k(n)$ after neglecting its lth element i.e. $r_{k,\,ll}$ and finally

$$\hat{\mathbf{w}}_{k\neg l}(n) = [\hat{w}_{k1}(n), \hat{w}_{k2}(n), \ldots, \hat{w}_{kl-1}(n), \hat{w}_{kl+1}(n-1), \ldots, \hat{w}_{kL}(n-1)]^T. \tag{38}$$

From (36) and (38) it is readily seen that each element of the kth row of $\mathbf{W}$ is updated at each time instance n, taking into account the most recent estimates of the remaining entries of the kth row in a cyclic manner. It is worthy to mention that this emerging iterative scheme, resulting from the espoused statistical independence among the elements of $\mathbf{W}$, can be viewed as a relevant to the cyclic coordinate-descent strategy [31]. Following the same premise, for the column sparsity promoting parameters we get from (A.11),

$$s_l(n) = \sqrt{\frac{\beta^{-1}(n-1)\delta_l(n)}{\hat{\mathbf{w}}_l^T(n)\mathbf{\Gamma}_l(n)\hat{\mathbf{w}}_l(n) + \sum_{k=1}^K \gamma_{kl}(n)\sigma^2_{\hat{w}_{kl}}(n) + q_{ll}(n)}}, \tag{39}$$

where $q_{ll}(n)$ is the lth diagonal element of $\mathbf{Q}(n)$. As for the hyperparameters δ_l's of the s_l's we have from (A.16), (A.18) the following recursive equation

$$\delta_l(n) = \frac{2\mu + (1-\lambda)^{-1} + K + 1}{2\nu + s_l^{-1}(n-1) + \delta_l^{-1}(n-1)}. \tag{40}$$

Note that in (40) the size of the effective time window i.e. $(1-\lambda)^{-1}$, is used in place of n, as in [22]. For γ_{kl}'s that independently favor sparsity on the entries of the subspace matrix $\mathbf{W}$, in an online scheme (A.13) takes the form,

$$\gamma_{kl}(n) = \sqrt{\frac{\rho_{kl}(n)}{\beta(n-1)s_l(n-1)\left(\hat{w}^2_{kl}(n) + \sigma^2_{\hat{w}_{kl}}(n)\right)}} \tag{41}$$

and for the hyperparameters ρ_{kl}'s, (A.17) and (A.19) yield

$$\rho_{kl}(n) = \frac{2(\psi+1)}{2\xi + \gamma_{kl}^{-1}(n-1) + \rho_{kl}^{-1}(n-1)}. \tag{42}$$

Finally, from (A.20) to (A.23) and applying some straightforward algebraic manipulations as in [22], we end up with the following efficient formula for computing the noise precision β, at each time iteration

$$\beta(n) = \frac{2\kappa + \frac{1}{1-\lambda}(K+L) + KL}{\left(2\theta + \sum_{k=1}^K \left(d_k(n) - \hat{\mathbf{w}}_k^T(n)\boldsymbol{t}_k(n) + \boldsymbol{\sigma}^T_{\hat{\mathbf{w}}_k}(n)\mathbf{r}_k(n)\right) + \sum_{l=1}^L s_l(n)q_{ll}(n)\right)} \tag{43}$$

where $\hat{\mathbf{w}}_k^T(n)$ is the kth row of $\hat{\mathbf{W}}(n)$, $\boldsymbol{t}_k(n)$ is the kth column of $\mathbf{T}(n)$, $\boldsymbol{\sigma}_{\hat{\mathbf{w}}_k}(n) = \mathrm{diag}(\mathbf{\Sigma}_{\hat{\mathbf{w}}_k}(n))$ and $\mathbf{r}_k(n) = \mathrm{diag}(\mathbf{R}_k(n))$.

As it can be seen, most of the above defined quantities resolve to efficient time-updating formulas. In doing so, the need for taking into consideration the whole bunch of data, which is computationally prohibitive in applications dealing with big data, is eliminated. By collecting and putting in a proper order the previously derived expressions, we are led to the new online variational Bayes sparse subspace learning (OVBSL) algorithm, which is summarized in Table 1. The algorithm provides at each time iteration not only the sought estimates $\hat{\mathbf{x}}(n)$ and $\hat{\mathbf{W}}(n)$, but also estimates for all parameters of the model described in Section 3. Note also that all these parameters are directly linked to specific distributions through the posterior inference analysis of Section 3. By carefully inspecting OVBSL in Table 1, it can be shown that its computational complexity is $\mathcal{O}(|\boldsymbol{\phi}(n)|L^2 + KL)$, where $|\boldsymbol{\phi}(n)|$ is the number of observed entries at time n. It should be emphasized that a significant reduction in the computational complexity has been achieved (which would be otherwise $\mathcal{O}(|\boldsymbol{\phi}(n)|L^3)$) by adopting the

Table 1
The OVBSL algorithm.

Initialize : $\hat{\mathbf{W}}(0), \mathbf{S}(0), \beta(0), \mathbf{\Gamma}_k(0), \mathbf{\Sigma}_{\hat{w}_k}(0), k = 1, 2, \ldots, K$
Set $\mathbf{T}(0) = \mathbf{0}, \mathbf{P}_k(0) = \mathbf{0}, d_k(0) = 0, k = 1, 2, \ldots, K$
Set $\mu = 10^{-6}, \nu = 10^{-6}, \psi = 10^{-6}, \xi = 10^{-6}, \kappa = 10^{-6}, \theta = 10^{-6}$
Set $\mathbf{Q}(0) = \mathbf{0}, \lambda$
for $n = 1, 2, \ldots$
Get$\mathbf{z}(n)$, $\boldsymbol{\phi}(n)$
$\mathbf{\Sigma}_{\hat{\mathbf{x}}}(n) = \beta^{-1}(n-1)\left(\hat{\mathbf{W}}^T(n-1)\mathbf{\Phi}_n\hat{\mathbf{W}}(n-1) + \sum_{k=1}^K \phi_k(n)\mathbf{\Sigma}_{\hat{\mathbf{w}}_k}(n-1) + \mathbf{S}(n-1)\right)^{-1}$
$\hat{\mathbf{x}}(n) = \beta(n-1)\mathbf{\Sigma}_{\hat{\mathbf{x}}}(n)\hat{\mathbf{W}}^T(n-1)\mathbf{z}(n)$
$\mathbf{\Sigma}(n) = \mathbf{\Sigma}_{\hat{\mathbf{x}}}(n) + \hat{\mathbf{x}}^T(n)\hat{\mathbf{x}}(n)$
$\mathbf{Q}(n) = \lambda\mathbf{Q}(n-1) + \mathbf{\Sigma}(n)$
$\mathbf{T}(n) = \lambda\mathbf{T}(n-1) + \hat{\mathbf{x}}(n)\mathbf{z}^T(n)$
for $k = 1, 2, \ldots, K$,
$\mathbf{P}_k(n) = \lambda\mathbf{P}_k(n-1) + \phi_k(n)\mathbf{\Sigma}(n)$
$\mathbf{R}_k(n) = \mathbf{P}_k(n) + \mathbf{\Gamma}_k(n-1)\mathbf{S}(n-1)$
$d_k(n) = \lambda d_k(n-1) + z_k^2(n)$
for $l = 1, 2, \ldots, L$,
$\hat{w}_{kl}(n) = \beta(n-1)\sigma_{\hat{w}_{kl}}(n-1)\left(t_{lk}(n) - \mathbf{r}^T_{k\neg l}(n)\hat{\mathbf{w}}_{k\neg l}(n)\right)$
$\sigma^2_{\hat{w}_{kl}}(n) = \beta^{-1}(n-1)r^{-1}_{k,ll}(n)$
$\rho_{kl}(n) = \dfrac{2(\psi+1)}{2\xi + \gamma_{kl}^{-1}(n-1) + \rho_{kl}^{-1}(n-1)}$
$\gamma_{kl}(n) = \sqrt{\dfrac{\rho_{kl}(n)}{\beta(n-1)s_l(n-1)\left(\hat{w}^2_{kl}(n) + \sigma^2_{\hat{w}_{kl}}(n)\right)}}$
end
Set $\mathbf{\Sigma}_{\hat{\mathbf{w}}_k}(n) = \mathrm{diag}\left([\sigma^2_{\hat{w}_{k1}}(n), \sigma^2_{\hat{w}_{k2}}(n), \ldots, \sigma^2_{\hat{w}_{kL}}(n)]^T\right)$
end
for $l = 1, 2, \ldots, L$,
$\delta_l(n) = \dfrac{2\mu + (1-\lambda)^{-1} + K + 1}{2\nu + s_l^{-1}(n-1) + \delta_l^{-1}(n-1)}$
$s_l(n) = \sqrt{\dfrac{\beta^{-1}(n-1)\delta_l(n)}{\hat{\mathbf{w}}_l^T(n)\mathbf{\Gamma}_l(n)\hat{\mathbf{w}}_l(n) + \sum_{k=1}^K \gamma_{kl}(n)\sigma^2_{\hat{w}_{kl}}(n) + q_{ll}(n)}}$
end
Set $\mathbf{S}(n) = \mathrm{diag}([s_1(n), s_2(n), \ldots, s_L(n)]^T)$
$\beta(n) = \dfrac{2\kappa + \frac{1}{1-\lambda}(K+L) + KL}{\left(2\theta + \sum_{k=1}^K \left(d_k(n) - \hat{\mathbf{w}}_k^T(n)\boldsymbol{t}_k(n) + \boldsymbol{\sigma}^T_{\hat{\mathbf{w}}_k}(n)\mathbf{r}_k(n)\right) + \sum_{l=1}^L s_l(n)q_{ll}(n)\right)}$
end

element-by-element estimation of $\hat{\mathbf{W}}$ via a coordinate-descent type procedure. As shown in Table 1, all hyperparameters of OVBSL are set and fixed to very small values at the initialization stage of the algorithm, as is the custom in sparse Bayesian learning schemes [32]. This way, prior distributions become non-informative; in line with the fact that no information for the respective parameters is a priori available.[7] Hence parameter fine tuning or cross-validation is entirely avoided and all parameters of the model are inferred from the data, rendering the proposed algorithm ideally accustomed for use in a real-time setting. In the next section the proposed algorithm is set in a unified framework with other related state-of-the-art techniques and its advantages in terms of performance and complexity are highlighted.

6. Relation with state-of-the-art

In this section we investigate and highlight the connection of the new Bayesian algorithm with two other closely related techniques that have recently appeared in the literature, namely the PETRELS algorithm presented in [4] and Algorithm 1 of Mardani et al. [17]. All three algorithms under study are second-order online subspace learning schemes that deal with (possibly highly) incomplete data. Out of the three schemes, only the proposed algorithm has the provision to impose sparsity to the unknown subspace matrix. Hence, to make comparisons more clear we relax this constraint, that is we set $\mathbf{\Gamma}_l = \mathbf{I}_K$ for $l = 1, 2, \ldots, L$ in our Bayesian model described in Section 3. As we shall see below, this Bayesian model can be considered as a unified framework from which all three schemes may originate. To be more specific, let us first recall the likelihood function of the model given in (12), which can be expressed as

$$p(\mathbf{Z} \mid \mathbf{X}, \mathbf{W}, \beta) \propto \exp\left(-\frac{\beta}{2}\left\|\mathbf{\Lambda}^{\frac{1}{2}}(\mathbf{Z} - \mathbf{\Phi} \odot (\mathbf{X}\mathbf{W}^T))\right\|_F^2\right). \tag{44}$$

Based on (44), the maximum likelihood (ML) estimator is obtained by minimizing w.r.t $\mathbf{X}$ and $\mathbf{W}$ the negative log-likelihood, resulting in the following minimization problem

$$\text{(P1)} \qquad \min_{\mathbf{X},\mathbf{W}} \frac{\beta}{2}\left\|\mathbf{\Lambda}^{\frac{1}{2}}(\mathbf{Z} - \mathbf{\Phi} \odot (\mathbf{X}\mathbf{W}^T))\right\|_F^2$$

The so-termed PETRELS algorithm presented in [4] solves (P1) through an online alternating (between $\mathbf{X}$ and $\mathbf{W}$) least squares (LS) technique, which provides both the estimates of the subspace matrix $\mathbf{W}(n)$ and the new vector of projection coefficients $\mathbf{x}(n)$ at each time iteration. However, by solving (P1) PETRELS does not take any special care for revealing the true rank of the sought subspace. The algorithm starts with an overestimate L of the rank (number of columns of $\mathbf{W}$) and the estimates returned by the algorithm are related to a subspace of rank L, which may be far from the true rank.

Let us now consider the likelihood function given in (44) and the first level (Gaussian) priors of $\mathbf{X}$ and $\mathbf{W}$ in our model given by (14) and (15) for $s_l = s$, $l = 1, 2, \ldots, L$, where s is a constant parameter and not a random variable that can be determined from data. Then (14) and (15) are rewritten as,

$$p(\mathbf{X} \mid s, \beta) \propto \exp\left(-\frac{\beta}{2}s\left\|\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{X}\right\|_F^2\right), \tag{45}$$

$$p(\mathbf{W} \mid s, \beta) \propto \exp\left(-\frac{\beta}{2}s\|\mathbf{W}\|_F^2\right). \tag{46}$$

[7] Actually, since those parameters are placed in the third and the fourth level of hierarchy, their values have no crucial role on the estimation of parameters of our interest i.e., the first level ones.

From the likelihood (44) and the priors (45) and (46) the maximum a-posteriori probability (MAP) estimator of $\mathbf{X}$ and $\mathbf{W}$ defined as,

$$\min_{\mathbf{X},\mathbf{W}}\{-\log p(\mathbf{X}, \mathbf{W} \mid \mathbf{Z})\} \equiv \min_{\mathbf{X},\mathbf{W}}\{-\log[p(\mathbf{Z} \mid \mathbf{X}, \mathbf{W}, \beta) \times p(\mathbf{X} \mid s, \beta)p(\mathbf{W} \mid s, \beta)]\}, \tag{47}$$

is expressed as,

$$\text{(P2)} \quad \min_{\mathbf{X},\mathbf{W}} \frac{\beta}{2}\left[\left\|\mathbf{\Lambda}^{\frac{1}{2}}(\mathbf{Z} - \mathbf{\Phi} \odot (\mathbf{X}\mathbf{W}^T))\right\|_F^2 + s\left\|\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{X}\right\|_F^2 + s\|\mathbf{W}\|_F^2\right].$$

The minimization problem (P2) is at the heart of the analysis in [17]. Algorithm 1 of Mardani et al. [17] is a second-order alternating ridge regression type scheme that solves (P2) sequentially and provides estimates of $\mathbf{W}(n)$ and $\mathbf{x}(n)$ at each time iteration. In [17], to promote the low-rank data representation, the minimization problem is originally formulated as

$$\text{(P2}') \quad \min_{\mathbf{V}} \beta\left[\frac{1}{2}\left\|\mathbf{\Lambda}^{\frac{1}{2}}(\mathbf{Z} - \mathbf{\Phi} \odot \mathbf{V})\right\|_F^2 + s\left\|\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{V}\right\|_*\right].$$

Then, in search for a nuclear-norm surrogate that would be amenable to online processing, $||\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{V}||_*$ in (P2′) is replaced by its upper bound $(||\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{X}||_F^2 + ||\mathbf{W}||_F^2)/2$, with $\mathbf{V} = \mathbf{X}\mathbf{W}^T$, thus leading to (P2). Even though, compared to PETRELS, a more direct promotion of the low-rankness of the underlying subspace is employed in [17], again an overestimate L of the true rank is used and Algorithm 1 of Mardani et al. [17] lacks a specific mechanism for imposing low-rankness explicitly by reducing the initial rank to the true rank as the algorithm evolves. In addition, special care should be taken for the parameter s that must be properly selected and updated in the framework of an online scheme.

Let us, finally, employ the complete Bayesian model of Section 3 (with the exception of the subspace matrix sparsity promoting parameters γ_{kl}'s which are set to 1). In such a case, as shown in Appendix B, the joint prior of $\mathbf{X}$ and $\mathbf{W}$ can be expressed as

$$p(\mathbf{X}, \mathbf{W} \mid \boldsymbol{\delta}, \beta) \propto \exp\left(-\beta^{\frac{1}{2}}\sum_{l=1}^{L}\delta_l^{\frac{1}{2}}\left(\|\mathbf{x}_l\|_{2,\mathbf{\Lambda}}^2 + \|\mathbf{w}_l\|_2^2\right)^{\frac{1}{2}}\right). \tag{48}$$

From (44) and (48) the MAP estimator for $\mathbf{X}$ and $\mathbf{W}$ is now obtained from the solution of the following minimization problem,

$$\text{(P3)} \quad \min_{\mathbf{X},\mathbf{W}}\left[\frac{\beta}{2}\left\|\mathbf{\Lambda}^{\frac{1}{2}}(\mathbf{Z} - \mathbf{\Phi} \odot (\mathbf{X}\mathbf{W}^T))\right\|_F^2 + \beta^{\frac{1}{2}}\sum_{l=1}^{L}\delta_l^{\frac{1}{2}}\left(\|\mathbf{x}_l\|_{2,\mathbf{\Lambda}}^2 + \|\mathbf{w}_l\|_2^2\right)^{\frac{1}{2}}\right].$$

Note that the regularizing summation term in (P3) corresponds to the *weighted* ℓ_2/ℓ_1 norm of the matrix $[(\mathbf{\Lambda}^{\frac{1}{2}}\mathbf{X})^T \ \mathbf{W}^T]^T$ [33], which is known to impose column sparsity [33] and thus explicitly reducing the rank of $\mathbf{W}$, leading to more consistent estimates. Derived from the Bayesian model of Section 3, the minimization problem (P3) is closely related to the analysis and the algorithm presented in the current paper. It should be emphasized though that the proposed algorithm is not a recursive alternating MAP estimation scheme, but a variational Bayes type technique that can be deemed as a generalization of the MAP approach. While a MAP procedure would provide the point estimates of the parameters of interest $\mathbf{X}$ and $\mathbf{W}$, the proposed algorithm returns in addition the approximate distribution of all parameters involved in the model, including the weighting parameters δ_l's, which are now estimated directly from the data. Summarizing and compared to Chi et al. [4] and Mardani et al. [17] the proposed algorithm a) is equipped with an inherent mechanism for inducing column sparsity and

Table 2
Computational complexity and memory storage requirements of online subspace learning algorithms.

Algorithm	GROUSE [7]	PETRELS [4]	Algorithm 1 of Mardani et al. [17]	Algorithm 2 of Mardani et al. [17]	OVBSL
Comp. complexity	$\mathcal{O}(\lvert\phi(n)\rvert L^2+KL)$	$\mathcal{O}(\lvert\phi(n)\rvert L^2)$	$\mathcal{O}(\lvert\phi(n)\rvert L^3)$	$\mathcal{O}(\lvert\phi(n)\rvert L^2+KL)$	$\mathcal{O}(\lvert\phi(n)\rvert L^2+KL)$
Memory requirements	$\mathcal{O}(KL)$	$\mathcal{O}(KL^2)$	$\mathcal{O}(KL^2)$	$\mathcal{O}(KL)$	$\mathcal{O}(KL^2)$

thus reducing the rank of the latent subspace matrix dynamically and b) is fully automatic as all parameters of the model are estimated from the data and thus any need for preselection (using heuristics) or fine tuning is entirely avoided.

In Table 2, OVBSL is compared in terms of computational complexity and memory storage requirements with other related state-of-the-art algorithms. Besides PETRELS and Algorithm 1 of Mardani et al. [17] mentioned above, two other algorithms are included, namely GROUSE reported in [7] and Algorithm 2 of Mardani et al. [17], which is a first-order stochastic approximation type scheme. We see from Table 2 that the proposed algorithm requires less computations per iteration than Algorithm 1 of Mardani et al. [17], while it has similar complexity with the remaining three algorithms. Note though that, as it will be also shown in the next section, PETRELS and GROUSE perform well under the condition that the true subspace rank $r(n)$ is known, while Algorithm 2 of Mardani et al. [17], being a first-order algorithm is expected to have a much slower convergence rate compared to the remaining second-order schemes included in Table 2.With regard to memory requirements, OVBSL demands more storage space compared to the rest state-of-the art algorithms, yet at the same order of magnitude with PETRELS and Algorithm 1 of Mardani et al. [17]. As expected, lower memory storage is required by the first order methods namely GROUSE and Algorithm 2 of Mardani et al. [17].

7. Experimental results

In this section, the effectiveness of the proposed algorithm is corroborated in a variety of experiments carried out on synthetic and real data.

7.1. Synthetic data experiments

In the following, two different experiments are presented. Our first goal is to illustrate the efficiency of OVBSL in tackling matrix completion. It should be noted that the sparsity imposition on the subspace matrix from OVBSL is purposely relaxed in this experiment, that is we set $\mathbf{\Gamma}_l=\mathbf{I}_K, \forall l=1,2,\ldots,L$. The performance of OVBSL in the challenging *sparse* subspace estimation problem is explored in the second experiment of this subsection. Therein, the aforementioned favorable characteristic of OVBSL algorithm, i.e., its potential to impose sparsity on the subspace matrix, is thoroughly investigated. To this end, the parameters γ_{kl}'s are then considered "active", normally taking their values according to the full Bayesian model analytically described above.

7.1.1. Online matrix completion

In order to assess the performance of OVBSL algorithm in recovering missing data, we simulate a low dimensional subspace $\mathbf{U}\in\mathcal{R}^{K\times r}$ with $K=500$ and $r=5$ and Gaussian i.i.d entries $u_{kl}\sim\mathcal{N}(0,\frac{1}{K})$. Next, 20000 $r\times 1$ projection coefficient vectors $\mathbf{c}(n)$ are produced according to a Gaussian distribution $c_l(n)\sim\mathcal{N}(0,1)$. The signal $\mathbf{y}(n)$ at time n is then generated by the product $\mathbf{Uc}(n)$, it is normalized so that its power is equal to 1, and then contaminated by i.i.d Gaussian noise $\mathbf{e}(n)\sim\mathcal{N}(\mathbf{0},\beta^{-1}\mathbf{I}_K)$. To model the missing entries, we randomly select a fraction π of the entries from each datum $\mathbf{y}(n)$, which are assumed to be known, whereas the rest $(1-\pi)\times 100\%$ of the elements are considered to be missing. To show the merits of the proposed OVBSL algorithm, we compare it to three state-of-the-art techniques, namely GROUSE with greedy step-size,[15], PETRELS [4] and Algorithm 1 of Mardani et al. [17]. It is worthy to mention that, as also previously mentioned, both GROUSE and PETRELS hinge on the assumption that the rank of the underlying subspace is known. Contrary, Algorithm 1 of Mardani et al. [17] utilizes ℓ_2-norm regularization as described in the previous section, that robustifies the algorithm in the absence of this knowledge. Finally, the true standard deviation of the noise is provided as input for adaptively estimating the step-size of GROUSE while the low-rank regularization parameter of Algorithm 1 of Mardani et al. [17] is set to 0.1 as is proposed in the relevant paper.

In the sequel, to make things more interesting, we adhere to the challenging but realistic scenario whereby the true rank of the underlying subspace is unknown. Along this line, the rank of the subspace matrix is accordingly initialized in all tested algorithms to an overestimate of the true rank, namely $L=10$. Our initial objective is to demonstrate the effectiveness of the proposed OVBSL algorithm when certain amounts of data are missing. To this end, we carry out two experiments corresponding to different fractions of the observed entries i.e. $\pi=\{0.25,0.75\}$, keeping the noise precision β fixed to 10^3. Since the competence of the subspace learning algorithms in tracking possible changes of the sought subspace is of crucial importance in many applications, an abrupt change of the subspace is induced at $n=10000$ for $\pi=0.25$. The performance of the tested algorithms is evaluated in terms of the normalized running average estimation error (NRAEE) defined as: $\mathrm{NRAEE}(n)=\frac{1}{100}\sum_{i=n-99}^{n}\frac{\|\hat{\mathbf{y}}(i)-\mathbf{y}(i)\|_2}{\|\mathbf{y}(i)\|_2}$ where $\hat{\mathbf{y}}(i)=\hat{\mathbf{W}}(i)\hat{\mathbf{x}}(i)$. The average NRAEE of 10 independent runs of the experiment is shown in Fig. 2a. It is clear that the proposed OVBSL algorithm outperforms its rivals for both values of the fraction of the observed data π. At the same time, OVBSL is proven to be competent in tracking sudden changes of the latent subspace, since the transient deterioration of its performance caused by the deliberate change induced at $n=10000$ is swiftly rectified in the subsequent iterations. Notably, in the lack of knowledge of the true rank of the subspace, PETRELS becomes unstable. Contrary, Algorithm 1 of Mardani et al. [17] and GROUSE with the greedy step-size scheme present a robust behavior (note that GROUSE is given the true standard deviation of the noise for updating its step-size), though with clearly less reconstruction accuracy compared to the proposed OVBSL algorithm.

Next, we examine the robustness of OVBSL to noise corruption. To do so, we keep the fraction of the observed entries fixed to $\pi=0.4$, focusing on the behavior of OVBSL and the competing schemes for three different values of the noise precision i.e. $\beta=\{10^5,10^3,10^2\}$. Fig. 2b depicts the average NRAEE of 10 executions of the experiment obtained by the tested algorithms in the three different cases examined. It is easily noticed that herein as well, OVBSL achieves higher reconstruction accuracy than the competing schemes for all different β's, thus corroborating its strength to various levels of noise corruption.

7.1.2. Online sparse subspace estimation

In the following, the compelling feature of OVBSL to favor *sparse* subspace estimates is thoroughly explored. To clearly demonstrate the merits of this key aspect of our algorithm, a sparse subspace

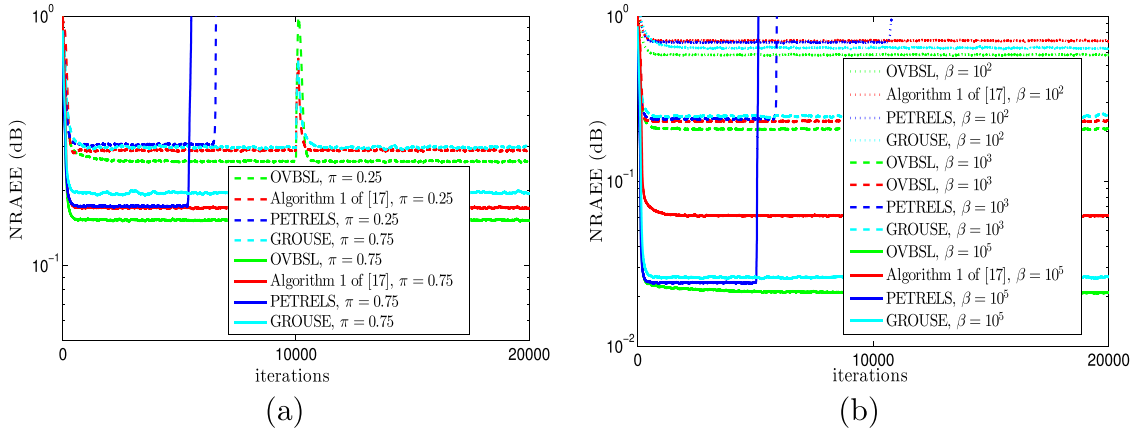


Fig. 2. Performance comparison among OVBSL, Algorithm 1 of Mardani et al. [17], PETRELS and GROUSE, [15], for the matrix completion problem. (a) Robustness to different fractions of the observed entries (π) (b) Sensitivity to different levels of noise corruption.

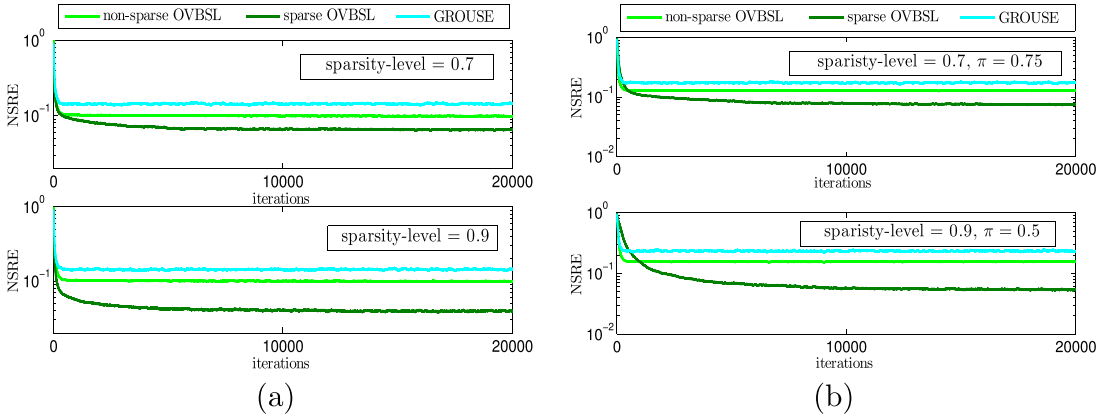


Fig. 3. Performance comparison between sparse and non-sparse versions of OVBSL and GROUSE, [15]. (a) Robustness to different sparsity levels of the subspace matrix and $\pi = 1$ (b) Robustness to different percentages of missing entries and subspace sparsity levels.

matrix $\mathbf{U}$ of rank $r = 5$ is modeled. Then, the same above-described process is adopted for producing $20,000$ projection coefficient vectors $\mathbf{c}(n)$, that give rise to the corresponding signals $\mathbf{Uc}(n)$. Finally, Gaussian i.i.d noise of precision $\beta = 10^3$ is assumed to contaminate the datums. For now, focusing on the subspace matrix estimation problem, we depart from the matrix completion problem considering that data are fully observed (hence the fraction of the observed entries π equals to 1) and we test two versions of OVBSL, that is, when sparsity of the subspace a) is taken into account and b) is disregarded in the same way explained earlier and the greedy step-size version of GROUSE, [15]. The estimates of the subspace are assessed as time evolves by means of the normalized subspace reconstruction error (NSRE) defined as $\text{NSRE}(n) = \frac{\|\mathcal{P}_{\hat{\mathbf{W}}^{\perp}(n)}\mathbf{U}\|_F}{\|\mathbf{U}\|_F}$.[8] The benefits emerging from taking into account the sparsity existing in the unknown subspace matrix, come to light by exploring OVBSL's performance for different levels of sparsity imposed on it, namely 0.7 and 0.9. In both cases, the subspace matrices are initialized to an overestimate of the rank, i.e., $L = 10$. Fig. 3a depicts the mean NSRE of 10 runs of the experiment obtained for the two versions of OVBSL and GROUSE as time evolves. As it can be readily seen, OVBSL achieves subspace estimates of higher accuracy compared to both its so to speak non-sparse version and GROUSE which, likewise, does not favor sparsity on the subspace matrix. It should be noted that the gains obtained by the sparse OVBSL are becoming abundantly clear as the sparsity level increases.

Next, OVBSL and GROUSE are probed in the challenging problem of sparse subspace estimation from partially observed data. Towards this, the same experimental setting described above is followed and two cases corresponding to two different combinations of sparsity level and fraction of observed entries are examined, namely a) sparsity-level=0.7 and $\pi = 0.75$ and b) sparsity-level=0.9 and $\pi = 0.5$. OVBSL is again evaluated for the two cases corresponding to its sparse and non-sparse version and GROUSE is also tested, initializing the rank L of subspace matrices to 5 and using NSRE as the performance metric. From Fig. 3b, it is verified that albeit data are incomplete, sparse OVBSL outperforms both its non-sparse version and GROUSE thus corroborating that taking advantage of the sparsity of the subspace matrix is still meaningful when the assumption of sparse subspace is valid.

7.2. Real data experiments

In this part of the paper, the efficiency of OVBSL algorithm is investigated on real data. More concretely, we conduct two different experiments corresponding a) to hyperspectral image reconstruction out of partially observed measurements and b) to the eigenface learning problem. In both experiments OVBSL is compared with the state-of-the-art Algorithm 1 of Mardani et al. [17] whose low-rank regularization parameter takes its value according to the heuristic rule that was also followed on the real data experiments of Mardani et al. [17].

[8] $\mathcal{P}_{\hat{\mathbf{W}}^{\perp}(n)}\mathbf{U}$ denotes the projection of the true subspace matrix $\mathbf{U}$ to the orthogonal complement of the subspace spanned by the columns of the estimated subspace matrix $\hat{\mathbf{W}}^{\perp}(n)$.

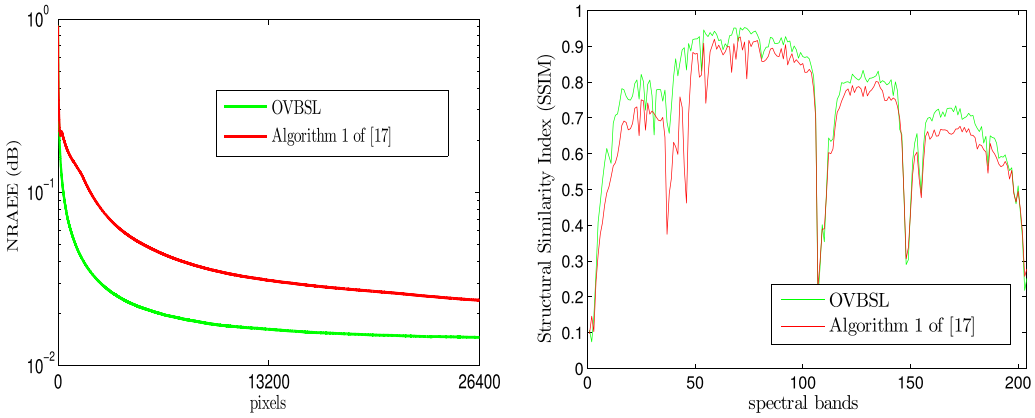
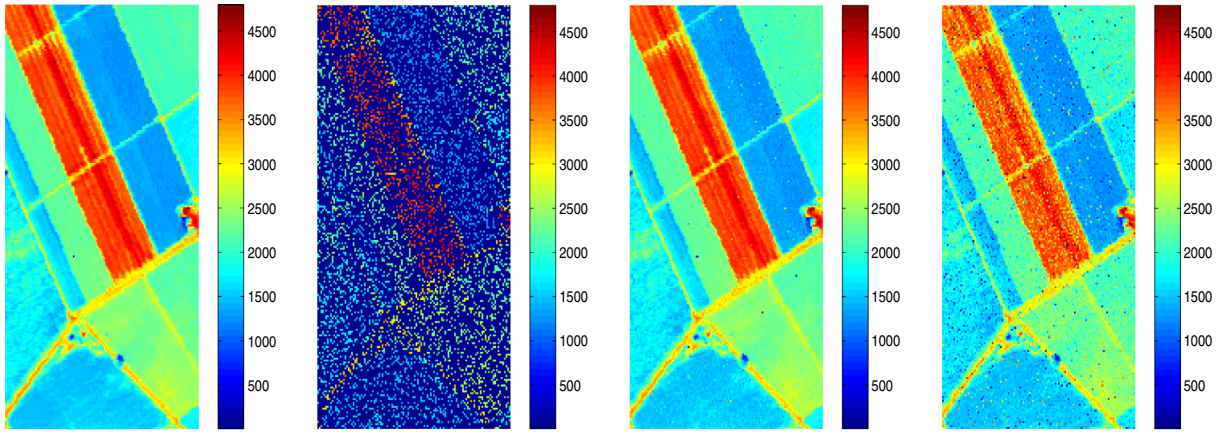
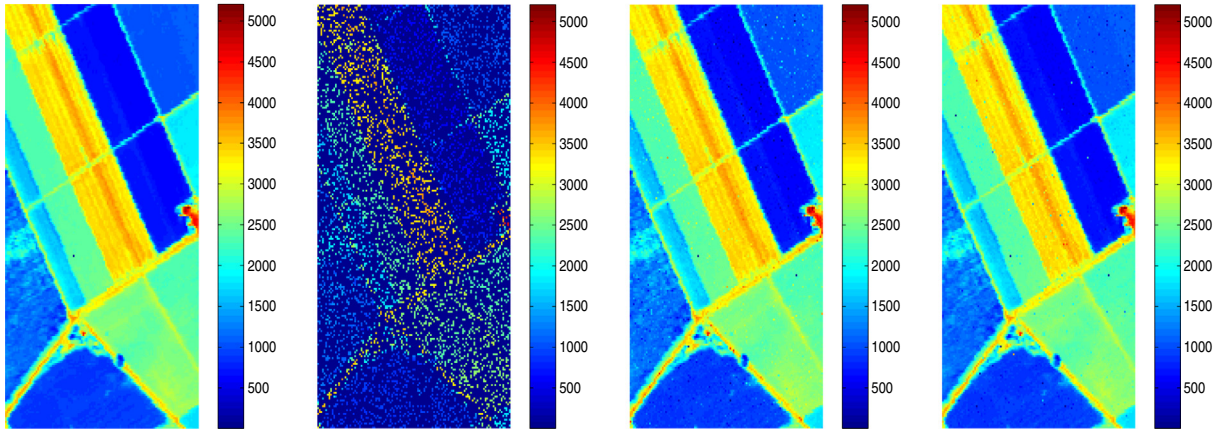


Fig. 4. Performance comparison between OVBSL and Algorithm 1 of Mardani et al. [17] in terms of NRAEE and SSIM. (a) NRAEE as the number of processed pixel increases (b) SSIM index per reconstructed band.

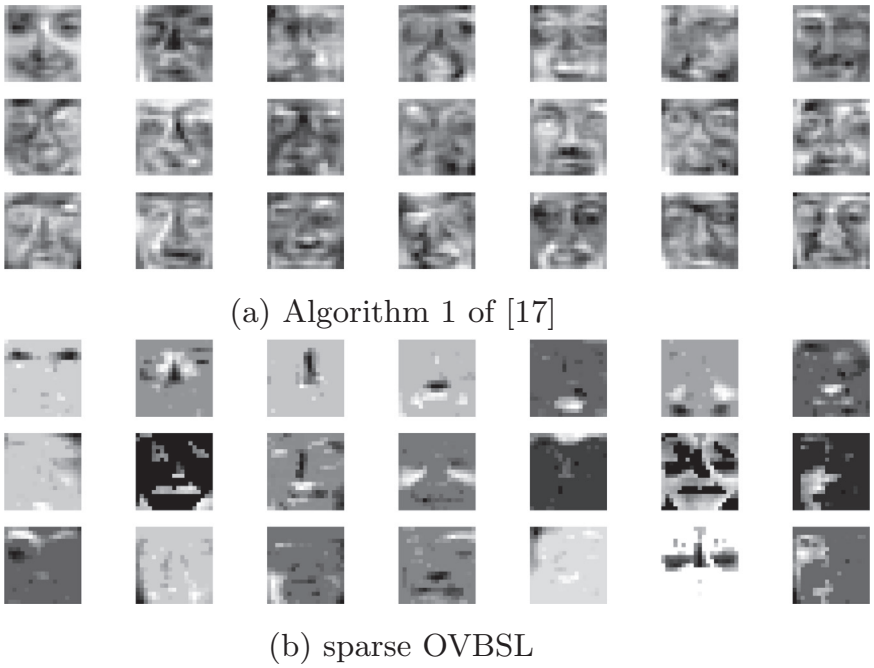


(a) true Salinas HSI, band 37 (b) incomplete band 37 (c) OVBSL, reconstructed band 37 (d) Algorithm 1 of [17], reconstructed band 37



(e) true Salinas HSI, band 31 (f) incomplete band 31 (g) OVBSL, reconstructed band 31 (h) Algorithm 1 of [17], reconstructed band 31

Fig. 5. Reconstruction of Salinas Valley HSI by OVBSL and Algorithm 1 of Mardani et al. [17], for $\pi = 0.2$.



(a) Algorithm 1 of [17]

(b) sparse OVBSL

Fig. 6. Eigenfaces obtained by Algorithm 1 of Mardani et al. [17] and OVBSL on MIT-CBCL dataset.

7.2.1. Pixel-by-pixel hyperspectral image recovery

A hyperspectral image (HSI) is a collection of multiple grayscale images captured at many contiguous spectral bands (channels), thus forming a so-called spectral cube. As a result of this, each pixel in a HSI is represented by a vector of size equal to the number of spectral bands and is called pixel *spectral signature*. The entries of this vector are the radiance values of the spatial area corresponding to the pixel in all spectral channels. A key characteristic of HSIs is the high degree of correlation they present, both in the spectral and the spatial domains, [34]. Given a HSI, let us form a matrix with its rows corresponding to the pixels of the HSI, and its columns to the spectral bands. In doing so, it can be easily seen that the underlying high coherence appearing both in columns and rows leads to a matrix that may be of very low rank, as compared to its dimensions. Actually, this fact gives us good grounds for exploiting the low-rank structure in favor of recovering HSIs, in cases that data either are partly missing or have suffered by severe noise corruption. In the following, we test the performance of OVBSL and Algorithm 1 of Mardani et al. [17] in recovering the Salinas Valley HSI, [34], out of a fraction $\pi = 0.2$ of its entries. Since both algorithms process data in an online fashion, we assume that the aforementioned time instances, hereafter, correspond to a sequence of all the pixels taken in a random order from the image. Put differently, the algorithms process the pixel spectral signatures (which are the rows of the formed matrix) one-by-one, as if they were becoming available in a streaming fashion. Notably, this type of processing aside from reducing the computational complexity, it alleviates the need for memory storage, thus paving the way for on-board processing. The rank of the subspace matrix is initialized to $L = 10$. To quantitatively assess the performance of the tested algorithms, we estimate the NRAEE (Fig. 4a), as the number of the processed pixels increases, and the structural similarity (SSIM) index values, [35], between the true and the reconstructed band images (Fig. 4b). It is clearly shown in Fig. 4a that OVBSL achieves higher reconstruction accuracy on average as compared to its rival in terms of NRAEE. Focusing on Fig. 5b, it can be noticed that OVBSL presents higher SSIMs in the majority of the spectral bands, with only few exceptions of bands, where the SSIM indexes obtained by OVBSL and Algorithm 1 of [17], either take close values or the latter gets slightly greater values, e.g. at band 31. In order to give a further insight at the reconstructed bands, we provide in Fig. 5a and e, the true bands 31 and 37, respectively, accompanied by their incomplete versions (Fig. 5b and f) that were provided as inputs to the tested algorithms. From Fig. 5c and d, it can be easily observed that, in good agreement with the SSIMs of the two algorithms at this band, OVBSL reconstructs the 37th band of Salinas HSI in remarkably higher accuracy than Algorithm 1 of [17]. As regards band 31 where the SSIM index of Algorithm 1 of Mardani et al. [17] is slightly higher than that of OVBSL, Fig. 5g and h show that the reconstructed images are quite similar for both algorithms. That said, OVBSL is favorably proven to be competent in processing this real HSI dataset, outperforming the state-of-the-art Algorithm 1 of Mardani et al. [17].

7.2.2. Online eigenface learning

In this section, we qualitatively evaluate the performance of sparse OVBSL as compared to the non-sparse Algorithm 1 of Mardani et al. [17] on another real dataset. Towards this, we use the MIT-CBCL face dataset [36], which contains $n = 2429$ face images of size 19×19 pixels. The tested algorithms process the images as K-dimensional vectors with $K = 361 (= 19^2)$, in an online fashion. The subspace matrix estimated by both algorithms can be deemed as a *learned dictionary* of faces. In doing so, each image can be reconstructed by a linear combination of the atoms (eigenfaces) contained in the subspace matrix. The rank of the subspace is initialized for both algorithms to 50. Fig. 6 shows the 21 more characteristic eigenfaces. Dark pixels correspond to negative values, while positive values are denoted with light colors. As it can be noticed, sparsity imposition from the sparse OVBSL leads to eigenfaces that present more localized features, contrary to those obtained by Algorithm 1 of Mardani et al. [17], where features are spread out over the image. It should be also noted that OVBSL converged to a subspace matrix of low-rank. This fact resulted from the inherent advantageous characteristic of OVBSL to eliminate components presenting low variance, hence offering negligible information.

8. Conclusions

In this paper, a novel online variational Bayes subspace learning (OVBSL) algorithm from incomplete data was presented. Two basic merits of the proposed approach are: a) the imposition of low-rankness on the sought subspace by utilizing a novel group sparsity based heuristic, and b) the sparsity promotion on the subspace matrix. The former characteristic makes the algorithm robust in the absence of the knowledge of the true rank while the latter renders it amenable to sparse dictionary learning problems. OVBSL belongs to the family of Bayesian algorithms thus, contrary to its deterministic counterparts, no parameter fine-tuning is required. The effectiveness of the proposed algorithm is verified in a variety of experiments conducted on simulated and real data. Subspace tracking from partial observations, treated in this paper, can be also viewed as an online matrix completion task that has a major impact in numerous applications. By suitably extending and modifying the proposed Bayesian model and methodology, similar problems of high importance such as online nonnegative matrix factorization and online robust PCA can be potentially tackled. This extension as well as the unification of all the different schemes under a common umbrella is the subject of our current investigation.

Appendix A

Due to the conjugacy of the respective prior distributions (14), (15) and the likelihood (12), the posterior distribution $q(\mathbf{x}(i))$ of the ith coefficient vector does turn out to be Gaussian, i.e.

$$q(\mathbf{x}(i)) = \mathcal{N}\big(\mathbf{x}(i) \mid \langle \mathbf{x}(i) \rangle, \mathbf{\Sigma}_{\mathbf{x}(i)}\big), \tag{A.1}$$

with mean $\langle \mathbf{x}(i) \rangle$ and covariance matrix $\mathbf{\Sigma}_{\mathbf{x}(i)}$ given by,

$$\langle \mathbf{x}(i) \rangle = \langle \beta \rangle \mathbf{\Sigma}_{\mathbf{x}(i)} \langle \mathbf{W} \rangle^T \mathbf{z}(i), \tag{A.2}$$

$$\mathbf{\Sigma}_{\mathbf{x}(i)} = \langle \beta \rangle^{-1} \big(\langle \mathbf{W}^T \mathbf{\Phi}_i \mathbf{W} \rangle + \langle \mathbf{S} \rangle \big)^{-1}, \tag{A.3}$$

where we recall that $\mathbf{\Phi}_i = \mathrm{diag}(\boldsymbol{\phi}(i))$. The expectation term $\langle \mathbf{W}^T \mathbf{\Phi}_i \mathbf{W} \rangle$ is expressed as,

$$\langle \mathbf{W}^T \mathbf{\Phi}_i \mathbf{W} \rangle = \langle \mathbf{W} \rangle^T \mathbf{\Phi}_i \langle \mathbf{W} \rangle + \sum_{k=1}^{K} \phi_{ik} \mathbf{\Sigma}_{\mathbf{w}_k} \tag{A.4}$$

where $\mathbf{\Sigma}_{\mathbf{w}_k} = \mathrm{diag}([\sigma^2_{w_{k1}}, \sigma^2_{w_{k2}}, \ldots, \sigma^2_{w_{kL}}]^T)$ by virtue of the statistical independence assumed for the elements of $\mathbf{W}$. Note that $\sigma^2_{w_{kl}}$ is the variance of w_{kl} whose posterior turns out also to be Gaussian, i.e.

$$q(w_{kl}) = \mathcal{N}(w_{kl} \mid \langle w_{kl} \rangle, \sigma^2_{w_{kl}}), \tag{A.5}$$

with

$$\langle w_{kl} \rangle = \langle \beta \rangle \sigma^2_{w_{kl}} \big(\langle \boldsymbol{x}_l \rangle^T \mathbf{\Lambda} \boldsymbol{z}_k - \langle \boldsymbol{x}_l^T \mathbf{\Lambda} \mathbf{\Phi}_k \mathbf{X}_{\neg l} \rangle \langle \mathbf{w}_{k\neg l} \rangle \big), \tag{A.6}$$

$$\sigma^2_{w_{kl}} = \langle \beta \rangle^{-1} \big(\langle \boldsymbol{x}_l^T \mathbf{\Lambda} \mathbf{\Phi}_k \boldsymbol{x}_l \rangle + \langle \gamma_{kl} \rangle \langle s_l \rangle \big)^{-1}. \tag{A.7}$$

$\mathbf{X}_{\neg l}$ and $\mathbf{w}_{k\neg l}$ in (A.6) are the quantities arising after removing the lth column and the lth element of $\mathbf{X}$ and $\mathbf{w}_k$, respectively and $\mathbf{\Phi}_k = \mathrm{diag}(\boldsymbol{\varphi}_k)$. As for the expectation terms appearing in (A.6) and (A.7), it holds,

$$\langle \boldsymbol{x}_l^T \mathbf{\Lambda} \mathbf{\Phi}_k \mathbf{X}_{\neg l} \rangle = \langle \boldsymbol{x}_l \rangle^T \mathbf{\Lambda} \mathbf{\Phi}_k \langle \mathbf{X}_{\neg l} \rangle + \sum_{i=1}^{n} \lambda^{n-i} \phi_{ik} \boldsymbol{\sigma}^T_{x(i)\neg l}, \tag{A.8}$$

$$\langle \boldsymbol{x}_l^T \mathbf{\Lambda} \mathbf{\Phi}_k \boldsymbol{x}_l \rangle = \langle \boldsymbol{x}_l \rangle^T \mathbf{\Lambda} \mathbf{\Phi}_k \langle \boldsymbol{x}_l \rangle + \sum_{i=1}^{n} \lambda^{n-i} \phi_{ik} \sigma_{x_{il}}, \tag{A.9}$$

with $\boldsymbol{\sigma}_{x(i)\neg l}$ standing for the lth column of $\mathbf{\Sigma}_{\mathbf{x}(i)}$ after removing its lth element $\sigma_{x_{il}}$.

Next, the posterior distributions of the variables s_l's and γ_{kl}'s belonging to the second hierarchical level are unfolded. From (24) it can be shown that the column sparsity promoting parameters s_l's are *a posteriori* distributed according to the following generalized inverse Gaussian distribution,

$$q(s_l) = \mathcal{GIG}\Big(s_l \mid -\frac{1}{2}, \langle \beta \rangle \big(\langle \boldsymbol{w}_l^T \mathbf{\Gamma}_l \boldsymbol{w}_l \rangle + \langle \boldsymbol{x}_l^T \mathbf{\Lambda} \boldsymbol{x}_l \rangle \big), \langle \delta_l \rangle \Big). \tag{A.10}$$

For the mean $\langle s_l \rangle$ of the $\mathcal{GIG}$ distribution it holds,

$$\langle s_l \rangle = \sqrt{\frac{\langle \delta_l \rangle}{\langle \beta \rangle \big(\langle \boldsymbol{w}_l^T \mathbf{\Gamma}_l \boldsymbol{w}_l \rangle + \langle \boldsymbol{x}_l^T \mathbf{\Lambda} \boldsymbol{x}_l \rangle \big)}}. \tag{A.11}$$

Likewise, the posterior distribution of γ_{kl}'s that promote independently sparsity on the elements of the subspace matrix $\mathbf{W}$ is the generalized inverse Gaussian

$$q(\gamma_{kl}) = \mathcal{GIG}\Big(\gamma_{kl} \mid -\frac{1}{2}, \langle \beta \rangle \langle s_l \rangle \langle w_{kl}^2 \rangle, \langle \rho_{kl} \rangle \Big), \tag{A.12}$$

with $\langle w_{kl}^2 \rangle = \langle w_{kl} \rangle^2 + \sigma^2_{w_{kl}}$. Hence,

$$\langle \gamma_{kl} \rangle = \sqrt{\frac{\langle \rho_{kl} \rangle}{\langle \beta \rangle \langle s_l \rangle (\langle w_{kl} \rangle^2 + \sigma^2_{w_{kl}})}}. \tag{A.13}$$

As far as the hyperparameters δ_l and ρ_{kl} of s_l and γ_{kl} respectively, are concerned, both are *a posteriori* Gamma distributed i.e.,

$$q(\delta_l) = \mathcal{G}(\delta_l \mid \bar{\mu}, \bar{\nu}_l) \tag{A.14}$$

with $\bar{\mu} = \mu + \frac{n+K+1}{2}$ and $\bar{\nu}_l = \nu + \frac{1}{2}\langle \frac{1}{s_l} \rangle$, and

$$q(\rho_{kl}) = \mathcal{G}\big(\rho_{kl} \mid \bar{\psi}, \bar{\xi}_{kl}\big) \tag{A.15}$$

with $\bar{\psi} = \psi + 1$ and $\bar{\xi}_{kl} = \xi + \frac{1}{2}\langle \frac{1}{\gamma_{kl}} \rangle$. For the expected values of δ_l and ρ_{kl}, that is $\langle \delta_l \rangle$ and $\langle \rho_{kl} \rangle$ we have,

$$\langle \delta_l \rangle = \frac{\mu + \frac{n+K+1}{2}}{\nu + \frac{1}{2}\langle \frac{1}{s_l} \rangle}, \tag{A.16}$$

$$\langle \rho_{kl} \rangle = \frac{\psi + 1}{\xi + \frac{1}{2}\langle \frac{1}{\gamma_{kl}} \rangle}. \tag{A.17}$$

Using the form of the distributions in (A.10) and (A.12), the expectation terms $\langle \frac{1}{s_l} \rangle$ and $\langle \frac{1}{\gamma_{kl}} \rangle$ arising in (A.16) and (A.17) can be obtained as,

$$\left\langle \frac{1}{s_l} \right\rangle = \frac{1}{\langle s_l \rangle} + \frac{1}{\langle \delta_l \rangle} \tag{A.18}$$

$$\left\langle \frac{1}{\gamma_{kl}} \right\rangle = \frac{1}{\langle \gamma_{kl} \rangle} + \frac{1}{\langle \rho_{kl} \rangle} \tag{A.19}$$

Concluding the posterior distributions of all the involved variables in our hierarchical model, it can be shown that the noise precision β is Gamma distributed as follows,

$$q(\beta) = \mathcal{G}\big(\beta \mid \bar{\kappa}, \bar{\theta}\big) \tag{A.20}$$

where $\bar{\kappa} = \kappa + \frac{n(K+L)+KL}{2}$

$$\bar{\theta} = \theta + \sum_{k=1}^{K} \Big(\langle \| \mathbf{\Lambda}^{\frac{1}{2}} \big(\boldsymbol{z}_k - \mathbf{\Phi}_k \mathbf{X} \mathbf{w}_k \big) \|_2^2 \rangle + \langle \mathbf{w}_k^T \mathbf{S} \mathbf{\Gamma}_k \mathbf{w}_k \rangle \Big) + \sum_{l=1}^{L} \langle s_l \rangle \langle \boldsymbol{x}_l^T \mathbf{\Lambda} \boldsymbol{x}_l \rangle. \tag{A.21}$$

and $\mathbf{\Gamma}_k = \mathrm{diag}([\gamma_{k1}, \gamma_{k2}, \ldots, \gamma_{kL}]^T)$. The expectation of β is given by $\langle \beta \rangle = \frac{\bar{\kappa}}{\bar{\theta}}$. As for the expectation terms arising in (A.21), it holds,

$$\langle \|\mathbf{\Lambda}^{\frac{1}{2}}\left(\boldsymbol{z}_k-\boldsymbol{\Phi}_k\mathbf{X}\mathbf{w}_k\right)\|_2^2\rangle$$
$$=\|\mathbf{\Lambda}^{\frac{1}{2}}\left(\boldsymbol{z}_k-\boldsymbol{\Phi}_k\langle\mathbf{X}\rangle\langle\mathbf{w}_k\rangle\right)\|_2^2+\mathrm{Tr}\left(\langle\mathbf{X}\rangle^T\mathbf{\Lambda}\boldsymbol{\Phi}_k\langle\mathbf{X}\rangle\mathbf{\Sigma}_{\mathbf{w}_k}\right)$$
$$+\langle\mathbf{w}_k\rangle^T\sum_{i=1}^{n}\phi_{ik}\lambda^{n-i}\mathbf{\Sigma}_{\mathbf{x}(i)}\langle\mathbf{w}_k\rangle+\mathrm{Tr}\left(\mathbf{\Sigma}_{\mathbf{w}_k}\sum_{i=1}^{n}\phi_{ik}\lambda^{n-i}\mathbf{\Sigma}_{\mathbf{x}(i)}\right) \tag{A.22}$$

$$\langle\mathbf{w}_k^T\mathbf{S}\mathbf{\Gamma}_l\mathbf{w}_k\rangle=\langle\mathbf{w}_k\rangle^T\langle\mathbf{S}\rangle\langle\mathbf{\Gamma}_l\rangle\langle\mathbf{w}_k\rangle+\sum_{l=1}^{L}\langle s_l\rangle\langle\gamma_{kl}\rangle\sigma_{w_{kl}}^2 \tag{A.23}$$

Appendix B

The joint prior of $\mathbf{X}$ and $\mathbf{W}$ is expressed as

$$p(\mathbf{X},\mathbf{W}\mid\boldsymbol{\delta},\beta,\mathbf{\Gamma})=\prod_{l=1}^{L}p(\boldsymbol{x}_l,\boldsymbol{w}_l\mid\delta_l,\beta,\mathbf{\Gamma}_l) \tag{B.1}$$

where

$$p(\boldsymbol{x}_l,\boldsymbol{w}_l\mid\delta_l,\beta,\mathbf{\Gamma}_l)=\int_0^{\infty}p(\boldsymbol{x}_l\mid s_l,\beta)p(\boldsymbol{w}_l\mid s_l,\beta,\mathbf{\Gamma}_l)p(s_l\mid\delta_l)ds_l \tag{B.2}$$

Using (14), (15) and (17) in (B.2) yields

$$p(\boldsymbol{x}_l,\boldsymbol{w}_l\mid\delta_l,\beta,\mathbf{\Gamma}_l)=\int_0^{\infty}(2\pi)^{-\frac{n+K}{2}}\beta^{\frac{n+K}{2}}|\mathbf{\Lambda}\mathbf{\Gamma}_l|^{\frac{1}{2}}s_l^{-\frac{3}{2}}$$
$$\times\exp\left(-\frac{\beta s_l}{2}(\|\boldsymbol{x}_l\|_{2,\mathbf{\Lambda}}^2+\|\boldsymbol{w}_l\|_{2,\mathbf{\Gamma}_l}^2)-\frac{\delta_l}{2s_l}\right)ds_l, \tag{B.3}$$

where $\|\boldsymbol{x}_l\|_{2,\mathbf{\Lambda}}^2=\boldsymbol{x}_l^T\mathbf{\Lambda}\boldsymbol{x}_l$ and $\|\boldsymbol{w}_l\|_{2,\mathbf{\Gamma}_l}^2=\boldsymbol{w}_l^T\mathbf{\Gamma}_l\boldsymbol{w}_l$. Using in (B.3) the expression of the GIG distribution for s_l with parameters $a=\beta(\|\boldsymbol{x}_l\|_{2,\mathbf{\Lambda}}^2+\|\boldsymbol{w}_l\|_{2,\mathbf{\Gamma}_l}^2)$, $b=\delta_l$ and $p=-1/2$, we easily get

$$p(\boldsymbol{x}_l,\boldsymbol{w}_l\mid\delta_l,\beta,\mathbf{\Gamma}_l)=(2\pi)^{-\frac{n+K}{2}}\beta^{\frac{n+K}{2}}|\mathbf{\Lambda}\mathbf{\Gamma}_l|^{\frac{1}{2}}2\mathcal{K}_{-\frac{1}{2}}$$
$$\times\left(\beta^{\frac{1}{2}}\delta_l^{\frac{1}{2}}(\|\boldsymbol{x}_l\|_{2,\mathbf{\Lambda}}^2+\|\boldsymbol{w}_l\|_{2,\mathbf{\Gamma}_l}^2)^{\frac{1}{2}}\right)$$
$$\times\left(\frac{\beta(\|\boldsymbol{x}_l\|_{2,\mathbf{\Lambda}}^2+\|\boldsymbol{w}_l\|_{2,\mathbf{\Gamma}_l}^2)}{\delta_l}\right)^{\frac{1}{4}} \tag{B.4}$$

By employing the identity,

$$\mathcal{K}_{-\frac{1}{2}}(x)=\left(\frac{\pi}{2x}\right)^{\frac{1}{2}}\exp(-x)$$

in (B.4) and after some straightforward calculations, we end up with the following expression for the joint distribution of $\mathbf{x}_l$ and $\mathbf{w}_l$,

$$p(\boldsymbol{x}_l,\boldsymbol{w}_l\mid\delta_l,\beta,\mathbf{\Gamma}_l)=(2\pi)^{-\frac{n+K-1}{2}}\beta^{\frac{n+K}{2}}|\mathbf{\Lambda}\mathbf{\Gamma}_l|^{\frac{1}{2}}\delta_l^{-\frac{1}{2}}$$
$$\times\exp\left(-\beta^{\frac{1}{2}}\delta_l^{\frac{1}{2}}(\|\boldsymbol{x}_l\|_{2,\mathbf{\Lambda}}^2+\|\boldsymbol{w}_l\|_{2,\mathbf{\Gamma}_l}^2)^{\frac{1}{2}}\right). \tag{B.5}$$

Then, from (B.1)

$$p(\mathbf{X},\mathbf{W}\mid\boldsymbol{\delta},\beta,\mathbf{\Gamma})=(2\pi)^{-\frac{(n+K-1)L}{2}}\beta^{\frac{(n+K)L}{2}}|\mathbf{\Lambda}|^{\frac{L}{2}}\left(\prod_{l=1}^{L}\delta_l^{\frac{1}{2}}|\mathbf{\Gamma}_l|^{\frac{1}{2}}\right)$$
$$\times\exp\left(-\beta^{\frac{1}{2}}\sum_{l=1}^{L}\delta_l^{\frac{1}{2}}(\|\boldsymbol{x}_l\|_{2,\mathbf{\Lambda}}^2+\|\boldsymbol{w}_l\|_{2,\mathbf{\Gamma}_l}^2)^{\frac{1}{2}}\right) \tag{B.6}$$

which is a multi-parameter (with respect to the δ_l's) Lalpace-type distribution defined on the columns of the matrix $[(\mathbf{\Lambda}^{1/2}\mathbf{X})^T(\mathbf{\Gamma}\odot\mathbf{W})^T]^T$. Such a distribution is known to impose column sparsity and thus, due to the form of the matrix, joint column sparsity on $\mathbf{X}$ and $\mathbf{W}$.

References

[1] M. Mardani, G. Mateos, G.B. Giannakis, Dynamic anomalography: tracking network anomalies via sparsity and low rank, Sel. Topics Signal Process. IEEE J. 7 (1) (2013) 50–66, doi:10.1109/JSTSP.2012.2233193.

[2] M. Elad, M. Aharon, Image denoising via sparse and redundant representations over learned dictionaries, Image Process. IEEE Trans. 15 (12) (2006) 3736–3745, doi:10.1109/TIP.2006.881969.

[3] S. Theodoridis, Machine Learning: A Bayesian and Optimization Perspective, Academic Press, 2015.

[4] Y. Chi, Y. Eldar, R. Calderbank, Petrels: parallel subspace estimation and tracking by recursive least squares from partial observations, Signal Process. IEEE Trans. 61 (23) (2013) 5947–5959, doi:10.1109/TSP.2013.2282910.

[5] N. Owsley, Adaptive data orthogonalization, in: Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '78., vol. 3, 1978, pp. 109–112, doi:10.1109/ICASSP.1978.1170425.

[6] J. Bunch, C. Nielsen, D. Sorensen, Rank-one modification of the symmetric eigenproblem, Numerische Mathematik 31 (1) (1978) 31–48, doi:10.1007/BF01396012.

[7] L. Balzano, R. Nowak, B. Recht, Online identification and tracking of subspaces from highly incomplete information, in: Communication, Control, and Computing, 2010 48th Annual Allerton Conference on, 2010, pp. 704–711, doi:10.1109/ALLERTON.2010.5706976.

[8] J. Mairal, F. Bach, J. Ponce, G. Sapiro, Online learning for matrix factorization and sparse coding, J. Mach. Learn. Res. 11 (2010) 19–60.

[9] E.J. Candès, B. Recht, Exact matrix completion via convex optimization, Found. Comput. Math. 9 (6) (2009) 717–772.

[10] S. Chouvardas, Y. Kopsinis, S. Theodoridis, Robust subspace tracking with missing entries: the set-theoretic approach, Signal Process. IEEE Trans. 63 (19) (2015) 5060–5070, doi:10.1109/TSP.2015.2449254.

[11] X. Su, T.M. Khoshgoftaar, A survey of collaborative filtering techniques, Adv. Artif. Intell. 2009 (2009) 4.

[12] S.F. Gull, G.J. Daniell, Image reconstruction from incomplete and noisy data, Nature 272 (20) (1978) 686–690.

[13] X. Doukopoulos, G. Moustakides, Fast and stable subspace tracking, Signal Process. IEEE Trans. 56 (4) (2008) 1452–1465, doi:10.1109/TSP.2007.909335.

[14] L. Balzano, S.J. Wright, Local convergence of an algorithm for subspace identification from partial data, Found. Comput. Math. (2014) 1–36.

[15] D. Zhang, L. Balzano, Global convergence of a grassmannian gradient descent algorithm for subspace estimation, in: The 19th International Conference on Artificial Intelligence and Statistics, 2016.

[16] B. Yang, Projection approximation subspace tracking, Signal Process. IEEE Trans. 43 (1) (1995) 95–107, doi:10.1109/78.365290.

[17] M. Mardani, G. Mateos, G.B. Giannakis, Subspace learning and imputation for streaming big data matrices and tensors, Signal Process. IEEE Trans. 63 (10) (2015) 2663–2677, doi:10.1109/TSP.2015.2417491.

[18] S. Babacan, M. Luessi, R. Molina, A. Katsaggelos, Sparse Bayesian methods for low-rank matrix estimation, Signal Process. IEEE Trans. 60 (8) (2012) 3964–3977, doi:10.1109/TSP.2012.2197748.

[19] V.Y. Tan, C. Févotte, Automatic relevance determination in nonnegative matrix factorization, SPARS'09-Signal Processing with Adaptive Sparse Structured Representations, 2009.

[20] D. Tzikas, C. Likas, N. Galatsanos, The variational approximation for Bayesian inference, Signal Process. Mag. IEEE 25 (6) (2008) 131–146, doi:10.1109/MSP.2008.929620.

[21] J.T. Parker, P. Schniter, V. Cevher, Bilinear generalized approximate message passing part i: derivation, IEEE Trans. Signal Process. 62 (22) (2014) 5839–5853.

[22] K.E. Themelis, A.A. Rontogiannis, K.D. Koutroumbas, A variational Bayes framework for sparse adaptive estimation, Signal Process. IEEE Trans. 62 (18) (2014) 4723–4736, doi:10.1109/TSP.2014.2338839.

[23] K.E. Themelis, A.A. Rontogiannis, K.D. Koutroumbas, Variational bayes group sparse time-adaptive parameter estimation with either known or unknown sparsity pattern, Signal Process. IEEE Trans. 64 (12) (2016) 3194–3206, doi:10.1109/TSP.2016.2543204.

[24] W. Yang, H. Xu, Streaming sparse principal component analysis, in: Proceedings of the 32nd International Conference on Machine Learning (ICML-15), 2015, pp. 494–503.

[25] N. Srebro, A. Shraibman, Rank, trace-norm and max-norm, in: Learning Theory, 18th Annual Conference on Learning Theory, Italy, June, 2005, pp. 545–560, doi:10.1007/11503415_37.

[26] P.V. Giampouras, A.A. Rontogiannis, K.E. Themelis, K.D. Koutroumbas, Online Bayesian low-rank subspace learning from partial observations, in: 23rd European Signal Processing Conference (EUSIPCO), 2015, pp. 2526–2530, doi:10.1109/EUSIPCO.2015.7362840.

[27] D. Cai, X. He, J. Han, Spectral regression: a unified approach for sparse subspace learning, in: Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on, IEEE, 2007, pp. 73–82.

[28] Y. Zhang, L.E. Ghaoui, Large-scale sparse principal component analysis with application to text data, in: Advances in Neural Information Processing Systems, 2011, pp. 532–539.
[29] T.P. Minka, Expectation propagation for approximate Bayesian inference, in: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence, in: UAI '01, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001, pp. 362–369.
[30] A.T. Cemgil, A Tutorial Introduction to Monte Carlo Methods, Markov Chain Monte Carlo and Particle Filtering, Elsevier Major Reference Works, 2012.
[31] D.P. Bertsekas, Nonlinear Programming, Athena Scientific, 1999.
[32] M. Tipping, Sparse Bayesian learning and the relevance vector machine, J. Mach. Learn. Res. 1 (2001) 211–244.
[33] M. Kowalski, Sparse regression using mixed norms, Appl. Comput. Harmon. Anal. 27 (3) (2009) 303–324. http://dx.doi.org/10.1016/j.acha.2009.05.006.
[34] P.V. Giampouras, K.E. Themelis, A.A. Rontogiannis, K.D. Koutroumbas, Simultaneously sparse and low-rank abundance matrix estimation for hyperspectral image unmixing, Geosci. Remote Sens. IEEE Trans. 54 (8) (2016) 4775–4789, doi:10.1109/TGRS.2016.2551327.
[35] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612, doi:10.1109/TIP.2003.819861.
[36] K.-K. Sung, Learning and Example Selection for Object and Pattern Recognition, PhD thesis, MIT, Artificial Intelligence Laboratory and Center for Biological and Computational Learning, Cambridge, MA, 1996 Ph.D. thesis.