HAL
open science

Indian Buffet Process Dictionary Learning : algorithms and applications to image processing

Hong-Phuong Dang, Pierre Chainais

To cite this version:

Hong-Phuong Dang, Pierre Chainais. Indian Buffet Process Dictionary Learning : algorithms and applications to image processing. International Journal of Approximate Reasoning, 2017. hal-01433609

HAL Id: hal-01433609

https://hal.science/hal-01433609v1

Submitted on 12 Jan 2017

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

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

Indian Buffet Process Dictionary Learning : algorithms and applications to image processing ☆

Hong-Phuong Dang[a], Pierre Chainais[a]

[a] *Univ. Lille, CNRS, Centrale Lille, UMR 9189 - CRIStAL*
Centre de Recherche en Informatique Signal et Automatique de Lille, F-59000 Lille, France

Abstract

Ill-posed inverse problems call for some prior model to define a suitable set of solutions. A wide family of approaches relies on the use of sparse representations. Dictionary learning precisely permits to learn a redundant set of atoms to represent the data in a sparse manner. Various approaches have been proposed, mostly based on optimization methods. We propose a Bayesian non parametric approach called IBP-DL that uses an Indian Buffet Process prior. This method yields an efficient dictionary with an adaptive number of atoms. Moreover the noise and sparsity levels are also inferred so that no parameter tuning is needed. We elaborate on the IBP-DL model to propose a model for linear inverse problems such as inpainting and compressive sensing beyond basic denoising. We derive a collapsed and an accelerated Gibbs samplers and propose a marginal maximum a posteriori estimator of the dictionary. Several image processing experiments are presented and compared to other approaches for illustration.

Keywords: sparse representations, dictionary learning, inverse problems, Indian Buffet Process, Bayesian non parametric.

☆Thanks to the BNPSI ANR project no ANR-13-BS-03-0006-01 and to the Fondation Ecole Centrale Lille for funding.

Email addresses: `hong_phuong.dang@ec-lille.fr` (Hong-Phuong Dang), `pierre.chainais@ec-lille.fr` (Pierre Chainais)

1. Introduction

Ill-posed inverse problems such as denoising, inpainting, deconvolution or super resolution in image processing do not have a unique solution due to missing information. External information is necessary to select a plausible solution. Prior information or regularization techniques often rely on the choice of a well suited representation space to identify a unique solution. In recent years, sparse representations [1, 2] have opened new avenues in signal and image processing. Sparsity refers to parsimonious representations where only a small number of components (or atoms) is used to describe the data in a possibly redundant dictionary. For instance, one can think of a continuous wavelet transform. Parsimony and sparsity have originated many successes in the solution of inverse problems.

Dictionary learning (DL) permits to learn such a sparse representation [1] from data themselves. Many works [3, 4, 5, 6] have shown the efficiency of DL to solve ill-posed inverse problems. Redundant dictionaries gather a number K of atoms potentially greater than the dimension P of the data space. In contrast with the mathematical construction of functional frames (e.g. wavelets), DL aims at learning an adaptive set of relevant atoms for sparse representation from data themselves.

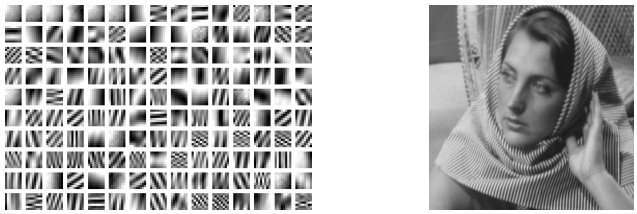
Many DL methods rooted in the seminal work by Olshausen & Field [2] are based on solving an optimization problem. Typically, the approaches in [3, 4, 5] proposed an optimal dictionary with a large size of 256 or 512 that is fixed in advance. Fast online dictionary learning has also been proposed in [6, 7]. Sparsity is typically promoted by an L0 or L1 penalty term on the set of encoding coefficients. Despite their good numerical efficiency, one main limitation of these approaches is that they most often set the size of the dictionary in advance, and need to know the noise level and tune the sparsity level. A few works have elaborated on the K-SVD approach [3] to propose adaptive dictionary learning (DL) methods that infer the size of the dictionary [8, 9, 10, 11] but they still often call for important parameter tuning.

Bayesian approaches have been much less studied. In [12], a Bayesian DL method named BPFA was proposed thanks to a Beta-Bernoulli model where sparsity is promoted through an adapted Beta-Bernoulli prior to enforce many encoding coefficients to zero. BPFA corresponds to a parametric approximation of the *Indian Buffet Process (IBP)* [13] since this approach works with a (large) fixed number of atoms. In [14], we introduced a truly *Bayesian non parametric (BNP)* approach namely the Indian Buffet Process dictionary learning (IBP-DL) thanks to the use of a true IBP prior. Such a prior both promotes sparsity and deals with an adaptive number of atoms. IBP-DL starts from an empty dictionary to learn a dictionary of growing adaptive size. It does not need to tune any parameter since the noise level and the sparsity level are sampled as well. The IBP-DL model in [14] aimed at solving the basic denoising problem only and detailed computations and algorithms were not described.

The present contribution presents the IBP-DL approach to solve linear inverse problems such as inpainting (missing pixels) or compressive sensing (random projections) in the presence of additive Gaussian noise. We derive several Gibbs sampling algorithms. Beyond the simple Gibbs sampler, we derive a collapsed Gibbs sampler and an accelerated Gibbs sampler in the spirit of [15] to solve the inpainting problem. Moreover, we propose a marginal maximum a posteriori estimate for inference of the dictionary and corresponding encoding coefficients. For reproducible research, Matlab codes will be made available from our websites[1].

Section 2 recalls about dictionary learning and the class of problems of interest. Section 3 presents the Indian Buffet Process (IBP) prior. Section 4 describes the IBP-DL observation model for linear Gaussian inverse problems. Section 5 describes various Gibbs sampling algorithms and the marginal maximum a posteriori (mMAP) estimator for inference. Section 6 illustrates the relevance of our approach on several image processing examples including denoising, inpainting and compressive sensing with comparison to other DL methods. Section 7

[1] `http://www.hongphuong-dang.com/publications.html`



(a)Dictionary : 150 atoms (b) A segment of Barbara image

Figure 1: IBP-DL dictionary of 150 atoms on a segment of size 256×256 of Barbara image

concludes and discusses some directions for future work.

2. Dictionary Learning (DL)

In image processing, it is usual to deal with local information by decomposing an image into a set of small patches, as in JPEG compression. Then each vector $\mathbf{y}_i \in \mathbb{R}^P$ represents a patch of size $\sqrt{P} \times \sqrt{P}$ (usually 8×8) casted in vectorial form according to lexicographic order. Let matrix $\mathbf{Y} \in \mathbb{R}^{P\times N}$ a set of N observations $\mathbf{y}_i$. For instance, fig. 1 displays a 256×256 segment of Barbara image from which a full data set of $N = (256-7)^2 = 62001$ overlapping 8×8 patches is extracted. The matrix $\mathbf{H}$ is the observation operator of patches $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_N] \in \mathbb{R}^{P\times N}$ in the initial image. It can be a binary mask in the inpainting problem or a random projection matrix in the case of compressive sensing. The additive noise $\boldsymbol{\varepsilon} \in \mathbb{R}^{P\times N}$ is assumed to be Gaussian i.i.d.. As a consequence, observations are modeled by

$$\begin{aligned} \mathbf{Y} &= \mathbf{H}\mathbf{X} + \boldsymbol{\varepsilon} \\ \mathbf{X} &= \mathbf{D}\mathbf{W} \end{aligned} \tag{1}$$

where $\mathbf{D}$=$[\mathbf{d}_1, ..., \mathbf{d}_K] \in \mathbb{R}^{P\times K}$ is the dictionary of K atoms and the encoding coefficients matrix $\mathbf{W}$= $[\mathbf{w}_1, ..., \mathbf{w}_N] \in \mathbb{R}^{K\times N}$. Dictionary learning can be seen as a matrix factorization problem: recovering $\mathbf{X}$ becomes equivalent to finding an (in some sense) optimal couple ($\mathbf{D}$, $\mathbf{W}$). The sparse representation of $\mathbf{x}_i$ is encoded by coefficients $\mathbf{w}_i$. Various approaches have been proposed, see [1]

for a review, to solve this problem by alternate optimization on $\mathbf{D}$ and $\mathbf{W}$. When working on image patches of size 8×8 (in dimension $P = 64$), these methods usually learn a dictionary of size 256 or 512 [3]. Sparsity is typically imposed through a L0 or L1-penalty terms on $\mathbf{W}$. Note that the weight of the regularization term is crucial and should decrease as the noise level $\sigma_{\boldsymbol{\varepsilon}}$ increases so that some parameter tuning is often necessary.

In the Bayesian framework, the problem is written in the form of a Gaussian likelihood according to the model (1). The prior $p(\mathbf{D}, \mathbf{W}, \sigma_{\boldsymbol{\varepsilon}})$ plays the role of a regularization and the joint posterior reads:

$$p(\mathbf{D}, \mathbf{W}, \sigma_{\boldsymbol{\varepsilon}} \mid \mathbf{Y}, \mathbf{H}) \propto p(\mathbf{Y} \mid \mathbf{H}, \mathbf{D}, \mathbf{W}, \sigma_{\boldsymbol{\varepsilon}}) p(\mathbf{D}, \mathbf{W}, \sigma_{\boldsymbol{\varepsilon}}) \tag{2}$$

Using Gibbs sampling, the problem can be solved by alternately sampling $\mathbf{D}$, $\mathbf{W}$, and $\sigma_{\boldsymbol{\varepsilon}}$. In the Bayesian non parametric (BNP) framework, the dictionary can be learnt without setting the size in advance nor tuning the noise level or the sparsity parameter. We present the general IBP-DL approach that generalizes our previous work [14], which dealt with denoising only, to linear inverse problems. The proposed approach uses an Indian Buffet Process (IBP) prior to both promote sparsity and deal with an adaptive number of atoms. Fig. 1 shows an example of the dictionary learnt from a segment of Barbara image without noise and the resulting reconstruction using the 150 atoms inferred by IBP-DL. Prior to a detailed description of the IBP-DL model for linear inverse problems, we briefly recall about the Indian Buffet Process.

3. Indian Buffet Process (IBP) and latent feature models

Bayesian non parametric methods permit to define prior distributions on random measures. Such random measures live in an infinitely dimensional space so and permit to deal with models of potentially infinite dimension. They offer an interesting alternative to reversible jump MCMC methods and model selection approaches.

The popular Chinese restaurant process [16] permits to deal with clustering problems without prior knowledge of the number of clusters. Recall that the

Chinese restaurant process can be built by integrating out a Dirichlet process and considering the resulting distribution over partitions of a set of points. The Dirichlet process is the De Finetti mixing distribution underlying the Chinese restaurant process.

Turning to latent feature problems, the IBP [17, 13] can be introduced as a non parametric Bayesian prior on sparse binary matrices $\mathbf{Z}$ with a potentially infinite number of rows. The matrix $\mathbf{Z}$ encodes the set of features that are assigned to each observation: $\mathbf{Z}(k,i)$=1 if observation $\mathbf{y}_i$ owns feature $\mathbf{d}_k$. In a formal presentation, Thibaux & Jordan [18] showed that the IBP can be obtained by integrating a Beta process in a *Beta-Bernoulli process*. The beta process was developed originally by Hjort [19] as a Lévy process prior for hazard measures. In [18], the *Beta process* was extended for use in feature learning; it appears to be the De Finetti mixing distribution underlying the Indian buffet process.

Let $\mathbf{BP}(c, B_0)$ a Beta process with a *concentration parameter* c and a *base measure* B_0. To draw B from a Beta process distribution, one draws a set of pairs (ω_k,π_k) from a marked Poisson point process on $\Omega \times [0,1]$, that can be represented as $B = \sum_k \pi_k \delta_{\omega_k}$. Here δ_{ω_k} is a Dirac distribution at $\omega_k \sim B_0$ with π_k its mass in B and $\sum_k \pi_k$ does not need to equal 1. When B_0 is *discrete*, for each atom ω_k, $\pi_k \sim \text{Beta}(cq_k, c(1-q_k))$ where $q_k \in (0,1)$ is the weight of the kth point (later on *feature* or *atom*) in measure B. Then one defines $\mathbf{z}_n \sim \mathbf{BeP}(B)$ i.i.d for $n = 1, ..., i-1$, a *Bernoulli process* with *hazard measure* B. If B is *discrete* then $\mathbf{z}_n = \sum_k b_k \delta_{\omega_k}$ where $b_k \sim \text{Bernoulli}(\pi_k)$. An important property is that the Beta process is conjugate to the Bernoulli process making $B \mid \mathbf{z}_1, ..., \mathbf{z}_{i-1}$ a Beta Process itself. Given a set of draws $\mathbf{z}_1, ..., \mathbf{z}_{i-1}$, a new draw $\mathbf{z}_i$ can be sampled according to

$$\mathbf{z}_i \mid \mathbf{z}_1, ..., \mathbf{z}_{i-1} \sim \mathbf{BeP}\left(\frac{c}{c+(i-1)} B_0 + \frac{1}{c+(i-1)} \sum_j^{i-1} \mathbf{z}_j\right) \tag{3}$$

where B has been marginalized out. A Bernoulli process is a Poisson process when the measure is *continuous*. Since a Bernoulli process is a particular kind

of Lévy process, it is the sum of two independent contributions with a mixed dicrete-continuous measure. When c=1 and B_0 is continuous with $B_0(\Omega)$=α, one recovers the classical generative process of the IBP introduced in [13] thanks to the following analogy. Let consider N customers (observations) entering an Indian restaurant where they select dishes (atoms) from an infinite buffet. The first customer tries a number Poisson(α) of dishes. Next i-th customer chooses the previously-selected dish k with a probability m_k/i, where m_k is the number of former customers who selected this dish k (before customer i). This step corresponds to the second term in eq.(3). Then the i-th customer tries an additional set of Poisson(α/i) new dishes; this corresponds to the first term in eq.(3). The behaviour of the IBP is governed by the parameter α which controls the expected number of dishes (features, atoms) used by N customers (observations, patches) since the expected total number of dishes is $\mathbb{E}[K] \simeq \alpha \log N$.

Table 1: List of symbols.

Symbol	**Description**
N, i	Number of observations, index of observations
P, ℓ	Dimension of an observation, index of dimension of an observation
K, k	Number of atoms, index of atoms
$\mathbf{Y}$, $\mathbf{y}_i$	Observation matrix, its i_{th} column observed vector
$\mathbf{W}$, $\mathbf{w}_i$,	Latent feature matrix, column of coefficients of observation i,
w_{ki}	k^{th} latent feature of observation i
$\mathbf{D}$, $\mathbf{d}_k$	Dictionary matrix, its kth column atom
$\mathbf{H}$, $\mathbf{H}_i$	Set of observation operators and operator matrix for i^{th} observation
$\boldsymbol{\Sigma}$, $\boldsymbol{\mu}$	Covariance matrix and mean vector
σ^2, μ	Variance and expected value
$\mathcal{P}$, $\mathcal{N}$, $\mathcal{U}$, $\mathcal{G}$, $\mathcal{IG}$	Poisson, Gaussian, Uniform, gamma and inverse gamma distributions

4. The IBP-DL model for linear inverse problems

The model under study can be described $\forall 1 \leq i \leq N$ by

$$\mathbf{y}_i = \mathbf{H}_i \mathbf{x}_i + \boldsymbol{\varepsilon}_i \tag{4}$$

$$\mathbf{x}_i = \mathbf{D}\mathbf{w}_i \text{ where } \mathbf{w}_i = \mathbf{z}_i \odot \mathbf{s}_i, \tag{5}$$

$$\mathbf{d}_k \sim \mathcal{N}(0, P^{-1}\mathbb{I}_P), \forall k \in \mathbb{N} \tag{6}$$

$$\mathbf{Z} \sim IBP(\alpha), \tag{7}$$

$$\mathbf{s}_i \sim \mathcal{N}(0, \sigma_s^2 \mathbb{I}_K), \tag{8}$$

$$\boldsymbol{\varepsilon}_i \sim \mathcal{N}(0, \sigma_{\boldsymbol{\varepsilon}}^2 \mathbb{I}_P). \tag{9}$$

where $\odot$ is the Hadamard product. Fig. 2 shows the graphical model. Notations are in Table 1. The observation matrix $\mathbf{Y}$ contains N column vectors of dimension P, only $Q \leq P$ in the compressive sensing (CS) case. The representation coefficients are defined as $\mathbf{w}_i = \mathbf{z}_i \odot \mathbf{s}_i$, in the spirit of a parametric Bernoulli-Gaussian model. The vector $\mathbf{z}_i \in \{0,1\}^K$ encodes which columns of $\mathbf{D}$ among K are used to represent $\mathbf{y}_i$; $\mathbf{s}_i \in \mathbb{R}^K$ represents the coefficients used for this representation. The sparsity properties of $\mathbf{W}$ are induced by the sparsity of $\mathbf{Z}$ drawn from the IBP prior. The IBP-DL model deals with a potentially infinite number of atoms $\mathbf{d}_k$ so that the size of the dictionary is not limited a priori. The IBP prior plays the role of a regularization term that penalizes the number K of active (non zero) rows in $\mathbf{Z}$ since $\mathbb{E}[K] \simeq \alpha \log N$ in the IBP. Except for σ_D^2 that is fixed to $1/P$ to avoid a multiplicative factor indeterminacy, conjugate priors are used for parameters $\boldsymbol{\theta} = (\sigma_{\boldsymbol{\varepsilon}}^2, \sigma_S^2, \alpha)$: vague inverse Gamma distributions for variances with very small hyperparameters (c_0=d_0=e_0=f_0=10^{-6}) are used for $\sigma_{\boldsymbol{\varepsilon}}^2$, σ_S^2, and a $\mathcal{G}(1,1)$ for α associated to a Poisson law in the IBP. The gamma distribution is defined as $\mathcal{G}(x; a, b) = x^{a-1} b^a \exp(-bx)/\Gamma(a)$ for $x > 0$. Linear operators $\mathbf{H}_i$ damage observations $\mathbf{y}_i$ since $\mathbf{H}_i$ may be a non invertible matrix. The simplest problem is denoising when $\mathbf{H}_i$=$\mathbb{I}_P$ [14]. In the inpainting problem, see fig. 3, $\mathbf{H}_i$ is a diagonal binary matrix of size $P \times P$ where zeros indicate missing pixels. In the case of compressive sensing, $\mathbf{H}_i$ is a rectangular (random) projection matrix of size $Q \times P$ (typically Q $\ll$ P). In the following, we

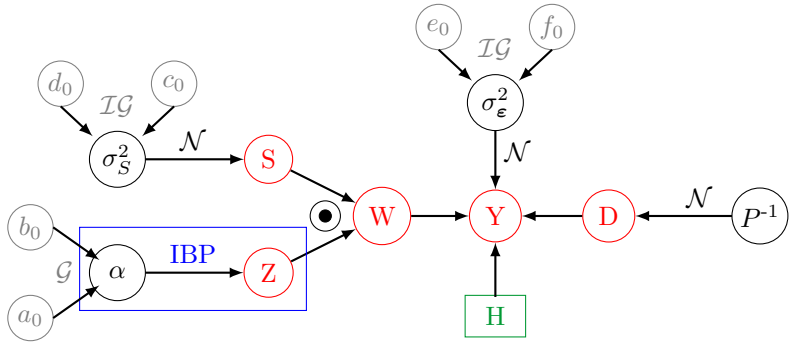
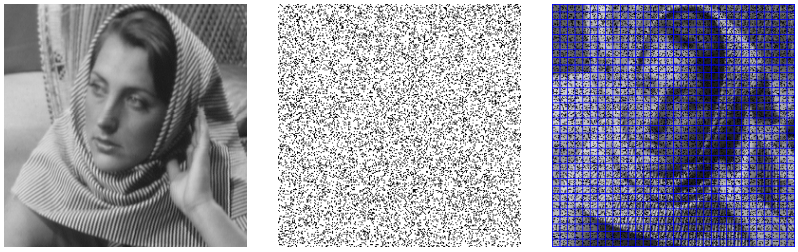


Figure 2: Graphical model of IBP-DL for linear inverse problems with additive Gaussian noise.



(a) (b) (c)

Figure 3: Inpainting problem: an image (a) is damaged by a mask (b) so that there are missing pixels in the observed image (c); the problem is solved by working on local patches.

describe Markov Chain Monte Carlo (MCMC) algorithms to generate samples according to the posterior distribution $p(\mathbf{Z}, \mathbf{D}, \mathbf{S}, \boldsymbol{\theta} \mid \mathbf{Y}, \mathbf{H}, \sigma_D^2)$.

5. MCMC algorithms for inference

This section details an MCMC sampling strategy to sample the posterior distribution $p(\mathbf{Z}, \mathbf{D}, \mathbf{S}, \boldsymbol{\theta} \mid \mathbf{Y}, \mathbf{H}, \sigma_D^2)$. Algorithm 1 summarizes this strategy. A Gibbs sampler is proposed. Some essential steps are described in Algorithms 2, 3 & 4. The sampling of $\mathbf{Z}$ with an IBP prior calls for some special care. Sampling $\mathbf{Z}$ goes in 2 steps: one first samples z_{ki} for active atoms, then a number of new atoms are sampled. In addition to Gibbs sampling, we also present the collapsed and accelerated Gibbs sampling for inference of IBP, see Algo. 2.

Init. : K=0, $\mathbf{Z}$=$\emptyset$, $\mathbf{D}$=$\emptyset$, α=1, σ_D^2=P^{-1}, σ_S^2=1, $\sigma_{\boldsymbol{\varepsilon}}$

Result: $\mathbf{D} \in \mathbb{R}^{P\times K}, \mathbf{Z} \in \{0;1\}^{K\times P}, \mathbf{S} \in \mathbb{R}^{K\times P}$, $\sigma_{\boldsymbol{\varepsilon}}$

for *each iteration t*

  for *data i=1:N*

    for $k \in k_{used} \leftarrow find(m_{-i} \neq 0)$

      Sample $\mathbf{Z}(k,i)$ see eq. (14)

    Infer new atoms, new coefficients see eq. (21)

  Sample $\mathbf{D}$, $\mathbf{S}$, see eq. (42), eq. (44)

  Sample $\boldsymbol{\theta} = (\sigma_{\boldsymbol{\varepsilon}}^2, \sigma_S^2, \alpha)$ see eq. (45), (46), (47)

Algorithm 1: Pseudo-algorithm of sampling for IBP-DL method.

5.1. Sampling of $\boldsymbol{Z}$

$\mathbf{Z}$ is a matrix with an infinite number of rows but only non-zero rows are kept in memory. In practice, one deals with finite matrices $\mathbf{Z}$ and $\mathbf{S}$. Let $m_{k,-i}$ the number of observations other than i using atom k. The 2 steps to sample $\mathbf{Z}$ are:

1. update the $z_{ki} = \mathbf{Z}(k,i)$ for 'active' atoms k such that $m_{k,-i} > 0$ (at least 1 patch other than i uses $\mathbf{d}_k$);
2. add new rows to $\mathbf{Z}$ which corresponds to activating new atoms in dictionary $\mathbf{D}$ thanks to a Metropolis-Hastings step.

We elaborate on previous works in [13, 15] where the binary-valued latent features model was considered to derive various versions of the Gibbs sampler. We describe usual Gibbs sampling for linear inverse problems, collapsed Gibbs sampling (CGS) for inpainting, and accelerated Gibbs sampling (AGS) for inpainting. Note that any linear inverse problem can be considered by using a usual Gibbs sampler. It appears that collapsed and accelerated Gibbs sampling can be derived for the inpainting (a fortiori denoising) problem which may not be the case for other problems.

5.1.1. Gibbs sampling

In this approach, $\mathbf{Z}$ is sampled from the posterior distribution.

$$P(\mathbf{Z} \mid \mathbf{Y}, \mathbf{H}, \mathbf{D}, \mathbf{S}, \sigma_{\boldsymbol{\varepsilon}}^2, \alpha) \propto p(\mathbf{Y}|\mathbf{H}, \mathbf{D}, \mathbf{Z}, \mathbf{S}, \sigma_{\boldsymbol{\varepsilon}}^2) p(\mathbf{Z} \mid \alpha) \tag{10}$$

Sampling z_{ki} for active atoms. $\mathbf{Z}(k, i)$ is simulated from a Bernoulli distribution weighted by likelihoods for each couple (patch i, atom k). The prior term is

$$P(z_{ki} = 1|\mathbf{Z}_{k,-i}) = \frac{m_{k,-i}}{N} \tag{11}$$

The likelihood $p(\mathbf{Y}|\mathbf{H}, \mathbf{D}, \mathbf{Z}, \mathbf{S}, \sigma_{\boldsymbol{\varepsilon}}^2)$ is easily computed from the Gaussian noise model, see Appendix A for a detailed derivation of the sampler. From Bayes'rule,

$$P(z_{ki}|\mathbf{Y}, \mathbf{H}, \mathbf{D}, \mathbf{Z}_{-ki}, \mathbf{S}, \sigma_{\boldsymbol{\varepsilon}}) \propto \mathcal{N}(\mathbf{y}_i|\mathbf{H}_i\mathbf{D}(\mathbf{z}_i \odot \mathbf{s}_i), \sigma_{\boldsymbol{\varepsilon}}^2) P(z_{ki}|\mathbf{Z}_{-ki}) \tag{12}$$

so that the posterior probabilities that $z_{ki} = 0$ or 1 are proportional to (p_0, p_1) defined by

$$\begin{cases} p_0 & = 1 - m_{k,-i}/N \\ p_1 & = \frac{m_{k,-i}}{N} \exp\left[-\frac{1}{2\sigma_{\boldsymbol{\varepsilon}}^2}(s_{ki}^2 \mathbf{d}_k^T \mathbf{H}_i^T \mathbf{H}_i \mathbf{d}_k - 2 s_{ki} \mathbf{d}_k^T \mathbf{H}_i^T (\mathbf{y}_i - \mathbf{H}_i \sum_{j \neq k}^{K} \mathbf{d}_j z_{ji} s_{ji})\right] \end{cases} \tag{13}$$

As a result, z_{ki} can be drawn from the Bernoulli distribution

$$z_{ki} \sim \text{Bernoulli}\left(\frac{p_1}{p_0 + p_1}\right) \tag{14}$$

Sampling new atoms. Following [20], we use a Metropolis-Hastings method to sample the number k_{new} of new atoms. This is equivalent to deal with rows of $\mathbf{Z}$ such that $m_{k,-i} = 0$: this happens either when an atom is not used (inactive, not stored) or when it is used by patch i only. Rows with *singletons* have a unique coefficient 1 and zeros elsewhere: $z_{ki} = 1$ and $m_{k,-i} = 0$. Sampling the number of new atoms is equivalent to sampling the number of singletons since when a new atom is activated, it creates a new singleton. Let k_{sing} the number of such singletons in matrix $\mathbf{Z}$ and $\mathbf{D}_{sing}$ and $\mathbf{S}_{sing}$ the corresponding k_{sing} atoms and the associated coefficients. Let $k_{prop} \in \mathbb{N}$ a proposal for the new number of singletons, $\mathbf{D}_{prop}$ the k_{prop} new proposed atoms, $\mathbf{S}_{prop}$ the new proposed coefficients

corresponding to $\mathbf{D}_{prop}$. Thus the proposal is $\zeta_{prop} = \{k_{prop}, \mathbf{D}_{prop}, \mathbf{S}_{prop}\}$. Let J the proposal distribution, we propose a move $\zeta_{sing} \rightarrow \zeta_{prop}$ with a probability having the form :

$$J(\zeta_{prop}) = J_K(k_{prop}) J_D(\mathbf{D}_{prop}) J_S(\mathbf{S}_{prop}) \tag{15}$$

The proposal is accepted, that is $\zeta_{new} = \zeta_{prop}$, if a uniform random variable $u \in (0, 1)$ verifies

$$u \leq \min\left(1, a_{\zeta_{sing} \rightarrow \zeta_{propo}}\right) \tag{16}$$

where the acceptance threshold is

$$a_{\zeta_{sing} \rightarrow \zeta_{prop}} = \frac{P(\zeta_{prop} \mid \mathbf{Y}, rest) J(\zeta_{sing})}{P(\zeta_{sing} \mid \mathbf{Y}, rest) J(\zeta_{prop})} = \frac{p(\mathbf{Y}|\zeta_{prop}, rest)}{p(\mathbf{Y}|\zeta_{sing}, rest)} a_K a_D a_S \tag{17}$$

where

$$a_K = \frac{\mathcal{P}(k_{prop}; \alpha/N) J_K(k_{sing})}{\mathcal{P}(k_{sing}; \alpha/N) J_K(k_{prop})} \tag{18}$$

$$a_D = \frac{\mathcal{N}(\mathbf{D}_{prop}; 0, \sigma_D^2) J_D(\mathbf{D}_{sing})}{\mathcal{N}(\mathbf{D}_{sing}; 0, \sigma_D^2) J_D(\mathbf{D}_{prop})} \tag{19}$$

$$a_S = \frac{\mathcal{N}(\mathbf{S}_{prop}; 0, \sigma_S^2) J_S(\mathbf{S}_{sing})}{\mathcal{N}(\mathbf{S}_{sing}; 0, \sigma_S^2) J_S(\mathbf{S}_{prop})} \tag{20}$$

If one uses the prior as the proposal on ζ_{prop}, the acceptance threshold is simply governed by the likelihood ratio

$$a_{\zeta_{sing} \rightarrow \zeta_{prop}} = \frac{p(\mathbf{y}_i|\zeta_{prop}, rest)}{p(\mathbf{y}_i|\zeta_{sing}, rest)} \tag{21}$$

since $a_K = a_D = a_S = 1$ in this case.

5.1.2. Collapsed Gibbs sampling for inpainting

Gibbs sampling may be convenient for a model with a constant number of degrees of freedom (the number K of atoms) [12]. The IBP prior permits to overcome this restriction by providing a model with a potentially infinite number of atoms. The mixing times may be long especially if dimension P is large. As far as possible a collapsed Gibbs sampler is desirable to reduce the state space and therefore the convergence time. This is possible for the problem of inpainting.

One can integrate the dictionary $\mathbf{D}$ out in closed form thanks to conjugacy properties (the Gaussian prior on matrix $\mathbf{D}$). When a variable is integrated out at some step, this variable must be sampled before reusing it [21]. Therefore $\mathbf{D}$ must be sampled immediately after $\mathbf{Z}$ that is sampled from the collapsed posterior

$$P(\mathbf{Z} \mid \mathbf{Y}, \mathbf{S}, \mathbf{H}, \sigma_{\boldsymbol{\varepsilon}}^2, \sigma_D^2, \alpha) \propto p(\mathbf{Y} \mid \mathbf{H}, \mathbf{Z}, \mathbf{S}, \sigma_{\boldsymbol{\varepsilon}}^2, \sigma_D^2) P(\mathbf{Z} \mid \alpha) \tag{22}$$

In the case of inpainting, $\mathbf{H}_i$ is a binary diagonal matrix of size $P \times P$. Let $\{\mathbf{F}_\ell\}$ the set of binary diagonal matrices of size N for ℓ=1,...,P. If each $\mathbf{H}_i$ is associated with each patch $\mathbf{Y}(:,i)$ then $\mathbf{F}_\ell$ is associated with pixels $\mathbf{Y}(\ell,:)$ at position ℓ in all patches. $\mathbf{F}_\ell(i,i)$ indicates whether pixel at location ℓ in patch i is observed or not so that $\mathbf{F}_\ell(i,i)$=$\mathbf{H}_i(\ell,\ell)$=$H_{i,\ell}$.

$$\begin{aligned} & p(\mathbf{Y} \mid \{\mathbf{H}_i\}_{i=1:N}, \mathbf{Z}, \mathbf{S}, \sigma_{\boldsymbol{\varepsilon}}^2, \sigma_D^2) = p(\mathbf{Y} \mid \{\mathbf{F}_\ell\}_{l=1:P}, \mathbf{Z}, \mathbf{S}, \sigma_{\boldsymbol{\varepsilon}}^2, \sigma_D^2) \\ & = \int p(\mathbf{Y} \mid \{\mathbf{F}_\ell\}_{l=1:P}, \mathbf{D}, \mathbf{Z}, \mathbf{S}, \sigma_{\boldsymbol{\varepsilon}}) p(\mathbf{D} \mid \sigma_D) d\mathbf{D} \\ & = \frac{1}{(2\pi)^{\|\mathbf{Y}\|_0/2} \sigma_{\boldsymbol{\varepsilon}}^{\|\mathbf{Y}\|_0 - KP} \sigma_D^{KP}} \prod_{\ell=1}^{P} |\mathbf{M}_\ell|^{1/2} \exp\left[-\frac{1}{2\sigma_{\boldsymbol{\varepsilon}}^2} (\mathbf{Y}_\ell (\mathbb{I} - \mathbf{F}_\ell^T \mathbf{W}^T \mathbf{M}_\ell \mathbf{W} \mathbf{F}_\ell) \mathbf{Y}_\ell^T\right] \end{aligned} \tag{23}$$

where $\mathbf{M}_\ell = (\mathbf{W}\mathbf{F}_\ell\mathbf{F}_\ell^T\mathbf{W}^T + \frac{\sigma_{\boldsymbol{\varepsilon}}^2}{\sigma_D^2}\mathbb{I}_K)^{-1}$. The full derivation is detailed in Appendix B. Thus, Bernoulli distribution in (14) to sample z_{ki} depends on

$$\begin{cases} p_0 & = (1 - \frac{m_{k,-i}}{N}) p(\mathbf{Y} \mid \{\mathbf{F}_\ell\}_{l=1:P}, \mathbf{Z}, \mathbf{S}, \sigma_{\boldsymbol{\varepsilon}}^2, \sigma_D^2) \\ p_1 & = \frac{m_{k,-i}}{N} p(\mathbf{Y} \mid \{\mathbf{F}_\ell\}_{l=1:P}, \mathbf{Z}, \mathbf{S}, \sigma_{\boldsymbol{\varepsilon}}^2, \sigma_D^2) \end{cases} \tag{24}$$

Here, we chose to integrate $\mathbf{D}$ out so that we don't need to propose new atoms $\mathbf{D}_{prop}$ in Metropolis-Hastings step[2]. The proposal is now $\zeta_{prop} = \{k_{prop}, \mathbf{S}_{prop}\}$ and the acceptance ratio is simply governed by the collapsed likelihood ratio, see (23),

$$a_{\zeta_{sing} \to \zeta_{prop}} = \frac{p(\mathbf{Y} | \zeta_{prop}, rest)}{p(\mathbf{Y} | \zeta_{sing}, rest)} \tag{25}$$

Unfortunately, the proposal according to the prior rarely proposes new atoms because the parameter α/N of the Poisson law as soon as the number of observations N is large. The sampler mixes very slowly. As a remedy we propose to

[2]Note that we can choose to integrate $\mathbf{D}$ or $\mathbf{S}$ out, but not both.

modify the proposal distribution for the number k_{new} of new atoms. The idea is to distinguish small values of k_{new} from large ones thanks to some K_{max}. Then we propose to use the following distribution:

$$J_K(k_{prop}) = \pi \mathbb{1}_{(k_{prop} > K_{max})} \mathcal{P}(k_{prop}; \frac{\alpha}{N}) \tag{26}$$

$$+(1-\pi)\mathbb{1}_{(k_{prop} \in [0:K_{max}])} \mathcal{M}(p_k(0:K_{max})) \tag{27}$$

where

$$\pi = P(k > K_{max}; \frac{\alpha}{N}) = \sum_{k=K_{max}+1}^{\infty} \mathcal{P}(k; \frac{\alpha}{N}) \tag{28}$$

$$p_k(x) = \mathcal{P}(x; \frac{\alpha}{N}) \mathcal{N}(\mathbf{y}_i; \mu_{\mathbf{y}i}, \Sigma_{\mathbf{y}i}) = \mathcal{P}(k; \frac{\alpha}{N}) \prod_{l=1}^{P} \mathcal{N}(\mathbf{y}_i(l); \mu_{\mathbf{y}il}, \Sigma_{\mathbf{y}il}) \tag{29}$$

and $\mathcal{M}$ is a multinomal distribution, $\mathcal{P}$ is a Poisson distribution. Now, the proposal is accepted with the acceptance threshold :

$$a_{\zeta_{sing} \to \zeta_{propo}} = \frac{P(\zeta_{propo} \mid rest, \mathbf{Y}) J(\zeta_{sing})}{P(\zeta_{sing} \mid rest, \mathbf{Y}) J(\zeta_{propo})} \tag{30}$$

$$= \frac{p(\mathbf{Y}|\zeta_{propo}, -)}{p(\mathbf{Y}|\zeta_{sing}, -)} \frac{\mathcal{P}(k_{prop}; \alpha/N) J_K(k_{sing})}{\mathcal{P}(k_{sing}; \alpha/N) J_K(k_{prop})} \tag{31}$$

One limitation of this approach is its computational cost. The complexity per-iteration of the IBP sampler is $O(N^3(K^2 + KP))$ due to the matrix in the exponent of the collapsed likelihood (23). This motivates next section which derives an accelerated Gibbs sampling following ideas from [15].

5.1.3. Accelerated Gibbs sampling for inpainting

In [22], even though a rank one update for matrix inversion is used, there is still an expensive matrix multiplication in the exponent of the collapsed likelihood. This is the reason why we have followed the recommendations in [15] to perform an accelerated Gibbs sampling. The two methods and their complexity were compared in [23]. A study comparing the speed of different Gibbs samplers for $\mathbf{Z}$ when $\{\mathbf{H}_i\}$ are identity matrices can be found in [15].

The computational time of the (collapsed) Gibbs sampler will be long due to the repeated computation of likelihoods. However one can derive an accelerated Gibbs sampler as suggested in [15] that achieves a computational cost of

$O(N(K^2 + KP))$ by proposing to maintain the $\mathbf{D}$ posterior instead of integrating $\mathbf{D}$ out completely. The derivation of this sampler is detailed in Appendix C. The implementation is described by Algorithm 2.

In section 5.1.2, $\mathbf{Z}$ was sampled by integrating $\mathbf{D}$ out. In the derivation of Appendix B, we show that the posterior of $\mathbf{D}$ has a Gaussian form.

$$p(\mathbf{D} \mid \mathbf{Y}, \mathbf{F}, \mathbf{Z}, \mathbf{S}, \sigma_{\boldsymbol{\varepsilon}}^2, \sigma_D^2) \propto \prod_{\ell=1}^{P} \mathcal{N}(\mathbf{D}(\ell,:); \boldsymbol{\mu}_{\mathbf{D}(\ell,:)}, \boldsymbol{\Sigma}_{\mathbf{D}(\ell,:)}) \tag{32}$$

$$\begin{aligned} \text{where} \quad & \boldsymbol{\Sigma}_{\mathbf{D}(\ell,:)} = \sigma_{\boldsymbol{\varepsilon}}^2 \mathbf{M}_\ell \\ & \boldsymbol{\mu}_{\mathbf{D}(\ell,:)} = \mathbf{Y}(\ell,:)\mathbf{F}_\ell^T \mathbf{W}^T \mathbf{M}_\ell \end{aligned} \tag{33}$$

The main idea is to work with *rows* of $\mathbf{D}$ in place of columns (atoms) as usual. The observations and the feature assignement matrices can split in two parts according to $\mathbf{Y} = [\mathbf{y}_i, \mathbf{Y}_{-i}]$ so that

$$\begin{aligned} & P(z_{ki} = 1 \mid \mathbf{Y}, \mathbf{H}, \mathbf{W}, \sigma_D, \sigma_{\boldsymbol{\varepsilon}}, \alpha) \propto \frac{m_{k,-i}}{N} \times \\ & \int p(\mathbf{y}_i \mid \mathbf{H}_i, \mathbf{w}_i, \mathbf{D}) p(\mathbf{Y}_{-i} \mid \mathbf{H}_{\neq i}, \mathbf{W}_{-i}, \mathbf{D}) p(\mathbf{D} \mid \sigma_D) d\mathbf{D} \\ & \propto \frac{m_{k,-i}}{N} \int p(\mathbf{y}_i \mid \mathbf{H}_i, \mathbf{D}, \mathbf{w}_i) \prod_{\ell=1}^{P} p(\mathbf{D}(\ell,:) \mid \mathbf{F}_\ell, \mathbf{W}_{-i}, \sigma_D) d\mathbf{D} \end{aligned} \tag{34}$$

One can show that the posterior of $\mathbf{D}$ is a gaussian distribution with expectation $\boldsymbol{\mu}_{\mathbf{D}\ell}$ and covariance $\boldsymbol{\Sigma}_{\mathbf{D}\ell}$. The posterior of $\mathbf{D}$ given all of the data *except data* i is also easily determined thanks to the use of *sufficient statistics*

$$\begin{aligned} g_{\mathbf{D}\ell} &= \boldsymbol{\Sigma}_{\mathbf{D}\ell}^{-1} = (1/\sigma_{\boldsymbol{\varepsilon}}^2)\mathbf{M}_\ell^{-1} \\ h_{\mathbf{D}\ell} &= \boldsymbol{\mu}_{\mathbf{D}\ell} g_{\mathbf{D}\ell} = (1/\sigma_{\boldsymbol{\varepsilon}}^2)\mathbf{Y}(\ell,:)\mathbf{F}_\ell^T \mathbf{W}^T \end{aligned} \tag{35}$$

This makes it easy to deal with the influence of one individual observation i apart. Indeed, one can define

$$g_{\mathbf{D}\ell,\pm i} = g_{\mathbf{D}\ell} \pm \sigma_{\boldsymbol{\varepsilon}}^{-2} H_{i,\ell}\ \mathbf{w}_i \mathbf{w}_i^{\mathrm{T}} \tag{36}$$

$$h_{\mathbf{D}\ell,\pm i} = h_{\mathbf{D}\ell} \pm \sigma_{\boldsymbol{\varepsilon}}^{-2} H_{i,\ell}\ y_i(\ell)\ \mathbf{w}_i^{\mathrm{T}} \tag{37}$$

as well as the corresponding $\boldsymbol{\mu}_{\mathbf{D}\ell,\pm i}$ and $\boldsymbol{\Sigma}_{\mathbf{D}\ell,\pm i}$. Since the likelihood is Gaussian, (34) yields

$$P(z_{ki} = 1 \mid \mathbf{Y}, \mathbf{H}, \mathbf{W}, \sigma_D, \sigma_{\boldsymbol{\varepsilon}}, \alpha) \propto \frac{m_{k,-i}}{N} \prod_{\ell=1}^{P} \mathcal{N}(\mathbf{y}_i(\ell); \mu_{yi\ell}, \sigma_{yi\ell}) \tag{38}$$

where

$$\mu_{yi\ell} = H_{i,\ell}\boldsymbol{\mu}_{\mathbf{D}(\ell,:),-i}\mathbf{w}_i \tag{39}$$

$$\sigma_{yi\ell} = H_{i,\ell}\mathbf{w}_i^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{D}(\ell,:),-i}\mathbf{w}_i + \sigma_{\boldsymbol{\varepsilon}}^2 \tag{40}$$

At each iteration on observation i, one uses (36)& (37) to remove/restore the influence of data i on the posterior distribution of $\mathbf{D}$ and therefore on the posterior of z_{ki}. Once $\mathbf{z}_i$ is sampled, we *restore* the influence of $\mathbf{y}_i$ into this posterior. The accelerated sampling [15] can reduce the complexity to $O(N(K^2 + KP))$, see Algorithm 2.

Init. : K=0, $\mathbf{Z}$=$\emptyset$, $\mathbf{D}$=$\emptyset$, α=1, σ_D^2=P^{-1}, σ_S^2=1, $\sigma_{\boldsymbol{\varepsilon}}$
Result: $\mathbf{D} \in \mathbb{R}^{P\times K}, \mathbf{Z} \in \{0;1\}^{K\times P}, \mathbf{S} \in \mathbb{R}^{K\times P}$, $\sigma_{\boldsymbol{\varepsilon}}$
for *each iteration t*
  Use information form of the $\mathbf{D}$ posterior according to (35)
  for *data i=1:N*
    Remove influence of data i from the $\mathbf{D}$ posterior via eq.(36),(37)
    $m_{-i} \in \mathbb{N}^{K\times 1} \leftarrow \sum \mathbf{Z}(:,-i)$
    for $k \in \{k : m_{-i} \neq 0\}$
      Infer $\mathbf{Z}(k,i)$, see **Algo. 3**
    Infer new atoms, new coefficients, see **Algo. 4**
    Restore influence of data i into the $\mathbf{D}$ posterior via eq.(36),(37)
    for *atoms k=1:K*
      Sample $\mathbf{d}_k$ eq. (42)
      Sample $\mathbf{s}_k$ eq. (44)
  Sample $\sigma_{\boldsymbol{\varepsilon}}$, σ_S, α see eq. (45), (46), (47)

Algorithm 2: Pseudo-algorithm of sampling by accelerated Gibbs sampling of IBP-DL method for inpainting. See also **Algo. 3** and **4** for details.

In practice, we need $\boldsymbol{\Sigma}_{\mathbf{D}(\ell,:)} = g_{\mathbf{D}(\ell,:)}^{-1}$ rather than $g_{\mathbf{D}(\ell,:)}$. This quantity can be directly updated thanks to the matrix inversion lemma. One can easily add or remove the influence of a single data i from $\boldsymbol{\Sigma}_{\mathbf{D}(\ell,:)}$, see Appendix D. Algorithm

3 presents the sampling algorithm of z_{ki} by accelerated Gibbs sampling. In practice, we work with the matrix $\mathbf{W}$ to save memory space.

Likelihood in case z_{ki}=1

- **if** $\mathbf{w}_i(k) = 0$
  - $\mathbf{w}_i(k) \sim \mathcal{N}(0, \sigma_S^2)$;
- $tmp \leftarrow \mathbf{w}_i(k)$;
- $\boldsymbol{\mu}_{yi} \leftarrow \mathbf{H}_i \boldsymbol{\mu}_{D,-i} \mathbf{w}_i$ (39) ;
- **for** *dimension ℓ = 1 : P*
  - $\Sigma_{yi}(\ell) \leftarrow H_{i,\ell} \mathbf{w}_i^{\mathrm{T}} \boldsymbol{\Sigma}_{\mathbf{D}-i}\{l\} \mathbf{w}_i + \sigma_{\boldsymbol{\varepsilon}}^2$ (40);
- $p_1 \leftarrow \dfrac{m_{k-i}}{N} \displaystyle\prod_{\ell=1}^{P} \mathcal{N}(\mathbf{y}_i(\ell); \boldsymbol{\mu}_{yi}(\ell), \Sigma_{yi}(\ell))$;

Likelihood in case z_{ki}=0

- $\mathbf{w}_i(k) \leftarrow 0$;
- $\boldsymbol{\mu}_{yi} \leftarrow \mathbf{H}_i \boldsymbol{\mu}_{D,-i} \mathbf{w}_i$ (39) ;
- **for** *dimension ℓ = 1 : P*
  - $\Sigma_{yi}(\ell) \leftarrow H_{i,\ell} \mathbf{w}_i^{\mathrm{T}} \boldsymbol{\Sigma}_{\mathbf{D}-i}\{l\} \mathbf{w}_i + \sigma_{\boldsymbol{\varepsilon}}^2$ (40);
- $p_0 \leftarrow \displaystyle\prod_{\ell=1}^{P} \mathcal{N}(\mathbf{y}_i(\ell); \boldsymbol{\mu}_{yi}(\ell), \Sigma_{yi}(\ell))$ (1-$\frac{m_{-i}(k)}{N}$);

$z_{ki} \sim \text{Bernoulli}\left(\dfrac{p_1}{p_1 + p_0}\right)$;

if $z_{ki} = 1$

- $\mathbf{w}_i(k) \leftarrow tmp$;

Algorithm 3: Algorithm for sample **Z**(k,i) of accelerated Gibbs sampling of IBP-DL method for inpainting, see **Algo. 2**

When sampling new atoms, the proposal can be either the prior distribution or the distribution we proposed in section 5.1.2. When a data i proposes k_{new} atoms, the acceptance threshold depends on the likelihood in (38) which itself depends on $\boldsymbol{\mu}_{\mathbf{D}_{new}(\ell,:),-i} = 0$ and $\boldsymbol{\Sigma}_{\mathbf{D}_{new}(\ell,:),-i} = \sigma_D^2 \mathbb{I}_{k_{new}}$. As a consequence, Algo. 4 uses prior distributions as proposal to sample new atoms as well as new coefficients.

Former singletons

$singletons \leftarrow \{k : m_{k-i} = 0 \ \& \ \mathbf{w}_i \neq 0\}$; % Find the singletons

$\boldsymbol{\mu}_{yi} \leftarrow \mathbf{H}_i \boldsymbol{\mu}_{D,-i} \mathbf{w}_i$;

for *each dimension* $\ell = 1 : P$

$\Sigma_{yi}(\ell) \leftarrow H_{i,\ell} \mathbf{w}_i^{\mathrm{T}} \boldsymbol{\Sigma}_{\mathbf{D}-i}\{l\} \mathbf{w}_i + \sigma_{\boldsymbol{\varepsilon}}^2$;

$p_{sing} \leftarrow \prod_{\ell=1}^{P} \mathcal{N}(\mathbf{y}_i(\ell); \boldsymbol{\mu}_{yi}(\ell), \Sigma_{yi}(\ell))$;

% eq.(38) with the singletons

New proposed singletons

$k_{prop} \sim \mathcal{P}(\alpha/N)$;

$\mathbf{w}_{prop} \leftarrow \mathbf{w}_i$;

$\mathbf{w}_{prop}(singletons) \leftarrow 0$; % Remove former singletons

$\mathbf{s}_{prop} \in \mathbb{R}^{k_{prop} \times 1} \sim \mathcal{N}(0, \sigma_S^2)$; % Propose new singletons

$\boldsymbol{\mu}_{yi} \leftarrow \mathbf{H}_i \boldsymbol{\mu}_{D,-i} \mathbf{w}_{prop}$;

for *each dimension* $\ell = 1 : P$

$\Sigma_{yi}(l) \leftarrow H_{i,\ell} \mathbf{w}_{prop}^{\mathrm{T}} \boldsymbol{\Sigma}_{\mathbf{D}-i}\{l\} \mathbf{w}_{prop} + \sigma_{\boldsymbol{\varepsilon}}^2 + H_{i,\ell} \mathbf{s}_{prop}^T \sigma_D^2 \mathbf{s}_{prop}$;

$p_{prop} \leftarrow \prod_{\ell=1}^{P} \mathcal{N}(\mathbf{y}_i(\ell); \boldsymbol{\mu}_{yi}(\ell), \Sigma_{yi}(\ell))$;

% eq.(38) with the new singletons

if $min\left(\frac{p_{prop}}{p_{sing}}, 1\right) > \mathcal{U}_{[0,1]}$

$\mathbf{w}_i = [\mathbf{w}_{prop}; \mathbf{s}_{prop}]$;

for *each dimension* $\ell = 1 : P$

$$\boldsymbol{\Sigma}_{\mathbf{D}-i}\{l\} \leftarrow \begin{bmatrix} \boldsymbol{\Sigma}_{\mathbf{D}-i}\{l\} & 0 \\ 0 & \sigma_D^2 \mathbb{I}_{k_{prop}} \end{bmatrix} ;$$

$h_{D,-i} \leftarrow [h_{D,-i} \quad zeros(P, k_{prop})]$;

Algorithm 4: Metropolis-Hastings algorithm using prior like proposal to infer new atoms and new coefficients when using the accelerated Gibbs Sampling of IBP-DL method for inpainting, see **Algo. 2**

5.2. Sampling $\boldsymbol{D}$

The dictionary $\mathbf{D}$ can be sampled by Gibbs sampling. The posterior of each atom $\mathbf{d}_k$ is given by

$$p(\mathbf{d}_k|\mathbf{Y},\mathbf{H},\mathbf{Z},\mathbf{S},\mathbf{D}_{-k},\boldsymbol{\theta}) \propto \prod_{i=1}^{N} \mathcal{N}(\mathbf{y}_i;\mathbf{H}_i\mathbf{D}\mathbf{w}_i,\sigma_{\boldsymbol{\varepsilon}}^2\mathbb{I}_{\|\mathbf{H}_i\|_0})\mathcal{N}(\mathbf{d}_k;0,P^{-1}\mathbb{I}_P) \quad (41)$$

so that

$$\begin{aligned} &p(\mathbf{d}_k|\mathbf{Y},\mathbf{H},\mathbf{Z},\mathbf{S},\mathbf{D}_{-k},\boldsymbol{\theta}) \propto \mathcal{N}(\boldsymbol{\mu}_{\mathbf{d}_k},\boldsymbol{\Sigma}_{\mathbf{d}_k}) \\ &\boldsymbol{\Sigma}_{\mathbf{d}_k} = (\sigma_D^{-2}\mathbb{I}_P + \sigma_{\boldsymbol{\varepsilon}}^{-2}\sum_{i=1}^{N} w_{ki}^2\mathbf{H}_i^T\mathbf{H}_i)^{-1} \\ &\boldsymbol{\mu}_{\mathbf{d}_k} = \sigma_{\boldsymbol{\varepsilon}}^{-2}\boldsymbol{\Sigma}_{\mathbf{d}_k}\sum_{i=1}^{N} w_{ki}(\mathbf{H}_i^T\mathbf{y}_i - \mathbf{H}_i^T\mathbf{H}_i\sum_{j\neq k}^{K}\mathbf{d}_j w_{ji}) \end{aligned} \quad (42)$$

5.3. Sampling $\boldsymbol{S}$

The posterior of each element s_{ki} of $\mathbf{S}$ is given in (44).

$$p(s_{ki}|\mathbf{Y},\mathbf{H},\mathbf{D},\mathbf{Z},\mathbf{S}_{k,-i},\boldsymbol{\theta}) \propto \mathcal{N}(\mathbf{y}_i;\mathbf{H}_i\mathbf{D}(\mathbf{s}_i\odot\mathbf{z}_i),\sigma_{\boldsymbol{\varepsilon}}^2\mathbb{I}_{\|\mathbf{H}_i\|_0})\mathcal{N}(\mathbf{s}_i;0,\sigma_S^2\mathbb{I}_K) \quad (43)$$

so that

$$\begin{aligned} &p(s_{ki}|\mathbf{Y},\mathbf{H}_i,\mathbf{D},\mathbf{Z},\mathbf{S}_{k,-i},\boldsymbol{\theta}) \propto \mathcal{N}(\mu_{s_{ki}},\Sigma_{s_{ki}}) \\ &z_{ki}=1 \Rightarrow \begin{cases} \Sigma_{s_{ki}} = (\sigma_{\boldsymbol{\varepsilon}}^{-2}\mathbf{d}_k^T\mathbf{H}_i^T\mathbf{H}_i\mathbf{d}_k + \sigma_S^{-2})^{-1} \\ \mu_{s_{ki}} = \sigma_{\boldsymbol{\varepsilon}}^{-2}\Sigma_{s_{ki}}\mathbf{d}_k^T(\mathbf{H}_i^T\mathbf{y}_i - \mathbf{H}_i^T\mathbf{H}_i\sum_{j\neq k}^{K}\mathbf{d}_j w_{ji}) \end{cases} \\ &z_{ki}=0 \Rightarrow \begin{cases} \Sigma_{s_{ki}} = \sigma_S^2 \\ \mu_{s_{ki}} = 0 \end{cases} \end{aligned} \quad (44)$$

5.4. Sampling σ_ε^2, σ_S^2, α

The other parameters are sampled according to their posterior that is easily obtained thanks to conjugacy properties:

$$\begin{aligned} &p(\sigma_{\boldsymbol{\varepsilon}}^{-2} \mid \mathbf{Y},\mathbf{H},\mathbf{D},\mathbf{W}) \propto \prod_{i=1}^{N}\mathcal{N}(\mathbf{y}_i \mid \mathbf{H}_i\mathbf{D}\mathbf{w}_i,\sigma_{\boldsymbol{\varepsilon}}^2\mathbb{I}_{\|\mathbf{H}_i\|_0})\mathcal{G}(\sigma_{\boldsymbol{\varepsilon}}^{-2} \mid c_0,d_0) \\ &\sigma_{\boldsymbol{\varepsilon}}^{-2} \sim \mathcal{G}\left(c_0 + \tfrac{1}{2}\sum_{i=1}^{N}\|\mathbf{H}_i\|_0, d_0 + \tfrac{1}{2}\sum_{i=1}^{N}\|\mathbf{y}_i - \mathbf{H}_i\mathbf{D}\mathbf{w}_i\|_2^2\right) \end{aligned} \quad (45)$$

$$\begin{aligned} &p(\sigma_S^{-2} \mid \mathbf{S}) \propto \prod_{i=1}^{N}\mathcal{N}(\mathbf{s}_i \mid 0,\sigma_S^2\mathbb{I}_K)\Gamma(\sigma_S^{-2} \mid c_0,d_0) \\ &\sigma_S^{-2} \sim \mathcal{G}\left(e_0 + \tfrac{KN}{2}, f_0 + \tfrac{1}{2}\sum_{i=1}^{N}\mathbf{s}_i^T\mathbf{s}_i\right) \end{aligned} \quad (46)$$

It is moreover possible to sample the concentration parameter of the Indian Buffet Process:

$$\begin{aligned} p(\alpha \mid K) &\propto \mathcal{P}(K \mid \alpha \sum_{j=1}^{N} \tfrac{1}{j})\mathcal{G}(\alpha \mid 1,1) \\ \alpha &\sim \mathcal{G}\left(1+K, 1+\sum_{j=1}^{N} 1/j\right) \end{aligned} \tag{47}$$

As a result, the proposed BNP approach is really non parametric since there remains no parameter to tune. Hyperparameters take very vague values and do not call for any optimal tuning.

5.5. Inference: marginalized MAP estimate

To alleviate notations, let $\boldsymbol{\theta} = (\sigma_{\boldsymbol{\varepsilon}}, \sigma_S, \alpha)$. A sequence $\{\mathbf{D}^{(t)}, \mathbf{Z}^{(t)}, \mathbf{S}^{(t)}, \boldsymbol{\theta}^{(t)}\}_{t=1}^{T_{MCMC}}$ is sampled by the MCMC algorithms. The purpose of this work is to restore damaged original $\mathbf{X}$ by using ($\mathbf{D}$, $\mathbf{W}$), see (1). The aim of this section is to define a relevant estimate of ($\mathbf{D}$, $\mathbf{W}$) for practical use in solving inverse problems. We derive in Appendix E the marginal posterior distribution resulting from the marginalization of the joint posterior distribution with respect to the nuisance parameters $\boldsymbol{\theta}$:

$$p(\mathbf{D}, \mathbf{Z}, \mathbf{S} \mid \mathbf{Y}, \mathbf{H}) = \int p(\mathbf{D}, \mathbf{Z}, \mathbf{S} \mid \mathbf{Y}, \mathbf{H}, \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta} \tag{48}$$

$$\propto \frac{1}{(2\pi\sigma_D^2)^{PK/2}} \exp\left(-\frac{\|\mathbf{D}\|_{\mathrm{F}}^2}{2\sigma_D^2}\right) \left(\sum_{i=1}^{N} \|\mathbf{y}_i - \mathbf{H}_i \mathbf{D} \mathbf{w}_i\|_{\mathrm{F}}^2\right)^{-N_0/2} \frac{\Gamma(NK/2)}{\pi^{NK/2} \|\mathbf{S}\|_{\mathrm{F}}^{NK}}$$

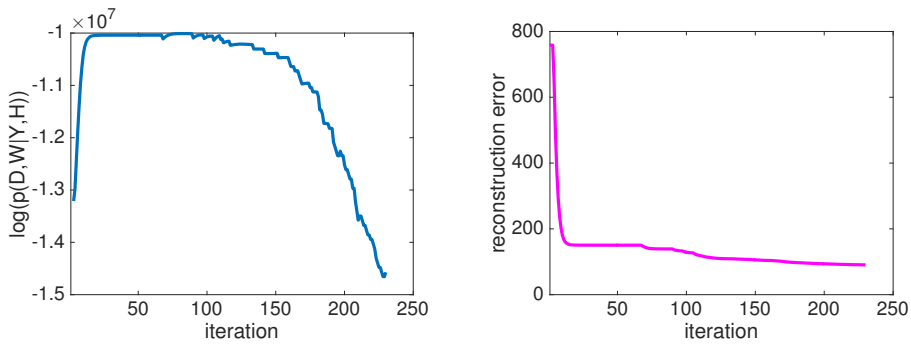
$$\frac{K!}{(H_N+1)^{K+1} \prod_{h=1}^{2^N-1} K_h!} \prod_{k=1}^{K} \frac{(N-m_k)!(m_k-1)!}{N!} \tag{49}$$

where $N_0 = \sum_{i=1}^{N} \|\mathbf{H}_i\|_0$, $H_N = \sum_{j=1}^{N} 1/j$. Then one can define the marginal maximum a posteriori (mMAP) estimator

$$(\mathbf{D}_{mMAP}, \mathbf{W}_{mMAP}) = \underset{\{\mathbf{D}^{(t)}, \mathbf{Z}^{(t)}\}_{t=1}^{T_{MCMC}}}{\operatorname{argmax}} \log p(\mathbf{D}, \mathbf{Z}, \mathbf{S} \mid \mathbf{Y}, \mathbf{H}) \tag{50}$$

Fig. 4(a) shows an example of the evolution of this marginalized posterior during the burn-in period for an inpainting experiment, see Section 6, with the Barbara

image with 50% missing data and $\sigma_{\boldsymbol{\varepsilon}} = 15$. Fig. 4(b) shows the evolution of the mean-square reconstruction error across Gibbs iterations. The reconstruction error nearly always decreases which means that the first step of our Monte Carlo simulation behave like an optimization algorithm first. One must take care of this long burn-in period when the marginalization posterior remarkably nearly always decreases. This behaviour can probably be explained by the evolution of the $1/\|\mathbf{S}\|_{\mathrm{F}}^{NK}$ in the mMAP distribution during first iterations (then K incresases from 0 and $\|\mathbf{S}\|_{\mathrm{F}}$ is expected to increase as well). Recall that an mMAP estimate can be properly defined only in the stationary regime of the Markov chain.



(a) (b)

Figure 4: (a) Logarithm of the marginalized posterior (burn-in); (b) reconstruction error on the result of the Barbara segment with 50% missing data and $\sigma_{\boldsymbol{\varepsilon}} = 15$.

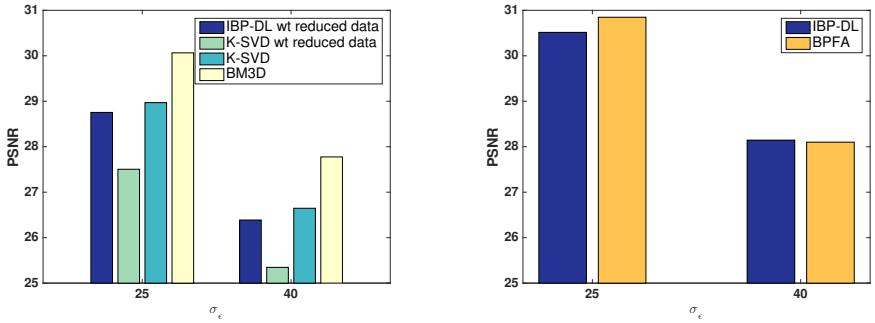
6. Experimental results in image processing.

6.1. Reconstruction of the original image

As a first illustration and consistency check, IBP-DL is trained on a segment of size 256×256 of Barbara image, see Fig. 1. A full data set of 62001 overlapping 8x8 patches is used. Fig. 1 shows the reconstruction of the original image by using the dictionary learnt by IBP-DL from the original image without noise. We use the mMAP estimate of $(\mathbf{D}, \mathbf{W})$ defined in section 5.5. The dictionary contains 150 atoms and the reconstruction is very accurate since one gets

PSNR=44.97 dB. For comparison, K-SVD [3] typically produces a dictionary of size 256 and a larger reconstruction error with PSNR = 43.97 dB. The Bayesian method proposed in [12] with a dictionary of maximal size 256 as well yields PSNR=42.92 dB. We remark that IBP-DL produces a relevant dictionary since it restores the image with an adapted yet smaller number of atoms to reach a better quality of approximation.

6.2. Denoising



(a) (b)

Figure 5: Denoising results of IBP-DL for noise levels $\sigma_{\boldsymbol{\varepsilon}} = 25$ and $\sigma_{\boldsymbol{\varepsilon}} = 40$: (a) average PSNR using IBP-DL learnt from a reduced training set, K-SVD with 256 atoms learnt from the reduced training set, or learnt from the full training set (as IBP-DL) and BM3D; (b) Average PSNR using IBP-DL and BPFA [12] learnt from the same full training set.

The most simple model where $\mathbf{H} = \mathbb{I}_P$ corresponds to the problem of image denoising. The results have already been presented in [14] and showed that IBP-DL denoising performances are similar to those of other state-of-the-art DL approaches. This was a first proof of the relevance of the learnt dictionaries. Fig. 5 (a) & (b) summarize denoising results, see [14] for details, by illustrating the PSNR averaged over 9 images for 2 noise levels $\sigma_{\boldsymbol{\varepsilon}}$=25 and 40 corresponding to PSNR=20.17dB and 16.08dB respectively. Fig. 5(a) compares the denoising performances of IBP-DL learnt from a reduced data set 50%-overlapping patches only with K-SVD based methods [3] and BM3D [24]. Results from BM3D [24]

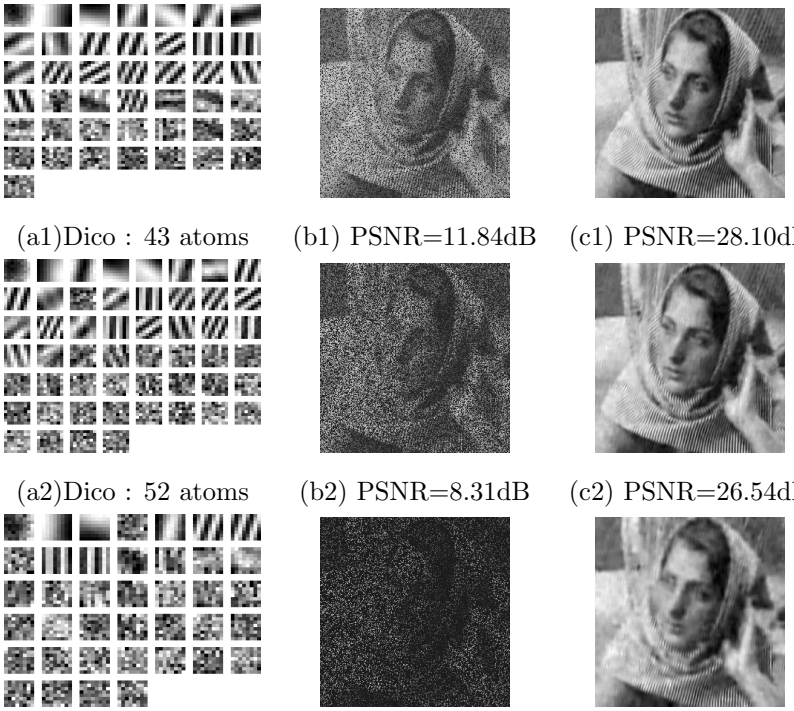
are used as a reference only since we do not expect to obtain better results here. The main observation is that IBP-DL performances are comparable to K-SVD. An important observation is that the performances of K-SVD drop dramatically when a reduced training set is used which may indicate a worse learning efficiency than IBP-DL. Fig. 5(b) compares the denoising performances of IBP-DL and BPFA [12] using the same full data set. They are comparable. One can observe that IBP-DL dictionary sizes strongly depend on the considered image. They are often smaller or only a little larger than 64 while K-SVD and BPFA usually fix the dictionary size at 256 or 512. Moreover, the number of non-zero coefficients is smaller as well when using IBP-DL. For instance, for the image *House* with a noise level of $\sigma = 25$, we found that BPFA led to 116380 non-zero coefficients using a dictionary of K=256 atoms (0.73% sparsity) while IBP-DL yields a dictionary of size K=57 associated to 67028 non zero coefficients (1.9% sparsity). This trend is general: IBP-DL produces smaller dictionaries than the standard 256 or 512 choice, and the number of non-zero coefficients is smaller as well. Despite a smaller dictionary, a very sparse and efficient representation is obtained, which is illustrated by restoration performances. We emphasize that the noise level is accurately estimated as well with an estimation error of at most 10% only. This is an essential benefit of the IBP-DL approach, see [14] for a detailed discussion on IBP-DL for denoising. We now turn to the more difficult inverse problems of image inpainting and compressive sensing.

6.3. Image inpainting

This section presents numerical experiments of image inpainting that is the restoration of missing pixels, e.g. due to some damaged sensor.

Fig. 6 displays several inpainting examples on a segment of *Barbara*. Atoms are ordered by decreasing weight of their coefficients in the reconstruction.

Table 2 gathers several restoration examples on the segment of Barbara image. It presents the size of the dictionary and the PSNR obtained by using IBP-DL for restoration for various proportions of missing pixels (from 0% to 80%) and various noise levels (0, 15, 25) for 8 bits images (gray levels range



(a1)Dico : 43 atoms (b1) PSNR=11.84dB (c1) PSNR=28.10dB

(a2)Dico : 52 atoms (b2) PSNR=8.31dB (c2) PSNR=26.54dB

(a3)Dico : 39 atoms (b3) PSNR=6.37dB (c3) PSNR=23.74dB

Figure 6: IBP-DL restoration of a Barbara segment. From top to bottom are restoration of the noisy ($\sigma_{\boldsymbol{\varepsilon}}$=25) masked image with 20%, 50% and 80% missing pixels. From left to right are IBP-DL dictionary, observed image, restored image.

$\sigma_{\boldsymbol{\varepsilon}}$ \ Missing		80%	50%	20%	0%
0	BPFA	26.87	35.60	40.12	42.94
	IBP-DL	57 - **27.49**	47 - **35.40**	40 - **38.87**	150 - **44.94**
15	BPFA	25.17	29.31	29.93	32.14
	IBP-DL	62 - **25.28**	58 - **28.90**	45 - **30.68**	121 - **31.87**
25	BPFA	23.49	26.79	27.58	29.30
	IBP-DL	39 - **23.74**	52 - **26.54**	43 - **28.10**	67 - **28.90**

Table 2: Restoration results of a Barbara grayscale segment. In each cell (top) PSNR (dB) using BPFA with 256 atoms to restore the image; (bottom) K - PSNR are the IBP-DL dictionary size K and the restoration PSNR (dB).

Missing	Cameraman	House	Peppers	Lena
80%	75 - **24.02**	46 - **31.00**	86 - **26.05**	24 - **30.98**
	22.87 - 24.11	28.38 - 30.12	23.51 - 25.92	28.57 - 31.00
50%	87 - **29.02**	52 - **37.86**	93 - **32.66**	84 - **36.66**
	26.56 - 28.90	33.40 - 38.02	28.36 - 32.58	33.25 - 36.94
20%	75 - **35.14**	56 - **42.37**	90 - **37.58**	44 - **39.20**
	27.56 - 34.70	34.66 - 43.03	30.09 - 37.73	34.37 - 41.27
Missing	Mandrill	Boat	F.print	Hill
80%	63 - **21.93**	29 - **27.86**	44 - **26.52**	98 - **29.33**
	21.24 -21.47	25.95 - 27.81	21.01 - 26.03	27.88 - 29.33
50%	48 - **25.70**	84 - **33.39**	45 - **33.74**	71 - **33.82**
	24.16 - 25.98	30.34 - 33.78	27.56 - 33.53	31.61 - 34.23
20%	65 - **29.48**	62 - **37.54**	86 - **39.88**	68 - **37.34**
	25.36 - 31.67	31.48 - 39.50	29.04 - 40.17	32.67 - 38.75

Table 3: Inpainting using IBP-DL or BPFA applied to gray-scale images: (top of cell) IBP-DL dictionary size K, IBP-DL restoration PSNR (dB) compared to (bottom of cell) KSVD and BPFA restoration PSNR in dB.

from 0 to 255). As a minimal reference, note that using only the constant atom for restoration, that is equivalent to a local averaging filter (or a nearest neighbor interpolation), yields a PSNR of 22 dB: at least, IBP-DL brings a significant improvement with respect to this basic method.

For 80% missing data without noise , BPFA yields a PSNR of 26.87 dB while IBP-DL yields a PSNR of 27.49 dB with 57 atoms; for 50% missing data and $\sigma_{\boldsymbol{\varepsilon}}$=25, $\text{PSNR}_{\text{BPFA}}$=26.79 dB and $\text{PSNR}_{\text{IBP-DL}}$=26.54 dB with K=52. This experiment clearly shows the relevance of IBP-DL that proposes an adapted and efficient dictionary for inpainting, even in the presence of additional noise.

Table 3 compares our results to those of BPFA [12] on a set of images. Depending on the considered image, the performance is in general either in favor of IBP-DL or BPFA for a difference of about ±0.1 dB. Larger differences

are sometimes observed for 20% missing pixels, more often in favor of BPFA. For 80% missing pixels in *House*, IBP-DL and BPFA yield a PSNR of 31.0 dB and 30.12 dB respectively, in favor of IBP-DL this time. Therefore inpainting performances are very similar while the number of atoms inferred by IBP-DL ($39 \leq K \leq 150$) is in general smaller than that of BPFA that is always close to its maximum value of 256 or 512 atoms. As a baseline for comparisons, we also compare our results with those obtained by using K-SVD [3] from the original image to learn a dictionary of size 256. Then an orthogonal matching pursuit algorithm is used to estimate the coefficients for restoration of the damaged image. From our experiments, IBP-DL always performs better than K-SVD.

Fig. 7 shows IBP-DL inpainting results for 3 images with 80%, 50% and 20% missing pixels leading to PSNR of 26.05 dB, 33.82 dB and 35.14 dB respectively. In addition to quantitative PSNR performances, qualitative results are visually fine.

6.4. Compressive sensing

In the compressive sensing experiment, we use the Castle grayscale image (481×321). Each patch $\mathbf{x}_i$ is observed through the same random Gaussian projection matrix $\mathbf{H}$. Then we use standard Gibbs sampling for inference according to model 1. The Castle image has 148836 overlapping 8×8 patches in dimension $P = 64$. The projection matrix $\mathbf{H} \in \mathbb{R}^{Q \times P}$, $Q \leq P$, is random with i.i.d. coefficients $\mathbf{H}(i,j) \sim \mathcal{N}(0,1)$. Fig. 8 displays the restoration of the Castle image with 50% compressive sensing rate, that is for $Q = P/2$. The estimated dictionary is made of 26 atoms only. The relative quadratic error is 0.004 corresponding to an SNR of 23.9 dB and PSNR = 32.9 dB which means a quite good restoration performance. Fig. 9 displays the restoration of the Castle using a random Gaussian i.i.d. dictionary of 26 atoms. The images are restored by averaging pixel estimates from overlapping patches reconstructed by Orthogonal Matching Pursuit (OMP). The restored image has an SNR of 17.37 dB to compare with the 23.9 dB using IBP-DL. The IBP method gives better performance.

Figure 7: Illustration of typical inpainting results obtained by using IBP-DL. From top to bottom are the IBP-DL dictionary, the masked and the inpainted images; (a) Peppers (80% missing), from a PSNR of 6.53 dB to 26.05 dB, (b) Hill (50% missing) from a PSNR of 8.70 dB to 33.82 dB, (c) Cameraman (20% missing) from a PSNR of 12.48 dB to 35.14 dB.

(a) (b) (c)

Figure 8: (a) Initial Castle image; (b) restored image with a relative reconstruction error from compressive sensing at 50% ($Q = P/2$) obtained by IBP-DL: SNR = 23.9 dB, PSNR = 32.9 dB; (c) the estimated dictionary is made of 26 atoms only.

(a) (b)

Figure 9: (a) Initial Castle image; (b) Restored image using random dictionary and OMP: SNR = 17.37 dB (c) the random dictionary of 26 atoms.

7. Conclusion

This article describes IBP-DL, a Bayesian non parametric method for dictionary learning to solve linear inverse problems. The proposed approach uses an Indian Buffet Process prior. It permits to learn a dictionary of adaptive size starting from an empty dictionary, except the trivial constant atom. Therefore a matrix factorization problem is solved in a really non parametric manner since no parameter tuning is needed in contrast with most optimization methods. Moreover we have formulated the dictionary learning problem in the context of linear inverse problems with Gaussian noise. Various MCMC algorithms have been proposed. In particular, we have presented a collapsed Gibbs sampler as well as an accelerated Gibbs sampler to solve the problem of image inpainting (completion of missing data). We have proposed a new method to sample new atoms. We have also derived a marginalized maximum a posteriori estimate for the dictionary. Numerical experiments have shown the relevance of the proposed approach in image processing for inpainting as well as compressive sensing. Future work will explore even more general models (e.g., extensions of the IBP) and other inference methods for scalability since the main practical limitation is the computational cost of Gibbs sampling.

Appendix A. Gibbs sampling

We derive the posterior over z_{ki} for 'active' atoms k, see (13).

$$
\begin{aligned}
&p(z_{ki}|\mathbf{Y},\mathbf{H},\mathbf{D},\mathbf{Z}_{-ki},\mathbf{S},\sigma_{\boldsymbol{\varepsilon}}) \propto \mathcal{N}(\mathbf{y}_i|\mathbf{H}_i\mathbf{D}(\mathbf{z}_i\odot\mathbf{s}_i),\sigma_{\boldsymbol{\varepsilon}}^2)P(z_{ki}|\mathbf{Z}_{-ki})\\
&\propto \exp\left[-\frac{1}{2\sigma_{\boldsymbol{\varepsilon}}^2}((\mathbf{y}_i-\mathbf{H}_i\mathbf{D}(\mathbf{z}_i\odot\mathbf{s}_i))^T(\mathbf{y}_i-\mathbf{H}_i\mathbf{D}(\mathbf{z}_i\odot\mathbf{s}_i))\right]P(z_{ki}|\mathbf{Z}_{-ki})\\
&\propto \exp\left[\frac{-1}{2\sigma_{\boldsymbol{\varepsilon}}^2}((z_{ki}s_{ki})^2\mathbf{d}_k^T\mathbf{H}_i^T\mathbf{H}_i\mathbf{d}_k-2z_{ki}s_{ki}\mathbf{d}_k^T\mathbf{H}_i^T(\mathbf{y}_i-\mathbf{H}_i\sum_{\substack{j=1\\j\neq k}}^{K}\mathbf{d}_jz_{ji}s_{ji})\right]P(z_{ki}|\mathbf{Z}_{-ki})
\end{aligned}
$$

Appendix B. Collapsed Gibbs sampling

We derive the collapsed likelihood $p(\mathbf{Y} \mid \{\mathbf{H}_i\},\mathbf{Z},\mathbf{S},\sigma_{\boldsymbol{\varepsilon}},\sigma_D)$ by integrating $\mathbf{D}$ out in (23). The integration must be carried out with respect to the

rows of $\mathbf{D}$ due to the presence of binary mask. We recall that $\{\mathbf{F}_\ell\}$ is the set of binary diagonal matrices of size N for ℓ=1,...,P. If each $\mathbf{H}_i$ is associated with each data $\mathbf{Y}(:,i)$ then $\mathbf{F}_\ell$ is associated with $\mathbf{y}_\ell$, dimension ℓ of data. $\mathbf{F}_\ell(i,i)$ indicates whether pixel at location ℓ in patch i is observed or not so that $\mathbf{F}_\ell(i,i)$=$\mathbf{H}_i(\ell,\ell)$=$H_{i,\ell}$.

$$\begin{aligned} p(\mathbf{Y} \mid \{\mathbf{H}_i\}, \mathbf{Z}, \mathbf{S}, \sigma_{\boldsymbol{\varepsilon}}, \sigma_D) &= p(\mathbf{Y} \mid \{\mathbf{F}_\ell\}, \mathbf{Z}, \mathbf{S}, \sigma_{\boldsymbol{\varepsilon}}, \sigma_D) \\ &= \int p(\mathbf{Y} \mid \{\mathbf{F}_\ell\}, \mathbf{D}, \mathbf{Z}, \mathbf{S}, \sigma_{\boldsymbol{\varepsilon}}) p(\mathbf{D} \mid \sigma_D) d\mathbf{D} \end{aligned} \tag{B.1}$$

Let $\mathbf{y}_\ell = \mathbf{Y}(\ell,:)$, $\mathbf{c}_\ell = \mathbf{D}(\ell,:)$, we have:

$$\begin{aligned} p(\mathbf{Y} \mid \{\mathbf{F}_\ell\}, \mathbf{D}, \mathbf{W}, \sigma_{\boldsymbol{\varepsilon}}) =& \frac{1}{(2\pi\sigma_{\boldsymbol{\varepsilon}}^2)^{\|\mathbf{Y}\|_0/2}} \exp\left[-\frac{1}{2\sigma_{\boldsymbol{\varepsilon}}^2}\sum_{l=1}^{P}(\mathbf{y}_\ell - \mathbf{c}_\ell\mathbf{W}\mathbf{F}_\ell)(\mathbf{y}_\ell - \mathbf{c}_\ell\mathbf{W}\mathbf{F}_\ell)^T\right] \\ p(\mathbf{D} \mid \sigma_D^2) =& \frac{1}{(2\pi\sigma_D^2)^{KP/2}} \exp\left[-\frac{1}{2\sigma_D^2}\sum_{l=1}^{P}\mathbf{D}(\ell,:)\mathbf{D}(\ell,:)^T\right] \end{aligned}$$

Then, the product in the integral (B.1) becomes

$$\begin{aligned} p(\mathbf{Y} \mid \{\mathbf{F}_\ell\}, \mathbf{D}, \mathbf{W}, \sigma_{\boldsymbol{\varepsilon}}) p(\mathbf{D} \mid \sigma_D^2) =& \frac{1}{(2\pi)^{(\|\mathbf{Y}\|_0+KP)/2}\sigma_{\boldsymbol{\varepsilon}}^{\|\mathbf{Y}\|_0}\sigma_D^{KP}} \\ \exp&\left[-\frac{1}{2}\sum_{l=1}^{P}\left(\frac{1}{\sigma_{\boldsymbol{\varepsilon}}^2}(\mathbf{y}_\ell - \mathbf{c}_\ell\mathbf{W}\mathbf{F}_\ell)(\mathbf{y}_\ell - \mathbf{c}_\ell\mathbf{W}\mathbf{F}_\ell)^T + \frac{1}{\sigma_D^2}\mathbf{c}_\ell\mathbf{c}_\ell^T\right)\right] \end{aligned} \tag{B.2}$$

$$\begin{aligned} &\frac{1}{\sigma_{\boldsymbol{\varepsilon}}^2}(\mathbf{y}_\ell - \mathbf{c}_\ell\mathbf{W}\mathbf{F}_\ell)(\mathbf{y}_\ell - \mathbf{c}_\ell\mathbf{W}\mathbf{F}_\ell)^T + \frac{1}{\sigma_D^2}\mathbf{c}_\ell\mathbf{c}_\ell^T \\ =&\frac{1}{\sigma_{\boldsymbol{\varepsilon}}^2}\mathbf{y}_\ell\mathbf{y}_\ell^T + \frac{1}{\sigma_{\boldsymbol{\varepsilon}}^2}\mathbf{c}_\ell\mathbf{W}\mathbf{F}_\ell\mathbf{F}_\ell^T\mathbf{W}^T\mathbf{c}_\ell^T - \frac{2}{\sigma_{\boldsymbol{\varepsilon}}^2}\mathbf{y}_\ell\mathbf{F}_\ell^T\mathbf{W}^T\mathbf{c}_\ell^T + \frac{1}{\sigma_D^2}\mathbf{c}_\ell\mathbf{c}_\ell^T \\ =&\frac{1}{\sigma_{\boldsymbol{\varepsilon}}^2}\mathbf{y}_\ell\mathbf{y}_\ell^T + \mathbf{c}_\ell(\frac{1}{\sigma_{\boldsymbol{\varepsilon}}^2}\mathbf{W}\mathbf{F}_\ell\mathbf{F}_\ell^T\mathbf{W}^T + \frac{1}{\sigma_D^2}\mathbb{I}_K)\mathbf{c}_\ell^T - \frac{2}{\sigma_{\boldsymbol{\varepsilon}}^2}\mathbf{y}_\ell\mathbf{F}_\ell^T\mathbf{W}^T\mathbf{c}_\ell^T \\ =&\frac{1}{\sigma_{\boldsymbol{\varepsilon}}^2}\mathbf{y}_\ell\mathbf{y}_\ell^T + \mathbf{c}_\ell(\sigma_{\boldsymbol{\varepsilon}}^2\mathbf{M}_\ell)^{-1}\mathbf{c}_\ell^T - \frac{2}{\sigma_{\boldsymbol{\varepsilon}}^2}\mathbf{y}_\ell\mathbf{F}_\ell^T\mathbf{W}^T\mathbf{c}_\ell^T \\ =&(\mathbf{c}_\ell - \mathbf{y}_\ell\mathbf{F}_\ell^T\mathbf{W}^T\mathbf{M}_\ell)(\sigma_{\boldsymbol{\varepsilon}}^2\mathbf{M}_\ell)^{-1}(\mathbf{c}_\ell - \mathbf{y}_\ell\mathbf{F}_\ell^T\mathbf{W}^T\mathbf{M}_\ell)^T + \frac{1}{\sigma_{\boldsymbol{\varepsilon}}^2}\boldsymbol{\Upsilon}_{\boldsymbol{\ell}} \end{aligned}$$

where

$$\mathbf{M}_\ell = (\mathbf{W}\mathbf{F}_\ell\mathbf{F}_\ell^T\mathbf{W}^T + \frac{\sigma_{\boldsymbol{\varepsilon}}^2}{\sigma_D^2}\mathbb{I}_K)^{-1}, \tag{B.3}$$

$$\boldsymbol{\Upsilon}_{\boldsymbol{\ell}} = \mathbf{y}_\ell(\mathbb{I} - \mathbf{F}_\ell^T\mathbf{W}^T\mathbf{M}_\ell\mathbf{W}\mathbf{F}_\ell)\mathbf{y}_\ell^T. \tag{B.4}$$

It can be shown that $\mathbf{c}_\ell = \mathbf{D}(\ell, :)$ can be drawn from a Gaussian distribution

$$\begin{aligned} p(\mathbf{c}_\ell \mid \mathbf{y}_\ell, \mathbf{F}_\ell, \mathbf{Z}, \mathbf{S}, \sigma_{\boldsymbol{\varepsilon}}, \sigma_D) &\propto \mathcal{N}(\boldsymbol{\mu}_{\mathbf{D}_\ell}, \boldsymbol{\Sigma}_{\mathbf{D}_\ell}) \\ \boldsymbol{\Sigma}_{\mathbf{D}_\ell} &= \sigma_{\boldsymbol{\varepsilon}}^2 \mathbf{M}_\ell \\ \boldsymbol{\mu}_{\mathbf{D}_\ell} &= \mathbf{y}_\ell \mathbf{F}_\ell^T \mathbf{W}^T \mathbf{M}_\ell \end{aligned} \tag{B.5}$$

and

$$p(\mathbf{D} \mid \mathbf{Y}, \mathbf{F}, \mathbf{Z}, \mathbf{S}, \sigma_{\boldsymbol{\varepsilon}}, \sigma_D) \propto \prod_{\ell=1}^{P} p(\mathbf{c}_\ell \mid \mathbf{y}_\ell, \mathbf{F}_\ell, \mathbf{Z}, \mathbf{S}, \sigma_{\boldsymbol{\varepsilon}}, \sigma_D) \tag{B.6}$$

Therefore, the integral in (B.1) yields (23)

$$\begin{aligned} p(\mathbf{Y} \mid \{\mathbf{F}_\ell\}, \mathbf{Z}, \mathbf{S}, \sigma_{\boldsymbol{\varepsilon}}^2, \sigma_D^2) &= \frac{1}{(2\pi)^{(\|\mathbf{Y}\|_0 + KP)/2} \sigma_{\boldsymbol{\varepsilon}}^{\|\mathbf{Y}\|_0} \sigma_D^{KP}} \exp\left[-\frac{1}{2}\sum_{l=1}^{P} \frac{1}{\sigma_{\boldsymbol{\varepsilon}}^2} \boldsymbol{\Upsilon}_{\boldsymbol{\ell}}\right] \\ &\times \int \exp\left[-\frac{1}{2}\sum_{l=1}^{P} \left((\mathbf{c}_\ell - \boldsymbol{\mu}_{\mathbf{D}_\ell}) \boldsymbol{\Sigma}_{\mathbf{D}_\ell}^{-1} (\mathbf{c}_\ell - \boldsymbol{\mu}_{\mathbf{D}_\ell})^T d\mathbf{D}\right)\right] \\ &= \frac{\prod_{\ell=1}^{P} (2\pi)^{K/2} |\boldsymbol{\Sigma}_{\mathbf{D}\ell}|^{1/2}}{(2\pi)^{(\|\mathbf{Y}\|_0 + KP)/2} \sigma_{\boldsymbol{\varepsilon}}^{\|\mathbf{Y}\|_0} \sigma_D^{KP}} \prod_{\ell=1}^{P} \exp\left[-\frac{1}{2\sigma_{\boldsymbol{\varepsilon}}^2} \boldsymbol{\Upsilon}_{\boldsymbol{\ell}}\right] \\ &= \frac{1}{(2\pi)^{\|\mathbf{Y}\|_0/2} \sigma_{\boldsymbol{\varepsilon}}^{\|\mathbf{Y}\|_0 - KP} \sigma_D^{KP}} \prod_{\ell=1}^{P} |\mathbf{M}_\ell|^{1/2} \exp\left[-\frac{1}{2\sigma_{\boldsymbol{\varepsilon}}^2} \boldsymbol{\Upsilon}_{\boldsymbol{\ell}}\right] \end{aligned} \tag{B.7}$$

Appendix C. Accelerated gibbs sampling

Here is the derivation of (38) to (40).

$$p(\mathbf{Z} \mid \mathbf{Y}, \mathbf{S}, \mathbf{H}, \sigma_{\boldsymbol{\varepsilon}}^2, \sigma_D^2, \alpha) \propto p(\mathbf{Z} \mid \alpha) \int p(\mathbf{Y} \mid \mathbf{H}, \mathbf{D}, \mathbf{Z}, \mathbf{S}, \sigma_{\boldsymbol{\varepsilon}}) p(\mathbf{D} \mid \sigma_D) d\mathbf{D} \tag{C.1}$$

The data is split into 2 sets according to $\mathbf{Y} = [\mathbf{y}_i, \mathbf{Y}_{-i}]$, $\mathbf{W} = [\mathbf{w}_i, \mathbf{W}_{-i}]$ and $\mathbf{H} = \{\mathbf{H}_i, \{\mathbf{H}_{\neq i}\}\}$.

$$\begin{aligned} p(\mathbf{Y} \mid \{\mathbf{H}_i\}, \mathbf{Z}, \mathbf{S}, \sigma_{\boldsymbol{\varepsilon}}, \sigma_D) &= \int p(\mathbf{Y} \mid \{\mathbf{H}_i\}, \mathbf{D}, \mathbf{Z}, \mathbf{S}, \sigma_{\boldsymbol{\varepsilon}}) p(\mathbf{D} \mid \sigma_D) d\mathbf{D} \\ &= \int p(\mathbf{y}_i, \mathbf{Y}_{-i} \mid \mathbf{H}_i, \{\mathbf{H}_{\neq i}\}, \mathbf{D}, \mathbf{z}_i, \mathbf{Z}_{-i}, \mathbf{s}_i, \mathbf{S}_{-i}, \sigma_{\boldsymbol{\varepsilon}}^2) p(\mathbf{D} \mid \sigma_D) d\mathbf{D} \\ &= \int p(\mathbf{y}_i \mid \mathbf{H}_i, \mathbf{D}, \mathbf{z}_i, \mathbf{s}_i, \sigma_{\boldsymbol{\varepsilon}}) p(\mathbf{Y}_{-i} \mid \{\mathbf{H}_{\neq i}\}, \mathbf{Z}_{-i}, \mathbf{S}_{-i}, \mathbf{D}, \sigma_{\boldsymbol{\varepsilon}}^2) p(\mathbf{D} \mid \sigma_D) d\mathbf{D} \end{aligned} \tag{C.2}$$

The likelihood $p(\mathbf{Y}_{-i} \mid \{\mathbf{H}_{\neq i}\}, \mathbf{Z}_{-i}, \mathbf{S}_{-i}, \mathbf{D}, \sigma_{\boldsymbol{\varepsilon}}^2)$ and the prior $p(\mathbf{D} \mid \sigma_D)$ are both Gaussian. We apply the Bayes rule. The posterior $p(\mathbf{D} \mid \mathbf{Y}_{-i}, \{\mathbf{H}_{\neq i}\}, \mathbf{Z}_{-i}, \mathbf{S}_{-i}, \sigma_{\boldsymbol{\varepsilon}}, \sigma_D)$ is also Gaussian :

$$\begin{aligned}
&p(\mathbf{Y} \mid \{\mathbf{F}_\ell\}, \mathbf{Z}, \mathbf{S}, \sigma_{\boldsymbol{\varepsilon}}^2, \sigma_D^2) \\
&\propto \int p(\mathbf{y}_i \mid \mathbf{H}_i, \mathbf{D}, \mathbf{z}_i, \mathbf{s}_i, \sigma_{\boldsymbol{\varepsilon}}) p(\mathbf{D} \mid \mathbf{Y}_{-i}, \{\mathbf{H}_{\neq i}\}, \mathbf{Z}_{-i}, \mathbf{S}_{-i}, \sigma_{\boldsymbol{\varepsilon}}, \sigma_D) d\mathbf{D} \\
&\propto \int p(\mathbf{y}_i \mid \mathbf{H}_i, \mathbf{D}, \mathbf{z}_i, \mathbf{s}_i, \sigma_{\boldsymbol{\varepsilon}}) \prod_{\ell=1}^{P} \mathcal{N}(\mathbf{c}_\ell; \boldsymbol{\mu}_{\mathbf{D}_{\ell,-i}}, \boldsymbol{\Sigma}_{\mathbf{D}_{\ell,-i}}) d\mathbf{D} \qquad \text{(C.3)}
\end{aligned}$$

$p(\mathbf{y}_i \mid \mathbf{H}_i, \mathbf{D}, \mathbf{z}_i, \mathbf{s}_i, \sigma_{\boldsymbol{\varepsilon}})$ and $p(\mathbf{D} \mid \mathbf{Y}_{-i}, \{\mathbf{H}_{\neq i}\}, \mathbf{Z}_{-i}, \mathbf{S}_{-i}, \sigma_{\boldsymbol{\varepsilon}}, \sigma_D)$ are both Gaussian so that the integral in equation (C.3) yields

$$\begin{aligned}
p(\mathbf{Y} \mid \{\mathbf{F}_\ell\}, \mathbf{Z}, \mathbf{S}, \sigma_{\boldsymbol{\varepsilon}}^2, \sigma_D^2) &\propto p(\mathbf{y}_i \mid \mathbf{H}_i, \mathbf{D}, \mathbf{z}_i, \mathbf{s}_i, \sigma_{\boldsymbol{\varepsilon}}, \boldsymbol{\mu}_{\mathbf{D}_{\ell,-i}}, \boldsymbol{\Sigma}_{\mathbf{D}_{\ell,-i}}) \\
&\propto \prod_{\ell=1}^{P} \mathcal{N}(\mathbf{y}_i(\ell); \mu_{yi\ell}, \sigma_{yi\ell}) \qquad \text{(C.4)}
\end{aligned}$$

$$\begin{aligned}
\text{where} \quad & \mu_{yi\ell} = H_{i,\ell} \boldsymbol{\mu}_{\mathbf{D}_{\ell,-i}} \mathbf{w}_i \\
& \sigma_{yi\ell} = H_{i,\ell} \mathbf{w}_i^{\mathrm{T}} \boldsymbol{\Sigma}_{\mathbf{D}_{\ell,-i}} \mathbf{w}_i + \sigma_{\boldsymbol{\varepsilon}}^2 \qquad \text{(C.5)}
\end{aligned}$$

Appendix D. Matrix inversion lemma

In section 5.1.3 we need to compute the inverse of $g_{\mathbf{D}\ell}$ and remove or restore the influence of each data i.

1. To remove the influence of data i one needs $\boldsymbol{\Sigma}_{\mathbf{D}_{\ell,-i}} = g_{\mathbf{D}\ell,-i}^{-1}$, see (36),

$$\begin{aligned}
g_{\mathbf{D}\ell,-i}^{-1} &= \left(g_{\mathbf{D}\ell} - \sigma_{\boldsymbol{\varepsilon}}^{-2} H_{i,\ell} \mathbf{w}_i \mathbf{w}_i^{\mathrm{T}}\right)^{-1} \\
&= g_{\mathbf{D}\ell}^{-1} - \frac{H_{i,\ell}}{H_{i,\ell} \mathbf{w}_i^T g_{\mathbf{D}\ell}^{-1} \mathbf{w}_i - \sigma_{\boldsymbol{\varepsilon}}^2} g_{\mathbf{D}\ell}^{-1} \mathbf{w}_i \mathbf{w}_i^{\mathrm{T}} g_{\mathbf{D}\ell}^{-1} \qquad \text{(D.1)}
\end{aligned}$$

2. Restore the influence of data i to recover $\boldsymbol{\Sigma}_{\mathbf{D}_\ell} = g_{\mathbf{D}\ell}^{-1}$ from $\boldsymbol{\Sigma}_{\mathbf{D}_{\ell,-i}}$

$$\begin{aligned}
g_{\mathbf{D}\ell}^{-1} &= \left(g_{\mathbf{D}\ell,-i} + \sigma_{\boldsymbol{\varepsilon}}^{-2} \mathbf{w}_i H_{i,\ell} \mathbf{w}_i^{\mathrm{T}}\right)^{-1} \\
&= g_{\mathbf{D}\ell,-i}^{-1} - \frac{H_{i,\ell}}{H_{i,\ell} \mathbf{w}_i^T g_{\mathbf{D}\ell,-i}^{-1} \mathbf{w}_i + \sigma_{\boldsymbol{\varepsilon}}^2} g_{\mathbf{D}\ell,-i}^{-1} \mathbf{w}_i \mathbf{w}_i^{\mathrm{T}} g_{\mathbf{D}\ell,-i}^{-1} \qquad \text{(D.2)}
\end{aligned}$$

Appendix E. Marginalized MAP

Let $\boldsymbol{\theta} = (\sigma_{\boldsymbol{\varepsilon}}, \sigma_S, \alpha)$. We compute:

$$p(\mathbf{D}, \mathbf{Z}, \mathbf{S} \mid \mathbf{Y}, \mathbf{H}) = \int p(\mathbf{D}, \mathbf{Z}, \mathbf{S} \mid \mathbf{Y}, \mathbf{H}, \boldsymbol{\theta}) p(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta}. \tag{E.1}$$

We have

$$p(\mathbf{D}, \mathbf{Z}, \mathbf{S} \mid \mathbf{Y}, \boldsymbol{\theta}) \propto p(\mathbf{Y} \mid \mathbf{H}, \mathbf{D}, \mathbf{Z}, \mathbf{S}, \boldsymbol{\theta}) p(\mathbf{D}) p(\mathbf{Z}) p(\mathbf{S}) \tag{E.2}$$

$$p(\boldsymbol{\theta}) = p(\alpha) p(\frac{1}{\sigma_{\boldsymbol{\varepsilon}}^2}) p(\frac{1}{\sigma_S^2}) \tag{E.3}$$

also

$$p(\mathbf{Y} \mid \mathbf{D}, \mathbf{Z}, \mathbf{S}, \sigma_{\boldsymbol{\varepsilon}}) = \frac{1}{(2\pi\sigma_{\boldsymbol{\varepsilon}}^2)^{N_0/2}} \exp\left(-\frac{1}{2\sigma_{\boldsymbol{\varepsilon}}^2} \sum_{i=1}^{N} \|\mathbf{y}_i - \mathbf{H}_i \mathbf{D} \mathbf{w}_i\|_F^2\right) \tag{E.4}$$

$$p(\mathbf{D} \mid \sigma_D) = \prod_{k=1}^{K} \frac{1}{(2\pi\sigma_D^2)^{P/2}} \exp\left(-\frac{1}{2\sigma_D^2} \|\mathbf{d}_k\|_2^2\right) \tag{E.5}$$

$$p(\mathbf{Z} \mid \alpha) = \frac{\alpha^K}{\prod_{h=1}^{2^N-1} K_h!} \exp\left(-\alpha H_N\right) \prod_{k=1}^{K} \frac{(N - m_k)!(m_k - 1)!}{N!} \tag{E.6}$$

$$p(\mathbf{S} \mid \sigma_S) = \prod_{i=1}^{N} \prod_{k=1}^{K} \frac{1}{(2\pi\sigma_S^2)^{1/2}} \exp\left(-\frac{s_{ki}^2}{2\sigma_S^2}\right) \tag{E.7}$$

$$p(\alpha) = \mathcal{G}(1, 1) \tag{E.8}$$

$$p(1/\sigma_{\boldsymbol{\varepsilon}}^2) = \mathcal{G}(c_0, d_0) \tag{E.9}$$

$$p(1/\sigma_S^2) = \mathcal{G}(e_0, f_0) \tag{E.10}$$

where $\mathcal{G}(x; a, c) = x^{a-1} b^a \exp(-bx)/\Gamma(a)$; $H_N = \sum_{j=1}^{N} \frac{1}{j}$; $N_0 = \sum_{i=1}^{N} \|\mathbf{H}_i\|_0$

Marginalize out α. We have $\alpha \sim \mathcal{G}(1, 1) = \exp(-\alpha)$.

$$\int_0^\infty \alpha^K \exp(-\alpha H_N) \exp(-\alpha) \mathrm{d}\alpha = \int_0^\infty \alpha^K \exp(-\alpha(H_N + 1)) \mathrm{d}\alpha \tag{E.11}$$

Since $\Gamma(x) = \int_0^\infty t^{x-1} \exp(-t) \mathrm{d}t$, we have

$$\int_0^\infty \alpha^K \exp(-\alpha(H_N + 1)) \mathrm{d}\alpha = \frac{\Gamma(K+1)}{(H_N + 1)^{K+1}} = \frac{K!}{(H_N + 1)^{K+1}} \text{ for } K \in \mathbb{N} \tag{E.12}$$

so that

$$\int p(\mathbf{Z}|\alpha)p(\alpha)d\alpha$$
$$= \frac{K!}{(H_N+1)^{K+1}} \frac{1}{\prod_{h=1}^{2^N-1} K_h!} \prod_{k=1}^{K} \frac{(N-m_k)!(m_k-1)!}{N!} \tag{E.13}$$

Marginalize out $\sigma_{\boldsymbol{\varepsilon}}$. Let $N_0 = \sum_{i=1}^{N} \|\mathbf{H}_i\|_0$.

$$\int_0^\infty p(\mathbf{Y} \mid \mathbf{H}, \mathbf{D}, \mathbf{Z}, \mathbf{S}, \sigma_{\boldsymbol{\varepsilon}}) p\left(\frac{1}{\sigma_{\boldsymbol{\varepsilon}}^2}\right) d\left(\frac{1}{\sigma_{\boldsymbol{\varepsilon}}^2}\right)$$
$$\propto \int \exp\left[-\frac{1}{\sigma_{\boldsymbol{\varepsilon}}^2}\left(d_0 + \frac{\sum_{i=1}^{N} \|\mathbf{y}_i - \mathbf{H}_i\mathbf{D}\mathbf{w}_i\|_F^2}{2}\right)\right] \left(\frac{1}{\sigma_{\boldsymbol{\varepsilon}}^2}\right)^{N_0/2+c_0-1} d\left(\frac{1}{\sigma_{\boldsymbol{\varepsilon}}^2}\right) \tag{E.14}$$

In practice, very small hyperparameters (c_0=d_0=10^{-6}) are used for $\sigma_{\boldsymbol{\varepsilon}}^2$.

$$\int_0^\infty p(\mathbf{Y} \mid \mathbf{H}, \mathbf{D}, \mathbf{Z}, \mathbf{S}, \sigma_{\boldsymbol{\varepsilon}}) p\left(\frac{1}{\sigma_{\boldsymbol{\varepsilon}}^2}\right) d\frac{1}{\sigma_{\boldsymbol{\varepsilon}}^2}$$
$$\propto \int \exp\left[-\frac{1}{\sigma_{\boldsymbol{\varepsilon}}^2}\left(\frac{\sum_{i=1}^{N} \|\mathbf{y}_i - \mathbf{H}_i\mathbf{D}\mathbf{w}_i\|_F^2}{2}\right)\right] \left(\frac{1}{\sigma_{\boldsymbol{\varepsilon}}^2}\right)^{N_0/2-1} d\frac{1}{\sigma_{\boldsymbol{\varepsilon}}^2}$$
$$\propto \frac{1}{\left(\frac{\sum_{i=1}^{N} \|\mathbf{y}_i - \mathbf{H}_i\mathbf{D}\mathbf{w}_i\|_F^2}{2}\right)^{N_0/2}} \Gamma(\frac{N_0}{2})$$
$$\propto \left(\sum_{i=1}^{N} \|\mathbf{y}_i - \mathbf{H}_i\mathbf{D}\mathbf{w}_i\|_F^2\right)^{-N_0/2} \tag{E.15}$$

Marginalize out σ_S*:.* Very small hyperparameters (e_0=f_0=10^{-6}) are also used for σ_S^2.

$$\int_0^\infty p(\mathbf{S} \mid \sigma_S) p(1/\sigma_S^2) d\frac{1}{\sigma_S^2}$$
$$\propto \int \frac{1}{(2\pi)^{NK/2}} \exp\left[-\frac{1}{\sigma_S^2}\left(f_0 + \frac{\|\mathbf{S}\|_F^2}{2}\right)\right] \left(\frac{1}{\sigma_S^2}\right)^{NK/2+e_0-1} d\frac{1}{\sigma_S^2}$$
$$\propto \frac{1}{(\pi)^{NK/2}} \frac{1}{(\|\mathbf{S}\|_F)^{NK}} \Gamma(\frac{NK}{2}) \tag{E.16}$$

Then, from (E.13),(E.15),(E.16) and (E.5), we obtain:

$$p(\mathbf{D},\mathbf{Z},\mathbf{S} \mid \mathbf{Y}) \propto \frac{K!}{(H_N+1)^{K+1}} \frac{1}{\prod_{h=1}^{2^N-1} K_h!} \prod_{k=1}^{K} \frac{(N-m_k)!(m_k-1)!}{N!}$$
$$\frac{1}{(2\pi\sigma_D^2)^{PK/2}} \exp\left(-\frac{\|\mathbf{D}\|_{\mathrm{F}}^2}{2\sigma_D^2}\right) \left(\sum_{i=1}^{N} \|\mathbf{y}_i - \mathbf{H}_i\mathbf{D}\mathbf{w}_i\|_{\mathrm{F}}^2\right)^{-N_0/2} \frac{1}{(\pi)^{NK/2}} \frac{1}{(\|\mathbf{S}\|_F)^{NK}} \Gamma(\frac{NK}{2}) \tag{E.17}$$

where $N_0 = \sum_{i=1}^{N} \|\mathbf{H}_i\|_0$, $H_N = \sum_{j=1}^{N} 1/j$.

References

[1] I. Tosic, P. Frossard, Dictionary learning : What is the right representation for my signal, IEEE Signal Process. Mag. 28 (2011) 27–38.

[2] B. Olshausen, D. Field, Emergence of simple-cell receptive properties by learning a sparse code for natural images, Nature 381 (1996) 607–609.

[3] M. Aharon, M. Elad, A. Bruckstein, K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation, IEEE Trans. Signal Process. 54 (2006) 4311–4322.

[4] M. Elad, M. Aharon, Image denoising via sparse and redundant representations over learned dictionaries, IEEE Trans. Image Process. 15 (2006) 3736–3745.

[5] J. Mairal, M. Elad, G. Sapiro, Sparse representation for color image restoration, IEEE Trans. Image Process. 17 (2008) 53–69.

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

[7] N. Rao, F. Porikli, A clustering approach to optimize online dictionary learning, in: ICASSP, 2012, pp. 1293–1296.

[8] R. Mazhar, P. Gader, EK-SVD: Optimized dictionary design for sparse representations: Optimized dictionary design for sparse representations, in: Proc. of ICPR, 2008, pp. 1–4.

[9] J. Feng, L. Song, X. Yang, W. Zhang, Sub clustering k-svd: Size variable dictionary learning for sparse representations, in: ICIP, 2009, pp. 2149–2152.

[10] C. Rusu, B. Dumitrescu, Stagewise k-svd to design efficient dictionaries for sparse representations, IEEE Signal Process. Lett. 19 (2012) 631–634.

[11] M. Marsousi, K. Abhari, P. Babyn, J. Alirezaie, An adaptive approach to learn overcomplete dictionaries with efficient numbers of elements, IEEE Trans. Signal Process. 62 (12) (2014) 3272–3283.

[12] M. Zhou *et al*, Nonparametric bayesian dictionary learning for analysis of noisy and incomplete images, IEEE Trans. Image Process. 21 (2012) 130–144.

[13] T. Griffiths, Z. Ghahramani, The Indian Buffet Process: An introduction and review, J. Mach. Learn. Res. 12 (2011) 1185–1224.

[14] H.-P. Dang, P. Chainais, A Bayesian non parametric approach to learn dictionaries with adapted numbers of atoms, in: IEEE Int. Workshop on MLSP, 2015, pp. 1–6.

[15] F. Doshi-Velez, Z. Ghahramani, Accelerated sampling for the indian buffet process, in: ICML, 2009, pp. 273–280.

[16] Y. W. Teh, Dirichlet processes, in: Encyclopedia of Machine Learning, Springer, 2010.

[17] T. Griffiths, Z. Ghahramani, Infinite latent feature models and the indian buffet process, in: Advances in NIPS 18, MIT Press, 2006, pp. 475–482.

[18] R. Thibaux, M. I. Jordan, Hierarchical beta processes and the indian buffet process., in: Int. Workshop on AISTATS, Vol. 11, 2007, pp. 564–571.

[19] N. L. Hjort, Nonparametric bayes estimators based on beta processes in models for life history data, Annals of Statistics 18 (3) (1990) 1259–1294.

[20] D. Knowles, Z. Ghahramani, Nonparametric Bayesian sparse factor models with application to gene expression modeling, The Annals of Applied Statistics 5 (2011) 1534–1552.

[21] D. A. van Dyk, T. Park, Partially collapsed Gibbs samplers, Journal of the American Statistical Association 103 (2008) 790–796.

[22] T. Griffiths, Z. Ghahramani, Infinite latent feature models and the indian buffet process, Tech. rep., University College London, Gatsby Computational Neuroscience Unit (2005).

[23] D. Andrzejewski, Accelerated gibbs sampling for infinite sparse factor analysis, Tech. rep., Lawrence Livermore National Laboratory (LLNL), Livermore, CA (2011).

[24] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image denoising by sparse 3-d transform-domain collaborative filtering, IEEE Trans. Image Process. 16 (2007) 2080–2095.