Research project as part of an internship carried out at IMT Atlantique (Lab-STIIC)

Designing transformations for spatialized prior distributions on the simplex in a bayesian context : application to spectral unmixing

---

Author:
M. Hector BLONDEL

Supervisors:
Dr. Lucas DRUMETZ
Pr. Thierry CHONAVEL

Version of April 8, 2026

Contents

Abstract

Abstract : This work, as part of a research internship carried out at IMT Atlantique, Lab-STICC, addresses hyperspectral unmixing problems. It uses a Bayesian framework that models spatial dependence between pixel abundances using Gaussian processes, which are particularly promising for uncertainty quantification. Though, as proportions lie on the simplex, these processes couldn't be defined in this constrained domain, but should be set on a latent space obtained via a bijective transformation. Our main contribution is the development and analysis of different transformations to adapt Langevin Monte Carlo algorithms for simplex valued processes, as well as a method for building confidence intervals on estimates. The new mathematical framework developed here allows efficient posterior sampling and uncertainty quantification, offering a principled Bayesian alternative for constrained spatial modeling in hyperspectral unmixing.

Résumé : Ce travail, réalisé dans le cadre d'un stage de recherche à l'IMT Atlantique, Lab-STICC, aborde les problèmes de démélange spectral dans un cadre bayésien modélisant la dépendance spatiale entre les abondances de pixels à l'aide de processus gaussiens. Néanmoins puisque ces abondances appartiennent au simplexe, ces processus gaussiens ne peuvent être définis directement dans ce domaine contraint, mais devraient exister dans un nouvel espace, obtenu par une transformation bijective. Notre principale contribution réside dans la formalisation des propriétés de différentes transformations adaptées à l'échantillonnage de type Langevin Monte Carlo sur le simplexe, et le développement d'une méthode de construction de régions de confiance. Le nouveau cadre mathématique développé ici permet un échantillonnage efficace de la loi a posteriori ainsi qu'une quantification naturelle de l'incertitude des estimations d'abondance, offrant ainsi une alternative bayésienne rigoureuse pour la modélisation spatiale contrainte dans l'imagerie hyperspectrale.

Remerciements / Acknowledgments

Je souhaiterais remercier chaleureusement mes encadrants Lucas Drumetz et Thierry Chonavel pour avoir proposé et encadré ce stage. Ils m'ont offert une grande liberté d'exploration, tout en veillant à ce que je garde le cap sur le problème principal lorsque cela s'imposait. Leur maîtrise du domaine d'application et des outils mathématiques, associée à leur rigueur, m'a offert les meilleures conditions pour apprendre et progresser durant ce stage.

Je souhaite également remercier l'IMT Atlantique, ses chercheurs, doctorants et stagiaires pour leur accueil au sein du laboratoire.

1 Introduction

1.1 Remote sensing and spectral unmixing

Remote sensing is the broad field of earth observation using, for example, optical (photographic) or radar imagery, acquired from airborne or satellite platforms.

More specifficaly, here we are interested in hyperspectral imaging (see [3] for more context) : one collects data, which has the form of an image, with a large amounts of wavelengths (say $L \geq 10$), mostly on the visible spectrum. In a scene, P different objects (called endmembers) with different reflectance, reflect and scatter light, of which we observe the different wavelength contributions (see 1). Some problems would be, among many others, interpolation (guessing missing abundance information within identified areas), noise reduction or unmixing (guessing abundances). We will focus on the later. A broad overview on this topic is given in [4].

We tackle our model as a regression. This constitutes a case of an inverse problem : search for parameters knowing the effects, with a specific model given. In the context of hyperspectral imaging, $L \gg P$ and the problem might be sufficiently determined.

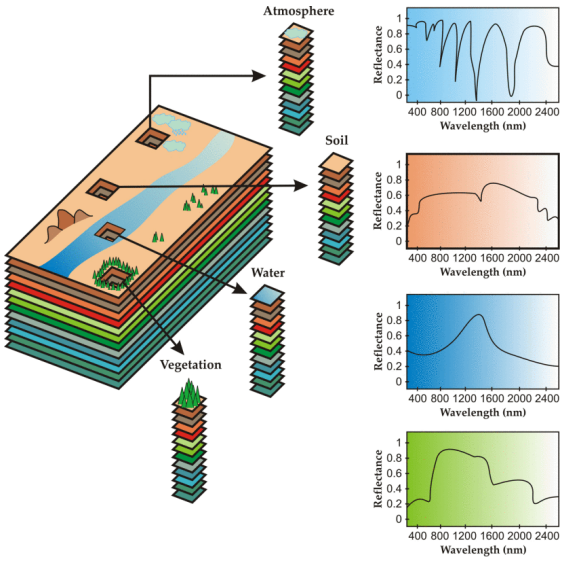


Figure 1: Hyperspectral imaging concept ([4])

1.2 Problem statement

More specifically, our research will focus on deducing abundance (the proportion of presence in each pixel) $A = [\mathbf{a}_1, \ldots, \mathbf{a}_N] \in \mathbb{R}^{P \times N}$ from observations $X = [\mathbf{x}_1 \ldots \mathbf{x}_N] \in \mathbb{R}^{L \times N}$ from the following linear model, assuming known endmember spectral signatures $S = [s_1 \ldots s_P] \in \mathbb{R}^{L \times P}$ (taking into account the reflectances for each element) :

$$\mathbf{x}_n = S\mathbf{a}_n + \mathbf{w}_n$$

with $n \in [\![1, N]\!]$ the pixels.

We guess here that S is knowned and fixed. It might have been guessed using specific methods.

Here, the (simplified) linear model exploits the physics properties of scatering and reflection. We guess we have eliminiated the dependance of the spectral observation to incident angle, accounting for instance on S not depending on n. In reality, a more complex model should be considered, for instance taking into account two reflections light paths.

In our case, we assume we know which (small) set of endmembers to search the proportions from. This means that formulating sparsity conditions is neither

necessary nor relevant here.

We want to solve the spectral unmixing problem, exploiting spatial dependence between abundances $\mathbf{a}_n$, which might prove useful especially for compensating noisy or altered observation. We also want to be able to quantify estimation uncertainty, so that the person using the data can understand how difficult the problem of unmixing is, and to what degree can the he rely on the validity of the output. This is hard in general when estimates might not have a clear interpretation (e.g. for outputs from optimization methods).

The complexity of this problem comes from the fact that abundances $\mathbf{a}_n$ lie in a constrained space, the simplex : $\forall n, \sum_k [A]_{n,k} = 1$.

For this purpose, we chose to use a bayesian approach, whose probabilistic nature allows us better exploit confidence.

Our proposed model here is to define a prior distribution as a gaussian process on the proportions, a very useful tool for spatial regression/interpolation, with a "natural" form allowing for variance and incertitude quantification (see [22]). However, since the abundances lie on a simplex—a constrained domain—this is not directly feasible. To address this, one can apply a bijective transformation that maps the simplex to an unconstrained space, in which the process is Gaussian.

To evaluate for our approach, here are the following problematics we want to explore :

- Gaussian process are really convenient to study, but they cannot be directly applied to the simplex, which is a constrained domain ! **How to define a bijective transformation ψ from the simplex to an euclidian space ? This would allow us to define a prior on A such that the transformed distribution $p(\psi(A))$ is a gaussian proces.**
- **Using this prior distribution, how to efficiently sample the posterior distribution, exploiting our transformation ?**
- **How would this framework allow us to evaluate the uncertainty of our estimations ?**

1.3 Research environment

The current internship is related to project GENESIS (GENerative modeling with kErnelS for Inverse problemS), belonging to the SequoIA cluster, focused on the

application of AI/ML technologies for ocean. This internship is carried out at IMT Atlantique.

The broad objective of the aforementioned project chair is to develop methods for remote sensing combining modern machine learning methods with gaussian processes, to exploit the latter's potential in uncertainty quantification.

Notations

- log is the natural logarithm : $\log(e) = 1$.

Let X be a random variable on $\mathbb{R}^N$.

- $p_X(\mathbf{x})$ is the density of random variable X in $\mathbf{x} \in \mathbb{R}^P$
- $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ is the real multivariate gaussian random variable with mean $\boldsymbol{\mu}$ and covariance matrix $\Sigma = \mathbb{V}(XX^T) : p(\mathbf{x}) = \frac{1}{(2\pi)^{N/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$

Matrices :

- $\langle .;. \rangle$ is the **euclidean** scalar product on $\mathbb{R}^P$
- $\|.\|$ is euclidean norm on vectors and $\|.\|_F$ the Frobenius norm on matrices
- $vec(A)$ the vectorized version of matrix $A \in \mathbb{R}^{m\times n} : [vec(A)]_{jm+i} = [A]_{i,j}$
- $A \otimes B$ kronecker product between $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times q}$: $[A \otimes B]_{im+k,jn+l} = A_{i,j}B_{k,l}$
- $\mathbf{1} := [1 \dots 1]^T$ with P ones.
- $G_P := I_P - \frac{1}{P}\mathbf{1}\mathbf{1}^T$ the projection onto the space $\mathbf{1}^\perp = \{\mathbf{x} \in \mathbb{R}^P | \mathbf{1}^T\mathbf{x} = \mathbf{0}\}$

Simplex :

- $\Delta^{P-1} := \{\mathbf{a} \in (\mathbb{R}_+^*)^P | \sum_{k=1}^P a_k = 1\}$ the dimension P simplex.
- $\check{\Delta}^{P-1} := \{\check{\mathbf{a}} \in (\mathbb{R}_+^*)^{P-1} | \sum_{k=1}^{P-1} a_k < 1\} = (\mathbb{R}_+^*)^{P-1} \cap \mathring{B}_{\|.\|_1}(0,1)$ the open set of $P-1$ first components of simplex elements, in bijection with Δ^{P-1}
- $\forall \mathbf{a} = [a_1, \dots, a_P] \in \Delta^{P-1}$, we define $\check{\mathbf{a}} := [a_1, \dots, a_{P-1}] \in \check{\Delta}^{P-1}$, and $\text{ext}(\check{a}) = [\check{a}_1, \dots, \check{a}_{P-1}, 1 - \sum_{k=1}^{P-1} \check{a}_k]^T \in \Delta^{P-1}$.

Gradient :

- ∇f is the gradient of a function f in an Euclidean space $\mathbb{R}^n$.
- $\text{grad}_C f$ is the gradient of f in $C \subset \mathbb{R}^n$ as Riemannian submanifold. $\text{Hess}_C f$ the Hessian. We have $\text{grad}_{\mathbb{R}^n} \sim \nabla$ and $\text{Hess}_{\mathbb{R}^n} \sim \nabla^2$.
- We will consider in particular $\text{grad}_{\Delta^{P-1}}$ and $\text{Hess}_{\Delta^{P-1}}$, noted grad_Δ and Hess_Δ respectively.

2 Litterature review

The problem of hyperspectral unmixing, more specifically supervised unmixing (searching abundances knowing the spectral signature of the endemembers) has been widely studied. Paper [4] gives an overview of proposed methods. Geometrical method allows to come back to the supervised unmixing case (our case, knowned S) from an unsupervised one, assuming that there exists one example pure of each endmember among all the pixels.

Bayesian approaches have also been widely studied. With posterior distribution often being too complex to interpret (expectancy and covariance not easily accessible), these papers often resort to Markov Chain Monte Carlo (MCMC) to sample them (see e.g. [10]). Additionally, sampling would allow them to gain more information of the posterior compared to the bare Maximum A Posteriori (MAP).

Nonetheless, to the best of our knowledge, few articles try to exploit the spatial dependence between the pixels in the prior distribution, as we would like to. Though, one interesting approach to mention would be for instance in [11], defining a finite number of regions in the image pixels, whose prior marginal distribution share the same mean and variance.

In the field of geostatistics (analysis of various spatial earth data), *kriging* is aiming at interpolating a quantity between known values. Though, the problem becomes harder when trying to interpolate compositional data (vectors summing to a constant). The work of Aitchison [1] studying the geometry of such data and Egozcue [13] defining *isometric log ratio (ilr)* transformation on the simplex have revealed very promising for compositional weights. In the field of geostatistics, [9] uses ilr to map proportions to a Euclidean space and perform classic Gaussian process regression on the new domain. Despite the fact that in the context of kriging in the article above, prediction and observation values lie in the same space, and samples are being directly drawn from the process, such a transformation might be of great interest for our problem.

3 Problem formulation

3.1 Spectral unmixing

Assume that for every pixel $n \in [\![1, N]\!]$ we have the model :

$$\mathbf{x}_n = S\mathbf{a}_n + \mathbf{w}_n$$

with $\mathbf{x}_n \in \mathbb{R}^L$ the observed wavelengths, $S \in \mathbb{R}^{L\times P}$ (with positive coefficients) a matrix contains endmembers spectral signature in its columns, $\mathbf{a}_n \in \mathbb{R}^P$ the proportions in several elements and $\mathbf{w}_n \in \mathbb{R}^L$ white gaussian noise. $\Sigma_W = \mathbb{V}[\mathbf{w}_n] := \mathbb{E}[\mathbf{w}_n\mathbf{w}_n^T] (= \sigma_w^2 I_L$ in practice) is the covariance matrix.

This extends to the whole image as the model :

$$X = SA + W$$

with $X \in \mathbb{R}^{L\times N}$ the observed wavelengths, $A \in (\Delta^{P-1})^N \subset \mathbb{R}^{P\times N}$ the proportions and $W \in \mathbb{R}^{L\times N}$ white gaussian noise.

The conditional probability of X given A is matrix-gaussian (see [15]) of mean SA and covariance matrix Σ_W between different wavelengths, assuming no dependency between separate pixels :

$$p(X|A) = \frac{1}{(2\pi)^{LN/2}|\Sigma_W|^{N/2}} \exp\left(-\frac{1}{2}Tr[(X-SA)^\top \Sigma_W^{-1}(X-SA)]\right) \quad (1)$$

$$= \frac{1}{(2\pi)^{LN/2}|\Sigma_W|^{N/2}} \exp\left(-\frac{1}{2}\left\|\Sigma_W^{-\frac{1}{2}}(X-SA)\right\|_F^2\right) \quad (2)$$

Our objective : estimate A knowing X

3.2 Bayesian framework

For this, we could simply search for the maximum likelihood estimate :

$$\hat{A} = \text{argmax}_A p(X|A,S) = \text{argmax}_{\mathbf{a}_1\dots\mathbf{a}_n} \prod_{k=1}^{N} p(\mathbf{x}_n|\mathbf{a}_n, S)$$

Instead, we want to use a Bayesian framework, and choose a prior $p(A)$ on the space $(\Delta^{P-1})^N$. Such a prior should exploit the mutual dependence between pixels.

After this, the problem would be to obtain the Maximum A Posteriori (MAP) estimator:

$$\hat{A} = \text{argmax}_{A \in (\Delta^{P-1})^N} p(A|X,S) = \text{argmax}_{A \in (\Delta^{P-1})^N} p(X|A,S)p(A)$$

Here we have used the Bayes formula for probability densities. The problem under study is not guaranteed to be convex, and it is constrained. Several methods could be considered to find a solution : classic constrained optimization methods (e.g. interior point method [23], projected gradient descent), Riemannian gradient descent ([6]) or transposition of the instance into an unconstrained space (e.g. Mirror Descent [19]).

On top of this MAP, we would also like to get a visualization of which A are credible. For instance it means finding a credible interval for A : a set $\mathcal{R} \in (\Delta^{P-1})^N$ such that $\mathbb{P}(A \in \mathcal{R}|X,S) \geq 1-\alpha$. For this, we would need to sample the posterior distibution, i.e. produce samples $A^{(k)}$ following the distribution $A^{(k)} \sim p(A|X,S)$.

4 Methodology

4.1 Description of our toolbox

We begin by detailing the Mirror Descent and Mirror Langevin algorithms, natural choices for tackling our constrained optimization/sampling problem on the simplex using transformations.

4.1.1 Optimizing with mirror descent

Assume we want to solve the following **constrained** optimization problem :

$$\arg\min_{\mathbf{a}\in C} f(\mathbf{a}),$$

where $C \subset \mathbb{R}^n$ is a compact convex set.

In our case, $C = \Delta^{P-1}$. One clever way to proceed in this context would be to find a bijective transformation (if possible) from the simplex to an unconstrained domain. Such a function would also allow us to exploit more the geometry of our initial set (the simplex) and to define Gaussian processes in the new domain.

In classical gradient descent we perform the following constrained convex optimization at each step, with β_t the step size :

$$\mathbf{a}_{t+1} \leftarrow \arg\min_{\mathbf{a}\in C} \langle \mathbf{a}, \nabla f(\mathbf{a}_t)\rangle + \frac{1}{\beta_t}||\mathbf{a} - \mathbf{a}_t||_2^2$$

which means trying to find a tradeoff between moving in the direction of $-\nabla f(\mathbf{a}_t)$ (going in the most promising descent direction), and not moving too far from current point $\mathbf{a}_t$.

Let $\varphi : C \to \mathbb{R}$ be a twice-differentiable strongly convex function such that $\nabla\varphi$ is bijective from C to $\mathbb{R}^n$. Its bijection is $\nabla\varphi^*$, with $\varphi^* : y \mapsto \sup_{z\in C} \langle z; y\rangle - \varphi(y)$, the convex conjugate of φ (see [14]). When satisfying aforementioned properties, φ is called a "mirror map". Here we want to use the $\nabla\varphi$ map to transfer the problem into an optimization on the Euclidean space $\mathbf{R}^n$.

Instead, mirror descent procedure performs the following update instead of classic gradient descent :

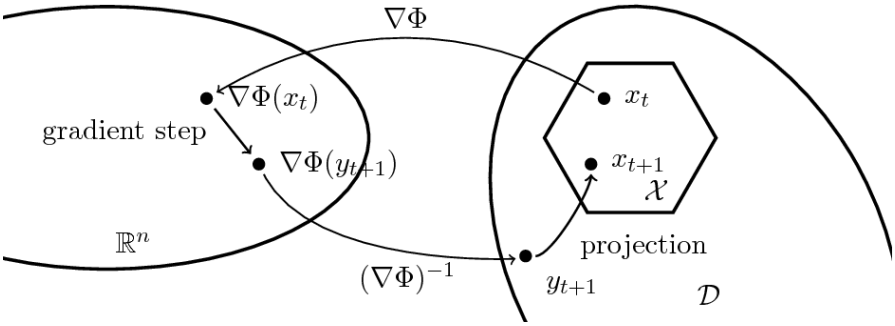


Figure 2: Mirror descent : optimization in the latent space (credits : [8])

$$\mathbf{a}_{t+1} \leftarrow \arg\min_{\mathbf{a}\in C} \langle \mathbf{a}, \nabla f(\mathbf{a}_t)\rangle + \frac{1}{\beta_t} B_\varphi(\mathbf{a}, \mathbf{a}_t) \tag{3}$$

We have replaced the Euclidean norm $\|.\|$ by the Bregman divergence generated by $\varphi : B_\varphi(\mathbf{a}, \mathbf{b}) := \varphi(\mathbf{a}) - \varphi(\mathbf{b}) - \nabla\varphi(\mathbf{b})^T(\mathbf{a} - \mathbf{b})$, which is a convex in parameter $\mathbf{a}$ when $\mathbf{b}$ is fixed. This Bregman divergence can be interpreted as a measure of how close is the function from linear between points a and b. The closer a and b are, the more linear is φ, and $B_\varphi(\mathbf{a}, \mathbf{a}) = 0$.

Above step is equivalent to performing the following algorithm :

$$\begin{cases} \mathbf{y}_{t+1} \leftarrow \nabla\varphi(\mathbf{a}_t) - \beta_t \nabla f(\mathbf{a}_t) \\ \mathbf{a}_{t+1} \leftarrow \nabla\varphi^*(\mathbf{y}_{t+1}) \end{cases} \tag{4}$$

Note
As shown in [21], performing the mirror descent (in the space $\mathbb{R}^n$) is equivalent to performing a Riemannian gradient descent on the manifold $(\nabla\varphi(\mathcal{D}), \langle \cdot; \nabla^2\varphi \cdot \rangle)$, with $\langle \cdot; \nabla^2\varphi \cdot \rangle$ the Riemannian metrics. It consists in following the update scheme :

$$y_{t+1} \leftarrow y_t - \beta_t (\nabla^2\varphi^*(y_t))^{-1} \nabla(f \circ \nabla\varphi^*)(y_t)$$

The previous mirror descent method would allow us to get the MAP

$$\text{argmax}_A \; p(A|X, S) = \text{argmin}_A \; (-\log p(A|X, S))$$

, but a minimizer would only give us the mode of the distribution. For more information about the dispersion, we would need to sample the posterior pdf.

4.1.2 Sampling with mirror langevin

In the same way that classical gradient descent extends to the *unadjusted Langevin* algorithm (ULA), mirror descent extends to so called *mirror langevin* algorithm for sampling, described in [16], which we summarize here.

Assume we want to sample distribution $\mathbf{a} \mapsto p(\mathbf{a})$ whose "potential" is $U(\mathbf{a}) = -\log p(\mathbf{a})$. The sampling adaptation comes down to performing the classic mirror descent, adding independent gaussian noise perturbation $\boldsymbol{\xi}_k \sim \mathcal{N}(\mathbf{0}_{\mathbb{R}^n}, I_n)$ at each step:

$$\begin{cases} \mathbf{y}_{t+1} \leftarrow \nabla\varphi(\mathbf{a}_t) - \beta_t \nabla(U \circ \nabla\varphi^*)(\nabla\varphi(\mathbf{a}_t)) + \sqrt{2\beta_t}\boldsymbol{\xi}_t \\ \mathbf{a}_{t+1} \leftarrow \nabla\varphi^*(\mathbf{y}_{t+1}) \end{cases}$$

After expanding $\nabla(U \circ \nabla\varphi^*)$, the update step can be rewritten and we have:

$$\begin{cases} \mathbf{y}_{t+1} \leftarrow \nabla\varphi(\mathbf{a}_t) - \beta_t [\nabla^2\varphi(\mathbf{a}_t)]^{-1}(\nabla U(\mathbf{a}_t) + \nabla \log\det \nabla^2\varphi(\mathbf{a}_t)) + \sqrt{2\beta_t}\boldsymbol{\xi}_t \\ \mathbf{a}_{t+1} \leftarrow \nabla\varphi^*(\mathbf{y}_{t+1}) \end{cases} \tag{5}$$

Complexity We notice here that this algorithms requires the computation of the inverse of the hessian $\nabla^2\varphi(\mathbf{a}_t)$ (of size $n \times n$ with n the dimension of the optimization space), which requires $\mathcal{O}(n^3)$ elementary operations in general, the time complexity bottleneck here. We might need to have a closed form expression of $[\nabla^2\varphi(\mathbf{a}_t)]^{-1}$ for high dimension n to reduce it to at least $\mathcal{O}(n^2)$ or even to $\mathcal{O}(n)$ when $[\nabla^2\varphi(\mathbf{a}_t)]^{-1}$ is diagonal for instance.

4.2 Prior model

Let's now explain how we are going to exploit the spatial dependence between the pixels in the prior.

We extend $\psi = \nabla\varphi : \Delta^{P-1} \to \mathbb{R}^{P-1}$ naturally into a transformation $(\Delta^{P-1})^N \to \mathbb{R}^{(P-1)\times N}$ by applying the transformation to each of the components.

A transformation from the simplex to a latent space would also allow us to define a process $(\mathbf{a}(\mathbf{u}))_{\mathbf{u}}$ on the spatial coordinates $\mathbf{u} = (x, y) \in \mathbb{R}^2$ (position on the image).

For the next we will note for every $l \in [\![1; P-1]\!], \alpha_l(\mathbf{u}) := [\psi(\mathbf{a}(\mathbf{u}))]_l$, the l's coefficient of the ψ-transformed weights at point $\mathbf{u}$.

We make the assumtion that coefficient $\alpha_l(\mathbf{u})$ follows a centered Gaussian process with Kernel function $\mathbf{u}, \mathbf{u}' \mapsto \mathbf{k}_l(\mathbf{u}, \mathbf{u}')$:

$$\alpha_l(\mathbf{u}) = [\psi(\mathbf{a}(\mathbf{u}))]_l \sim \mathcal{GP}(0, k_l(.,.)) \Leftrightarrow \begin{cases} \forall \mathbf{u} \in \mathbb{R}^2, \alpha_l(\mathbf{u}) \text{Gaussian} \\ \forall \mathbf{u} \in \mathbb{R}^2, \mathbb{E}(\alpha_l(\mathbf{u})) = 0 \\ \forall \mathbf{u}, \mathbf{u}' \in \mathbb{R}^2, \text{Cov}(\alpha_l(\mathbf{u}), \alpha_l(\mathbf{u}')) = k_l(\mathbf{u}, \mathbf{u}') \end{cases}$$

The kernel k_l can be any symmetric positive definite mapping

Let's also fix the covariance between the coefficients $l, m, l \neq m$ for pixels $\mathbf{u}$ and $\mathbf{u}'$:

$$\text{Cov}(\alpha_l(\mathbf{u}), \alpha_m(\mathbf{u}')) = k_{l,m}(\mathbf{u}, \mathbf{u}')$$

We can regroup all of these kernels into

$$k : \mathbf{u}, \mathbf{u}' \mapsto \begin{bmatrix} k_1(\mathbf{u}, \mathbf{u}') & k_{1,2}(\mathbf{u}, \mathbf{u}') & \dots & k_{1,P-1}(\mathbf{u}, \mathbf{u}') \\ k_{2,1}(\mathbf{u}, \mathbf{u}') & k_{2,2}(\mathbf{u}, \mathbf{u}') & & \vdots \\ \vdots & & \ddots & \vdots \\ k_{P-1,1}(\mathbf{u}, \mathbf{u}') & \dots & \dots & k_{P-1}(\mathbf{u}, \mathbf{u}') \end{bmatrix}$$

Our prior model for N pixels $\mathbf{u}_1, \dots \mathbf{u}_N$ is then :

$$vec(\psi(A)) = \begin{bmatrix} \psi(\mathbf{a}_1) \\ \vdots \\ \psi(\mathbf{a}_N) \end{bmatrix} = \begin{bmatrix} \psi(\mathbf{a}(\mathbf{u}_1)) \\ \vdots \\ \psi(\mathbf{a}(\mathbf{u}_N)) \end{bmatrix} \sim \mathcal{N}(0, K)$$

where we have gathered evaluations of kernel k on pixels $\mathbf{u}_1, \dots \mathbf{u}_N$:

$$K = \begin{bmatrix} k(\mathbf{u}_1, \mathbf{u}_1) & \dots & k(\mathbf{u}_1, \mathbf{u}_N) \\ \vdots & & \vdots \\ k(\mathbf{u}_N, \mathbf{u}_1) & \dots & k(\mathbf{u}_N, \mathbf{u}_N) \end{bmatrix}$$

We make the **separability assumption**, that K can be decomposed as :

$$K = K_N \otimes K_P$$

where $K_N \in \mathbb{R}^{N\times N}$ fixes the covariance between different pixels and $K_P \in \mathbb{R}^{(P-1)\times(P-1)}$ the covariances between different proportions inside one pixel. This hypothesis assumes that kernel functions $\mathbf{u}, \mathbf{u}' \mapsto k_l(\mathbf{u}, \mathbf{u}')$ and $\mathbf{u}, \mathbf{u}' \mapsto k_{l,m}(\mathbf{u}, \mathbf{u}')$ are all multiple of a same function $k(.,.)$ with signature K_P on the points $\mathbf{u}_1 \dots \mathbf{u}_N$.

Using the change of variable formula, the prior probability density function of A writes :

$$\begin{aligned} p_A(A) &= p_{\psi(a)}(\psi(A)) \left|\det \nabla\psi(A)\right| \\ &= \frac{1}{(2\pi)^{\frac{(P-1)N}{2}} \left|K_N\right|^{\frac{P-1}{2}}} \left(\prod_{l=1}^{N} \left|\det \nabla\psi(\mathbf{a}_l)\right|\right) \exp\left[-\frac{1}{2}\left\|K_P^{-\frac{1}{2}}\psi(A)K_N^{-\frac{1}{2}}\right\|_F^2\right] \end{aligned}$$

Fixing $K_P = I_{P-1}$ means to impose that proportions are uncorrelated, and we finally get the prior pdf :

$$p_A(A) = \frac{1}{(2\pi)^{\frac{(P-1)N}{2}} \left|K_N\right|^{\frac{P-1}{2}}} \left(\prod_{l=1}^{N} \left|\det \nabla\psi(\mathbf{a}_l)\right|\right) \exp\left[-\frac{1}{2}\left\|\psi(A)K_N^{-\frac{1}{2}}\right\|_F^2\right] \quad (6)$$

Our objective is now to find the right tranformation to use to define the prior, and to design a method for sampling method to draw from the posterior distribution.

4.3 Several transformation candidates for mirror descent

The simplex $\Delta^{P-1} := \{\mathbf{a} \in (\mathbb{R}_+^*)^P | \sum_{k=1}^P a_k = 1\}$ is a polygon included in a dimension $P-1$ hyperplane of $\mathbb{R}^{P-1}$ (see figure 3 below for case $P = 3$). With this in mind, we would like to show that the simplex is isometric to a dimension $P-1$ space, with isometries having the right properties.

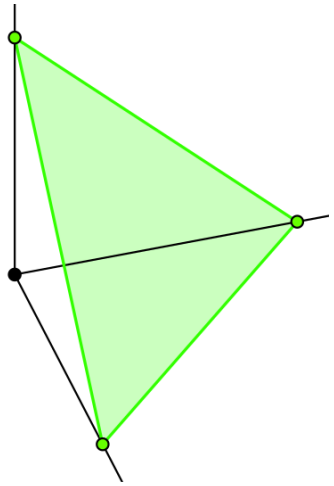


Figure 3: $P = 3$-simplex ; source : wikipedia

Where $g : \mathbf{a} \mapsto (\prod_{k=1}^{P} a_k)^{\frac{1}{P}}$ is the geometric mean of the proportions.

What kind of transformation from the simplex are we searching for ?

- ψ should match the simplex Δ^{P-1} to an euclidean space (or a hyperplane)
- ψ should be invariant by permutation of the components (such that the prior $\psi(A) \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ is the least informative possible). We define the "isotropy property" below.
- ψ should define a natural isometry to $\mathbb{R}^{P-1}$
- ψ We should check that the transformation is the gradient of a mirror map for being able to use mirror descent
- If so, are the hessian and it's inverse diagonal ? It would greatly simplify the computation of the product in 5, puting the complexity down to vector operations : $\mathcal{O}(P)$ for each iteration.

Definition *Isotropy*

- A random variable A on $\mathcal{D} \in \mathbb{R}^n$ subset of an euclidean space is said to be **permutation-invariant** if and only if for every permutation matrix $P \in \{0;1\}^{n\times n}$, PA has the same probability density as A
- A transformation $\psi : \Delta^{P-1} \to \mathbb{R}^{P-1}$ is said to be **isotropic** if and only if ψ^{-1} preserves permutation-invariance (B permutation-invariant $\Rightarrow \psi^{-1}(B)$ permutation-invariant)

We have the feeling that the entropy :

$$h : \begin{cases} \Delta^{P-1} \to \mathbb{R} \\ \mathbf{a} \mapsto \sum_{k=1}^{P} a_k \log a_k \end{cases} = \begin{cases} \Delta^{P-1} \to \mathbb{R} \\ \mathbf{a} \mapsto \sum_{k=1}^{P} a_k \log a_k + (1 - \sum_{k=1}^{P} a_k) \end{cases} \tag{7}$$

might be a suitable mirror map for the simplex, as it is often used when dealing with distributions, or more generally elements summinng to one. Though, as Δ^{P-1} is a close set of $\mathbb{R}^P$ (with in fact the structure of a "cut hyperplane plane") one cannot differentiate the entropy on it. Each of the transformations presented below offers a distinct method to circumvent this difficulty and guaranty the differentiability of the mirror map.

4.3.1 Additive Log Ratio (ALR) a.k.a. logit

A first clever way to overcome the simplex being a closed set of Δ^{P-1} would be to parametrize the simplex by its $P-1$ first components :

$$\check{\Delta}^{P-1} := \{\check{\mathbf{a}} \in (\mathbb{R}_+^*)^{P-1} | \sum_{k=1}^{P-1} \check{a}_k < 1\}$$

which is an open set of $\mathbb{R}^{P-1}$ in trivial bijection with Δ^{P-1}.

The logit transformation, also known as the additive log-ratio (alr) transformation, stems from the entropy map considered in this new space :

$$\check{h} : \check{\mathbf{a}} \mapsto h\left(\begin{bmatrix} \check{a}_1 \\ \vdots \\ \check{a}_{P-1} \\ 1 - \sum_{k=1}^{P-1} \check{a}_k \end{bmatrix}\right) = h(\text{ext}(\check{\mathbf{a}})) = \sum_{k=1}^{P-1} \check{a}_k \log \check{a}_k + \left(1 - \sum_{k=1}^{P-1} \check{a}_k\right) \log\left(1 - \sum_{k=1}^{P-1} \check{a}_k\right) \tag{8}$$

Differentiating gives us the following transformation :

$$\text{logit} : \begin{cases} \check{\Delta}^{P-1} \to \mathbb{R}^{P-1} \\ \check{\mathbf{a}} \mapsto \left(\log \check{a}_k - \log\left(1 - \sum_{l=1}^{P-1} \check{a}_l\right)\right)_{k=1,\dots,P-1} \\ \quad = (\log a_k - \log a_P)_{k=1,\dots,P-1} \end{cases} \tag{9}$$

where $\mathbf{a} = \text{ext}(\check{\mathbf{a}})$.

The inverse of the LOGIT transformation writes :

$$\text{logit}^{-1} : \begin{cases} \mathbb{R}^{P-1} \to \check{\Delta}^{P-1} \\ (b_1, \dots, b_{P-1}) \mapsto \left(\frac{\exp(b_1)}{1+\sum_{k=1}^{P-1} \exp(b_k)}, \dots, \frac{\exp(b_{P-1})}{1+\sum_{k=1}^{P-1} \exp(b_k)}\right) \end{cases}$$

The LOGIT transformation **not isotropic**, as we can see in 4.

hessian Using the definition of *logit* as deriving from mirror map in 8, we can reuse the Hessian of the original entropy (19) :

$$\nabla \text{logit}(\check{\mathbf{a}}) = \nabla^2 \check{h}(\check{\mathbf{a}}) = \begin{bmatrix} \frac{1}{\check{a}_P} + \frac{1}{\check{a}_1} & \cdots & \frac{1}{\check{a}_P} \\ \vdots & \ddots & \vdots \\ \frac{1}{\check{a}_P} & \cdots & \frac{1}{\check{a}_P} + \frac{1}{\check{a}_{P-1}} \end{bmatrix}$$

Using formula 21 from the annex for computing its determinant, we get :

$$\det(\nabla \text{logit}(\check{\mathbf{a}})) = (\prod_{k=1}^{P-1} \frac{1}{\check{a}_k}) \frac{1}{1 - \sum_{k=1}^{P-1} \check{a}_k}$$

hessian inverse Using 22,we can get a closed form of $\nabla \text{logit}(\check{\mathbf{a}}) = \text{Diag}(\frac{1}{a_1}, \dots \frac{1}{a_{P-1}}) + \frac{1}{a_P} 1 1^T$.

$$(\nabla \text{logit}(\check{\mathbf{a}}))^{-1} = \text{Diag}(\check{\mathbf{a}}) - \check{\mathbf{a}}\check{\mathbf{a}}^T = [\delta_{i,j}a_i - a_i a_j]_{i,j}$$

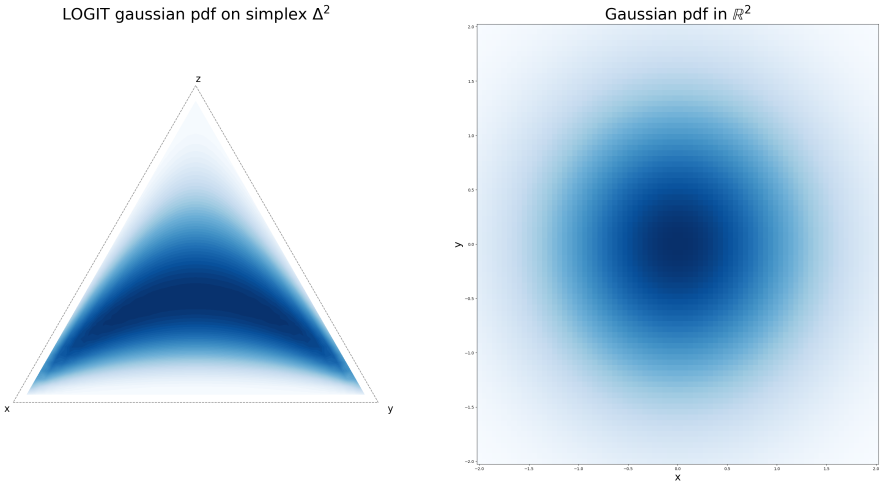


Figure 4: left : isotropic zero-mean gaussian pdf on $\mathbb{R}^2$; right : corresponding LOGIT-gaussian pdf ; for $P = 3$, $\boldsymbol{\mu} = [0\ 0]^T; \Sigma = I_2$

Mirror Langevin sampling with LOGIT Sampling the distribution $p(\mathbf{a}|\mathbf{x}, S) \propto \exp(-\check{U}(\check{\mathbf{a}}))$ (with $\check{U}(\check{\mathbf{a}}) := U(\text{ext}(\check{\mathbf{a}}))$) with mirror Langevin and ALR transformation for one pixel reverts to performing the following iterations :

$$\text{logit}(\check{\mathbf{a}}_{t+1}) \leftarrow \text{logit}(\check{\mathbf{a}}_t) - \beta_t(\text{Diag}(\check{\mathbf{a}}_t) - \check{\mathbf{a}}_t\check{\mathbf{a}}_t^T)(\nabla\check{U}(\check{\mathbf{a}}_t) - \check{\mathbf{a}}_t^{-1} + \frac{1}{a_{t,P}}\mathbf{1}) + \sqrt{2\beta_t}\boldsymbol{\xi}_t$$

where $\boldsymbol{\xi}_t \sim \mathcal{N}(0, I_P)$, and noting $\check{\mathbf{a}}_t{}^{-1} := [\frac{1}{\check{a}_{t,k}}]_k$, and $a_{t,P} := 1 - \sum_{k=1}^{P-1}\check{a}_{t,k}$

This also writes :

$$\text{logit}(\check{\mathbf{a}}_{t+1}) \leftarrow \text{logit}(\check{\mathbf{a}}_t) - \beta_t(\check{\mathbf{a}}_t \odot \nabla\check{U}(\check{\mathbf{a}}_t) - \left\langle \check{\mathbf{a}}_t; \nabla\check{U}(\check{\mathbf{a}}_t)\right\rangle \check{\mathbf{a}}_t + (P-1)\check{\mathbf{a}}_t - \mathbf{1}) + \sqrt{2\beta_t}\boldsymbol{\xi}_t$$

The update only uses vector sums, hadamard and scalar products, iteration requires only $\mathcal{O}(P)$ elementary operations.

4.3.2 Centralized Log Ratio (CLR)

Even if the entropy $h : \Delta^{P-1} \to \mathbb{R}$ is not differentiable in a classic way on the closed set $\Delta^{P-1} \subset (\mathbb{R}_+^*)^P$, we can compute its gradient in the Riemannian submanifold $(\Delta^{P-1}; \langle ., . \rangle)$ [1], and get the following Centered log ratio (CLR) transformation :

$$\text{clr} = \text{grad}_\Delta h \tag{10}$$

The CLR transformation is :

$$clr : \begin{cases} \Delta^{P-1} \to \mathbb{R}^P \\ \mathbf{a} \mapsto G_P \log \mathbf{a} = \left[\log a_k - \frac{1}{P}\sum_{l=1}^P \log a_l\right]_{1 \le k \le P} \end{cases}$$

Property 4.1
$clr : \Delta^{P-1} \to \mathbf{1}^\perp$ is an **isometry** from the Aitchison simplex [13] $(\Delta^{P-1}, \langle ; \rangle_A)$ to the euclidean hyperplane space $(\mathbf{1}^\perp; \langle ; \rangle)$:

$$\forall \mathbf{a}, \mathbf{b} \in \Delta^{P-1}, \langle \text{clr}(\mathbf{a}); \text{clr}(\mathbf{b}) \rangle = \langle \mathbf{a}; \mathbf{b} \rangle_A$$

With the following operations in the Aitchison simplex : $\forall \mathbf{a}, \mathbf{b} \in \Delta^{P-1}, \forall \lambda \in \mathbb{R}$,

$$\mathbf{a} \oplus \mathbf{b} := \frac{1}{\sum_{k=1}^P a_k b_k}(a_1 b_1, \dots a_P b_P); \lambda \cdot \mathbf{a} := \frac{1}{\sum_{k=1}^P a_k^\lambda}(a_1^\lambda, \dots, a_P^\lambda); \langle \mathbf{a}; \mathbf{b} \rangle_A := \sum_{k=1}^P \log(\frac{a_k}{g(\mathbf{a})}) \log(\frac{b_k}{g(\mathbf{b})})$$

The inverse of the clr transformation

$$\text{clr}^{-1} : \begin{cases} \mathbf{1}^\perp \to \Delta^{P-1} \\ (b_1, \dots, b_{P-1}) \mapsto \left(\frac{\exp(b_1)}{\sum_{k=1}^P \exp(b_k)}, \dots, \frac{\exp(b_P)}{\sum_{k=1}^P \exp(b_k)}, \right) \end{cases}$$

The CLR transformation is **isotropic**, as we can see in 5.

Hessian The Hessian of the entropy mirror map is here (using B) :

$$\text{grad}_\Delta \text{clr}(\mathbf{a}) = \nabla^2 h(\mathbf{a}) = G_P \text{Diag}(\frac{1}{a_1}, \dots \frac{1}{a_p}) G_P \tag{11}$$

[1] see annex B for a development of these mathematical tools

where the gradient is here taken into the simplex Δ^{P-1} as a Riemannian submanifold of $(\mathbb{R}^P; \langle ; \rangle)$. Here, the hessian is written as a symmetric endomorphism $\mathbb{R}^P \to \mathbb{R}^P$, with kernel being $\mathbf{1}^\perp$. Though, a more natural way would be to consider its restriction to $\mathbf{1}^\perp$, the tangent space to the simplex. The ILR transform which will proposed later simply consists in writing the gradients and hessians in a suitable basis for $\mathbf{1}^\perp$, described by matrix a H_P

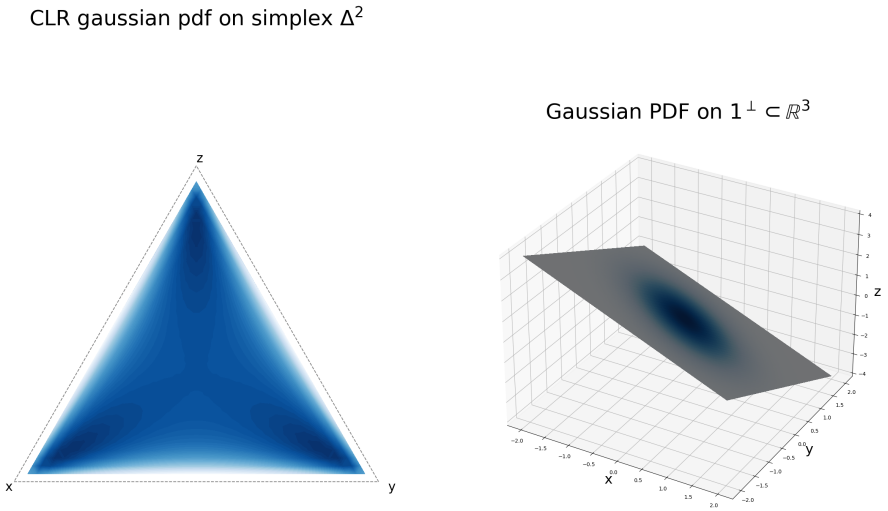


Figure 5: left : gaussian pdf on hyperplane $(\begin{bmatrix}1\\1\\1\end{bmatrix})^\perp \subset \mathbb{R}^3$; right : corresponding Clr-gaussian pdf ; for $P = 3$, $\boldsymbol{\mu} = [0; 0]^T; \Sigma = I_2$

Mirror Langevin sampling with CLR Sampling the distribution $p(\mathbf{a}|\mathbf{x}, S) \propto \exp(-U(\mathbf{a}))$ with mirror Langevin and CLR transformation for one pixel reverts to performing the following iterations :

$$\mathrm{clr}(\mathbf{a}_{t+1}) \leftarrow \mathrm{clr}(\mathbf{a}_t) - \beta_t(\mathrm{Diag}(\mathbf{a}_t) - \mathbf{a}_t\mathbf{a}_t^T)(\mathrm{grad}_\Delta U(\mathbf{a}_t) - {\mathbf{a}_t}^{-1}) + \sqrt{2\beta_t} G_P \boldsymbol{\xi}_t$$

where $\boldsymbol{\xi}_t \sim \mathcal{N}(0, I_P)$, and noting ${\mathbf{a}_t}^{-1} := [\frac{1}{a_{t,k}}]_k$.

which writes :

$$\text{clr}(\mathbf{a}_{t+1}) \leftarrow \text{clr}(\mathbf{a}_t) - \beta_t(\mathbf{a}_t \odot \text{grad}_\Delta U(\mathbf{a}_t) - \langle \mathbf{a}_t; \text{grad}_\Delta U(\mathbf{a}_t)\rangle \, \mathbf{a}_t + P\mathbf{a}_t - \mathbf{1}) + \sqrt{2\beta_t} G_P \boldsymbol{\xi}_t$$

As the update only uses vector sums and hadamard products, iteration requires $\mathcal{O}(P)$ elementary operations.

4.3.3 Isometric Log Ratio (ILR)

The ilr transformation (first introduced in [12]) simply stems from the expression of CLR in an orthonormal basis of $\mathbf{1}^\perp$ [2] :

$$\text{ilr}(\mathbf{a}) = H_P \text{grad}_\Delta h = H_P \cdot \text{clr}(\mathbf{a}) (= [\langle \mathbf{v}_i, \text{clr}(\mathbf{a})\rangle]_{1 \le i \le P-1}) \tag{12}$$

where $H_P \in \mathbb{R}^{(P-1)\times P}$ is a matrix whose rows are vectors from an orthogonal basis in $\mathbf{1}^\perp$.

$$H_P = \begin{bmatrix} \mathbf{v}_1^T \\ \vdots \\ \mathbf{v}_{P-1}^T \end{bmatrix}$$

The ILR transformation provides an **isometric** mapping from the simplex Δ^{P-1} to $\mathbb{R}^{P-1}$.

Example *P=3 [12]*

$$H_3 = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & -\frac{2}{\sqrt{6}} \end{bmatrix}$$

As the CLR transformation, the ILR transformation is **isotropic**, as we can see in 6.

Hessian

$$\text{grad}_\Delta \text{ilr}(\mathbf{a}) = H_P G_P \text{Diag}(\frac{1}{a_1}, \dots \frac{1}{a_p}) G_P^T H_P^T = H_P \text{Diag}(\frac{1}{a_1}, \dots \frac{1}{a_p}) H_P^T \tag{13}$$

[2] H_P has the following properties : $H_P H_P^T = I_{P-1}$ $H_P^T H_P = G_P$ $G_P H_P = H_P$

Using successively 21 and 23 the determinant of this matrix can be computed as :

$$\det(\text{grad}_\Delta \text{ilr}(\mathbf{a})) = \frac{1}{P}\prod_{k=1}^{P}\frac{1}{a_k} \tag{14}$$

hessian inverse using successively 22 and 24, we can express the hessian inverse in closed form :

$$(\text{grad}_\Delta \text{ilr}(\mathbf{a}))^{-1} = H_P(\text{Diag}(\mathbf{a}) - \mathbf{a}\mathbf{a}^T)H_P^T \tag{15}$$

Mirror Langevin sampling with ILR Sampling the distribution $p(\mathbf{a}|x,S) \propto \exp(-U(\mathbf{a}))$ with mirror Langevin and ILR transformation reverts to performing the following iterations :

$$\text{ilr}(\mathbf{a}_{t+1}) \leftarrow \text{ilr}(\mathbf{a}_t) - \beta_t H_P(\text{Diag}(\mathbf{a}_t) - \mathbf{a}_t\mathbf{a}_t^T)H_P^T(H_P\text{grad}_\Delta U(\mathbf{a}_t) - H_P{\mathbf{a}_t}^{-1}) + \sqrt{2\beta_t}\boldsymbol{\xi}_t \tag{16}$$

$$= \text{ilr}(\mathbf{a}_t) - \beta_t H_P(\text{Diag}(\mathbf{a}_t) - \mathbf{a}_t\mathbf{a}_t^T)(\text{grad}_\Delta U(\mathbf{a}_t) - {\mathbf{a}_t}^{-1}) + \sqrt{2\beta_t}\boldsymbol{\xi}_t \tag{17}$$

where $\boldsymbol{\xi}_t \sim \mathcal{N}(\mathbf{0}, I_{P-1})$, and noting ${\mathbf{a}_t}^{-1} := [\frac{1}{a_{t,k}}]_k$.

The update requires multiplying by H_P, which needs at least around $\mathcal{O}(P^2)$ elementary operations at each step.

However, since the ILR transformation is simply a reparameterization of the CLR transformation, changing only the basis of the tangent space (in which the gradient and hessian are written), performing mirror Langevin sampling with ILR is trivially equivalent to doing so with CLR. The only difference lies in the form of the update step, which, in the ILR case, is computationally more efficient.

In practice, one will then only consider CLR update steps 4.3.2 for comparisons.

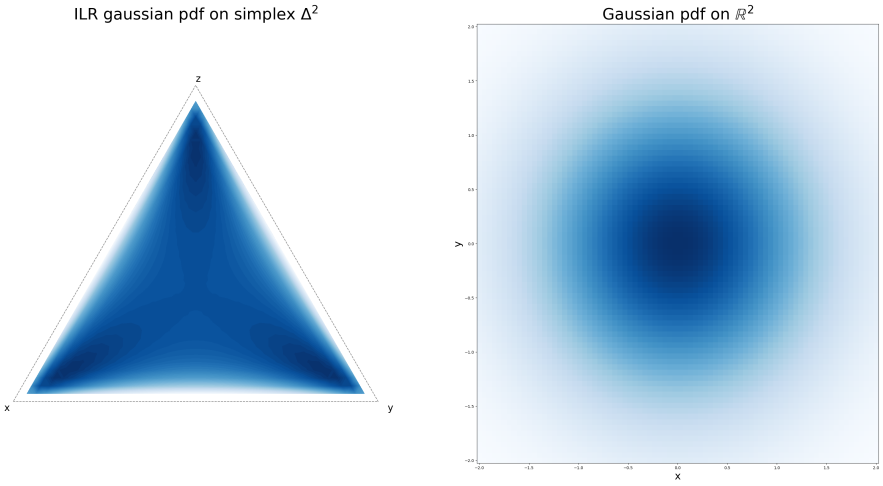


Figure 6: left : gaussian pdf on $\mathbb{R}^2$; right : corresponding ILR-gaussian (equivalent to CLR-gaussian) pdf ; for $P = 3$, $\boldsymbol{\mu} = [0; 0]^T; \Sigma = I_2$

4.3.4 Log (on compositional data)

Let's propose here a new, original way to transform the simplex, by extending it to the whole space $\mathbb{R}^P$ of compositional data.

This approach has been inspired by the multiplicative weight update (MWU) algorithms [2], a computationally efficient case of mirror descent with projection, for optimizing on the simplex. In our case, we found a way to perform mirror Langevin as in [16] (without projection, the latent space being in bijection with the simplex), which would allow us to exploit the convergence results of this paper.

The LOG transformation we are going to define derives from the mirror map :

$$\bar{h} : \begin{cases} {\mathbb{R}_+^*}^P \to \mathbb{R}^P \\ \mathbf{c} \mapsto \sum_{k=1}^P c_k \log c_k - \sum_{k=1}^P c_k \end{cases}$$

(whose restriction on Δ^{P-1} is the classic entropy function : $h : \mathbf{a} \in \Delta^{P-1} \to \sum_{k=1}^P a_k \log a_k$)

LOG transformation is then simply :

$$\log : \begin{cases} {\mathbb{R}_+^*}^P \to \mathbb{R}^P \\ \mathbf{c} \mapsto [\log c_k]_k \end{cases} \tag{18}$$

It is defined on the cone
$(\mathbb{R}_+^*)^P = \cup_{s\in\mathbb{R}_+^*} s\Delta^{P-1} = \cup_{s\in\mathbb{R}_+^*} \{(c_k)_k, \sum_{k=1}^P c_k = s\}$.
Which could be seen as the set of compositional data (i.e. values suming to a constant).

Hessian Gradient of the log transform on $({\mathbb{R}_+}^*)^P$ is :

$$\nabla \log(\mathbf{c}) = \nabla^2 \bar{h}(\mathbf{c}) = \begin{bmatrix} \frac{1}{c_1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \frac{1}{c_P} \end{bmatrix} \tag{19}$$

which is diagonal, with determinant $\prod_{k=1}^P \frac{1}{c_k}$

Mirror Langevin sampling with LOG Here we propose an original method to sample **a** using the log transformation defined above.

To sample the random variable $\mathbf{A}|\mathbf{X}$ in Δ^{P-1}, the trick is to introduce a new random variable $\mathbf{C}$ on $(\mathbb{R}_+^*)^P$:

Fixing S (independent of $\mathbf{A}$) a random variable on $\mathbb{R}_+^*$, with $s \mapsto p_S(s)$, one can sample distribution $\mathbf{C} := S\mathbf{A}$ in ${\mathbb{R}_+^*}^P$ with its pdf.

The pdf of $\mathbf{C}|X$ is knowned in closed form, so $\mathbf{C}$ can be sampled. :

$$\begin{aligned} p(\mathbf{c}|\mathbf{x}) &= p(s\mathbf{a}|\mathbf{x}) = \frac{1}{s^{P-1}} p_S(s) p_{\mathbf{A}|\mathbf{X}}(\mathbf{a}) \\ \Rightarrow U(\mathbf{c}) &= -\log p(\mathbf{c}) = U_A(\frac{\mathbf{c}}{\|\mathbf{c}\|_1}) + U_s(\|\mathbf{c}\|_1) + (P-1)\log(\|\mathbf{c}\|_1) \end{aligned} \tag{20}$$

where we have used here a change of variable for the second equality.

It will be possible to recover $\mathbf{A}$ samples : $\frac{\mathbf{C}}{\|\mathbf{C}\|_1} = \mathbf{A}$ almost surely.

Mirror Langevin with CLR transformation reverts to performing the following iterations :

$$\begin{aligned}\log(\mathbf{c}_{t+1}) &\leftarrow \log(\mathbf{c}_t) - \beta_t \mathrm{Diag}(\mathbf{c}_t)(\nabla U(\mathbf{c}_t) - {\mathbf{c}_t}^{-1}) + \sqrt{2\beta_t}\boldsymbol{\xi}_t \\ &= \log(\mathbf{c}_t) - \beta_t(\mathbf{c}_t \odot \nabla U(\mathbf{c}_t) - \mathbf{1}) + \sqrt{2\beta_t}\boldsymbol{\xi}_t\end{aligned}$$

As the update only uses vector sums and hadamard products, iteration requires only $\mathcal{O}(P)$ elementary operations, like LOGIT or CLR. Though, this very simple update step might show slighly more efficient.

4.3.5 Summary of the transformation properties

In 1 is a summary of the transformations we have defined :

$\psi = \nabla\varphi$	domain	expression	$\nabla^2\varphi$	$[\nabla^2\varphi]^{-1}$	$\det(\nabla^2\varphi)$
alr (logit) :	$\check{\Delta}^{P-1} \to \mathbb{R}^{P-1}$	$\mathbf{a} \mapsto [\log a_k - \log a_P]_{1\le k\le P-1}$	$[\frac{\delta_{i,j}}{a_{i,j}} + \frac{1}{a_P}]_{1\le i,j\le P-1}$	$\mathrm{Diag}(\check{\mathbf{a}}) - \check{\mathbf{a}}\check{\mathbf{a}}^T$	$\prod_{k=1}^P \frac{1}{a_k}$
clr :	$\Delta^{P-1} \to \mathbf{1}^\perp \subset \mathbb{R}^P$	$\mathbf{a} \mapsto \left[\log a_k - \frac{1}{P}\sum_{l=1}^P \log a_l\right]_{1\le k\le P}$	$G_P\mathrm{Diag}(1/\mathbf{a})G_P^T$ [a]	$\mathrm{Diag}(\mathbf{a}) - \mathbf{a}\mathbf{a}^T$ [a]	$\frac{1}{P}\prod_{k=1}^P \frac{1}{a_k}$
ilr :	$\Delta^{P-1} \to \mathbb{R}^{P-1}$	$\mathbf{a} \mapsto H_P\mathrm{clr}(\mathbf{a})$	$H_P\mathrm{Diag}(1/\mathbf{a})H_P^T$	$H_P(\mathrm{Diag}(\mathbf{a}) - \mathbf{a}\mathbf{a}^T)H_P^T$	$\frac{1}{P}\prod_{k=1}^P \frac{1}{a_k}$
log :	${\mathbb{R}_+^*}^P \to \mathbb{R}^P$	$\mathbf{c} \mapsto [\log c_k]_{1\le k\le P}$	$\mathrm{Diag}(1/\mathbf{c})$	$\mathrm{Diag}(\mathbf{c})$	$\prod_{k=1}^P \frac{1}{c_k}$

[a] One of the possible matrices, representing an endomorphism $\mathbb{R}^P \to \mathbb{R}^P$ stable with respect to the subspace $1^\perp \subset \mathbb{R}^P$.
Inverse is taken only on the $\mathbf{1}^\perp$ space.

Table 1: Transformations

Here, we have noted $1/\mathbf{a} := [\frac{1}{a_k}]_{1\le k\le P}$

Their respective properties are summarized in table 2 :

Transformation ψ	**Isometric** (Aitchison simplex)	**Isotropic**	Time complexity of $v \mapsto [\nabla^2\varphi(\mathbf{a})]^{-1}v$
alr (logit) : $\check{\Delta}^{P-1} \to \mathbb{R}^{P-1}$	✗	✗	$\sim 6P = \mathcal{O}(P)$
clr : $\Delta^{P-1} \to \mathbf{1}^\perp \subset \mathbb{R}^P$	✓	✓	$\sim 5P = \mathcal{O}(P)$
ilr : $\Delta^{P-1} \to \mathbb{R}^{P-1}$	✓	✓	$\mathcal{O}(P^2)$
log : ${\mathbb{R}_+^*}^P \to \mathbb{R}^P$	✗	✓	$\sim 2P = \mathcal{O}(P)$

Table 2: Transformation properties

From our development above, we observe that ILR/CLR and LOG transorms seem to be best candidate for the prior (unlike LOGIT, which is biased). LOGIT,

CLR and LOG transforms in mirror descent show less time complexity, requiring only $\mathcal{O}(P)$ elementary operations. We might also have the intuition that using the same transformation for the prior and the optimization / sampling would be the best, at least for later interpretability of our algorithm.

For now, using CLR transform with CLR-gaussian prior would seem to be the best choice in our setup.

In the next chapter, we are going to implement mirror descent and confront these various methods, against a more classical / straightforward sampling method.

4.4 Estimating confidence intervals

In our original problem, we know the closed-form expression of the posterior distribution $p(A|S, X)$. Though, this distribution this expression is too complex to study analytically, and the marginal distributions $p(A_{i,j}|X, S)$ for pixel i, j might not have a closed form when using a spatial dependance for the prior !

Information about dispersion is crucial to our problem, as it's necessary to assess for the confidence in our estimation.

An example of complex posterior distribution for one pixel case is shown in figure 7.

As we see for example in this case, the distribution potential might, in general, have a complex shape, and convexity of it might not be guaranteed !

So, we might need to sample it to study its properties : modes and dispersion around these modes.

How to get a credible interval out of samples from the posterior distribution ?

In Bayesian inference, confidence intervals used in a "frequentist" approach are replaced by *credible regions* :

> **Definition** *Credible region*
> A credible region $\mathcal{I}_\alpha$ with parameter $0 \leq \alpha \leq 1$ is defined as a subset of the domain such that it contains the true value $\bar{X}$ with probability at least $1 - \alpha$ under the posterior distribution [5]:
>
> $$\mathbb{P}(X \in \mathcal{I}_\alpha) \geq 1 - \alpha.$$

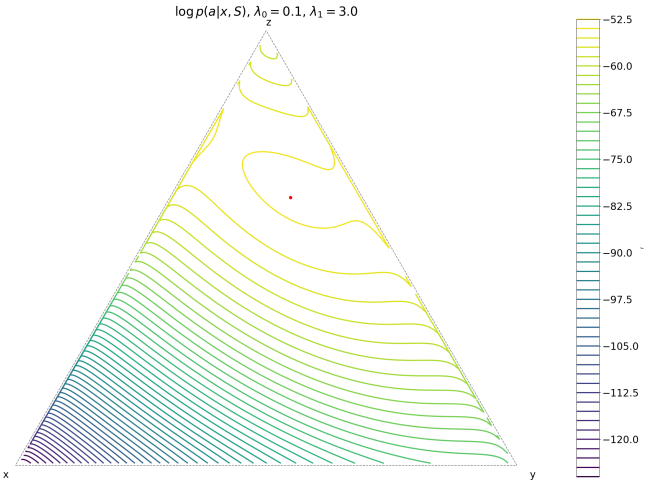


Figure 7: $\log p(\mathbf{a}|x,s)$ for one point (no kernel), red point : true value of **a** ILR-gaussian prior with $\lambda_0 = 0.1$ and confidence in the data $\lambda_1 = 3$

The ideal credible region would be the one with minimal volume. In fact, it turns out that such a region has a spectific form : it can be defined as the set of the points of the space whose density is greater than a certain threshold.

Property 4.2
For a distribution with density f on domain $\mathcal{D}$, the **Highest Density Region (HDR)** is defined as:

$$\mathcal{I}_\alpha^{\text{minimal}} = \{x \in \mathcal{D} \mid f(x) \geq f_\alpha\},$$

where f_α is the largest threshold such that

$$\mathbb{P}\left(X \in \mathcal{I}_\alpha^{\text{minimal}}\right) = 1 - \alpha.$$

This region is a minimal volume credible region.

The above statement is mentioned in [7], p.123, and proven for dimension 1 in [18].

An example is given in Figure 8, where the HDR correctly captures the high-probability modes of a multimodal posterior distribution.

Since the posterior distribution in our case does not admit a closed-form, we rely

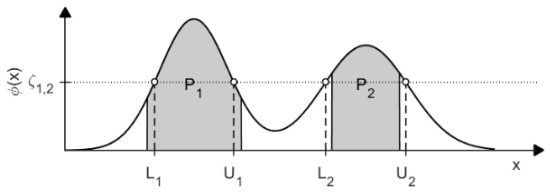


Figure 8: Multimodal distribution. In gray : HDR

on Monte Carlo samples $A_1, \ldots, A_n$ to approximate the credible region.

Following [17], one can estimate the threshold f_α as

$$\hat{f}_\alpha := f_{(\lfloor \alpha n \rfloor)},$$

where $f_{(j)}$ denotes the j-th largest value among $\{f(A_1), \ldots, f(A_M)\}$. Then, the estimated HDR is given by:

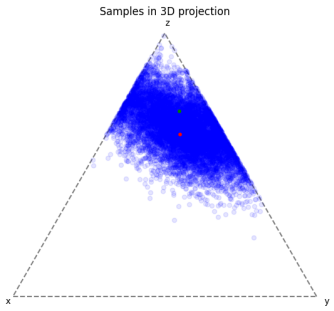
$$\hat{\mathcal{I}}_\alpha = \{A_i \mid f(A_i) \geq \hat{f}_\alpha\}.$$

In practice, this means that once posterior samples are obtained, we can sort them by their posterior density values and retain the top $(1 - \alpha)$ proportion of them.

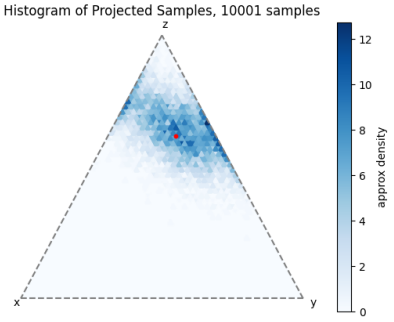
Algorithm: Estimating the HDR from samples.

1. Draw M samples $A_1, \ldots, A_M$ from the posterior $p(A|X)$.
2. Evaluate the posterior density $f(A_i)$ at each sample.
3. Sort the values $f(A_i)$ in decreasing order.
4. Find $\hat{f}_\alpha$ as the $\lfloor \alpha n \rfloor$-th largest value.
5. Define $\hat{\mathcal{I}}_\alpha = \{A_i \mid f(A_i) \geq \hat{f}_\alpha\}$.

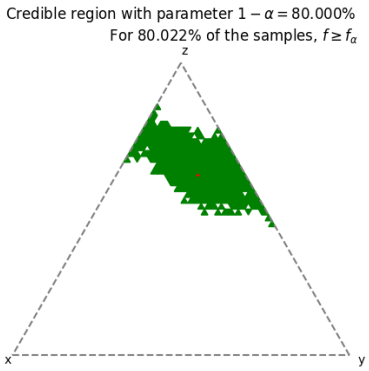
Below is an example, in the $P = 3$ simplex, of $n_{samples} = 10001$ samples and the corresponding credible region.



(a) Samples



(b) Approximate pdf



(c) Confidence region, $\alpha = 20\%$

Here, we have based our credible interval building on pdf estimates (as we don't know it's expressions in this precise case).

The threshold for the probability density has been estimated to $\simeq 3.12$ here.

5 Experimental results

The code used for numerical studies is provided in :

https://gitlab.imt-atlantique.fr/h25blond/logit_gaussian_remote_sensing

5.1 Setup : Houston dataset

For the following, we will mainly use the Houston dataset. From a dictionnary of $P = 4$ endmembers gathered from a spectral study, we compose the expected satellite visible image using the spectral signature of $L = 144$ wavelengths, measured for each endmembers.

The full image is shown here :

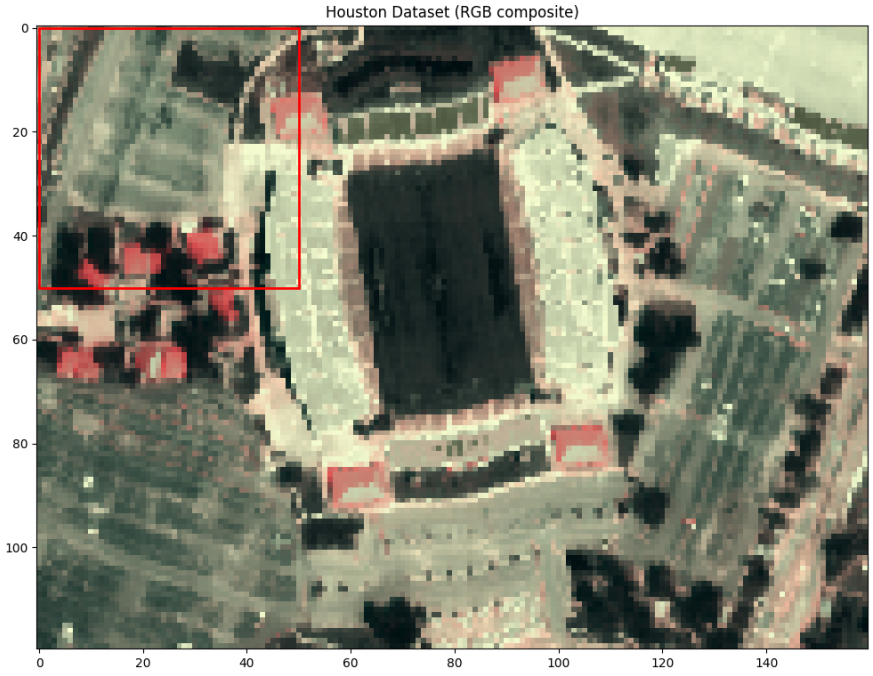


Figure 10: full and 50 x 50 sub-image

and corresponding endmembers :

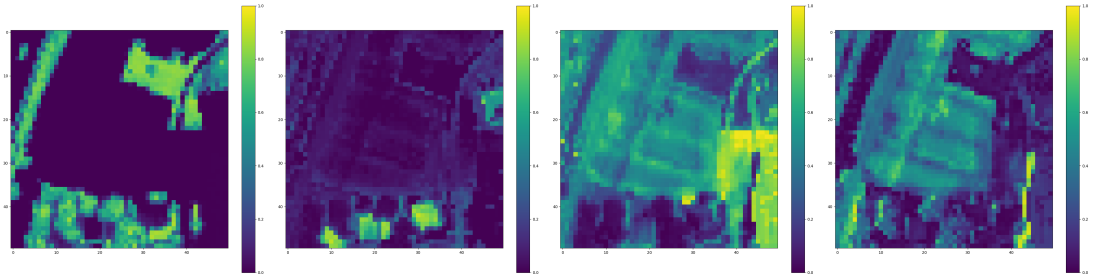


Figure 11: $P = 4$ endmembers for the 50x50 sub-image

The first endmember is the vegetation element, the second is brick, third concrete and fourth one bitumen.

5.2 One pixel case simple examples

First, we focus our study on unmixing one pixel, whose location is shown in 12.

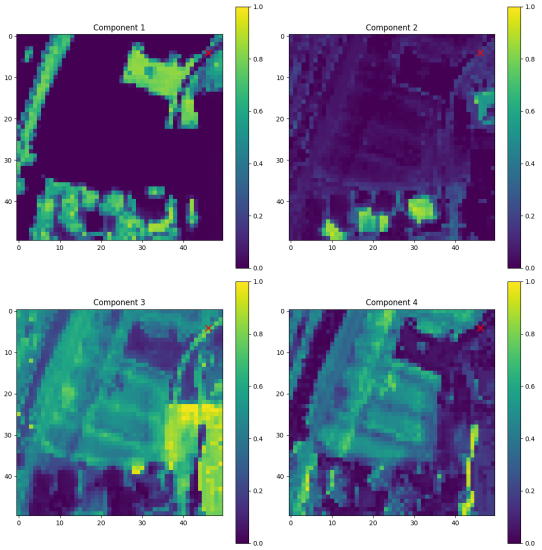


Figure 12: Pixel location

Let's first study the simple case: unmixing the elements in one pixel between $P = 4$ endmembers. For each simulation, we perform mirror Langevin for

sampling 1001 points, for an ilr-Gaussian distribution, using a step size $\beta = 10^{-3}$. We estimate the posterior density and build a credible interval thanks to the method developped in 4.4.

We vary the model hyperparameters :

- $\lambda_0 := \|K^{-1}\|_F$ is the confidence in the prior (opposite of the prior variance in the latent space) : $-\log p_{\psi(\mathbf{A})}(\psi(\mathbf{a})) = \frac{\lambda_0}{2} \|\psi(\mathbf{a})\|^2 \propto \lambda_0$
- $\lambda_1 := \|\Sigma_W^{-1}\|_F$ is the confidence in the measurements (opposite of the noise variance) : $-\log p_{\mathbf{A}|\mathbf{X}}(\mathbf{a}|\mathbf{x}) = \frac{\lambda_1}{2} \|\mathbf{a} - S\mathbf{x}\|^2 \propto \lambda_1$

to investigate their influence on the posterior distribution :

$$\left\{ -\log p_{\psi(\mathbf{A})}(\psi(\mathbf{a})) = \frac{\lambda_0}{2} \|\psi(\mathbf{a})\|^2 \propto \lambda_0 \right.$$

In the case $\lambda_0 = 10$ and $\lambda_1 = 1000$, the prior remains relatively spread, but the high value of λ_1 strongly enforces the data fidelity. As a consequence, the samples concentrate around the true abundances, with limited variability.

For $\lambda_0 = 10$ and $\lambda_1 = 100$, the confidence in the data is lower than in the previous case. The posterior distribution is therefore more spread, with a greater variability of the samples around the true abundances.

Finally, in the case $\lambda_0 = 100$ and $\lambda_1 = 100$, the prior becomes much sharper (less spread), which constrains the abundances more strongly. The samples are less dispersed due to the stronger prior, but the lower value of λ_1 still leads to a relatively spread posterior compared to the first case.

From the previous figures, we observe as we would have expected that:

- the bigger λ_1 is, the more confidence we grant to the data compared to the prior;
- the bigger λ_0 is, the less spread will be the prior;
- the bigger λ_1 is, the less spread will be the points around the truth, meaning we give less confidence, i.e. the posterior distribution is more spread.

(a) $\lambda_0 = 10; \lambda_1 = 1000$ **(b)** $\lambda_0 = 10; \lambda_1 = 100$ **(c)** $\lambda_0 = 100; \lambda_1 = 100$

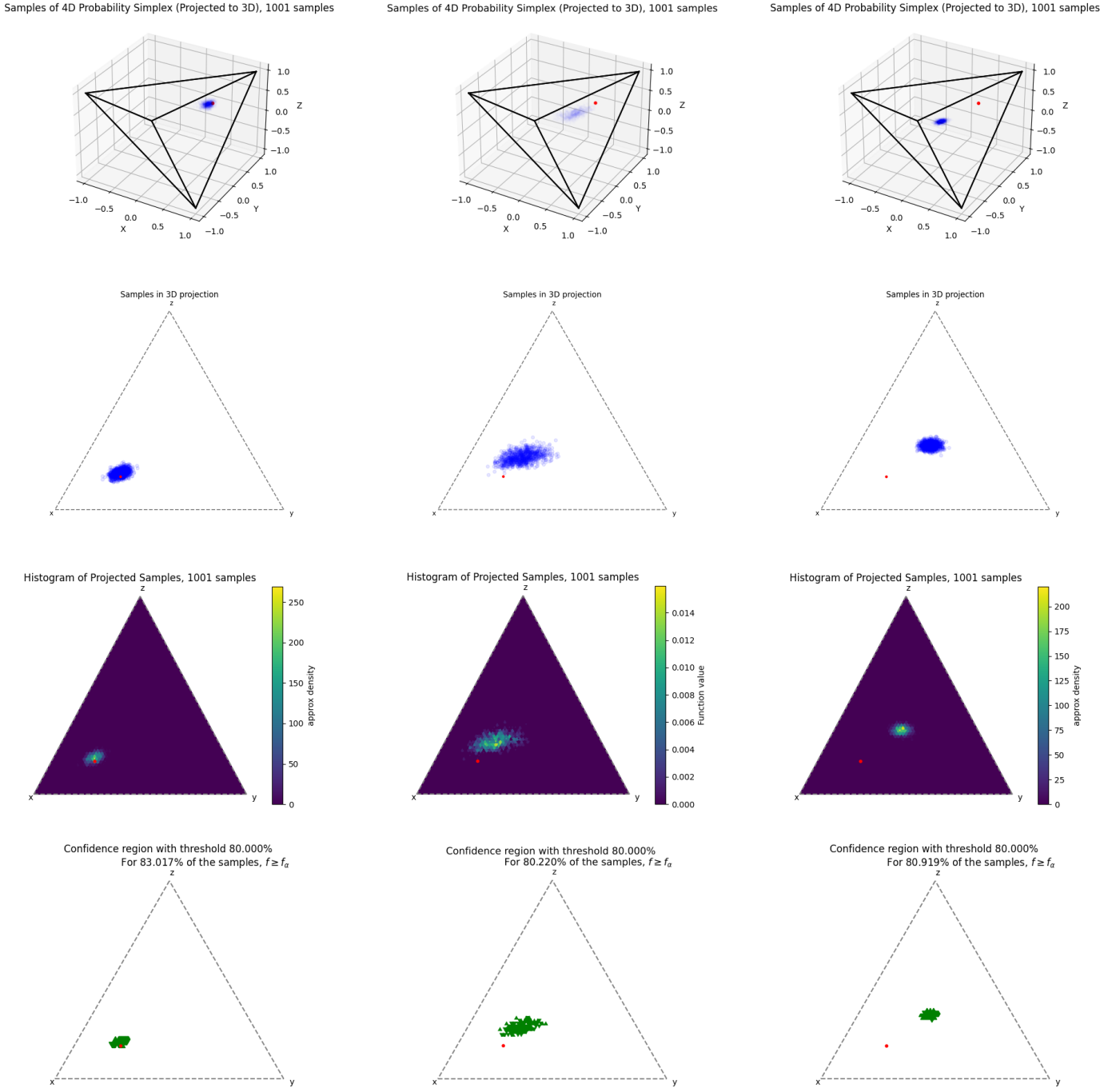


Figure 13: Comparison of one pixel cases for different (λ_0, λ_1) configurations. Each subfigure contains the 3D samples, 2D simplex projection, corresponding estimated histogram, and confidence region.

5.3 Performance of the different transformations for one pixel unmixing (without kernel)

Let's compare different transformations on randomly-generated proportions $\mathbf{a}$, and no kernel in the prior. Although we don't study unmixing on a full map with spatial dependance here, our simulations are crucial to understand convergence and overall parameter impact.

Estimation accuracy In the following plots, we confront the following sampling methods :

- Mirror Langevin with log transformation (with $\|c\| \sim \gamma(P,1)$ [3] and ilr-gaussian prior ($\text{ilr}(\mathbf{a}) \sim \mathcal{N}(0, I_P)$).
- Mirror Langevin with logit transformation and logit prior ($\text{logit}(\mathbf{a}) \sim \mathcal{N}(0, I_P)$)
- Mirror Langevin with clr transformation and clr prior ($\text{clr}(\mathbf{a}) \sim G_P\mathcal{N}(0, I_P)$, which is equivalent to using ILR prior)
- Mirror Langevin with logit transformation and clr prior
- Classic projected Langevin algorithm with a uniform prior on Δ^{P-1}

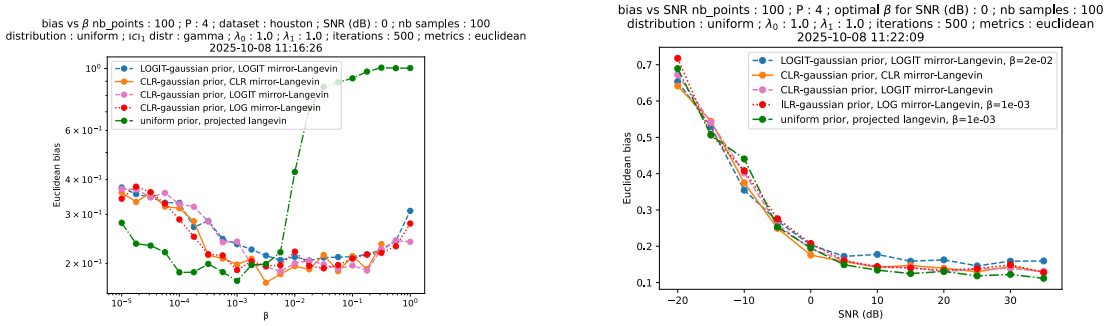
Our objective would be to show that mirror descent procedure performs as good as classic projected Langevin algorithm, and to evidence the fact that ILR transformation might be the best transformation for overall performance.

For a fixed S, we draw ground truth $\mathbf{a}_{\text{true}}$, and gather $n_s = 100$ samples $\hat{\mathbf{a}}^{(n_s)} := (\hat{\mathbf{a}}_l)_{1 \leq l \leq n_s}$. Euclidean distance bias $\|(\frac{1}{n_s}\sum_{l=1}^{n_s} \hat{\mathbf{a}}_l) - \mathbf{a}_{\text{true}}\|_2$ is computed as a metrics for the accuracy of the estimation. [4]

Average is done over 100 sampling algorithm realizations, for fixed endmembers. $SNR := \frac{\mathbb{E}(\|SA\|^2)}{\mathbb{E}(\|W\|^2)}$ is set to 0dB.

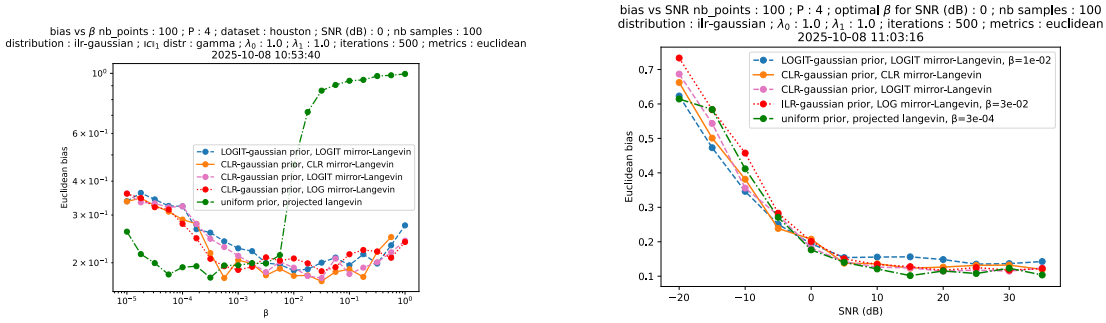
[3] with pdf $f(s) = \frac{\beta^P s^{P-1}}{\Gamma(P-1)} e^{-\beta s}$, $\Gamma(z) := \int_0^{+\infty} t^{z-1}e^{-t}dt$ being Euler's gamma function.

[4] An other way to characterize estimation accuracy, inspired by [20], would be to use the Aitchison distance : $b_\Delta(\hat{\mathbf{a}}^{(n_s)}, \mathbf{a}_{\text{true}}) := \left\|\text{Mcen}(\hat{\mathbf{a}}^{(n_s)}) - \mathbf{a}_{\text{true}}\right\|_A$, with $\text{Mcen}(\hat{\mathbf{a}}^{(n_s)}) := \text{argmin}_{\mathbf{a}\in\Delta^{P-1}} \frac{1}{n_s}\sum_{l=1}^{n_s} \|\hat{\mathbf{a}}_l - \mathbf{a}\|_a^2 = \text{ilr}^{-1}(\frac{1}{n_s}\sum_{l=1}^{n_s} \text{ilr}(\hat{\mathbf{a}}_l))$. This approach might reveal more suited to the geometry, but as these values are harder to interpret, we have chosen to use "euclidean" bias for our plots.



(a) Bias vs β **(b)** Bias vs SNR, fixed β

Figure 14: Truth $\mathbf{a}$ to estimate drawn from $\mathcal{U}(\Delta^{P-1})$



(a) Bias vs β **(b)** Bias vs SNR, fixed β

Figure 15: Truth $\mathbf{a}$ to estimate drawn from ilr-Gaussian distribution: $\text{ilr}(\mathbf{a}) \sim \mathcal{N}(0, I_{P-1})$

In the results above, we observe similar optimal performance for the different algorithms, which would have been expected : in general, mirror Langevin is not meant to converge faster to the right posterior distribution (in fact, Mirror Descent in optimization has similar convergence rates as classic gradient descent [8]).

All algorithms perform better than random (which would lead to a mean bias of 0.35 for uniformly generated truth proportions $\mathbf{a}$). Optimum performance is achieved for different stepsizes β, depending on the nature of the algorithm. Too big β makes the descent diverge, which is the case for $\beta \geq 1$ here. For β too small, the algorithms doesn't have the time to converge within 500 iterations.

On other studies, ILR-transform and CLR-transform mirror Langevin perform the same, which is reassuring as they are perfectly equivalent algorithms.

Both scenarios show that ILR mirror Langevin - ILR prior is advantaged, and LOGIT mirror Langevin - LOGIT prior disadvataged, probably because of the inherent assymetry of its prior.

We would expect uniform prior and projected langevin to be advantaged for matching real distribution generating **a**, but this is not clearly the case here. At lower SNR, where the prior might gain more importance in the posterior distribution, we might observe a slight advantage for matching prior (corresponding figures are not shown in this report)

In 15 we observe that CLR and LOGIT mirror Langevin perform similar with CLR prior, which was expected. Indeed, for same posterior potential U, the convexity of $\mathbf{y} \mapsto U(\nabla\varphi(\mathbf{y}))$, which plays in the convergence rate of the mirror langevin algorithm ([16]) is identical considering $\nabla\varphi = \text{logit}$ or $\nabla\varphi = \text{clr}$. Though, these algorithms converges for slightly different β ranges.

In other experiments, we observed that modifying the prior distribution on $S := \|C\|_1$ (see equation 6) did not significantly affect the convergence of the LOG algorithm.

Algorithm complexity As expected, LOGIT, ILR and LOG algorithms have a time complexity of about $\mathcal{O}(P)$, while ILR iterations require $\mathcal{O}(P^2)$ due to the multiplication by matrix H_P. That is true even with torch internal optimizations performed, which might play for small P here. The fact that LOGIT is consistently faster than CLR or LOG stems from its update steps being performed in the space $\mathbb{R}^{P-1}$, whereas CLR and LOG operate in the higher-dimensional space $\mathbb{R}^P$.

Performance - complexity compromise Overall, our results show that CLR-mirror Langevin with CLR prior is the best option for solving our unmixing problem.

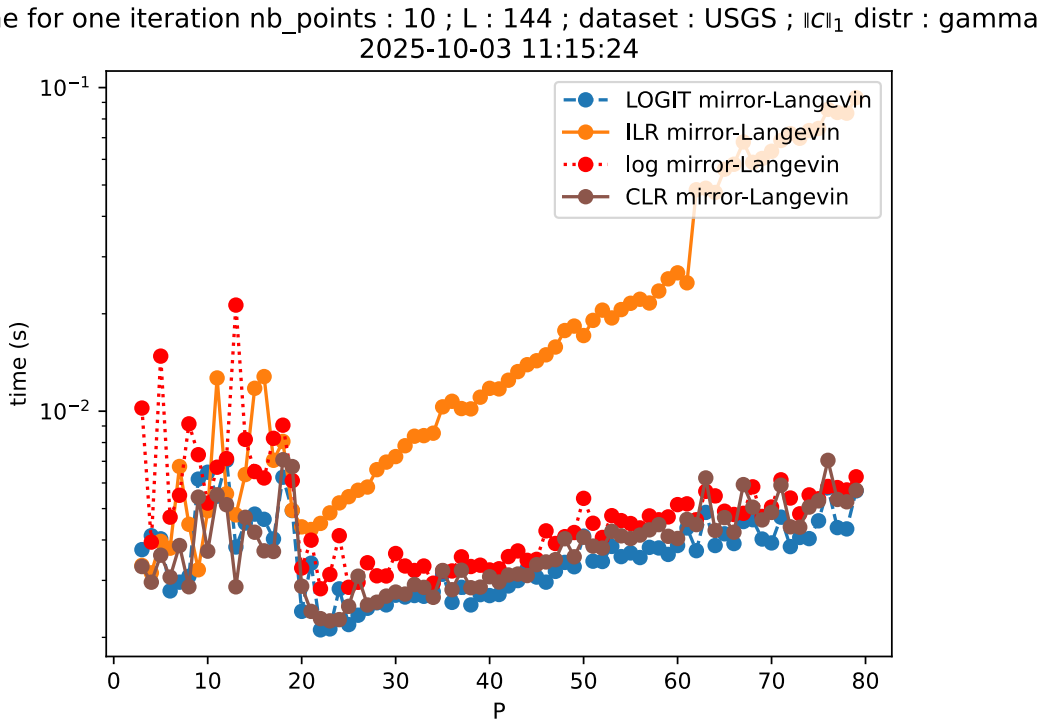


Figure 16: Execution time of the Mirror-Langevin procedure for increasing P

5.4 Experimental results for gaussian process priors

We now wish to exploit mirror-Langevin with our transforms for unmixing a full image, using our transformations to define gaussian processes on priors.

Here, keeping the same 50 x 50 image as in 10 with $N = 50 \times 50 = 250$ pixels, we put the prior that proportions $A = [\mathbf{a}_1, \dots \mathbf{a}_N], \mathbf{a}_1 \dots \mathbf{a}_N \in \Delta^{P-1}$ follow a CLR-Gaussian process :

$$\text{vec}(\text{clr}(\mathbf{a})) = \text{vec}([clr(\mathbf{a}_1) \dots clr(\mathbf{a}_N)]) \sim \mathcal{N}(0, K_N \otimes I_{P-1})$$

with $K_N = \begin{bmatrix} k(\mathbf{u}_1, \mathbf{u}_1) & \dots & k(\mathbf{u}_1, \mathbf{u}_N) \\ \vdots & & \vdots \\ k(\mathbf{u}_N, \mathbf{u}_1) & \dots & k(\mathbf{u}_N, \mathbf{u}_N) \end{bmatrix}$), using the exponential kernel

$k(\mathbf{u}, \mathbf{u}') = \frac{1}{\lambda_0} \exp(-\frac{1}{\sigma_k} \|\mathbf{u} - \mathbf{u}'\|_2)$, with the kernel lengthscale being $\sigma_k = 6 \simeq \frac{50}{8}$, expressed in numbers of pixels, roughly the size of an element in the scene. λ_0 is

the confidence in the prior.

In our tests, the classic Gaussian kernel seemed to diverge easily, maybe due to it's bigger covariance coefficients. That's why we chose the exponential kernel.

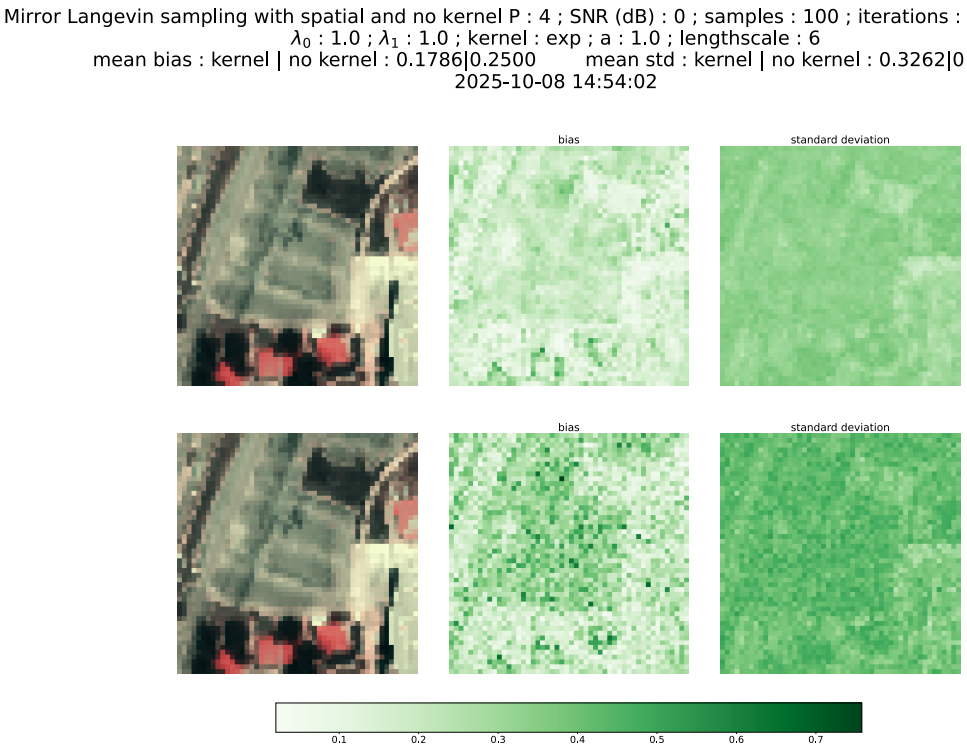


Figure 17: Spetral unmixing with CLR-Gaussian process, SNR $=$ 0dB. Top line : bias and variance using kernel ; Bottom line : bias and variance with no kernel ($K_N = I_N$)

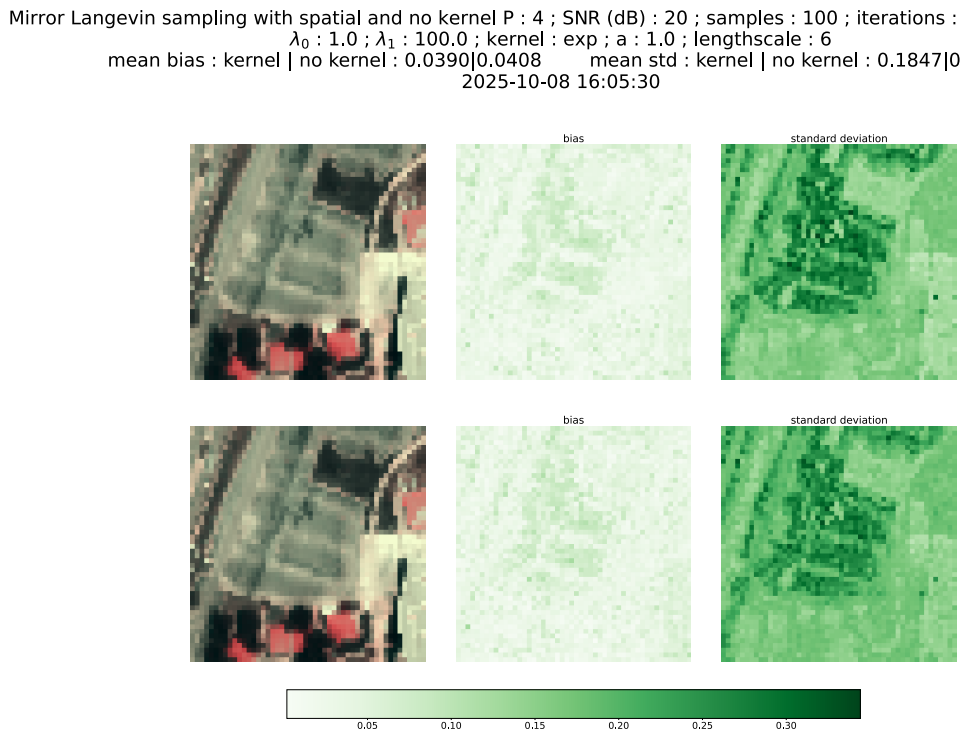


Figure 18: Spetral unmixing with CLR-Gaussian process, SNR $=$ 10dB. Top line : bias and variance using kernel ; Bottom line : bias and variance with no kernel ($K_N = I_N$)

In both figures 17 and 18, introducing spatial covariances in the kernel improves the bias. Indeed, using a Gaussian process as prior "smooths" the samples by encouraging compositions $\mathbf{a}$ to be similar to their neighbors.

As expected, the sampling benefits even more from this spatial model when noise power is higher (for SNR $=$ 0dB in figure 17) here : the average bias between samples and ground truth is $\simeq 0.09$ without kernel compared to $\simeq 0.06$ using a kernel !

Here, the environment shows a lot of discontinuities, and the spatial dependance might bring even more for continuously varying spectral scenes.

In 18, the large variance observed on the parking spaces might come from the fact that spectral signature of concrete and bitument might be similar.

6 Conclusion

Thoughout this research project, we have managed to develop and test a complete methodology to :

- Sample a posterior distribution of proportions from a hyperspectral image, using a transformation suited to the geometry of the proportions
- Take into account the very nature of our data observation, a 2D spectral image, using Gaussian processes for the prior.
- Build confidence intervals from these samples, allowing us to visualize the accuracy of sampling (and estimation)

Our main contribution has been to discuss transformation choices, for the Mirror-Langevin procedure. Our analysis and numerical study reveals an advantage for CLR-mirror Langevin with CLR prior, while ALR-transform being also possible with CLR prior.

In 4.3.4, we have also proposed an original method to sample efficiently from the whole composition space $(\mathbb{R}_+^*)^P$ with the logarithmic transformation. Further investigation is needed to determine whether this approach offers clear advantages.

Here are some open questions and unexplored this research project :

- Imagine an analytic way to build confidence intervals, without relying on samples and estimated marginal pdf.
- Perform all explained algorithms directly in the latent space. As shown in [21], peforming mirror descent is always equivalent to performing Riemannian gradient descent in the latent space, with its new Riemannian metrics.
- Further explore the dependence of the prior on $s = \frac{\mathbf{c}}{\|c\|_1}$ in our original method .
- Which choice should be made for the kernel, is there Is there a way to "learn" kernels, adapted to different scales ?

7 Hindsight on my research internship

I will now develop some reflections on my research experience, illustrating them with tangible examples.

7.1 Scientific methodology : the work of researcher

As a research intern, like every researcher :

1. I need to perform a state of the art study to ensure that the problem I'm trying to solve is useful and hasn't been solved before (or proven non solvable). Even though I could trust my supervisor that the problem was relevant, I still decided to try to do some part on my own. It was challenging, as I had to make sure the problem was useful and hadn't already been solved (or proven impossible).

2. I had to imagine models and methods to solve this problem, sometimes requiring to read manuals from the literature to learn about some mathematical tools.

3. I need to test models via numerical realizations, the objective being to have results matching what was expected from the theory. The purely-engineering competence is crucial in this part : even in a research environment, building clear code and design simulation process is essential. When badly performed, simulation step can make the researcher loose a lot of time.

4. I had to present models and results to my supervisors. In my case, we set up weekly meetings with my tutors for showing my work. For this part, the communication is key as it is important to go straight to the point for a more constructive meeting with other researchers. Project presentation during classes at CentraleSupélec revealed extremely useful for grasping this issue. I also understood that one should always prepare slides for a more structured talk.

5. It was also necessary to format these results in a final deliverable (which can be a report like the one you are reading, a paper, presentation ...). Though, I learned that form matters throughout the entire research process, so I made sure to keep well-structured slides or parts of the report updated

every week, rather than just informal notes or drafts. This might have prevented me from going off track.

7.2 The role of the supervisor in a research environment

Even if, as I've realized during this experience, the research work can seem solitary, the contact with the supervisors is of utmost importance.

The role of my main supervisor is to realign my research with the overall problematics. With it's knowledge of the field of remote sensing, he can bring :

- Before everything, provide me for the problematics and first methods to solve it.
- Provide basis knowledge in its own field of expertise. My supervisor allowed me to attend a class he was teaching, about sampling methods. This allowed me to get a structured introduction on this topic, which was completely new for me. Without this proper introduction, learning these concepts by only reading references would likely have been much more laborious and would have made me focus only on specific points.
- Bring new methods to the discussions throughout the whole internship. During my project at IMT Atlantique, my supervisor suggested me to investigate the LOGIT, CLR and ILR transformations, based on an article he had found. These new concepts revealed crucial for the problem.
- Provide me the good metrics to use to evaluate the performance of methods, and preferably confront it to the existing literature. In my case, I was suggested to compare the mirror descent to classic mirror Langevin. Achieving similar, if not better, performance for my algorithms and classic mirror Langevin shows quite a consistency in the results. It is always important to confront a new method to already existing ones. Though, it would have been interesting in this project for instance to compare the estimation accuracy from the sampling to existing unmixing campaigns.
- Bring a deeper analysis of the experimental results. For instance, in my project, with its expertise, my tutor could comment my curves and help me understand the results : whether it is consistent or if there there might be code errors.

7.3 Step back

Overall, I've learned a lot from this experience as part of IMT Atlantique. Even if the finality of the work of the engineer and the researcher are different, their respective tools prove very useful in each of the two contexts, and these two occupations are not opposed. I applied many of the skills gained at CentraleSupélec, both in theory, methodology and programming tools.

I understand that as a future engineer, and maybe researcher, experience is key : it's only though experiences, projects, internships, and later full-time position that one can learn the best way to be an engineer or researcher.

A Formula sheet

Determinant and inverse of a perturbated matrix

Determinant of a pertubated matrix The following determinant formula ([22], p.201) can be proven using Weinstein–Aronszajn identity. It specifies how the determinant changes when updating a matrix $A \in \mathbb{R}^{n\times n}$ by a low rank matrix UCV^T where $C \in \mathbb{R}^{m\times m}, U, V \in \mathbb{R}^{n\times m}, m \leq n$:

$$\det(A + UCV^T) = \det(A)\det(C)\det(C^{-1} + V^T A^{-1} U) \tag{21}$$

Woodbury identity The following matrix formula can prove useful when trying to compute the inverse of the low-rank matrix $A + UCV$, where $C \in \mathbb{R}^{m\times m}, U, V \in \mathbb{R}^{n\times m}, m \leq n$:

$$(A + UCV^T)^{-1} = A^{-1} - A^{-1}U(C^{-1} + V^T A^{-1} U)^{-1} V^T A^{-1} \tag{22}$$

Application : useful determinant and inverse

For any complex numbers $d_1, \dots d_P$ and x, let $M = \begin{bmatrix} x+d_1 & x & \dots & x \\ x & x+d_2 & \dots & x \\ \vdots & & \ddots & \vdots \\ x & \dots & x & x+d_P \end{bmatrix}$.

Using 21 in the case $d_1, \dots d_P \neq 0$:

$$det(M) = [\sum_{k=1}^{P} (\prod_{l\neq k} d_l)]x + \prod_{k=1}^{P} d_k = (\prod_{k=1}^{P} d_k)[1 + \sum_{k=1}^{P} \frac{x}{d_k}] \tag{23}$$

where the last equality is valid only when $d_1, \dots d_P \neq 0$.

Using 22, if $d_1, \dots, d_P \neq 0$ the inverse of M can be expressed as

$$
\begin{aligned}
M^{-1} &= \mathrm{Diag}\left(\frac{1}{d_1},\dots,\frac{1}{d_P}\right) - \frac{x}{1+\sum_{k=1}^{P}\frac{x}{d_k}}\begin{bmatrix}\frac{1}{d_1}\\ \vdots \\ \frac{1}{d_P}\end{bmatrix}\begin{bmatrix}\frac{1}{d_1} & \cdots & \frac{1}{d_P}\end{bmatrix} \\
&= \left[\frac{\delta_{ij}}{d_i} - \frac{x}{1+\sum_{k=1}^{P}\frac{x}{d_k}}\frac{1}{d_i d_j}\right]_{1\le i,j\le P}.
\end{aligned} \tag{24}
$$

B The simplex Riemannian submanifold

Gradient and Hessian in the simplex Riemannian submanifold

$(\Delta_{P-1};\langle .;.\rangle)$ is a submanifold of the euclidean space $(\mathbb{R}_+^{*\,P};\langle .;.\rangle)$.

Tangent spaces at point $\mathbf{a}\in\Delta^{P-1}$ is $T_a = 1^{\perp} = \{x\in\mathbb{R}^P | 1^T x = 0\}$.

Assume f accepts a smooth extension to an open set of $\mathbb{R}^P$ $U\supset\Delta^{P-1}$ (e.g. $U=(\mathbb{R}_+^*)^P$)) . The riemannien gradient of $f:\Delta^{P-1}\to\mathbb{R}$ is then :

$$
\mathrm{grad}_{\Delta} f = P_{T_a}(\nabla\bar{f}) = G_P\nabla\bar{f} \tag{25}
$$

where $\nabla\bar{f}$ its classic euclidean gradient of $\bar{f}$.

$\mathbf{a}\mapsto\mathrm{grad}_{\Delta}f(\mathbf{a})$ admits a smooth extension $\overline{\mathrm{grad}_{\Delta}f}$ in the open set $\mathbb{R}_+^{*\,P}$, so the Hessian of $f:\Delta^{P-1}\to\mathbb{R}$ is the linear map (see [6], p. 96, Corollary 5.16.) :

$$
\begin{cases} T_a \to T_a \\ \mathbf{u}\mapsto H(\mathbf{u}) = P_{T_a}(D\overline{\mathrm{grad}_{\Delta}f}(\mathbf{a})\mathbf{u}) \end{cases}
$$

where D is the differentiation operator. The hessian operator can be simplified as :

$$
H_a(\mathbf{u}) = G_P(D(G_P\nabla\bar{f})(\mathbf{a})\mathbf{u}) = G_P(G_P D(\nabla\bar{f})(\mathbf{a})\mathbf{u} = G_P D\nabla\bar{f}(\mathbf{a})G_P\mathbf{u} \tag{26}
$$

as $G_P\mathbf{u}=\mathbf{u}\in\mathbf{1}^{\perp}$.

The Hessian matrix can be written as :

$$
\mathrm{Hess}_{\Delta}f(\mathbf{a}) = G_P\nabla^2\bar{f}(\mathbf{a})G_P^T
$$

Expressed in a natural basis for $\mathbf{1}^\perp$ (whose representant matrix is H_P), the hessian becomes :

$$H(\mathbf{a}) = H_P G_P [\nabla^2 \bar{f}(\mathbf{a})] G_P^T H_P^T = H_P [\nabla^2 \bar{f}(\mathbf{a})] H_P^T$$

Convexity in Riemannian manifolds

Let's first extend some convexity concepts on manifolds, allowing us to characterize global minima and later to adapt Mirror descent algorithm to the simplex geometry.

> **Definition** *Geodesically convex set ([6], Definition 11.2)*
> A subset C of a Riemannian manifold $\mathcal{M}$ is geodesically convex if, for every $x, y \in C$, there exists a geodesic segment $c : [0,1] \to \mathcal{M}$ such that $c(0) = x, c(1) = y$ and $\forall t \in [0,1], c(t) \in C$.

Δ^{P-1} is geodesically convex : its geodesics are the segment functions $c_{\mathbf{a},\mathbf{b}} : t \in [0;1] \mapsto (1-t)\mathbf{a} + t\mathbf{b}$, which are all in Δ^{P-1} (classical convexity).

> **Definition** *Geodesically convex function ([6], Definition 11.3)*
> A function $f : C \subset \mathcal{M} \to \mathbb{R}$ is geodesically convex if C is geodesically convex and if $f \circ c : [0,1] \mapsto \mathbb{R}$ is (strictly) convex for each geodesic segment $c : [0,1] \to \mathcal{M}$ whose image is in C (with $c(0) \neq c(1)$).

> **Property B.1** *(Local minimizers are global minimizers ([6], Corollary 11.22))*
> Let $\mathcal{M}$ be a convex and opened manifold. Let $f : \mathcal{M} \to \mathbb{R}$ be a strictly differentiable convex function on this manifold. Then :
>
> $$a \in \mathcal{M} \text{ is a global minimizer of } f \text{ if and only if } \operatorname{grad}_C f(a) = 0$$

This property closely mirrors the situation in classical optimization over Euclidean spaces. In general, when the manifold is not an open set, the forward implication holds only if a a lies in the interior of the domain. However, the backward implication always remains valid. Both implications can be established using first-order optimality conditions.

Actually, convexity within simplex Riemannian submanifolds can be easily

established once a smooth extension is known :

Property B.2
Let f be a function from $\Delta^{P-1} \to \mathbb{R}$. If f admits a smooth extension $\bar{f}$, (strictly) convex in $(\mathbb{R}_+^*)^P$, then $f : \Delta^{P-1} \to \mathbb{R}$ is (strictly) geodesically convex

Proof. Assume there exists a smooth convex extension of f noted $\bar{f} : {\mathbb{R}_+^*}^P \to \mathbb{R}$. On Δ^{P-1}, geodesics of are exactly the functions $c_{a,b} : t \in [0;1] \mapsto (1-t)a + tb$ for $a, b \in \Delta^{P-1}$ (admitted without proof). fixing a and b, we have : $\forall t \in [0;1], \forall u, v \in [0;1]$

$$\begin{aligned} f \circ c_{a,b}((1-t)u + tv) &= \bar{f} \circ c_{a,b}((1-t)u + tv) \\ &= \bar{f}\big([1 - ((1-t)u + tv)]a + [(1-t)u + tv]b\big) \\ &= \bar{f}((1-t)[(1-u)a + ub] + t[(1-v)a + vb]) \\ &\leq (1-t)\bar{f}((1-u)a + ub) + t\bar{f}((1-v)a + vb) \end{aligned}$$

(From the convexity of $\bar{f} : {\mathbb{R}_+^*}^P \to \mathbb{R}$)

$$\begin{aligned} &= (1-t)(\bar{f} \circ c_{a,b})(u) + t(\bar{f} \circ c_{a,b})(v) \\ &= (1-t)(f \circ c_{a,b})(u) + t(f \circ c_{a,b})(v) \end{aligned}$$

and $f \circ c_{a,b}$ is convex on $[0;1]$.

The demonstration adapts with strictly convex extension $\bar{f}$ by changinf $\leq$ by $<$ in the inequality. □

For φ a stricly convex function on Δ^{P-1}, let's define the Bergman divergence :

$$\begin{aligned} \forall \mathbf{a}, \mathbf{b} \in \Delta^{P-1}, B_\varphi(\mathbf{a}, \mathbf{b}) &:= \varphi(\mathbf{a}) - \varphi(\mathbf{b}) - \langle \mathrm{grad}_\Delta \varphi(\mathbf{b}); \mathbf{a} - \mathbf{b} \rangle \\ &= \bar{\varphi}(\mathbf{a}) - \bar{\varphi}(\mathbf{b}) - \langle G_P \nabla \bar{\varphi}(\mathbf{b}); \mathbf{a} - \mathbf{b} \rangle \\ &= \bar{\varphi}(\mathbf{a}) - \bar{\varphi}(\mathbf{b}) - \nabla \bar{\varphi}(\mathbf{b})^T G_P(\mathbf{a} - \mathbf{b}) \\ &= \bar{\varphi}(\mathbf{a}) - \bar{\varphi}(\mathbf{b}) - \nabla \bar{\varphi}(\mathbf{b})^T(\mathbf{a} - \mathbf{b}) (\geq 0) \end{aligned}$$

using the fact that the scalar product here is the Euclidean scalar product $\langle .;. \rangle$.

Mirror Descent in the simplex Riemannian submanifold

Let $\varphi : C \to \mathbb{R}$ be a twice-differenciable convex function such that grad_Δ is bijective from C to $\mathbb{R}^n$.

We can then define the Bergman divergence B_h and $\forall \mathbf{a}_t \in \Delta^{P-1}, \mathbf{a} \mapsto B_h(\mathbf{a}, \mathbf{a}_t)$ is stricly geodesically convex and so is the expression $\mathbf{a} \mapsto \langle \mathbf{a}, \text{grad}_\Delta f(\mathbf{a}_t) \rangle + \frac{1}{\beta_t} B_h(\mathbf{a}, \mathbf{a}_t)$ is convex. Then, by property the following optimization problem admits at most one solution on Δ^{P-1} :

$$\arg\min_{\mathbf{a} \in \Delta^{P-1}} \langle \mathbf{a}, \text{grad}_\Delta f(\mathbf{a}_t) \rangle + \frac{1}{\beta_t} B_\varphi(\mathbf{a}, \mathbf{a}_t) \tag{27}$$

We have :

$$\begin{aligned} &\arg\min_{\mathbf{a} \in \Delta^{P-1}} \langle \mathbf{a}, \text{grad}_\Delta f(\mathbf{a}_t) \rangle + \frac{1}{\beta_t} B_\varphi(\mathbf{a}, \mathbf{a}_t) \\ = &\arg\min_{\mathbf{a} \in \Delta^{P-1}} \langle \mathbf{a}, \text{grad}_\Delta f(\mathbf{a}_t) \rangle + \frac{1}{\beta_t} (\varphi(\mathbf{a}) - \langle \text{grad}_\Delta(\mathbf{a}_t); \mathbf{a} \rangle) \end{aligned}$$

By property B.1, a solution to this optimization problem if and only if the following gradient is set to 0 (note that by strict convexity of f there will be at most one minimizer):

$$\mathbf{a} + \frac{1}{\beta_t} (\text{grad}_\Delta \varphi(\mathbf{a}) - \text{grad}_\Delta \varphi(\mathbf{a}_k))$$

A solution $\mathbf{a}^*$ of the one step minimization problem would lead to a null gradient :

$$\begin{aligned} &\text{grad}_\Delta f(\mathbf{a}_t) + \frac{1}{\beta_t} (\text{grad}_\Delta \varphi(\mathbf{a}^*) - \text{grad}_\Delta \varphi(\mathbf{a}_t)) = 0 \\ &\Leftrightarrow \text{grad}_\Delta \varphi(\mathbf{a}^*) = \text{grad}_\Delta \varphi(\mathbf{a}_t) - \beta_t \text{grad}_\Delta f(\mathbf{a}_t) \\ &\Leftrightarrow \mathbf{a}^* = \text{grad}_\Delta \varphi^* [\text{grad}_\Delta \varphi(\mathbf{a}_t) - \beta_t \text{grad}_\Delta f(\mathbf{a}_t)] \end{aligned}$$

with the later expression being possible thanks to $\text{grad}_\Delta \varphi$ being a bijective transformation.

That achieves proving that the mirror descent step

$$\mathbf{a}_{t+1} := \arg\min_{\mathbf{a} \in \Delta^{P-1}} \langle \mathbf{a}, \text{grad}_\Delta f(\mathbf{a}_t) \rangle + \frac{1}{\beta_t} B_\varphi(\mathbf{a}, \mathbf{a}_t)$$

is consistent, and gives us the classical expression of the mirror descent step, with riemannian gradients (4) .

Entropy map The entropy map h is geodesically convex on the Riemannian submanifold Δ^{P-1}, since its extension

$$\bar{h} : \begin{cases} \Delta^{P-1} \to \mathbb{R} \\ \mathbf{a} \mapsto \sum_{k=1}^{P} a_k \log a_k - \sum_{k=1}^{P} a_k \end{cases} \quad \text{is stricly convex.}$$

Furthermore, since ∇h, the centered log-ratio (clr) transformation

$$\text{clr} : \Delta^{P-1} \to \mathbf{1}^\top \subset \mathbb{R}^P,$$

is a bijection, one can perform mirror descent on the simplex using this transformation.

Extension to sampling

Previous development can be used to extend mirror langevin to the simplex Riemannian manifold.

References

[1] J. AITCHISON : The statistical analysis of compositional data. *Journal of the Royal Statistical Society: Series B (Methodological)*, 44(2):139–160, 1982.

[2] Sanjeev ARORA, Elad HAZAN et Satyen KALE : The multiplicative weights update method: a meta-algorithm and applications. *Theory of computing*, 8(1):121–164, 2012.

[3] Anuja BHARGAVA, Ashish SACHDEVA, Kulbhushan SHARMA, Mohammed H. ALSHARIF, Peerapong UTHANSAKUL et Monthippa UTHANSAKUL : Hyperspectral imaging and its applications: A review. *Heliyon*, 10(12):e33208, 2024.

[4] José M. BIOUCAS-DIAS, Antonio PLAZA, Nicolas DOBIGEON, Mario PARENTE, Qian DU, Paul GADER et Jocelyn CHANUSSOT : Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(2):354–379, avril 2012.

[5] William M. BOLSTAD et James M. CURRAN : *Introduction to Bayesian Statistics*. John Wiley & Sons, 3rd édition, 2016. Chapter 1.4.

[6] Nicolas BOUMAL : *An Introduction to Optimization on Smooth Manifolds*. Cambridge University Press, 1 édition, mars 2023.

[7] G.E.P. BOX et G.C. TIAO : *Bayesian Inference in Statistical Analysis*. Wiley Classics Library. Wiley, 2011. Good overview of Bayesian credible intervals (cumulative vs Highest Probability Density (HPD) based), with an original method to compute efficiently Highest Probability Density regions.

[8] Sébastien BUBECK : Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8:231–357, janvier 2015.

[9] Lucia CLAROTTO, Denis ALLARD et Alessandra MENAFOGLIO : A new class of α -transformations for the spatial analysis of compositional data. *Spatial Statistics*, 47:100570, mars 2022.

[10] Nicolas DOBIGEON, Said MOUSSAOUI, Margot COULON, Jean-Yves TOURNERET et Alfred HERO : Joint bayesian endmember extraction and linear unmixing for hyperspectral imagery. *IEEE Transactions on Signal Processing*, 57:4355–4368, 2009.

[11] Olivier ECHES, Nicolas DOBIGEON et Jean-Yves TOURNERET : Enhancing hyperspectral image unmixing with spatial correlations. *IEEE Transactions on Geoscience and Remote Sensing*, 49(11):4239–4247, novembre 2011.

[12] J. J. EGOZCUE et V. PAWLOWSKY-GLAHN : Simplicial geometry for compositional data. *Geological Society, London, Special Publications*, 264(1):145–159, janvier 2006.

[13] J J EGOZCUE, V PAWLOWSKY-GLAHN, G MATEU-FIGUERAS et C BARCELO-VIDAL : Isometric logratio transformations for compositional data analysis.

[14] Olivier FERCOQ, Pascal BIANCHI et Anne SABOURIN : Convex analysis. Technical report / course notes, Institut Mines-Télécom / Télécom-ParisTech, CNRS LTCI.

[15] A.K. GUPTA et D.K. NAGAR : *Matrix Variate Distributions*. Monographs and Surveys in Pure and Applied Mathematics. Taylor & Francis, 1999.

[16] Ya-Ping HSIEH, Ali KAVIS, Paul ROLLAND et Volkan CEVHER : Mirrored langevin dynamics. (arXiv:1802.10174), décembre 2020. arXiv:1802.10174 [cs].

[17] Rob J. HYNDMAN : Computing and Graphing Highest Density Regions. *The American Statistician*, 50(2):120–126, 1996. Publisher: [American Statistical Association, Taylor & Francis, Ltd.].

[18] Orhan KESEMEN, Buğra Kaan TI˙RYAKI˙, Eda ÖZKUL et Özge TEZEL : Determination of the confidence intervals for multimodal probability density functions. *Gazi university journal of science*, 2017.

[19] Arkadij Semenovič NEMIROVSKIJ et David Borisovich YUDIN : Problem complexity and method efficiency in optimization. 1983.

[20] V. PAWLOWSKY-GLAHN et J. J. EGOZCUE : Geometric approach to statistical analysis on the simplex. *Stochastic Environmental Research and Risk Assessment*, 15(5):384–398, octobre 2001.

[21] Garvesh RASKUTTI et Sayan MUKHERJEE : The information geometry of mirror descent. *IEEE Transactions on Information Theory*, 61(3):1451–1457, mars 2015.

[22] Carl Edward RASMUSSEN et Christopher K. I. WILLIAMS : *Gaussian processes for machine learning*, volume 2. MIT Press, United States, 2006.

[23] Margaret WRIGHT : The interior-point revolution in optimization: History, recent developments, and lasting consequences. *Bulletin of the American Mathematical Society*, 42(1):39–56, 2004.