Gaussian Markov random fields for MRF sampling

(and inference)

Jean-Baptiste Courbot[1] & Hugo Gangloff[2]

July 3, 2025

[1] IRIMAS - UR 7499 – Université de Haute-Alsace, Mulhouse, France
[2] Université Paris-Saclay, AgroParisTech, INRAE, UMR MIA Paris-Saclay, Palaiseau, France

I. Introduction: MRF, GMRF

II. Sampling

II.1 Method

II.2 Some results

III. Inference

IV. Next steps

Introduction: MRF, GMRF

MRF in image processing

Let $\mathcal{S}$ be the lattice of n sites in an image. $\mathbf{X} = \{X_s\}_{s\in\mathcal{S}}$ is a MRF if and only if, $\forall s \in \mathcal{S}$:

$$p(x_s | \mathbf{x}_{\mathcal{S}\setminus s}) = p(x_s | \mathbf{x}_{N_s})$$

where N_s denotes the neighborhood of the s site.

The Hammersley-Clifford theorem[1] allows to write,considering only pairwise site interactions ψ:

$$p(\mathbf{x}) = \frac{1}{\gamma} \exp\left(-\sum_{s\in\mathcal{S}} \sum_{s'\in N_s} \psi(x_s, x_{s'}) \right).$$

Computing $\gamma > 0$ is intractable in general, so realizations $\mathbf{X} = \mathbf{x}$ are obtained through iterative sampling techniques, such as Gibbs sampling[2].

[1] Clifford and Hammersley, "Markov fields on finite graphs and lattices", 1971.

[2] Geman and Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images", 1984.

GRF and GMRF

$\mathbf{Z}$ is a Gaussian random field on $\mathcal{S}$ iff $\mathbf{Z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, with $\boldsymbol{\mu} \in \mathbb{R}^n$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times n}$ a covariance matrix.

Upon known conditions on $\boldsymbol{\Sigma}$[3], GRF are also Markovian, and are then referred to as Gaussian Markov random fields (GMRF). Depending on $\boldsymbol{\Sigma}$, GMRF can be sampled with:

- Cholesky decomposition: $\mathcal{S}$ is small.
- Fourier sampling: $\mathcal{S}$ is a torus and $\boldsymbol{\Sigma}$ is circulant.
- Spectral sampling: $\boldsymbol{\Sigma}$ belongs to an extended Gneiting class of covariances[4].

[3] Rozanov, *Markov random fields*, 1982, p. 120.

[4] Allard et al., "Simulating space-time random fields with nonseparable Gneiting-type covariance functions", 2020.

Our proposition

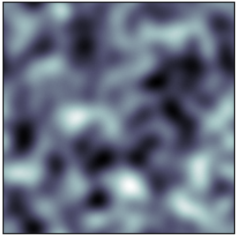
Sampling a GMRF in $\mathbb{R}^n$ is computationally efficient:

- We propose to use GMRF as proxies for discrete fields in Ω^n
- What are the properties of the resulting field?
- How can this help for inference?

Sampling

The initial idea

'If you look at a thresholded GMRF, it looks like a MRF"



(a) $\mathbf{Z} = \mathbf{z}$, GMRF real.

(b) $\{z_s \geq 0\}_{s \in \mathcal{S}}$

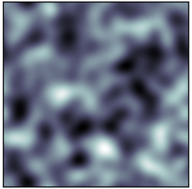
(c) MRF real.

Moving to $K > 2$ classes

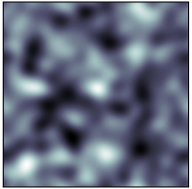
How to "threshold" to have $K > 2$ classes, ensuring class balance ?

- we need Z_s to lie in a higher dimension
- Design $\mathbf{Z}$ as a $(K-1)-$valued GMRF, such that now $\mathbf{z} \in \mathbb{R}^{(K-1)n}$, and at each site s $\mathbf{z}_s \in \mathbb{R}^{K-1}$
- Example in the $K = 3$ classes cases

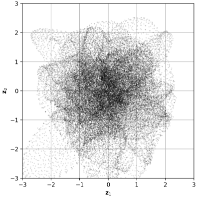
Thus, in dimension $K-1$ we can split into K balanced classes.



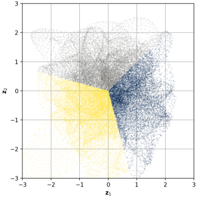
(a) $\mathbf{z}_1$, GMRF real.



(b) $\mathbf{z}_2$, GMRF real.



(c) location of each $\mathbf{z}_s$ in $\mathbb{R}^2$



(d) Splitting in 3 classes

Formalizing (I)

We need to formalize:

- the splitting within $\mathbb{R}^{K-1}$
- a proper writing of $\mathbf{X}$ given $\mathbf{Z}$

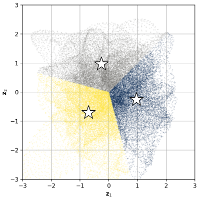


Fig. 1: Unit 2-simplex and splitting

Definition (Unit simplex)

A unit P-simplex is a regular simplex belonging in $\mathbb{R}^P$, whose $P+1$ vertices lie on a unit sphere. From[5] we take its vertices $\mathbf{v} \in \mathbb{R}^P$ as $\mathbf{v}_{P+1} = \frac{-1}{\sqrt{P}}\mathbf{1}$ and :

$$\mathbf{v}_j = \sqrt{\frac{P+1}{P}}\mathbf{e}_j - \frac{\sqrt{P+1}-1}{P\sqrt{P}}\mathbf{1} \ \forall 1 \leq j \leq P$$

with $\mathbf{1} \in \mathbb{R}^P$ a vector of ones and $\mathbf{e}_j \in \mathbb{R}^P$ the $j-$th basis vector.

[5] Anderson and Thron, "Coordinate Permutation-Invariant Unit N-Simplexes in N dimensions", 2021.

Formalizing (II)

Definition (Gaussian Unit-simplex Markov random field (GUM))

Let K be the number of classes to sample from, and $\mathbf{U}_{K-1}$ the K vertices of a unit $(K-1)$-simplex. Let also $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ be a GMRF taking values in $\mathbb{R}^{n(K-1)}$, such that $\mathbf{Z} = \{\mathbf{Z}_s\}_{s\in\mathcal{S}}$ and $\mathbf{Z}_s$ takes values in $\mathbb{R}^{K-1}$.

We define $\phi_{K,c} \colon \mathbb{R}^{n(K-1)} \mapsto \mathbb{R}^n$ such that:

$$\phi_{K,c}(\mathbf{Z}) = \sum_{i=1}^{K} \omega_i \pi_i^c(\mathbf{Z})$$

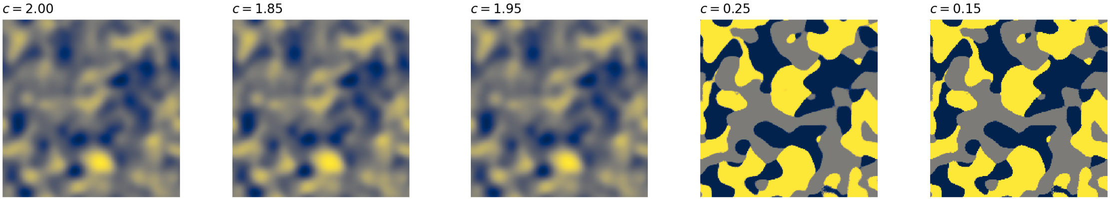
π_i^c indicates, site-wise, the distance between $\mathbf{Z}$ and the $i-$th vertices $\mathbf{v}_i$ of $\mathbf{U}_{K-1}$, such that $\forall s \in \mathcal{S}$:

$$\pi_i^c(\mathbf{Z}_s) = \frac{\exp(-c^{-2}\|\mathbf{Z}_s - \mathbf{v}_i\|^2)}{\sum_{k=1}^{K} \exp(-c^{-2}\|\mathbf{Z}_s - \mathbf{v}_k\|^2)}$$

with $c > 0$, $\mathbf{v}_k, \mathbf{v}_i \in \mathbf{U}_{K-1}$ unit simplex vertices, and $\omega_i \in \Omega \subset \mathbb{N}$. $\phi_{K,c}(\mathbf{Z})$ is named a *GUM random field*.

GUM : illustration

A look at the values taken by $\phi_{K,c}(z)$:



(a) (b) (c) (d) (e)

Property 1: $\mathbf{Z}$ being a GMRF, then $\phi_{K,c}(\mathbf{Z})$ is also a Markov field.

Property 2: when $c \to 0$, $\phi_{K,c}(\mathbf{Z})$ get close to a mixture of Dirac masses:

$$\phi_{K,c}(\mathbf{Z}) \underset{c \to 0}{\longrightarrow} \sum_{i=1}^{K} \omega_i \delta_{\left[\|\mathbf{Z}-\mathbf{v}_i\|_2 \leq \|\mathbf{Z}-\mathbf{v}_k\|_2,\ \forall \mathbf{v}_k \in \mathbf{U}_{K-1}\right]}$$

Property 2: when $c \to 0$, $\phi_{K,c}(\mathbf{Z})$ get close to a mixture of Dirac masses:

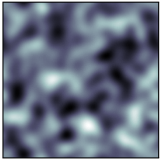
$$\phi_{K,c}(\mathbf{Z}) \xrightarrow[c \to 0]{} \sum_{i=1}^{K} \omega_i \delta_{\left[\|\mathbf{Z}-\mathbf{v}_i\|_2 \leq \|\mathbf{Z}-\mathbf{v}_k\|_2,\ \forall \mathbf{v}_k \in \mathbf{U}_{K-1}\right]}$$

Rewording $\lim_{c \to 0} \phi_{K,c}(\mathbf{Z}) = \mathbf{X} = \{X_s\}_{s \in \mathcal{S}}$, we have $\forall s \in \mathcal{S}$:

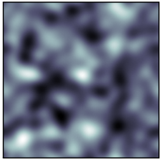
$$X_s = \omega_{k^*} \text{ with } k^* \text{ chosen such that } \mathbf{v}_{k^*} = \arg\min_{\mathbf{v} \in \mathbf{U}_{K-1}} \|\mathbf{Z}_s - \mathbf{v}\|_2$$

Some insights

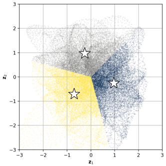
- $\mathbf{X}$ *should* be Markovian too
- related to a multivariate sigmoid and / or a transformed gaussian random field



(a) $\mathbf{z}_1$



(b) $\mathbf{z}_2$



(c) Location & labels

(d) $\mathbf{x}$

Numerical results

Implementation: Python & JAX, in CPU and GPU. Available on Github:
`https://github.com/HGangloff/mrfx`

We want to assess:

- sampling speed
- statistical properties

We evaluate:

- DGUMS based on Fourier GMRF sampling[6]
- and on spectral GMRF sampling[7]
- MRF with *chromatic* Gibbs sampling[8]

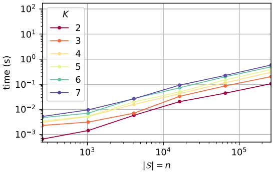
---

[6] Rue and Held, *Gaussian Markov random fields: theory and applications*, 2005.

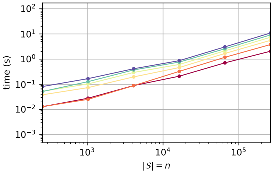
[7] Allard et al., "Simulating space-time random fields with nonseparable Gneiting-type covariance functions", 2020.

[8] Gonzalez et al., "Parallel Gibbs sampling: From colored fields to thin junction trees", 2011.

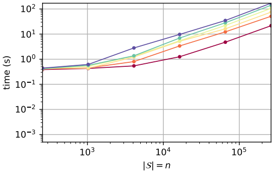
Numerical results: sampling speed



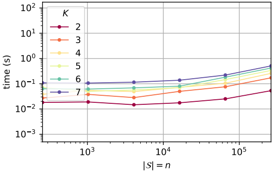
(a) CPU: Fourier sampling



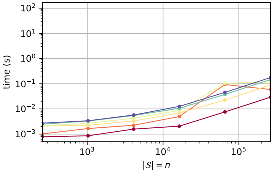
(b) Spectral sampling



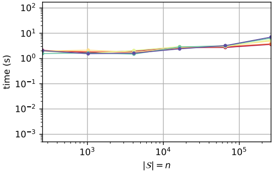
(c) Gibbs sampling



(d) GPU: Fourier sampling



(e) Spectral sampling



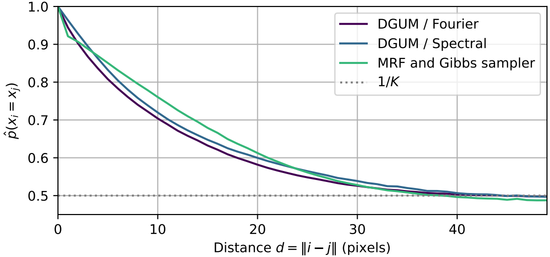
(f) Gibbs sampling

Numerical results: statistical properties

Pointwise measures, denoting $\pi_k = p(x_s = k)$

	K	2	3	4
DGUM / Fourier	$\hat{\pi}_0$	0.4924	0.3278	0.2369
	$\text{std}(\hat{\pi}_0)$	0.0650	0.0497	0.0533
DGUM / Spectral	$\hat{\pi}_0$	0.5036	0.3296	0.2534
	$\text{std}(\hat{\pi}_0)$	0.0935	0.0780	0.0711
Gibbs sampling	$\hat{\pi}_0$	0.4980	0.3470	0.2480
	$\text{std}(\hat{\pi}_0)$	0.0399	0.0581	0.0505

Pairwise statistics :



Inference

Inferring with GUMs

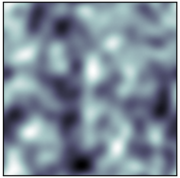
Observation model: we assume $\mathbf{y}$ is obtained as a noisy version of $\mathbf{x}$. For instance: class-wise parameters changes within Gaussian or Poisson distributions.

We conjecture that inferring with a GUM prior is not tractable.

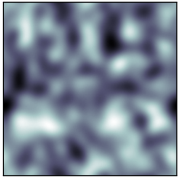
The GUM inverse problem relies on:

$$p(\mathbf{y}, \mathbf{x}) \propto p(\mathbf{y}|\mathbf{x})p(\mathbf{x}) = p(\mathbf{y}|\mathbf{x}) \int_{\mathbb{R}^{n(K-1)}} p(\mathbf{x}|\mathbf{z}) d\mathbf{z}$$

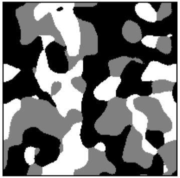
- not that simple
- too much information is lost in $p(\mathbf{x}|\mathbf{z})$



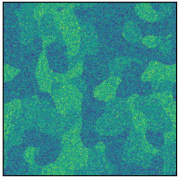
(a) $\mathbf{z}_1$



(b) $\mathbf{z}_2$



(c) $\mathbf{x}$



(d) $\mathbf{y}$

The problems & solutions: inference with Kernel MLE

$\forall s$ we can write a likelihood:

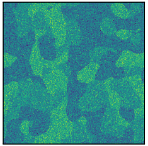
$$L_k(y_s) = p(y_s | x_s = \omega_k)$$

We then design a maximum of kernel MLE, $\forall s$:

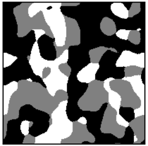
$$\hat{\mathbf{x}}_\rho = \arg\max_k \left(\mathbf{\Sigma}_\rho L_k(\mathbf{y})\right)_s$$

with $\mathbf{\Sigma}_\rho$ depending on a covariance function of range parameter ρ.

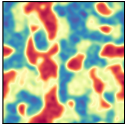
Then: how to determine ρ ?



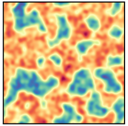
(a) $\mathbf{y}$



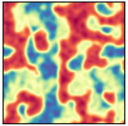
(b) $\mathbf{x}$



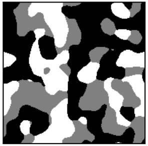
(c) $\mathbf{\Sigma}_\rho L_0(\mathbf{y})$



(d) $\mathbf{\Sigma}_\rho L_1(\mathbf{y})$



(e) $\mathbf{\Sigma}_\rho L_2(\mathbf{y})$



(f) $\hat{\mathbf{x}}_\rho$

The problems & solutions (II): optimization problem

How to determine ρ ? $\rightarrow$ a MAP estimator:

$$\rho^* = \arg\max_{\rho>0} \; p(\mathbf{y}|\hat{\mathbf{x}}_\rho)p(\hat{\mathbf{x}}_\rho)$$

Let us assume a Potts-like potential for $\mathbf{x}$, assuming $\lambda > 0$:

$$p(\mathbf{x}) \propto \exp\left(-\lambda U(\mathbf{x})\right), \text{ and } U(\mathbf{x}) = \sum_{s\in\mathcal{S}} \sum_{s'\in N_s} \mathbb{1}_{\{x_s = x_{s'}\}}$$

Then the MAP estimator can be rewritten:

$$\arg\min_{\rho>0} \; -\log(p(\mathbf{y}|\hat{\mathbf{x}}_\rho)) + \lambda U(\hat{\mathbf{x}}_\rho) \qquad (\mathcal{P}_\lambda)$$

This can be solved quite easily numerically, for a given λ, because computing $\hat{\mathbf{x}}_\rho$ is fast.

Then: how to determine λ ?

The problems & solutions (III): homotopy approach ⚠

How to determine λ ? $\rightarrow$ Proximity to sparse optimization problems:

- Solutions of $(\mathcal{P}_\lambda)$ lies on the Pareto frontier
- Homotopy methods probe this frontier until a solution is found

Purpose : we can trade the choice for λ with the choice for a ℓ_2-related parameter.

$$\arg\min_{\lambda} \left| 1 - \left\| \frac{\mathbf{y} - \boldsymbol{\mu}_{\hat{\mathbf{x}}_\rho}}{\boldsymbol{\sigma}_{\hat{\mathbf{x}}_\rho}} \right\|_2 \right| \text{ such that } \rho \text{ is solution to } (\mathcal{P}_\lambda) \qquad (\mathcal{P}_{\text{hom}})$$

Rephrasing : correctness of solution found for λ with respect to the known noise parameter (mean, variance) gathered under $\boldsymbol{\Theta}$?

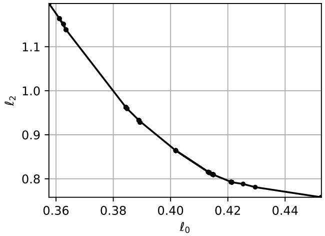


Fig. 2: Pareto frontier: solutions to $(\mathcal{P}_\lambda)$ are under-optimal above the curve and unattainable below.

Inference: wrapping up

Several layers of nested problems:

- The core problem: an estimator $\hat{\mathbf{x}}_\rho$ that is computationally cheap, for a given ρ
- ρ is found solving $(\mathcal{P}_\lambda)$ for a given λ
- λ is found solving $(\mathcal{P}_{\mathrm{hom}})$ given noise parameters $\boldsymbol{\Theta}$
- $\boldsymbol{\Theta}$ is estimated by an alternating scheme (not detailed here)

Preliminary results:

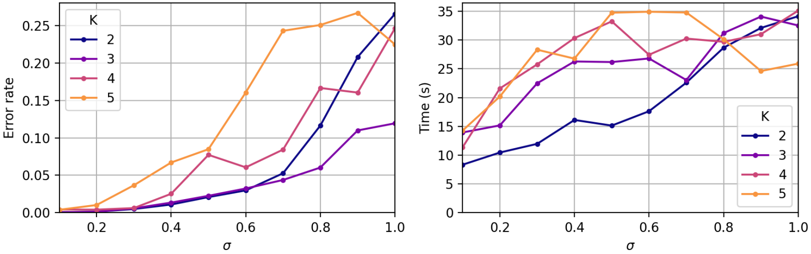


Fig. 3: Error rate (left) and computation time(right) under Gaussian noise with means at $\{1, 2, \dots, K\}$ and varying standard deviation σ.

Next steps

Next steps

Sampling:

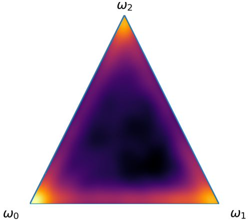
- Properties of **X**
- Class unbalance

Inference:

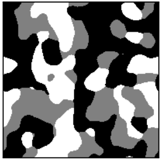
- Relation between inference formalism and GUMS (if any)
- Formalize the problem nesting & relation to sparse optimization
- Numerical results & real microscopy images

Representations:

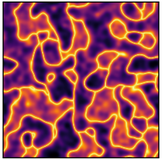
- The probability simplex
- Uncertainty quantification



(a) Pointwise DGUM density in the probability simplex ($K = 3$).



(b) $\hat{\mathbf{x}}_\rho$



(c) Tentative uncertainty

Resources

Take-home message: to avoid Gibbs-induced computational costs, we propose:

- a fast sampling method that mimic Potts models
- a fast nested inference method, suitable for unsupervised inverse problems

Resources so far:

- Sampling was described in SSP 2025[9]
- Sampling code is available[10]

In progress:

- journal paper to wrap this up
- related: a tutorial on Markov models in image processing (with Julien Stoehr)

[9] Courbot and Gangloff, "Gaussian Unit-simplex Markov random fields as a fast proxy for MRF sampling", 2025.

[10] `https://github.com/HGangloff/mrfx`