Bayesian Inversion via Probabilistic Cellular Automata: an application to image denoising

Danilo Costarelli*, Michele Piconi†, and Alessio Troiani‡

Department of Mathematics and Computer Science, University of Perugia

Abstract

We propose using Probabilistic Cellular Automata (PCA) to address inverse problems with the Bayesian approach. In particular, we use PCA to sample from an approximation of the posterior distribution. The peculiar feature of PCA is their intrinsic parallel nature, which allows for a straightforward parallel implementation that allows the exploitation of parallel computing architecture in a natural and efficient manner. We compare the performance of the PCA method with the standard Gibbs sampler on an image denoising task in terms of Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM). The numerical results obtained with this approach suggest that PCA-based algorithms are a promising alternative for Bayesian inference in high-dimensional inverse problems.

AMS subject classification: 60J22, 62F15, 62M20, 62M40

Key words: Bayesian inversion; Markov Chain Monte Carlo, Probabilistic Cellular Automata; Gibbs sampler; image denoising

1 Introduction

Loosely speaking, an inverse problem consists of determining the *root cause* of an observed phenomenon. Several examples fall into this category, spanning a wide range of fields.

Problems of this type arise very frequently in remote sensing, for instance, in the context of large-scale monitoring of Essential Climate Variables (ECVs) such as Soil Moisture (SM) [5, 11], Freeze-Thaw state (FT) [23, 28], and Above Ground Biomass (ABG) [7, 8]. In this context, it is common to have two-dimensional arrays of some measured backscattered signal that one wants to convert into levels of some measured physical quantity, such as soil permittivity, and, in turn, to quantitative values of the ECV of interest. Note that the measured signal will, in general, be affected by noise. Problems like this are typically ill-posed, and several strategies have been developed to tackle them (see [1, 24, 32] for comprehensive reviews).

Among such strategies, we consider the Bayesian approach to retrieval (as in [20, 26]), which has the peculiar feature of treating all quantities at stake as random variables. The key object of the Bayesian approach is the so-called *posterior distribution*, which is a probability distribution of the possible root causes given the observed data. The retrieval is then achieved by sampling from the posterior distribution, typically by collecting samples from a Markov chain with the

*danilo.costarelli@unipg.it
†michele.piconi@unipg.it
‡alessio.troiani@unipg.it

posterior distribution as its stationary distribution. This approach is known as the Monte Carlo Markov Chain (MCMC).

In this paper we propose an MCMC approach where the considered Markov Chain is a Probabilistic Cellular Automaton (PCA), that is a Markov Chain that at each step, updates all components of the state vector independently one from another ([9, 10, 14, 22]) with a probability that is affected only by the *local* effect of the update of each component. Due to its parallel nature, a PCA can be simulated effectively on a parallel computing architecture such as a GPU ([21]) where, in principle, each computing core may take care of the update of a single component of the state vector. All these updates may be performed simultaneously. In general, it is not immediate to define a chain of this type having the stationary distribution equal to the sought-for posterior distribution. However, we will see how it is possible to define a PCA whose stationary distribution is close to the desired one (see, e.g., [12]).

Our aim is to show that the PCA approach can be a viable alternative to the use of other MCMC approaches, such as the Gibbs Sampler and the Metropolis-Hastings algorithm, where to update a subset of the components of the state space, the *global* effect of the update must be evaluated. Since evaluating this global effect can be computationally expensive, it is common in the literature to consider only single-component sequential updates.

To support the viability of our approach, we will consider simple cases of noisy image restoration. To this extent, we will consider one of the scenarios discussed in the seminal paper [16], which contributed greatly to the popularity of the Bayesian approach to inversion.

This paper aims to be an attempt to evaluate the effectiveness of PCA in addressing inverse problems within the Bayesian framework. In this respect, the results we present here are quite promising. In particular, we show that the PCA is able to produce slightly better results with respect to the Gibbs sampler in terms of Peak Noise to Noise Ratio (PSNR) and Structural Similarity Index (SSIM).

In Section 2, we provide some background on the Bayesian approach to inversion and describe how Markov Chain Monte Carlo (MCMC) techniques can be used to draw samples from the posterior distribution. In particular, in this section, we recall the definition of the Gibbs Sampler and show the relation between its stationary measure and the posterior distribution.

In Section 3 we give the definition of Probabilistic Cellular Automata, characterize their stationary measure, and show how this distribution can be directed towards the desired posterior distribution.

In Section 4 we describe the prior distribution of the test images, whereas Section 5 we present the details of the retrieval algorithms we considered.

In Section 6 we present a comparison of the results obtained with the PCA approach with those of the Gibbs sampler in terms of Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index (SSIM).

Finally, in Section 7 we highlight future lines of research.

2 Background

Bayesian Inversion. An inverse problem concerns the estimation of some object of interest x based on some observed or measured quantity g. The relationship between x and g is given in

terms of a mathematical model

$$g = \psi(x, \varepsilon)$$

where ε is a term that accounts for noise affecting the observation, as well as other unobserved or poorly known quantities. In this context, solving an inverse problem amounts to finding an "inverse" of the function ψ.

Both x and g may be quite general objects. For example, in the context of remote sensing, x could refer to soil permittivity, which is related to essential climate variables like Soil Moisture and Freeze-Thaw state, while g might be the measured backscatter of a signal. In the context of image processing, x may be the "true" image of interest, whereas g represents the degraded observed version of that image affected by factors such as noise (the ε term) or blur (a suitable form of ψ).

In many applications, the problem of inverting ψ is not well posed, and several approaches have been developed to tackle it (see [4] for an overview in the context of parameter estimation and [6] for an introduction in the field of imaging). Here we consider the so-called Bayesian approach (for a comprehensive introduction, see the monograph [19]). In the Bayesian framework, all quantities at stake are treated as (in general, multivariate) random variables linked by the relation

$$G = \psi(X, N).$$

and the inverse problem consists of determining the so-called posterior distribution

$$\pi_{\text{post}}(G = g | X = x)$$

In the following, we will always denote random variables with capital letters, whereas lowercase letters will denote the values taken by the (usually corresponding) random variables. When it does not give rise to ambiguity, we do not write explicitly the name of the random variable and write expressions as the previous one as $p_{\text{post}}(g|x)$. By Bayes' theorem

$$\pi_{\text{post}}(x|g) = \frac{\pi(x, g)\pi_{\text{prior}}(x)}{\pi(g)}$$

where π is the joint distribution of X and G and π_{prior}, called the prior distribution of X, models the knowledge we have on the quantity we want to estimate *before* the outcome of the experiment (measured physical quantity, noisy image, ...) has been observed.

Here we only consider models of the form

$$G = \phi(X) \odot N \tag{1}$$

where $\odot$ is an invertible function such as addition or multiplication. Further, we will always assume N and X to be independent. Denoting by Φ the inverse of $\odot$, we have

$$\pi(g|x) = \pi_{\text{noise}}\{N = \Phi(g, \phi(x))\} \tag{2}$$

which is called the *likelihood* of the observed data given the *ground truth* x.

Observing that, conditional on $G = g$, the marginal density $\pi(g)$ is a constant, it is easily seen that

$$\pi_{\text{post}}(x|g) \propto \pi_{\text{prior}}(x)\pi_{noise}(\Phi(g, \phi(x)))$$

where $\propto$ means "proportional to". The posterior distribution is, therefore, determined by the prior probability assumed for X and the model chosen to describe the noise.

Note that we provided the expression of $\pi_{\text{post}}(x|g)$ only up to a multiplicative factor that would turn the previous expression into a proper probability density. In the previous case, this multiplicative constant is the marginal density $\pi(g)$. In general, this normalizing constant is hard to evaluate explicitly. However, using the so-called Markov Chain Monte Carlo (MCMC) approach (see below), it is possible to sample from probability distributions only known up to a multiplicative factor. In what follows, we will often write un-normalized probability densities (using the symbol $\propto$) when the knowledge of the normalizing constant is not needed in the retrieval process.

In this paper, we will consider prior distributions of the form

$$\pi_{\text{prior}}(u) = \frac{e^{-\beta H(x)}}{Z} \tag{3}$$

A distribution of this type is called a *Gibbs* distribution and originates in statistical mechanics. The parameter $\beta > 0$ is called *inverse temperature*. The *energy* function H is called Hamiltonian, and Z is a normalizing constant (referred to as *partition function* in the context of statistical mechanics).

As for the likelihood, we will consider (additive) Gaussian noise with mean μ and variance σ^2. Then, the likelihood is of type

$$\pi_{\text{noise}}(g|x) \propto e^{-\frac{1}{2\sigma^2}\|\mu - \Phi(g,\phi(x)\|_2^2} \tag{4}$$

and the posterior distribution, therefore, of type

$$\pi_{\text{post}}(x|g) \propto e^{-\beta\left[H(x) + \frac{1}{\beta}\frac{1}{2\sigma^2}\|\mu - \Phi(g,\phi(x)\|_2^2\right]} \tag{5}$$

That is, it is again a Gibbs distribution with Hamiltonian

$$H_g(x) = H(x) + \frac{1}{\beta}\frac{1}{2\sigma^2}\|\mu - \Phi(g,\phi(x)\|_2^2 \tag{6}$$

In this paper, we refer to the Hamiltonian defining the posterior Gibbs distribution as the *posterior Hamiltonian*.

Bayesian inference. In a deterministic framework, the inversion process aims to produce a single "recovered" value. On the other hand, in the Bayesian framework, the outcome of the inversion process is the posterior distribution, which is a probability distribution over the entire set of values that can be taken by the variable of interest X. However, to obtain useful information from the posterior distribution, we have to specify how this information must be "extracted" from this distribution. Typical choices are the so-called *conditional mean estimate* (CM) and *maximum a posteriori estimate* (MAP), both providing a point estimate for the quantity to be recovered. The former is defined as

$$\hat{x}_{\text{MAP}} := \arg\max_x \left\{\pi_{\text{post}}(x|g)\right\}, \tag{7}$$

that is, it is the value that maximizes the posterior density, whereas the latter is

$$\hat{x}_{\text{CM}} := \mathbb{E}(x|g) = \int u \cdot \pi_{\text{post}}\left(u \mid g\right)\mathrm{d}u, \tag{8}$$

that is the average value with respect to the posterior density of the quantity one wants to recover.

Although, in principle, computing the maximum a posteriori and conditional mean estimates is straightforward, in many real situations, solving the optimization problem or computing the integral may be a challenging task. This is due to the fact that the problem lives in a high-dimensional space. A common scenario is $x \in S^n$ where S may be finite or even infinite. As an example, in the case where x represents a grayscale image, S is the number of allowed gray levels for each pixel, and n is the number of pixels. Even for pure black-and-white images with a very limited size (e.g. 32×32), this state space is huge.

Both the optimization and integration problems can be solved using a Monte Carlo (MC) approach. For the computation of the conditional mean, it is possible to exploit the law of large numbers. Indeed, consider a random variable $X \in \mathcal{X}$ and let $x_1, x_2, \ldots, x_N$ be a collection of points drawn from the probability distribution f_X of X. Then, as N gets large,

$$\mathbb{E}\, h(X) = \int_{\mathcal{X}} h(x) f_X \,\mathrm{d}\, x \approx \frac{1}{N} \sum_{i=1}^{N} h\left(x_i\right). \tag{9}$$

On the other hand, to solve the optimization problem, it would be possible to draw samples from the neighborhood of the maximizers of f_X.

Consequently, a key point to effectively perform Bayesian inversion is the ability to collect samples from the posterior distribution. One way to achieve this task is to consider the MCMC approach: define a Markov Chain $X^{(t)}$ living on $\mathcal{X}$ having a prescribed distribution π as invariant measure. Then, under mild conditions on $X^{(t)}$, (irreducibility and aperiodicity), the limiting distribution of the chain, also called the stationary measure, is π, independently on the initial state of the chain (see [17] for a proof).

When the state space of the chain is $\mathcal{X} = S^n$, one possibility to have a prescribed π as invariant measure is to consider the Gibbs-Sampler algorithm. According to this sampling scheme, the i-th component of X is updated according to the conditional distribution $\pi(X_i = \cdot | \{X_j = x_j, j \neq i\})$. More properly:

$$\begin{aligned}\mathbb{P}\left(X^{(t+1)} = (x_1^{(t)}, \ldots x_{i-1}^{(t)}, x_i, x_{i+1}^{(t)}, \ldots x_n^{(t)}) \middle| X^{(t)} = (x_1^{(t)}, \ldots x_n^{(t)})\right) \\ = \pi\left(X_i = x_i \middle| \{X_j = x_j^{(t)}, j \neq i\}\right). \end{aligned} \tag{10}$$

The component to be updated may be chosen at random, or the components may be updated in a systematic way, one after another (systematic Gibbs sampler). In analogy with the statistical mechanics parlance, we call this sampling scheme the (systematic) *single spin flip* Gibbs sampler.

Consider the case where π is a Gibbs distribution with Hamiltonian H over the space S^n. If H has the form $H(x) = \sum_{i=1}^{n} f_i(x, x_i)$, then the conditional probabilities $\pi(X_i = s | \{X_j = x_j, j \neq i\})$ are of type

$$\pi(X_i = s | \{X_j = x_j, j \neq i\}) = \frac{e^{\beta f_i(x,s)}}{\sum_{s \in S} e^{\beta f_i(x,s)}} \tag{11}$$

Their computation is, therefore, straightforward (at least in the case where S is "small"). Further, it is commonn that f_i only depends on the values of x in a *neighborhood* of component i (e.g., the pixels that are close to pixel i in the case of image) and, hence, its computation is fast even if the number of components is large.

Note that, usually, the interesting time scale of the chain is not the scale where a single update takes place, but, rather, the scale defined by *sweeps*, that is, sequences of n updates. Therefore, in a sweep, the systematic Gibbs sampler samples a new value for each component of x.

The Geman-and-Geman paper. In their seminal paper [16], Geman and Geman showed that in the case of Gibbs prior distribution and noise independent of the "true" signal, in the limit $\beta \to \infty$, the stationary measure of the Gibbs Sampler is the uniform distribution over the maxima of the posterior distribution, provided the cooling schedule is "slow enough". In particular, they showed that the result holds if the inverse temperature β goes to infinity as $\log k$, where k is the number of steps of the chain. Unfortunately, this cooling scheme is too slow for practical applications. However, their results showed that the MCMC approach can be a viable option for tackling optimization problems. Furthermore, their successful application of the Gibbs Sampler to image restoration demonstrated that even practically feasible cooling schemes can be beneficial in several applications.

3 Probabilistic Cellular Automata

In principle, for the Gibbs sampler to work, it is not necessary to update a single component of X_i at a time. Consider $X \in S^n$ (that is, $X = (X_1, X_2, \dots X_n)$) and let π be a probability distribution on S^n. Let $I \subset \{1, 2, \dots, n\}$ and call $J = \{1, \dots n\} \setminus I$. Then, a Gibbs sampler is a Markov chain whose transition probabilities are given by

$$\mathbb{P}\left(X^{(t+1)} = (x_i,\, i \in I; x_j^{(t)}, j \in J) \middle| X^{(t)} = (x_1^{(t)}, \dots x_n^{(t)})\right) = \pi\left(\{X_i = x_i,\, i \in I\} \middle| \{X_j = x_j^{(t)},\, j \in J\}\right). \quad (12)$$

Also in this case, there is a certain freedom to choose the set I. One possibility is to fix the size of I to be k and sample, at each step, a new subset of indices to be updated of size k.

However, in general, the computation of the conditional probabilities in (12) may be highly demanding from a computational point of view when I is not a singleton.

A drawback of the single spin flip Gibbs sampler is the fact that the updates of two different components of X are, in general, not independent. This dependence does not allow for the exploitation of the computational capabilities of parallel processors (such as GPUs) at their fullest. It would be tempting to update all components of X *in parallel* and perform a sweep in a single pass, that is, set[1]

$$\mathbb{P}\left(X^{(T+1)} = (x_1, \dots x_n) \middle| X^{(T)} = (x_1^{(T)}, \dots x_n^{(T)})\right) = \prod_{i=1}^{n} \pi(X_i^{(T+1)} = x_i | X^{(T)} = (x_1^{(T)}, \dots x_n^{(T)})). \quad (13)$$

Unfortunately, this update rule does not have the measure π as the stationary distribution and, actually, it may not be related at all to the Gibbs measure. Further, its long-term behavior may be highly dependent on the initial condition.

A Markov Chain on S^n with transition probability matrix defined as

$$\mathrm{P}(x, w) := \mathbb{P}\left(X^{(T+1)} = (w_1, \dots w_n) \middle| X^{(T)} = (x_1, \dots x_n)\right) = \prod_{i=1}^{n} \mathbb{P}(X_i^{(T+1)} = w_i | X^{(T)} = x) \quad (14)$$

is called a Probabilistic Cellular Automaton (PCA). A Markov Chain of this kind can be effectively simulated on a parallel computing architecture. This is because, given the current

[1] Note that here (and in what follows), we used the capital letter T to denote the generic time step of the chain in place of the lower case letter t. This is to highlight that with this update rule, the time-scale is that of a full *sweep:* all components have a chance to be updated.

configuration, the value of each component can be sampled independently from the others. A dedicated computing unit can determine each new value.

The PCA that we will describe in the following sections will have a stationary distribution that is *close* to the actual posterior distribution. This feature is achieved by introducing an inertial term in the Hamiltonian, which prevents too big changes at each step of the chain. Although this sampling scheme yields a stationary distribution that is only an approximation to the desired posterior distribution, the increased computing efficiency makes this approach worth further investigation.

Consider a Hamiltonian $H(x) = \sum_{i=1}^{n} f_i(x, x_i)$ and let X be a Markov Chain on S^n with transition probability matrix

$$\mathrm{P}(x, w) = \frac{e^{-\beta H(x,w)}}{\sum_w e^{-\beta H(x,w)}} \tag{15}$$

where $H(x, w) = \sum_i f(x_i, w_i)$ is a *lifted* two-arguments version of the original Hamiltonian H. Then it is immediate to show that (15) defines a PCA in the sense of (14). Indeed,

$$\mathrm{P}(x, w) \propto e^{-\beta H(x,w)} = e^{-\beta \sum_{i=1}^{n} f_i(x,w_i)} = \prod_{i=1}^{n} e^{-\beta f_i(x,w_i)}. \tag{16}$$

We have

$$\mathrm{P}(X_i^{(T+1)} = w_i | X^{(T)} = x) = \frac{\mathbb{P}(X^{(T+1)} = w_i, X^{(T)} = w)}{\mathbb{P}(X^{(T)} = x)} \tag{17}$$

Note that

$$\begin{aligned} \mathbb{P}(X^{(T)} = x) &= \sum_w \mathbb{P}(X^{(T+1)} = w | X^{(T)} = x) \\ &\propto \sum_{w_1 \in S} \cdots \sum_{w_n \in S} \prod_{j=1}^{n} e^{-\beta f_j(x,w_j)} \\ &= \prod_{j=1}^{n} \sum_{w_j \in S} e^{-\beta f_j(x,w_j)}. \end{aligned} \tag{18}$$

Similarly,

$$\mathbb{P}(X^{(T+1)=w_i}, X^{(T)=w}) \propto e^{-\beta f_i(x,w_i)} \prod_{j \neq i} \sum_{w_j \in S} e^{-\beta f_j(x,w_j)} \tag{19}$$

Then the claim follows with

$$\mathbb{P}(X_i^{(T+1)} = w_i | X^{(T)} = x) = \frac{e^{-\beta f_i(x,w_i)}}{\sum_{s \in S} e^{-\beta f_i(x,s)}}. \tag{20}$$

Further note that, if $H(x, w) = H(w, x)$, then it is easy to show that the stationary measure of this chain is

$$\pi(x) = \frac{Z_x}{Z} \coloneqq \frac{\sum_w e^{-H(x,w)}}{\sum_{x,w} e^{-H(x,w)}}. \tag{21}$$

To prove this it is enough to show that the detailed balance condition $\pi(x)\,\mathrm{P}(x, w) = \pi(w)\,\mathrm{P}(w, x)$ holds (see, e.g. [17]). For the transition matrix (15) we immediately have

$$\pi(x)\,\mathrm{P}(x, w) = \frac{Z_x}{Z} \frac{e^{-H(x,w)}}{Z_x} = \frac{e^{-H(x,w)}}{Z} = \frac{e^{-H(w,x)}}{Z} = \frac{Z_w}{Z} \frac{e^{-H(w,x)}}{Z_w} = \pi(w)\,\mathrm{P}(w, x) \tag{22}$$

The *lazy* PCA As already mentioned, if we simply "lift" the Hamiltonian of the posterior distribution and update the current state of the chain according to the transition matrix defined in (16), the stationary measure of the chain is not the posterior distribution. To counter this issue, we modify the posterior Hamiltonian by adding an inertial term that puts a constraint on the freedom with which each component is updated at each step of the chain. We remark, however, that conditionally on the current state, each component of the chain is updated independently of all others at each step.

In particular, we use a pair Hamiltonian of type

$$\tilde{H}(x,w) = H(x,w) + q\|x-w\| \tag{23}$$

where q is a positive parameter and $\|\cdot\|$ is a suitable "norm". We take $q\|x-w\| = \sum_i q|x_i - w_i|^p$, where we use the convention $0^0 = 0$. In general, a Markov Chain of this type does not have the Gibbs distribution as a stationary measure. However, if the inertia of the system is big enough, the stationary measure of the chain is close to the Gibbs measure. This argument has been made precise in the discrete context of the Ising model in [12, 27, 2].

Note that, if $H(x,w) = \sum_{i=1}^n f_i(x,w_i)$, then

$$\tilde{H}(x,w) = \sum_{i=1}^n f_i(x,w_i) + \sum_{i=1}^n q|x_i - w_i|^p = \sum_{i=1}^n \tilde{f}_i(x,w_i) \tag{24}$$

and, hence, the Markov chain with transition probabilities

$$\mathrm{P}(x,w) = \frac{e^{-\beta\tilde{H}(x,w)}}{\sum_w e^{-\beta\tilde{H}(x,w)}} \propto \prod_{i=1}^n e^{-\beta\tilde{f}_i(x,w_i)} \tag{25}$$

defines, again, a PCA in the sense of (14).

The PCA for the posterior distribution In this paper we want to apply *lazy* PCA approach to sample from the posterior distribution (5). We assume that the posterior Hamiltonian can be written in the form $H_g(x) = \sum_{i=1}^n f_i(x,x_i)$. Then to define the lazy PCA we first lift $H_g(x)$ to $H_g(x,w)$ as in (15) and then add to it the term $q\|x-v\|$ to obtain the posterior Hamiltonian

$$\tilde{H}_g = H_g(x,w) + q\|x-v\|. \tag{26}$$

Then, we can define the transition probabilities as

$$\tilde{\mathrm{P}}(x,w) = \frac{e^{-\beta\tilde{H}_g(x,w)}}{\sum_w e^{-\beta\tilde{H}_g(x,w)}} = \frac{e^{-\beta\sum_i f_i(x,w_i)+q|x_i-w_i|^p}}{\sum_w e^{-\beta\sum_i f_i(x,w_i)+q|x_i-w_i|^p}} = \prod_{i=1}^n \frac{e^{-\beta[f_i(x,w_i)+q|x_i-w_i|^p]}}{\sum_{s\in S} e^{-\beta[f_i(x,s)+q|x_i-s|^p]}}. \tag{27}$$

The shape of f_i that must be plugged in (27) depends on the particular choice of the prior distribution. In the cases we consider in this paper (see Section 6), f_i is such that $\tilde{H}_g(x,w) = \tilde{H}_g(w,x)$. Then, arguing as for (21), it follows that the previous transition matrix defines a Markov Chain with stationary measure

$$\tilde{\pi} = \frac{\sum_w \tilde{\mathrm{P}}(x,w)}{\sum_{x,w}\tilde{\mathrm{P}}(x,w)}. \tag{28}$$

Distinctive fatures of PCA The primary advantage of the PCA approach to inversion is that the update rule operates completely in parallel. This means that, in principle, the computations

needed to update a single component of the Markov chain could be carried out by a dedicated computing unit, such as a GPU core.

It's important to note that this type of parallelism arises not from a specialized implementation of the algorithm, but from an intrinsic feature of the algorithm itself. As a result, the computational efficiency of PCA is likely to benefit "for free" from the technological advancements occurring in parallel computing architectures across various fields.

The counterpart to the parallel evolution is that the stationary distribution of the chain does not coincide with the sought-for posterior distribution, but it is only an approximation. The goodness of the approximation depends on the parameter q and improves as q grows larger.

Currently, there are only asymptotic results, showing that the stationary measure of the PCA converges to the Gibbs measure for the Ising model, which can be viewed as a case of pure black-and-white images when considering image restoration, as $q \to \infty$. However, some numerical results suggest that PCA can provide quite accurate estimates for finite values of q [13, 30], and the findings presented here appear to support this observation.

4 The Image Model

In this paper, we considered gray-scale images. However, our approach could be extended to RGB images by treating each color layer separately. Each image is encoded as a square matrix Λ where each entry represents the *luminance* of a pixel. Pixels take value in the range $[0, 1]$ where 0 represents pure black and 1 pure white. However, the values that can be taken by each pixel are not arbitrary, but are *quantized* due to the fact that only a finite number of bits is used to encode the "intensity" of a pixel. We denote by ℓ the number of possible levels of gray.

In what follows, we to explicitly identify the position and the values of the pixels of an image. Note that it is possible to add an arbitrary ordering to the elements of Λ (for instance, the standard *column-major* ordering). When we use a single index such as i or j to denote an element (pixel) of Λ we refer to this arbitrary *linear* ordering, whereas when we use a double indexing (r, c) when we refer to the pixel at row r and column c. We call this ordering *cartesian*. Clearly, there is an obvious bijection $i \leftrightarrow (r_i, c_i)$between the linear and the cartesian ordering. Consequently, we write x_i to denote the intensity of the i-th pixel or $x_{(r,c)}$ for explicitely denoting the intensity of pixel at row r and column c.

As a prior distribution for the images taken into account, we considered the Gibbs measure with prior Hamiltonian

$$H(x) = \sum_{(r,c)\in\Lambda} f_{(r,c)}(x, x_{(r,c)}) \tag{29}$$

where

$$f_{(r,c)} = \sum_{(p,q)\in\mathcal{F}(r,c)} V(x_{(p,q)}, x_{(r,c)}) \tag{30}$$

where $\mathcal{F}(r, c)$ is a *neighborhood* of (r, c) and

$$V(z, w) = \begin{cases} -J & \text{if } z = w \\ +J & \text{otherwise} \end{cases} \tag{31}$$

with $J > 0$. F the neighborhood $\mathcal{F}$ we took

$$\mathcal{F}(r, c) = \{(p, q) : \max\{|r - p|, |c - q|\} = 1\} \tag{32}$$

that is, the neighbors of (r, c) are the 8 sites of Λ "surrounding" (r, c) (see Figure 1). Note that the number of neighbors of sites next to the boundary of the image is less than that in the "center".

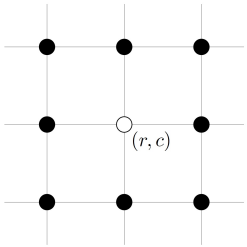


Figure 1: Neighborhood structure $\mathcal{F}(r, c)$: each pixel (center) interacts with its 8 surrounding neighbors.

As mentioned in Section 2, the Gibbs measure associated with this Hamiltonian can be written as

$$\pi_{\text{prior}}(x) \propto e^{-\beta H(x)} = \prod_{i \in \Lambda} \exp\{-\beta f_i(x, x_i)\} = \prod_{i \in \Lambda} \exp\left\{\beta \sum_{j \sim i} J \cdot 2(\mathbb{1}_{\{x_i = x_j\}} - 1)\right\} \tag{33}$$

where by $j \sim i$ we mean the sites of Λ belonging to $\mathcal{F}(i)$ and $J2(\mathbb{1}_{\{x_i = x_j\}})$ is an alternative way of writing (31). This prior probability measure favors configurations (images) where pixels are aligned with their neighbors. This tendency arises from the attractive nature of the potential defined in equation (31). The use of the "double minus sign"—with one appearing in front of the Hamiltonian in the exponent of the probability measure and the other in front of the J that tunes the strength of the interaction—is common in statistical mechanics. This notation emphasizes that the configurations that occur most frequently are those that minimize the energy of the system.

As far as the degraded image is concerned, in this paper we only considered the simple case of Gaussian white noise, that is the functions ϕ and $\odot$ are, respectively, the identity and *addition*. This means that we consider $X - G$ (that is the difference between the observed and the original value of a pixel) to be normally distributed, i.e,

$$\pi(x, g) \propto e^{-\frac{1}{2\sigma^2}\|g - x\|_2^2} \tag{34}$$

Consequently, the posterior Hamiltonian can be written as

$$H_g = H(x) + \frac{1}{\beta}\frac{1}{2\sigma^2}\sum_{i \in \Lambda}(g_i - x_i)^2 \tag{35}$$

and the posterior distribution as

$$\pi_g^\beta(x) := \pi^\beta(x|g) \propto e^{-\beta H_g(x)} = \prod_{i \in \Lambda} \exp\left\{\sum_{j \sim i} \beta J \cdot 2(\mathbb{1}_{\{x_i = x_j\}}) - \frac{1}{2\sigma^2}(g_i - x_i)^2\right\} \tag{36}$$

Note that the posterior distribution is characterized by a probabilistic price to pay to have values for the pixels that are different from the observed ones.

5 Retrieval Algorithms

To test our PCA approach to retrieval, we compare it to the standard Gibbs Sampler algorithm.

5.1 Retrieval via Gibbs Sampler

As outlined in the Introduction, the Gibbs Sampler updates the pixel at site i according to the conditional posterior probability as

$$\mathbb{P}\left(X_i = s | g, \{X_j = x_j, j \neq i\}\right) = \pi_g(X_i = s | \{X_j = x_j, j \neq i\}) \tag{37}$$

which can be written as

$$\pi(X_i = s | g, \{X_j = x_j, j \neq i\}) = \frac{\exp\left\{\left(\sum_{j\sim i} \beta J \cdot 2(\mathbb{1}_{\{s=x_j\}})\right) - \frac{1}{2\sigma^2}(g_i - s)^2\right\}}{\sum_{s\in S} \exp\left\{\left(\sum_{j\sim i} \beta J \cdot 2(\mathbb{1}_{\{s=x_j\}})\right) - \frac{1}{2\sigma^2}(g_i - s)^2\right\}} \tag{38}$$

where S is the set of possible gray levels.

The retrieval procedure performs a sequence of *sweeps* where each sweep consists in a sequence of steps, each as in (38), performed on the sites of Λ taken in a predetermined order (for instance the column-major one).

The parameter β, which is the inverse of the temperature, can, in principle be varied during the retrieval process. The so-called *simulated annealing* is performed by increasing β trhoughout the evolution of the chain according to a suitable schedule.

5.2 Retrieval via PCA

As discussed in Section 3, a PCA is a Markov chain where transition probabilities are of the form of (16) and are defined in terms of a suitable "pair Hamiltonian". The key feature of this pair Hamiltonian is that it can be written in terms of a sum over the sites of Λ. In an inversion problem such as the one we consider in this paper, we want the pair Hamiltonian to be a "lifted" version of the posterior Hamiltonian (35) as explained at the beginning of Section 3. Further, to put forward our *lazy* PCA approach, we add the posterior pair Hamiltonian an inertial term which is proportional to a suitable *distance* between the configuration corresponding to the two arguments of the pair Hamiltonian. Here we consider the following *posterior pair Hamiltonian*

$$\begin{aligned}\tilde{H}_g(x, w) &= H_g(x, w) + q\|x - w\|_0 \\ &= \sum_{i\in\Lambda}\left(-\sum_{j\sim i} J \cdot 2(\mathbb{1}_{\{\mathtt{x_i}=\mathtt{x_j}\}})\right) + \frac{1}{\beta}\frac{1}{2\sigma^2}\sum_{i\in\Lambda}(g_i - x_i)^2 + q\sum_i \mathbb{1}_{\{x_i\neq w_i\}}\end{aligned} \tag{39}$$

where q is a positive parameter.

With these definitions, the retrieval procedure consists of a sequence of steps where at each step the probability of going from x to w is

$$\begin{aligned}\mathrm{P}(x, w) &\propto e^{-\beta\tilde{H}_g(x,w)} \\ &= \prod_{i\in\Lambda} \frac{\exp\left\{\left(\sum_{j\sim i} \beta J \cdot 2(\mathbb{1}_{\{s=x_j\}})\right) - \frac{1}{2\sigma^2}(g_i - s)^2 - \beta q \mathbb{1}_{\{s\neq x_j\}}\right\}}{\sum_{s\in S} \exp\left\{\left(\sum_{j\sim i} \beta J \cdot 2(\mathbb{1}_{\{s=x_j\}})\right) - \frac{1}{2\sigma^2}(g_i - s)^2 - \beta q \mathbb{1}_{\{s\neq x_j\}}\right\}}\end{aligned} \tag{40}$$

where, again, S is the set of possible gray levels. Note that, at each step, all pixels are potentially updated.

The effect of the additional inertial terms $-\beta q \mathbb{1}_{\{s \neq x_j\}}$ in the exponent of each factor of the transition probabilities is to add a probabilistic price every time the value of a pixel changes. Since the potential (31) we are taking into account is only concerned with whether two pixels in the same neighborhood have the same value or not, the L_0 *norm* we consider for the inertial term seems to be appropriate. Note that this is not a restriction and other norms such as the L_1 or L_2 can be considered. These norms are, likely, more suitable when L_1 or L_2 norms appear in the prior. Then the probabilistic price to update the value of a pixel will be proportional to the *size* of the change.

6 Experimental Results

To test our algorithm, we considered the restoration of noisy images. In particular, the images we considered are, ideally, samples of a Markov Random Field (MRF) to which we added Gaussian Noise. Note that a MRF is a probability measure that can be specified via a Gibbs measure $\pi_{\text{prior}}(x) = \frac{e^{-\beta H(x)}}{Z}$ (see [16] for more details).

This is the same setup as the simpler cases considered in [16], which we considered suitable to make an initial assessment of the PCA approach to inverse problems.

To generate the MRF, we started a chain from a set of pixels randomly chosen from ℓ gray levels (equally spaced from 0, pure black to 1, pure white) and let it evolve according to the Gibbs Sampler algorithm with Gibbs measure

$$\pi_{\text{prior}} = \frac{e^{-\beta H(x)}}{Z} \tag{41}$$

with H as in (33) and $J = \frac{1}{3}$. We changed β manually during the evolution of the chain to obtain "not too noisy" original images. Then the degraded images have been obtained by adding independent normally distributed values (with mean zero and standard deviation σ) to each pixel and rounding the obtained result to the nearest gray level.

Both algorithms were run for a fixed number of steps (sweeps in the case of Gibbs Sampler) and we considered the last sample as the retrieved image. The aim is to perform a MAP estimate of the original image. We considered two images with 5, one with 9 gray levels, and one with 33 gray levels. We considered a noise with standard deviation 0.25 for the images with 5 gray levels, standard deviation 0.2 for the image with 9 levels and standard deviation 0.1 for the image with 33 gray levels. Note that, in all cases, the standard deviation of the noise is larger than the gap between two consecutive levels. All images are 256×256 pixels.

For both algorithms and all test cases, we performed 1000 steps with the following heuristics cooling schedule for β: we started with $\beta = 1.25$ and increased it by 0.25 every 250 steps. We set σ equal to the value of the added noise. For the PCA, we set $q = 0.51$.

The results of the retrieval are provided in Figures 2, 3, 4 and 5

Both algorithms have been evaluated in terms of the Structural similarity index measure (SSIM) and the Peak signal to noise ratio (PSNR).

If x is the original $m \times n$ image and $\tilde{x}$ is its noisy version, the PSNR of $\tilde{x}$ with respect to x is

$$\mathrm{PSNR}(\tilde{x}) = 20 \log_{10} \left(\frac{\max x}{\sqrt{\mathrm{MSE}(x, \tilde{x})}} \right) \tag{42}$$

where $\max x$ is the maximum pixel value of the original image,

$$\mathrm{MSE}(x, \tilde{x}) = \frac{1}{m \cdot n} \sum_{i=1}^{m} \sum_{j=1}^{n} \left[x_{(i,j)} - \tilde{x}_{(i,j)} \right]^2 \tag{43}$$

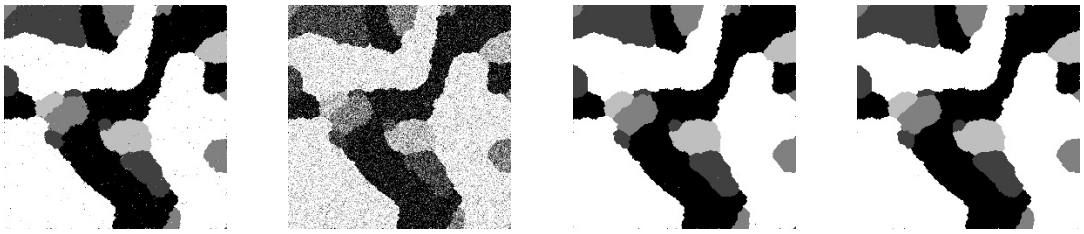
is the Mean Square Error of x and $\tilde{x}$, and $x_{(i,j)}$, $\tilde{x}_{(i,j)}$ are the luminances of the pixel with coordinates (i, j).

SSIM has been introduced in [33] to measure the *perceived similarity* between two images x and y and it is defined as

$$\mathrm{SSIM}(x, y) = \frac{(2\mu_x \mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{44}$$

where μ_x, μ_y are the sample means of x and y, σ_x, σ_y are their sample standard deviations and σ_{xy} is the sample covariance. Constants c_1 and c_2 depend, essentially on the number of possible gray levels and are meant to stabilize the results in the case of small denominators.

The obtained results are provided in Table 1. The results we obtained with the PCA appear to be slightly better than those obtained with the Gibbs Sampler. In particular, the SSIM of the PCA restored images is always higher than those of the images restored using the Gibbs sampler.



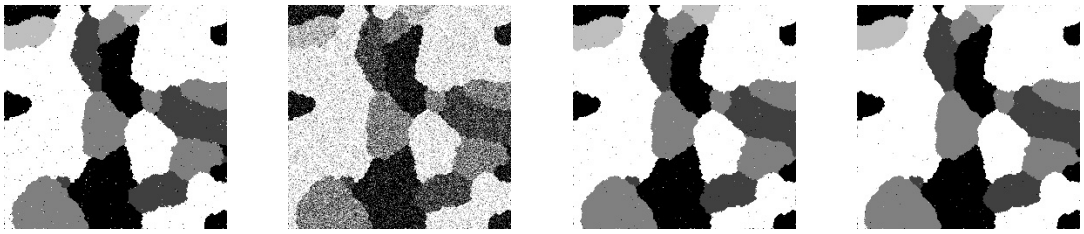
(a) Original image. **(b)** Noisy image. **(c)** Image denoised with Gibbs Sampler. **(d)** Image denoised with PCA.

Figure 2: Image mrf_n256_l5_i00 - 5 levels of gray; the standard deviation of the white noise is $\sigma = 0.25$. (First example)

7 Discussion of results and future developments

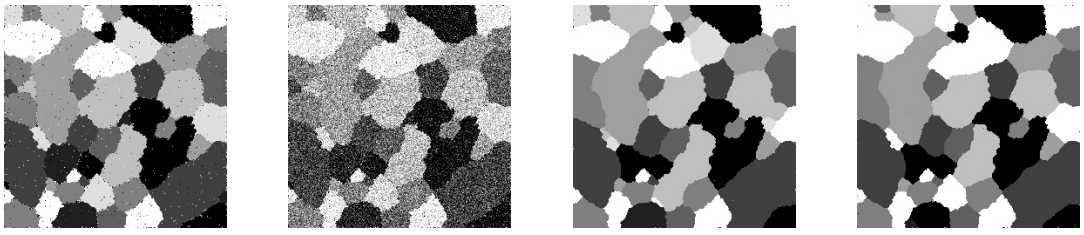
The paper aims to assess whether PCA are a viable option to tackle inverse problems. In this respect, we believe that the results we obtained support this hypothesis. In particular, we showed that it might be worth losing the exact control of the stationary measure of the Markov chain in order to use an inherently parallel algorithm. This work paves the road for PCA Markov Chain to the same type of evolution *sequential* Markov Chains Monte Carlo followed over the last decades. In this direction, we plan to further develop this research project along several lines

One of these lines concerns taking into account more realistic images together with more complex degradation types (such as multiplicative noise and combinations of noise and blur). To tackle



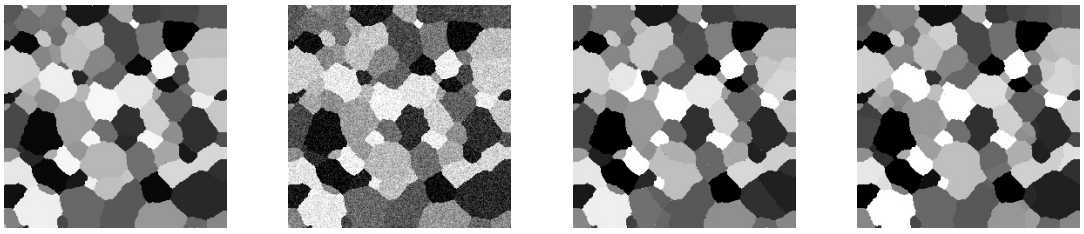
(a) Original image. **(b)** Noisy image. **(c)** Image denoised with Gibbs Sampler. **(d)** Image denoised with PCA.

Figure 3: mrf_n256_l5_i002 - 5 levels of gray, the standard deviation of the white noise is $\sigma = 0.25$. (Second example)



(a) Original image. **(b)** Noisy image. **(c)** Image denoised with Gibbs Sampler. **(d)** Image denoised with PCA.

Figure 4: mrf_n256_l9_i001 - 9 levels of gray; the standard deviation of the white noise is $\sigma = 0.20$.



(a) Original image. **(b)** Noisy image. **(c)** Image denoised with Gibbs Sampler. **(d)** Image denoised with PCA.

Figure 5: mrf_n256_l33_i001 - 9 levels of gray; the standard deviation of the white noise is $\sigma = 0.10$.

image id	N	σ	ℓ	**algo**	**SSIM**	**PSNR**
mrf_n256_l5_i001	256	0.25	5	GS	0.8388	24.2604
				PCA	0.8457	24.5715
mrf_n256_l5_i002	256	0.25	5	GS	0.7659	22.6702
				PCA	0.7721	22.9506
mrf_n256_l9_i001	256	0.20	9	GS	0.6146	20.3747
				PCA	0.6207	20.1542
mrf_n256_l33_i001	256	0.10	33	GS	0.8900	28.5466
				PCA	0.8945	28.9641

Table 1: SSIM and PSNR values for the images de-noised via the Gibbs Sampler (GS) and the Probabilistic Cellular Automaton (PCA)

these cases, a different prior distribution will have to be taken into account. In particular, this will require taking into account neighborhoods for the pixels that are more complex than the neighborhood of (32) (visually represented in Figure 1). Further, we should also consider an interaction between pixel that is more complex than that of described by (31) (which is, essentially an L_0 "norm") such as, for instance, L_1 and L_2 norms. Likewise, different norms should be considered for the inertial term of the posterior Hamiltonian (39) of the PCA.

A second line of research should focus on the computational aspects of the problem. In this work, our implementation of the PCA, though parallel from a theoretical point of view, is, in practice, sequential from the purely computational point of view (the new value of each pixel is stored in a temporary matrix which is reset as the current matrix at the end of each step). A true parallel implementation of the PCA where pixel values are updated with a GPU should be considered, and a comparison of the execution times with those of the Gibbs Sampler should be put forward. In this direction, results obtained with PCA in the context of combinatorial optimization, where the MCMC approach is also fruitfully applied, are rather encouraging (see [3, 15, 18, 25, 29, 31]).

Finally, we should put forward a comparison of the PCA approach with families of MCMC algorithms such as Hamiltonian Monte Carlo and Langevin Monte Carlo

Acknowledgment

All the authors are members of the Gruppo Nazionale per l'Analisi Matematica, la Probabilità e le loro Applicazioni (GNAMPA) of the Istituto Nazionale di Alta Matematica (INdAM). D. Costarelli and M. Piconi are also memebers of the UMI (Unione Matematica Italiana) group T.A.A. (Teoria dell' Approssimazione e Applicazioni), and of the network RITA (Research ITalian network on Approximation).

Funding

The authors have been supported within the project PRIN 2022 PNRR: "RETINA: REmote sensing daTa INversion with multivariate functional modeling for essential climAte variables characterization", funded by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGen-

erationEU, under the Italian Minister of University and Research MUR (Project Code: P20229SH29, CUP: J53D23015950001).

Author's contribution

All authors contributed equally to this work for writing, reviewing, and editing. All authors have read and agreed to the published version of the manuscript.

Conflict of interest/Competing interests

The authors declare that they have no conflict of interest and competing interests.

Copyright

The images have been generated by the authors.

Availability of data and material and Code availability

All the data generated for this study were stored in our laboratory and are not publicly available. Researchers who wish to access the data directly (including the ".tif" version of the images) can contact the corresponding author.

References

[1] Laura Angeloni, Domenico Daniele Bloisi, Paolo Burghignoli, Davide Comite, Danilo Costarelli, Michele Piconi, Anna Rita Sambucini, Alessio Troiani, and Alessandro Veneri. Microwave remote sensing of soil moisture, above ground biomass and freeze-thaw dynamic: Modeling and empirical approaches. *in print: Modern Mathematical Methods - arXiv preprint arXiv:2412.03523*, 2025.

[2] Valentina Apollonio, Roberto D'Autilia, Benedetto Scoppola, Elisabetta Scoppola, and Alessio Troiani. Criticality of measures on 2-d ising configurations: from square to hexagonal graphs. *Journal of Statistical Physics*, 177(5):1009–1021, 2019.

[3] Valentina Apollonio, Roberto D'Autilia, Benedetto Scoppola, Elisabetta Scoppola, and Alessio Troiani. Shaken dynamics: an easy way to parallel markov chain monte carlo. *Journal of Statistical Physics*, 189(3):39, 2022.

[4] Richard C Aster, Brian Borchers, and Clifford H Thurber. *Parameter estimation and inverse problems*. Elsevier, 2018.

[5] Anna Balenzano, Francesco Mattia, Giuseppe Satalino, and Malcolm WJ Davidson. Dense temporal series of c-and l-band sar data for soil moisture retrieval over agricultural crops. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 4(2):439–450, 2010.

[6] Mario Bertero, Patrizia Boccacci, and Christine De Mol. *Introduction to inverse problems in imaging*. CRC press, 2021.

[7] Alexandre Bouvet, Stéphane Mermoz, Thuy Le Toan, Ludovic Villard, Renaud Mathieu, Laven Naidoo, and Gregory P Asner. An above-ground biomass map of African savannahs and woodlands at 25 m resolution derived from ALOS PALSAR. *Remote Sens. Env.*, 206:156–173, Mar. 2018.

[8] Lin Chen, Chunying Ren, Bai Zhang, Zongming Wang, and Yanbiao Xi. Estimation of forest above-ground biomass by geographically weighted regression and machine learning with sentinel imagery. *Forests*, 9(10):582, 2018.

[9] Emilio Nicola Maria Cirillo, Vanessa Jacquier, and Cristian Spitoni. Metastability of synchronous and asynchronous dynamics. *Entropy*, 24(4):450, 2022.

[10] Emilio NM Cirillo, Francesca R Nardi, and Cristian Spitoni. Metastability for reversible probabilistic cellular automata with self-interaction. *Journal of Statistical Physics*, 132:431–471, 2008.

[11] Maria Paola Clarizia, Nazzareno Pierdicca, Fabiano Costantini, and Nicolas Floury. Analysis of cygnss data for soil moisture retrieval. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2227–2235, 2019.

[12] Paolo Dai Pra, Benedetto Scoppola, and Elisabetta Scoppola. Sampling from a gibbs measure with pair interaction by means of pca. *Journal of Statistical Physics*, 149:722–737, 2012.

[13] Roberto D'Autilia, Louis Nantenaina Andrianaivo, and Alessio Troiani. Parallel simulation of two-dimensional ising models using probabilistic cellular automata. *Journal of Statistical Physics*, 184:1–22, 2021.

[14] Roberto Fernández, Pierre-Yves Louis, and Francesca R Nardi. Overview: Pca models and issues. *Probabilistic Cellular Automata: theory, applications and future perspectives*, pages 1–30, 2018.

[15] Bruno Hideki Fukushima-Kimura, Satoshi Handa, Katsuhiro Kamakura, Yoshinori Kamijima, Kazushi Kawamura, and Akira Sakai. Mixing time and simulated annealing for the stochastic cellular automata. *Journal of Statistical Physics*, 190(4):79, 2023.

[16] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741, 1984.

[17] Olle Häggström. *Finite Markov chains and algorithmic applications*, volume 52. Cambridge University Press, 2002.

[18] Marco Isopi, Benedetto Scoppola, and Alessio Troiani. On some features of quadratic unconstrained binary optimization with random coefficients. *Bollettino dell'Unione Matematica Italiana*, pages 1–21, 2024.

[19] Jari Kaipio and Erkki Somersalo. *Statistical and computational inverse problems*, volume 160. Springer Science & Business Media, 2006.

[20] Yann H Kerr, Philippe Waldteufel, Philippe Richaume, Jean Pierre Wigneron, Paolo Ferrazzoli, Ali Mahmoodi, Ahmad Al Bitar, François Cabot, Claire Gruhier, Silvia Enache Juglea, et al. The smos soil moisture retrieval algorithm. *IEEE transactions on geoscience and remote sensing*, 50(5):1384–1403, 2012.

[21] Carlo Lancia and Benedetto Scoppola. Equilibrium and non-equilibrium ising models by means of pca. *Journal of Statistical Physics*, 153:641–653, 2013.

[22] Joel L Lebowitz, Christian Maes, and Eugene R Speer. Statistical mechanics of probabilistic cellular automata. *Journal of statistical physics*, 59:117–170, 1990.

[23] Kyle C McDonald and John S Kimball. 53: estimation of surface freeze–thaw states using microwave sensors. *Encyclopedia of Hydrological Sciences. John Wiley & Sons, Inc., Hoboken, NJ*, pages 1–15, 2005.

[24] Isabella Mereu, Mariarosaria Natale, Michele Piconi, Alessio Troiani, Vincenzo Suriani, Domenico Daniele Bloisi, Paolo Burghignoli, Danilo Costarelli, Alessandro Veneri, Davide Comite, et al. Interpolation theory and artificial intelligence. a roadmap for satellite data augmentation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, pages 1–28, 2025.

[25] Daiki Okonogi, Satoru Jimbo, Kota Ando, Thiem Van Chu, Jaehoon Yu, Masato Motomura, and Kazushi Kawamura. Apc-sca: A fully-parallel annealing algorithm with autonomous pinning effect control. In *2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 414–420, 2022.

[26] Simonetta Paloscia, Paolo Pampaloni, Simone Pettinato, and Emanuele Santi. A comparison of algorithms for retrieving soil moisture from envisat/asar images. *IEEE Transactions on Geoscience and Remote Sensing*, 46(10):3274–3284, 2008.

[27] Aldo Procacci, Benedetto Scoppola, and Elisabetta Scoppola. Probabilistic cellular automata for low-temperature 2-d ising model. *Journal of Statistical Physics*, 165:991–1005, 2016.

[28] Kimmo Rautiainen, Davide Comite, Juval Cohen, Estel Cardellach, Martin Unwin, and Nazzareno Pierdicca. Freeze–thaw detection over high-latitude regions by means of gnss-r data. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–13, 2021.

[29] Benedetto Scoppola and Alessio Troiani. Gaussian mean field lattice gas. *Journal of Statistical Physics*, 170:1161–1176, 2018.

[30] Benedetto Scoppola, Alessio Troiani, and Matteo Veglianti. Shaken dynamics on the 3d cubic lattice. *Electronic Journal of Probability*, 27:1–26, 2022.

[31] Alessio Troiani. Probabilistic cellular automata monte carlo for the maximum clique problem. *Mathematics*, 12(18):2850, 2024.

[32] Alessandro Veneri, Vincenzo Suriani, Alessio Troiani, Domenico Daniele Bloisi, Paolo Burghignoli, Danilo Costarelli, Isabella Mereu, Mariarosaria Natale, Michele Piconi, and Davide Comite. A review on retrieval methods for microwave remote sensing of soil moisture, above-ground biomass, and freeze-thaw dynamic. *10.36227/techrxiv.174000579.94856636/v1*, 2025.

[33] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.