Nearly accurate solutions for Ising-like models using Maximal Entropy Random Walk

Jarek Duda
Jagiellonian University, Golebia 24, 31-007 Krakow, Poland, Email: dudajar@gmail.com

Abstract—While one-dimensional Markov processes are well understood, going to higher dimensions there are only a few analytically solved Ising-like models, in practice requiring to use relatively costly, uncontrollable and inaccurate Monte-Carlo methods. There is discussed analytical approach for e.g. width $\times \infty$ approximation of lattice, also exploiting Hammersley-Clifford theorem to generate random Gibbs/Markov field through scanning line-by-line using local statistical model as in lossless image compression. While its conditional distributions could be found with Monte-Carlo methods, there is discussed use of Maximal Entropy Random Walk (MERW) to calculate them from approximation of lattice as infinite in one direction and finite in the remaining. Specifically, in the finite directions there is built alphabet of all patterns, then transition matrix containing energy for all pairs of such patterns is built, from its dominant eigenvector getting probability distribution of pairs of patterns, which can be translated into local statistical model for line-by-line scan. Such inexpensive models, requiring seconds on a laptop for attached implementation and directly providing probability distributions of patterns, were tested for mean entropy and energy per node, getting maximal $\approx 0.02$ error from analytical solution near critical point, which quickly improves to extremely accurate e.g. $\approx 10^{-10}$ error for $J \approx 0.2$.

Keywords: information theory, statistical mechanics, Markov fields, Ising model, Gibbs field, Hammersley-Clifford theorem, Monte-Carlo, lossless image compression

I. INTRODUCTION

Gibbs fields are the basic models of statistical mechanics, condensed matter physics. In 1971 through Hammersley-Clifford theorem [1], they turned out equivalent with providing different perspective Markov fields, stimulating both fundamental information theory research as being interesting topic very far from complete understanding, but also having potential for new practical tools and applications especially around image analysis, segmentation, compression [2], [3], or maybe for storing information in constrained media like Bit Patterned Media Recording [4].

As there are only few models with known analytical solution like zero-field 2D Ising model [5] awarded Nobel prize, in practice there are mainly used Monte-Carlo methods [6] which are costly to provide uncorrelated fields, agreement with desired probability distribution, high accuracy for estimated properties, can get trapped in local minima preventing from exploring full configuration space [7].

Hence it is valuable to develop intermediate methods e.g. analytical for approximated problem, like discussed here visualized in Fig. 1, which combines two concepts that can

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{Top: we focus on Gibbs fields - defined by Boltzmann probability distribution $Pr(s) \propto e^{-\beta E(s)}$ of 2D lattice configurations $s : \mathbb{Z}^2 \rightarrow A$ with short-range interactions: energy $E$ being sum of functions of values in each node and edge (visualized with orange color). By Hammersley-Clifford theorem, Gibbs field is equivalent with Markov field: in which conditional probability distributions based on values in some region (blue), depend only on values in the boundary of this region (green). Below: we use the presented MERW formula to find probability distribution of $\{u, v\}$ pairs of neighboring patterns of finite width - getting analytical solution for Boltzmann distribution among their infinite sequences in vertical direction. Summing over the "*" positions, and dividing by sum over "?" values, we get the model: probability of '?' value based on a few neighboring values marked green. We can use such calculated local conditional probability distribution model e.g. to generate random field through such scanning line-by-line, calculate parameters like mean entropy or energy per node, probability distribution of patterns, data compress such configuration or store information in it. Bottom: errors (differences from accurate values) of calculated entropy (left) and energy (right) per node found this way for various $J = J_0 = J_1$ coupling parameters ($\beta = 1, \mu = 0$), using various model size (up) and widths used to calculate the model (down). These errors reach $\approx 0.02$ near critical point $J \approx 0.44$, but reduce orders of magnitude if moving away from this point. We can see characteristic beak near critical point, which might be useful to localize it in general case.}
\end{figure}
be also applied separately:

- Markov fields allow to simplify calculation of conditional probability distribution to be based only on local situation, what is very convenient from perspective of generating them by scanning line-by-line as in lossless image compression like LOCO-I [8].
- Such model of local conditional distributions requires to know probability distribution of patterns - they could be found with Monte-Carlo, but it is costly to get high accuracy this way. Here they are found with Maximal Entropy Random Walk (MERW) providing analytical solution for approximation of lattice to infinite in one direction and finite in the remaining.

Presented methodology was originally introduced [9] by the author in 2006 for 2D Fibonacci coding (Hard Square) problem - for storing information in lattice of bits where it is forbidden to use two neighboring ’1’s, what could e.g. allow to improve HDD capacity. Statistical model found with MERW allowed to work \( \approx 10^{-9} \) bits/node (for \( a = b = 5 \) here) below theoretical \( 0.5878911617753406 \) bits/node [10] entropy threshold, and first Asymmetric Numeral Systems were introduced there for such (reversed) entropy coding purpose. This article expands it from uniform distribution among allowed patterns into Boltzmann distribution for general Ising-like models. Appendix contains example implementation.

II. Problem and Methodology

A. Gibbs fields with Ising model as practical example

Information theoretic discussion of the proposed methodology can be found in [9] from perspective of 2D Fibonacci/Hard Square problem forbidding some patterns. Here we would like to expand it to more general Boltzmann distribution of patterns in Gibbs field.

We focus on (infinite) \( d \)-dimensional lattice \( X = \mathbb{Z}^d \), being interested in its space of configurations: \( s : X \to \mathcal{A} \) for some alphabet \( \mathcal{A} \), e.g. \( \{0,1\} \) or \( \{-1,1\} \). We focus on discrete alphabets, but there are also considered continuous.

We would like to consider Boltzmann distribution among such configurations:

\[
\Pr(s) \propto \exp(-\beta E(s)) \tag{1}
\]

where \( \beta \) can be fixed here to \( \beta = 1 \), \( E(s) \) is called energy of configuration and usually is defined in a translationally invariant way, for example in popular 2D Ising model we will focus on here: \( \mathcal{A} = \{-1,1\} \), energy \( E(s) = -\mu \sum_{x,y \in \mathcal{N}} s_{x,y} - J_h \sum_{x,y \in \mathcal{N}} s_{x,y}s_{x+1,y} - J_v \sum_{x,y \in \mathcal{N}} s_{x,y}s_{x,y+1} \) \tag{2}

where \( \mu \) corresponds to external field, \( J_h, J_v \) are horizontal and vertical coupling constants, usually the same: the presented tests are for \( J_h = J_v = J, \beta = 1, \mu = 0 \).

Some properties of interest to be found are average entropy, energy, value per node - there are known analytical formulas for \( \mu = 0 \) case ([5], [10]) used in tests here. As the lattice is infinite, such local averages can be defined by limit of averages from finite size lattices. Here it is avoided by imagining that all values were chosen by translationally invariant conditional probability distribution model: for line-by-line scanning, what allows to obtain averages from probability distribution of its local situations.

From information theory perspective it is also valuable to be able to use the presented approach to calculate probability distributions of patterns for such ensembles, what defines their Markov type [11].

All such questions can be approximately answered and in practice usually are using Monte-Carlo methods like Metropolis-Hastings [9]. They generate random fields on finite lattice e.g. with cyclic boundary conditions by repeating large number of times: take random position and randomly modify its value or not accordingly to calculated probability depending on its neighbors. While we have certainty of asymptotically getting field from the assumed probability distribution, in practice only finite numbers of steps are used - bringing difficult questions of probability distribution it has actually used and autocorrelations with the previously generated field, in addition to contribution of its approximation with finite lattice.

Additionally, even if being from perfect desired distribution, estimation of probability distribution of patterns from random samples has error decreasing with square root of the sample size - making it impractical to get e.g. \( \approx 10^{-10} \) accuracy, which often can be inexpensively achieved with the discussed approach as in evaluation in Fig. [7].

B. Hammersley-Clifford theorem and Markov property

While Gibbs fields seem not very convenient to understand local statistical behavior, fortunately through Hammersley-Clifford theorem [11] they turn out equivalent with Markov fields, which generalize property from Markov processes into multidimensional case, can be written e.g. as:

\[
\Pr(s_{x,y} | X \setminus \{x,y\}) = \Pr(s_{x,y} | \mathcal{N}(x,y)) \tag{3}
\]

where \( \mathcal{N} \) is neighborhood accordingly to the used interaction/constraints in energy formula, e.g. 4 neighbors in (2) of Ising model

This is local Markov property, visualized in the left diagram in top of Fig. [2] Applying it multiple times we get global formulation from the middle diagram: conditional distribution in some set of nodes depends only of values at its boundary according to the used interaction/constraints.

Finally we can also use it for scanning (right diagram): generate random field through taking succeeding line-by-line random values like in lossless image compression. Markov property says that conditional distribution based on already fixed values in fact depends only on the boundary values. In \( \mathbb{Z}^2 \) lattice this boundary is infinite, but we can approximate using local: a few values before and after - correspondingly \( b \) and \( a \) values as in this diagram.
For such fixed $b$ values before and $a$ after, we can put the entire behavior into size $|A|^{a+b}$ table - of '?' conditional probability based on all $a+b$ values. To find such model we need to estimate probability distribution of all size $a+b+1$ patterns including '?', then divide it by sum over all values of '?'. In Fig. 1 error plot we can see that while calculated entropy strongly depends on such $a, b$ context size, energy nearly does not.

We could find such pattern distributions with Monte-Carlo, but it would be costly to get decent accuracies this way - here we use MERW instead.

C. Maximal Entropy Random Walk (MERW)

MERW ([9], [12]) analytically solves general 1D problem, is further applied to graph of all width $w$ patterns. Originally for a graph adjacency matrix $M$ it provides stochastic matrix assuming uniform distribution of paths on this graph, or equivalently maximizing entropy rate. As sketched below, it can be generalized from uniform to Boltzmann distribution [13] by replacing adjacency matrix

$$M_{uv} = \exp(-\beta(E_u/2 + E_{uv} + E_v/2))$$

(4)

where $E_u$ is energy of vertex/pattern $u$, $E_{uv}$ of $u-v$ edge/interaction. In statistical physics it is referred as transition matrix [10] and averaged properties are calculated from its eigenvalues. Here we would like to find stochastic model, what requires its dominant eigenvector instead:

$$M\psi = \lambda\psi$$

for maximal $|\lambda|$, $\sum \psi_i^2 = 1$ (5)

From Perron-Frobenius theorem this eigenvector should be unique and can be chosen as real, allowing to approximate $M^p \approx \lambda^p \psi\psi^T$ for large powers $p$. From the other side from [4], $M^p$ can be seen as sum (partition function) over all length $2p$ paths using Boltzmann distribution. Fixing their central position to $u$ and performing $p \to \infty$ limit, we get probability distribution of patterns $Pr(u) \propto (\psi_u)^2$. Analogously fixing its neighboring two central positions to $u, v$ and performing $p \to \infty$ limit we get $Pr(u,v) \propto \psi_u M_{uv} \psi_v$. Normalizing them to sum to 1, thanks to $\sum \psi_i^2 = 1$ and $M\psi = \lambda\psi$, we get probability distributions for one and two neighboring vertices/patterns:

$$Pr(u) = (\psi_u)^2$$

$$Pr(u,v) = \psi_u \frac{M_{uv}}{\lambda} \psi_v$$

(6)

D. Calculating the model

Formula (6) allows us to calculate probability distribution of pairs of neighboring values assuming Boltzmann distribution among their infinite sequences.

To apply it to a multidimensional lattice case, we can approximate the remaining directions with finite width $w$ like in Fig. 1 getting random walk in large size $|A|^w$ set of states $A^w$. We can assume cyclic boundary conditions - approximating $\mathbb{Z}^2$ lattice with infinite length cylinder surface. Otherwise we approximate with $w \times \infty$ rectangle.

Now for the assumed interaction model we need to calculate individual energy of each $A^w$ pattern, and interaction energy between all their pairs, constructing transition matrix [11]. Calculating its dominant eigenvector we can find probability distribution of pairs $Pr(u,v)$ using (6).

Now we can calculate the model by summing over values in unused positions - marked by stars in Fig. 1 and dividing by sums over the searched position '?'. Appendix contains optimized implementation of this procedure in Wolfram Mathematica, Fig. 2 contains examples of such models.

E. Applying the scanning model

Discussed model provides conditional probability distribution for the currently considered position, based on local already chosen values in scanning line-by-line.

For low dimension and short range interactions it can
be inexpensively calculated with discussed MERW-based approach, providing nearly accurate values unless being close to a critical point. In remaining cases we can always search for such model with Monte-Carlo, however, getting high accuracy might become computationally very costly.

Having such model, a basic application is calculating average energy, entropy, value per node, e.g. by averaging such properties for currently chosen value over estimated probability distribution of considered size $a + b$ context - e.g. as realized in Appendix.

Another application is generation of random field using such model through line-by-line scan, for boundary values using model with reduced context. As being generated from scratch, we get practically uncorrelated configuration this way. There remains question of its agreement with assumed distribution, in any case there can be later applied some reduced number of Monte-Carlo steps to improve this distribution - details of savings which can be obtained this way need further investigation.

Other possible applications of such scanning models is data compression of random fields, or storing information in such constrained media by using entropy coder instead of taking random values, as in the original motivation for the discussed approach [9].

F. Alternative MERW calculation of model

MERW formulas [6] for probability distribution of symbols/patterns and their pairs can be analogously extended to formulas for longer: length $l$ sequences of symbols/patterns by using $l - 1$ appearances of $M/\lambda$ matrix with corresponding indexes:

$$\Pr(u_1u_2 \ldots u_l) = \psi_{u_1} \frac{M_{u_1u_2}}{\lambda} \frac{M_{u_2u_3}}{\lambda} \ldots \frac{M_{u_{l-1}u_l}}{\lambda} \psi_{u_l} \tag{7}$$

In practice we might want to allow multiple patterns in above intermediate indexes, e.g. all with fixed '+1' in some node. This summation can be generally written by building $\Pi_l$ matrices with '1' on diagonal for all allowed cases for given position, and zeros beside:

$$\Pr(\Pi_1 \ldots \Pi_l) = \psi^T \Pi_1 \frac{M}{\lambda} \Pi_2 \frac{M}{\lambda} \ldots \frac{M}{\lambda} \Pi_l \psi \tag{8}$$

Considered various $\Pi_l$ for a given position should sum over all alternatives to identity matrix. The $\psi$ at the ends can be imagined as a result of infinite sequence of multiplication of $M/\lambda$, of propagators from both infinities.

We could use the above formula for alternative calculation of model parameters: by placing considered nodes (green dots and '?' in Fig. [1]) in vertical instead of horizontal way, and using [3] formula for $l = a + b - 1$ length.

Intuitively it could lead to more accurate probabilities thanks to being further from boundaries, however, it would require many multiplication operations for very large dense matrices - what is computationally very expensive.

III. DISCUSSION AND FURTHER WORK

There was discussed perspective for Gibbs/Markov fields as being generated through scanning line-by-line like in lossless image compression. Hammersley-Clifford theorem allows to optimize the necessary context for conditional probability distributions. MERW-based approach allows to inexpensively calculate quite accurate models at least in some situations. Having such model we can e.g. estimate parameters, probability distributions of patterns, inexpensively generate nearly uncorrelated random fields, data compress them or store information in such constrained media.

While this approach can be adapted to various short range interactions in 2D (also long in pattern direction), it quickly becomes impractical for longer range interactions and higher dimensions due to exponentially growing space of possible patterns. It might be possible to overcome e.g. by working on some spaces of features or classes of abstraction of possible patterns. Even more difficult question regards continuous case, requiring to model continuous conditional probabilities this way, what could be handled by decomposing into mixed moments and their relations [13].

REFERENCES

[1] J. M. Hammersley and P. Clifford, “Markov fields on finite graphs and lattices,” Unpublished manuscript, Vol. 46, 1971.
[2] C. A. Bouman, K. Sauer, and S. Saquib, “Markov random fields and stochastic image models,” 1995.
[3] Z. Kato, J. Zebuia et al., “Markov random fields in image segmentation,” Foundations and Trends in Signal Processing, Vol. 5, No. 1–2, pp. 1–155, 2012.
[4] H. Richter, A. Dobin, O. Heinonen, K. Gao, R. vd Veerendonk, R. Dinch, D. Weller, P. Asselion, M. Erden et al., “Recording on bit-patterned media at densities of 1 Tbit/in2 and beyond,” IEEE Transactions on Magnetics, Vol. 42, No. 10, pp. 2255–2260, 2006.
[5] L. Onsager, “Crystal statistics. i. a two-dimensional model with an order-disorder transition,” Physical Review, Vol. 65, No. 3–4, p. 117, 1944.
[6] K. Binder, D. M. Ceperley, J.-P. Hansen, M. Kalos, D. Landau, D. Levesque, H. Mueller-Krumbhaar, D. Stauffer, and J.-J. Weis, Monte Carlo methods in statistical physics. Springer Science & Business Media, 2012, vol. 7.
[7] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” science, Vol. 220, No. 4598, pp. 671–680, 1983.
[8] M. J. Weinberger, G. Seroussi, and G. Sapiro, “The loco-I lossless image compression algorithm: Principles and standardization into jpeg-Is,” IEEE Transactions on Image Processing, Vol. 9, No. 8, pp. 1309–1324, 2000.
[9] J. Duda, “Optimal encoding on discrete lattice with translational invariant constraints using statistical algorithms,” arXiv preprint arXiv:0710.3861, 2007.
[10] R. J. Baxter, Exactly solved models in statistical mechanics. Elsevier, 2016.
[11] Y. Baryshnikov, J. J. Duda, and W. Szpankowski, “Types of markov fields and tilings,” IEEE Transactions on Information Theory, Vol. 62, No. 8, pp. 4361–4375, 2016.
[12] Z. Barda, J. Duda, J.-M. Luck, and B. Wacław, “Localization of the maximal entropy random walk,” Physical review letters, Vol. 102, No. 16, p. 160602, 2009.
[13] J. Duda, “Extended maximal entropy random walk,” Ph.D. dissertation, Ph. D. dissertation, Jagiellonian University, 2012.[Online]. Available: https://fasia.uj.edu.pl/documents/41628/de3e0b7-e7b1-4eba-8a5a-d97f4256fd065, 2012.
[14] J. Duda, R. Syrek, and H. Gurgul, “Modelling bid-ask spread conditional distributions using hierarchical correlation reconstruction,” arXiv preprint arXiv:1911.02361, 2019.
This Appendix contains optimized Wolfram Mathematica implementation of discussed method, also calculation of average entropy, energy and magnetization, and analytical formulas to find their exact values for tests. For width=13 it needs about 3 seconds, for width=14 about 15 seconds. For higher widths memory requirements quickly grows due to the needed dense $2^w \times 2^w$ size matrix \texttt{prob} which is used for both transition matrix and pair probabilities here.

\begin{verbatim}
(* findprob - find Pr(u,v) of pairs of stripes *)
(* needs Jv, Jh, mu, beta, width, cyclic *)
findprob := ( patn = Power[2, width];
    pat = Table[2 IntegerDigits[i,2,width]-1.,{i, 0, patn-1}];
    tmp = Transpose[pat]; If[cyclic, AppendTo[tmp, tmp[[1]]]]; 
    prodt = Transpose[tmp[[1 ;; -2]]*tmp[[2 ;; -1]]];
    paten = Table[-Jh*Total[prodt[[i]]]-mu*Total[pat[[i]]],{i, patn}];
    (* calculate transfer matrix as prob: *)
    prob = KroneckerProduct[Table[1., patn], paten/2];
    prob += Transpose[prob] - pat.Transpose[Jv*pat];
    prob = Exp[(-beta)*prob];
    {{lam}, {psi}} = Abs[Eigensystem[prob, 1, Method -> "Arnoldi"]];
    prob *= KroneckerProduct[psi/lam, psi];)

(* find model: pr. of '?' based on bef 'b's before *)
(* [mid-bef,mid-1] and aft 'a's after [mid,mid+aft-1]: *)
(* *********** previous pattern (i) in (ij) pairs *)
(* ************ next pattern (j) *)
h[p_] := -p*Log[2, p] - (1-p)Log[2, 1-p];
findmodel := (tpat = Round[Transpose[pat]]/2 + 1/2;
    mid = Ceiling[width/2 + 1];
    ipat = Table[Power[2,i],{i, bef, 0, -1}].tpat[[mid-bef;;mid]] + 1;
    jpat = Table[Power[2,i],{i, aft-1, 0, -1}].tpat[[mid;;mid+aft-1]] + 1;
    rprob = SparseArray[Table[{ipat[[i]], i} -> 1., {i, patn}]].prob.
    SparseArray[Table[{j, jpat[[j]]} -> 1., {j, patn}]].prob;
    cprob = rprob[[1 ;; -1 ;; 2]] + rprob[[2 ;; -1 ;; 2]];
    model = rprob[[2 ;; -1 ;; 2]]/cprob; (* find the model *)
    H = Total[Total[cprob*h[model]]]; (* entropy per node *)
    mag = Total[Total[cprob - rprob][[1 ;; -1 ;; 2]]]; * magnetization *)
    U = Total[Total[rprob - rprob[[1 ;; -1 ;; 2]]]*cprob];
    {U, H, mag});

(* calculates accurate values for U and H *)
accUH[J_, beta_: 1] := (If[J == 0, {0., 1.},
    k = 1/Pow[Sinh[2*beta*J], 2];
    U = -k*Log[2]/(2*beta) - (1/2*beta) + k;
    F = -Log[2]/(2*beta) - (1/2*beta) - k*Log[2]/(2*beta);
    {U, beta*(U - F)/Log[2]})

(* example of application *)
Jv = Jh = 1; mu = 0; beta = 1;
width = 10; cyclic = True;
findprob;
bef = 3; aft = 3;
Print[accUH[Jv, beta], " accurate {U,H}, found:"]
findmodel (*return {U,H,magnetization}*)
\end{verbatim}