Critical Temperature and Critical Exponents from Monte Carlo Methods and from Machine Learning

Dimitra Pefkou[1]

[1] *Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.*

The phase transition classification of physical systems can be studied in several ways, including Markov Chain Monte Carlo methods and machine learning neural networks. We study both approaches in order to classify the phase of a $2D$ Ising model, and to calculate its unitless critical temperature and the critical exponent ν. The Monte Carlo method results are $T_c = 2.285(52)$ and $\nu = 0.74(39)$, and the neural network results are $Tc = 2.239(39)$ and $\nu = 0.63(40)$, thus both approaches yield results of similar accuracy.

I. INTRODUCTION

When a physical system transitions from a distinct state to another state with different physical properties, it undergoes a phase transition. The phase transition occurs at a specific critical value of the relevant varying parameter, which for many systems is the temperature T. The behavior of the system near the critical point T_c is described by a set of critical exponents, for example its correlation length scales like $t^{-\nu}$, where $t = \frac{T-T_c}{T_c}$.

The critical temperature and critical exponents can be analytically obtained only for a few simple physical systems in low dimensions. For that reason several numerical and computational methods have been developed to study the phase transitions of complex systems. The most important are Markov Chain Monte Carlo (MCMC) methods, which allow one to efficiently generate sets of uncorrelated configurations of a system using importance sampling, e.g by the Metropolis-Hastings algorithm [1]. Then one can estimate the expectation values of physical quantities by replacing the integration over values weighted by their Boltzmann weight with a simple sum over their measurements in the generated ensemble of configurations. The critical temperature and exponents can then be obtained via several different approaches, including the finite scaling method [2].

MCMC methods are very effective to study the phase transition of systems that have an order parameter, however, they are not always useful for e.g a system without an order parameter. In recent years, machine learning (ML) methods have been utilized in relation to critical systems [3]. Our particular problem of finding the critical temperature of a phase transition can be readily be described as a classification problem in ML.

In this work we explore a simple MCMC approach to obtain the critical temperature and critical exponent ν of the 2D Ising model. We then follow the work of Ref. [3] and use a neural network (NN) to train a model on a set of Ising model configurations to classify the phase of the configuration and estimate the critical temperature and exponent ν. Finally, we explore whether the model trained on the Ising model can correctly classify 2D configurations of the 3-state Potts model, also obtained by MCMC simulation.

II. ISING AND POTTS MODELS

The archetype system that demonstrates a second-order phase transition is the Ising model for N 2-state spins σ in d dimensions, which is described by the Hamiltonian

$$H = -J \sum_{<ij>} \sigma_i \sigma_j, \tag{1}$$

with $\sigma = \{-1, 1\}$ and we sum over the nearest neighbors of each spin. The Ising model has been thoroughly studied and the its critical temperature can been obtained in the thermodynamic limit $N \to \infty$ for several dimensions d. For $d = 2$ which is the focus of this work, the exact dimensionless critical temperature can be obtained analytically and is equal to $T_c = \frac{J}{\ln(1+\sqrt{2})} \approx 2.27$.

The Potts model can be considered a generalization of the Ising model for different spins. Its Hamiltonian is given by

$$H = -J_P \sum_{<ij>} \delta_{s_i, s_j} \tag{2}$$

δ_{s_i,s_j} is the Kronecker delta and the sum is over nearest neighbors. The simplest case is the 3-state spin, which in 2D it has dimensionless critical temperature $T_c \approx 1.00$.

III. MONTE CARLO SIMULATION

For this project, we investigate 2D square lattices of length $L = \{6, 12, 18, 24, 30\}$ at 100 different dimensionless temperatures T equally spaced in the range $[1., 7.]$ for the Ising model and in $[0.2, 4.2]$ for the Potts model. For each lattice size and temperature, we generate an ensemble of 3000 configurations for the Ising model and 1000 for the 3-state Potts model, using a simple Metropolis-type algoritm:

- Start with a lattice with randomly assigned spin values at each site.
- For each site $\{i, j\}$, propose a spin flip $\sigma_{i,j} \to -\sigma_{i,j}$.
- If the energy change associated with this spin flip $\Delta E < 0$, accept it and update the spin. If $\Delta E > 0$, accept it with probability $e^{-\Delta E/T}$.

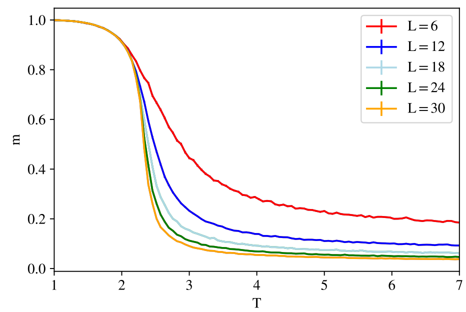

FIG. 1: Magnetization as a function of unitless temperature. The error bars are obtained via bootstrap resampling.

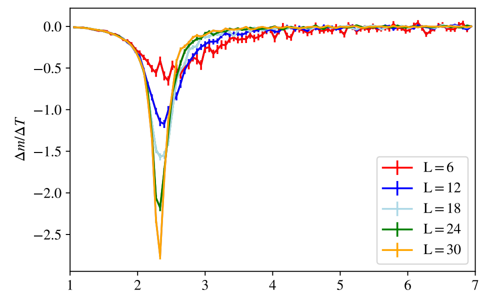

FIG. 2: Magnetization gradient as a function of unitless temperature. The error bars are obtained via bootstrap resampling.

- Repeat with new lattice.

We allow our ensemble generation to thermalize for 1000 steps before starting to save the configurations. The configurations were saved every 10 steps in order to reduce their correlation with each other.

We estimate the critical temperature of the Ising model in the thermodynamic limit by investigating the average magnetization per spin m, which can be obtained by averaging over the spins of each configuration, and then averaging the absolute values over the entire ensemble. Since we have a limited sample, to estimate the error of m we use the standard resampling method of bootstrapping, with 200 bootstraps of 3000 configurations each (Fig. 1). The transition temperature T_0 demonstrates finite size scaling, which we can utilize to extract T_c via

$$T_0(L) = T_c(1 + x_0 L^{-1/\nu}). \tag{3}$$

where $T_0(L)$ is taken to be the bootstrap averaged values at which the absolute value of the derivative of m, expressed as

$$\frac{\Delta m(T)}{\Delta T} = \left| \frac{m(T+\delta T) - m(T-\delta T)}{2\delta T} \right| \tag{4}$$

and shown in Fig. 2, is maximum. The parameters T_c, ν and x_0 are then fit by least squares regression.

We obtain $T_c = 2.285(52)$ and $\nu = 0.74(39)$. Even though we find T_c to be really close to its real value, we are not able to constrain ν, which is analytically predicted to be 1, as well. That is likely due our small number of L values considered and the inefficiently large spacing in temperature, as well as unaddressed systematics related to the choice of a single T_0 value for each L. In Fig. 3, the m scaling functions "collapsing" on top of each other near the critical point visually confirm that our results are adequate.

IV. NEURAL NETWORK

We now use our ensemble to train a feed-forward neural network (NN) to classify the phases of the Ising model.

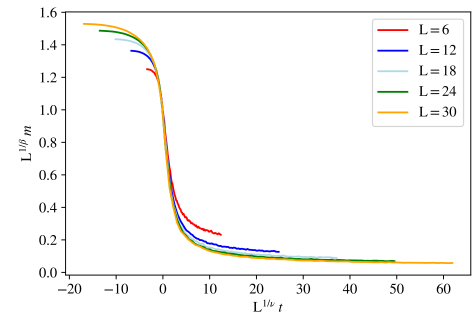

FIG. 3: Collapsed magnetization scaling functions.

We use the same NN as in Ref. [3], implemented using Pytorch [4]. The input is the flattened L spin configurations, and the net (Fig. 4) consists of a single fully-connected linear layer of A units followed by a sigmoid activation function

$$\sigma(z) = \frac{1}{1+e^{-z}} , \tag{5}$$

and an output linear layer of two neurons followed by a final sigmoid function. Each of the output neurons is associated with one of the two classes of our system. The final sigmoid layer normalizes the sum of the output of the two neurons to be equal to 1. The training labels are given in the form of one-hot encoding, with configurations of temperature under T_c being labeled as $[1, 0]$ and configurations above T_c as $[0, 1]$.

We find that varying the number of units of the middle layer and the learning rate had no observable effect on the result, and therefore we set $A = 100$ and learning rate $= 0.01$. The choice of loss function that is minimized at each epoch to update the weights of the NN is found to be crucial. Most choices for a loss function result in numerical instabilities of the classification. The most stable results are obtained by removing the final sigmoid activation function from the NN and including it in the loss function, and then acting with a final sigmoid function on the results. In this way, the loss function for output

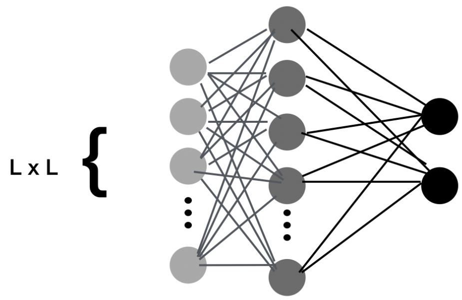

FIG. 4: Schematic of the neural net used.

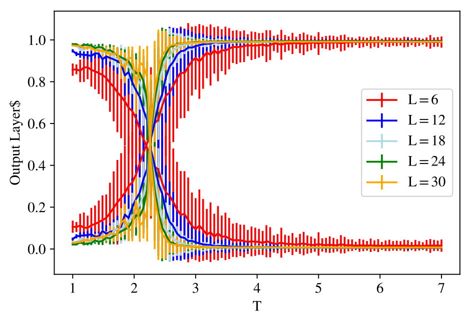

FIG. 5: The results of the output layers with varying temperature for a test set. The error bars shown are the standard deviations over the test set.

x and expected output y can be written as

$$l(x, y) = -[y \ln \sigma(x) + (1 - y) \ln(1 - \sigma(x)] \ . \tag{6}$$

We use the Adam optimizer [5] for the training, and find that at least 10000 epochs are needed for a successful classification of a testing set drawn randomly from the remaining configurations. The training size was not crucial for the classification, but it was for obtaining a fitted value for T_c.

Our results for the two nodes of the output layer averaged over a testing set of size 500 are shown in Fig 5. The model is successful in classifying the two phases, however the signals are much noisier than those in Ref.[3], which is likely due to the limited size of our ensemble, the wider spacing of temperatures and the lower L range we use. We get an estimate of the transition temperature at each length by considering the temperature at which the two outputs cross for each L and configuration, and taking the mean over all configurations in the test set. Fitting Eqn. 3, we get $T_c = 2.239(39)$ and $\nu = 0.63(40)$. We see that despite the fluctuation of the signals, the critical value estimates are of similar accuracy as in the MC approach.

Finally, we attempt to use the NN trained in 1500 Ising model configuration to classify the Potts model, however the results are unsuccessful (Fig. 6). Since one of the main motivations of this approach is to use models

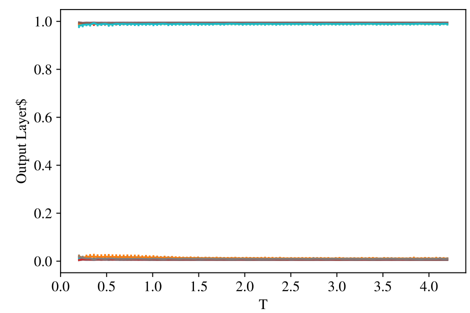

FIG. 6: The NN trained on the Ising model is unsuccessful at classifying the 3-state Potts model that has critical temperature close to 1.

trained in simple well-understood physical systems in order to predict the phase transition of more complex systems like those not described by an order parameter, we note that for the two models explored here the weights of the simpler model did not correctly classify the more complex model. This method has been successfully applied to other systems though, like e.g the triagonal Ising model in Ref. [3], so it is possible that our model was not sufficiently trained and that this is not a fundamental limitation.

V. CONCLUSION

We explored two different approaches of classifying the phase of a $2D$ Ising model, traditional Monte Carlo methods and artificial neural network. We found that for the limited size of out data set, they were both successful at predicting the critical temperature and the critical exponent ν with about the same accuracy. We attempted to use the model trained on the Ising configurations in order to classify the phase transition of the Potts model configurations, but were unsuccessful, likely due to inefficient training and limitations of our data sets.

[1] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[2] M. E. J. Newman and G. T. Barkema. *Monte Carlo methods in statistical physics.* Clarendon Press, Oxford, 1999.

[3] Juan Carrasquilla and Roger G. Melko. Machine learning phases of matter. *Nature Physics*, 13(5):431–434, Feb

2017.

[4] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[5] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.