Utrecht University

Master Thesis

Unsupervised Learning of Physical Models: Uses and Limitations of Principal Component Analysis

Author:
António Rebelo

Supervisor:
Dr. Lars Fritz

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Complex Systems Studies
Institute for Theoretical Physics

December 15, 2017

UTRECHT UNIVERSITY

Abstract

Faculty of Science
Institute for Theoretical Physics

Master of Science

Unsupervised Learning of Physical Models: Uses and Limitations of Principal Component Analysis

by António Rebelo

In the bulk of this Thesis, we apply an Unsupervised learning algorithm, namely Principal Component Analysis (PCA) to five models: the Ising ferromagnet, the Ising antiferromagnet, the Ising antiferromagnet on a Triangular and Kagome lattice and finally for the XY model. For this, we feed the algorithm with configurations generated through Monte Carlo algorithms. We show that the PCA is able to detect meaningful features of the models for all but the Kagome lattice model. We give a description of the mechanism through which PCA is able to find these features and conclude that PCA finds the Fourier modes of the system. Lastly, we repeat this analysis using a Neural Network in a Confusion Learning Scheme.

Acknowledgements

First and foremost I would like to acknowledge my supervisor Dr. Lars Fritz for embarking with me unto the unknown territory of machine learning. For always willing to give a helping hand when necessary, always making sure I knew the direction in which to go. For trusting in my ability to work independently and for guiding me through this Complex Systems Profile adventure in general.

Secondly, I would like to thank my parents and their huge mental and economic support. For always doing everything in their power to make sure I was doing well, both personally and academically.

Lastly, I would like to thank my sisters, my closest friends and all those that gave me an extra push when the winds of motivation grew weak.

Contents

Chapter 1

Introduction

This thesis is about the use of unsupervised learning, namely Principal Component Anlysis, in the context of physics. In the last years, machine learning techniques have been revolutionizing practically every field of science. It has been present in subjects as diverse as image recognition [6,8], finance and marketing [9], pharmaceutical engineering [25], biology [28, 36] or astrophysics [5]. With increasing computer power the range of alternatives that machine learning allows us to explore has been growing exponentially. Physics has been lagging somewhat behind other fields in this regard, but it is finally catching up with a huge flood of papers on the subject coming through in the last years. Of course a lot of it, has to do with the fact that some of the more powerful machine learning methods lie within the realm of supervised learning, where the features that are meant to be learned are known *a priori*. You could guess that this doesn't suit physics because the ultimate goal of employing machine learning in this field is to be able to identify features previously unknown to us. Even so, a lot of practical applications have been found for these algorithms (I'll expand on this in just a bit). More and more, new models and states of matter are becoming harder to crack. Having a tool as powerful as machine learning to aid us in that goal would be a tremendous achievement.

To be a little more specific about recent advances in machine learning in the field of physics, we'll now try to give an overview of recent developments. On the supervised learning side of things, regression approaches are being used to predict crystal structures [15], to approximate density functionals [38], to solve quantum impurity problems [4] and to identifying polymer states [45], just to give a few examples. Neural networks are being used to recognize phases of matter and solve many-body physics problems [10–13, 37, 39, 46]. On a tangentially related subject, there are connections being established between what we know of physical statistics and the behavior of deep neural networks [16, 29], showing that the field of machine learning also can benefit from physical insights. This is related with the fact that scaling and renormalization principles are a crucial tool in making sense of macroscopic behaviors that are a consequence of intricate and complex microscopical properties. It should come as no surprise that Neural Networks make use of the same tools to find structure in big datasets.

On the other hand, unsupervised learning (and this is where we get closer to the subject of this thesis) has also been making its way into physics. It's been used to make video predictions of object trajectories [18]. Using a confusion learning scheme, Neural Networks have been used to recognize phase transitions [40]. In the appendix, we show some results that we obtained by analyzing some models with this method. Finally, in the paper that sparked our interest in this topic, phase transitions were recognized using the same method we study here, Principal Component Analysis [42]. Since then there have been a couple of papers that made the same use of PCA to study phase

transitions in physical models [21, 41]. These were of great importance in providing valuable insights for the analysis that is done here.

As stated previously, the goal of using this method is to be able to, in the future, be able to flesh out important properties of a certain model just by feeding the algorithm with the raw configurations of the system. However, since using this method in the context of physics is something relatively new, there's much to be gained by studying how it fares in cases of models for which we already know the details and physical properties of. This is what we propose ourselves to do in this Thesis: to see what and in which way PCA is able to learn properties of understood physical models so that in the future, interpretation of its applications elsewhere come along easier. Another goal of our research is to identify the limitations of PCA so we can know what to expect when analyzing a new model.

To fulfill these goals, we studied five models:

- The Ising Ferromagnet
- The Ising Antiferromagnet
- The Ising Antiferromagnet on a Triangular Lattice
- The Ising Antiferromagnet on a Kagome Lattice
- The XY model

Each of these models was chosen with a goal in mind. We sought to increasingly raise their difficulty to test the PCA's pattern recognition abilities.

Structure

In this section we outline the way this thesis is structured and give a quick summary of each chapter.

In Ch.(2) we go into details about each particular model outlining their main physical properties so that context for understanding the results is not lacking.

In Ch.(3) we talk about the adopted methodology. The Monte Carlo methods through which the system configurations fed into the machine learning algorithms were generated are explained. An overview of PCA is given, explaining the mathematical process behind it with a quick example to provide intuition on how it works. The same is done for the Confusion Learning Scheme.

Ch.(4) makes up the bulk of this thesis. It's in this section that we show the results of our analysis and try to provide their interpretation

Finally in Ch.(5) we wrap up the discussion outlining the main results of this Thesis and point out what are the future prospects in this line of research.

Chapter 2

The Models

In this work, we study a variety of models. To better understand the results here we introduce some of it's basic properties. We will not delve to deep in these models, as that could be a subject for another thesis in itself. We only intend to give an overview of them so that readers that are less acquainted with them, have some context when going through the results of their analysis.

2.1 Ising model

The first model to be tested was the two-dimensional classical square-lattice Ising model, designed to study the paramagnet-ferromagnet transition. It describes an arrangement of magnetic dipole moments of atomic spins that at any given moment have values that are either +1 or -1. Each spin interacts with its nearest neighbours and with an external magnetic field pointing in the direction perpendicular to the lattice. This model is of huge historical relevance for several reasons:

- it first helped to consolidate the position of the field of statistical physics among the scientific community when Ising first solved the one-dimensional case his 1925 thesis [22].
- Lars Onsager latter solved it for the 2-dimensional case in the celebrated paper dated from 1944 [35], developing tools and insights instrumental for the development of modern statistical physics.

It's probably the most studied model in all of physics, which can be attributed to the fact that despite it being very simple it displays rich physics, making it ideal to develop intuitions about complicated constructions. All of these factors make it the ideal starting point for our investigative analysis of machine learning phase transitions.

Moving on to the details of the model, the total energy of the system can be obtained by the familiar Hamiltonian:

$$H = -J \sum_{<ij>} S_i S_j - h \sum_i S_i$$

$< ij >$ denoting nearest neighbors, and where the spin $S_i = \pm 1$ and h is the external magnetic field, which in this study will always be zero. The ferromagnetic model (when $J > 0$) has a second order phase transition at $T_c/J = 2/ln(1+\sqrt{2}) \approx 2.27$ [35] above which there is a disordered phase; below, the system presents an ordered phase that favors configurations with aligned (spins with the same value). Typical ferromagnetic

Ising configurations are presented in Fig. 2.1. The same is true for the antiferromagnetic model ($J < 0$) except that the ordered phase consists of anti-aligned spins. Note that the given value for the critical temperature is only true in the thermodynamic limit where the lattice size is infinite.

Important features of this model are encapsulated in its **critical exponents**. Critical exponents establish universality classes for phase transitions. Transitions within the same class have certain thermodynamic quantities that scale with the same power law near the critical point, the exponent of the power law being the critical exponent:

$$\xi \sim |T - T_c|^{-\nu} \tag{2.1}$$
$$M \sim |T - T_c|^{\beta} \tag{2.2}$$
$$C \sim |T - T_c|^{-\alpha} \tag{2.3}$$
$$\chi \sim |T - T_c|^{-\gamma} \tag{2.4}$$

The critical exponents are ν β α and γ. ξ is the correlation length, M the magnetization, C the heat capacity and χ the magnetic susceptibility. We do not go through the calculation of these exponents here but suffice it to say there are several ways of going about it. Historically, the first being the one presented in Onsager's solution and there's also a more modern way where you map this problem to a conformal field theory (in $d = 2$ the CFT in question is the minimal model). The values for the critical exponents are listed bellow

$$\nu = 1$$
$$\beta = 1/8$$
$$\alpha = 0$$
$$\gamma = 7/4$$

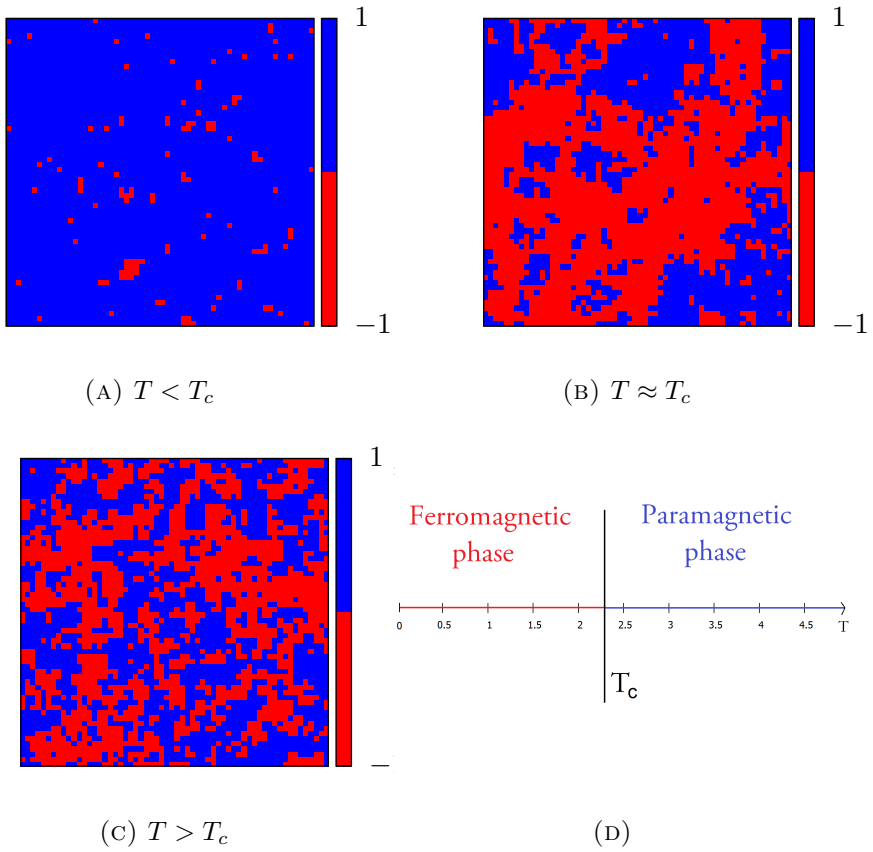
These values were taken from [26].

Finite size scalling

When the lattice size is finite, there is no "true" phase transition because thermodynamic potentials become fully analytic, presenting no divergences. Of course the behavior of the system still resembles that of a phase transition. It's possible to define an analogue of the critical temperature,$T_c'(L)$ for finite size as the value of the temperature for which the specific heat is maximum. The full finite size scaling behavior is laid out in [17]. In the particular case of the correlation length, in the thermodynamic limit it becomes infinite at T_c. However, for a finite system when $\xi \sim L$ the system is already ordered, so we can establish the relationship:

$$|T_c - T_c'|^{-\nu} \sim L \tag{2.5}$$

If in the limit of $L \to \infty$ the magnetic susceptibility becomes infinite at the critical temperature it's reasonable to argue that when L is finite, χ has a maximum at T_c'. Meaning that if we measure χ for several values of L and get several values for T_c' we



(A) $T < T_c$ (B) $T \approx T_c$

(C) $T > T_c$ (D)

FIGURE 2.1: (a-c)Usual configurations for the ferromagnetic Ising model bellow,at and above the critical point (d) Phase diagram for the Ising model.

can estimate T_c and ν.

The order parameter [1] for the ferromagnetic Ising is the total magnetization per site, however, since for finite size this quantity becomes,on average, zero for all temperatures, a more useful parameter can be defined as it's modulus:

$$m_f = \left\langle \left| \frac{1}{N} \sum_{i=1}^{N} S_i \right| \right\rangle$$

for the antiferromagnetic case, the order parameter is the staggered magnetization per site. It's equivalent reads:

$$m_a = \left\langle \left| \frac{1}{N} \sum_{i=1}^{N} (-1)^i S_i \right| \right\rangle$$

Later on we will see how Principal Component Analysis, the algorithm used for the bulk of our analysis, is able to recover and recognize these quantities.

[1] In case the reader is not familiar with the concept of order parameter, it is a quantity associated with the phase transition that usually encases the phase behaviour of the system by taking non-zero values in the ordered phase and zero in the disordered.

2.2 Ising Antiferromagnet on a Triangular Lattice

The Antiferromagnet (AF) Ising model on a triangular lattice is a classical example of a frustrated system. The Hamiltonian takes the same form as in the Ising model:

$$H = -J \sum_{<ij>} S_i S_j - h \sum_i S_i$$

the sum over $< ij >$ runs through all 6 nearest neighbors. Again h will always be zero in the study presented here.

The most relevant feature of this model for our analysis is its frustration. Frustration is a phenomenon commonplace among AF systems and it happens when two conditions are fulfilled:

- it's impossible to satisfy all of the conditions that correspond to a lower energy configuration.
- the ground state presents a considerable degree of degeneracy.

To see how this happens for this model in particular, it's helpful to look at Fig. 2.2. The basic cell of this model are triangles just like the one depicted. The two spins that are drawn are anti-aligned as is energetically favorable, but the spin represented by a question mark will always be aligned with one of the other two. The three can't be all anti-aligned. Of all the possible 8 states, 2 correspond to high-energy configurations of spins that are entirely aligned and the other 6 are degenerate ground-states of the system. If instead of one triangle, we consider a full lattice, and if for every triangle there are 6 degenerate ground-state configurations, you can begin to imagine the amount of possible ground-states of the lattice.

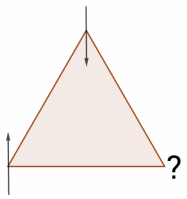


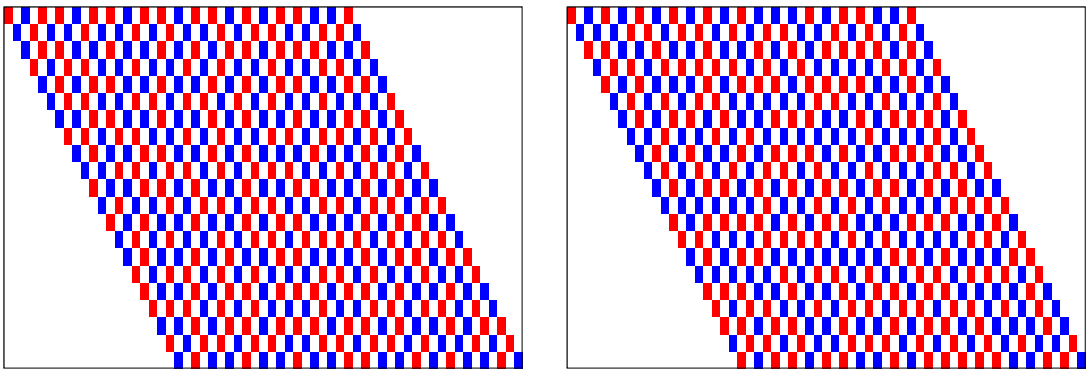
FIGURE 2.2: Antiferromagnetic spins in a triangular lattice.

A usual measure of the frustration of a given model is the residual entropy per site, S_{res} (the ground state entropy). Since Ising spins have two possible orientations (± 1) that occur with equal probability in the disordered phase, in the infinite temperature limit, S_{res} is just $ln(2)$. Because of this, the ratio $S_{res}/ln(2)$ is a measure of the ground state degeneracy, and hence of the model frustration. It ranges from 0 for a unique or sub-extensively degenerate ground-state to 1 for a system without any ground state constraints, so the higher it is the more frustrated, in principle, a model is. For the Triangular antiferromagnet, this quantity takes the value [27]

$$\frac{S_{res}}{ln(2)} \approx 0.47$$

There are other ways in which frustration can arise. Because in this case the frustration comes from the layout of the system it's called geometric frustration.

This model was first studied by Wannier [43] and by Newell [31]. It remains in a paramagnetic phase for every **finite** temperature meaning it doesn't have any sort of long range order. This is why this model was chosen to be analyzed since it will be harder for pattern recognition algorithms to lock in into something.



(A) $T < T_c$ (B) $T > T_c$

FIGURE 2.3: Usual configurations for the triangular antiferromagnet for (a)low and (b)high temperature.

This model is also studied because of its non-trivial zero temperature properties. The system becomes critical at this temperature with spin-spin correlations decaying with distance as power laws. In contrast, they decay exponentially for finite temperature.

In Fig. 2.3 we can see some configurations for low and high temperature. The differences are hard to spot, but upon closer inspection, it can be seen that there are no triangles with three aligned spins the for low temperature, while for the high temperature case it's easy enough to spot triangles like these. This is noted here only because it will be of some relevance in the analysis.

One last thing that might interest the reader is that this model can be mapped onto a compact QED photon theory [30].

2.3 Ising Antiferromagnet on a Kagome Lattice

The Kagome lattice consists of a modified version of the triangular Ising with a different arrangement where every site has 4 nearest neighbors instead of 6, but where the building blocks are still made up of triangles (see Fig. 2.4).

It shares some of its properties with the triangular model. It remains a paramagnet for every finite temperature and it presents a more highly degenerate ground state manifold, with residual entropy (see previous page) [23, 27]

$$\frac{S_{res}}{ln(2)} \approx 0.72$$

This makes it an even more frustrated model than the Triangular antiferromagnet. The reason behind it is the low connectivity between each site. Since there are only 4 neighbors for a given site, strong (or weak) correlations are much harder to establish and die off extremely quickly.

Similarly to the Triangular antiferromagnet, this model also becomes critical at zero temperature.

All of this should make it an harder challenge to our pattern recognition algorithms to overcome and it's the reason this model in particular was chosen to be analyzed.

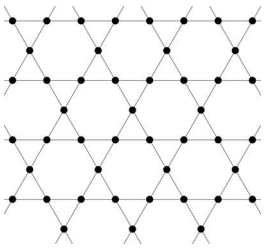
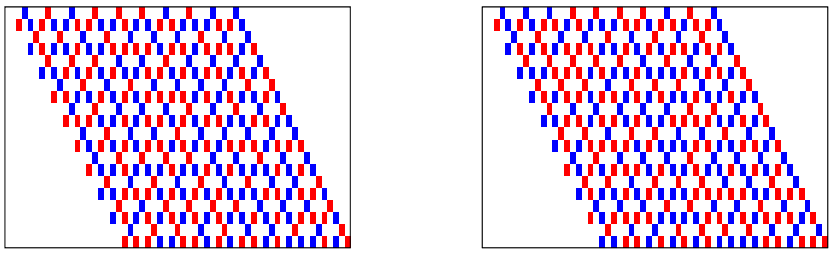


FIGURE 2.4: Depiction of a Kagome lattice.



(A) (B)

FIGURE 2.5: Typical Kagome configurations for (a) low and (b) high temperature.

2.4 XY model

The classical 2D XY model is a description of two component vector spins that lie in a square lattice. Contrary to what happens in the Ising model, the spin is no longer a binary variable. Instead the possible states lie on a continuous range of values. The spins can orient themselves in any direction confined to the plane and the state of a given spin is given by the vector $\mathbf{S}_i$. It's characterized by its norm and angle $\theta \in [0, 2\pi]$ that the spin does with a reference axis that can be arbitrarily defined. The energy of a given state is given by

$$H = -J \sum_{<ij>} \mathbf{S}_i \cdot \mathbf{S}_j = -JS^2 \sum_{<ij>} \cos(\theta_i - \theta_j)$$

where <> denotes nearest neighbors. We will always use $S^2 = 1$ in this thesis.

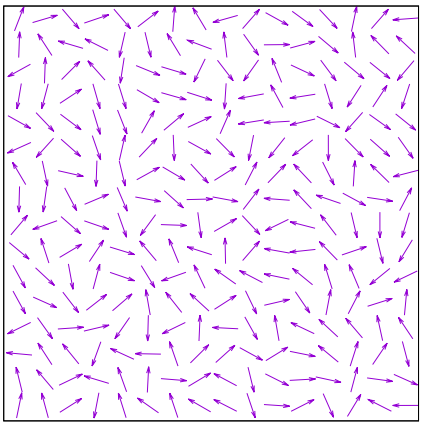


FIGURE 2.6: Configuration of the XY model for $T/J = 0.4$.

This model is famous for being the first known case of an infinite order phase transition, the Kosterlitz–Thouless (KT) transition. There are several systems that have transitions that belong to the same universality class as the XY, most notably the superfluid transition in two-dimensional helium [3, 24]. The transition goes from a low temperature phase where spins are aligned, presenting quasi-long range order with the correlation function decaying with a power law:

$$\langle \mathbf{S}_i \cdot \mathbf{S}_j \rangle \propto r^{-\eta}$$

and a disordered high temperature phase where correlations decay exponentially

$$\langle \mathbf{S}_i \cdot \mathbf{S}_j \rangle \propto e^{-r/\xi}$$

where r is distance, $\eta = T/2\pi J$ and and ξ the correlation length.

This transition can be explained in terms of vortices. A vortex is a topological defect and it can be quantified through its vorticity v

$$v = \oint_C \nabla\theta \cdot d\mathbf{l} = 2\pi k$$

where k is the winding number and where $\nabla\theta$ can be approximated by angle differences between near sites. C is a countour around the vortex center. A vortex corresponds to

a structure with $k = 1$ and an antivortex to $k = -1$. To visualize what these structures look like see Fig. 2.7.

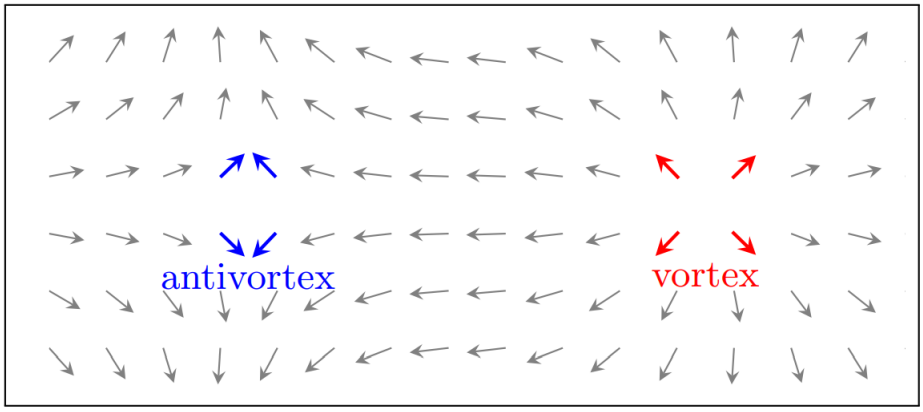


FIGURE 2.7: Depiction of vortex and antivortex. Image taken from reference [7].

The vortex has a high configurational energy and naively one should not expect to observe them when temperature is low and there's not enough energy availiable to create these structures. However, it turns out that vortex-antivortex pairs have a much lower energy, and occur spontaneously for low temperatures due to thermal fluctuations. They behave like particles in a dilute gas, propagating and occasionally bumping against each other. As temperature increases, more pairs spontaneously form until, suddenly, at the transition temperature T_{KT}, vortices and antivortices decouple. Uncoupled pairs, make the system more disordered and from there stems the exponentially decaying correlation function.

All derivatives of the free energy are finite at the transition and the transition temperature is $T_{KT}/J = 0.893$ [14, 34]. The magnetization in this model is a vector, given by

$$\mathbf{M} = \frac{1}{N}\sum_i (\cos(\theta_i), \sin(\theta_i))$$

It's also worth pointing out that the magnetization evaluated at the critical point scales as $L^{-1/8}$ [14]. This will come into play as Principal Component analysis, will be able to detect this feature.

For this model, defining an order parameter is trickier but doable, it is the *superfluid density*. However to go into detail as to how this parameter behaves, one would have to go about introducing to many complex variables and so we refer to references [33, 44] for a full treatment.

Chapter 3

Methods

3.1 Monte Carlo

Monte Carlo (MC) techniques are a standard and fundamental part of every computational physicist's toolbox. In this thesis, all of the samples that were analyzed were generated using MC algorithms, so here we give an overview of how they work and go into some of the specifics the simulations. For further details regarding Monte Carlo techniques see reference [32].

The problem that MC solves, presents itself when trying to deal with many body systems. While each individual body usually obeys a simple set of equations of motion that are easy enough to express mathematically, the sheer number of equations that need to be solved in order to get the full dynamics of the system makes it an almost impossible endeavor. However, the fact that there are some macroscopical properties that behave in a very predictable fashion, such as the pressure of a gas composed of many particles, tells us that there's a statistical character to the nature of the solutions. Collectively, the range of probable behaviors of the system is small so it is likely that the system will display a behavior within this range.

More specifically, it is known from statistical mechanics that at **equilibrium**, the probability p_μ that a system is in a given state μ with energy given by E_μ is given by the Boltzmann distribution:

$$p_\mu = \frac{e^{-\beta E_\mu}}{Z}$$

where $\beta = 1/k_B T$ and Z is the partition function,

$$Z = \sum_\mu e^{-\beta E_\mu}$$

The Monte Carlo goal is to make an algorithm that samples states of the system in which the probability of any particular one appearing is proportional to its Boltzmann weight. To achieve this goal, a Markov process is used, which is to say that given a state μ, a new state ν is generated randomly. Then from ν another state is generated and so on and so on producing a Markov chain of states. There's a **transition probability** $P(\mu \to \nu)$ that given a state μ the system will transition into ν. A Markov process is defined by imposing two constraints on these probabilities: that they are constant over time and that they have no memory i.e. , they are not dependent on the properties of the current states μ and ν.

The probabilities are chosen so that starting from any possible state, if this process is repeated enough times, a succession of states that appear with probabilities given by the Boltzmann distribution will be produced. This is the computer analogue of reaching equilibrium and in a way it's no so different from what happens in reality. In order to achieve this, we must impose further constraints on the Markov process, namely the conditions of **ergodicity** and **detailed balance**

The condition of ergodicity says it should be possible for the system to reach any state from any other state, if run for long enough. This is imposed because every state has a non-zero probability in the Boltzmann distribution so it should always be reachable.

The condition of detailed balance is a bit trickier to explain. This condition ensures that after reaching equilibrium, it is the Boltzmann distribution that is reached. Saying that a system as reached equilibrium, is equivalent of saying that the rate at which system makes transitions into and out of any state μ must be equal. In mathematical terms this can be expressed as

$$\sum_{\nu} p_\mu P(\mu \to \nu) = \sum_{\nu} p_\nu P(\nu \to \mu)$$

since $\sum_\nu P(\mu \to \nu) = 1$ the previous expression reduces to

$$p_\mu = \sum_{\nu} p_\nu P(\nu \to \mu)$$

However, this condition is not enough to actually guarantee detailed balance because the system may enter in what's called a limit cycle, which is to say that it can enter in a trajectory that's a closed cycle of states. If this happens, then the sampled states will not obey the Boltzmann distribution.

To prevent this from happening, the **condition of detailed balance** is then imposed:

$$p_\mu P(\mu \to \nu) = p_\nu P(\nu \to \mu)$$

or as it is most commonly written

$$\frac{P(\mu \to \nu)}{P(\nu \to \mu)} = e^{-\beta(E_\nu - E_\mu)}$$

The Metropolis algorithm

The above mentioned condition is all that is necessary in order for it work, but notice there is still a lot of freedom in how to choose the transition probabilities. To construct a good algorithm one must be smart in the way he chooses the shape of P. It is often the case that each system has a optimal way of doing it and that specifically tailored algorithms will be must faster than the standard ones. But the standard ones are a good starting point to analyze any system and even those have some nuances to them. The most standard and widely used is the Metropolis algorithm. In this thesis, for all of the models except for the Ising ferromagnet this was the algorithm of choice.

Before getting into the gist of it, it's useful to separate $P(\mu \to \nu)$ in two parts:

$$P(\mu \to \nu) = g(\mu \to \nu) A(\mu \to \nu)$$

$g(\mu \to \nu)$ is called the selection probability and it is the likelihood that given a state μ the state ν will be generated. $A(\mu \to \nu)$ is called the acceptance ratio. Given that the target state ν was generated, it will transition into it a fraction $A(\mu \to \nu)$ of the times (when it doesn't change, it just stays in μ).

$g(\mu \to \nu)$ is related with how the new state ν is generated and it varies from model to model. In general the goal is to maximize the acceptance ratios so that the system goes from state to state fast, giving us a good sample of different states as opposed to being stuck in the same one for a long number of iterations. To that goal it is useful to choose a state ν that is not far from μ in terms of energy. In the Ising model for example, ν is obtained through μ by flipping one random spin. If there are N spins then there are N possible ν states that can be reached from ν and so

$$g(\mu \to \nu) = g(\nu \to \mu) = \frac{1}{N}$$

the equation of detailed balance then simplifies to

$$\frac{A(\mu \to \nu)}{A(\nu \to \mu)} = e^{-\beta(E_\nu - E_\mu)}$$

The acceptance ratios can have any value between zero and one. As it was pointed out just now, the way to get the most efficient algorithm is to maximize them. The way to achieve this, is to set the larger of the two to 1, and let the other one take whatever value is necessary to satisfy the detailed balance condition. $A(\mu \to \nu)$ is only smaller than $A(\nu \to \mu)$ if $E_\mu < E_\nu$. That leaves us with

$$A(\mu \to \nu) = \begin{cases} e^{-\beta(E_\nu - E_\mu)} & \text{if} E_\mu < E_\nu \\ \\ 1 & \text{if} E_\mu > E_\nu \end{cases}$$

This sums up the Metropolis algorithm. A new state is randomly generated, the difference in energy from the current one is then evaluated. If it's lower, the change to that state is accepted, if not, the change is made with probability $e^{-\beta(E_\nu - E_\mu)}$. Rinse and repeat and you'll have an algorithm that travels nimbly though state space (most of the times), sampling the desired Boltzmann distribution.

Simulation Details

In this section some of the details of the simulations that were performed are outlined.

First, a note on the algorithm that was used for the Ising ferromagnet. The Metropolis algorithm suffers from critical slowing down in this model. For this reason, and because a better alternative was readily available, a Wolff algorithm was used. The Wolff algorithm is a modification of the Metropolis and the general idea behind it is that instead of attempting a single spin flip, new states are generated by flipping a cluster of spins. The precise details of the algorithm can be found in reference [32].

The algorithms used to analyze the data require sampling a large number of configurations per temperature value. The samples should be completely independent from one another, so to guarantee this, after each sample is taken, the system was reset to

a random configuration and then rethermalized to take the next sample. Also worth noting, is that the computational cost of a Monte Carlo step varies with system size and from model to model. Frustrated models[1], for example, take longer to achieve equilibrium and so are more costly, but that is not the only factor. The XY model, for instance, takes longer than the Ising because its states lie in a continuous range of values. Because of this, even the generation of new states is fundamentally different. While for models with a discrete phase space it's easy enough to define the new state ν (say, single spin flip in Ising), for the XY model there's a much wider range of reasonable choices. To implement Metropolis in this case the new state was constructed by changing the angle of one of the spins by $\Delta\theta$. To assure a good evolution of the system this value was dynamically adapted so that the acceptance rate of new states stayed in the interval $[0.3; 0.7]$ which is typically the desired interval.

All of these factors were considered in choosing the number of samples taken per temperature, the used lattice sizes and the amount of Monte Carlo steps per site. We sought to have a balance between having large enough datasets to be able to do meaningful analysis, nicely thermalized data while keeping the program runtimes within reasonable timeframes.

TABLE 3.1: Details of the numerical simulations used to generate the datasets. L refers to lattice size, N_0 is the number of samples per temperature, MCS is the number of MC steps per site, T_{min}/T_{max} is the range in which the samples were taken and ΔT the interval between temperature values.

Model	L	N_0	MCS $\times 10^{-3}$	T_{min}/T_{max}	ΔT
Ising F	10,15,20,40,60	5000	21	1.9/2.9	0.05
Ising AF	20,40,60	5000	30	1.9/2.9	0.05
Triangular Ising AF	15,21,27	2000	80	0.1/2.2	0.05
Kagome Ising AF	12,16,20	2000	175	0.3/2.2	0.05
XY F	8,16,20,25	1500	22	0.4/2.3	0.05

Also, Periodic Boundary conditions were employed in all of the models to minimize finite size effects.

Lastly a note on how the triangular lattice was constructed. For this we simply started out from square lattices and coupled each site to its diagonal neighbors besides the usual four. To construct the Kagome, we set the adequate sites of the Triangular lattice to zero, killing all of their contributions to the energy and interactions.

[1] Frustrated models are models that have a high degeneracy of the ground-state and where it's impossible to meet all the conditions that correspond to a lower energy. The Triangular and Kagome lattice Ising antiferromagnets are examples of frustrated models.

3.2 Learning algorithms

In this section we go about describing how the algorithms that were used to analyze the data work, but first it's worth it to make a quick note about how these algorithms are usually classified.

Artificial Intelligence algorithms usually fall within one of two categories: supervised or unsupervised machine learning. There is a big distinction between these two.

Machine learning problems always rely on a training dataset. The datasets that are fed into **supervised** and **unsupervised** learning algorithms are different. For the supervised learning case, each data sample is labeled with the features that the algorithm is intended to learn. As an example, suppose we want to train a neural network to distinguish pictures with blue cars, from pictures with red cars. We build a dataset composed of many images of cars and label each sample with either blue or red according to what is shown in the image. The neural network will use this information to learn how to predict the color of an unlabeled sample.

On the other hand, unsupervised learning doesn't rely on any labels to learn features of the data. If we intended to distinguish between blue and red cars with an unsupervised learning algorithm, the only thing fed to it would be the pictures of the cars. The algorithm would then learn by itself that the differences between the images have to do with the color of the cars and would also be able to correctly predict it for an image that was not part of the training dataset.

Usually, different problems are better suited to either supervised or unsupervised learning, however, if we intend to use machine learning as a tool to analyze physical models and learn new features, unsupervised learning will be the algorithm of choice.

3.2.1 Principal Component Analysis

Principal Component Analysis (PCA) is one of the simplest methods that can be used to do unsupervised learning. It is in sum, a linear transformation, yet its power will become obvious throughout this chapter. We'll first describe what PCA consists of while trying to give some intuition about it. For further details regarding PCA we refer to reference [1].

We start out by building the input matrix X with the data that will be analyzed. PCA has numerous applications that span a wide variety of fields. Because of this, the structure of X varies quite a bit depending on the context in which it's used. In our application of physical feature extraction, the input will almost always be the raw configurations of the system. If the case study is the Ising model for example, then the matrix X will have dimensions NxM, where $N = L^2$ is the number of lattice sites and M is the total number of independent configurations that are fed onto the PCA. For each temperature, N_0 uncorrelated configurations are sampled through Monte Carlo algorithms and added to the input matrix. It is also useful to calculate the column average $m_j = 1/M \sum_i X_{ij}$ and subtract it to the column entries to create a centered and column-wise zero mean matrix (this is just a standard data cleaning procedure when dealing with PCA).

PCA is a dimensionality reduction procedure that finds a specific basis to represent the data that does so more efficiently than the original basis (note that one data point, in the example of the Ising model, corresponds to one configuration, or in other words, a point in a L^2 dimensional space). The word efficiency here is used in the context of

variance. The goal is to find an orthogonal basis in which the first axis will point along the direction that has highest variance, the second will point in the direction that has highest variance with the restriction that it's orthogonal to the first one and so on and so on. If we then project the data in the new basis it's often the case that for high dimensional data, most of the total variance is encased in the first few directions: the principal components. This also means, that since most of the informational content of the data will be in the first few directions, it will frequently happen that for many of the directions the data is just randomly distributed, close to the origin, making them irrelevant. If the directions along which the variance is minimal are discarded, this process is useful in diminishing the amount of dimensions that are needed to represent the data without significant loss of information, but besides that, it's a process that's useful in fleshing out features of the data. Variance is usually a consequence of the change in a certain feature of the data, so it's possible to take some insight on the structure of the data by seeing in which direction the principal components point. This will hopefully become more clear with the first examples, for now let's move on to the precise definition of the procedure.

As already noted, the principal components will be mutually orthogonal vectors that point in the direction along which the variance in the data is the greatest. The i$^{\text{th}}$ component y_i is obtained through:

$$\mathbf{y}_i = \mathbf{X}\mathbf{w}_{(i)}$$

$\mathbf{w}_{(i)}$ is the so called weight of the first component and is the vector that needs to be computed.

We can iteratively find the first weight vector, $\mathbf{w}_{(1)}$, by computing

$$\mathbf{w}_{(1)} = \arg\max_{\|w\|=1} \left\{ \sum_i (\mathbf{x}_i \cdot \mathbf{w})^2 \right\}$$

where arg max stands for arguments of the maxima, meaning the domain points at which the function inside the brackets is a maximum. The subsequent components can be found by subtracting the already calculated components and go through the iteration again. While this process is somewhat more intuitive, it can be shown (see [2]) that there's an equivalent and simpler way to do it. The principal component weights, will be the eigenvectors of the matrix $X^{\mathbf{T}}X$:

$$X^{\mathbf{T}}X\mathbf{w}_{(i)} = \lambda_i \mathbf{w}_{(i)} \tag{3.1}$$

the eigenvalues, λ_i also hold a meaning. They are a measure of the variance reproduced by its corresponding component and so it is convenient to order them decreasingly: $\lambda_1 \geqslant \lambda_2 \geqslant ... \geqslant \lambda_N$. Usually one refers to the normalized eigenvalue, or as it's commonly called in the literature, the **explained variance ratio**

$$\widetilde{\lambda}_i = \lambda_i / \sum \lambda_i$$

To illustrate and visualize how the process works in practice, a simple example of PCA analysis can be seen in Fig. 3.1. Keep in mind that in our analysis the PCA will be used in a high-dimensional context, so depending on the value of the first few

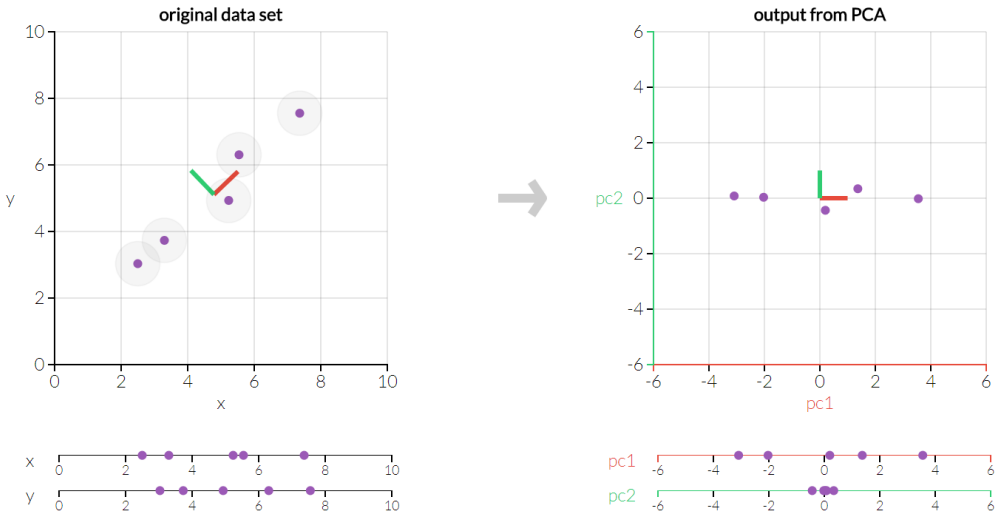


FIGURE 3.1: Example of PCA analysis of a 2-dimensional toy dataset. Note the projections of the data in the principal components. Most of the variance happens along the $pc1$ direction. Image by Victor Powell, n.d., retrieved in 13/11/2017 from http://setosa.io/ev/principal-component-analysis/

explained variance ratios, the projection on the plane/3D-space/... made by the first two/three/... principal components might be more useful than the one-dimensional projection.

3.2.2 Confusion Learning

Principal component analysis provides us with a great tool, however it also presents some limitations, as we will outline in our analysis. Neural Networks (NN from here on) are in general more powerful in learning features of a dataset, but they too have some drawbacks: they are hard to interpret and usually work only within the supervised learning scope.

It's possible however, to use the power of NNs to our advantage, mixing unsupervised with supervised learning and cleverly circumvent the latter problem. This is what happens in a confusion learning scheme, but note that the goal is now solely to detect if there's a phase transition and to get an estimate of the critical temperature, as physical insights are still hard to attain.

Moving on to the explanation of how this method actually works, suppose that we are working within the frame of the presented physical models. Now the data alone won't suffice and it's necessary to label each configuration according to which phase they are for the NN to be able to correctly classify a given unlabeled configuration. Usually, the way to measure the effectiveness of the NN is to perform cross-validation of the data. In other words, the training data is split into a test set and the actual training set (note that both have the data labeled). The NN is trained with the training set and it's asked to predict the classification of the configurations in the testing set. The percentage of configurations that the NN predicts correctly is called the **score**, S of the NN. Note that 50% is the worst score a NN can have since it means it just predicts randomly.

Now, when correctly labeled data is fed int the NN it's common to have scores larger than 90%. The NN will, in many cases, correctly learn the data features and is able to correctly predict which phase the system is in. However if we knew how to correctly label the data, all of this would be besides the point. As stated before, the goal is to not have to recur to any *a priori* knowledge of the system. So how can this be achieved?

Assume that the varying parameter is the temperature and that we have a training data set in the range (T_l, T_h). The only thing that is known to us is which temperature each configuration belongs to and that at a certain temperature T_c the system undergoes a phase transition ($T_l < T_c < T_h$). Since T_c is not known, the data is labeled according to an arbitrary fake critical temperature that we call as T_c', so everything bellow T_c' is 0 and everything above is 1 (accordingly let's say that the data bellow the real critical point has feature 0 and that the high temperature data has feature 1). The NN is trained with this labeling and its score $S(T_c')$ is measured. We repeat this process for every temperature.

It turns out that, $S(T_c')$ will always have a universal W-shape (see Fig. A.2) and the critical point will be located at the middle of the W, giving us a way to estimate T_c. The explanation goes like this: when $T_c' = T_l$, all of the data is labeled 1, so when asked to predict something, the NN will always predict 1 because it was the only thing it learned, hence it will have a perfect score. Similarly the case repeats itself for $T_c' = T_h$. In the intermediate points, the rule the NN follows is that if most of the data that has feature x is labeled as y then it will classify all data with feature x as y. This means that, for example, if 70% of the data with feature 0 is labeled 1, the NN will wrongly classify the 30% that has label 0, decreasing the prediction score. So the score has the expression:

$$S(T_c') = 100 \times \begin{cases} 1 - \dfrac{min(T_c - T_c', T_c' - T_l)}{T_c - T_l} & T_l < T_c' < T_c \\ 1 - \dfrac{min(T_c' - T_c, T_h - T_c')}{T_h - T_c} & T_c < T_c' < T_h \end{cases}$$

If $T_l < T_c < T_h$, as is the case, the function will always have said W-shape.

Note that the data doesn't need to be sampled symmetrically around T_c. If it's not, then the dips in the W will be asymmetric but the peak will still be at T_c.

We are now equipped with the tools to proceed for the data analysis.

Chapter 4

Results and Discussion

In this chapter, the obtained results are laid out. Here, we give our interpretation of the results, and how they are related to the characteristics of the tested models.

4.1 Principal Component Analysis

The goal in this section is to evaluate precisely what the PCA can and cannot do. To this end, we start out with the simplest case, the Ising ferromagnet and then we test out how it fares for the antiferromagnet case of the same model. From there we go on to frustrated models, testing the Triangular and Kagome Ising antiferromagnets. Finally we test it out for the XY model.

4.1.1 Ising model

We'll first outline the main results that were obtained and subsequently give the interpretation and analysis of these results.

The PCA was fed a matrix X in which each of the rows is a configuration sample for a given temperature (the spin values are just aligned in a row vector and put together to create X). 5000 samples per temperature value were generated in this case. The PCA was carried out by solving the eigenvalue problem

$$X^{\mathbf{T}} X \mathbf{w}_{(i)} = \lambda_i \mathbf{w}_{(i)}$$

as discussed in the Methods chapter.

Results

It can be seen in Fig. 4.1 that the value of the first explained variance ratio is dominant over the others. If we project the data onto the first two components we get an idea of what the PCA found, as it is seen in Fig. 4.2a.

Immediately we see three distinct clusters, two with low temperature configurations and one with high temperature ones. More on that later. The weight corresponding to the first component is constant throughout the lattice and there's obviously structure in the weight of the second component. Oddly enough the projection of the data in the second and third components (since they have similar explained ratios, it makes sense to group them together) groups the high and low temperature together with intermediate temperatures spreading out "outward".

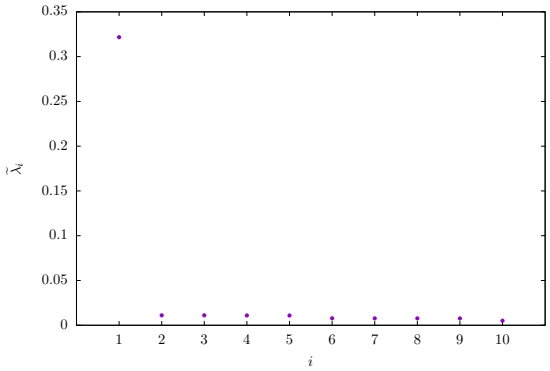


FIGURE 4.1: First 10 explained variance ratios for $L = 40$. They denote their relative importance in reproducing the data.

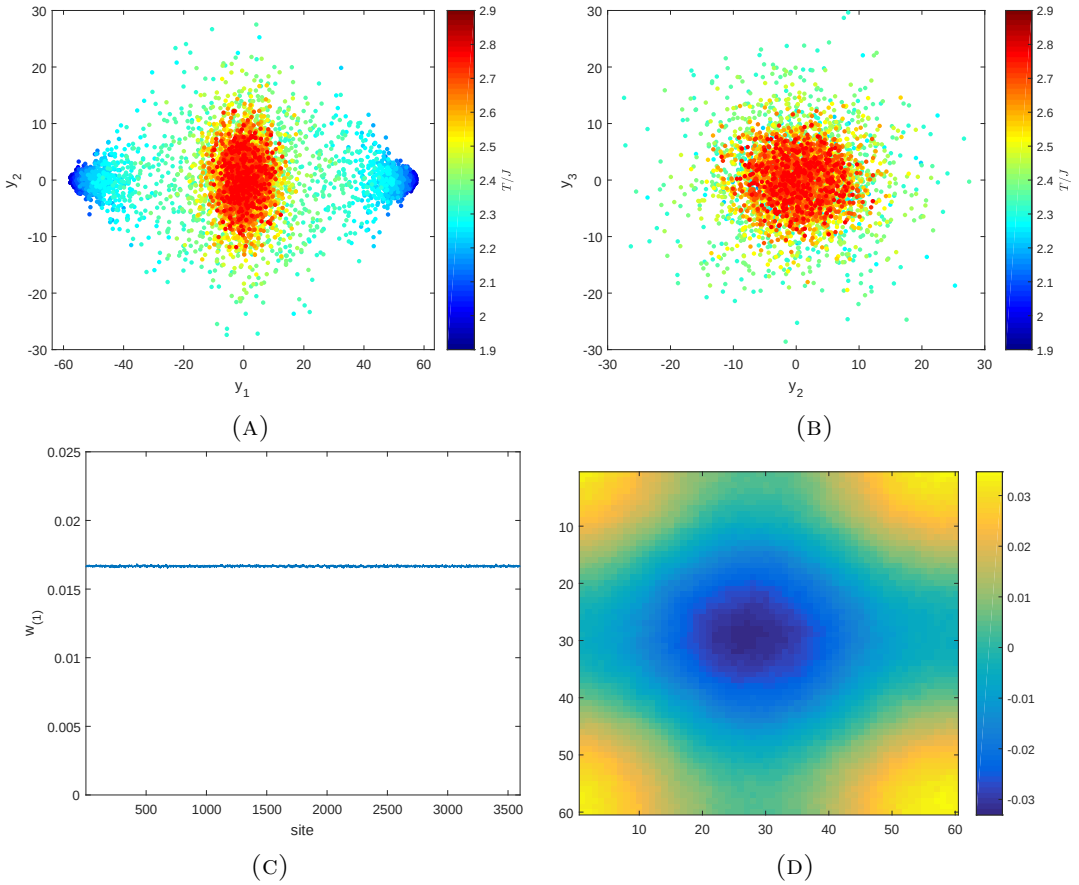


FIGURE 4.2: (a)Projection of the ferromagnetic Ising configurations on the two main principal components for $L = 60$. Only 300 configurations per temperature are represented to not obscure the image. (b)Projection of the ferromagnetic Ising configurations on the second and third principal components. (c)Weights of the first component for every lattice site for $L = 60$. Notice that $1/L \approx 0.017$ (d)Weights for the second component plotted on the lattice.

Discussion

We now take a moment to interpret and make sense of the results described in the previous section.

Looking at the projection of the data in the two main principal components we immediately see three distinct clusters: one for high temperature and two for low temperature. This hints that the clusters for low temperature correspond to the two possible alignments of the groundstate of the Ising model for $T < T_c$ (all up, or all down). In fact the weights of the first principal component are approximately constant in every lattice site (see Fig. 4.2c). Since the weight vector is normalized, this means that the first principal component is proportional to the total magnetization of the system. To show this, let c be the constant of the first weight, y be the coordinate of the first principal component for a given configuration $\mathbf{x}$ taken from one of the rows of $\mathbf{X}$. We have

$$y = \mathbf{x}\mathbf{w}_{(1)} = c \begin{bmatrix} S_1 & S_2 & S_3 & ... \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ ... \end{bmatrix} = c \sum_{i=1}^{N} S_i$$

where S_i is the value of the spin in site i. The normalization condition of $\mathbf{w}_{(1)}$, implies that $c^2 L^2 = 1$, so we get:

$$y = \frac{1}{L} \sum_{i=1}^{N} S_i$$

meaning that if we plot $|y| / L$ in function of the temperature, we should get something very similar to m_f. Indeed this is what can be seen in Fig 4.4a, meaning the PCA is able to somehow flesh out the main characteristics of the data, namely its order parameter in this case. It might not seem very surprising since in this case the order parameter is very simple function, but that isn't all the PCA is able to discern. We are also leaving out the mechanism through which the PCA is able to find structure, more on that later. Looking at the explained variance ratios, one can be tempted to consider only the first component and dismiss the others as noise since there's a such a big difference between the first and second explained variances. Careful analysis however, reveals that further components also encase relevant information about the system. Here we follow the treatment in [21]. The weights of the second component can be seen in Fig. 4.2d. It has a striking similarity with Fig 4.3, which is obtained by plotting the following expression:

$$\mathbf{w}'_{(2)} = \frac{1}{L} \left[cos(\mathbf{r}_1 \mathbf{k}_1), cos(\mathbf{r}_2 \mathbf{k}_1), ... \right] + \frac{1}{L} \left[cos(\mathbf{r}_1 \mathbf{k}_2), cos(\mathbf{r}_2 \mathbf{k}_2), ... \right]$$

where $\mathbf{r}_i$ is the lattice site and $\mathbf{k}_1 = [0, 2\pi/L]$, $\mathbf{k}_2 = [2\pi/L, 0]$ are the lowest Fourier wave vectors. It's possible to associate the first component with $\mathbf{k} = [0, 0]$, so it appears that the PCA Fourier transforms the data. This is allowed, since after all a Fourier transform is a linear transformation as the PCA is.

After the fact, this might seem trivial, but one should consider that the PCA was not instructed to represent the data in k space and this is **not** what PCA does in general. In a way this tells us something we already knew, that representing the data in Fourier space is a great method to get insights of our models. Our analysis doesn't finish here though. For once we should analyze how effectively the PCA performs this transformation and what else can we extract from the these results.

Another point of this discussion is that it's useful to take look at the modulus of

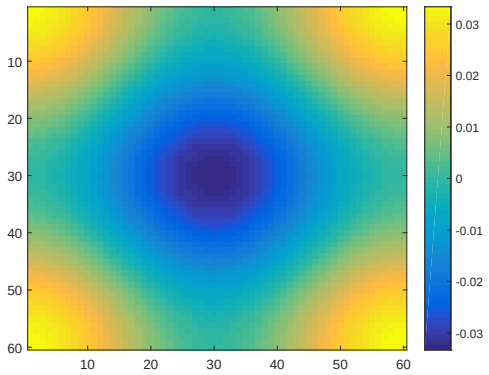


FIGURE 4.3: Plot of $\mathbf{w}'_{(2)}$. The similarity with Fig. 4.2d is obvious.

the second component in function of the temperature as portrayed in Fig. 4.4b. The Magnetic susceptibility is proportional to the expectation value of the Fourier mode that's associated with the second component. If this is true, then we can use the scaling relation (2.5) to get an estimate of the critical temperature in thermodynamic limit by using:

$$T'_c = T_c + kL^{-1} \tag{4.1}$$

where the fit parameters are T_c and k. Taking the maxima of the plots to be the analogue critical temperature T'_c, and plot them against $1/L$ (see Fig. 4.4c), we get the linear relation from which it's possible to extract the value

$$T_c = 2.266(14)$$

This value is in accordance with the theoretical result $T_c/J = 2/ln(1+\sqrt{2}) \approx 2.269$ [35] [1].

There's something else to be learned in the analysis of the mechanism that allows the PCA to find weight vectors that are meaningful. In this, reference [41] was helpful in shedding light on the subject and we shall do a similar analysis to the one performed there.

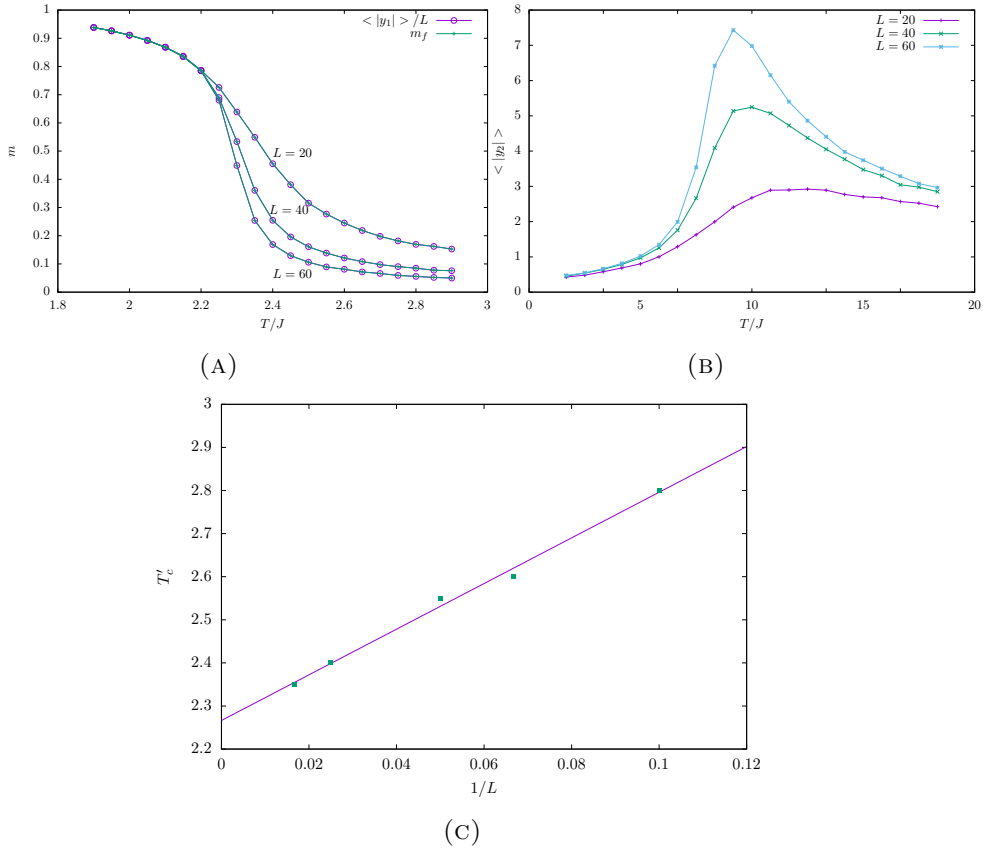
Suppose that instead of a full lattice the only thing passed to the PCA would be the configurations of two interacting spins. This is our toy model. If the lattice only has two spins, and if S_1 is the spin of the first lattice site, then the favored configurations for the ordered phase take the form

$$x_l = (S_1, S_1)$$

and for the disordered phase

$$x_{h1} = (S_1, S_1) \qquad x_{h2} = (S_1, -S_1)$$

[1] The error in this quantity is just the the one obtained through the linear regression. This is because the true error depends on the resolution of the temperature used and of the error of the average used to calculate $|y_2|$ for a given value of temperature. The combination of these two makes it impossible to have a realistic estimate for the error in T'_c.



(A) (B)

(C)

FIGURE 4.4: (a) The magnetization per site m_f and the magnetization obtained through PCA, $< |y_1| > /L$ (the average $<>$ is over the 5000 samples per temperature), in function of temperature for different lattice sizes. The two quantities are nearly indistinguishable. (b) Average of the modulus of the second component in function of the temperature.(c) Critical temperature analogue T_c' taken from the maxima of Fig. 4.4b in function of the inverse of the lattice size. The y intercept of the fit is

$$T_c = 2.266(14)$$

are equally likely. Suppose we feed configurations that span both phases, then $X^{\mathbf{T}}X \equiv A$ from eq 3.1 needs to be computed:

$$A = X^{\mathbf{T}}X = \frac{1}{M}\sum_n x_n^T x_n$$

if M is the number of total configurations x_n is the nth configuration. If a fraction p of the data corresponds to the ordered phase, then $1-p$ corresponds to the disordered one and we can write:

$$A = pA_h + (1-p)A_l \tag{4.2}$$

if we have a large enough data set then,

$$A_l = \langle x_l^T x_l \rangle$$

where <> denotes the average over the values that x_l can take. So in this case

$$A_l = \frac{1}{2}\sum_{S_1} S_1^2 \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

For A_h:

$$A_h = \frac{1}{2}\left(\langle x_{h1}^T x_{h1}\rangle + \langle x_{h2}^T x_{h2}\rangle\right)$$

$$A_h = \mathbb{I}$$

Solving equation 3.1 will yield the weight vectors and corresponding eigenvalues. A_h is the identity matrix so the largest, and consequently the principal eigenvalue will come from A_l, which has:

$$\lambda_1 = 2 \Rightarrow \mathbf{w}_{(1)} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\lambda_2 = 0 \Rightarrow \mathbf{w}_{(2)} = \frac{1}{\sqrt{2}}\begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

The important thing to note is that similarly to what happens in the case presented above, the first weight vector is constant. It's easy to generalize what was just explained to an LxL sized lattice. In that case A_l is just a matrix of ones with dimensions NxN. The characteristic polynomial of this matrix is $(\lambda - N)\lambda^{N-1}$ so the highest eigenvalue will always be $\lambda_1 = N$ and its eigenvector

$$\mathbf{w}_{(1)} = \frac{1}{L}\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

which is precisely what was obtained through the data analysis. Of course the data representation here is overly simplified. Low temperature configurations are more than spins either all up or down and that's why the second eigenvalue is not meaningful in this treatment even though it is so in the data analysis.

To achieve the same analytic result for the second component, the simplified picture of starting out from a lattice with only two spins would no longer work. The second component will be related with the first excited state which in a full lattice corresponds to flipping only one of the spins while the other remain aligned, but in a lattice with just two spins it would be nonsensical. If one of the two spins is flipped, the result is a configuration that could be just as easily a high temperature configuration. The PCA, however looks at the full dataset in a full lattice, so it's somewhat intuitive that if it is able to detect the zeroth mode through the method just described, then it's able to detect spin-wave excitations in a similar fashion.

There is one final evidence that the PCA manages to do this effectively and that has to do with the relation of the value of the explained variance and the Fourier coefficients of the Fourier transformed data. Note that the explained variance is different from the explained variance ratio, the difference being that the explained variance is not normalized to the sum of every eigenvalue, or in other words it's simply the eigenvalue

of matrix $\mathbf{X}^T\mathbf{X}$. In practice they're not that different, but the explained variance makes the point of this comparison more obvious.

TABLE 4.1: Comparison of the modulus of the highest Fourier coefficients $|c(\mathbf{k})|$ of the Fourier transformed data for $L = 20$ with the highest explained variance values. To obtain $|c(\mathbf{k})|$, each sample was Fourier transformed and then they were averaged over the 5000 samples per temperature)

$\lvert c(\mathbf{k})\rvert$	λ_i
206	151.0
43	7.7
43	7.7
43	7.7
43	7.7
35	4.8
35	4.8

Regarding the Fourier coefficients, there's one that tops all the rest (of course it's the zeroth mode), then there are 4 modes that presumably have the same energy, hence, similar coefficients and the list goes on for lower energy modes. The same pattern can be seen in the explained variance values. The relative difference between the first coefficient and the second one is not the same as for the the first and second explained variances, but it turns out that theirs squares are related as can be seen in Table 4.2. What happens is that because the explained variance values come from the eigenvalues of $\mathbf{X}^T\mathbf{X}$ then, if one is to make a comparison with the Fourier coefficients, it's necessary to square them.

TABLE 4.2: Comparison of the ratio between squared Fourier coefficients with the corresponding ratios of the explained variance. The i and j indices refer to values that are in the same range. As an example, looking at table 4.1, $|c(\mathbf{k})| = 206$ would be labeled as c_1 and the four subsequent values as c_2. The same applies for the values of the explained variance λ_i. δ is the percent relative error.

i/j	c_i^2/c_j^2	λ_i/λ_j	$\delta(\%)$
1/2	22.5	19.6	13
1/3	34.4	31.3	9
1/4	55.6	52.4	6
2/3	1.5	1.6	4
2/4	2.5	2.7	8

Not only is the PCA able to detect the Fourier modes, but it is also able to correctly estimate a quantity proportional to the Fourier coefficients. We were unable to take any meaning from the proportionality constant.

The ratios compared in Table 4.2 are right within a $\approx 10\%$ error margin. It is likely the case that if more data were fed to the PCA this error value would decrease. This is not a trivial result and shows the robustness of PCA.

That concludes the analysis for the Ising ferromagnet. Next we see how PCA fares in the Ising Antiferromagnet model.

4.1.2 Antiferromagnetic Ising

As noted previously, in physical terms there is not much difference between the anti-ferromagnetic and ferromagnetic Ising models. The antiferromagnet presents some frustration and the ordered phase is more complex, consisting of a checkerboard pattern. This should then present a further test in the abilities of the PCA to detect patterns in the data and to meaningfully transform it into principle components.

The algorithm was fed a matrix X in which each of the rows is a configuration sample for a given temperature (the spin values are just aligned in a row vector and put together to create X). 5000 samples per temperature value were generated in this case. The PCA was carried out by solving the eigenvalue problem

$$X^{\mathbf{T}} X \mathbf{w}_{(i)} = \lambda_i \mathbf{w}_{(i)}$$

as discussed in the Methods chapter.

Results

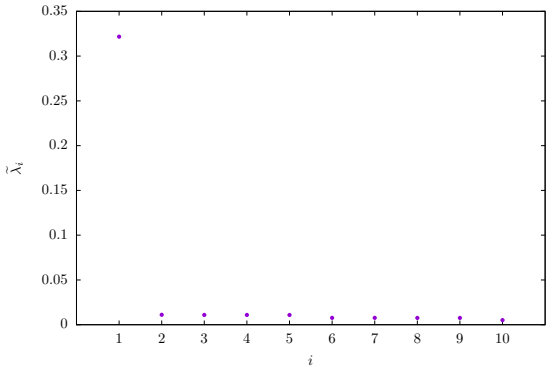


FIGURE 4.5: First 10 explained variance ratios for $L = 40$. They denote their relative importance in reproducing the data.

The obtained explained variance ratios can be seen in Fig. 4.5. Similarly to what happened in the ferromagnet, one component dominates over all others.

Fig. 4.6 sums up the results for the Ising antiferromagnet. Again the first principal component separates the configurations in clusters for low and high temperature and we can already see from the shape of the second weight that it will correspond to a Fourier mode of the system. The figures are very similar to what we can see for the Ising model, which is only a consequence of the underlying physics being very similar. This model posed as much of a challenge to the PCA as the regular ferromagnet did.

Discussion

We repeat some of the analysis made for the Ising model. First we note that the weights of the second component correspond to the plot of the following expression:

$$\begin{aligned}\mathbf{w}'_{(2)} = &\frac{1}{L}\left[\exp(i\mathbf{r}_1 \cdot \mathbf{c})\cos(\mathbf{r}_1\mathbf{k}_1), \exp(i\mathbf{r}_2 \cdot \mathbf{c})\cos(\mathbf{r}_2\mathbf{k}_1), ...\right] + \\ &+\frac{1}{L}\left[\exp(i\mathbf{r}_1 \cdot \mathbf{c})\sin(\mathbf{r}_1\mathbf{k}_2), \exp(i\mathbf{r}_2 \cdot \mathbf{c})\sin(\mathbf{r}_2\mathbf{k}_2), ...\right]\end{aligned}$$

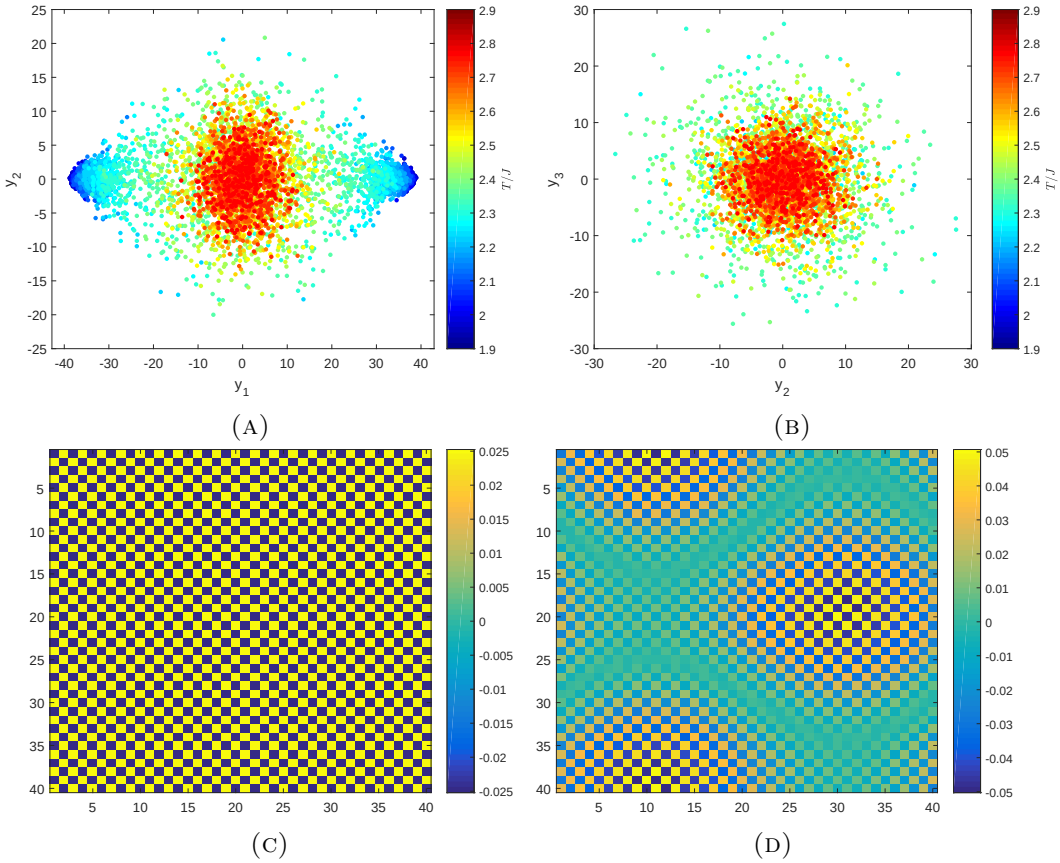


FIGURE 4.6: (a)Projection of the antiferromagnetic Ising configurations on the two main principal components for $L = 40$. Only 300 configurations per temperature are represented to not obscure the image. (b)Projection of the second and third component. (c)Weights of the first component for every lattice site for $L = 60$. Notice that $1/L \approx 0.017$ (d)Weights for the second component plotted on the lattice.

with $\mathbf{c} = [\pi, \pi]$, $\mathbf{k}_1 = [2\pi/L, 0]$ and $\mathbf{k}_2 = [0, 2\pi/L]$. The plot of this expression can be seen in Fig. 4.7 and the similarity with the actual plot of the second weight (Fig. 4.6d) is obvious.

The plot of the modulus of the first component (Fig. 4.8a) vs temperature, along with the fact that the weight of the first component is a checkerboard pattern (Fig. 4.6c) is a clear indication that the first component finds the staggered magnetization of the system. The modulus of the second component vs temperature has the exact same shape as in the Ising ferromagnet (Fig. 4.4b). It was concluded there that that shape was characteristic of the specific heat of the system, being proportional to it. There's no reason for it be any different in this case.

We Fourier transformed the data and compared the values of the explained variance values with the Fourier coefficients and we found exactly the same is in Table 4.1. This further proves the point of the PCA doing Fourier transformation.

The take away from the analysis of this model is perhaps that PCA is robust enough to detect more complex patterns than the ones found in the Ising ferromagnet, which might seem somewhat trivial. The PCA is not confused by the change from ferromagnet to antiferromagnet and is able to identify that the underlying physics is the same for

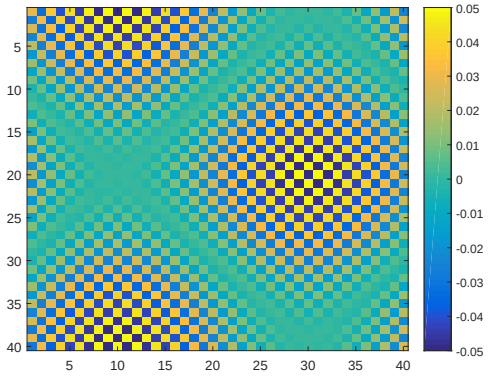
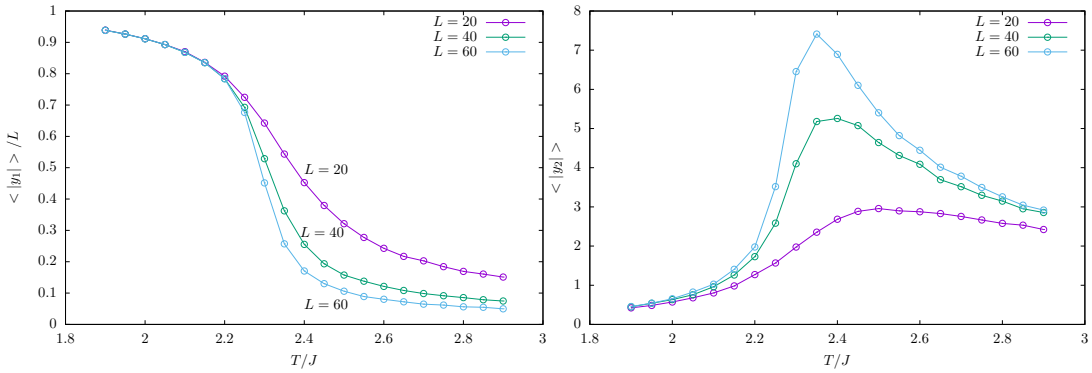


FIGURE 4.7: (d)Plot of $\mathbf{w}'_{(2)}$. The similarity with Fig. 4.6d is obvious.



(A) (B)

FIGURE 4.8: (a) Average of the modulus of the first component divided by the lattice size, $< |y_1| > /L$ (the average $<>$ is over the 5000 samples per temperature), in function of temperature for different lattice sizes. This plot is no different than that of the staggered magnetization for this system. (b)Average of the modulus of the second component $< |y_2| > /L$ in function of the temperature. It's clearly proportional to the specific heat of the system.

both cases. Of course it all comes down to the fact that ultimately what PCA does is to Fourier transform the data, but nothing grants that because it was able to do it for the ferromagnet it would also be able do it for the antiferromagnet and in a way, that is what we intended to test. Next, we see how PCA behaves when confronted with a seriously frustrated model, where disorder is prevalent, patterns are scarce and much harder to pick up on.

4.1.3 Triangular Ising Antiferromagnet

As was just mentioned, the Triangular Ising Antiferromagnet (TIAM) is a frustrated model, remaining in a paramagnet phase for all finite temperatures, where disorder is prevalent. Here we take a dab at analyzing it with PCA.

The PCA was fed a matrix X in which each of the rows is a configuration sample for a given temperature (the spin values are just aligned in a row vector and put together to create X). 2500 samples per temperature value were generated in this case.The PCA was carried out by solving the eigenvalue problem

$$X^{\mathbf{T}} X \mathbf{w}_{(i)} = \lambda_i \mathbf{w}_{(i)}$$

as discussed in the Methods chapter.

Results

Looking at the explained variance ratios in Fig. 4.9, the analysis reveals the presence of two dominant principal components, even though its explained ratios are relatively much smaller when compared with the previously tested models.

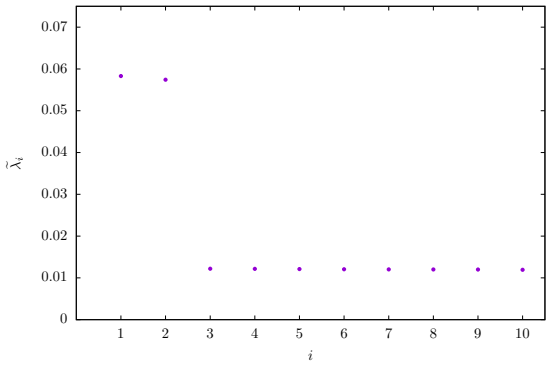


FIGURE 4.9: First 10 explained variance ratios for $L = 21$. They denote their relative importance in reproducing the data.

Projecting the data on the main components yields Fig. 4.10a. The PCA seems to distinguish the data into low and high temperature clusters with the low temperature cluster having the particularity of having a circular shape.

It's hard to visualize the shape of the two main weights because of the mapping from triangular to square lattice, so in Fig. 4.11 a schematic representation of the weights in a triangular lattice is depicted.

Discussion

The TIAM remains disordered at every finite temperature so it should come as no surprise that the values for the explained ratios are smaller. There's no obvious pattern that the PCA can lock into, so the patterns that it is able to find are not very good at reproducing the real data causing a low explained ratio. If anything, it should come as a surprise that the PCA is able to find any pattern at all in a system that is always disordered.

Looking at the data projected in the first two principal components it almost appears as if the PCA is able to distinguish between two phases, but that analysis is misleading.

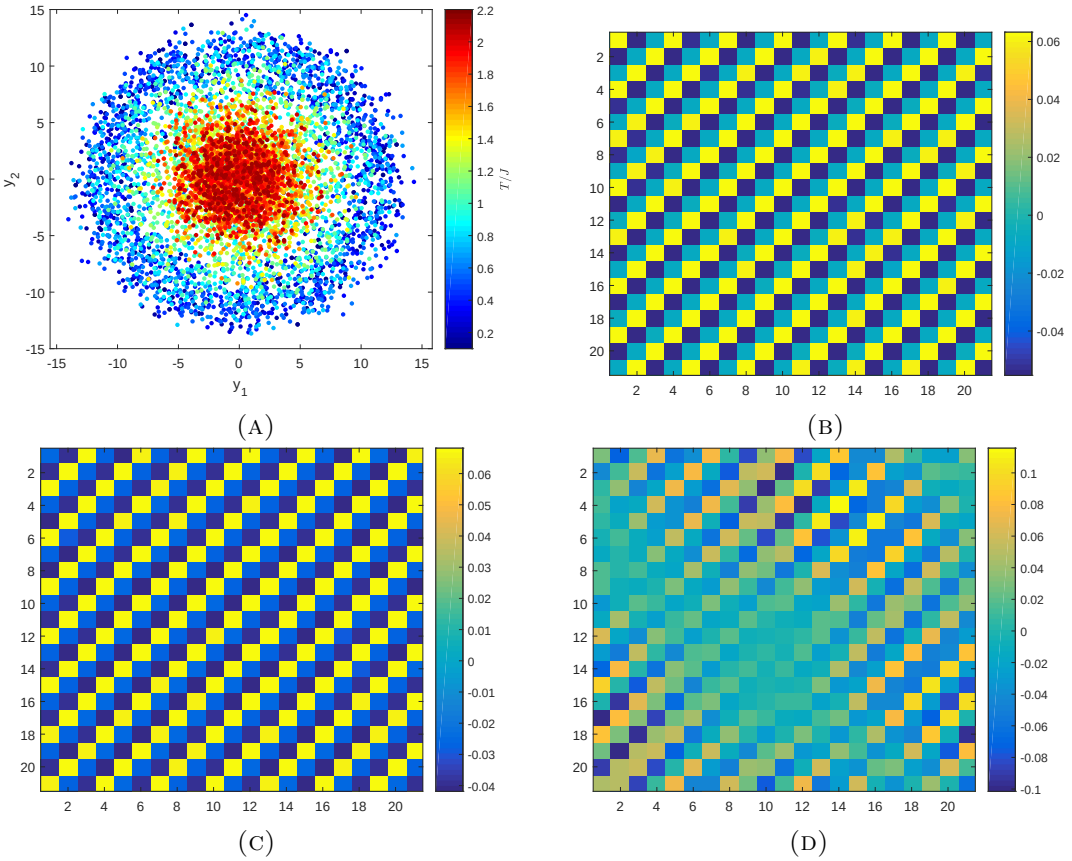


FIGURE 4.10: (a)Projection of the triangular configurations onto the first two principal components for a lattice with $L = 21$. (b)Plot of the weight corresponding to the first principal component vs the lattice site. The configurations are displayed in a square because the triangular lattice was mapped onto a triangular lattice in the simulation. (c)Plot of the weight corresponding to the second principal component vs the lattice site. (d)Plot of the weight corresponding to the third principal component vs the lattice site.

To see where the first two components come from, we resort to the same matrix analysis we've used earlier for the Ising model. The idea is then to construct the matrix A as in eq. **??**. To quickly recap:

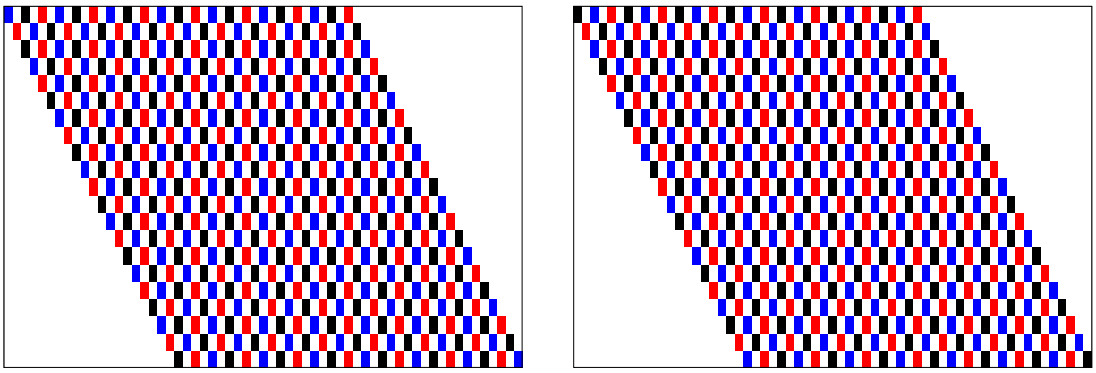
$$A = X^{\mathbf{T}} X = \frac{1}{M} \sum_n x_n^T x_n$$

where M is the number of total configurations x_n is the nth configuration. If a fraction p of the data corresponds to the ordered phase, then $1-p$ corresponds to the disordered one and we can write:

$$A = pA_h + (1-p)A_l \tag{4.3}$$

if we have a large enough data set then,

$$A_l = \langle x_l^T x_l \rangle$$



(A) (B)

FIGURE 4.11: Schematic representation of the weights on a triangular lattice. Similar values in the weight matrix are represented by the same color.

where <> denotes the average over the values that x_l can take. The same holds for A_h.

For that we should construct the states x_h/x_l corresponding to low and high temperature. Let's suppose the system is only comprised of a single triangle. Then, of the $2^3 = 8$ possible states, 6 are energetically favorable for low temperature: the three states represented in the figure bellow plus their flipped version.

If the value of one of the spins is S, then the vectors corresponding to the above configurations are

$$x_{l1} = (S, -S, S) \qquad x_{l2} = (S, S, -S) \qquad x_{l3} = (-S, S, S)$$

the ones regarding high temperature include the all of the x_l plus the two where the 3 spins are aligned:

$$x_{h4} = (S, S, S)$$

To compute A_l,

$$A_l = \frac{1}{3}\left(\langle x_{l1}^T x_{l1}\rangle + \langle x_{l2}^T x_{l2}\rangle + \langle x_{l3}^T x_{l3}\rangle\right)$$

$$A_l = \frac{1}{3}\sum_S S^2 \left(\begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 1 & -1 \\ 1 & 1 & -1 \\ -1 & -1 & 1 \end{bmatrix} + \begin{bmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \\ -1 & 1 & 1 \end{bmatrix} \right)$$

$$A_l = \frac{1}{3}\begin{bmatrix} 3 & -1 & -1 \\ -1 & 3 & -1 \\ -1 & -1 & 3 \end{bmatrix}$$

The calculation of A_h is not interesting since it will once again be the identity matrix, as you can check for yourself, and it will contribute eigenvalues that won't be the largest for the matrix A.

We now compute the eigenvalues and eigenvectors of A_l according to eq. 3.1.

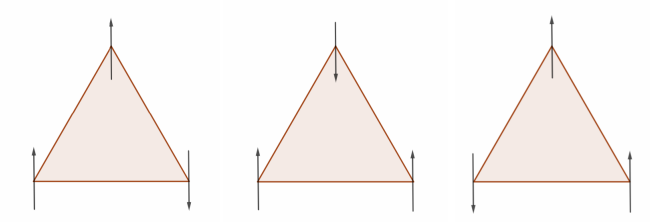


FIGURE 4.12: Three of the six possible configurations corresponding to x_l.

$$\lambda_1 = 4 \Rightarrow \mathbf{w}'_{(1)} = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

$$\lambda_2 = 4 \Rightarrow \mathbf{w}'_{(2)} = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}$$

$$\lambda_3 = 1 \Rightarrow \mathbf{w}'_{(3)} = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

This results can trivially be extended to the whole lattice in a process equivalent to what was done for the Ising model. One need only to be careful to guarantee that every triangle in the lattice is one of the x_l configurations. The obtained double valued eigenvalue corresponds to the two leading principal components. In actuality, what is depicted in figure 4.10b is the full lattice equivalent of a linear combination of $\mathbf{w}'_{(1)}$ and $\mathbf{w}'_{(2)}$, but that's just a consequence in the freedom that exists in the choice of the basis used to define the hyperplane spanned by $\mathbf{w}_{(1)}$ and $\mathbf{w}_{(2)}$.

Since the two leading components are really two faces of the same coin, it's useful to combine them in a more useful quantity. Inspired by the circular shape of the projection of the data, we define R_{12} as being:

$$R_{12} = \langle y_1^2 + y_2^2 \rangle$$

where the <> denotes the average over the 2500 values for each temperature. This quantity is depicted in Fig. 4.13.

This whole analysis suggests that the two main principal components measure the likelihood of the system being in the full lattice equivalent of the low energy states shown in Fig. 4.12. In order to test this hypothesis, we'll calculate this probability, compare it with R_{12} and check if there's any similarity. The probability is given by the Boltzmann weight of the configurations divided by the partition function:

$$P(T) = \frac{n_l e^{-H_l/T}}{n_l e^{-H_l/T} + n_h e^{-H_h/T}} = \frac{6e^{1/T}}{6e^{1/T} + 2e^{-3/T}}$$

Where n_l and n_h are the number of states with low and high energy respectively and H_l/H_h is the energy of said states.

Compared with figure 4.13 it's easy to notice that the drops occur at around the same values of temperature. Furthermore, the fact that the probability only declines

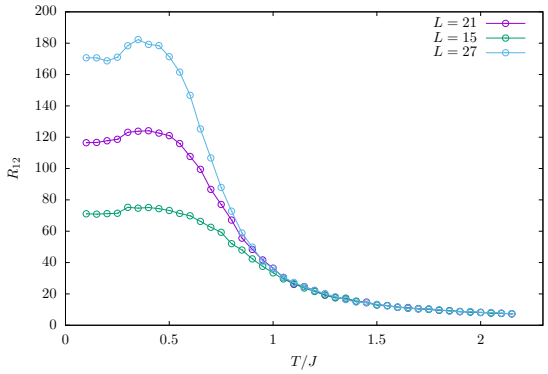


FIGURE 4.13: Plot of R_{12} in function of the temperature where $R_{12} = \langle y_1^2 + y_2^2 \rangle$

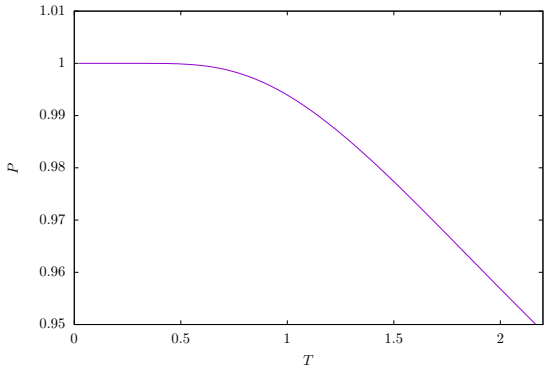


FIGURE 4.14: Probability of being in one of the low energy states in function of temperature.

ever so slightly may account for the low value of the explained variance ratio that these components have. It should be noted that the little bump for very low temperatures in the plot of R_{12} is probably just an artifact of the data that unfortunately we were not able to extinguish.

While it's obvious that the remaining components present some structure, it's also hard to make any sense of it. If we were to feed more data we would probably find the patterns corresponding to spin-waves.

There's one final aspect that's perhaps worth mentioning. The values of the explained variance ratios seem to be related with the "how much" of the pattern represented by its correspondent weight is in the data, but it also appears to be dependent on the lattice size. Oddly enough it seems that the value of $\widetilde{\lambda}_i L$ remains constant for different lattice sizes, but I can't provide any explanation for this or for the value that it takes.

To test if this is an exclusive property of this model, the calculation was repeated for the previously tested models and it turns out that this is the only instance of where the explained variance ratios seem to be inversely proportional to the lattice size.

We conclude the analysis of this model here. Next we see how PCA fares in cracking the king of frustrated models, analyzing what happens for the Ising Antiferromagnet

TABLE 4.3: First explained variance ratio times the lattice length.

$\widetilde{\lambda}_1 \times 10^2$	L	$\widetilde{\lambda}_1 L$
4.55	27	1.23
5.83	21	1.22
8.07	15	1.21

in a Kagome lattice.

4.1.4 Antiferromagnetic Ising in a Kagome lattice

To construct the Kagome lattice, the starting point is the mapping from square to triangular lattice that was done for the triangular AF. By setting the correct spins in the square lattice to zero, a Kagome lattice was obtained. These null spins were not removed before feeding the configurations into the PCA because they have, in practice, no impact on the final result. If each site is seen as one of the N dimensions of the data, the sites that were set to zero span a subspace for which the values of every configurations are the same, so the PCA will have no variance to detect along those dimensions. The only side effect is that the there will be explained ratios detected for those dimensions that will be extremely low. The best way to understand this is to give the example of a random data set with three dimensions (x, y, z) but that has $z = 0$ for every point. If you perform PCA on it, since it is just an orthogonal transformation, the z direction will also have to be represented in the new basis, but it will have a low explained variance ratios, because there is no variance to explain in that direction. The same will happen here.

The PCA was fed a matrix X in which each of the rows is a configuration sample for a given temperature (the spin values are just aligned in a row vector and put together to create X). 2000 samples per temperature value were generated in this case. The PCA was carried out by solving the eigenvalue problem

$$X^{\mathbf{T}} X \mathbf{w}_{(i)} = \lambda_i \mathbf{w}_{(i)}$$

as discussed in the Methods chapter.

Results

Looking at the values of the explained variance ratios (Fig. 4.15c) we see that this time the PCA has a much harder time detecting any structure in the data. The ratios have all very small values, with the highest, accounting for little over 0.5% of the variance of the data for $L = 20$ (comparing with the Ising ferromagnet and the Triangular antiferromagnet, the first component accounted for more than 30% and 5% respectively). The explained variance ratios slowly decrease until the 300th value and then suddenly drop off to zero.

Furthermore, projecting the data onto the first two components (Fig. 4.15a), there is no clustering of high/low temperature data, indicating that the algorithm can't really distinguish between high and low temperature regimes. The plot of the weight corresponding to the first component (Fig. 4.15b) contributes to this same analysis. There is no clear pattern and the picture is very noisy.

Discussion

First, it's worth pointing out that the last 100 explained ratios are essentially zero because of the sites that are set to zero in building the Kagome lattice as was explained above. If we were to randomly assign values for the remaining sites, then each explained variance ratio should have an equal value of $1/N_{sites}$, N_sites being the number of actual lattice sites. For $L = 20$ this would be $1/300 \approx 3.3 \times 10^{-3}$ but the first explained ratio is around double of that value. This doesn't mean that the PCA was successful though, as it's still a very small value, and if that was the case we would be able to see

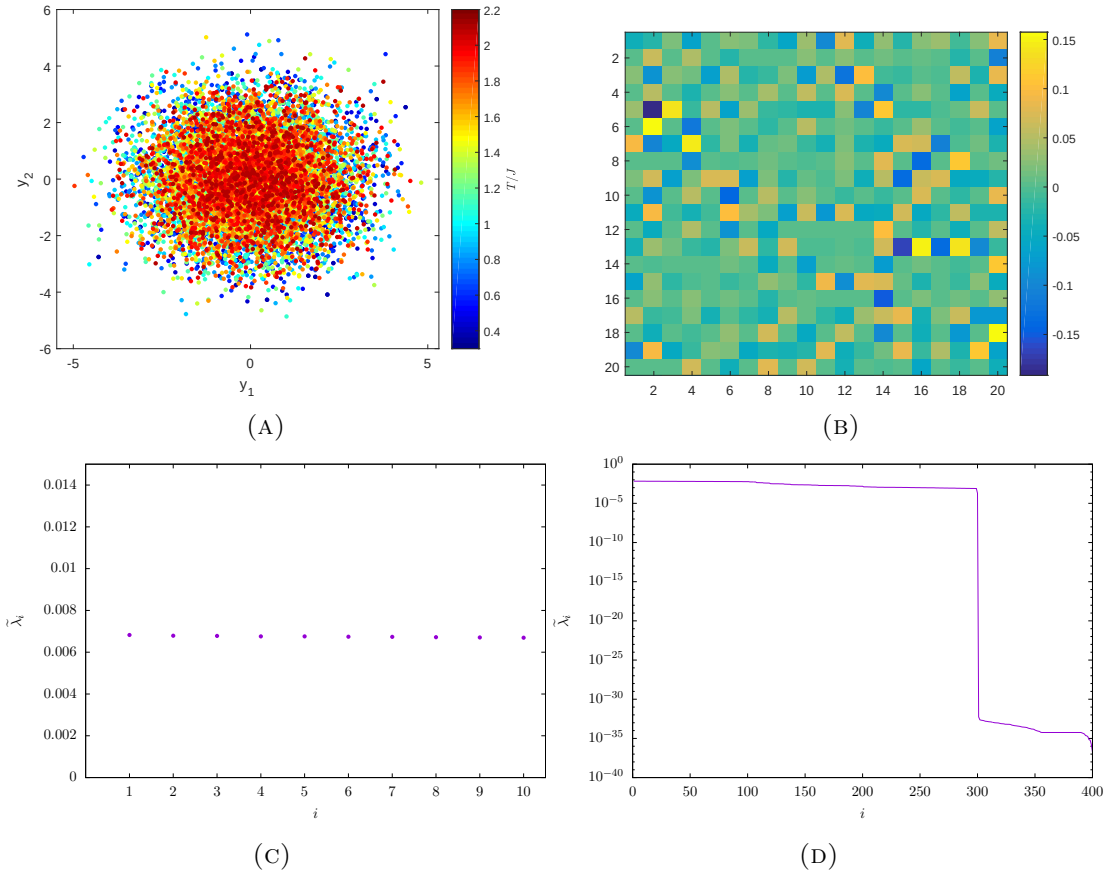


FIGURE 4.15: (a)Projection of the Kagome configurations onto the first two principal components for a lattice with $N = 300$. Only 300 samples per temperature are depicted to not obscure the image. (b) Plot of the weight corresponding to the first principal component. $\mathbf{w}_{(1)}$. (c) Explained variance ratios of the first 10 principal components for $L = 20$ (d) Explained variance ratios for the first 400 principal components for $L = 20$.

a pattern in the weight of the component. Looking at the weight in Fig. 4.15b, we see that there's a very blurred sort of checkerboard pattern, but only in some areas of the lattice. The fact that is very blurred tells us that it's probably just a consequence of some configurations having similar patterns by chance and so PCA distinguished them from the rest, but that doesn't mean these configurations were special.

Ultimately this result is intimately related with the physics of this model. The system is truly disordered for all temperatures, truly here meaning that even if perturbed it will remain disordered (the same doesn't apply for the triangular case for example). Given the analysis that was given for the triangular case, one might expect that the same explanation might apply here, since we could also define basic triangular cells. However, it would seem that the disorder in the system disturbs the aligned triangle formation enough so that the PCA is not able to detect anything. The sites in the Kagome are less connected than in the triangular lattice (they have only four nearest neighbors instead of the six from the triangular). Lower connectivity means lower correlations as well, and apparently this is enough to throw the PCA off.

In a way this result is not surprising given the nature of the Kagome model, but it

is important in establishing the boundaries of what the PCA can and can't do. Next we move on to a continuous model. Intuitively, it's easy enough to understand that it must be easier to detect patterns in quantities that have discrete binary values than in quantities that span a range of continuous values, so the next step is to test whether or not PCA is able to overcome that challenge.

4.1.5 XY model

The goal here is to test how PCA fares for models with continuous states. Also, it will be interesting to see if the PCA is able to detect more complex non-linear features of the data, namely vorticity. For this model, since the values of the states taken by each site are no longer binary, each row of the matrix X fed into the PCA is a list with two values per site, namely $[\cos\theta_i, \sin\theta_i]$. This should make the analysis easier to interpret as opposed to feeding it only the θ_i values. The PCA was carried out by solving the eigenvalue problem

$$X^{\mathbf{T}} X \mathbf{w}_{(i)} = \lambda_i \mathbf{w}_{(i)}$$

as discussed in the Methods chapter.

Results

As can be seen in Fig. 4.16b, the PCA detected two dominant components, together accounting for 24% of the total variance in the data. Projecting the data in these components, we see a clear separation of the data in low and high temperature regimes. The low temperature cluster has a circular shape. The analysis done for the previous models allows us to make an educated guess of what might be happening here. Probably, the PCA is detecting two lowest energy Fourier modes that correspond to the highest two eigenvalues, and it is also likely that if we performed the matrix eigenvalue analysis done for the Ising model, one would find two double valued eigenvalues that top all the others. This is indeed the case as will be shown in the discussion section.

The weights of the first four components are depicted in Fig. 4.17. The PCA seems to find only correlations between the x and y components of the spins separately, so the weights were also separated to get a better grasp of the real picture. While it is true that the PCA seems to make a distinction between the horizontal and vertical components of the spins, it is also clear that they are not completely independent as they present similar patterns for all the 4 depicted components.

Discussion

The circular shape of the lower temperature samples in the projection into the first components inspires us to transform the data according to

$$R_{12} = \left\langle y_1^2 + y_2^2 \right\rangle / N$$

Plotting R_{12} as we did for the triangular lattice, we get a plot reminiscent of the magnetization. An easy way to check that it is indeed the magnetization is to check that at T_{KT}, R_{12} scales with $L^{-1/8}$ [14]. This was done in Fig. 4.16d and it is clear that this relationship holds.

To see where the first two components come from, we resort to the same matrix analysis we've used earlier for the Ising models. The idea is then to construct the matrix A as in eq. **??**. To quickly recap,

$$A = X^{\mathbf{T}} X = \frac{1}{M} \sum_n x_n^T x_n$$

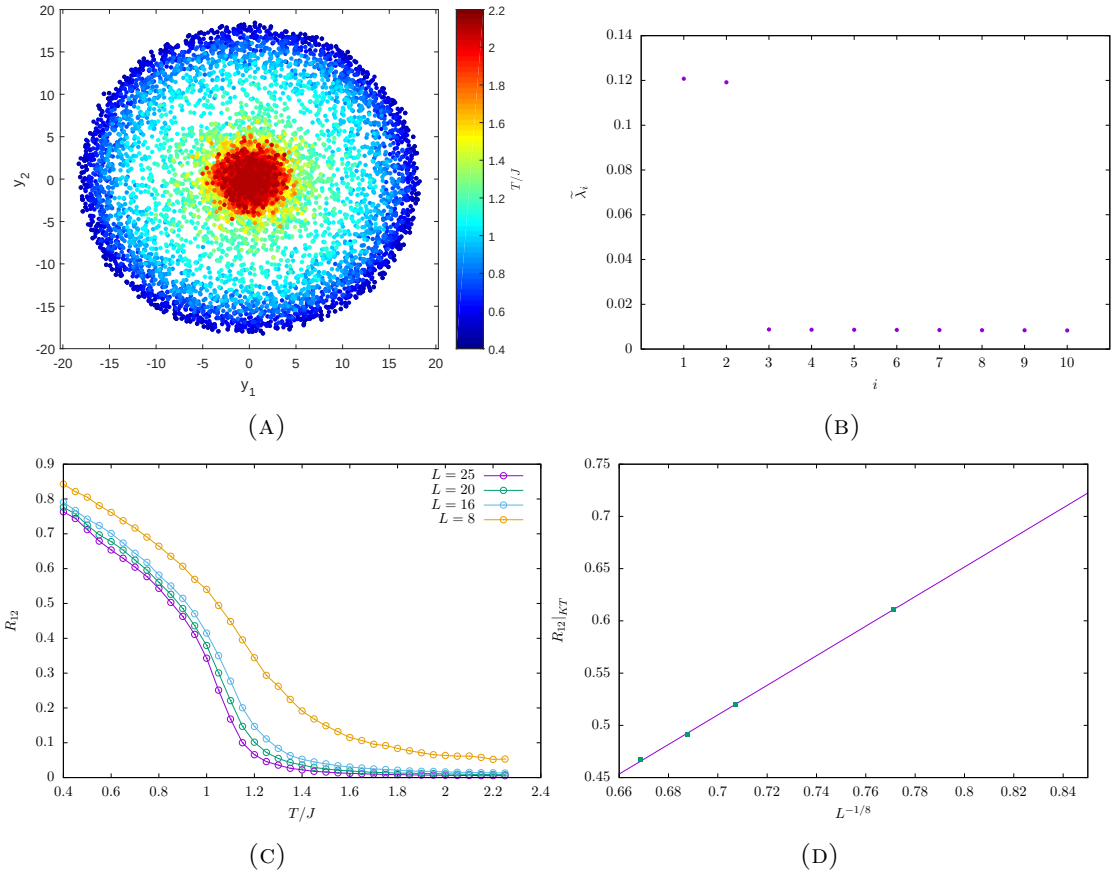


FIGURE 4.16: (a)Projection of the XY configurations onto the first two principal components for a lattice with $L = 20$. Only 300 samples per temperature are depicted so as to not obscure the picture. (b) Explained variance ratios of the first 10 principal components for $L = 20$. (c) Plot of R_{12} in function of the temperature where R_{12} is given by $R_{12} = \langle y_1^2 + y_2^2 \rangle / N$ (d) Values of R_{12} take at T_{KT} in function of $L^{-1/8}$. Checking this scaling relationship also serves the purpose of testing the accuracy of the PCA. The linear relationship is clear, further confirmed by the value of reduced chi-squared $\chi^2 = 2 \times 10^{-6}$. The values of the parameters in the linear regression $y = mx + b$ are $m = 1.42(2)$ and $b = -0.48(1)$.

if M is the number of total configurations x_n is the nth configuration. If a fraction p of the data corresponds to the ordered phase, then $1 - p$ corresponds to the disordered one and we can write:

$$A = pA_h + (1 - p)A_l \tag{4.4}$$

if we have a large enough data set then,

$$A_l = \langle x_l^T x_l \rangle$$

where <> denotes the average over the values that x_l can take. The same holds for A_h.

In order to proceed, it's necessary to construct the states x_h/x_l corresponding to low and high temperature configurations. In this case, we start out considering a lattice

with only 2 sites A and B. Their configurations are given by the vector:

$$x_n = [\cos\theta_A, \sin\theta_A, \cos\theta_B, \sin\theta_B]$$

For low temperature configurations we expect to see the spins aligned, so $\theta_A \approx \theta_B$. Note that they can align themselves in any direction so $\theta_A \in [0, 2\pi]$ The low temperature vector will then be

$$x_l = [\cos\theta_A, \sin\theta_A, \cos\theta_A, \sin\theta_A]$$

and also,

$$A_l = \langle x_l^T x_l \rangle = \frac{1}{2\pi} \int_0^{2\pi} x_l^T x_l d\theta_A$$

finally

$$x_l^T x_l = \begin{bmatrix} \cos^2\theta_A & \cos\theta_A\sin\theta_A & \cos^2\theta_A & \cos\theta_A\sin\theta_A \\ \cos\theta_A\sin\theta_A & \sin^2\theta_A & \cos\theta_A\sin\theta_A & \sin^2\theta_A \\ \cos^2\theta_A & \cos\theta_A\sin\theta_A & \cos^2\theta_A & \cos\theta_A\sin\theta_A \\ \cos\theta_A\sin\theta_A & \sin^2\theta_A & \cos\theta_A\sin\theta_A & \sin^2\theta_A \end{bmatrix}$$

taking the integral, the result is

$$A_l = \frac{1}{2} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

For high temperature, it is assumed that they will have random values, and so A_h will again be the identity. The important eigenvalues and eigenvectors come from A_l then. In this case there's two double valued eigenvalues, one of them being zero, so not very relevant since we're looking for the highest possible. In sum

$$\lambda_1 = 2 \Rightarrow \mathbf{w}'_{(1)} = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

$$\lambda_2 = 2 \Rightarrow \mathbf{w}'_{(2)} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Projecting high temperature data into $(\mathbf{w}'_{(1)}, \mathbf{w}'_{(2)})$ will result in a vector proportional to $(\sin\theta_A + \sin\theta_B, \cos\theta_A + \cos\theta_B)$ and projecting low temperature data will result in something proportional to $(\sin\theta_A, \cos\theta_A)$. Hence the circular shape observed in Fig. 4.16a.

This argument can be easily extended to the whole lattice and explains the values of the first two components seen in Fig. 4.17.

Unfortunately it would seem like the PCA is not able to detect properties involving the vortices nor the superfluid density. In a way this is somewhat expected, because

these are non-linear features of the inputs $[\cos\theta_i, \sin\theta_i]$, and the PCA is only a linear transformation so it has trouble detecting these.

Lastly, by looking at Fig. 4.17 it is very apparent that we are looking at Fourier modes of the system, or to be more precise, their x and y components. This is not a new feature of our analysis, as we concluded the same happened for all models, but the Kagome antiferromagnet.

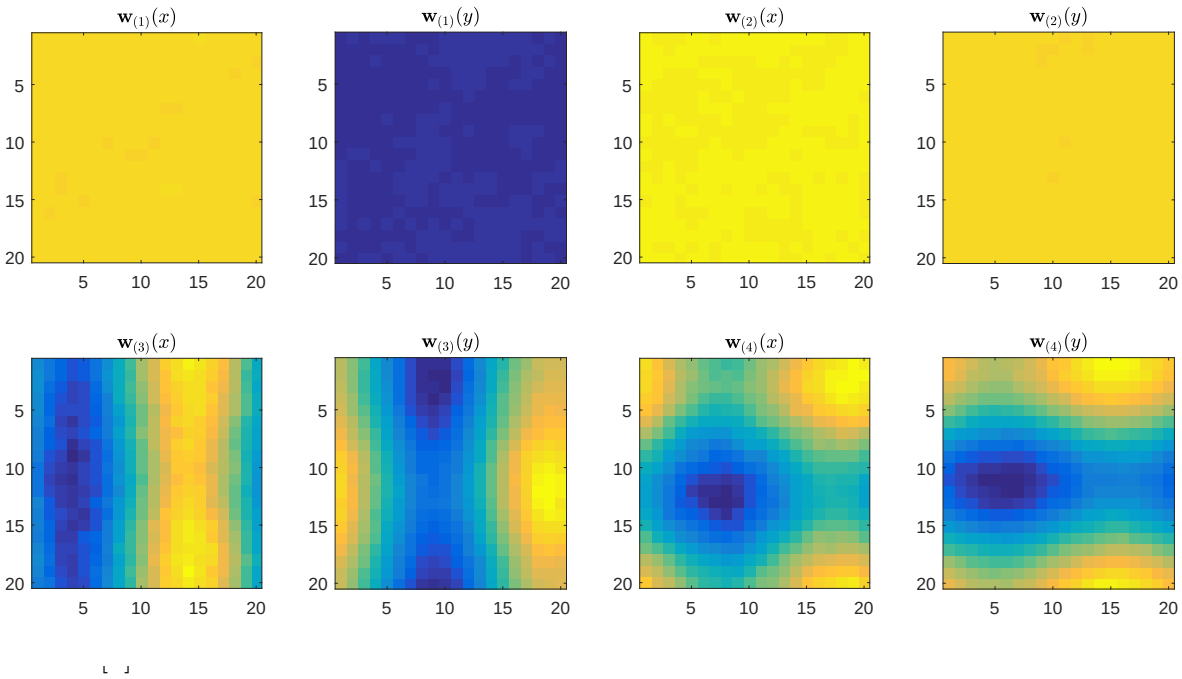


FIGURE 4.17: Weights of the PCA vs lattice site for the first four principal components of the XY model. Since each site is represented by it's x and y components, the PCA also detects correlations between them separately. To better visualize the results here the x and y components were also separated.

Chapter 5

Conclusion

In this section we summarize our results and give an outlook as to what can be done in the future.

In the broader sense of things, the main result from this Thesis is that PCA is able to successfully flesh out important physical properties of the probed models. In the case of the Ising model, we learned that the PCA is able to detect the order parameter of the system. It does so through the detection of the zeroth Fourier mode and encasing it in the main principal component, but analysis to subsequent components reveals that they correspond to higher energy Fourier modes. We also found that the PCA successfully finds the Fourier modes of the system for the Ising model in a Triangular Lattice and in the XY model. Existing literature [21, 41, 42] fails to put proper emphasis on this significant finding.

It must be noted that this is not a trivial result. Performing Fourier transformation is not what PCA does in general. Of all the possible subspaces in which is possible to represent the data the PCA picks out k-space because it efficiently encodes many of the significant properties of the system.

This is not something new. We've known for more than a century that Fourier representations of a system possess vital information about the underlying physics of a given model, but in a way it reassures us in the sense that it's not only us that think in that way, machines think it too.

An important remark, is that PCA is able to do all this with enough precision, such that physical quantities like the critical temperature in the Ising model, or the magnetization in the XY model can be calculated with high degree of accuracy.

Another aspect worth pointing out is that the PCA is somehow sensitive to the physics of the model. The reason behind this statement is related with the results for the Kagome Antiferromagnet. Even though the *matrix* analysis that was done for the Triangular Antiferromagnet should also fit the Kagome case and cause two dominant main components, that is not what is observed. Instead the PCA somehow detects the high degree of frustration in the Kagome model. This is manifest in the fact that all components are very low, with none particularly dominating over the other.

Regarding the XY model, it should be said that we believe PCA is not able to detect properties related with the the vorticity of the model, which are fundamental in the description of the phase transition present in this model. Reference [41] outlines the difficulty that PCA has in detecting non-linear features of the data. This being said, we recently received a suggestion from the co-supervisor of this Thesis, Gerard Barkema, that might be worth pursuing. If instead of the X and Y components of each spin, the phase is fed to the PCA, and if we look to components beyond the fourth, we might eventually find some components related with vorticity. The reasoning behind

this supposition is that the energy associated with vortices is lower than the energy associated with the lowest Fourier modes. PCA might detect some modes before it detects vorticity, but eventually it might detect it! Unfortunately we found ourselves in a position where the time to make such an analysis is lacking.

Another future research prospect is to explore non-linear generalizations of PCA. Namely methods like Principal Component Curves and Manifolds [19] where by constructing an embedding manifold and by encoding using standard geometric projection onto the manifold, the natural geometric interpretation of the PCA is extended to account for non-linear features. Or a more common approach, the kernel PCA [20], where using a kernel technique, the originally linear operations of PCA are performed in a reproducing kernel Hilbert space. Reference [41] states that they intend to release a second part to the paper where they use just this method, using a Neural-Network to choose the proper kernel.

There is certainly a good outlook in what pertains machine learning techniques used in the context of physics. PCA is just one of many methods (and perhaps one of the simplest) to which we can resort to learn new features of a given system. The prospect of a world where physicists are dethroned by machines seems somewhat ridiculous now, but with the advent of Artificial Intelligence who knows where the limit lies.

Appendix A

Confusion Learning Scheme

In this appendix we show the results we got for the Confusion Learning Scheme. Only the models presenting phase transitions were tested (Ising and the XY), since the goal here was to understand if the this scheme can recognize the phase transition and the critical temperature.

A.1 Ising model

The analogue critical temperature for the Ising model can be found through eq. 4.1. For $L = 60$ this value is

$$T_c(L) \approx 2.36$$

this is the value we're looking to estimate. Bellow we see the obtained result.

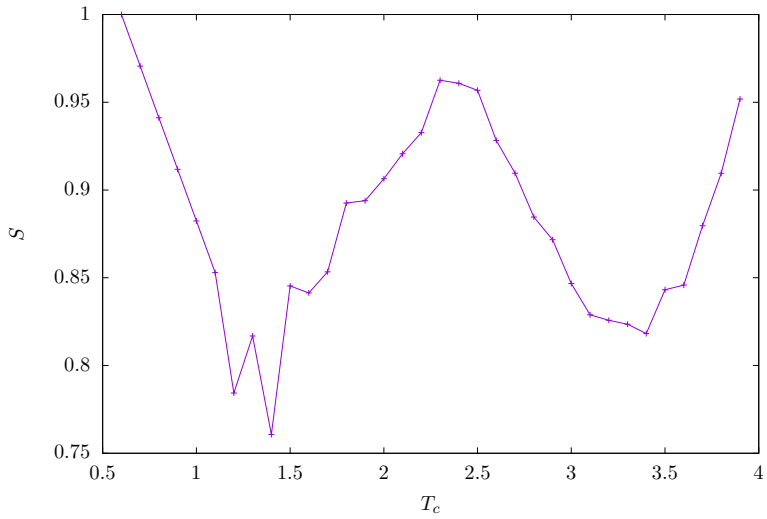


FIGURE A.1: Results for the Ising model with $L = 60$ using a network with initial learning rate $\alpha = 0.02$ and l2 regularization parameter $l_2 = 5 \times 10^{-3}$

We should point out that to get this results, an optimization of the values of the l2 regularization parameter and the initial learning rate α was necessary. A grid search looking for the values corresponding to the highest peak of the middle of the w shape was done. The amount of hidden units for the network was chosen to be equal to the amount of sites in the lattice.

The critical temperature obtained through this analysis is

$$T_c = 2.30(5)$$

the error here being is taken to be the temperature step ΔT. The Confusion Scheme is able to correctly estimate the value of the critical temperature, unfortunately it is hard to get any insight from this result. The reason why will be explained in the next subsection.

A.2 XY model

The analogue critical temperature of the XY model can be calculated through the following formula:

$$T_c(L) = T_{KT} + \frac{\pi^2}{4cln^2L}$$

with $c \approx 1.0926$ [**?**]. For $L = 20$, this value is

$$T_c(20) \approx 1.14$$

below we see the results obtained.

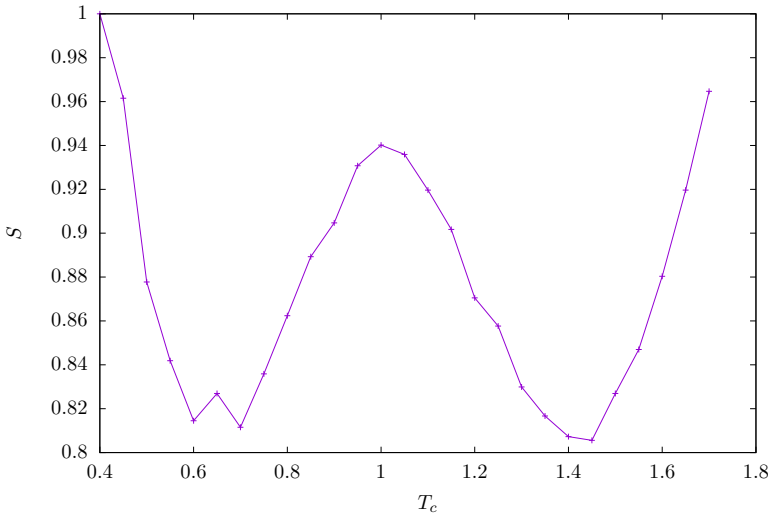


FIGURE A.2: Results for the xy model with $L = 20$ using a network with learning rate $\alpha = 0.07$ and l2 regularization parameter $l_2 = 5 \times 10^{-4}$

The peak in Fig **??** is our estimate for the the analogue critical temperature. It seems that the confusion Scheme predicts a critical temperature of

$$T_c(20) = 1.00(5)$$

the error here being is taken to be the temperature step ΔT. While close, this value is not quite correct, but it's obvious the network is able to find some feature related with the criticality of this model. Unfortunately the weights of the neurons can't tell us enough for us to conclude what this property is. As is often the case with Neural-Networks, these weights work has a black box. The properties are detected through the

combined work of the neurons so looking at their individual weights is not helpful in this case.

Bibliography

[1] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[2] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[3] JF Allen and AD Misener. Flow of liquid helium ii. *Nature*, 141(3558):75, 1938.

[4] Louis-François Arsenault, Alejandro Lopez-Bezanilla, O Anatole von Lilienfeld, and Andrew J Millis. Machine learning for many-body physics: the case of the anderson impurity model. *Physical Review B*, 90(15):155136, 2014.

[5] Nicholas M Ball and Robert J Brunner. Data mining and machine learning in astronomy. *International Journal of Modern Physics D*, 19(07):1049–1106, 2010.

[6] Marian Stewart Bartlett, Gwen Littlewort, Mark Frank, Claudia Lainscsek, Ian Fasel, and Javier Movellan. Recognizing facial expression: machine learning and application to spontaneous behavior. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 568–573. IEEE, 2005.

[7] Matthew JS Beach, Anna Golubeva, and Roger G Melko. Machine learning vortices at the kosterlitz-thouless transition. *arXiv preprint arXiv:1710.09842*, 2017.

[8] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[9] Indranil Bose and Radha K Mahapatra. Business data mining—a machine learning perspective. *Information & management*, 39(3):211–225, 2001.

[10] Peter Broecker, Juan Carrasquilla, Roger G Melko, and Simon Trebst. Machine learning quantum phases of matter beyond the fermion sign problem. *arXiv preprint arXiv:1608.07848*, 2016.

[11] Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017.

[12] Juan Carrasquilla and Roger G Melko. Machine learning phases of matter. *Nature Physics*, 2017.

[13] Kelvin Ch'ng, Juan Carrasquilla, Roger G Melko, and Ehsan Khatami. Machine learning phases of strongly correlated fermions. *Physical Review X*, 7(3):031038, 2017.

[14] SG Chung. Essential finite-size effect in the two-dimensional xy model. *Physical Review B*, 60(16):11761, 1999.

[15] Stefano Curtarolo, Dane Morgan, Kristin Persson, John Rodgers, and Gerbrand Ceder. Predicting crystal structures with data mining of quantum calculations. *Physical review letters*, 91(13):135503, 2003.

[16] Andreas Engel. *Statistical mechanics of learning*. Cambridge University Press, 2001.

[17] Arthur E. Ferdinand and Michael E. Fisher. Bounded and inhomogeneous ising models. i. specific-heat anomaly of a finite lattice. *Phys. Rev.*, 185:832–846, Sep 1969.

[18] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in Neural Information Processing Systems*, pages 64–72, 2016.

[19] Alexander N Gorban, Balázs Kégl, Donald C Wunsch, Andrei Y Zinovyev, et al. *Principal manifolds for data visualization and dimension reduction*, volume 58. Springer, 2008.

[20] Heiko Hoffmann. Kernel pca for novelty detection. *Pattern Recognition*, 40(3):863–874, 2007.

[21] Wenjian Hu, Rajiv R. P. Singh, and Richard T. Scalettar. Discovering phases, phase transitions, and crossovers through unsupervised machine learning: A critical examination. *Phys. Rev. E*, 95:062122, Jun 2017.

[22] Ernst Ising. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik A Hadrons and Nuclei*, 31(1):253–258, 1925.

[23] Kenzi Kanô and Shigeo Naya. Antiferromagnetism. the kagomé ising net. *Progress of theoretical physics*, 10(2):158–172, 1953.

[24] P Kapitza. Viscosity of liquid helium below the l-point. *Nature*, 141(3558):74, 1938.

[25] Ross D King, Stephen Muggleton, Richard A Lewis, and MJ Sternberg. Drug design by machine learning: The use of inductive logic programming to model the structure-activity relationships of trimethoprim analogues binding to dihydrofolate reductase. *Proceedings of the national academy of sciences*, 89(23):11322–11326, 1992.

[26] Jean Claude Le Guillou and Jean Zinn-Justin. Accurate critical exponents from field theory. *Journal de Physique*, 50(12):1365–1370, 1989.

[27] Yen Lee Loh, Dao-Xin Yao, and Erica W Carlson. Thermodynamics of ising spins on the triangular kagome lattice: Exact analytical method and monte carlo simulations. *Physical Review B*, 77(13):134402, 2008.

[28] Brett A McKinney, David M Reif, Marylyn D Ritchie, and Jason H Moore. Machine learning for detecting gene-gene interactions. *Applied bioinformatics*, 5(2):77–88, 2006.

[29] Pankaj Mehta and David J Schwab. An exact mapping between the variational renormalization group and deep learning. *arXiv preprint arXiv:1410.3831*, 2014.

[30] R Moessner and SL Sondhi. Ising models of quantum frustration. *Physical Review B*, 63(22):224401, 2001.

[31] Gordon Frank Newell. Crystal statistics of a two-dimensional triangular ising lattice. *Physical Review*, 79(5):876, 1950.

[32] MEJ Newman and GT Barkema. *Monte Carlo Methods in Statistical Physics chapter 1-4*. Oxford University Press: New York, USA, 1999.

[33] Takao Ohta and David Jasnow. Xy model and the superfluid density in two dimensions. *Physical Review B*, 20(1):139, 1979.

[34] Peter Olsson. Monte carlo analysis of the two-dimensional xy model. ii. comparison with the kosterlitz renormalization-group equations. *Physical Review B*, 52(6):4526, 1995.

[35] Lars Onsager. Crystal statistics. i. a two-dimensional model with an order-disorder transition. *Physical Review*, 65(3-4):117, 1944.

[36] Nicholas P Schafer, Bobby L Kim, Weihua Zheng, and Peter G Wolynes. Learning to fold proteins using energy landscape theory. *Israel journal of chemistry*, 54(8-9):1311–1337, 2014.

[37] Frank Schindler, Nicolas Regnault, and Titus Neupert. Probing many-body localization with neural networks. *arXiv preprint arXiv:1704.01578*, 2017.

[38] John C Snyder, Matthias Rupp, Katja Hansen, Klaus-Robert Müller, and Kieron Burke. Finding density functionals with machine learning. *Physical review letters*, 108(25):253002, 2012.

[39] Giacomo Torlai and Roger G Melko. Learning thermodynamics with boltzmann machines. *Physical Review B*, 94(16):165134, 2016.

[40] Evert PL van Nieuwenburg, Ye-Hua Liu, and Sebastian D Huber. Learning phase transitions by confusion. *Nature Physics*, 13(5):435–439, 2017.

[41] Ce Wang and Hui Zhai. Machine learning of frustrated classical spin models. i. principal component analysis. *Phys. Rev. B*, 96:144432, Oct 2017.

[42] Lei Wang. Discovering phase transitions with unsupervised learning. *Physical Review B*, 94(19):195105, 2016.

[43] GH Wannier. Antiferromagnetism. the triangular ising net. *Physical Review*, 79(2):357, 1950.

[44] Hans Weber and Petter Minnhagen. Monte carlo determination of the critical temperature for the two-dimensional xy model. *Physical Review B*, 37(10):5986, 1988.

[45] Qianshi Wei, Roger G Melko, and Jeff ZY Chen. Identifying polymer states by machine learning. *Physical Review E*, 95(3):032504, 2017.

[46] Yi Zhang and Eun-Ah Kim. Quantum loop topography for machine learning. *Physical Review Letters*, 118(21):216401, 2017.