Universidade Federal do Rio Grande do Sul
Instituto de Física
Trabalho de Conclusão de Curso

Aprendizado Não Supervisionado em Sistemas de Spins

Pedro Henrique Mendes Duarte

Trabalho de Conclusão de curso apresentado como requisito parcial para obtenção do título de Bacharel em Física.

Orientador:
Prof. Dr. Heitor Carpes Marques Fernandes.

Porto Alegre, RS
Outubro de 2022

Mendes, Pedro Henrique
Aprendizado Não Supervisionado em Sistemas de Spins / Pedro Henrique Mendes Duarte. -- 2022.
46

Orientador: Dr. Heitor Carpes Marques Fernandes.

Trabalho (Conclusão de curso) - Universidade Federal do Rio Grande do Sul, Instituto de Física, Porto Alegre, BR-RS, 2022.

Monte Carlo, Modelo de Ising, Modelo de Potts, Aprendizado Não Supervisionado, Análise dos Componentes Principais, Clusterização. I. Fernandes, Heitor C.M., orient.

Agradecimentos

Este trabalho não é só meu, ele também é de todas as pessoas que, mesmo minimamente, fizeram parte da minha vida. Eu acredito que cada pessoa que conhecemos deixa um pedaço dela em nós, assim quero agradecer aos tantos que me compõem.

Aos amigos que conheci na faculdade, João Oliveski, Julia Martins, Mateus Guimarães e Matheus Demarchi. Me viram trilhar esse caminho acadêmico desde o inicio. Quero levar nossa amizade para a vida toda.

Aos meus amigos mais antigos, João "Bolinho" Rosa, Bryan "Ed" Campello e Lucca Cunha, com quem sempre pude contar e me fazem ver a vida de um jeito melhor.

Ao professor Heitor Fernandes que me acompanhou por grande parte da minha graduação. Sempre esteve disposto a me ensinar, orientar e dar todo suporte necessário.

Aos meus irmãos, Arthur, Maria Eduarda e Marina e Ramiro. Que sirva de lembrança que a educação é a ferramenta mais transformadora das nossas vidas.

Aos meus padrinhos, Camila e Fábio, que sempre estiveram presentes e nunca deixaram faltar nada. Nos meus momentos mais difíceis foram vocês que eu pude chamar de pais.

A minha namorada Thauany, o maior presente que a UFRGS me deu. Que sempre acreditou em mim, até quando nem mesmo eu acreditei. Me da força e me alegra todos os dias. Te amo.

A minha vó, Maria da Glória. Eu poderia escrever mil páginas e ainda sim não seriam suficientes. Minha maior motivadora e a razão pela qual eu estudo. Todo o esforço é dedicado a ti, eu não seria quem sou sem ti e este trabalho não existiria sem ti. Continue me vendo de perto.

E a tantos outros que aqui não foram citados, mas que estão presentes em mim.

Obrigado.

「俺はルフィ！海賊王になる男だ！」
尾田 栄一郎

Resumo

Neste trabalho utilizamos algoritmos de aprendizado não supervisionado, uma classe de aprendizado de máquina, no estudo dos modelos de Ising e Potts. Iniciamos o estudo com o modelo de Ising, que possui uma vasta literatura, para familiarização dos conceitos envolvidos. Os resultados da simulação são condizentes com os encontrados na bibliografia e a aplicação das técnicas de aprendizado não supervisionado encontraram resultados similares aos obtidos por ferramentas já difundidas. As técnicas de aprendizado não supervisionado utilizados reduziram a dimensão dos dados, encontraram a região onde ocorre a transição de fase e identificaram comportamentos do sistema reduzido. As ferramentas também forneceram concepções sobre o parâmetro de ordem do modelo de Ising.

Na aplicação ao modelo de Potts também foram reduzidas as dimensões dos sistemas e identificadas as regiões críticas, que variam para cada q-estados. Foram definidos parâmetros de ordem no espaço dos dados reduzidos que possuem características dos já conhecidos. Uma análise relativa ao tamanho da rede foi realizada e os parâmetros definidos possuem semelhança aos definidos na literatura. Em ambos modelos foi possível identificar correlações que não seriam facilmente identificadas com a grande quantidade de dados originais.

Palavras-chave: Monte Carlo, Modelo de Ising, Modelo de Potts, Aprendizado Não Supervisionado, Análise dos Componentes Principais, Clusterização.

Abstract

In this work we use unsupervised learning algorithms, a class of machine learning, in the study of Ising and Potts models. We started with the study of the Ising model, which presents a vast literature, to familiarize with the concepts involved. The simulation results are consistent with those previously found in the literature and the application of unsupervised learning techniques found similar results to those obtained by common tools. The unsupervised learning techniques used reduced the dimensionality of the data, being able to locate the phase transition and identify behaviors in the reduced system. Those tools also provided insights into the order parameter of the Ising model.

In the application to the Potts model, the dimensions of the systems were also reduced and the critical regions were identified, which vary for each q-states. Order parameters were defined in the space of the reduced data that have characteristics with those we already known. An analysis related to the lattice size were performed and the parameters are similar to those defined in the literature. In both models it was possible to identify correlations that would not be easily identified with the large amount of original data.

Keywords: Monte Carlo, Ising Model, Potts Model, Unsupervised Learning, Principal Component Analysis, Clusterization.

Sumário

Capítulo 1

Introdução

Com o objetivo geral de reconhecer padrões nos dados disponíveis e tentar prever resultados não é surpresa que o aprendizado de máquina tenha chamado atenção da comunidade de físicos. A utilização destas ferramentas aplicadas em diversas áreas da física tem crescido nos últimos anos [1–8]. Já se sabe que o aprendizado de máquina pode ser eficiente na determinação, tanto do ponto onde a transição de fase ocorre [8], quanto do tipo de simetria que é quebrada [9]. Entretanto, apesar do sucesso destas abordagens, não é claro como os modelos e algoritmos reconhecem variações nos parâmetros de controle e nos observáveis de interesse.

O estudo de sistemas de spins se faz necessário na compreensão de fenômenos de diferentes áreas. Um sistema de spins é definido pela discretização do espaço que as partículas do sistema ocupam. Cada partícula ocupa um sítio e assume um valor. A capacidade de simplificar as características é importante quando lidamos com um sistema complexo. A simplificação das características permite tratar os sistemas de forma mais acessível sem perder a capacidade de compreender os fenômenos que os compõem. Usualmente definidos em uma rede, como os deste trabalho, buscam reproduzir comportamentos de sistemas físicos dentro de tamanhos de redes finitos. O exemplo mais conhecido é o Modelo de Ising [10], um modelo que busca reproduzir um imã, que possui grande literatura e conta com resultados exatos [11].

O Método de Monte Carlo (MC) é amplamente utilizado no estudo deste tipo de sistemas. A grande vantagem do método de MC está em ser capaz de simular os sistemas e estudar seus resultados mesmo quando uma solução analítica não está disponível. Nas últimas décadas foram desenvolvidas muitas ferramentas para a compreensão dos siste-

mas de spins, utilizando uma combinação dos (raros) resultados exatos conhecidos e de abordagens numéricas e computacionais. Nesta situação os resultados exatos são de fundamental importância, pois permitem a comparação destes com os obtidos por meio dos métodos de aproximação numéricos desenvolvidos. Com isto é possível avaliar a ordem da precisão que o método numérico propicia.

Reconhecimento de imagem, tradução de automática de linguagem e previsões do mercado financeiro são exemplos de aplicações de aprendizado de máquina. Os modelos geralmente são utilizados como caixas-pretas, onde apenas sabemos a entrada e a saída. Porém, nas questões relativas a aplicações na Física, há a necessidade de um entendimento mais profundo [12, 13] para que possamos introduzir princípios físicos na criação dos modelos.

Este trabalho tem o objetivo de estudar os sistemas de spins utilizando técnicas de Aprendizado Não Supervisionado, uma classe do aprendizado de máquina. Se propõe estudar dois modelos: Ising e Potts. Tais modelos apresentam transição de fase, definidas analiticamente, e possuem vasta literatura. O modelo de Ising pela sua simplicidade será utilizado em um primeiro momento para a familiarização, e então, avançaremos para o estudo do Potts. O modelo de Potts apresenta duas ordens de transição, contínuas e descontínuas, assim o método pode ser usado para o estudo de diferentes ordens. Estudar modelos que apresentam diferentes ordens de transição é importante quando estamos lidando com um sistema que não entendemos. Por serem modelos já estudados, esperamos encontrar resultados conhecidos utilizando técnicas que ainda não são amplamente utilizadas.

Capítulo 2

Metodologia

Neste capítulo serão apresentados os métodos utilizados na elaboração do trabalho. Também serão introduzidos conceitos necessários para o entendimento do trabalho.

2.1 Método de Monte Carlo

O Método de Monte Carlo (MC) é um método de simulação, realizada computacionalmente, que busca reproduzir os fenômenos físicos dos sistemas estudados [14]. O método se baseia em simular o sistema em contato com um banho térmico, ou no ensemble desejado, e fazer com que ele explore uma variedade de estados com os pesos estatísticos apropriados.

Na formulação de Monte Carlo via Cadeias de Markov, MCMC, precisamos escolher a dinâmica que o sistema adotará ao longo da simulação, isto é, a regra de aceitação para trocar de um estado para outro. Devemos escolher taxas de transição $A(\mu \rightarrow \nu)$ de um estado (μ) para outro estado (ν) que, no equilíbrio, representem a distribuição de Boltzmann [14]. Por meio dessas taxas é possível considerar os estados que o sistema acessará durante a simulação, e nestes estados podemos realizar as medidas de interesse.

A grande vantagem do método de MC está em não ser necessário incluir cada estado do sistema. Amostrando apenas uma pequena quantidade desses estados, escolhidos apropriadamente, obteremos resultados corretos. Ou seja, o método de MC se baseia em amostragem por importância. As medidas são tomadas como médias dos estados que o sistema passa. O método de MC, em muitos casos, não possui cálculos complexos, mas

cálculos simples que são repetidos diversas vezes.

As implementações das simulações de MC deste trabalho serão na rede quadrada com interações entre primeiros vizinhos e possuem condições de contorno periódicas. As redes possuem $N = L \times L = L^2$ sítios, cada um com um spin que pode assumir um determinado valor, o qual depende do modelo simulado. Os sistemas são inicializados aleatoriamente e o sistema evolui até o equilíbrio pelo tempo transiente, neste tempo não são tomadas medidas.

2.1.1 Algoritmo de Metropolis

Um dos algoritmos de evolução mais simples é o de Metropolis. Introduzido nos meados dos anos 50 por Metropolis e seus colegas [15], tem sido utilizado até hoje por sua simplicidade e eficiência. O algoritmo aplica de forma clara os conceitos das simulações de MC por cadeira de Markov (MCMC), como descrito na literatura [14, 16, 17].

Assim, queremos utilizar um algoritmo para alterar os estados do sistema que obedeça as condições de uma simulação MCMC. O algoritmo irá mudar aleatoriamente o estado de um único spin entre uma das opções disponíveis. Algoritmos que alteram o sistema modificando um spin por vez são denominados como algoritmos de *single spin flip*. Precisamos também considerar que o sistema se encontra em uma temperatura T e precisa obedecer a distribuição de Boltzmann. Logo, o algoritmo proposto possui a forma

$$A(\mu \to \nu) = \begin{cases} \exp\left(-\beta \Delta E\right), & \text{se} \quad \Delta E > 0 \\ 1, & \text{se} \quad \Delta E < 0 \end{cases}, \tag{2.1}$$

onde $\beta = 1/k_B T$ e $\Delta E = E_\nu - E_\mu$. A equação (2.1.1) foi definida para a aplicação no modelo de Ising, podendo apresentar mudanças para outros modelos. O algoritmo da Equação (2.1) nos diz que a troca de estados depende apenas da diferença de energia entre eles, da temperatura do banho térmico e da constante de Boltzmann. Se a diferença de energia for menor que zero, a transição é sempre aceita. Se for maior que zero a transição é aceita com uma probabilidade relativa ao peso de Boltzmann, que leva em consideração a diferença de energia dos estados. O algoritmo satisfaz as relações de balanço detalhado e ergodicidade. As condições de balanço e ergodicidade não discutidos neste trabalho, mas seguem detalhados na Ref. [14].

O tempo de simulação é medido em MCS (*Monte Carlo Steps*), passos de Monte Carlo, onde o algoritmo é repetido L^2 vezes. A repetição do algoritmo de aceitação L^2 vezes permite que cada spin, em média, tenha a chance de tentar trocar seu valor a cada passo.

2.1.2 Algoritmo Swendsen-Wang

Entretanto existem algoritmos de dinâmica mais eficientes, em certas regiões, que o de Metropolis para a simulação de sistemas de spins. Um exemplo é o algoritmo de Swendsen-Wang [14], um algoritmo de *cluster flip*, inversão de múltiplos spins ao mesmo tempo. A utilização do algoritmo de Swendsen-Wang passa pelo entendimento da formação de *clusters*, que em seguida serão propostos a uma inversão de valores. A formação segue a forma: Escolhemos um sítio aleatório e olhamos seus vizinhos, se os vizinhos possuírem o mesmo valor do sítio inicial adicionaremos ele ao *cluster* com uma probabilidade

$$P_{add} = 1 - \exp\left(-\beta J\right), \tag{2.2}$$

onde $\beta = 1/k_B T$. A equação (2.2) foi definida para a aplicação no modelo de Potts, podendo apresentar mudanças para outros modelos. Então invertemos o valor dos spins do *cluster* com uma probabilidade de 1/2. É interessante apontar que a identificação de todos *clusters* acontece antes das propostas de inversão de valores. Caso estejamos interessados em aplicar esse algoritmo para casos onde possuímos múltiplas opções de spins, montamos o *cluster* e então propomos a troca de valor do spin com probabilidade $1/q$, onde q é o número de possíveis estados para os spins.

O tempo de simulação é medido em MCS (*Monte Carlo Steps*), passos de Monte Carlo, onde o algoritmo é repetido apenas uma vez. Diferentemente do algoritmo de Metropolis, não repetiremos o algoritmo L^2 pois os spins do *cluster* trocarão para algum valor, nem que seja o mesmo valor que ele já se encontra.

2.2 Aprendizado Não Supervisionado

Arthur Samuel define Aprendizado de Máquina, como o "campo de estudo que possibilita aos computadores a habilidade de aprenderem sem serem explicitamente programados" [18]. Aprendizado não supervisionado é uma classe de aprendizado de máquina onde obtemos resultados ao utilizar dados que não possuem rótulos. Existem diversos algoritmos e modelos de aprendizado não supervisionado, entre estes métodos, destacamos a redução de dimensionalidade e clusterização. A redução de dimensionalidade, útil quando tratamos com dados que possuem muitas dimensões, tornando mais fácil a visualização ou compreensão. A redução de dimensionalidade se baseia no fato que muitos dados podem carregar qualidades de baixa relevância, podendo ser eliminados sem perder muita informação [19]. Já a clusterização é utilizada quando queremos agrupar os dados em grupos com características parecidas. A clusterização é uma tarefa que pode ser realizada por meio de diversos algoritmos, e a aplicação deles depende do formato dos nossos dados.

Quando tratamos de questões relativas a aplicações na física, não estamos interessados puramente na classificação [12, 13], queremos entender os procedimentos envolvidos para poder introduzir princípios físicos nos modelos propostos. Podemos citar a Máquina Restrita de Boltzmann (RBM), que se baseia em modelos de energia, e a distribuição de probabilidade, segue a distribuição de Boltzmann e possui utilização no estudo de sistemas de spins [20].

Neste trabalho utilizaremos algoritmos de aprendizado não supervisionado para obter informações dos sistemas de spins. Tomaremos algoritmos dessa classe pois queremos avaliar os sistemas sem assumir a existência da transição de fase. Os dados das configurações finais dos sistemas serão usados como entrada, nenhuma outra informação será dada ao modelo. Nos últimos anos tem se consolidado cada vez mais a utilização de técnicas de aprendizado de máquina no estudo de Física Estatística [4, 6–9] e em outras áreas da física [2, 5, 21].

É importante citar que não serão criados códigos de implementação dos algoritmos, mas aplicaremos os já disponíveis nas bibliotecas de aprendizado de máquina já conhecidas (Refs. [22, 23]). Há diversas vantagens ao utilizar essas bibliotecas, sendo a mais notável, a grande comunidade de usuários. Aplicar ferramentas adotadas pela comunidade científica

atual faz com que o projeto seja de fácil debate.

2.2.1 Análise dos Componentes Principais

A Análise dos Componentes Principais (PCA) é uma técnica de redução de dimensionalidade que identifica os componentes principais (PC) que contém maior variância e, então, projeta os dados nestes componentes [13]. Os componentes principais, encontrados através de uma transformação linear, são ortogonais entre si e representam menos variância conforme a ordem do componente aumenta. A transformação obtém um vetor $W = (w_1, w_2, \cdots, w_N)$, que são os pesos dos componentes principais, obtido da forma

$$X^T X w_l = \lambda_l w_l \tag{2.3}$$

onde X é a matriz original, os dados, w_l são os pesos e λ_l são os autovalores, neste caso são os componentes principais. Um fator importante ao utilizarmos o PCA é a razão de variância explicada, descrita por

$$\tilde{\lambda_l} = \frac{\lambda_l}{\sum_i^N \lambda_i} \tag{2.4}$$

A equação (2.4) nos diz quanto que cada componente principal representa de variação do sistema. Podemos utilizar os valores para decidir em quantos componentes principais podemos projetar nossos dados. Utilizamos os pesos obtidos na equação (2.3) para fazer a projeção dos dados. A projeção é descrita por

$$y_l = X w_l \tag{2.5}$$

Assim, obteremos o vetor Y, que é um novo conjunto de dados que possui dimensão igual ao número de componentes principais que julgamos como sendo importante.

A Figura 2.1 ilustra um exemplo da identificação das direções da variação dos dados e das projeções nas direções. O tamanho do vetor indica a maior variação dos dados e, por consequência, a ordem do componente principal (PC). Note que nas projeções, na Figura 2.1, os vetores possuem magnitudes iguais, mas o primeiro componente representa uma porcentagem de variação maior.

O exemplo da Figura 2.1 apresenta apenas duas características, podendo ser descrito em duas coordenadas. O PCA apenas confirma as variações que já são visíveis

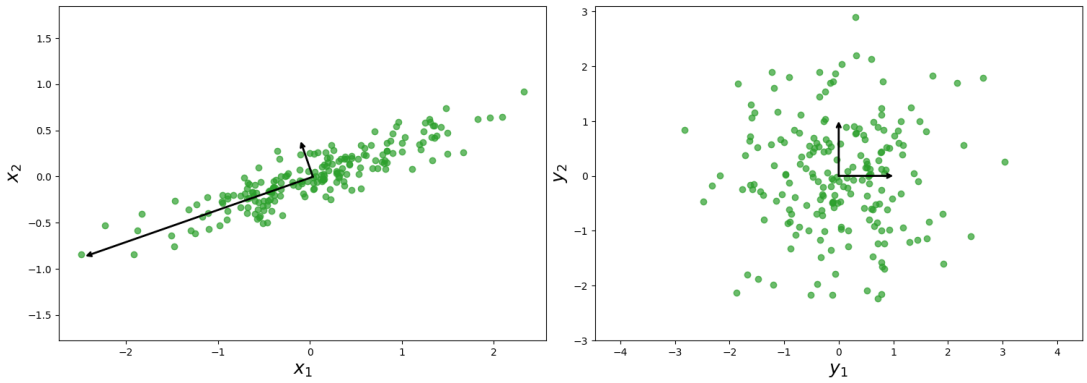

FIGURA 2.1. Exemplo de aplicação do PCA. Na esquerda podemos ver os dados em sua forma original e os componentes principais com suas magnitudes preservadas. À direita vemos as projeções dos dados nos componentes (componentes já ortonormais).

inicialmente. A grande vantagem estará em aplicar esse algoritmo em dados com muitas colunas de características, onde seria praticamente impossível notar variações, ou até mesmo visualizar graficamente os dados.

2.2.2 Algoritmo DBSCAN

O algoritmo DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) [24], é um algoritmo para identificar *clusters* na distribuição espacial dos nossos dados, baseado na densidade de pontos. O algoritmo identifica as áreas com maior densidade de pontos e os agrupa em *clusters*. Utilizaremos esse algoritmo porque ele leva em consideração os pontos de ruído, que aparecerão em nossas simulações de MC. Outro fator importante é que este algoritmo não necessita que seja indicado o número de *clusters* a serem identificados. São necessários dois hiperparâmetros, argumentos que controlam o modelo, para o DBSCAN. Os hiperparâmetros necessários são N_{min}, o número mínimo de pontos para que um *cluster* seja considerado um *cluster*, e ε, a distância máxima para pontos serem considerados vizinhos[1].

O algoritmo classifica os pontos em três tipos : (i) Pontos núcleos, que possuem N_{min} vizinhos dentro de uma distância ε. (ii) Pontos de fronteira, que são acessíveis através dos pontos núcleo, mas não possuem N_{min} vizinhos próximos a uma distância ε. (iii) Pontos de ruído, que não são núcleos e nem acessados por eles. A Figura 2.2

[1]Os nomes dos hiperparâmetros são comumente utilizados como **Min_samples** e **eps**, para N_{min} e ε respectivamente.

(retirada da Ref. [25]) mostra um exemplo da classificação dos pontos na hora de aplicar a clusterização.

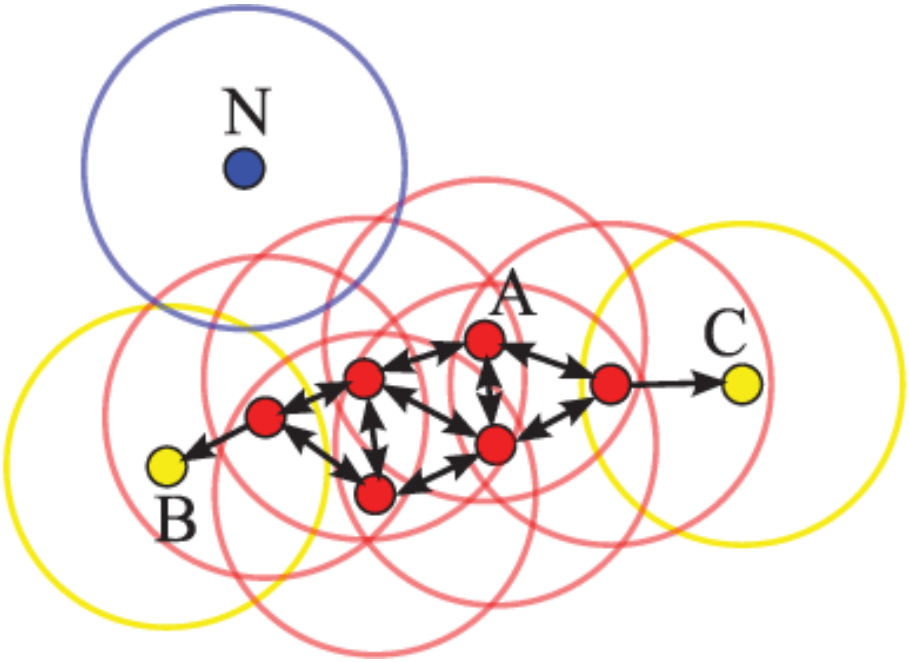

Figura 2.2. Exemplo de aplicação do algoritmo DBSCAN com $N_{min} = 4$. Neste diagrama o ponto A e os outros pontos vermelhos são pontos núcleo, pontos B e C, em amarelo, são pontos de fronteira, enquanto o ponto N, em azul, é um ponto de ruído. Figura retirada da Ref. [25].

Tendo em vista as definições dos pontos podemos aplicar o algoritmo. Então, escolhemos um ponto aleatório e determinamos seus vizinhos utilizando o ε. Se este ponto possuir o número mínimo, isto é, N_{min}, começa a formação do *cluster*. Os pontos desse *cluster* são visitados para avaliar se são pontos núcleo ou pontos de fronteira, seus possíveis vizinhos são adicionados ao *cluster*. Caso o ponto não possua o número mínimo de pontos vizinhos ele é denotado como ponto de ruído. Este ponto de ruído pode ser revisto futuramente e ser marcado como pertencendo a outro *cluster*. Então selecionamos outro ponto aleatório que não tenha sido visitado anteriormente e repetimos o processo. O algoritmo finaliza quando todos pontos forem visitados. Os hiperparâmetros N_{min} e ε podem ser definidos por conhecimento de domínio e/ou utilizar técnicas de otimização [26].

Neste trabalho partiremos de um N_{min} definido e aplicaremos técnicas para identificar o melhor valor para ε. O primeiro passo é calcular a distância para os n pontos mais próximos para cada ponto. Para calcular a distância dos pontos utilizaremos o algoritmo

de vizinhos próximos [27]. Ao visualizar essa distância, devemos tomar como ε o valor onde observamos uma mudança mais expressiva, isto é, onde temos a maior inclinação. Para obtermos maior precisão sera utilizado, na aferição do ponto com maior inclinação, o algoritmo desenvolvido pela Ref [28].

Capítulo 3

Aplicação ao Modelo de Ising 2D

Neste capítulo apresentaremos o Modelo de Ising, discutiremos os resultados da simulação por Monte Carlo e do aprendizado não supervisionado utilizando o PCA e a Clusterização.

3.1 O Modelo

O Modelo de Ising é um modelo estatístico que representa as propriedades magnéticas da matéria [10]. O modelo descreve uma rede de lado L com $N = L \times L = L^2$ sítios que interagem com sua vizinhança. Assumiremos que o campo magnético externo é nulo ($B = 0$) e que cada sítio interage com a mesma intensidade J. Os spins σ_i podem assumir os valores $+1$ (*spin up*) e -1 (*spin down*). O modelo que será utilizado é descrito pelo hamiltoniano de Ising

$$H = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j. \tag{3.1}$$

Onde a notação $\langle i, j \rangle$ significa que os sítios i e j são vizinhos próximos.

Existem dois observáveis de nosso interesse, e que serão utilizados neste trabalho, são: a magnetização e a energia. A energia é calculada diretamente do hamiltoniano dado em (3.1), enquanto a magnetização é definida pela soma dos spins

$$M = \sum_{i=1}^{N} \sigma_i. \tag{3.2}$$

Na fase ferromagnética a magnetização apresenta um comportamento com flutuações, isto é, o sistema alterna entre o máximo e o mínimo da magnetização. É importante

apontar que a intensidade das flutuações da magnetização depende do tamanho de rede. Para redes grandes a quantidade de tentativas de inversão de spin que são necessárias aumentam com a rede. Já na fase paramagnética o sistema está desordenado, e, por possuir um número similar de spins *up* e *down*, a magnetização estará próxima de zero.

O modelo apresenta transição de fase em $T = T_C \simeq 2{,}269$, para $J = k_B = 1$, onde k_B é a constante de Boltzmann. Para temperaturas superiores a temperatura critica ($T > T_C$) o modelo apresenta uma fase paramagnética ($M \approx 0$) e para temperaturas inferiores ($T < T_C$), apresenta uma fase ferromagnética ($M \neq 0$).

3.2 Resultados da Simulação

Para utilizar os dados da simulação como fonte do aprendizado não supervisionado é necessário primeiro avaliar os resultados obtidos. Foram realizadas simulações de Monte Carlo do modelo de Ising utilizando o algoritmo de Metropolis 2.1.1. Foram simulados três tamanhos de lado L: 20, 40 e 80 e condições de contorno periódicas. Tomamos estes tamanhos de rede pois sabemos que os resultados apresentam efeitos de tamanho finito e estamos interessados em analisar as diferenças. As temperaturas de simulação variam entre 1,6 e 2,9, com uma diferença de 0,01. Para cada temperatura foram realizadas 100 simulações, com diferentes sementes para o gerador de números aleatórios, assim, teremos 13100 simulações para cada tamanho de rede. Foram utilizados os seguintes parâmetros para todas simulações: tempo transiente e tempo de medida foram tomados como 10^5 MCS e 10^6 MCS, respectivamente; e $k_B = J = 1$. Os valores de tempo de transiente e medida seguem os valores tomados na literatura [14].

A Figura 3.1 mostra as séries temporais dos observáveis de relevância, energia e magnetização, em simulações com $T = 2{,}2$. A energia está bem centrada em um ponto específico, salvo por flutuações estatísticas. Já magnetização flutua entre seus dois extremos. Ambos resultados são esperados para as simulações, uma vez que elas foram realizadas em $T < T_C$. Podemos também visualizar as distribuições dos observáveis, obtidas das equações (3.1) e (3.2). A Figura 3.2 mostra histogramas dos observáveis após a simulação. A energia apresenta comportamento gaussiano, o que é compatível com o algoritmo utilizado para a dinâmica [16]. Porém a magnetização possui dois picos distintos,

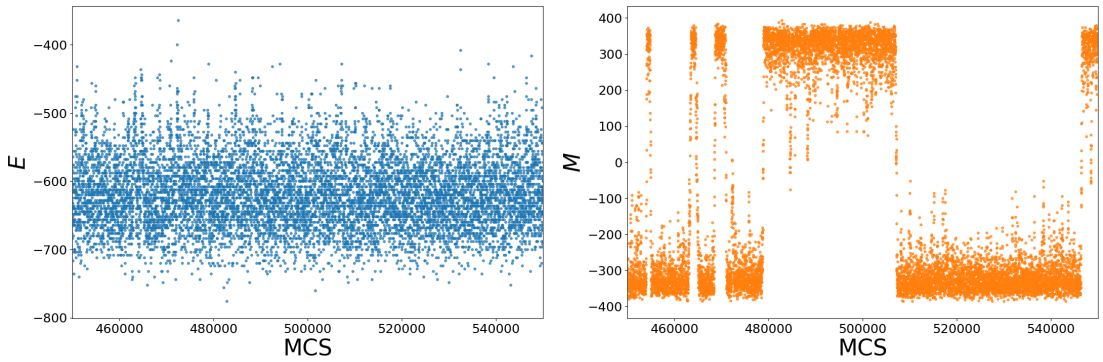

Figura 3.1. Trechos de séries temporais de simulações do Modelo de Ising. O gráfico da esquerda representa a energia e o da direita a magnetização. Os dados são da mesma simulação, onde $L = 20$ e $T = 2{,}2$.

comportamento que também é esperado dado o tamanho do sistema. A magnetização teria pontos simétricos caso o tempo de simulação tendesse ao infinito. A Figura 3.3

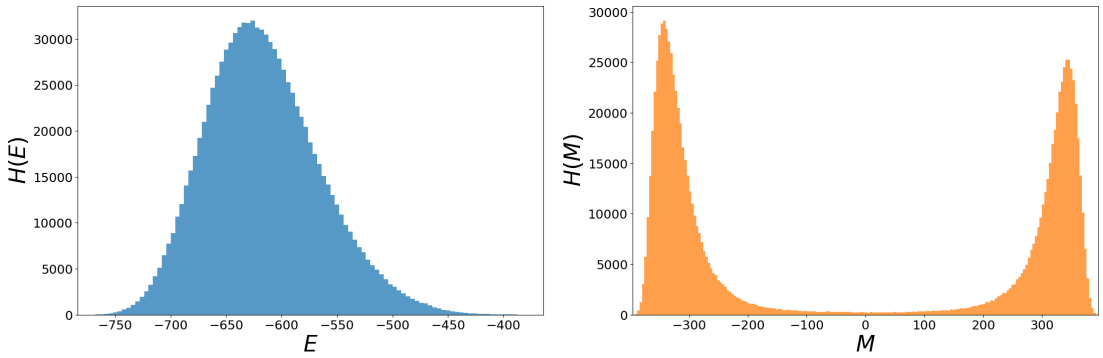

Figura 3.2. Histogramas da energia, à esquerda, e magnetização, à direita, das séries temporais da Figura 3.1, onde $L = 20$ e $T = 2{,}2$.

mostra a comparação das distribuições de energia e os resultados exatos [29] para dois tamanhos de rede $L = 32$ e $L = 64$.

Um dos objetivos é analisar o comportamento dos observáveis em função da temperatura. Na Figura 3.4 é possível ver o comportamento suave da energia, enquanto a magnetização possui dois focos de pontos. Para $T < T_C$ o sistema tende a orientar seus sítios para um valor não nulo de magnetização e seu valor oposto. Porém, para $T > T_C$ a magnetização vai a zero, também visível na Figura 3.4. Ambos valores de magnetização, o valor e o seu oposto, são equiprováveis, a orientação, positivo ou negativo, dependerá do número aleatório dentro da simulação. Cada ponto da figura representa uma média do observável sobre todo tempo de medida. Fazemos assim pois ao tomar a média durante

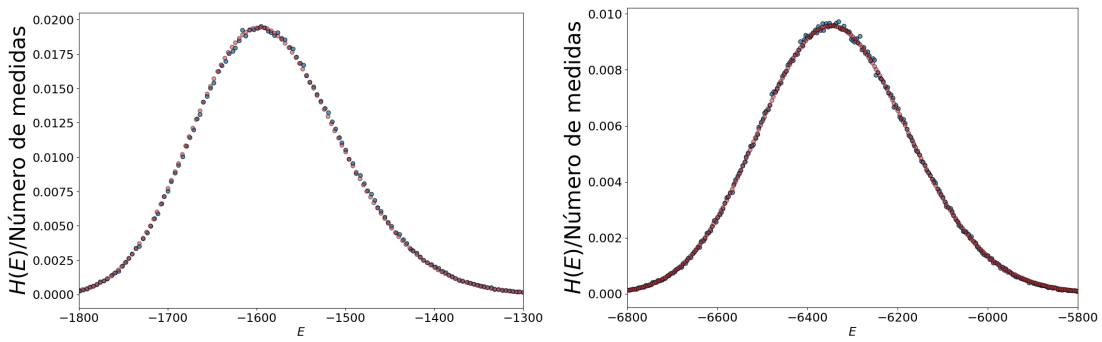

FIGURA 3.3. Resultados exatos e de simulação para dois tamanhos de rede, $L = 32$, à esquerda, e $L = 64$, à direita, ambos com temperatura $T = 2{,}2$. Em ambos gráficos os pontos azuis representam os histogramas de energia e os vermelhos os resultados exatos.

o tempo de simulação, é esperado que ele represente bem a medida. Logo, cada ponto é uma média durante 10^6 MCS. Notamos que perto da transição de fase o sistema apresenta muitas flutuações. Esse resultado é esperado, uma vez que na região crítica o tempo de correlação se torna maior e o algoritmo de Metropolis apresenta menor eficiência [14]. Ainda que possam ser notadas as flutuações, o comportamento está claro e condiz com o encontrado na literatura. Podemos ver que nossa simulação representa os dados exa-

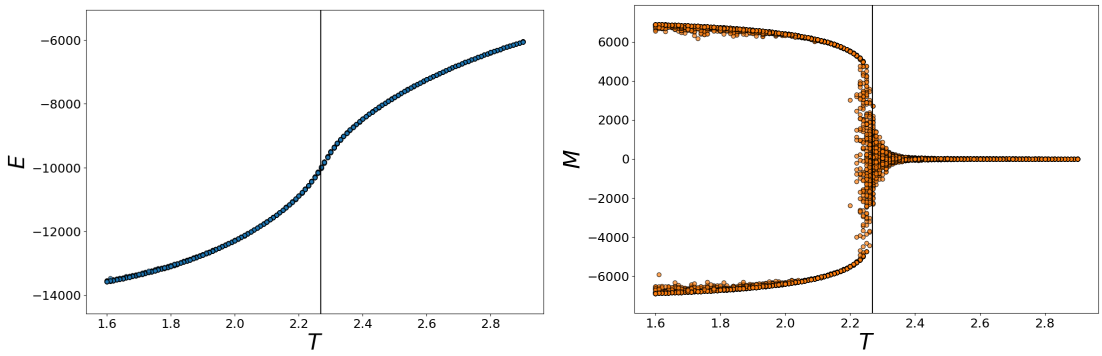

FIGURA 3.4. Magnetização, na esquerda, e energia, na direita, em função da temperatura T de simulação, $L = 80$. A linha vertical preta representa a temperatura crítica $T = T_C \simeq 2{,}269$.

tos e os comportamentos conhecidos do modelo de Ising encontrados na literatura [14]. Sabendo que nossas simulações estão corretas, podemos utilizar as configurações finais destas para o aprendizado não supervisionado.

Os dados que estamos interessados são de configurações finais dos sistemas pois com esses dados montaremos a matriz X descrita na equação (2.3). A matriz X tem a

forma $M \times N$, onde M é o número de amostras e N é o número de spins, isto é L^2. A matriz é composta pelo estado final de cada spin e ordenada da seguinte forma

$$X = \begin{pmatrix} \uparrow & \uparrow & \uparrow & \cdots & \downarrow & \uparrow & \downarrow \\ & & & \vdots & & & \\ \downarrow & \uparrow & \uparrow & \cdots & \uparrow & \uparrow & \downarrow \end{pmatrix}_{M \times N} . \tag{3.3}$$

Assim, na equação (3.3) cada linha é uma simulação completa.

3.3 Resultados do PCA

A Figura 3.5 mostra *snapshots* de configurações finais, com diferentes T, para sistemas com $L = 80$. Vemos, na 3.5, três exemplos de sistemas, um com $T < T_C$, um com $T \approx T_C$ e outro com $T > T_C$. Na configuração $T < T_C$ é observado a predominância de um spin, quando temos $T \approx T_C$ notamos a formação de grandes *clusters* e quando $T > T_C$ o sistema está desordenado. Analisando apenas três configurações é visível a diferença, porém estamos interessados em analisar uma grande quantidade de configurações com diferentes temperaturas.

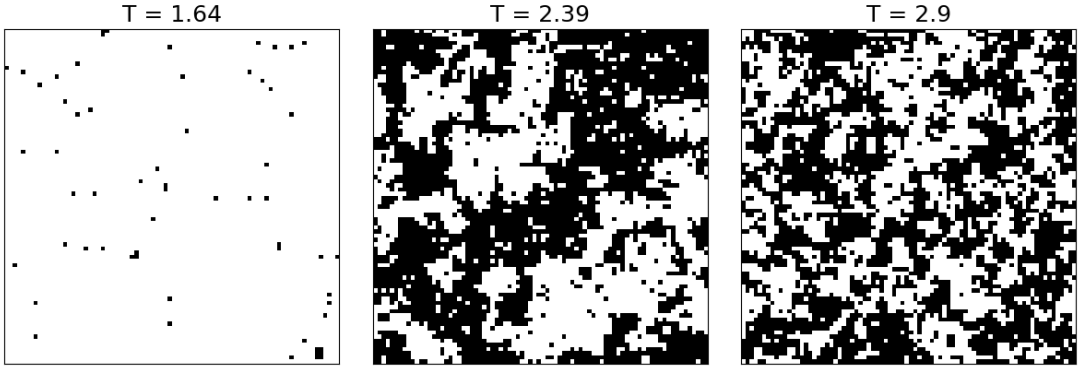

Figura 3.5. *Snapshots* de configurações finais de diferentes simulações para diferentes temperaturas. Simulações com $L = 80$. É possível ver a grande diferença de comportamento entre os sistemas.

O primeiro passo ao aplicar um método de redução de dimensionalidade é entender em quantas características é possível representar a variação dos nossos dados. A Figura 3.6 mostra as primeiras razões de variância explicada, da equação (2.4), para os três tamanhos de rede utilizados. É visível que o método identifica apenas um componente principal dominante. Isso significa que podemos expressar parte da variação contida no

sistema utilizando apenas um valor. Entretanto, perderíamos a dispersão dos dados caso apresentássemos em uma dimensão. Assim, vamos utilizar os dois primeiros componentes principais para visualizar nossos dados.

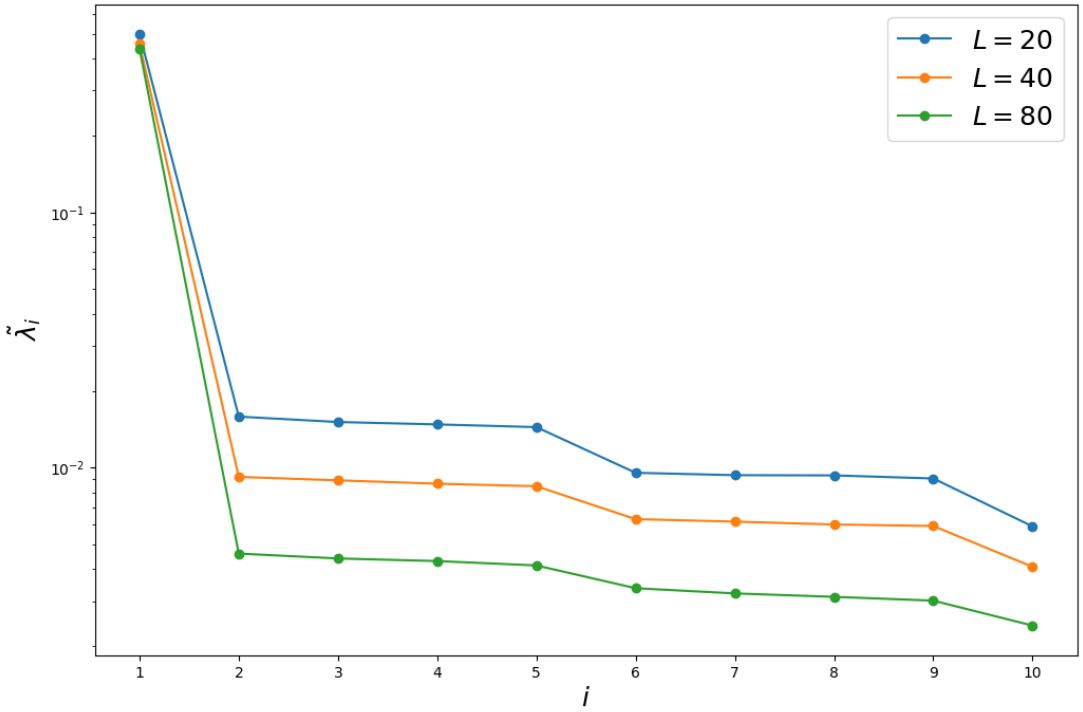

FIGURA 3.6. Razão da variação explicada para diferentes tamanhos de rede. O eixo $\tilde{\lambda}_i$ está na escala logarítmica para melhor visualização.

Precisamos projetar nossos dados nos componentes principais para analisar o comportamento, a projeção segue a equação (2.5). A Figura 3.7 nos mostra que, ao projetar nossos dados nos seus respectivos componentes principais, eles apresentam uma tendência a se agrupar com pontos relativos à sua temperatura de simulação. Pontos de maior temperatura se agrupam em $y_1 \approx 0$ e os de menor temperatura em $|y_1| > 0$. Também é visível que a dispersão em y_1 é proporcional ao tamanho da rede.

O próximo passo é atribuir significado físico a este componente principal identificado. Podemos observar o primeiro componente principal, y_1, em função da temperatura de simulação. Analisando a gráfico da Figura 3.8 é possível ver que o primeiro componente principal representa a magnetização do sistema. Isto significa que o algoritmo PCA identifica que a característica mais relevante do modelo de Ising, tendo como fonte os *snapshots* do sistema, é a magnetização.

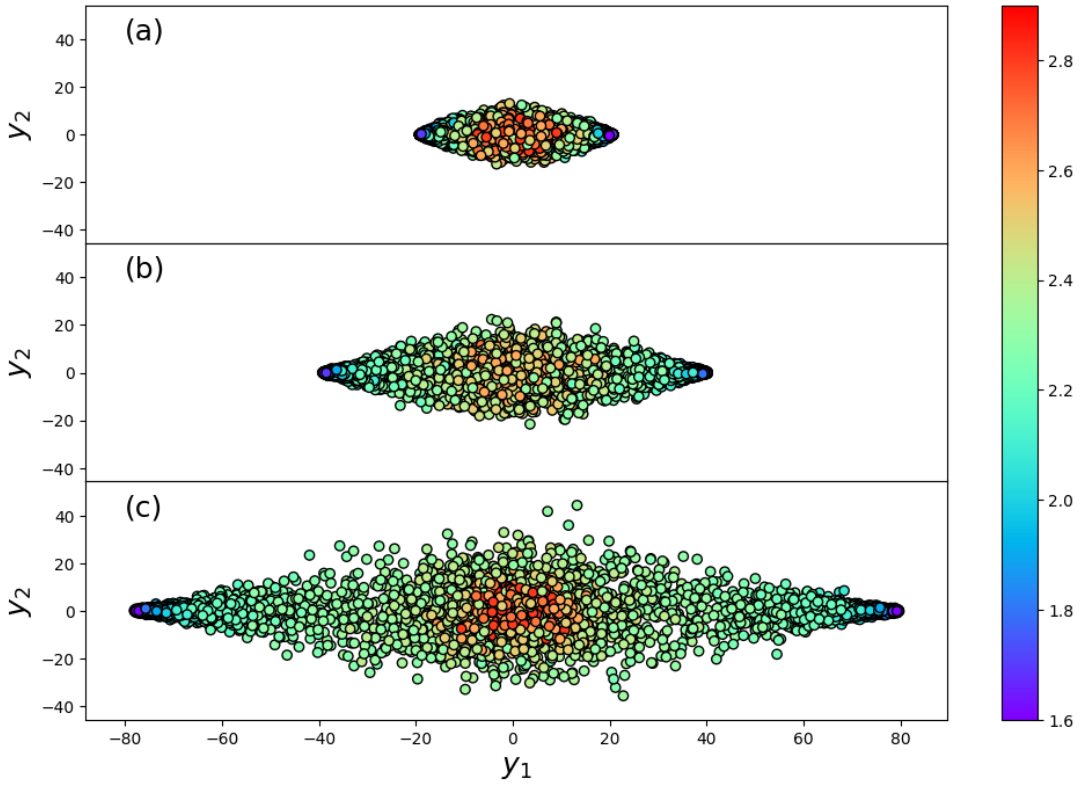

Figura 3.7. Projeções nos componentes principais para cada rede de lado L. Ao lado a escala de cores representando a temperatura de simulação T. Os painéis, (a), (b) e (c), representam as projeções de $L = 20$, $L = 40$ e $L = 80$, respectivamente.

Utilizando apenas as configurações do sistema como fonte do estudo, sem fornecer nenhum alvo para o algoritmo, foi possível reduzir a ordem do sistema e encontrar um parâmetro de ordem para transição de fase. As simulações representam sistemas com $L \times L = L^2$ dimensões e seria necessário um gráfico com esse número de dimensões para representar, ou $L^2!/(L^2 - 2)!$ gráficos de duas dimensões. Porém, independentemente do tamanho de rede, foi possível reduzir o sistema para duas dimensões, tornando a visualização das características do sistema mais evidentes.

3.4 Clusterização

Iremos utilizar o algoritmo DBSCAN para identificar *clusters* nas projeções dos componentes principais da rede de lado $L = 80$. Utilizaremos as projeções do maior tamanho de rede pois ele possui maior dispersão, mas o método pode ser aplicado a qualquer tamanho de rede. Uma vez que possuímos um novo conjunto de dados, com

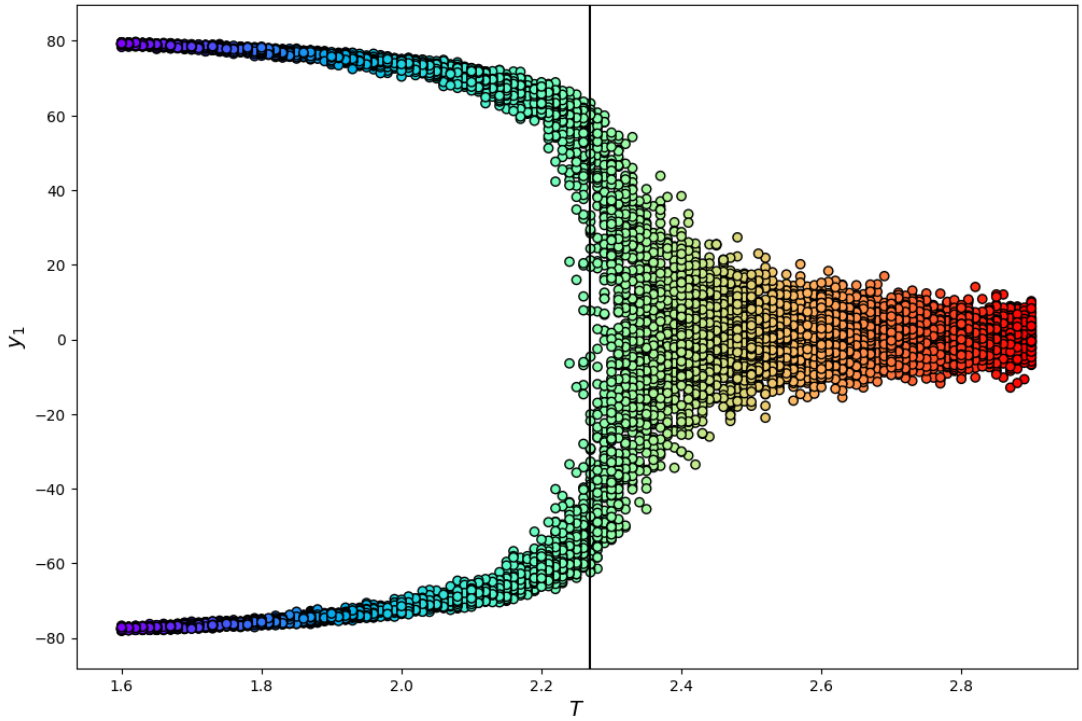

FIGURA 3.8. Primeiro componente principal, y_1 em função da temperatura T de simulação ($L = 80$). O comportamento similar a magnetização, salvo de dispersões, é visível.

duas dimensões, podemos realizar a clusterização e então aplicar os resultados no conjunto original dos dados. Com a aplicação do algoritmo de clusterização no modelo de Ising espera-se familiarizar com o método para a utilização em outros modelos.

Como descrito na seção 2.2.2, para utilizar o algoritmo são necessários dois hiperparâmetros: N_{min}, o número mínimo de pontos para um *cluster* ser considerado, e ε, distância máxima para dois pontos serem considerados vizinhos. N_{min} foi tomado como 25 por que estamos lidando com uma grande quantidade de dados. Para o valor de ε foi utilizado um algoritmo capaz de apontar o melhor valor para o hiperparâmetro (ver Refs. [26, 28]). A Figura 3.9 mostra a distância do vizinho mais próximo em função do número de vizinhos próximos. Precisamos escolher o ponto com maior curvatura [26], e este ponto foi tomado como $\varepsilon = 3{,}142$.

Com os hiperparâmetros escolhidos podemos aplicar o algoritmo DBSCAN. Iremos aplicar o algoritmo nas projeções, e mesmo identificado que o componente principal é a magnetização, não utilizaremos o módulo. Não considerar o módulo das projeções significa não levar em conta a simetria do modelo de Ising, mesmo sabendo que ela existe.

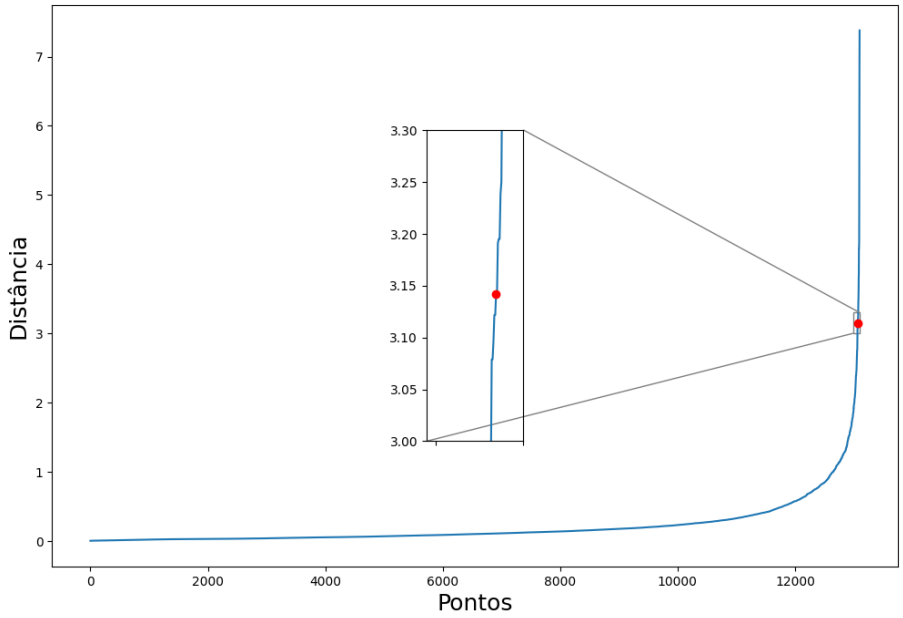

Figura 3.9. Distância em função dos n-pontos próximos próximos. Ampliado a região onde temos maiores curvaturas, com destaque para o ponto Distância $= \varepsilon = 3{,}142$.

Podemos ver na Figura 3.10 que o algoritmo identificou três *clusters* e 861 pontos de ruído. Esse número de pontos de ruído representa uma porcentagem de 7% no total de amostras. Possuímos três *clusters*, mas podemos notar que dois destes representam a região de baixa temperatura, logo os três *clusters* representam dois comportamentos do sistema. Como apontado na Figura 3.4, o nosso sistema apresenta dois comportamentos, podemos analisar a magnetização com a escala de cores referente aos resultados do resultado da clusterização.

A Figura 3.11 mostra que os *clusters*, quando representados na magnetização, representam os dois comportamentos do nosso sistema, antes e depois da temperatura crítica. Os pontos de ruído se encontram próximos a T_C, não pertencendo totalmente a nenhuma região. Logo a clusterização identificou ambas fases do sistema e a região de transição.

Partindo das projeções nos primeiros dois componentes principais para as simulações de $L = 80$ conseguimos identificar *clusters* com significado físico claro do nosso modelo. O *cluster* onde os pontos estão próximos de $|y_1| \approx 0$ representa os pontos onde $T > T_C$, os *clusters* onde $|y_1| \gg 0$ são relacionados a $T < T_C$, e os pontos de ruído representam a região $T \approx T_C$. Podemos ver que, sem nenhum alvo prévio o algoritmo conseguiu identificar que o sistema de entrada, projeções dos componentes, possui dois

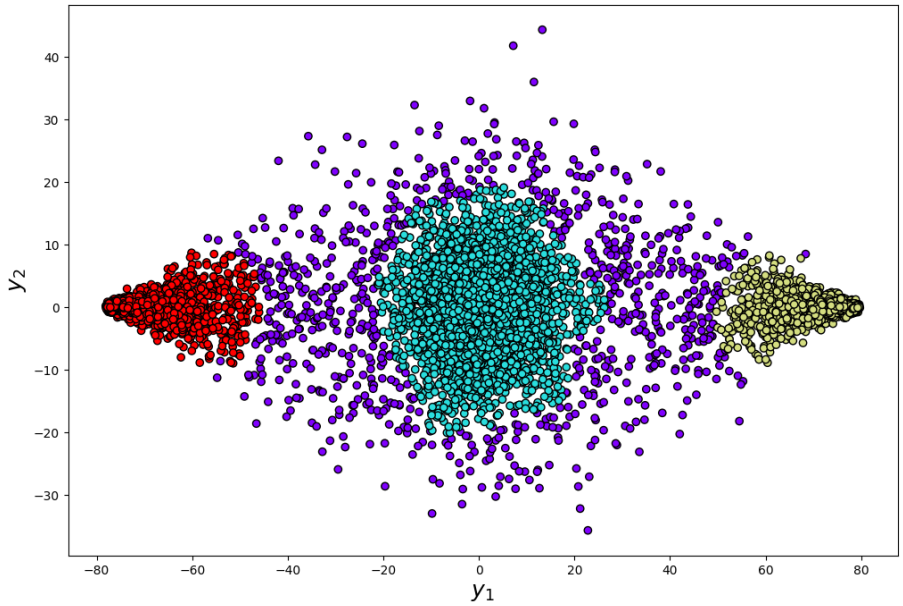

FIGURA 3.10. *Clusters*, em vermelho, amarelo e ciano, e ruído, em roxo, identificados pelo algoritmo DBSCAN. Nota-se que temos dois *clusters* nas regiões de baixa temperatura, um na região de alta temperatura e pontos de ruído na região entre eles.

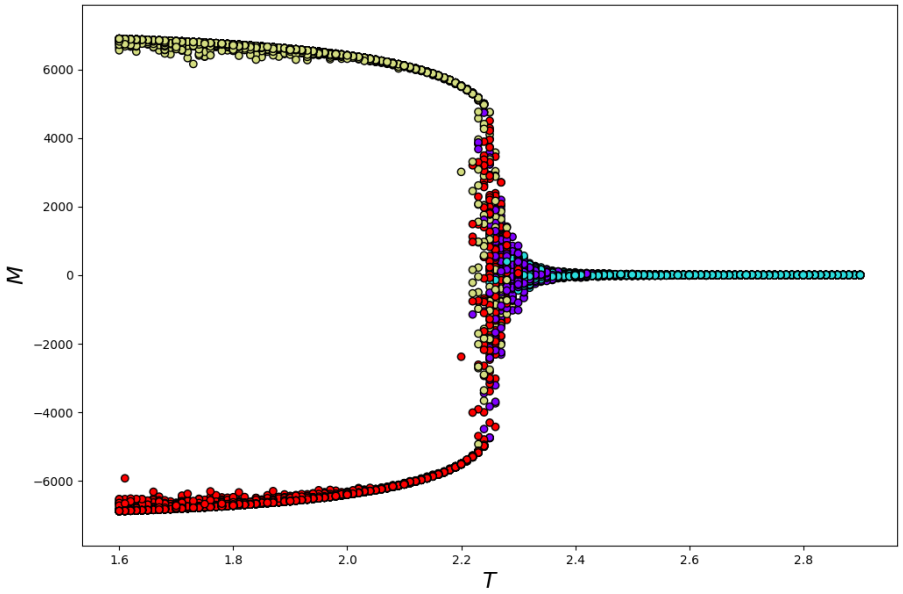

FIGURA 3.11. Magnetização em função da temperatura T ($L = 80$) de simulação, porém com escala de cores do resultado do DBSCAN.

comportamentos.

Capítulo 4

Aplicação ao Modelo de Potts 2D

Neste capítulo apresentaremos o Modelo de Potts, discutiremos os resultados da simulação por Monte Carlo e do aprendizado não supervisionado utilizando o PCA.

4.1 O Modelo

O Modelo de Potts é uma generalização do Modelo de Ising, onde os sítios s_i podem tomar valores diferentes de ± 1 [14]. Assim, o Modelo de Potts de q-Estados é aquele onde o sítio pode assumir os valores

$$s_i = 0, \cdots, q-1. \tag{4.1}$$

Podemos visualizar também como um sistema onde spins estão confinados em um plano, cada um apontado para direções igualmente espaçadas que são especificadas pelos ângulos [30]

$$\theta_i = \frac{2\pi s_i}{q}. \tag{4.2}$$

Neste modelo a interação depende se os ângulos dos spins são diferentes, isto é, se seus valores são diferentes. Caso sejam iguais, dois sítios vizinhos contribuem com uma energia $-J$; e não contribuem se eles possuem valores diferentes[1]. O modelo de Potts que será utilizado é descrito por

$$H = -J \sum_{\langle i,j \rangle} \delta_{\theta_i,\theta_j}, \tag{4.3}$$

[1]Apresentamos o modelo de Potts utilizando a notação de Clock Model, (4.2), onde os spins são pensados como spins apontando para uma direção. Utilizaremos essa notação pois ela será mais útil na construção da matriz que forneceremos ao PCA.

onde $\delta_{\theta_i,\theta_j}$ é a delta de Kronecker, que é 1 quando $\theta_i = \theta_j$ e 0 quando $\theta_i \neq \theta_j$. O modelo de Potts é equivalente ao de Ising quando $q = 2$, a menos de uma constante. A equivalência dos modelos é importante pois podemos utilizar os resultados exatos do Ising [29], de tamanhos finitos, para avaliar os resultados das simulações do Potts.

A transição de fase do modelo de Potts depende do valor de q adotado, o ponto onde ocorre é dado por

$$T_C = \frac{J}{\ln(1+\sqrt{q})}, \tag{4.4}$$

com $k_B = J = 1$. Similar ao Ising, para temperaturas inferiores à temperatura crítica observamos um comportamento ferromagnético, e para temperaturas superiores a temperatura crítica observamos um comportamento paramagnético[2]. É importante apontar que o modelo de Potts apresenta transições contínuas (de segunda ordem) para $q \leq 4$ e transições descontínuas (de primeira ordem) para $q > 4$ [30].

4.2 Resultados da Simulação

Para utilizar os dados da simulação como fonte do aprendizado não supervisionado é necessário avaliar os resultados obtidos. Foram realizadas simulações de Monte Carlo do modelo de Potts utilizando o algoritmo de Swendsen-Wang 2.1.2. Os *clusters* foram construídos utilizando o algoritmo Hoshen–Kopelman. Mesmo que tenhamos definido o valor dos spins pela equação (4.2), é conveniente simular o modelo de Potts utilizando os valores dos spins como descrito na equação (4.1). Foram tomados três tamanhos de lado L: 20, 40 e 80. Utilizaremos estes tamanhos de rede pois sabemos que os resultados apresentam efeitos de tamanho finito e estamos interessados em analisar as diferenças. As temperaturas de simulação variaram entre $T_C - 0{,}5$ e $T_C + 0{,}5$, com uma variação de 0,01. Para cada temperatura foram realizadas 50 simulações, com diferentes sementes para o gerador de números aleatórios, assim, teremos 5000 simulações para cada tamanho de rede. Foram utilizados os seguintes parâmetros para todas simulações: tempo transiente como 10^6 MCS e $k_B = J = 1$. O tempo de medidas só será utilizado para a avaliação dos resultados do modelo com os resultados exatos [29]. É importante citar que o modelo de Potts possui um tempo de execução maior que o modelo de Ising, isto porque ele é

[2] O ferromagnetismo aqui é definido como o estado em que a média dos valores dos spins é diferente de zero, enquanto no paramagnetismo a média equivale a $q/2$.

modelo mais complexo e possui algoritmos de dinâmicas que necessitam de um tempo maior para realizar suas execuções. Por necessitar um tempo maior para sua execução, foram realizados um número menor de amostras para esse modelo.

A Figura 4.1 mostra um trecho de uma série temporal da energia, em uma simulação com $q = 2$, $T = 1{,}1$ e $L = 32$ e $L = 64$. Assim como no modelo de Ising, a energia está bem centrada em um ponto específico, com algumas flutuações estatísticas. Podemos também visualizar as distribuições da simulação a fim de comparar com os resultados exatos [29]. Para fazer a comparação, precisamos utilizar $q = 2$, multiplicar os valores de energia por dois, para que a diferença de energia passe de 4 para 8, e adicionar $2L^2$, para que o estado fundamental do Potts seja igual ao do Ising. Também é necessário utilizar metade da temperatura na simulação do Potts para comparar com o resultado exato do Ising. A Figura 4.1 mostra a comparação das distribuições de energia e os resultados exatos, para $L = 32$. Como esperado, a energia apresenta um comportamento gaussiano, compatível com o algoritmo da dinâmica 2.1.2. Uma vez que a simulação representa os dados exatos e

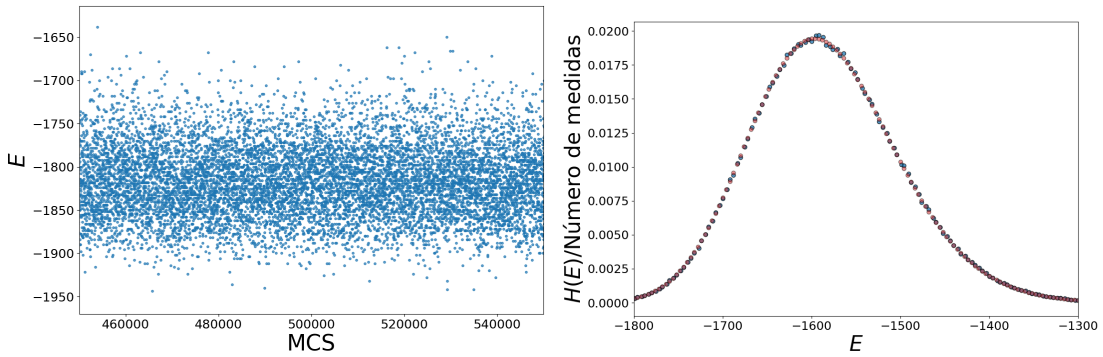

Figura 4.1. Trecho de série temporal da energia em uma simulação do Modelo de Potts. Resultados exatos e de simulação, do trecho, para $L = 32$ com temperatura $T = 1{,}1$. Os pontos azuis representam os histogramas de energia e os vermelhos os resultados exatos, estes com $T = 2{,}2$.

os comportamentos obtidos na literatura poderemos utilizar as configurações finais destas para o aprendizado não supervisionado. Os dados de interesse são as configurações finais dos sistemas, com estes podemos montar a matriz X, descrita na equação (2.3). A matriz X tem a forma $M \times N$, onde M é o número de amostras e N é o número de spins, isto é

L^2. A matriz é composta pelo estado final de cada spin e ordenada da seguinte forma

$$X = \begin{pmatrix} \theta_{1,1} & \theta_{1,2} & \cdots & \theta_{1,N-1} & \theta_{1,N} \\ & & \vdots & & \\ \theta_{M,1} & \theta_{M,2} & \cdots & \theta_{M,N-1} & \theta_{M,N} \end{pmatrix}_{M\times N} . \tag{4.5}$$

Assim, na equação (4.5) cada linha é uma simulação completa. Note que cada elemento da matriz pode assumir um valor descrito pela equação (4.2). Entretanto, para uma melhor visualização espacial dos futuros resultados do PCA, será utilizado uma versão modificada da matriz (4.5). Utilizaremos as direções dos spins nos eixos, da forma

$$X = \begin{pmatrix} \cos\theta_{1,1}, \sin\theta_{1,1} & \cdots & \cos\theta_{1,N}, \sin\theta_{1,N} \\ & \vdots & \\ \cos\theta_{M,1}, \sin\theta_{M,1} & \cdots & \cos\theta_{M,N}, \sin\theta_{M,N} \end{pmatrix}_{M\times 2N} . \tag{4.6}$$

Essa mudança não interfere no resultado, apenas modifica o arranjo espacial das projeções do PCA. A matriz utilizada será a descrita pela equação (4.6), onde foram adicionadas L^2 colunas, logo a forma dela é $M \times N$, onde M é o número de amostras e N é o número de projeções dos spins, isto é $2L^2$.

4.3 Resultados do PCA

Vamos começar visualizando exemplos de configurações finais, que serão a fonte dos nossos estudos. A Figura 4.2 mostra *snapshots* de configurações finais, com diferentes T, para o modelo de Potts de três diferentes q-estados 3, 4 e 7, todos com $L = 80$. Nas configurações $T < T_C$ é observado a predominância de um tipo de spin, quando temos $T \approx T_C$ notamos a formação de *clusters* do mesmo valor, e quando possuímos $T > T_C$ os sistemas estão desordenados. É visível a diferença, porém estamos interessados analisar uma grande quantidade de configurações com diferentes temperaturas.

Com uma grande quantidade de configurações finais podemos montar a matriz da equação (4.6). É necessário analisar em quantas características captamos a maior quantidade de variação. A Figura 4.3 mostra as primeiras razões de variância explicada, para $q = 3$, $q = 4$ e $q = 7$. Note que, diferente dos resultados do Modelo de Ising, do Capítulo 3, identificamos dois componentes principais dominantes. Esse fato está ligado que, a matriz utilizada não é a do valor dos spins, mas sim das direções deles nos eixos. Uma vez que possuímos uma informação e a transformamos em duas, é esperado obter

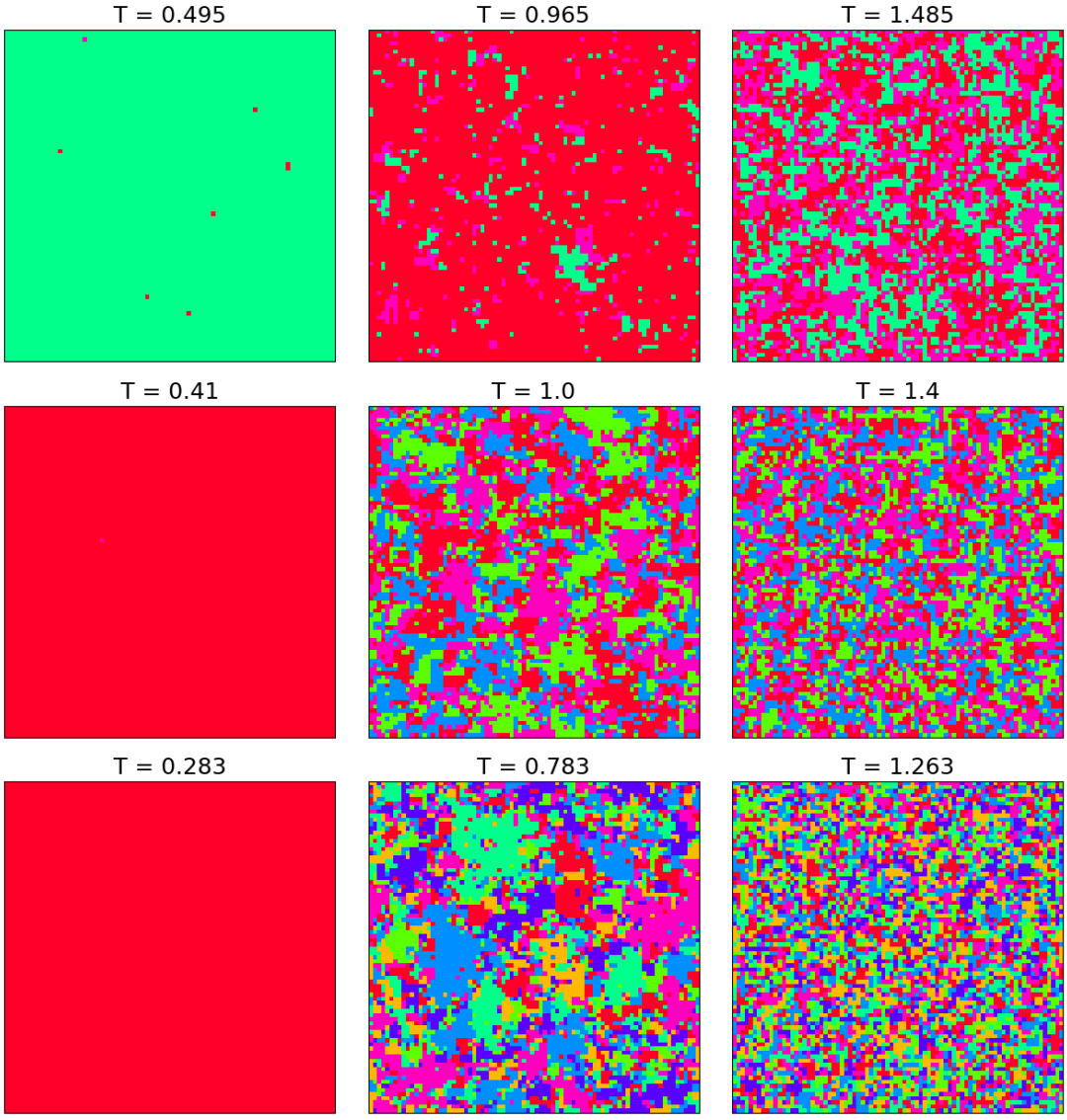

Figura 4.2. *Snapshots* de configurações finais de diferentes simulações para diferentes temperaturas, $q = 3$ (no topo), $q = 4$ (no centro) e $q = 7$ (na base), todos com $L = 80$. A grande diferença entre as temperaturas de simulação é visível.

o dobro de componentes principais relevantes. Serão nestes dois componentes principais que projetaremos os dados da matriz descrita na equação 4.6.

Então, fazendo a projeção dos dados nos componentes principais, seguindo a equação (2.5), obteremos os dados com as dimensões reduzidas. As Figuras 4.4,4.5 e 4.6 ilustram as projeções dos dados em seus componentes principais. Os pontos se agrupam com pontos de temperatura de simulação similar. Similarmente ao modelo de Ising, teremos pontos de maior temperatura se agrupam em $y_1 \approx 0$ e os de menor temperatura em $|y_1| > 0$. A mudança de um grupo de maior temperatura ocorre na região de temperatura

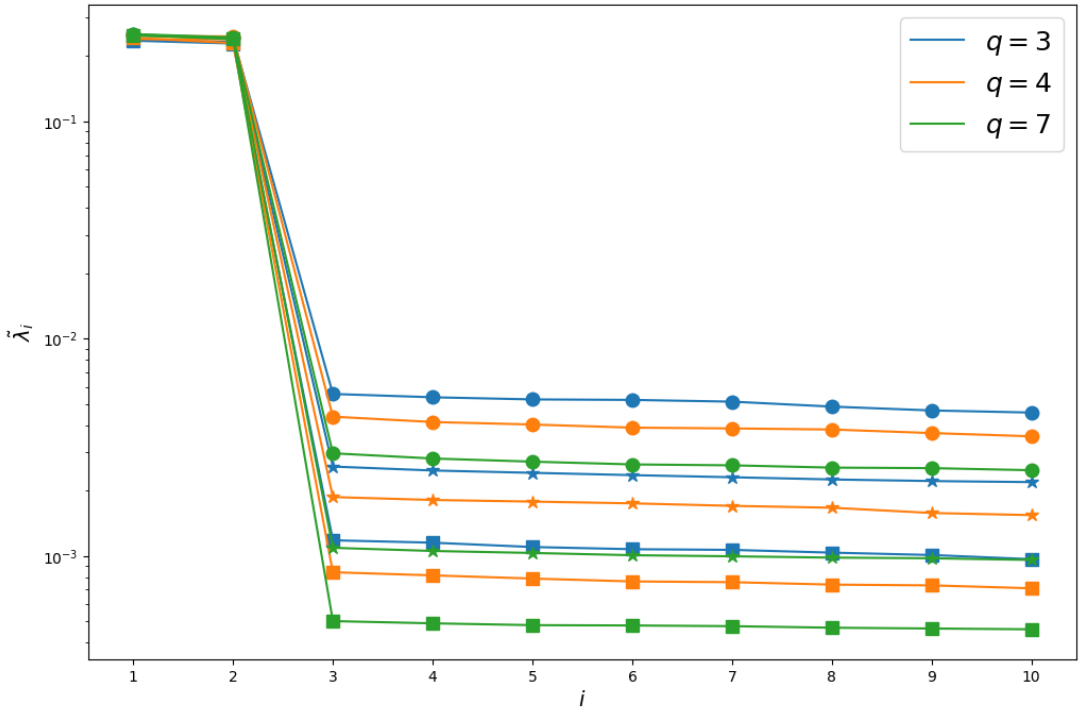

FIGURA 4.3. Razão da variação explicada, para $q = 3$, $q = 4$ e $q = 7$ com diferentes tamanhos de rede. O eixo $\tilde{\lambda}_i$ está na escala logarítmica para melhor visualização. Os pontos em círculo representam $L = 20$, em estrela os valores de $L = 40$ e em quadrado os valores de $L = 80$.

crítica. A transição de fase ocorre em $T \approx 0{,}995$, para $q = 3$, $T \approx 0{,}910$, para $q = 4$ e $T \approx 0{,}773$, para $q = 7$. Entretanto, diferente do resultado do modelo de Ising, temos q regiões de baixa temperatura. O número de regiões de baixa temperatura é igual ao número de q-estados do modelo de Potts utilizado.

Partindo apenas de configurações finais de sistemas como base do aprendizado não supervisionado foi possível reduzir a dimensão do sistema e encontrar comportamentos relacionados ao número de q-estados. A mudança da matriz descrita na equação (4.5) para a descrita na equação (4.6) transforma as informações que seriam inicialmente descritas em uma componente em duas, sem perder generalidade. Com dois componentes principais podemos visualizar melhor os padrões relativos aos q-estados. Independente do tamanho da rede de lado L foi observado o mesmo comportamento, embora para tamanhos maiores de rede o comportamento é mais visível.

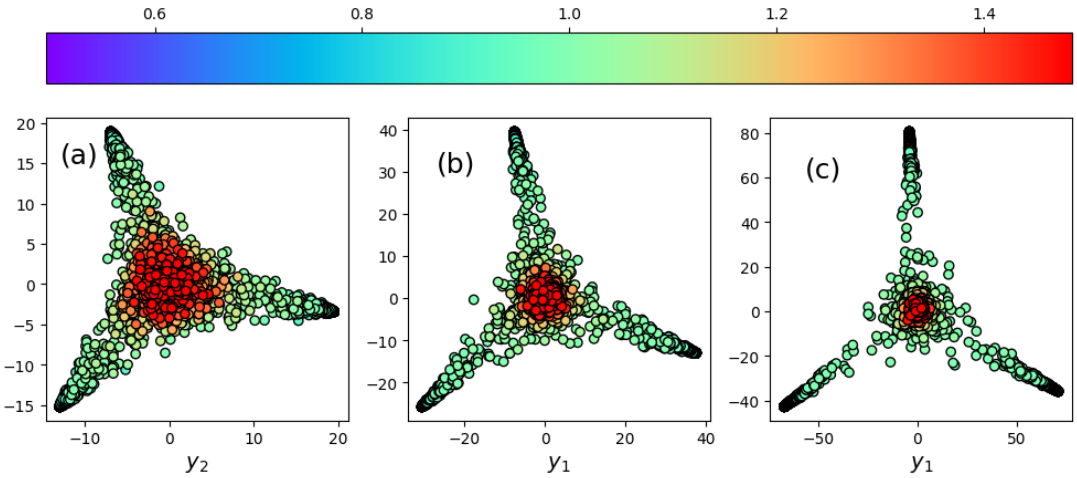

Figura 4.4. Projeções nos componentes principais para cada rede de lado L. A escala de cores representando a temperatura de simulação T e $q = 3$. Os painéis, (a), (b) e (c), representam as projeções de $L = 20$, $L = 40$ e $L = 80$, respectivamente.

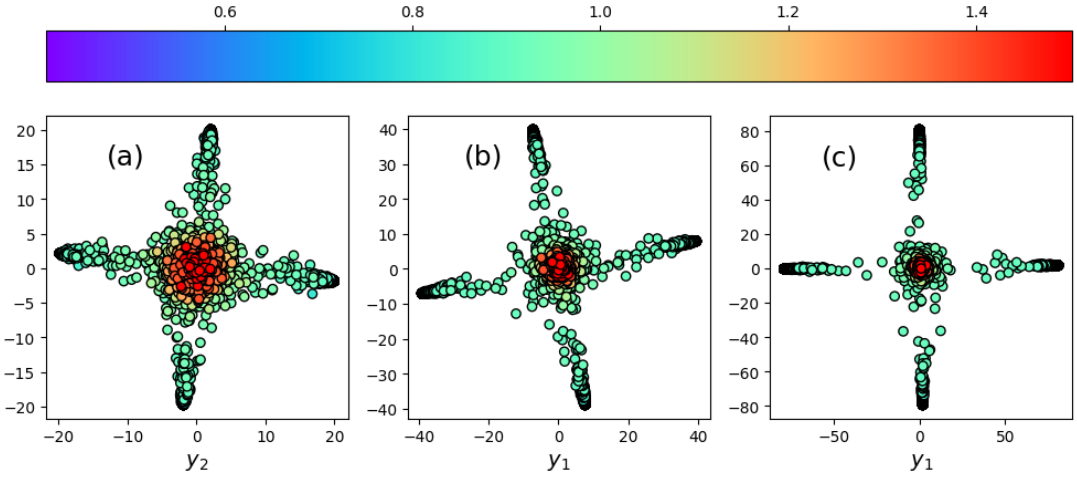

Figura 4.5. Projeções nos componentes principais para cada rede de lado L. A escala de cores representando a temperatura de simulação T e $q = 4$. Os painéis, (a), (b) e (c), representam as projeções de $L = 20$, $L = 40$ e $L = 80$, respectivamente.

4.4 Clusterização

Novamente iremos utilizar o algoritmo DBSCAN para identificar *clusters* nas projeções dos componentes principais da rede de lado $L = 80$, para os três q-estados utilizados, 3, 4 e 7. Precisamos definir dois hiperparâmetros necessários para aplicação do algoritmo: N_{min}, o número mínimo de pontos para um *cluster* ser considerado, e ε, distância máxima para dois pontos serem considerados vizinhos. N_{min} foi tomado como 25 por que estamos lidando com uma grande quantidade de dados e o valor de ε foi definido utilizando um

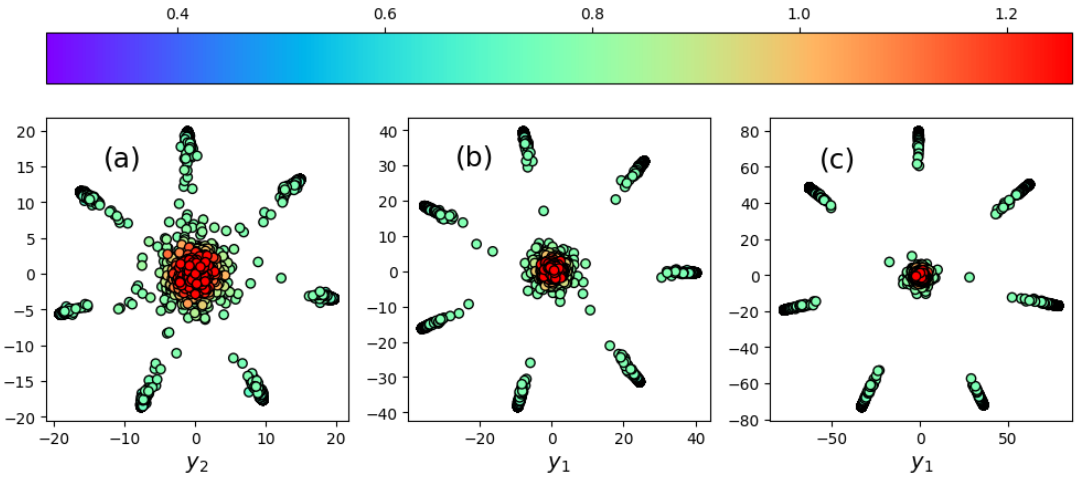

FIGURA 4.6. Projeções nos componentes principais para cada rede de lado L. A escala de cores representando a temperatura de simulação T e $q = 7$. Os painéis, (a), (b) e (c), representam as projeções de $L = 20$, $L = 40$ e $L = 80$, respectivamente.

algoritmo que necessita do N_{min} com fonte [26, 28]. A Figura 4.7 mostra a distância do vizinho mais próximo em função do número de vizinhos próximos. Precisamos escolher o ponto com maior curvatura [26], e este ponto foi tomado como $\varepsilon = 4{,}170$, para $q = 3$, $\varepsilon = 4{,}193$, para $q = 4$ e $\varepsilon = 4{,}186$, para $q = 7$.

Aplicando o DBSCAN nas projeções confirmamos os resultados que esperamos, q *clusters* de baixa temperatura e um *cluster* de alta temperatura. A Figura 4.8 mostra os *clusters* diferidos cada um com uma cor. Os pontos de ruído se encontram entre os *clusters* de baixa temperatura e o *cluster* de alta temperatura. A zona entre os *clusters* se encontra na região crítica, onde os pontos não pertencem a fase ordenada ou desordenada.

Partindo das projeções nos dois componentes principais, confirmamos os resultados que esperávamos inicialmente. Podemos encontrar q *clusters* de baixa temperatura e um de alta. Os pontos de ruído são da região $T \approx T_C$, onde os pontos não são classificados como pertencendo em nenhuma fase.

4.5 Parâmetros de Ordem

Para seguir a analise, vamos definir um análogo do parâmetro de ordem no espaço do PCA. Utilizaremos as projeções dos dados, da equação (2.5), para definir um fator γ_x,

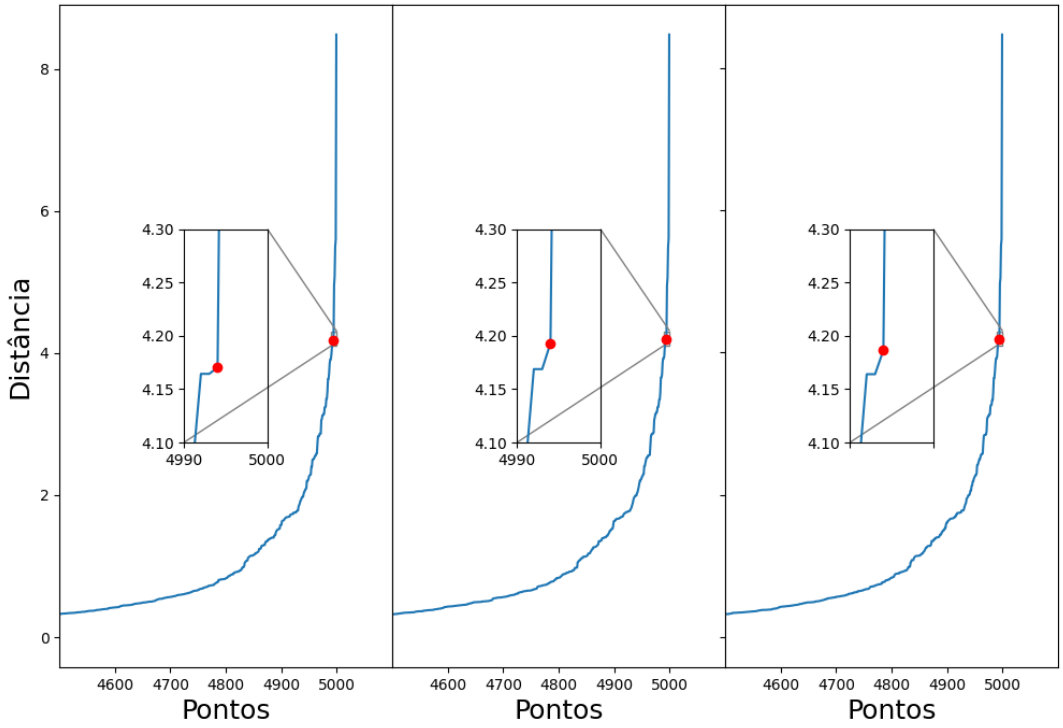

Figura 4.7. Distância em função dos n-pontos próximos próximos. Ampliado as regiões onde temos maiores curvaturas, com destaque para os pontos que tomaremos como ε. Os painéis, (a), (b) e (c), representam os modelos de Potts com $q = 3$, 4 e 7, respectivamente.

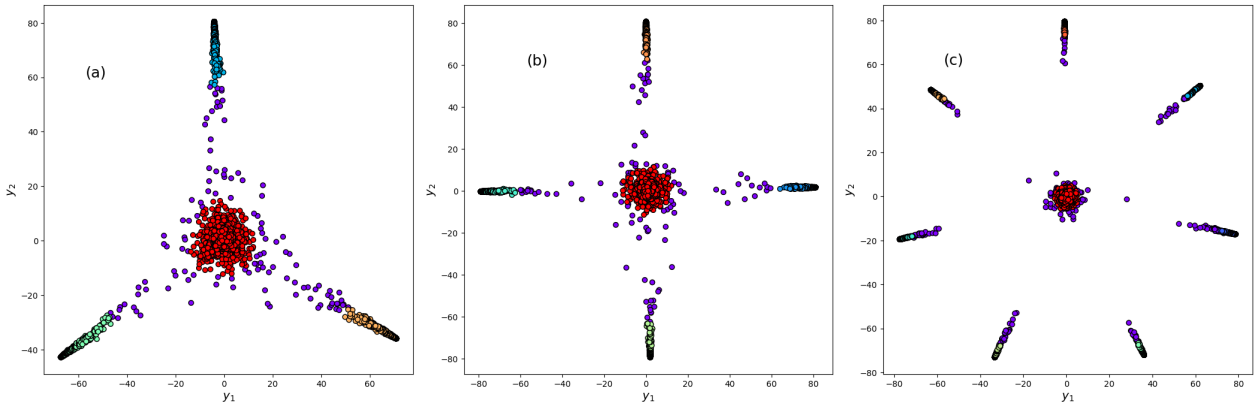

Figura 4.8. Resultado da clusterização utilizando o algoritmo DBSCAN. Podemos ver a formação de $q + 1$ *clusters* e a zona de ruído. Os painéis, (a), (b) e (c), representam os modelos de Potts com $q = 3$, 4 e 7, respectivamente.

da forma

$$\gamma_x = \sqrt{|y_1^x|^2 + |y_2^x|^2}, \tag{4.7}$$

onde y_i^x, com $i = 1, 2$ é a projeção do *snapshot* x. Utilizando a equação (4.7), e tomando a média das M amostras com temperatura fixa T, obtemos a a definição do parâmetro

de ordem [4]

$$\Gamma(T) = \frac{1}{ML}\sum_{x=1}^{M}\gamma_x(T) = \left\langle \frac{\gamma_x(T)}{L} \right\rangle_T . \tag{4.8}$$

Na Figura 4.9 vemos os parâmetros de ordem, calculados através da equação (4.8), das nossas simulações. O parâmetro de ordem possui um comportamento similar em todos tamanhos de rede, porém é mais evidente para $L = 80$. Este comportamento evidência os efeitos de tamanho finito das simulações. Nota-se uma grande flutuação estatística na simulação $q = 4$ com $L = 80$. Uma vez que obtemos um parâmetro de ordem para

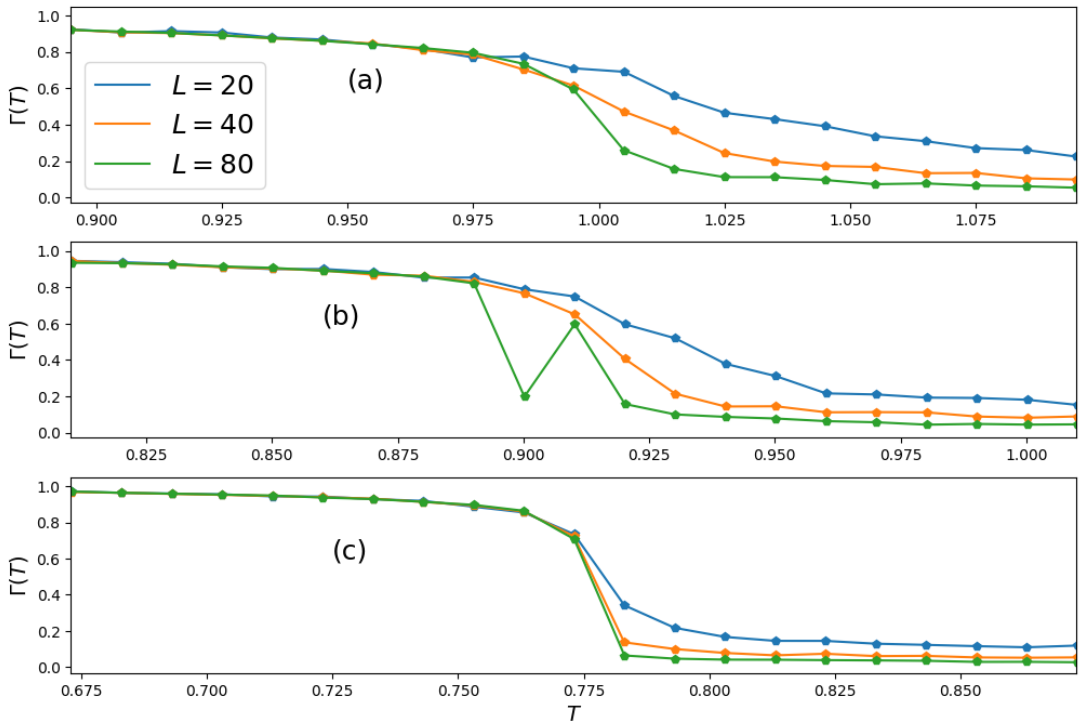

FIGURA 4.9. Parâmetros de ordem para simulações com rede de lado $L = 20$, $L = 40$ e $L = 80$. Os painéis, (a), (b) e (c), representam os modelos de Potts com $q = 3$, 4 e 7, respectivamente.

os sistemas, podemos estudar o comportamento utilizando alguma medida que apresente comportamento assintótico para diferentes regiões. Utilizaremos o parâmetro de Binder [31], que possui comportamento assintótico ao redor da temperatura crítica T_C e é independente do tamanho da rede de lado L. O parâmetro de Binder utilizado é definido por

$$B(T,L) = 1 - \frac{\left\langle \left(\gamma_x(T)/L\right)^4 \right\rangle_T}{3\left\langle \left(\gamma_x(T)/L\right)^2 \right\rangle_T^2}, \tag{4.9}$$

onde os fatores são os segundo e o quarto momento do parâmetro de ordem, da equação 4.8. Na definição do parâmetro de Binder fica claro a dependência do tamanho da rede L e da temperatura T. A vantagem de utilizar tal parâmetro é que, independente do tamanho da rede, possuiremos curvas de mesmos limites que se intersectam no ponto crítico T_C [31]. A Figura 4.9 mostra os parâmetros de Binder para diferentes q-estados, 3, 4 e 7, e diferentes tamanhos de rede de lado L. Notamos que, independente do tamanho

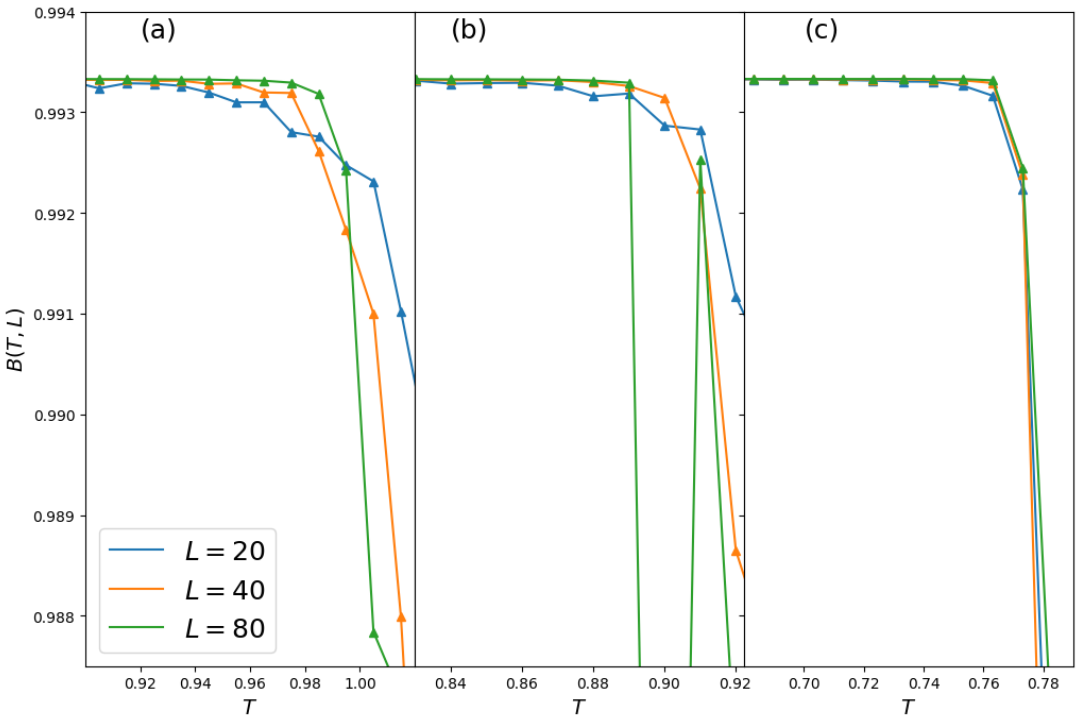

Figura 4.10. Parâmetros de Binder para simulações com rede de lado $L = 20$, $L = 40$ e $L = 80$. Os painéis, (a), (b) e (c), representam os modelos de Potts com $q = 3$, 4 e 7, respectivamente.

de rede, as curvas tendem a se intersectar na região da temperatura crítica. Também é visível a mudança significativa do comportamento do parâmetro ao redor de T_C. A flutuação de 4.8 também é vista aqui.

Partindo das projeções das configurações finais foi possível definir um parâmetro de ordem para o nosso modelo de Potts, dentro do espaço gerado pelas projeções do PCA. O parâmetro de ordem possui um comportamento similar para todos os tamanhos de rede, porém é mais claro para redes maiores. O parâmetro de ordem se assemelha muito com a magnetização do modelo de Potts [32], que não foi definido neste trabalho. O

parâmetro de Binder definido dentro do espaço do PCA elimina os efeitos de tamanho de rede, eles possuem um comportamento que apresenta mudança exatamente quando $T \approx T_C$. O comportamento do parâmetro de Binder no espaço do PCA se assemelha ao do original [31].

Capítulo 5

Conclusões e Perspectivas

5.1 Conclusões

Em um primeiro momento foram obtidos conhecimentos essenciais de simulação e programação de algoritmos de aprendizado de máquina. Reproduzimos os resultados já esperados do modelo de Ising, porém utilizando algoritmos que ainda não estão amplamente difundidos na física. Foi possível compreender que, ao aplicar o PCA, o algoritmo identifica que o modelo possui uma característica mais relevante que outras. Em uma análise nota-se que esta característica é a magnetização, condizente com o que já era esperado. Utilizando algoritmos de clusterização conseguimos separar os comportamentos em seus devidos grupos, de baixa e alta temperatura.

Para o modelo de Potts pudemos notar que a posição espacial das projeções depende do número de q-estados que estamos utilizando. A projeção dos dados cria *clusters* igualmente espaçados, o que é plausível com a definição utilizada. Ainda no modelo de Potts foi possível definir e avaliar a região crítica utilizando parâmetros definidos dentro do espaço criado pelas projeções do PCA.

É interessante apontar que a única fonte do aprendizado pelo PCA foram as configurações finais dos sistemas. Em ambos modelos, Ising e Potts, foi possível reduzir a dimensão dos sistemas, avaliar a região crítica e tomar percepções sobre os parâmetros de ordem. Entretanto, os métodos não identificaram a diferença de ordem da transição de fase do Modelo de Potts.

5.2 Perspectivas

A aplicação de algoritmos de aprendizado não supervisionados nos modelos de Ising e Potts abrem espaço para o pensamento de aplicação em outros modelos. Modelos como os de Gás de Rede, Heisenberg ou XY podem ser estudados utilizando esta mesma linha de estudo. Uma vez que compreendemos a aplicação do método nos modelos simples, podemos tentar obter novos resultados em modelos mais complexos. A diferença na ordem da transição de fase do modelo de Potts não foi identificada, assim é possível buscar formas de entender se o PCA consegue identificar a ordem transição utilizando configurações como fonte de estudo.

Ainda na parte de aprendizado de máquina podemos pensar em aplicar os resultados obtidos no PCA em outros algoritmos. Uma vez que reduzimos a dimensão do sistema com o PCA podemos aplicar em outros modelos, como redes neurais, e tentar obter informações sem fornecer rótulos [8].

Outra possibilidade seria pensar em aplicar os princípios físicos aqui obtidos na criação das redes neurais, que são algoritmos mais complexos. É possível tanto estudar os modelos físicos utilizando aprendizado de máquina quanto estudar aprendizado de máquina por uma abordagem física [12, 13]. Nos últimos anos tem-se notado um crescimento nesta área, uma vez que a difusão do aprendizado de máquina tem se tornado mais acessível.

Referências Bibliográficas

[1] P. G. Breen, C. N. Foley, T. Boekholt, and S. P. Zwart, "Newton versus the machine: solving the chaotic three-body problem using deep neural networks," *Monthly Notices of the Royal Astronomical Society*, vol. 494, p. 2465–2470, Apr 2020.

[2] S. Alexander, S. Gleyzer, H. Parul, P. Reddy, M. W. Toomey, E. Usai, and R. V. Klar, "Decoding dark matter substructure without supervision," 2020.

[3] F. D'Angelo and L. Böttcher, "Learning the ising model with generative neural networks," *Physical Review Research*, vol. 2, Jun 2020.

[4] A. Tirelli, D. O. Carvalho, L. A. Oliveira, J. P. Lima, N. C. Costa, and R. R. d. Santos, "Unsupervised machine learning approaches to the q-state potts model," 2021.

[5] G. Carleo and M. Troyer, "Solving the quantum many-body problem with artificial neural networks," *Science*, vol. 355, p. 602–606, Feb 2017.

[6] T. Morishita and S. Todo, "Randomized-gauge test for machine learning of ising model order parameter," *Journal of the Physical Society of Japan*, vol. 91, apr 2022.

[7] A. Tanaka and A. Tomiya, "Detection of phase transition via convolutional neural networks," *Journal of the Physical Society of Japan*, vol. 86, p. 063001, jun 2017.

[8] K. Fukushima and K. Sakai, "Can a cnn trained on the ising model detect the phase transition of the q-state potts model?," 2021.

[9] L. Wang, "Discovering phase transitions with unsupervised learning," *Phys. Rev. B*, vol. 94, p. 195105, Nov 2016.

[10] S. Salinas, *Introduction to statistical physics*. Springer Science & Business Media, 2001.

[11] R. J. Baxter, *Exactly solved models in statistical mechanics*. London: Academic, 1982.

[12] L. Zdeborová, "Understanding deep learning is also a job for physicists," *Nature Physics*, vol. 16, pp. 602–604, Jun 2020.

[13] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, "Machine learning and the physical sciences," *Reviews of Modern Physics*, vol. 91, Dec 2019.

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

[15] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The journal of chemical physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[16] D. P. Landau and K. Binder, *A guide to Monte Carlo simulations in statistical physics*. Cambridge university press, 2014.

[17] W. Krauth, *Statistical Mechanics: Algorithms and Computations*. Oxford Master Series in Physics, Oxford University Press, UK, 2006.

[18] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, 1959.

[19] A. Geron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Inc., 2nd ed., 2019.

[20] L. Wang, "Exploring cluster monte carlo updates with boltzmann machines," *Physical Review E*, vol. 96, nov 2017.

[21] A. Tanaka, A. Tomiya, and K. Hashimoto, *Deep Learning and Physics*. Mathematical Physics Studies, Springer, 2021.

[22] scikit-learn: scikit-learn.org.

[23] PyTorch: pytorch.org.

[24] M. Ester, H. P. Kriegel, J. Sander, and X. Xiaowei, "A density-based algorithm for discovering clusters in large spatial databases with noise," 12 1996.

[25] E. Schubert, J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "Dbscan revisited, revisited," *ACM Transactions on Database Systems (TODS)*, vol. 42, pp. 1 – 21, 2017.

[26] N. Rahmah and I. S. Sitanggang, "Determination of optimal epsilon (eps) value on DBSCAN algorithm to clustering data on peatland hotspots in sumatra," *IOP Conference Series: Earth and Environmental Science*, vol. 31, p. 012012, jan 2016.

[27] D. Sculley, "Web-scale k-means clustering," pp. 1177–1178, 01 2010.

[28] V. Satopaa, J. Albrecht, D. Irwin, and B. Raghavan, "Finding a "kneedle"in a haystack: Detecting knee points in system behavior," in *2011 31st International Conference on Distributed Computing Systems Workshops*, pp. 166–171, 2011.

[29] P. D. Beale, "Exact distribution of energies in the two-dimensional ising model," *Phys. Rev. Lett.*, vol. 76, pp. 78–81, Jan 1996.

[30] F. Y. Wu, "The potts model," *Rev. Mod. Phys.*, vol. 54, pp. 235–268, Jan 1982.

[31] L. Barone, E. Marinari, G. Organtini, and F. Ricci-Tersenghi, *Scientific Programming: C-language, Algorithms and Models in Science*. World Scientific, 2013.

[32] E. E. Ferrero, J. P. D. Francesco, N. Wolovick, and S. A. Cannas, "q-state potts model metastability study using optimized GPU-based monte carlo algorithms," *Computer Physics Communications*, vol. 183, pp. 1578–1587, aug 2012.