

---

UNIVERSIDADE ESTADUAL DE MARINGÁ  
DEPARTAMENTO DE FÍSICA

---

ARTHUR AUGUSTO BARIZON PESSA

DESENVOLVIMENTO E APLICAÇÃO DE  
MÉTODOS ORDINAIS À ANÁLISE DE  
IMAGENS E SÉRIES TEMPORAIS

Maringá, Março de 2023.

---

---

UNIVERSIDADE ESTADUAL DE MARINGÁ  
DEPARTAMENTO DE FÍSICA

---

ARTHUR AUGUSTO BARIZON PESSA

DESENVOLVIMENTO E APLICAÇÃO DE  
MÉTODOS ORDINAIS À ANÁLISE DE  
IMAGENS E SÉRIES TEMPORAIS

*Tese de doutorado apresentada ao Programa de Pós-  
Graduação em Física da Universidade Estadual de  
Maringá.*

Orientador: Prof. Dr. Haroldo Valentin Ribeiro

Maringá, Março de 2023.

---

Dados Internacionais de Catalogação-na-Publicação (CIP)  
(Biblioteca Central - UEM, Maringá - PR, Brasil)

P475d

Pessa, Arthur Augusto Barizon

Desenvolvimento e aplicação de métodos ordinais à análise de imagens e séries temporais / Arthur Augusto Barizon Pessa. -- Maringá, PR, 2023.  
119 f.: il. color., figs., tabs.

Orientador: Prof. Dr. Haroldo Valentin Ribeiro.

Tese (Doutorado) - Universidade Estadual de Maringá, Centro de Ciências Exatas, Departamento de Física, Programa de Pós-Graduação em Física, 2023.

1. Entropia de permutação. 2. Redes ordinais. 3. Análise de séries temporais. 4. Análise de imagens. 5. Aprendizagem estatística. I. Ribeiro, Haroldo Valentin, orient. II. Universidade Estadual de Maringá. Centro de Ciências Exatas. Departamento de Física. Programa de Pós-Graduação em Física. III. Título.

CDD 23.ed. 530.13

Marinalva Aparecida Spolon Almeida - 9/1094

ARTHUR AUGUSTO BARIZON PESSA

**DESENVOLVIMENTO E APLICAÇÃO DE MÉTODOS ORDINAIS À ANÁLISE DE  
IMAGENS E SÉRIES TEMPORAIS**

Tese apresentada à Universidade Estadual de Maringá, como requisito parcial para a obtenção do título de doutor.

Aprovado em: Maringá, 28 de março de 2023.

**BANCA EXAMINADORA**

---

Prof. Dr. Haroldo Valentin Ribeiro  
Universidade Estadual de Maringá – UEM

---

Prof. Dr. Osvaldo Anibal Rosso  
Universidade Federal de Alagoas – UFAL

---

Prof. Dr. Luciano José Zunino  
Universidad Nacional de La Plata – Argentina

---

Prof. Dr. Renio dos Santos Mendes  
Universidade Estadual de Maringá – UEM

---

Prof. Dr. Rafael Soares Zola  
Universidade Tecnológica Federal do Paraná – UTFPR/Apucarana  
Programa de Pós-Graduação em Física – PFI/UEM

## Resumo

Como consequência da notável dificuldade em definir o conceito de complexidade formalmente, diferentes técnicas têm sido propostas para tentar quantificar a complexidade de diversos sistemas. A entropia de permutação, resultado da união de ideias de teoria de informação e análise de séries temporais não lineares, é uma medida de complexidade de séries temporais que deu origem a um conjunto de ferramentas de análise de dados coletivamente chamado de formalismo de Bandt e Pompe ou de métodos ordinais. A esse formalismo faltava, entretanto, uma implementação computacional suficientemente abrangente e na forma de código aberto. Assim, a primeira contribuição resultante desta tese é suprir essa demanda com o desenvolvimento do pacote `ordpy`, um módulo de funções escrito em linguagem Python. O Capítulo 1 apresenta as funções que compõem o `ordpy` por meio de uma revisão e replicação dos principais resultados do formalismo de Bandt e Pompe. Dentre os métodos implementados no `ordpy`, destacamos nossa proposta de generalização do algoritmo de redes ordinais que viabiliza o mapeamento de imagens em redes complexas. No Capítulo 2, avaliamos restrições fundamentais de conectividade para determinar a forma exata de redes ordinais mapeadas a partir de dados bidimensionais completamente aleatórios. Em seguida, aplicamos esse mapeamento a imagens geradas por simulações computacionais de ornamentos geométricos periódicos, movimento browniano fracionário bidimensional, superfícies de Ising e também a um conjunto de texturas naturais, a fim de caracterizar esses sistemas usando propriedades das redes ordinais. A robustez de redes ordinais mapeadas de imagens ruidosas é examinada em cenários de classificação de imagens. No Capítulo 3, apresentamos uma aplicação de redes ordinais para caracterizar texturas de cristais líquidos, na qual os pesos das ligações de redes ordinais são usados como variáveis preditivas junto a algoritmos de aprendizagem estatística para determinar propriedades desses materiais. Especificamente, mostramos que nossa abordagem permite identificar e classificar transições de fase, determinar concentrações de dopantes e também a temperatura de amostras de cristal líquido. Por fim, no Capítulo 4, investigamos o complexo movimento que pequenos pedaços de papel exibem ao cair pelo ar. Para isso, calculamos a entropia e a complexidade de séries temporais relacionadas à área observável desses papéis em queda em diversos experimentos. Nossos resultados mostram que quedas caóticas ou em forma de tombos sucessivos estão localizadas em regiões distintas do plano complexidade-entropia. Além disso, verificamos que essas duas medidas, aliadas a um algoritmo de agrupamento, são capazes de discriminar entre os dois tipos de movimento com precisão comparável a outras abordagens baseadas em quantidades físicas mais tradicionais, com a vantagem de não requererem a reconstrução das trajetórias tridimensionais das quedas dos papéis.

**Palavras-chave:** Entropia de permutação. Redes ordinais. Análise de séries temporais. Análise de imagens. Cristais líquidos. Papéis em queda livre. Aprendizagem estatística.

## Abstract

Due to the notable difficulty in formally defining complexity, different techniques have been proposed for measuring and trying to quantify the complexity of various systems. Permutation entropy, a combination of ideas stemming from information theory and nonlinear time series analysis, is a complexity measure for time series that gave origin to a set of data analysis tools collectively known as the Bandt and Pompe formalism or ordinal methods. This formalism lacked, however, a sufficiently broad and open-source computational implementation. Thus, the first contribution resulting from this thesis is to reduce this gap with the development of `ordpy`, a Python module that implements several of the principal ordinal methods. Chapter 1 presents the functions implemented in `ordpy` alongside a revision and replication of some of the most important results in the Bandt and Pompe formalism. Between the methods implemented in `ordpy`, we highlight our proposal for a generalization of the ordinal network algorithm for mapping images into complex networks. In Chapter 2, we analyze fundamental connectivity constraints in ordinal networks to determine their topology when mapped from a bidimensional random field. Next, we apply our algorithm to images generated from computer simulations of periodic geometric ornaments, bidimensional fractional Brownian motion, Ising surfaces, and a set of natural textures, in order to characterize these systems using ordinal network properties. The robustness of ordinal networks mapped from noisy images is examined in image classification tasks. In Chapter 3, we present an application of ordinal networks to liquid crystals' textures in which edge weights of ordinal networks are used as predictors alongside statistical learning algorithms to determine physical properties of these complex materials. Specifically, we show that our approach can identify and classify phase transitions as well as determine doping concentration and temperature in liquid crystal samples. Finally, in Chapter 4, we investigate the complex movements displayed by slips of paper when falling through the air. To do so, we evaluate entropy and complexity from time series related to the observable area of these falling papers in several experiments. Our results show that chaotic and tumbling motions are localized in different regions of the complexity-entropy plane. Furthermore, combined with a clustering algorithm, these two metrics are capable of discriminating between the two types of movements with precision comparable to other approaches based on more traditional physical quantities, but with the advantage of not requiring the reconstruction of the three-dimensional falling trajectories.

**Keywords:** Permutation entropy. Ordinal networks. Time series analysis. Image analysis. Liquid crystals. Falling papers. Statistical learning.

<b>Introdução</b>		<b>7</b>
<b>1 <code>ordpy</code>: um pacote Python para análise de dados com entropia de permutação e redes ordinais</b>		<b>10</b>
1.1 <code>ordpy</code> . . . . .		10
1.2 Distribuições ordinais, entropia de permutação e o plano complexidade-entropia		11
1.3 Aplicações do formalismo de Bandt e Pompe usando <code>ordpy</code> . . . . .		19
1.4 Padrões ordinais faltantes . . . . .		25
1.5 Entropias de permutação de Tsallis e Rényi . . . . .		27
1.6 Redes ordinais . . . . .		31
1.7 Aplicações de redes ordinais usando <code>ordpy</code> . . . . .		38
<b>2 Estrutura topológica de redes ordinais mapeadas de dados bidimensionais e aplicações</b>		<b>43</b>
2.1 Mapeamento de imagens em redes ordinais . . . . .		43
2.2 Restrições de conectividade em redes ordinais . . . . .		45
2.3 Redes ordinais aleatórias . . . . .		48
2.4 Redes ordinais de ornamentos periódicos ruidosos . . . . .		50
2.5 Redes ordinais de imagens brownianas fracionárias . . . . .		52
2.6 Redes ordinais de superfícies de Ising . . . . .		54
2.7 Redes ordinais de texturas naturais . . . . .		56
2.8 Comparação entre redes ordinais e métodos tradicionais de análise de imagem		58
<b>3 Aplicação de redes ordinais à predição de propriedades físicas de cristais líquidos</b>		<b>62</b>

3.1	Aprendizagem de máquina, redes ordinais e cristais líquidos . . . . .	62
3.2	Apresentação dos dados e procedimentos experimentais . . . . .	63
3.3	Classificação de transições de fase em cristais líquidos usando redes ordinais .	65
3.4	Determinação de concentrações de dopantes em cristais líquidos usando redes ordinais . . . . .	68
3.5	Previsão da temperatura de cristais líquidos usando redes ordinais . . . . .	70
<b>4</b>	<b>Agrupamento de movimentos de papéis em queda livre usando entropia e complexidade</b>	<b>73</b>
4.1	O problema dos papéis em queda livre . . . . .	73
4.2	Apresentação dos dados . . . . .	74
4.3	Classificação de movimentos de queda livre . . . . .	76
	<b>Conclusões e Perspectivas</b>	<b>81</b>
	<b>A Seleção de parâmetros de <i>embedding</i> na análise de séries temporais</b>	<b>83</b>
	<b>B Sistemas dinâmicos</b>	<b>86</b>
	<b>C Processos estocásticos</b>	<b>88</b>
	<b>D Matrizes de coocorrência de nível de cinza</b>	<b>91</b>
	<b>E Cristais líquidos</b>	<b>93</b>
	<b>F Aprendizagem estatística</b>	<b>95</b>
	<b>Referências Bibliográficas</b>	<b>102</b>

Historicamente, alguns dos conceitos mais fundamentais da ciência são também os mais difíceis de serem definidos [1]. Entretanto, o aspecto operacional, que determina como uma quantidade é medida, pode ser, muitas vezes, mais importante do que uma definição suficientemente abrangente e precisa [1]. De certa forma, é isso o que acontece com o conceito de complexidade, em torno do qual toda uma ciência recentemente se desenvolveu [2]. Intuitivamente, a ideia de complexidade deve descrever um objeto ou processo entre o determinismo e a aleatoriedade, uma vez que a geração e descrição (mesmo que estatística) de ambos os tipos de processos pode ser considerada simples [3, 4]. Apesar disso, décadas de pesquisa têm demonstrado que o conceito permanece elusivo e sem qualquer definição universalmente aceita [2]. Dessa forma, muito tem sido feito para o desenvolvimento de medidas de complexidade, as quais abundam na literatura científica [5]. Tal como ocorre no estudo de sistemas complexos, muitas medidas de complexidade são constituídas a partir de uma combinação de elementos teóricos de várias áreas do conhecimento, principalmente física estatística, ciência da computação, teoria de informação e sistemas dinâmicos [2]. É ainda bastante comum que essas medidas sejam calculadas a partir de séries temporais associadas a um objeto ou sistema, fazendo da avaliação da complexidade de uma sequência numérica uma estimativa substitutiva da complexidade do objeto [2, 5].

Nesse contexto, a entropia de permutação, proposta por Bandt e Pompe [6], é uma medida de complexidade de séries temporais que surgiu de uma combinação de ideias advindas do estudo de séries temporais não lineares [7, 8] e teoria de informação [9, 10]. Seus principais méritos são a simplicidade, robustez à adição de ruído e eficiência computacional [6]. Em palavras, a entropia de permutação é a entropia de Shannon [9] calculada a partir de uma distribuição de probabilidade de um conjunto de permutações (ou padrões ordinais) obtidos a partir da simbolização de pequenas partições na qual uma série temporal é subdividida. Esses padrões ordinais informam acerca de relações de amplitude entre observações em cada

uma das partições. Para ilustrar melhor essa abordagem, consideramos a seguinte sequência de quatro números:  $x_t = \{8, 1, 6, 4\}$ . Ao subdividirmos essa sequência em partições contendo duas observações, encontramos:  $(8, 1); (1, 6); (6, 4)$ . Notamos que as observações da primeira e terceira partições estão em ordem decrescente e, por isso, associamos a permutação  $(1, 0)$  a essas partições. A segunda partição, contrariamente, tem seus elementos ordenados crescentemente e a permutação  $(0, 1)$  é associada a ela. A frequência relativa de ocorrência dessas permutações dentro da série temporal ( $p_{01} = 1/3$  e  $p_{10} = 2/3$ ) resulta em uma distribuição de probabilidade que quando aliada à entropia de Shannon dá origem à entropia de permutação [6]. Esse engenhoso procedimento para a obtenção de uma distribuição de probabilidade a partir de uma série temporal é o chamado método de simbolização de Bandt e Pompe [6].

Desde que foi proposta em 2002, a entropia de permutação difundiu-se enormemente entre pesquisadores trabalhando com séries temporais oriundas de diversos sistemas e disciplinas. Isso pode ser observado em aplicações bem-sucedidas em ciências biomédicas [11], econofísica [12], ciências físicas [13] e engenharia [14]. Essas e outras aplicações da entropia de permutação também mostram uma grande pluralidade no uso desta medida, como no monitoramento do regime dinâmico de um sistema mecânico (verificação de sinais de desgaste mecânico) [14], na detecção de anomalias em séries temporais (registros equivocados em medidas geológicas de temperatura) [13], na caracterização de séries temporais (distinção entre ruído ou caos) [11] e em testes de independência entre observações de uma sequência numérica [15].

Para além do próprio sucesso da entropia de permutação como uma medida de complexidade, o procedimento de simbolização de Bandt e Pompe inspirou e possibilitou muitas outras medidas de complexidade e ferramentas utilizadas em estudos de séries temporais e imagens. No caso de séries temporais, algumas modificações da proposta original de Bandt e Pompe utilizam a distribuição de probabilidade de padrões ordinais conjuntamente a outras formas entrópicas, como aquelas propostas por Tsallis [16] e Rényi [17]. Outras tentativas de criar novas medidas generalizam o algoritmo de simbolização de Bandt e Pompe a fim de obter padrões ordinais em múltiplas escalas de tempo [18–20], incluem informação associada à amplitude do sinal [21–24] ou consideram diferentes maneiras de lidar com igualdades dentro de uma partição [25, 26]. Além disso, a generalização do procedimento de simbolização de Bandt e Pompe para duas dimensões [27, 28] tornou possível o uso da entropia de permutação para a análise de imagens. Desde então, a aplicabilidade desse método a imagens tem sido corroborada por diversas investigações relacionadas a, por exemplo, obras de arte [29] e cristais líquidos [30]. Mais recentemente, a ideia de padrões ordinais também tem sido utilizada no mapeamento de séries temporais e imagens em redes complexas [31–33]. Essas redes são conhecidas como redes ordinais e fazem parte de um conjunto de transformações inicialmente chamadas de redes de séries temporais [34].

Considerando o nível de desenvolvimento e a diversidade de aplicações, o formalismo da

entropia de permutação representa um conjunto de ferramentas com grande potencial de uso em análise de dados, especialmente quando consideramos o aumento da disponibilidade de grandes conjuntos de dados [35, 36] e a demanda por métodos computacionalmente eficientes na extração de informação útil dos mesmos [37, 38].

Sendo assim, esta tese tem como principal objetivo contribuir para o desenvolvimento do formalismo da entropia de permutação e fomentar o uso desses métodos por meio de aplicações a sistemas físicos. Com esse fim, este trabalho encontra-se organizado como segue. No Capítulo 1, realizamos uma revisão do desenvolvimento do formalismo de Bandt e Pompe nas duas décadas desde o artigo seminal desses pesquisadores. Essa revisão é realizada concomitantemente à apresentação do módulo de funções `ordpy` [39], um pacote de código aberto escrito em Python que desenvolvemos com o intuito de implementar os principais métodos para o estudo de séries temporais e imagens pertencentes ao formalismo da entropia de permutação. Os métodos são formalizados matematicamente e alguns dos principais resultados da literatura são replicados enquanto o uso do pacote é ilustrado por pequenos trechos de código. No Capítulo 2, características topológicas de redes ordinais, derivadas do procedimento de simbolização de Bandt e Pompe, quando aplicadas a imagens, são expostas e discutidas detalhadamente [33]. Experimentos com alguns conjuntos de imagens geradas via simulações computacionais ilustram como propriedades de redes ordinais podem ser exploradas na procura por padrões em imagens. No Capítulo 3, investigamos a aplicação da transformação de redes ordinais na caracterização de texturas ópticas obtidas experimentalmente de amostras de cristais líquidos [40]. Mais especificamente, estudamos se a transformação é suficientemente informativa para, aliada a algoritmos de aprendizagem de máquina, ser utilizada como característica preditiva de propriedades físicas como a fase, concentração de dopante e temperatura de amostras de cristais líquidos. No Capítulo 4, entropia de permutação e complexidade estatística são empregadas como características discriminatórias no estudo do movimento de queda livre de pequenos pedaços de papel. Em um contexto no qual não existe uma classificação definitiva quanto a movimentos de queda caóticos ou que se dão em tombos sucessivos (*tumbling*), o plano complexidade-entropia oferece uma classificação consistente para esses dois tipos de movimento. Por fim, encerramos este trabalho com considerações finais acerca dos tópicos apresentados.

Para tornar este trabalho autocontido, alguns apêndices complementam o texto principal quanto à escolha apropriada de parâmetros no formalismo da entropia de permutação (Apêndice A) e trazem definições de sistemas dinâmicos (Apêndice B) e processos estocásticos (Apêndice C) apresentados ao longo do texto. Além disso, definimos as matrizes de coocorrência de nível de cinza (Apêndice D), fazemos uma breve revisão sobre conceitos básicos de cristais líquidos (Apêndice E) e apresentamos aspectos essenciais relacionados a algoritmos de aprendizagem estatística (Apêndice F).

---

## `ordpy`: um pacote Python para análise de dados com entropia de permutação e redes ordinais

---

Neste primeiro capítulo, os principais métodos de análise de dados associados ao formalismo da entropia de permutação são apresentados. Essas ferramentas são formalizadas matematicamente e suas implementações no pacote Python `ordpy` [39] são apresentadas em fragmentos curtos de código. Conjuntamente, utilizando o `ordpy`, reproduzimos resultados encontrados na literatura a fim de validar nosso código fonte.

### 1.1 `ordpy`

A despeito do razoável intervalo de tempo entre o presente trabalho e o artigo seminal de Bandt e Pompe [6], não havia uma implementação computacional em código aberto, coesa e razoavelmente abrangente da entropia de permutação e dos métodos ordinais derivados dessa técnica. Dessa forma, o pacote `ordpy` foi desenvolvido a fim de solucionar esse problema. Para além de preferências pessoais, o `ordpy` foi escrito na linguagem de programação Python tendo em vista seu uso disseminado em computação científica [41] e extensivo suporte da comunidade de usuários [42]. O módulo `ordpy` tem como única dependência o pacote `numpy` [41], uma biblioteca Python fundamental que otimiza cálculos com vetores e funções matemáticas que operam sobre esses vetores.

No restante deste capítulo, apresentamos as funções contidas no pacote `ordpy` e ilustramos seus usos ao mesmo tempo que uma revisão dos desenvolvimentos relativos à entropia de permutação é apresentada. Alternamos entre a formalização das diferentes técnicas e a apresentação das funções por meio de fragmentos de códigos que as implementam, além

de replicarmos resultados contidos na literatura. O código fonte do pacote `ordpy` está livremente disponível em um repositório do GitHub ([github.com/arthurpessa/ordpy](https://github.com/arthurpessa/ordpy)), bem como a documentação de todas as suas funções ([arthurpessa.github.io/ordpy](https://arthurpessa.github.io/ordpy)). O módulo também pode ser instalado usando o *Python Package Index* (PyPI) via:

```
$ pip install ordpy
```

Os códigos e dados necessários para a replicação de todos os experimentos, análises e figuras apresentados neste primeiro capítulo estão detalhados na forma de um *Jupyter notebook*, uma ferramenta de que permite a execução interativa de códigos Python [43, 44]. Esse *notebook* também pode ser encontrado no repositório do pacote `ordpy`.

## 1.2 Distribuições ordinais, entropia de permutação e o plano complexidade-entropia

Como já mencionamos, a entropia de permutação é a entropia de Shannon calculada a partir de uma distribuição de probabilidade dos padrões ordinais (permutações) de uma série temporal. Essa distribuição de probabilidade é chamada de distribuição ordinal ou distribuição de padrões ordinais e o processo de simbolização utilizado na estimativa dessa distribuição é o procedimento de simbolização de Bandt e Pompe<sup>1</sup>. Para descrever esse procedimento, consideraremos uma série temporal<sup>2</sup> arbitrária  $\{x_t\}_{t=1,\dots,N_x}$ . Primeiramente, dividimos essa série temporal em  $n_x = N_x - (d_x - 1)\tau_x$  partições constituídas de  $d_x > 1$  observações distantes  $\tau_x \geq 1$  intervalos de tempo entre si. Fixados os valores de  $d_x$  e  $\tau_x$ , cada partição pode ser representada por

$$w_p = (x_p, x_{p+\tau_x}, x_{p+2\tau_x}, \dots, x_{p+(d_x-2)\tau_x}, x_{p+(d_x-1)\tau_x}), \quad (1.1)$$

com  $p = 1, \dots, n_x$  indicando o índice da partição. Os parâmetros  $d_x$  e  $\tau_x$ , os dois únicos parâmetros do método de Bandt e Pompe, são conhecidos como *embedding dimension* e *embedding delay*, respectivamente<sup>3</sup>.

Em seguida, para cada partição  $w_p$ , avaliamos a permutação  $\pi_p = (r_0, r_1, \dots, r_{d_x-1})$  dos índices de posição  $(0, 1, \dots, d_x - 1)$  que ordena os elementos de  $w_p$  em ordem ascendente,

<sup>1</sup>O uso de procedimentos de simbolização em sequências numéricas no contexto de sistemas dinâmicos data do fim do século XIX com os primeiros trabalhos de análise de órbitas planetárias de Henri Poincaré. Uma revisão e explanação de procedimentos de simbolização podem ser encontradas na referência [45].

<sup>2</sup>Uma série temporal é um conjunto de observações de uma variável realizadas sequencialmente no tempo [46]. Séries temporais são comuns, por exemplo, em economia (séries históricas de indicadores econômicos ou séries financeiras como a cotação diária do dólar) e meteorologia (séries de temperatura e outras séries climáticas).

<sup>3</sup>Na proposta original de Bandt e Pompe  $\tau_x$  é fixado como  $\tau_x = 1$  [6]. A formalização da entropia de permutação para outros valores de *embedding delay* foi posteriormente realizada por Cao *et al.* [47].

isto é, a permutação dos índices de posição definida pelas desigualdades  $x_{p+r_0} \leq x_{p+r_1} \leq \dots \leq x_{p+r_{d_x-1}}$ . Em caso de empates, mantemos a ordem de ocorrência dos elementos da partição<sup>4</sup>, ou seja, caso  $x_{p+r_{k-1}} = x_{p+r_k}$ , fazemos  $r_{k-1} < r_k$  para  $k = 1, \dots, d_x - 1$  [47]. Para ilustrar o significado da notação anterior, suponha que tenhamos  $x_t = (5, 3, 2, 2, 7, 9)$  e fixemos  $d_x = 4$  e  $\tau_x = 1$ . A primeira partição da série é  $w_1 = (5, 3, 2, 2)$ . Ordenando seus elementos, encontramos  $2 \leq 2 < 3 < 5$  ou  $x_{1+2} \leq x_{1+3} < x_{1+1} < x_{1+0}$ . Assim, a permutação, ou padrão ordinal, associada a  $w_1$  é  $\pi_1 = (2, 3, 1, 0)$ . De modo geral, séries temporais apresentam resolução suficiente para tornar irrelevantes problemas com igualdades entre observações consecutivas. Entretanto, esse problema pode se tornar crítico para sinais com baixa resolução [48].

Após a avaliação das permutações associadas a cada uma das partições, obtemos uma sequência de padrões ordinais (sequência simbólica)  $\{\pi_p\}_{p=1, \dots, n_x}$ . Como a inspeção dessa sequência pode ser de interesse, a função `ordinal_sequence` do pacote `ordpy` retorna essa sequência simbólica conforme ilustrado no seguinte fragmento de código:

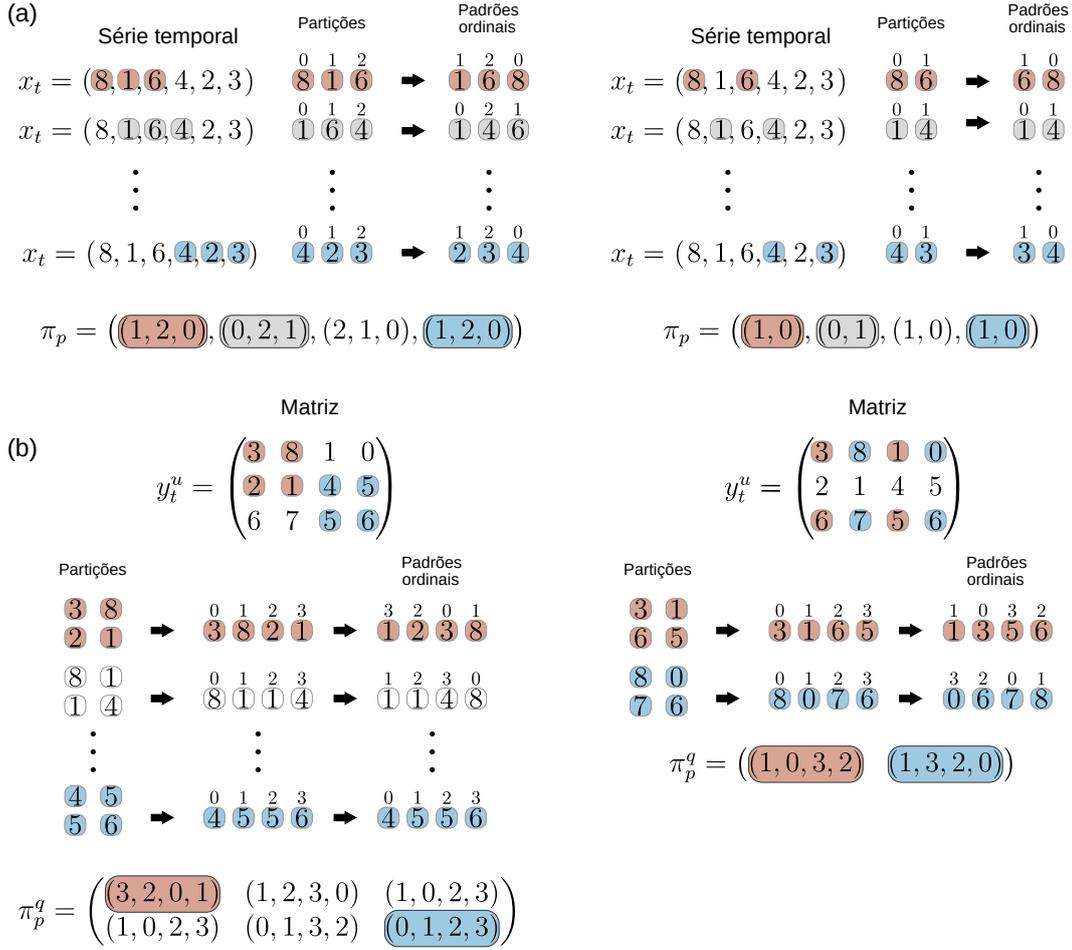
```
>>> from ordpy import ordinal_sequence
>>> x = [5, 3, 2, 2, 7, 9]
>>> ordinal_sequence(x, dx=4, tau=1)
array([[2, 3, 1, 0],
       [1, 2, 0, 3],
       [0, 1, 2, 3]])
>>> ordinal_sequence([1.55, 1.54, 1.53], dx=2)
array([[1, 0], [1, 0]])
>>> ordinal_sequence([1.55, 1.54, 1.53], dx=2, tie_precision=1)
array([[1, 0], [0, 1]])
```

Nos exemplos anteriores, os dois últimos casos ilustram o uso do parâmetro `tie_precision`, o qual pode restringir o número de casas decimais consideradas ao avaliar as relações de amplitude. Esse parâmetro está disponível na maioria das funções do `ordpy` e é particularmente relevante quando consideramos séries temporais cuja resolução é próxima da precisão estabelecida pela representação de vírgula flutuante (*floating-point representation*). Na Figura 1.1(a), a aplicação do procedimento de simbolização de Bandt e Pompe a uma série temporal é ilustrado para dois pares diferentes de valores de  $d_x$  e  $\tau_x$ .

A partir da sequência simbólica de uma série temporal podemos construir a distribuição de probabilidade de padrões ordinais  $P = \{\rho_i(\Pi_i)\}_{i=1, \dots, n_\pi}$ . Essa distribuição é simplesmente a distribuição das frequências relativas das permutações que ocorrem na sequência simbólica,

---

<sup>4</sup>Outra maneira de lidar com empates entre observações dentro de uma partição consiste em adicionar um ruído de amplitude menor que a menor resolução da série temporal. Essa estratégia, inicialmente proposta no trabalho seminal de Bandt e Pompe [6], é pouco utilizada na literatura e preterida em favor daquela empregada em nosso trabalho.



**Figura 1.1: Procedimento de simbolização de Bandt e Pompe.** (a) Ilustração do método de Bandt e Pompe aplicado a uma série temporal e a sequência ordinal resultante para os parâmetros de *embedding*  $d_x = 3$  e  $\tau_x = 1$  (esquerda) e  $d_x = 2$  e  $\tau_x = 2$  (direita). (b) Aplicação do método de Bandt e Pompe a uma matriz (dado bidimensional) e as sequências (matrizes) simbólicas resultantes para os parâmetros de *embedding*  $d_x = d_y = 2$  e  $\tau_x = \tau_y = 1$  (esquerda) e  $d_x = d_y = 2$  e  $\tau_x = \tau_y = 2$  (direita). Em ambos os painéis, diferentes cores são usadas para ressaltar a relação entre o esquema de particionamento dos dados e o mapeamento dessas partições no conjunto correspondente de permutações.

ou seja,

$$\rho_i(\Pi_i) = \frac{\text{número de partições do tipo } \Pi_i \text{ em } \{\pi_p\}}{n_x}, \quad (1.2)$$

sendo  $\Pi_i^5$  cada uma das  $n_\pi = d_x!$  permutações diferentes que podem ocorrer em  $\{\pi_p\}$ . O seguinte fragmento de código demonstra como obter uma distribuição ordinal usando a função `ordinal_distribution` contida no `ordpy`:

```
>>> from ordpy import ordinal_distribution
>>> x = [5, 3, 2, 2, 7, 9]
```

<sup>5</sup>Usamos  $\Pi$  para indicar as diferentes permutações que podem ocorrer em uma sequência simbólica, em contraposição a  $\pi$  que representa as permutações que compõem uma sequência simbólica. Por exemplo, para  $d_x = 2$  temos apenas  $\Pi_1 = (0, 1)$  e  $\Pi_2 = (1, 0)$ , a despeito da sequência simbólica  $\{\pi_p\}_{p=1, \dots, n_x}$  em questão.

```

>>> pis, rho = ordinal_distribution(x, dx=3)
>>> pis
array([[0, 1, 2],
       [1, 2, 0],
       [2, 1, 0]])
>>> rho
array([0.5 , 0.25, 0.25])

```

Os dois vetores (*arrays*) retornados por `ordinal_distribution` contém os padrões ordinais e suas frequências relativas, respectivamente. Por padrão, a função `ordinal_distribution` retorna apenas os padrões ordinais que ocorrem no dado (isto é, padrões tais que  $\rho_i(\Pi_i) \neq 0$ ). Entretanto, o parâmetro `return_missing` pode modificar esse comportamento como mostrado a seguir:

```

>>> from ordpy import ordinal_distribution
>>> x = [5, 3, 2, 2, 7, 9]
>>> pis, rho = ordinal_distribution(x, dx=3, return_missing=True)
>>> pis
array([[0, 1, 2],
       [1, 2, 0],
       [2, 1, 0],
       [0, 2, 1],
       [1, 0, 2],
       [2, 0, 1]])
>>> rho
array([0.5 , 0.25, 0.25, 0. , 0. , 0. ])

```

Permutações que não ocorrem no dado são sempre retornadas como os últimos elementos do vetor.

Uma vez de posse da distribuição ordinal  $P$ , podemos calcular sua entropia de Shannon [9] e definir a entropia de permutação como

$$S(P) = - \sum_{i=1}^{n_\pi} \rho_i(\Pi_i) \log \rho_i(\Pi_i), \quad (1.3)$$

com  $\log(\dots)$  sendo o logaritmo na base 2 e a entropia sendo medida em bits [10]. Uma interpretação da Equação 1.3 é que a entropia de permutação quantifica a aleatoriedade na distribuição de amplitudes das observações dentro das partições de uma série temporal. Desse modo, o limite superior  $S = \log n_\pi$  indica que os dados são aleatoriamente distribuídos, enquanto o limite inferior  $S = 0$  indica uma distribuição de amplitudes regular ou determinística. Ainda, considerando que o valor máximo para a entropia de Shannon é

$S_{\max} = \log n_\pi$ , podemos definir a entropia de permutação normalizada como

$$H(P) = \frac{S(P)}{\log n_\pi}, \quad (1.4)$$

restringindo os valores de  $H$  ao intervalo  $[0, 1]$ . No módulo `ordpy`, a função `permutation_entropy` calcula os valores de  $S$  e  $H$  diretamente de uma série, conforme ilustrado a seguir:

```
>>> from ordpy import permutation_entropy
>>> x = [5, 3, 2, 2, 7, 9]
>>> permutation_entropy(x)
0.5802792108518123
>>> permutation_entropy(x, normalized=False, base='e')
1.0397207708399179
```

A função `permutation_entropy` calcula a entropia de permutação como definida na Equação 1.3, utilizando o logaritmo na base 2, mas esse comportamento pode ser modificado por meio do parâmetro `base` que possibilita o uso do logaritmo natural.

Antes de propriamente aplicarmos a entropia de permutação a uma série temporal, é necessário considerarmos a escolha apropriada dos parâmetros  $d_x$  e  $\tau_x$ . Uma vez que o parâmetro de *embedding dimension* define o número de permutações possíveis ( $n_\pi = d_x!$ ), Bandt e Pompe recomendam [6] valores de  $d_x \in \{3, 4, 5, 6, 7\}$ . Esses valores comumente satisfazem a condição  $d_x! \ll N_x$ , o que permite estimativas confiáveis da distribuição de probabilidade de padrões ordinais. Outra estratégia, embora menos comum, consiste em escolher  $d_x$  tal que  $5d_x! \leq N_x$  [49]. Mais recentemente, entretanto, Cuesta-Frau *et al.* [50] mostraram que essas restrições à escolha de  $d_x$  podem ser consideravelmente relaxadas em várias situações nas quais a entropia de permutação é usada para fins de classificação de séries temporais. Quanto ao parâmetro de *embedding delay*, este define uma escala de intervalos de tempo no dado sob análise. Esse parâmetro é frequentemente escolhido como  $\tau_x = 1$ , mas diferentes valores de  $\tau_x$  podem informar acerca de mecanismos de resposta retardada e estruturas de correlação temporal [20]. Uma discussão mais detalhada sobre as escolhas de  $d_x$  e  $\tau_x$  é apresentada no Apêndice A deste trabalho.

Para além de séries temporais, o formalismo da entropia de permutação foi estendido a dados bidimensionais por Ribeiro e colaboradores [27, 28]. Para apresentar essa generalização, consideramos uma matriz arbitrária  $\{y_t^u\}_{t=1, \dots, N_x}^{u=1, \dots, N_y}$  cujos elementos podem representar os pixels de uma imagem. Definimos as *embedding dimensions*  $d_x$  e  $d_y$  correspondentes às direções horizontal e vertical (respectivamente) e, similarmente, os *embedding delays*  $\tau_x$  e  $\tau_y$ . Analogamente ao caso unidimensional, subdividimos a matriz dos dados em partições de

tamanho  $d_x \times d_y$  definidas por

$$w_p^q = \begin{pmatrix} y_p^q & y_{p+\tau_x}^q & \cdots & y_{p+(d_x-1)\tau_x}^q \\ y_p^{q+\tau_y} & y_{p+\tau_x}^{q+\tau_y} & \cdots & y_{p+(d_x-1)\tau_x}^{q+\tau_y} \\ \vdots & \vdots & \ddots & \vdots \\ y_p^{q+(d_x-1)\tau_x} & y_{p+\tau_x}^{q+(d_x-1)\tau_x} & \cdots & y_{p+(d_x-1)\tau_x}^{q+(d_y-1)\tau_y} \end{pmatrix}, \quad (1.5)$$

com os índices  $p = 1, \dots, n_x$  e  $q = 1, \dots, n_y$  e o número de partições sendo limitado por  $n_x = N_x - (d_x - 1)\tau_x$  e  $n_y = N_y - (d_y - 1)\tau_y$ . Para associarmos uma permutação a cada partição bidimensional, concatenamos as linhas das partições a fim de representá-las como um vetor unidimensional, ou seja,

$$w_p^q = \left( y_p^q, y_{p+\tau_x}^q, \dots, y_{p+(d_x-1)\tau_x}^q, \right. \\ \left. y_p^{q+\tau_y}, y_{p+\tau_x}^{q+\tau_y}, \dots, y_{p+(d_x-1)\tau_x}^{q+\tau_y}, \dots, \right. \\ \left. y_p^{q+(d_x-1)\tau_x}, y_{p+\tau_x}^{q+(d_x-1)\tau_x}, \dots, y_{p+(d_x-1)\tau_x}^{q+(d_y-1)\tau_y} \right). \quad (1.6)$$

Como o procedimento anterior independe da partição, simplificamos a notação escrevendo  $w_p^q$  como

$$w_p^q = (\tilde{y}_0, \tilde{y}_1, \dots, \tilde{y}_{d_x d_y - 2}, \tilde{y}_{d_x d_y - 1}), \quad (1.7)$$

com  $\tilde{y}_0 = y_p^q$ ,  $\tilde{y}_1 = y_{p+\tau_x}^q$  e assim por diante. Em seguida, avaliamos a permutação associada a cada partição da mesma maneira como no caso unidimensional a fim de definir uma sequência simbólica  $\{\pi_p^q\}_{p=1, \dots, n_x}^{q=1, \dots, n_y}$  correspondente ao dado [a Figura 1.1(b) ilustra esse procedimento]. A partir dessa sequência de permutações, calculamos a frequência relativa para todas as  $n_\pi = (d_x d_y)!$  permutações  $\Pi_i$  via

$$\rho_i(\Pi_i) = \frac{\text{número de partições do tipo } \Pi_i \text{ em } \{\pi_p^q\}}{n_x n_y}, \quad (1.8)$$

com  $i = 1, \dots, n_\pi$ . Novamente, a distribuição ordinal é  $P = \{\rho_i(\Pi_i)\}_{i=1, \dots, n_\pi}$ . É notável o fato de que o procedimento de ordenamento dos elementos das matrizes não é único como no caso unidimensional, visto que encontraríamos sequências simbólicas diferentes caso concatenássemos as colunas das partições ao invés de suas linhas. Entretanto, esses procedimentos diferentes de concatenação não modificam a entropia da distribuição ordinal, somente permutam as probabilidades entre esses padrões [27].

Igualmente ao caso de séries temporais, a entropia de permutação de um dado bidimensional é a entropia de Shannon da distribuição ordinal  $P = \{\rho_i(\Pi_i)\}_{i=1, \dots, n_\pi}$  e as Equações 1.3 e 1.4 permanecem válidas, salvo a diferença no número total de padrões ordinais [ $n_\pi = (d_x d_y)!$  no caso bidimensional]. Mais uma vez, os valores de  $d_x$  e  $d_y$  são usualmente restringidos pela

condição  $(d_x d_y)! \ll N_x N_y$  para se obter uma estimativa confiável da distribuição ordinal  $P$  [27, 28]. Naturalmente, a generalização da entropia de permutação para duas dimensões recupera a formulação original fixando  $d_y = \tau_y = 1$ . No pacote `ordpy`, a generalização para o caso bidimensional também está implementada nas funções `ordinal_sequence`, `ordinal_distribution` e `permutation_entropy`, como demonstrado em:

```
>>> from ordpy import ordinal_sequence, ordinal_distribution,
... permutation_entropy
>>> y = [[5, 3, 2], [2, 7, 9]]
>>> ordinal_sequence(y, dx=2, dy=2)
array([[2, 1, 0, 3],
       [1, 0, 2, 3]])
>>> ordinal_distribution(y, dx=2, dy=2)
(array([[1, 0, 2, 3],
       [2, 1, 0, 3]]), array([0.5, 0.5]))
>>> permutation_entropy(y, dx=2, dy=2)
0.21810429198553155
```

Uma outra ferramenta importante e bastante popular no formalismo da entropia de permutação é o plano complexidade-entropia proposto por Rosso *et al.* [51]. Esse método foi inicialmente introduzido com o propósito de distinguir entre séries caóticas e estocásticas e tem sido usado com sucesso em vários contextos [30, 52–61]. O plano complexidade-entropia combina a entropia de permutação normalizada  $H$  (Equação 1.4) com uma medida de complexidade estatística  $C$  (também calculada usando a distribuição ordinal) para construir um espaço de representação bidimensional com os valores de  $H$  e  $C$ . A medida de complexidade estatística  $C$  usada por Rosso *et al.* é inspirada no trabalho de Lopez-Ruiz *et al.* [62] e é definida como sendo o produto entre a entropia de permutação normalizada e a divergência normalizada de Jensen-Shannon [63] calculada entre a distribuição ordinal  $P = \{\rho_i(\Pi_i)\}_{i=1, \dots, n_\pi}$  e a distribuição uniforme  $U = \{1/n_\pi\}_{i=1, \dots, n_\pi}$ . Matematicamente, podemos escrever essa medida como

$$C(P) = \frac{D(P, U)H(P)}{D^{\max}}, \quad (1.9)$$

sendo

$$D(P, U) = S[(P + U)/2] - \frac{1}{2}S(P) - \frac{1}{2}S(U) \quad (1.10)$$

a divergência de Jensen-Shannon e

$$D^{\max} = -\frac{1}{2} \left( \frac{n_\pi! + 1}{n_\pi!} \log(n_\pi! + 1) - 2 \log(2n_\pi!) + \log n_\pi! \right)$$

uma constante de normalização. Essa última constante expressa o máximo valor de  $D(P, U)$ ,

que ocorre quando  $P = \{\delta_{1,i}\}_{i=1,\dots,n_\pi}$  [64, 65], sendo  $\delta_{ij} = \begin{cases} 1 & \text{se } i = j \\ 0 & \text{se } i \neq j \end{cases}$  a função delta de Kronecker.

Diferentemente da entropia de permutação, a complexidade estatística  $C$  é nula nas situações extremas de completo determinismo ou ordem (quando uma única permutação ocorre com probabilidade 1) e aleatoriedade ou desordem (quando todas as permutações são equiprováveis). O valor de  $C$  quantifica complexidade estrutural<sup>6</sup> e contém informação adicional não expressa no valor de  $H$ . Além disso,  $C$  não é uma função trivial de  $H$ , no sentido que distribuições de mesma entropia  $H$  podem levar a diferentes valores de  $C$  [51, 64, 65]. Isso acontece uma vez que  $H$  e  $D$  são expressas por diferentes somas de  $\rho_i(\Pi_i)$  e, portanto, não há motivos para assumir uma relação unívoca entre  $H$  e  $C$ .

Para ilustrar esse fato, substituiremos (por simplicidade) a divergência de Jensen-Shannon pela distância euclidiana entre  $P$  e  $U$  (assim como no artigo seminal de Lopez-Ruiz *et al.* [62]), isto é,  $D(P, U) = \sum_{i=1}^{n_\pi} (\rho_i(\Pi_i) - 1/n_\pi)^2$ . Nesse caso, a complexidade estatística é

$$C(P) \propto - \left( \sum_{i=1}^{n_\pi} \rho_i(\Pi_i) \log \rho_i(\Pi_i) \right) \left( \sum_{i=1}^{n_\pi} (\rho_i(\Pi_i) - 1/n_\pi)^2 \right),$$

e podemos prontamente observar que diferentes distribuições ordinais  $P = \{\rho_i(\Pi_i)\}_{i=1,\dots,n_\pi}$ , de mesmo valor de  $H$ , podem ter diferentes valores de complexidade estatística  $C$  (ou vice-versa). Para deixar esse fato ainda mais claro, consideramos uma distribuição de probabilidade de três estados<sup>7</sup>, isto é,  $P = \{a, b, 1 - (a + b)\}$  com  $a > 0$  e  $b > 0$  números reais tais que  $a + b \leq 1$  (para assegurar a normalização de  $P$ ). Nesse caso, temos

$$S = -a \log a - b \log b - [1 - (a + b)] \log[1 - (a + b)]$$

e

$$D = (a - 1/3)^2 + (b - 1/3)^2 + ([1 - (a + b)] - 1/3)^2.$$

Assim, supondo  $a = 0,79$  e  $b = 0,18$  ou  $a = 0,80$  e  $b = 0,16$  encontramos o mesmo valor de  $H = S/\log 3 \approx 0,55$ , mas valores diferentes para  $D$  (0,32 no primeiro caso e 0,33 no segundo) e, conseqüentemente, para  $C$ .

No pacote `ordpy`, a função `complexity_entropy` retorna simultaneamente os valores de  $H$  e  $C$  de uma série temporal:

```
>>> from ordpy import complexity_entropy
>>> complexity_entropy([4, 7, 9, 10, 6, 11, 3], dx=2)
```

<sup>6</sup>Por complexidade estrutural nos referimos à “complexidade das relações entre os componentes de um sistema” [66].

<sup>7</sup>Essa situação seria equivalente a uma distribuição ordinal com  $d_x! = 3$  ou  $(d_x d_y)! = 3$ , caso isso fosse possível.

```
(0.9182958340544894, 0.06112816548804511)
```

A generalização do procedimento de simbolização de Bandt e Pompe também permite a aplicação do plano complexidade-entropia na discriminação de dados bidimensionais [27,28]. Sendo assim, a função `complexity_entropy` também é capaz de calcular a complexidade e entropia de um dado em forma de matriz:

```
>>> from ordpy import complexity_entropy
>>> complexity_entropy([[1,2,1],[8,3,4],[6,7,5]], dx=2, dy=2)
(0.3271564379782973, 0.2701200547320647)
```

### 1.3 Aplicações do formalismo de Bandt e Pompe usando **ordpy**

Feitas as definições necessárias relacionadas ao formalismo da entropia de permutação, passamos a apresentar aplicações práticas das funções contidas no `ordpy`, replicando alguns resultados importantes encontrados na literatura.

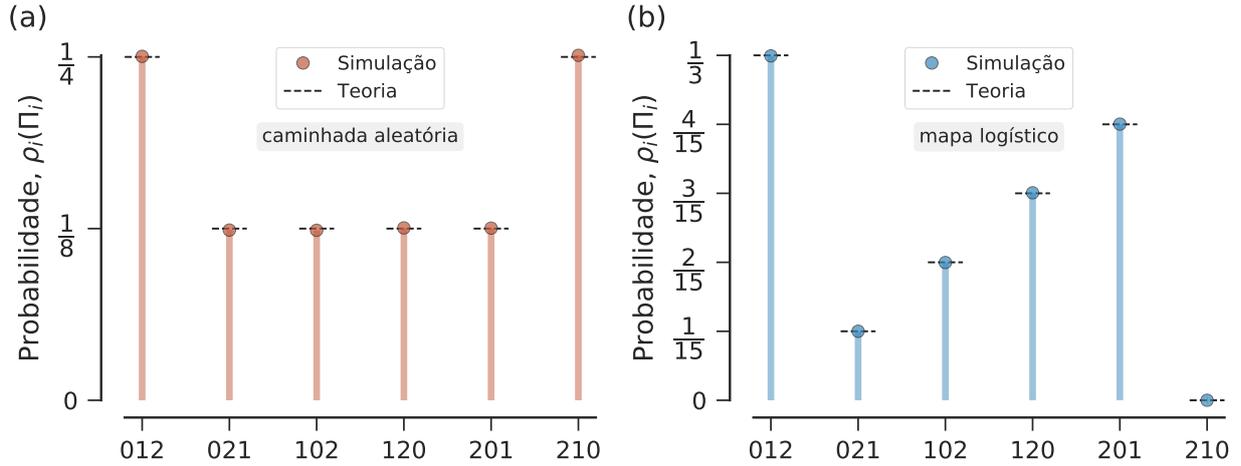
Começamos determinando as distribuições de probabilidade de padrões ordinais (permutações) de duas séries temporais, uma de natureza caótica e outra estocástica, a saber, uma caminhada aleatória de passos gaussianos e o mapa logístico<sup>8</sup> em seu estado de caos completamente desenvolvido. Esses dois exemplos foram escolhidos pois as distribuições ordinais teóricas são conhecidas para alguns valores de  $d_x$  e  $\tau_x$ . Especificamente, para  $d_x = 3$  e  $\tau_x = 1$ , as distribuições de probabilidade associadas às permutações  $\{(0, 1, 2), (0, 2, 1), (1, 0, 2), (1, 2, 0), (2, 0, 1), (2, 1, 0)\}$  são  $P_{\text{caminhada}} = \{1/4, 1/8, 1/8, 1/8, 1/8, 1/4\}$  e  $P_{\text{logístico}} = \{1/3, 1/15, 2/15, 3/15, 4/15, 0\}$  para a caminhada aleatória [67] e o mapa logístico no estado de caos completo [68], respectivamente.

Para estimar numericamente as distribuições ordinais desses dois processos, geramos computacionalmente uma caminhada aleatória gaussiana e iteramos o mapa logístico. Para ambas as séries, simulamos  $10^6$  observações e estimamos suas distribuições ordinais usando a função `ordinal_distribution`. A Figura 1.2 mostra que nossos resultados estão em excelente acordo com as distribuições ordinais teoricamente esperadas em cada caso. É intrigante observar que o padrão ordinal  $(2, 1, 0)$  (permutação descendente) não ocorre na série logística (tem probabilidade zero). Esse fato é entendido como uma característica diretamente ligada à dinâmica determinística do mapa [68, 69]. Como discutiremos na próxima seção, investigações acerca desses “padrões ordinais faltantes” são úteis na caracterização da dinâmica de uma série temporal.

Continuando as aplicações, ilustramos o uso da função `permutation_entropy` reproduzindo parcialmente uma análise do mapa logístico devida a Bandt e Pompe (Figura 2

---

<sup>8</sup>Esse mapa é definido no Apêndice B.

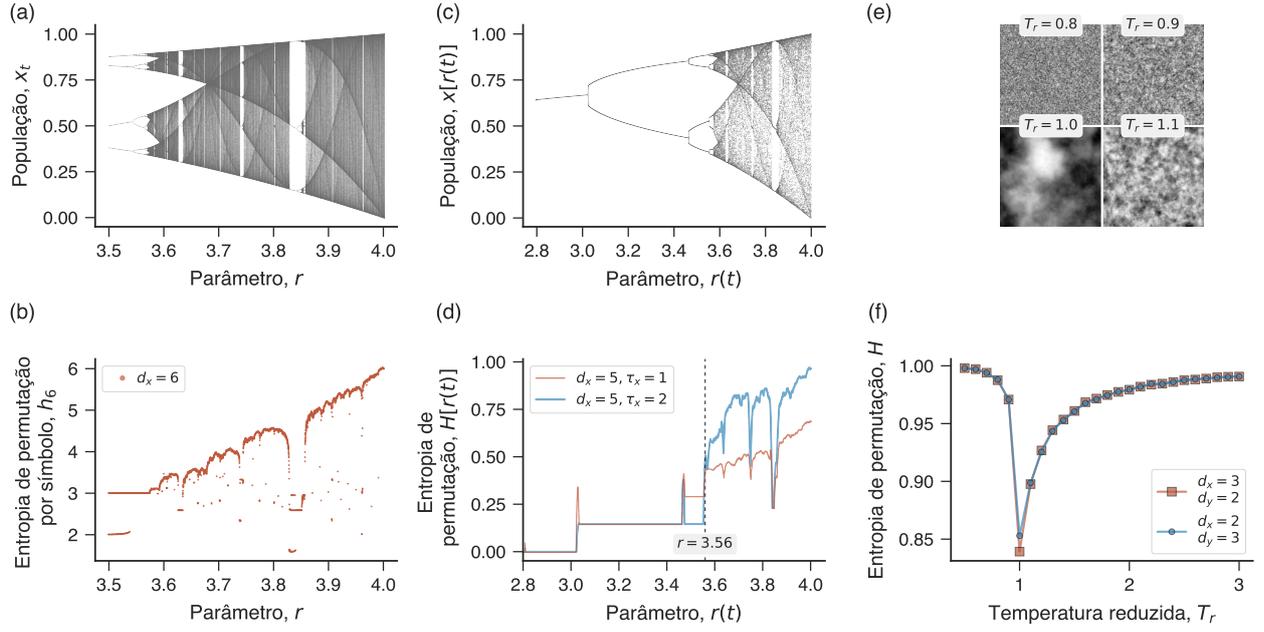


**Figura 1.2: Distribuições de probabilidade de padrões ordinais em séries estocásticas e determinísticas.** (a) Comparação entre uma distribuição ordinal empírica obtida de uma caminhada aleatória gaussiana simulada com  $10^6$  passos e a distribuição teórica  $P_{\text{caminhada}}$  (linhas horizontais tracejadas) quando  $d_x = 3$  e  $\tau_x = 1$ . (b) Comparação entre uma distribuição ordinal empírica obtida após  $10^6$  iterações do mapa logístico no regime completamente caótico e a distribuição teórica  $P_{\text{logístico}}$  (linhas horizontais tracejadas) para  $d_x = 3$  e  $\tau_x = 1$ .

na referência [6]). Para tanto, simulamos séries temporais compostas de  $10^6$  iterações do mapa logístico para cada valor do parâmetro  $r \in \{3,5; 3,5001; 3,5002; \dots; 4,0\}$ . Em seguida, calculamos a entropia de permutação  $S$  para cada uma dessas 5001 séries usando a função `permutation_entropy` com os parâmetros  $d_x = 6$  e  $\tau_x = 1$ . Ainda, dividimos o resultado da entropia de permutação por 5 para obter a entropia de permutação por símbolo de ordem 6, isto é,  $h_6 = S/5$ , conforme definido no trabalho de Bandt e Pompe [6]. A Figura 1.3(a) mostra o conhecido diagrama de bifurcação do mapa logístico, enquanto a Figura 1.3(b) mostra os valores de  $h_6$  em função do parâmetro  $r$ . Notamos que a entropia de permutação por símbolo apresenta uma tendência geral de crescimento em função do parâmetro  $r$  marcada por quedas abruptas em intervalos de  $r$  relacionados a comportamentos periódicos do mapa. Conforme apontado por Bandt e Pompe, o comportamento da entropia de permutação é similar àquele observado para o expoente de Lyapunov [6].

Em outro exemplo com a função `permutation_entropy`, replicamos um experimento numérico de Cao *et al.* [47] (veja a Figura 1 dessa referência) que procura por mudanças na dinâmica de uma série temporal do mapa logístico transiente<sup>9</sup> e demonstra o papel do parâmetro *embedding delay*  $\tau_x$  na detecção dessas mudanças. Seguindo o trabalho original, iteramos o mapa logístico transiente começando com  $x_0 = 0,65$  e  $r_0 = 2,8$  e incrementamos o parâmetro logístico  $r(t)$  em  $10^{-5}$  (a cada iteração) até atingirmos  $r(t) = 4$ . Esse processo gera uma série temporal com 120001 observações que é mostrada na Figura 1.3(c). Usando essa série temporal, calculamos a entropia de permutação normalizada dentro de janelas com

<sup>9</sup>Esse mapa é definido no Apêndice B.



**Figura 1.3: Entropia de permutação de dados unidimensionais e bidimensionais.** (a) Diagrama de bifurcação do mapa logístico para o parâmetro  $r \in \{3,5; 3,5001; 3,5002; \dots; 4,0\}$ . (b) Entropia de permutação por símbolo de ordem seis ( $h_6$ ) calculada para as mesmas séries temporais do mapa logístico ( $10^6$  observações, condição inicial aleatória) usadas na confecção do diagrama de bifurcação. Os parâmetros de *embedding* são  $d_x = 6$  e  $\tau_x = 1$ . (c) Série temporal do mapa logístico transiente obtida das condições iniciais  $x_0 = 0,65$  e  $r_0 = 2,8$  e incrementando o parâmetro  $r(t)$  em  $10^{-5}$  a cada iteração. Apesar de parecer bastante similar ao diagrama de bifurcação, esse painel mostra uma série temporal na qual cada observação  $x[r(t)]$  corresponde a um valor  $r(t)$ . (d) Dependência da entropia de permutação normalizada  $H[r(t)]$  (avaliada em janelas deslizantes de 1024 observações) com o parâmetro  $r(t)$  (ao fim de cada janela). As diferentes curvas mostram os resultados para  $H[r(t)]$  com  $d_x = 5$  e  $\tau_x = 1$  (em vermelho) e  $d_x = 5$  e  $\tau_x = 2$  (em azul). A linha vertical tracejada em  $r = 3,56$  indica uma dobra de período da série. (e) Superfícies de Ising obtidas após  $10^6$  passos de Monte Carlo com temperaturas reduzidas  $T_r \in \{0,8; 0,9; 1,0; 1,1\}$ . Nessas superfícies, tons escuros de cinza indicam sítios de alta amplitude enquanto tons claros de cinza indicam o oposto. (f) Entropia de permutação normalizada em função da temperatura reduzida  $T_r \in \{0,5; 0,6; \dots; 3,0\}$  para superfícies de Ising (tamanho  $250 \times 250$ ) obtidas após  $10^6$  passos de Monte Carlo. As curvas mostram os resultados para parâmetros de *embedding*  $d_x = 3, d_y = 2$  (em vermelho) e  $d_x = 2, d_y = 3$  (em azul) com  $\tau_x = \tau_y = 1$  em ambos os casos.

1024 observações com  $d_x = 5$  e dois valores de *embedding delay*,  $\tau_x = 1$  e  $\tau_x = 2$ .

Assim como Cao *et al.* [47], denotamos os valores da entropia de permutação como  $H[r(t)]$ , com  $r(t)$  representando o parâmetro logístico no fim da janela de 1024 observações. A Figura 1.3(d) mostra os valores de  $H[r(t)]$  que, conjuntamente com a Figura 1.3(c), indicam claramente que oscilações abruptas em  $H[r(t)]$  estão associadas a mudanças dinâmicas na série temporal. Apesar da similaridade dos resultados, notamos que  $H[r(t)]$  calculada com

o parâmetro  $\tau_x = 2$  identifica melhor essas mudanças dinâmicas do que no caso em que  $\tau_x = 1$ . Isso é evidenciado, por exemplo, pela dobra de período que ocorre em  $r \approx 3,56$  que é capturada apenas para  $\tau_x = 2$  [47].

Como mencionamos, uma generalização da entropia de permutação para imagens foi proposta por Ribeiro *et al.* [27]. Para ilustrar o uso da função `permutation_entropy` com esse tipo de dado, replicamos um experimento numérico com superfícies de Ising<sup>10</sup> (Figura 8 na referência [27]). Essas superfícies representam a soma acumulada de variáveis de *spin* do modelo canônico de Ising em duas dimensões em uma simulação de Monte Carlo. A Figura 1.3 mostra quatro exemplos dessas superfícies (redes quadradas de tamanho  $250 \times 250$ ) obtidas após  $10^6$  passos de Monte Carlo para diferentes temperaturas reduzidas  $T_r$ .

Notamos a emergência de padrões não triviais quando a temperatura reduzida é igual à temperatura crítica ( $T_r = 1$ ) para o modelo de Ising [70]. A fim de replicar os resultados de Ribeiro *et al.*, geramos uma superfície de Ising para cada temperatura reduzida  $T_r \in \{0,5; 0,6; \dots; 3,0\}$  e calculamos suas entropias de permutação normalizadas usando os parâmetros  $d_x = 3, d_y = 2$  e  $d_x = 2, d_y = 3$ , com  $\tau_x = \tau_y = 1$  em ambos os casos. Em acordo com Ribeiro *et al.* [27], a Figura 1.3(f) mostra que a entropia de permutação identifica precisamente a transição de fase do modelo de Ising (a repentina diminuição da entropia quando o sistema se encontra próximo à temperatura crítica) e que essas superfícies de Ising são simétricas sob uma permutação de *embedding dimensions*.

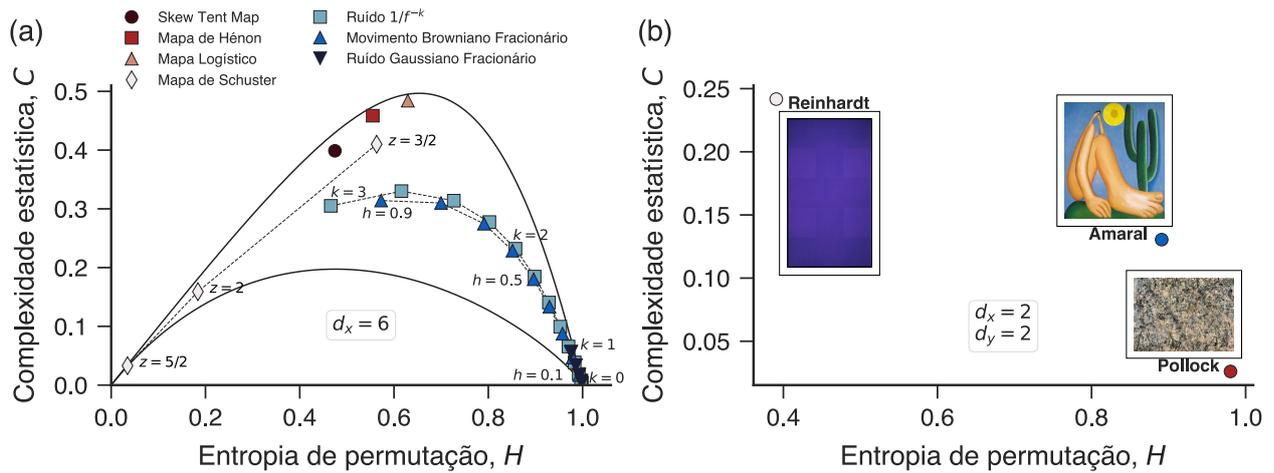
A função `complexity_entropy` calcula simultaneamente a entropia de permutação normalizada e a complexidade estatística para séries temporais ou imagens. Para ilustrar seu uso, reproduzimos parcialmente os resultados de Rosso *et al.* [51] (Figura 1 nessa referência) relacionados à distinção entre séries temporais caóticas e estocásticas. Seguindo esse artigo, iteramos quatro mapas discretos para gerar séries caóticas<sup>11</sup>. Especificamente, utilizamos séries caóticas do *skew tent map* (parâmetro  $w = 0,1847$ ), mapa de Hénon (componente  $x$ , parâmetros  $a = 1,4$  e  $b = 0,3$ ), mapa logístico ( $r = 4$ ) e mapa de Schuster (parâmetro  $z \in \{3/2, 2, 5/2\}$ ). Usamos também séries temporais de três processos estocásticos<sup>12</sup>: ruído com espectro de potência  $1/f^k$  (para  $k \in \{0,00; 0,25; \dots; 3,00\}$ ), movimento browniano fracionário (expoente de Hurst  $h \in \{0,1; 0,2; \dots; 0,9\}$ ) e ruído fracionário gaussiano (também com  $h \in \{0,1; 0,2; \dots; 0,9\}$ ). Para cada um desses mapas e processos estocásticos, geramos 10 séries temporais com  $2^{15}$  observações e condições iniciais aleatórias. Usamos a função `complexity_entropy` com os parâmetros de *embedding*  $d_x = 6$  e  $\tau_x = 1$  para calcular suas complexidades estatísticas e entropias de permutação (média sobre as 10 amostras em cada caso).

A Figura 1.4(a) mostra que séries caóticas usualmente apresentam alta complexidade estatística e baixa entropia enquanto séries estocásticas apresentam alta entropia e com-

<sup>10</sup>A definição desse processo estocástico é apresentada no Apêndice C.

<sup>11</sup>As definições desses mapas são apresentadas no Apêndice B.

<sup>12</sup>As definições desses processos estocásticos são apresentadas no Apêndice C



**Figura 1.4: Plano complexidade-entropia para dados uni e bidimensionais.** (a) Valores médios de complexidade estatística  $C$  e entropia de permutação normalizada  $H$  para dez realizações de séries temporais de mapas caóticos e processos estocásticos. Os parâmetros de *embedding* são  $d_x = 6$  e  $\tau_x = 1$ . Duas linhas sólidas representam os valores máximo e mínimo de complexidade para uma dada entropia. (b) Localização de três pinturas no plano complexidade-entropia com parâmetros de *embedding*  $d_x = d_y = 2$  e  $\tau_x = \tau_y = 1$ .

plexidade intermediária ou baixa. É interessante notar que as séries temporais estocásticas aproximam-se do canto inferior direito do plano complexidade-entropia ( $H \rightarrow 1$  e  $C \rightarrow 0$ ) conforme a autocorrelação das séries diminui [51]. Esses resultados mostram ainda que algumas séries estocásticas e caóticas têm valores próximos de entropia mas complexidades estatísticas diferentes (o movimento browniano fracionário com  $h = 0,9$  e o mapa de Schuster com  $z = 3/2$ , por exemplo), confirmando que a complexidade estatística captura informação complementar à entropia de uma distribuição ordinal. Ressaltamos que nem todos os valores do plano complexidade-entropia são acessíveis, uma vez que distribuições de probabilidade com uma dada entropia restringem os valores da complexidade estatística a um intervalo determinado [65]. Essa região acessível é delimitada por duas linhas sólidas na Figura 1.4. As funções `maximum_complexity_entropy` e `minimum_complexity_entropy`, implementadas no `ordpy`, geram os pontos que compõem essas curvas como mostrado a seguir:

```
>>> from ordpy import
... maximum_complexity_entropy,
... minimum_complexity_entropy
>>> maximum_complexity_entropy(dx=4)
array([[ -0.          ,  -0.          ],
       [ 0.21810429,  0.19670592],
       [ 0.34568712,  0.28362016],
       ...,
       [ 0.98660828,  0.02388382]])
```

```
>>> minimum_complexity_entropy(dx=4)
array([[ -0.00000000e+00,  -0.00000000e+00],
       [ 2.67076969e-02,  2.55212327e-02],
       ...,
       [ 1.00000000e+00,  -3.66606083e-16]])
```

Como dissemos, a função `complexity_entropy` também é capaz de atuar em dados bidimensionais. Para ilustrar esse fato, seguimos Sigaki *et al.* [29] e investigamos padrões em pinturas de diferentes movimentos artísticos. Devido à imensa quantidade de dados investigada por esses autores, não reproduzimos nenhum resultado do trabalho e apenas usamos suas ideias para ilustrar como complexidade e entropia extraem informação dessas imagens. Para tanto, escolhemos três pinturas do site [wikiart.org](http://wikiart.org) (no estudo original, os autores estudaram 137364 imagens obtidas do mesmo site). Essas pinturas são: Blue, 1953 por Ad Reinhardt (768 × 435 pixels), pertencente ao movimento de arte abstrata Campo de Cor [71]; Abaporu, 1928 por Tarsila do Amaral (1200 × 1026 pixels), pintura do Modernismo brasileiro [72]; Number 1, 1950 (Lavender Mist), 1950 por Jackson Pollock (749 × 1024 pixels) [73], uma pintura do movimento artístico chamado Expressionismo Abstrato. As três imagens estão em formato JPEG com 24 bits por pixel (8 bits para vermelho, verde e azul no sistema de cores RGB<sup>13</sup>). Sendo assim, calculamos a média das intensidades dos pixels nessas três camadas de modo a representar cada imagem como uma matriz simples. A partir dessas matrizes, calculamos a complexidade estatística e a entropia de permutação de cada uma das três obras de arte usando os parâmetros de *embedding*  $d_x = d_y = 2$  e  $\tau_x = \tau_y = 1$ .

A Figura 1.4(b) mostra a localização dessas imagens no plano complexidade-entropia (inserções retratam as obras de arte). Em acordo com as tendências globais observadas por Sigaki *et al.* [29], esses resultados mostram que pinturas retratando objetos com bordas claramente definidas (os quadrados na obra de Reinhardt) tendem a apresentar valores altos de complexidade estatística e baixa entropia. Em outro extremo, pinturas com contornos borrados e difusos (tais como as *drip paintings* de Pollock) têm alta entropia e baixa complexidade. Entre esses comportamentos opostos, temos todo um contínuo de imagens, como exemplificado aqui pelo trabalho da pintora brasileira Tarsila do Amaral. Como defendido por Sigaki *et al.* [29], o plano complexidade-entropia mapeia o grau de ordem local (entre pixels vizinhos) das pinturas em uma escala de ordem-desordem e simplicidade-complexidade que é similar a descrições qualitativas de pinturas propostas por historiadores da arte como Heinrich Wölfflin (a dicotomia linear versus pictórico) e Alois Riegl (a dicotomia háptico versus óptico).

---

<sup>13</sup>No sistema de cores RGB, uma imagem é representada por três matrizes, uma para cada cor primária, que são combinadas em uma tela. Para cada cor e em todos os pontos da imagem, a intensidade é determinada por um número inteiro entre 0 e 255 (8 bits de informação para cada cor).

## 1.4 Padrões ordinais faltantes

Como comentamos quando da análise da distribuição de probabilidade de padrões ordinais do mapa logístico completamente caótico, esse sistema não realiza a “permutação descendente”  $(2, 1, 0)$  quando  $d_x = 3$  [vide a Figura 1.2(b)]. Essa característica não é uma particularidade apenas do mapa logístico. Na verdade, esses padrões ordinais faltantes (ou padrões ordinais proibidos<sup>14</sup>) ocorrem em diferentes sistemas e estatísticas simples associadas a eles têm se mostrado indicadores úteis e confiáveis da dinâmica de um sistema [12, 74–77].

Essa linha de investigação de padrões ordinais é devida ao trabalho seminal de Amigó *et al.* [68, 69] e, seguindo seu esquema de classificação, podemos dividir esses padrões proibidos em duas categorias: verdadeiros ou falsos [69]. Padrões verdadeiramente proibidos [como  $(2, 1, 0)$  no caso do mapa logístico] são uma marca de determinismo na dinâmica de uma série temporal e representam uma característica intrínseca do processo dinâmico subjacente à série [68], ou seja, esses padrões não são um artefato decorrente do tamanho finito de uma série de observações [69]. Ao contrário, padrões falsamente proibidos estão relacionados ao tamanho finito de séries temporais e podem ocorrer mesmo em processos completamente aleatórios [69].

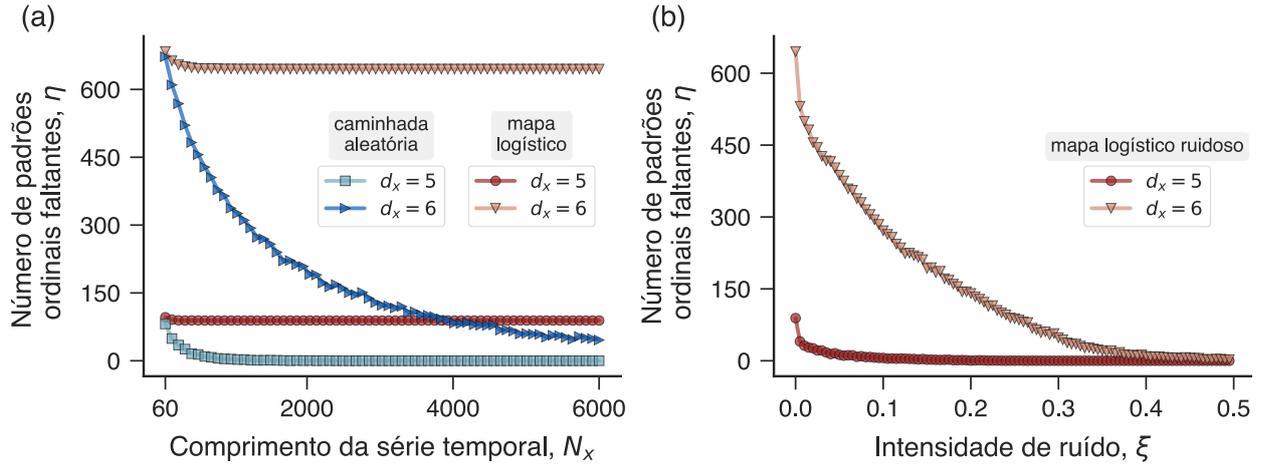
Essa distinção não é tão clara quando lidamos com dados empíricos e uma análise típica consiste em investigar o número de padrões faltantes ( $\eta$ ) em função do tamanho da série temporal ( $N_x$ ). O comportamento dessa curva é útil para discriminar séries temporais. No pacote `ordpy`, a função `missing_patterns` identifica padrões ordinais faltantes e estima sua frequência relativa como em:

```
>>> from ordpy import missing_patterns
>>> missing_patterns([4, 7, 9, 10, 6, 11, 3, 5, 6, 2, 3, 1], dx=3)
(array([[0, 2, 1],
        [2, 1, 0]]),
 0.3333333333333333)
```

Para ilustrar melhor o uso dessa função, investigamos padrões ordinais faltantes em séries temporais obtidas do mapa logístico completamente caótico ( $r = 4$ ) e caminhadas aleatórias gaussianas. Em ambos os casos, usamos *embedding dimensions*  $d_x = 5$  e  $d_x = 6$  (com  $\tau_x = 1$ ) e tamanhos de séries  $N_x \in \{60, 150, 240, \dots, 6000\}$ . A Figura 1.5(a) mostra os resultados. Observamos que o número de padrões faltantes aproxima-se de zero para as caminhadas aleatórias mais longas. Contrariamente, o número de permutações faltantes apresenta uma queda inicial (mais facilmente perceptível para  $d_x = 6$ ) com o tamanho das séries do mapa logístico, mas estabiliza-se em valores consideravelmente altos. Esse resultado reafirma a proposição de que os padrões faltantes estão intrinsecamente associados ao determinismo do mapa [69].

---

<sup>14</sup>Em inglês, *missing ordinal patterns* ou *forbidden ordinal patterns*.



**Figura 1.5: Padrões ordinais faltantes em séries temporais.** (a) Número de padrões ordinais faltantes ( $\eta$ ) em caminhadas aleatórias (em azul) e séries do mapa logístico (em vermelho) como função do tamanho das sequências ( $N_x$ ) para os parâmetros de *embedding*  $d_x = 5$  e  $d_x = 6$ , ambos com  $\tau_x = 1$ . Os resultados representam o número médio de padrões ordinais faltantes para 10 réplicas de séries temporais para cada  $N_x \in \{60, 150, 240, \dots, 6000\}$ . (b) Dependência do número de padrões ordinais faltantes com a intensidade de ruído ( $\xi$ ) para séries logísticas com 6000 observações. O ruído adicionado às séries logísticas é uniformemente distribuído no intervalo  $[-\xi, \xi]$  com  $\xi \in \{0; 0,005; 0,010; \dots; 0,500\}$ . Os resultados representam médias calculadas para dez séries temporais para cada nível de ruído. As *embedding dimensions* são indicadas na figura e o *embedding delay* é fixado em  $\tau_x = 1$ . Usamos condições iniciais aleatórias e fixamos o parâmetro  $r = 4$  em todas as simulações com o mapa logístico que levaram aos resultados apresentados nessa figura.

Em outra aplicação com a função `missing_patterns`, replicamos um resultado de Amigó *et al.* [69] (Figura 4 dessa referência) para, mais uma vez, evidenciar que o número de padrões faltantes é um bom indicador de determinismo em séries temporais [49,69]. Seguindo o trabalho original, geramos séries temporais do mapa logístico (6000 iterações,  $r = 4$ ) e adicionamos ruído uniformemente distribuído no intervalo  $[-\xi, \xi]$ , sendo  $\xi$  um parâmetro que controla a amplitude do ruído. Em seguida, estimamos o número médio de padrões faltantes (em 10 réplicas) para cada nível de ruído  $\xi \in \{0; 0,005; 0,010; \dots; 0,500\}$  e *embedding dimensions*  $d_x = 5$  e  $d_x = 6$  (com  $\tau_x = 1$ ). A Figura 1.5(b) mostra o número de padrões ordinais faltantes em função da amplitude do ruído  $\xi$  para ambas as *embedding dimensions*. Observamos que o número de padrões ordinais faltantes relacionados a essas séries determinísticas contaminadas com ruído aproxima-se de zero com o aumento da amplitude do ruído. Entretanto, é necessário um nível significativo de ruído ( $\approx 50\%$  da amplitude do sinal original<sup>15</sup> para  $d_x = 6$ ) para remover qualquer sinal de determinismo capturado pela ausência de permutações [69].

<sup>15</sup>Os valores da série do mapa logístico estão restritos ao intervalo  $[0, 1]$ .

## 1.5 Entropias de permutação de Tsallis e Rényi

Além da entropia de permutação e da complexidade estatística, pesquisadores propuseram o uso de outras estatísticas juntamente à distribuição ordinal [16, 17, 78, 79]. Assim como verificamos explicitamente para a complexidade estatística, esses diferentes quantificadores devem extrair informação adicional acerca da dinâmica de uma série temporal não capturada pelas métricas já utilizadas. Nesse contexto, uma abordagem produtiva passa por considerar generalizações paramétricas da entropia de Shannon, tais como aquelas propostas por Tsallis [80] e Rényi [81]. O trabalho de Zunino *et al.* [16] foi o primeiro a considerar a entropia de Tsallis como substituta da entropia de Shannon e, assim, definir a entropia de permutação de Tsallis como

$$S_\beta(P) = \frac{1}{\beta - 1} \sum_{i=1}^{n_\pi} (\rho_i(\Pi_i) - [\rho_i(\Pi_i)]^\beta), \quad (1.11)$$

sendo  $\beta$  um número real (no limite em que  $\beta \rightarrow 1$  recuperamos a entropia de Shannon e a entropia de permutação usual).

A entropia de Tsallis tem seu máximo para a distribuição uniforme, de modo que  $S_\beta^{\max} = \frac{1 - (n_\pi)^{1-\beta}}{\beta-1}$ . Assim, a entropia de permutação de Tsallis normalizada é

$$H_\beta(P) = (\beta - 1) \frac{S_\beta(P)}{1 - (n_\pi)^{1-\beta}}. \quad (1.12)$$

Similarmente, Liang *et al.* [17] propuseram a entropia de permutação de Rényi

$$S_\alpha(P) = \frac{1}{1 - \alpha} \ln \left( \sum_{i=1}^{n_\pi} [\rho_i(\Pi_i)]^\alpha \right), \quad (1.13)$$

sendo  $\alpha > 0$  um número real. A entropia de Rényi converge para a entropia de Shannon quando  $\alpha \rightarrow 1$  e também é maximizada pela distribuição uniforme ( $S_\alpha^{\max} = \ln n_\pi$ , assim como a entropia de Shannon). Logo, a entropia de permutação de Rényi normalizada é

$$H_\alpha(P) = \frac{S_\alpha(P)}{\ln n_\pi}. \quad (1.14)$$

Em ambos os casos, a forma entrópica generalizada é monoparamétrica e possui um termo no qual as probabilidades dos padrões ordinais são expostas a uma operação de potenciação, na qual o expoente é o parâmetro entrópico (isto é,  $\rho_i^\beta$  e  $\rho_i^\alpha$ ). Esses parâmetros atribuem diferentes pesos às probabilidades dos padrões ordinais permitindo-nos ressaltar diferentes escalas de grandeza dentre as probabilidades e produzir um conjunto de quantificadores (variando esses parâmetros) dada uma distribuição ordinal. No módulo `ordpy`, as funções `tsallis_entropy` e `renyi_entropy` implementam essas duas alternativas à entropia

de permutação usual:

```
>>> from ordpy import tsallis_entropy, renyi_entropy
>>> tsallis_entropy([4,7,9,10,6,11,3], q=[1,2], dx=2)
... #O parâmetro beta é representado por q na função.
array([0.91829583, 0.88888889])
>>> renyi_entropy([4,7,9,10,6,11,3], alpha=[1,2], dx=2)
array([0.91829583, 0.84799691])
```

Em um contexto similar, temos a proposta das chamadas curvas complexidade-entropia por Ribeiro *et al.* [82] e Jauregui *et al.* [83]. Esses trabalhos estenderam o conceito do plano complexidade-entropia ao considerar as entropias de Tsallis e Rényi combinadas às correspondentes generalizações de complexidade estatística [65]. Assim, ao invés de localizar uma série temporal por um ponto em um plano complexidade-entropia, Ribeiro *et al.* [82] e Jauregui *et al.* [83] criaram curvas paramétricas variando os parâmetros entrópicos ( $\beta$  ou  $\alpha$ ) e simultaneamente calculando as entropias e complexidades generalizadas.

Para definir as curvas complexidade-entropia de Tsallis [82], estendemos a medida de complexidade estatística (Equação 1.9) usando a entropia de Tsallis:

$$C_\beta(P) = \frac{D_\beta(P, U)H_\beta(P)}{D_\beta^{\max}}, \quad (1.15)$$

com

$$D_\beta(P, U) = \frac{1}{2}K_\beta\left(P\left|\frac{P+U}{2}\right.\right) + \frac{1}{2}K_\beta\left(U\left|\frac{P+U}{2}\right.\right) \quad (1.16)$$

representando a divergência de Jensen-Tsallis [65] escrita em termos da divergência de Kullback-Leibler [65, 84] correspondente,

$$K_\beta(V|R) = \frac{1}{\beta-1} \sum_i^{n_\pi} v_i^\beta [r_i^{1-\beta} - v_i^{1-\beta}], \quad (1.17)$$

na qual  $V = \{v_i\}_{i=1, \dots, n_\pi}$  e  $R = \{r_i\}_{i=1, \dots, n_\pi}$  são duas distribuições de probabilidade arbitrárias. Na Equação 1.15,

$$D_\beta^{\max} = \frac{2^{2-\beta}n_\pi - (1+n_\pi)^{1-\beta} - n_\pi(1+1/n_\pi)^{1-\beta} - n_\pi + 1}{2^{2-\beta}n_\pi(1-\beta)}$$

é uma constante de normalização igual ao valor máximo de  $D_\beta(P, U)$ , o qual ocorre quando  $P = \{\delta_{1,i}\}_{i=1, \dots, n_\pi}$  (assim como para a divergência de Jensen-Shannon). Seguindo Ribeiro *et al.* [82], construímos uma representação paramétrica de pares ordenados  $(H_\beta(P), C_\beta(P))$  com  $\beta > 0$  para obter as curvas complexidade-entropia de Tsallis.

Similarmente, para definir as curvas complexidade-entropia de Rényi [83], generalizamos

a complexidade estatística no formalismo da entropia de Rényi como

$$C_\alpha(P, U) = \frac{D_\alpha(P, U)H_\alpha(P)}{D_\alpha^{\max}}, \quad (1.18)$$

com

$$D_\alpha(P, U) = \frac{1}{2}K_\alpha\left(P\left|\frac{P+U}{2}\right.\right) + \frac{1}{2}K_\alpha\left(U\left|\frac{P+U}{2}\right.\right) \quad (1.19)$$

representando a divergência de Jensen-Rényi [65] escrita em termos da divergência de Kullback-Leibler [65, 85] correspondente:

$$K_\alpha(V|R) = \frac{1}{\alpha - 1} \ln \left( \sum_{i=1}^{d!} v_i^\alpha r_i^{1-\alpha} \right). \quad (1.20)$$

A constante de normalização

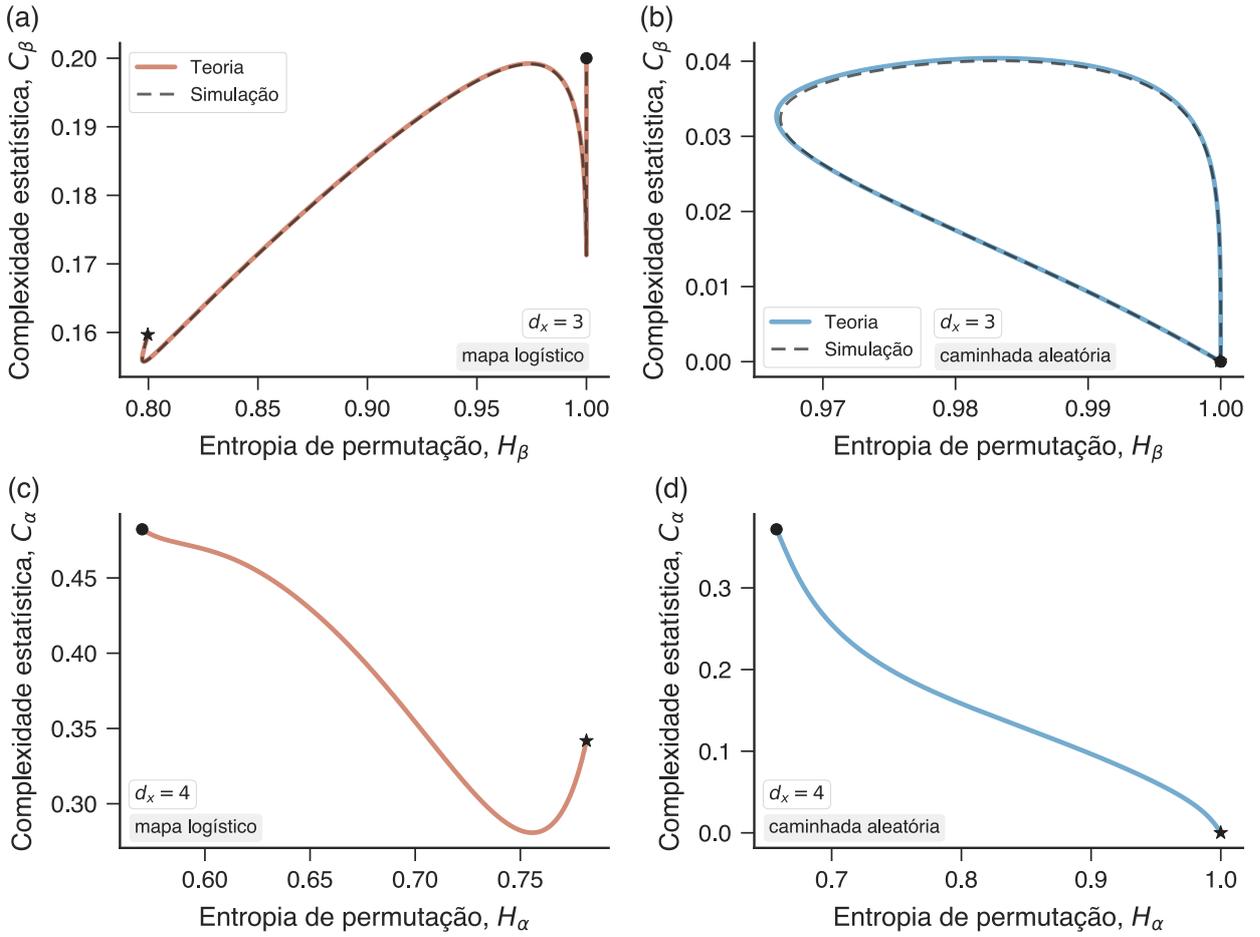
$$D_\alpha^{\max} = \frac{1}{2(\alpha - 1)} \ln \left[ \frac{(n_\pi + 1)^{1-\alpha} + n_\pi - 1}{n_\pi} \left( \frac{n_\pi + 1}{4n_\pi} \right)^{1-\alpha} \right],$$

mais uma vez, corresponde ao valor máximo de  $D_\alpha(P, U)$ , que ocorre quando  $P = \{\delta_{1,i}\}_{i=1,\dots,n_\pi}$ . Sendo assim, construímos uma representação paramétrica dos pares ordenados  $(H_\alpha(P), C_\alpha(P))$  para  $\alpha > 0$  que dá origem às curvas complexidade-entropia de Rényi propostas por Jauregui *et al.* [83].

As funções `tsallis_complexity_entropy` e `renyi_complexity_entropy` implementam as curvas complexidade-entropia de Tsallis e Rényi, como mostra o seguinte fragmento de código:

```
>>> from ordpy import tsallis_complexity_entropy,
... renyi_complexity_entropy
>>> tsallis_complexity_entropy([4, 7, 9, 10, 6, 11, 3], dx=2, q=[1, 2])
... #0 parâmetro beta é representado por q na função.
array([[0.91829583, 0.06112817],
       [0.88888889, 0.07619048]])
>>> renyi_complexity_entropy([4, 7, 9, 10, 6, 11, 3], dx=2, alpha=[1, 2])
array([[0.91829583, 0.06112817],
       [0.84799691, 0.08303895]])
```

Para ilustrar o uso dessas funções, replicamos alguns experimentos numéricos envolvendo o mapa logístico e caminhadas aleatórias contidos nos trabalhos de Ribeiro *et al.* [82] (Figuras 1 e 6 nessa referência) e Jauregui *et al.* [83] (Figuras 1 e 3 desse trabalho). Começamos iterando o mapa logístico ( $r = 4$ , condições iniciais aleatórias) e simulando uma caminhada aleatória de passos gaussianos. Para a série do mapa logístico descartamos as primeiras  $10^4$  iterações para evitar efeitos transientes e iteramos outras  $10^6$  vezes. A caminhada aleató-



**Figura 1.6: Curvas complexidade-entropia de Tsallis e Rényi.** Curvas complexidade-entropia de Tsallis para séries temporais obtidas (a) do mapa logístico ( $r = 4$ ) e (b) de uma caminhada aleatória de passos gaussianos. Em ambos os casos usamos os parâmetros de *embedding*  $d_x = 3$  e  $\tau_x = 1$ . As linhas sólidas representam os resultados empíricos e as linhas tracejadas indicam as formas exatas dessas curvas complexidade-entropia. Os painéis (c) e (d) mostram as curvas complexidade-entropia de Rényi obtidas a partir das mesmas séries temporais com parâmetros de *embedding*  $d_x = 4$  e  $\tau_x = 1$ . Em todos os painéis, marcadores em forma de estrela indicam o começo das curvas ( $\beta \approx 0$  ou  $\alpha \approx 0$ ), enquanto marcadores circulares indicam o fim das curvas (maiores valores de  $\beta$  e  $\alpha$ ).

ria também contém  $10^6$  observações. Usando essas séries temporais, traçamos suas curvas complexidade-entropia de Tsallis para  $d_x = 3$  e  $\tau_x = 1$ , amostrando 1000 valores (logaritmicamente espaçados) para o parâmetro entrópico  $\beta$  no intervalo  $[0,01, 100]$  para o mapa logístico e entre  $[0,001, 100]$  para a caminhada aleatória.

As Figuras 1.6(a) e 1.6(b) mostram as curvas complexidade-entropia empíricas em comparação com a forma exata esperada. Essas curvas teóricas podem ser determinadas para essas séries temporais porque as distribuições ordinais do mapa logístico ( $P_{\text{logístico}} = \{1/3, 1/15, 2/15, 3/15, 4/15, 0\}$ ) e da caminhada aleatória ( $P_{\text{walk}} = \{1/4, 1/8, 1/8, 1/8, 1/8, 1/4\}$ ) são exatamente conhecidas para  $d_x = 3$  [67, 68]. Observamos que os resultados empíricos estão em

excelente acordo com a teoria. Conforme discutido por Ribeiro *et al.* [82], séries aleatórias tendem a formar curvas complexidade-entropia fechadas [Figura 1.6(b)], enquanto séries caóticas são usualmente representadas por curvas complexidade-entropia abertas [Figura 1.6(a)]. Essas características emergem como consequência direta da existência (ou não) de padrões ordinais faltantes capturados pelo comportamento limite de  $H_\beta$  quando  $\beta \rightarrow 0$  e  $\beta \rightarrow \infty$  [82].

Similarmente, também estimamos as curvas complexidade-entropia de Rényi para as mesmas duas séries temporais usando  $d_x = 4$  e  $\tau_x = 1$ . As Figuras 1.6(c) e 1.6(d) mostram essas curvas. Diferentemente das curvas complexidade-entropia de Tsallis, curvas complexidade-entropia de Rényi são sempre abertas [83] e o uso dessas curvas na distinção entre séries caóticas e estocásticas depende de uma característica mais sutil. Jauregui *et al.* [83] sugeriram que a curvatura inicial das curvas complexidade-entropia de Rényi ( $dC_\alpha/dH_\alpha$  para  $\alpha$  pequeno) pode ser utilizada como um indicador de determinismo em séries temporais. Especificamente, eles encontraram que curvaturas positivas estão associadas a séries temporais de natureza estocástica, enquanto curvaturas negativas estão associadas a fenômenos caóticos. Esse padrão também está presente nos resultados das Figuras 1.6(c) e 1.6(d).

## 1.6 Redes ordinais

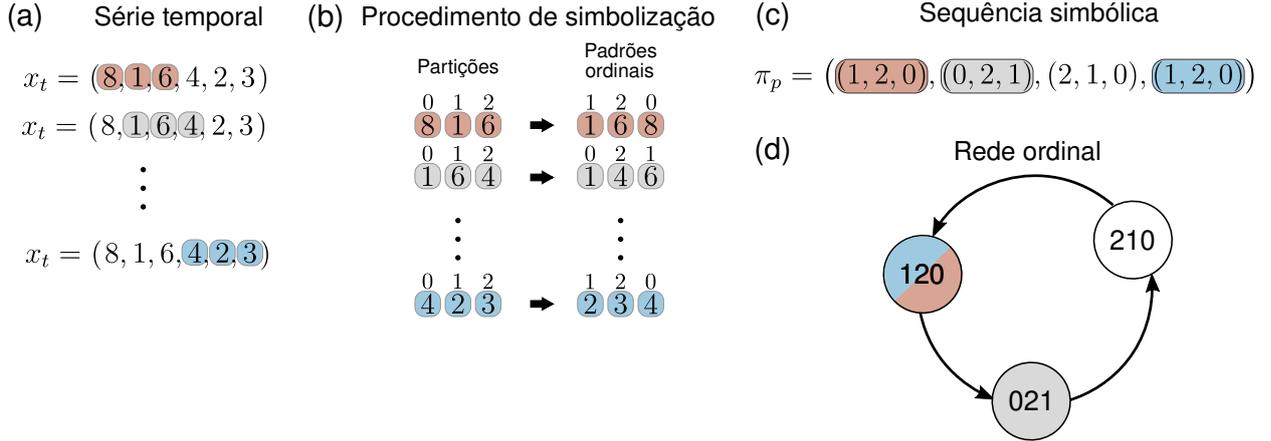
As redes ordinais estão entre os desenvolvimentos mais recentes do formalismo de Bandt e Pompe. Essas redes foram primeiramente propostas por Small [31] com o objetivo de investigar séries temporais de sistemas dinâmicos [86–89]. As redes ordinais pertencem a uma classe mais geral de métodos, coletivamente conhecidos como redes de séries temporais [34], que tem como objetivo principal mapear séries temporais em redes complexas. Para além da descrição estatística de padrões ordinais, essa abordagem considera a vizinhança de cada símbolo ordinal pertencente a uma sequência simbólica. Na representação de rede ordinal, permutações diferentes de uma série são mapeadas em nós de uma rede. As arestas entre esses nós indicam permutações adjacentes na sequência simbólica. Essas arestas podem ainda ser direcionadas, de acordo com a sucessão temporal de padrões ordinais, e pesadas pelas frequências relativas de ocorrência dessas sucessões em uma sequência simbólica [86].

Após a aplicação do método de Bandt e Pompe com parâmetros de *embedding*  $d_x$  e  $\tau_x$  a uma série temporal  $\{x_t\}_{t=1,\dots,N_x}$  e a consequente obtenção de uma sequência simbólica  $\{\pi_p\}_{p=1,\dots,n_x}$ , podemos definir os elementos da matriz de adjacência<sup>16</sup> pesada que define a rede ordinal como [32, 86]

$$\rho_{i,j} = \frac{\text{número de transições } \Pi_i \rightarrow \Pi_j \text{ em } \{\pi_p\}_{p=1,\dots,n_x}}{n_x - 1}, \quad (1.21)$$

---

<sup>16</sup>Considerando um grafo direcionado e pesado, os elementos  $\rho_{i,j}$  de sua matriz de adjacência são números reais que indicam uma ligação direcionada do  $i$ -ésimo para o  $j$ -ésimo vértice do grafo [90].



**Figura 1.7: Mapeamento de séries temporais em redes ordinais.** (a) O particionamento de uma série temporal  $\{x_t\}_{t=1, \dots, N_x}$  de tamanho  $N_x = 6$  é mostrado. (b) Ilustração do procedimento de simbolização para os parâmetros de *embedding*  $d_x = 3$  e  $\tau_x = 1$ . (c) Sequência simbólica  $\{\pi_p\}_{p=1, \dots, n_x}$  ( $n_x = 4$ ) resultante da simbolização da série temporal. (d) Representação da série temporal em (a) como uma rede ordinal. Nesse exemplo, todas as sucessões entre permutações ocorrem apenas uma vez, de modo que todas as arestas da rede têm o mesmo peso.

com  $i, j = 1, 2, \dots, n_\pi$  ( $n_\pi = d_x!$ ),  $\Pi_i$  e  $\Pi_j$  representando padrões ordinais distintos e o denominador  $n_x - 1$  indicando o número de transições ordinais ocorrendo em  $\{\pi_p\}_{p=1, \dots, n_x}$ . A Figura 1.7 ilustra o processo de construção de uma rede ordinal a partir de uma série temporal. No `ordpy`, a função `ordinal_network` retorna os nós, arestas e pesos de uma rede ordinal mapeada de uma série temporal:

```

>>> from ordpy import ordinal_network
>>> ordinal_network([4, 7, 9, 10, 6, 11, 8, 3, 7], dx=2, normalized=False)
(array(['0|1', '1|0'], dtype='<U3'),
 array([[ '0|1', '0|1'],
        [ '0|1', '1|0'],
        [ '1|0', '0|1'],
        [ '1|0', '1|0']], dtype='<U3'),
 array([2, 2, 2, 1]))
  
```

Ressaltamos que o algoritmo de redes ordinais originalmente proposto por Small [31] utiliza de uma abordagem ligeiramente diferente do procedimento de simbolização de Bandt e Pompe para a obtenção da sequência simbólica. Ao invés de definir partições de dados que se sobrepõem (Equação 1.1), Small avalia os padrões ordinais em partições não sobrepostas de tamanho  $d_x$  (o parâmetro de *embedding delay* também não está presente no algoritmo original). Além disso, as arestas são não direcionadas e não pesadas nessa primeira formulação. Apesar dessa implementação não ser tão comum quanto aquelas que seguem mais

diretamente o formalismo de Bandt e Pompe [86–89], o parâmetro `overlapping` da função `ordinal_network` torna possível a construção de redes ordinais que seguem a abordagem inicial de Small:

```
>>> from ordpy import ordinal_network
>>> ordinal_network([4, 7, 9, 10, 6, 11, 8, 3, 7], dx=2, normalized=False,
... overlapping=False)
(array(['0|1', '1|0'], dtype='<U3'),
array(['0|1', '0|1'], ['0|1', '1|0']),
dtype='<U3'), array([2, 1]))
```

Nesta tese, propomos uma generalização do método de redes ordinais, tornando-as capazes de mapear dados bidimensionais. O Capítulo 2 apresenta essa generalização [33] conjuntamente a uma série de aplicações que demonstram seu uso em imagens. Entretanto, para manter a exposição do pacote `ordpy` autocontida, iremos agora apresentar os aspectos formais dessa técnica e também as funções do `ordpy` que implementam esse método. Para tanto, iniciamos com a aplicação da prescrição bidimensional do procedimento de simbolização de Bandt e Pompe [27] (veja as Equações 1.5, 1.6 e 1.7) a uma matriz  $\{y_t^u\}_{t=1, \dots, N_x}^{u=1, \dots, N_y}$  definindo as *embedding dimensions* ( $d_x$  e  $d_y$ ) e *embedding delays* ( $\tau_x$  e  $\tau_y$ ) a fim de obter uma sequência ordinal  $\{\pi_p^q\}_{p=1, \dots, n_x}^{q=1, \dots, n_y}$ .

Similarmente ao caso unidimensional, cada permutação  $\Pi_i$  [ $i = 1, \dots, n_\pi$ , com  $n_\pi = (d_x d_y)!$ ] está associada a um nó da rede ordinal e arestas direcionadas conectam nós (permutações) que são verticalmente ( $\pi_p^q \rightarrow \pi_p^{q+1}$  para  $q = 1, \dots, n_y - 1$ ) ou horizontalmente ( $\pi_p^q \rightarrow \pi_{p+1}^q$  para  $p = 1, \dots, n_x - 1$ ) adjacentes na sequência simbólica. Uma aresta direcionada entre um par de permutações  $\Pi_i$  e  $\Pi_j$  é pesada pela probabilidade de ocorrência dessa transição particular. Sendo assim, os elementos da matriz de adjacência pesada que definem a rede ordinal mapeada a partir de um dado bidimensional são dados por [33]

$$\rho_{i,j} = \frac{\text{número de transições } \Pi_i \rightarrow \Pi_j \text{ em } \{\pi_p^q\}_{p=1, \dots, n_x}^{q=1, \dots, n_y}}{2n_x n_y - n_x - n_y}, \quad (1.22)$$

com  $i, j = 1, \dots, n_\pi$  e o denominador representando o número total de transições horizontais e verticais. A Figura 1.8 ilustra o processo de construção de uma rede ordinal a partir de dados bidimensionais. A função `ordinal_network`, contida no módulo `ordpy`, implementa esse algoritmo de mapeamento de dados bidimensionais em redes como em:

```
>>> from ordpy import ordinal_network
>>> ordinal_network([[1, 2, 1], [8, 3, 4], [6, 7, 5]], dx=2, dy=2,
... normalized=True)
(array(['0|1|3|2', '1|0|2|3', '1|2|3|0'],
dtype='<U7'),
```

```
array([[ '0|1|3|2', '1|0|2|3'],
       [ '0|1|3|2', '1|2|3|0'],
       [ '1|0|2|3', '0|1|3|2'],
       [ '1|2|3|0', '0|1|3|2']], dtype='<U7'),
array([0.25, 0.25, 0.25, 0.25]))
```

Além do mapeamento anterior, também é possível construir redes ordinais que consideram apenas transições horizontais (redes ordinais horizontais) ou verticais (redes ordinais verticais) entre permutações na sequência simbólica. Conforme veremos no Capítulo 2, propriedades dessas duas redes são úteis para explorar simetrias em imagens. Utilizando o pacote `ordpy`, essas redes podem ser obtidas alterando o parâmetro `connections` na função `ordinal_network`:

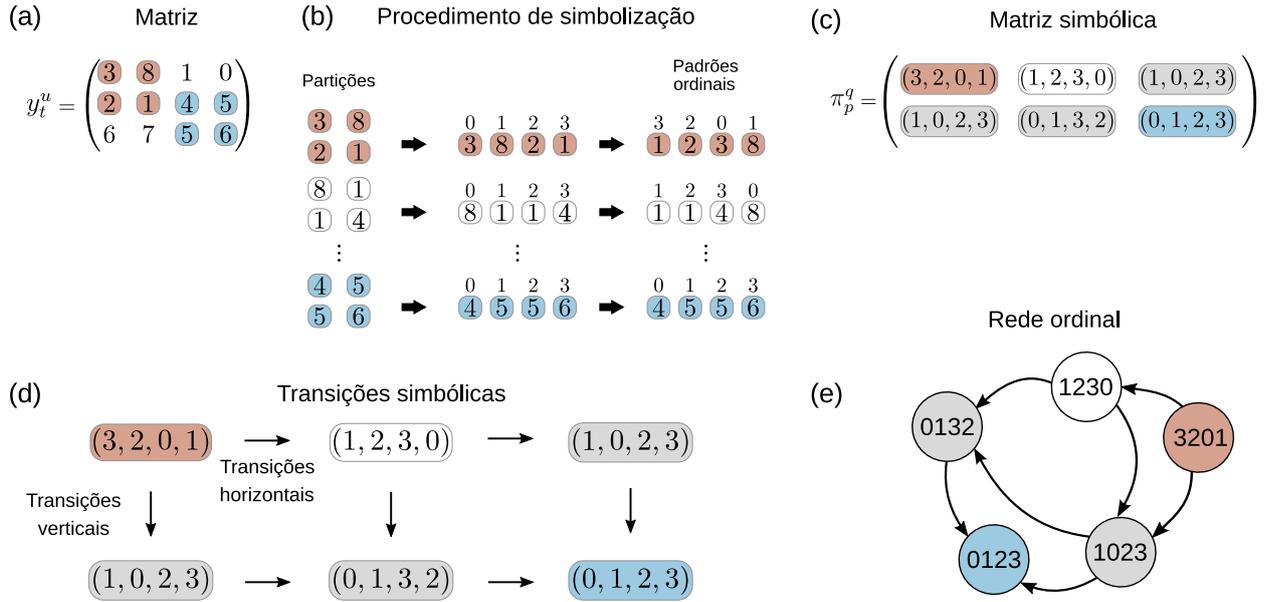
```
>>> from ordpy import ordinal_network
>>> ordinal_network([[1,2,1],[8,3,4],[6,7,5]], dx=2, dy=2,
... normalized=False, connections='horizontal')
(array([ '0|1|3|2', '1|0|2|3', '1|2|3|0'],
dtype='<U7'),
array([[ '0|1|3|2', '1|0|2|3'],
       [ '1|2|3|0', '0|1|3|2']], dtype='<U7'),
array([1, 1]))
```

Uma característica intrigante das redes ordinais é a existência de restrições intrínsecas de conectividade entre os nós dessas redes [32,33], consequência do procedimento de simbolização de Bandt e Pompe. Esses vínculos estão diretamente relacionados ao fato de que partições adjacentes compartilham observações (ou pixels), de modo que relações de amplitude em uma dada partição influenciam parcialmente partições adjacentes. Para séries temporais, essas restrições implicam que todos os nós em uma rede ordinal têm graus de saída e entrada<sup>17</sup> limitados a números entre 0 e  $d_x$ . Consequentemente, o número máximo de arestas possíveis nesse tipo de rede ordinal é  $d_x \times (d_x!)$  [32]. Ainda mais, redes ordinais mapeadas de séries temporais podem apresentar autoarestas apenas em nós associados a padrões ordinais exclusivamente ascendentes [por exemplo, (0,1), (0,1,2) e (0,1,2,3)] ou exclusivamente descendentes [por exemplo, (1,0), (2,1,0), (3,2,1,0)] [32].

Uma consequência direta dessas restrições de conectividade é que as redes ordinais mapeadas de dados aleatórios (em uma ou duas dimensões) não formam redes aleatórias ou grafos completos [32,33]. Ainda mais contraintuitiva é a existência de diferentes pesos nessas redes ordinais, apesar de todas as permutações serem equiprováveis para esse tipo de dado [32,33]. Essa propriedade não trivial é resultado do fato de que entre todas as possíveis relações de amplitude envolvidas em uma transição ordinal entre uma dada permutação e todas as suas

---

<sup>17</sup>Em inglês, *in-degree* e *out-degree*.



**Figura 1.8: Mapeamento de dados bidimensionais em redes ordinais.** (a) O particionamento de uma matriz de dados  $\{y_t^u\}_{t=1, \dots, N_x}^{u=1, \dots, N_y}$  de tamanho  $N_x = 3$  e  $N_y = 4$  é mostrado. (b) Ilustração do procedimento de simbolização para os parâmetros de *embedding*  $d_x = d_y = 2$  e  $\tau_x = \tau_y = 1$ . (c) Matriz simbólica  $\{\pi_p^q\}_{p=1, \dots, n_x}^{q=1, \dots, n_y}$  ( $n_x = 3$  e  $n_y = 2$ ) resultante da simbolização da matriz de dados. (d) Transições (sucessões verticais e horizontais) entre padrões ordinais que ocorrem na matriz simbólica. (e) Representação da matriz de dados em (a) como uma rede ordinal. Nesse exemplo, todas as sucessões entre permutações ocorrem apenas uma vez, de modo que todas as arestas da rede têm o mesmo peso.

possíveis permutações vizinhas, algumas transições são mais prováveis que outras. Para o caso unidimensional, as arestas de redes ordinais aleatórias apresentam apenas dois pesos distintos:  $1/(d_x + 1)!$  e  $2/(d_x + 1)!$  (o denominador representa a soma dos pesos) [32]. Uma regra geral para determinar as arestas com maior peso é escolher todas as transições nas quais o índice de posição igual a “ $d_x - 1$ ” na próxima permutação encontra-se na mesma posição do índice “0” na primeira permutação [32]. Por exemplo, considerando uma rede ordinal mapeada de um dado aleatório ( $d_x = 4$ ), o peso da aresta entre as permutações  $(3, 2, 1, 0)$  e  $(2, 1, 0, 3)$  deve ser maior que qualquer uma das outras três ligações que partem de  $(3, 2, 1, 0)$ . Redes ordinais mapeadas de dados bidimensionais apresentam mais de dois pesos diferentes, mas não existe uma regra simples para encontrar esses pesos [33]. Entretanto, seus valores podem ser calculados numericamente [33], como veremos no Capítulo 2.

As transições verticais e horizontais em redes ordinais mapeadas a partir de dados bidimensionais impõem restrições de conectividade similares, porém mais desafiadoras [33] (como também discutiremos no Capítulo 2). Nesse caso, os números máximos de ligações são  $C(d_x d_y, d_y) \times d_y!$  e  $C(d_x d_y, d_x) \times d_x!$  ao considerarmos, respectivamente, apenas transições horizontais e apenas transições verticais, sendo  $C(n, s)$  o número de formas distintas de escolhermos  $s$  elementos de um conjunto de  $n$  elementos. Entretanto, os conjuntos de transi-

ções horizontais e verticais não são disjuntos e sua união define todas as arestas que podem emanar de um nó. Encontrar uma expressão geral para essa última operação entre conjuntos mostra-se uma tarefa bastante complicada, uma vez que ela depende especificamente do padrão ordinal sob análise. Assim, apesar de limitado, o número máximo de arestas varia entre os padrões ordinais e precisa ser obtido numericamente [33]. Além disso, diferentemente do caso unidimensional, redes ordinais mapeadas de dados bidimensionais podem apresentar autoarestas em vários nós [33].

No módulo `ordpy`, a função `random_ordinal_network` gera a forma exata das redes ordinais esperadas do mapeamento de dados aleatórios unidimensionais e bidimensionais com *embedding dimensions* arbitrárias  $d_x$  e  $d_y$  (forma esperada do mapeamento de um dado aleatório suficientemente grande). O seguinte fragmento de código ilustra o uso dessa função:

```
>>> from ordpy import random_ordinal_network
>>> random_ordinal_network(dx=2)
(array(['0|1', '1|0'], dtype='<U3'),
 array([[ '0|1', '0|1'],
        [ '0|1', '1|0'],
        [ '1|0', '0|1'],
        [ '1|0', '1|0']], dtype='<U3'),
 array([0.16666667, 0.33333333,
        0.33333333, 0.16666667]))
```

Os três vetores retornados representam os nós, arestas e pesos da rede ordinal, respectivamente. Vale a pena notar que as restrições de conectividade desaparecem quando consideramos as partições não sobrepostas da proposta inicial de Small [31]. Nesse caso, redes ordinais mapeadas de dados aleatórios suficientemente grandes são representadas por grafos nos quais todos os nós conectam-se entre si, além de conectarem-se a si mesmos com autoarestas, com os pesos de todas as arestas apresentando o mesmo valor. A função `random_ordinal_network` retorna esses grafos se alterarmos o parâmetro `overlapping`:

```
>>> from ordpy import random_ordinal_network
>>> random_ordinal_network(dx=2,
... overlapping=False)
(array(['0|1', '1|0'], dtype='<U3'),
 array([[ '0|1', '0|1'],
        [ '0|1', '1|0'],
        [ '1|0', '0|1'],
        [ '1|0', '1|0']], dtype='<U3'),
 array([0.25, 0.25, 0.25, 0.25]))
```

Similarmente, *embedding delays* maiores que um modificam a maneira como dados são compartilhados entre partições adjacentes, impondo outros tipos de restrições. Por conta

disso, a função `random_ordinal_network` está restrita ao caso em que  $\tau_x = \tau_y = 1$  e as partições do dado se sobrepõem.

O principal motivo para o mapeamento de séries temporais ou imagens em redes ordinais é o uso de métricas de redes complexas na caracterização de conjuntos de dados. Além das muitas medidas originadas na ciência de redes [90], a natureza probabilística dos nós e arestas em redes ordinais motivou duas novas medidas entrópicas [32, 79, 87]. A primeira é uma medida local, definida ao nível dos nós, conhecida como entropia local do vértice [32, 33, 79, 87]

$$s_i = - \sum_{j \in \mathcal{O}_i} \rho'_{i,j} \log \rho'_{i,j}, \quad (1.23)$$

com o índice  $i$  referindo-se a um nó relacionado a uma dada permutação  $\Pi_i$ ,  $\rho'_{i,j} = \rho_{i,j} / \sum_{k \in \mathcal{O}_i} \rho_{i,k}$  representando a probabilidade de transição renormalizada do nó  $i$  ao nó  $j$  (permutações  $\Pi_i$  e  $\Pi_j$ ) e  $\mathcal{O}_i$  a vizinhança de saída do nó  $i$  (conjunto de todas as arestas direcionadas de  $i$  para outros nós). Essa quantidade mede o determinismo das transições ordinais ao nível do nó, de modo que  $s_i$  é máximo quando todas as arestas saindo de  $i$  têm o mesmo peso, enquanto  $s_i = 0$  caso exista apenas uma aresta saindo do nó  $i$ . Usando a entropia local do vértice, podemos ainda definir a entropia global dos vértices [32, 33, 79, 87]

$$S_G = \sum_{i=1}^{n_\pi} \rho_i s_i, \quad (1.24)$$

com  $\rho_i$  sendo a probabilidade de encontrar a permutação  $\Pi_i$  (Equações 1.2 e 1.8). Assim,  $S_G$  representa uma média ponderada do determinismo local envolvendo todos os vértices de uma rede ordinal<sup>18</sup>.

Contrariamente à entropia de permutação [6], e em razão das restrições intrínsecas de conectividade, a entropia global dos vértices não é máxima para redes ordinais mapeadas de dados aleatórios [32, 33]. Para dados unidimensionais completamente aleatórios, a entropia global dos vértices calculada de uma sequência suficientemente longa é [32]

$$S_G^{\text{aleatório}} = \log(d_x + 1) - (\log 4)/(d_x + 1). \quad (1.25)$$

Apesar de não haver expressão equivalente para dados bidimensionais, é possível calcular  $S_G^{\text{aleatório}}$  usando redes ordinais aleatórias geradas numericamente [33]. Sendo assim, a entropia global dos vértices pode sempre ser normalizada por  $S_G^{\text{aleatório}}$ , isto é,  $H_G = S_G / S_G^{\text{aleatório}}$ .

No `ordpy`, a função `global_node_entropy` avalia  $S_G$  diretamente de uma série temporal ou imagem ou, ainda, de uma rede ordinal retornada por `ordinal_network`. O seguinte fragmento de código mostra alguns exemplos simples do uso da função `global_node_entropy`:

---

<sup>18</sup>Veja também Unakafov e Keller [91] para a definição da entropia condicional de padrões ordinais.

```

>>> from ordpy import global_node_entropy
>>> global_node_entropy([1,2,3,4,5,6,7,8,9], dx=2)
0.0
>>> global_node_entropy(ordinal_network([1,2,3,4,5,6,7,8,9], dx=2))
0.0
>>> global_node_entropy(np.random.uniform(size=100000), dx=3)
1.4988332319747597
>>> global_node_entropy(random_ordinal_network(dx=3))
1.5

```

## 1.7 Aplicações de redes ordinais usando ordpy

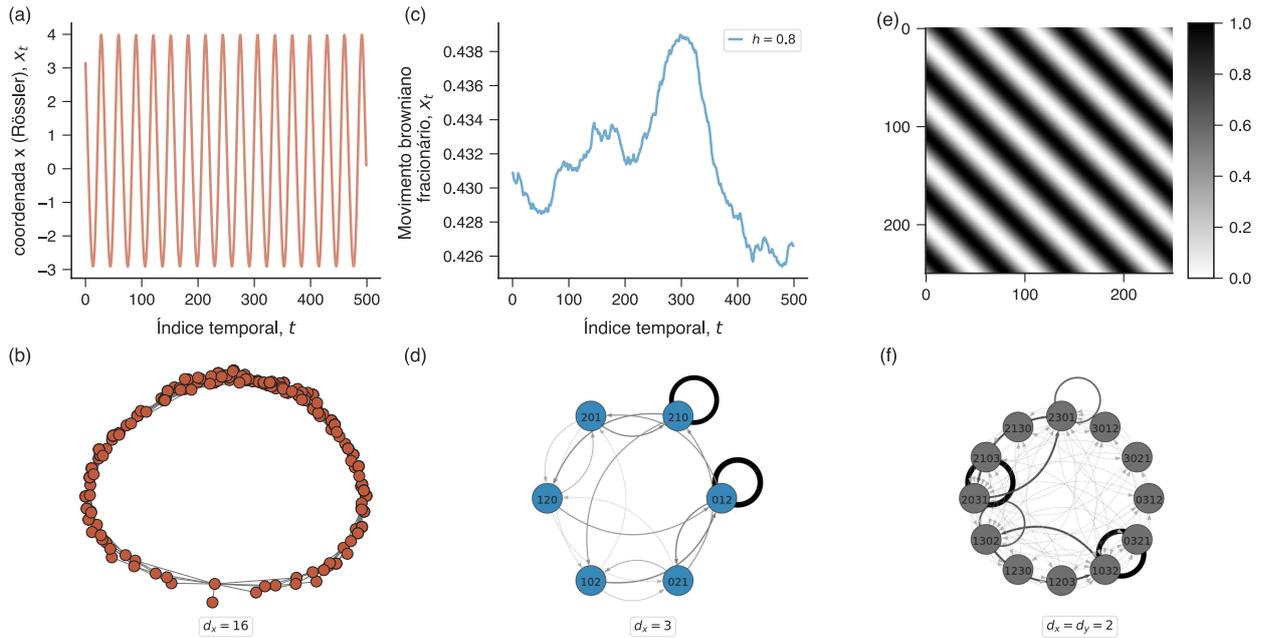
Para ilustrar o uso do `ordpy` no contexto de redes ordinais, revisamos e replicamos alguns resultados encontrados na literatura. Antes de começar, entretanto, ressaltamos que o pacote `ordpy` não contém funções para análise de redes ou visualização das mesmas. A função `ordinal_network` apenas gera dados de saída (nós, arestas e listas de pesos) que podem ser usados juntamente a pacotes Python especializados em grafos, tais como `graph-tool` [92], `networkx` [93] e `igraph` [94]. Nesta tese, utilizamos os pacotes `networkx` e `igraph`.

Começamos reproduzindo parcialmente o trabalho pioneiro de Small [31] no qual as “redes de partição ordinal” surgiram pela primeira vez (Figura 3 nessa referência). Seguindo Small [31], resolvemos numericamente as equações diferenciais que constituem o sistema de Rössler<sup>19</sup> (com parâmetros  $a = 0,3$ ,  $b = 2$  e  $c = 4$ ) e amostramos a coordenada  $x$  para obter uma série temporal com  $10^5$  observações. A Figura 1.9(a) ilustra o comportamento periódico dessa sequência. Em seguida, mapeamos esse dado em uma rede ordinal com parâmetros de *embedding*  $d_x = 16$  e  $\tau_x = 1$ . Recordamos que o algoritmo de Small utiliza de partições não sobrepostas e que as arestas da rede ordinal resultante não são direcionadas ou pesadas. O parâmetro `overlapping` em `ordinal_network` deve ser definido como `False` para o uso apropriado do algoritmo de Small. A Figura 1.9(b) mostra uma visualização dessa rede ordinal, na qual a estrutura circular alude à periodicidade da série temporal original.

Em outro exemplo com redes ordinais, replicamos parcialmente resultados obtidos por Pessa e Ribeiro [32] usando amostras de movimento browniano fracionário (Figura 6 nessa referência). Para tanto, geramos uma série temporal desse processo estocástico com expoente de Hurst  $h = 0,8$  e  $2^{16}$  observações, conforme ilustrado na Figura 1.9(c). Em seguida, mapeamos essa série temporal em uma rede ordinal com parâmetros de *embedding*  $d_x = 3$  e  $\tau_x = 1$  (agora com partições sobrepostas como no procedimento usual de Bandt e Pompe). A Figura 1.9(d) mostra uma visualização da rede ordinal resultante, na qual o comportamento

---

<sup>19</sup>Esse sistema dinâmico é definido no Apêndice B.



**Figura 1.9: Redes ordinais mapeadas a partir de séries temporais e imagens.**

(a) Série temporal obtida da coordenada  $x$  do sistema de Rössler (parâmetros  $a = 0,3$ ,  $b = 2$  e  $c = 4$ ). Aqui, mostramos apenas as últimas 500 observações de todas as 100000 que constituem a série. Essa série temporal exibe um comportamento periódico após um curto transiente. (b) Visualização da rede ordinal mapeada da coordenada  $x$  do sistema de Rössler com parâmetros de *embedding*  $d_x = 16$  e  $\tau_x = 1$ . Essa rede ordinal usa do mapeamento original devido a Small, caracterizado por partições que não se sobrepõem e arestas não direcionadas e sem pesos. (c) Série temporal obtida de uma realização de um movimento browniano fracionário com  $h = 0,8$  (apenas as últimas 500 de  $2^{16}$  observações são mostradas). (d) Rede ordinal representando a série do movimento browniano fracionário com  $d_x = 3$  e  $\tau_x = 1$ . Foram usadas partições sobrepostas e as espessuras das arestas feitas proporcionais aos seus pesos. (e) Exemplo de um ornamento geométrico periódico de tamanho  $250 \times 250$  pixels. (f) Rede ordinal representando o ornamento com parâmetros de *embedding*  $d_x = d_y = 2$  e  $\tau_x = \tau_y = 1$ . Nessa visualização, a espessura das arestas é feita proporcional aos pesos.

persistente imposto pelo expoente de Hurst ( $h = 0,8$ ) é capturado pelas autoarestas intensas associadas aos padrões ordinais  $(0, 1, 2)$  e  $(2, 1, 0)$  (marcando as tendências ascendentes e descendentes ao longo dessa série temporal). Pessa e Ribeiro [32] também mostraram que propriedades de redes ordinais (por exemplo, o caminho mais curto médio) são efetivas na estimativa do expoente de Hurst de séries temporais e alcançaram desempenho superior a métodos comumente usados como a análise de flutuação destendenciada<sup>20</sup> [95].

Continuando, também consideramos redes ordinais mapeadas de dados bidimensionais. Mapeamos um ornamento periódico previamente explorado na referência [33] (Figura 2 dessa referência). A Figura 1.9(e) mostra um ornamento geométrico<sup>21</sup> de tamanho  $250 \times 250$ ,

<sup>20</sup>Do inglês, *detrended fluctuation analysis* (DFA).

<sup>21</sup>A definição desse ornamento é apresentada no Apêndice C.

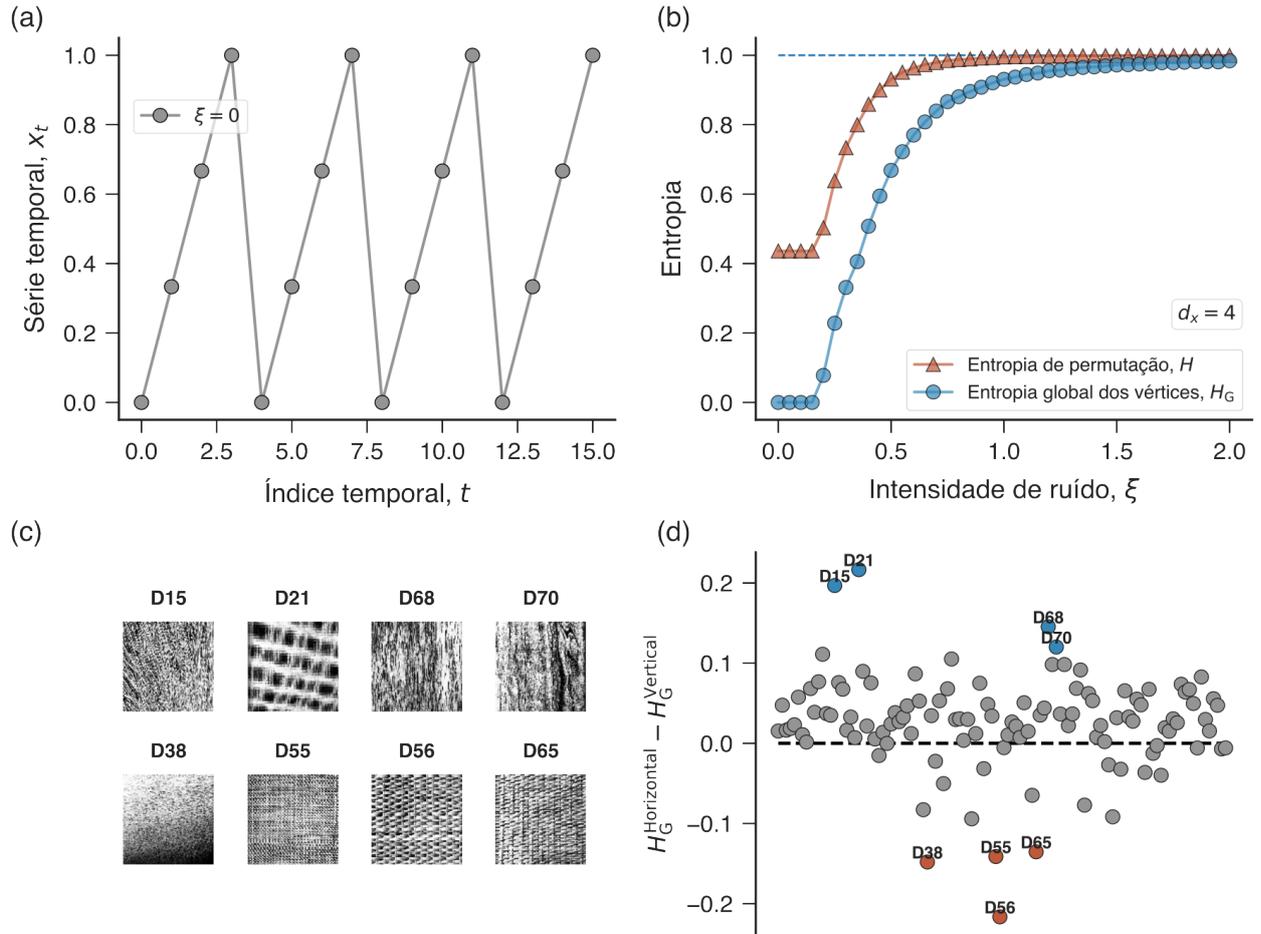
enquanto a Figura 1.9(f) apresenta uma visualização da rede ordinal correspondente com parâmetros de *embedding*  $d_x = d_y = 2$  e  $\tau_x = \tau_y = 1$ . Fizemos as espessuras das arestas proporcionais aos seus pesos (Equação 1.22) para ressaltar que algumas poucas arestas concentram a maior parte da probabilidade de transição na rede. Além disso, observamos que essa rede contém 12 nós e 72 arestas, isto é, somente uma fração de todos os nós (24) e arestas (416) possíveis de ocorrer em redes ordinais quando  $d_x = d_y = 2$  e  $\tau_x = \tau_y = 1$ .

Em adição aos exemplos anteriores, mais qualitativos, replicamos alguns resultados relacionados à entropia global dos vértices em redes ordinais. Para séries temporais, seguimos Pessa e Ribeiro [32] (Figura 5 nessa referência) e geramos um sinal periódico tipo dente de serra [Figura 1.10(a)] com  $10^5$  observações e adicionamos ruído branco uniformemente distribuído no intervalo  $[-\xi, \xi]$ , sendo  $\xi$  o parâmetro de amplitude do ruído. Geramos séries dente de serra ruidosas para cada  $\xi \in \{0; 0,05; 0,1; \dots; 2\}$  e determinamos os valores médios da entropia de permutação normalizada ( $H$ ) e da entropia global dos vértices normalizada ( $H_G$ ) para dez réplicas de séries temporais com parâmetros  $d_x = 4$  e  $\tau_x = 1$ .

A Figura 1.10(b) mostra os valores médios de  $H$  e  $H_G$  em função da amplitude de ruído  $\xi$ . Notamos que ambas as medidas aproximam-se de 1 com o aumento da amplitude do ruído. Entretanto, a entropia de permutação satura para  $\xi \approx 1$ , enquanto a entropia global dos vértices requer valores significativamente maiores de  $\xi$  para que o mesmo ocorra. Esse resultado indica que a entropia global dos vértices é mais robusta à adição de ruído e tem um maior poder discriminatório que a entropia de permutação [32].

Para demonstrar o uso da função `global_node_entropy` em dados bidimensionais, vamos antecipar um resultado do Capítulo 2 e calcular a entropia global dos vértices para um conjunto de 112 imagens de texturas naturais, as texturas normalizadas de Brodatz [96,97]. A Figura 1.10(c) mostra exemplos dessas imagens, enquanto a Figura 1.10(d) mostra a diferença entre a entropia global dos vértices para redes ordinais horizontais ( $H_G^{\text{Horizontal}}$ ) e verticais ( $H_G^{\text{Vertical}}$ ) mapeadas das texturas de Brodatz com  $d_x = d_y = 2$  e  $\tau_x = \tau_y = 1$ . Ressaltamos oito texturas com valores particularmente grandes para essa diferença. A maioria dessas imagens é caracterizada por faixas predominantemente orientadas nas direções vertical e horizontal, sugerindo que estatísticas das redes ordinais verticais e horizontais são capazes de detectar simetrias simples em imagens.

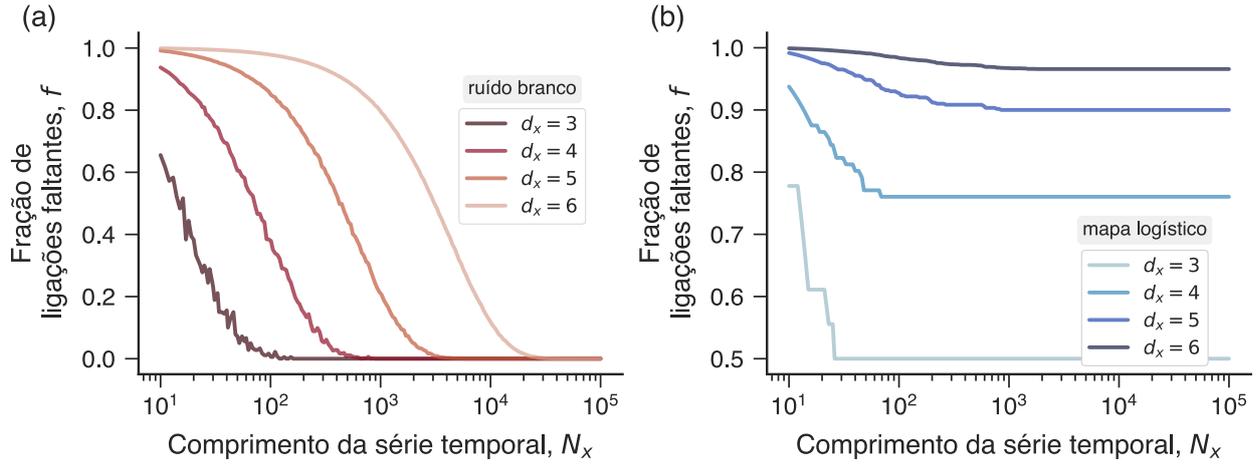
Em uma última aplicação usando redes ordinais, exploramos o conceito de ligações faltantes ou transições faltantes entre padrões ordinais [32]. Similarmente aos padrões ordinais faltantes descritos por Amigó *et al.* [68,69], redes ordinais podem apresentar transições proibidas entre padrões ordinais que são verdadeiras ou falsas. Mais uma vez, ligações verdadeiramente faltantes estão relacionadas à dinâmica intrínseca do processo sob análise, enquanto ligações falsamente faltantes estão associadas ao tamanho finito de um conjunto de dados. Uma vez que sabemos a forma exata de redes ordinais aleatórias e, conseqüentemente, todas as conexões possíveis em uma rede ordinal de mesmos parâmetros de *embedding* [32,33],



**Figura 1.10: Entropia global dos vértices aplicada a séries temporais e imagens.** (a) Primeiras observações de uma série dente de serra definida por  $x_t = \{0; 1/3; 1/6; 1; 0; 1/3; 1/6; 1; \dots\}$ . (b) Entropia de permutação normalizada ( $H$ ) e entropia global dos vértices normalizada ( $H_G$ ) em função da intensidade ( $\xi$ ) do ruído branco uniforme adicionado aos sinais periódicos tipo dente de serra. As diferentes curvas representam médias de  $H$  e  $H_G$  sobre dez realizações para  $\xi = \{0; 0,05; 0,1; \dots; 2\}$ . (c) Oito exemplos de texturas normalizadas de Brodatz. Essas são imagens em escala de cinza (256 níveis de cinza) de tamanho  $640 \times 640$  pixels [96]. (d) Diferenças entre a entropia global dos vértices avaliada a partir das redes ordinais horizontal e vertical ( $H_G^{\text{Horizontal}} - H_G^{\text{Vertical}}$ ) mapeadas de cada uma das 112 texturas de Brodatz. Oito texturas [mostradas no painel (c)] com as maiores diferenças (em módulo) são ressaltadas.

podemos prontamente encontrar todas as transições faltantes em uma rede ordinal empírica. A função `missing_links`, implementada no `ordpy`, avalia todas as transições faltantes diretamente de uma série temporal ou imagem ou, ainda, a partir dos vetores (`arrays`) retornados por `ordinal_network`:

```
>>> from ordpy import missing_links
>>> missing_links([4, 7, 9, 10, 6, 11, 3], dx=2, return_fraction=False)
(array([[ '1|0', '1|0' ]], dtype='<U3'), 1)
>>> missing_links(ordinal_network([4, 7, 9, 10, 6, 11, 3], dx=2),
```



**Figura 1.11: Ligações faltantes em redes ordinais.** (a) Relação entre a fração de ligações faltantes ( $f$ ) estimada a partir de ruído branco gaussiano em função do comprimento da série temporal ( $N_x$ ). (b) Dependência entre a fração de ligações faltantes ( $f$ ) estimada a partir do mapa logístico completamente caótico ( $r = 4$ ) e o comprimento da série temporal ( $N_x$ ). Em ambos os painéis, as diferentes curvas representam valores médios para 10 realizações (para cada um dos diferentes tamanhos de séries) e *embedding dimensions*  $d_x \in \{3, 4, 5, 6\}$  com  $\tau_x = 1$ . Os valores de  $N_x$  usados em ambos painéis correspondem a 194 números logaritmicamente espaçados no intervalo  $[10, 10^5]$ .

```
... dx=2, return_fraction=True)
(array([[ '1|0', '1|0']], dtype='<U3'), 0.25)
```

Para demonstrar o uso da função `missing_links`, replicamos os resultados de Pessa e Ribeiro [32] (Figura 4 nessa referência) acerca de ligações faltantes em redes ordinais mapeadas a partir de séries de ruído branco gaussiano. Geramos séries temporais com comprimento  $N_x$  variando logaritmicamente entre 10 e  $10^5$  e estimamos valores médios da fração de ligações faltantes usando dez réplicas para cada tamanho de série. Esse processo é repetido para as *embedding dimensions*  $d_x \in \{3, 4, 5, 6\}$  e  $\tau_x = 1$ . A Figura 1.11(a) mostra essas frações de ligações faltantes em função do comprimento da série temporal. Observamos que essa quantidade aproxima-se de zero conforme  $N_x$  se torna suficientemente grande. Além disso, quanto menor a *embedding dimension*, mais rapidamente desaparecem as ligações faltantes. Esse padrão é uma marca de ligações falsamente faltantes. Também realizamos a mesma análise com séries temporais geradas do mapa logístico ( $r = 4$ ) cujos resultados são apresentados na Figura 1.11(b). Diferentemente do ruído branco, o mapa logístico produz redes ordinais com ligações faltantes que persistem mesmo em séries consideravelmente longas, um comportamento típico de ligações verdadeiramente faltantes.

---

## Estrutura topológica de redes ordinais mapeadas de dados bidimensionais e aplicações

---

Neste segundo capítulo, focamos no estudo de nossa proposta para mapear dados bidimensionais em redes ordinais [33]. Primeiramente, discutimos em detalhe algumas restrições de conectividade que são intrínsecas a essas redes. Na sequência, aplicamos a transformação de redes ordinais a um conjunto de sistemas teste para mostrar a versatilidade de nossa técnica na análise de imagens. Por fim, em um cenário de classificação de imagens, comparamos as acurácias obtidas a partir do emprego de variáveis preditivas geradas por redes ordinais com outras obtidas de preditores tradicionalmente aplicados ao estudo de texturas.

### 2.1 Mapeamento de imagens em redes ordinais

Nas últimas duas décadas, a ciência de redes estabeleceu-se como um campo de pesquisa bastante vibrante e bem-sucedido [98]. O fato de que os mais diversos sistemas complexos são oportunamente descritos por conjuntos de vértices e arestas [99,100], combinado a avanços na aquisição e processamento de dados, tem levado à aplicação disseminada de redes complexas a uma imensa variedade de sistemas de origem biológica, tecnológica e social. Para além disso, desenvolvimentos recentes na ciência de redes têm originado maneiras novas e mais abstratas de definir redes complexas nas quais vértices e arestas não necessariamente representam os constituintes e as interações de um sistema mas, ao invés disso, têm significados diversos [34, 101].

Algoritmos criados para mapear séries temporais em redes complexas são uma classe particularmente importante de tais redes [34] e têm sido utilizados, por exemplo, na caracte-

rização de séries temporais de natureza caótica e estocástica obtidas experimentalmente ou a partir de simulações. Essas abordagens são coletivamente conhecidas como redes de séries temporais e exemplos proeminentes incluem os grafos de visibilidade [102], as redes de recorrência [103] e as redes ordinais<sup>1</sup> [31]. Os grafos de visibilidade e suas variantes [102, 105–108] mapeiam cada observação de uma série temporal em um vértice do grafo, e pares de vértices são conectados caso suas respectivas amplitudes na série temporal satisfaçam uma “condição de visibilidade” [102]. As redes de recorrência, por sua vez, são uma reinterpretação dos gráficos de recorrência [109], uma ferramenta gráfica desenvolvida no contexto de análise de séries temporais não lineares [7, 8]. Vértices de redes de recorrência representam um conjunto de estados obtidos de pequenos segmentos de séries temporais e arestas são alocadas entre vértices (pares de estados) que são suficientemente próximos de acordo com uma medida de distância [103]. As redes ordinais foram propostas mais recentemente e originam-se igualmente do estudo de séries temporais não lineares. Seus vértices representam permutações associadas a segmentos da série temporal e as ligações são estabelecidas com base nas sucessões dessas permutações ao longo da série (como vimos à Seção 1.6) [31, 32, 86, 110].

Enquanto o uso de algoritmos de mapeamento de séries temporais em redes tem ganho popularidade entre a comunidade científica (veja, por exemplo, Zou *et al.* [34] para uma revisão desses métodos), poucas propostas têm sido dedicadas à extensão dessas ideias a outros tipos de dados, como imagens. A maioria dos trabalhos nessa direção tem se restringido a representações de imagens como redes espaciais nas quais pixels (ou grupos de pixels) são mapeados em vértices ligados por arestas que refletem a distância e similaridade de cor entre eles. Esses métodos têm se provado úteis na detecção de arestas [111], extração de *features* de texturas [112] e segmentação de imagens [113], entre outras aplicações práticas [114–116]. Temos um número ainda menor de alternativas mais elaboradas de representações de imagens como redes. Exemplos notáveis incluem o trabalho de Xiao *et al.* [117] que trata da distribuição de grau de grafos de visibilidade “linha-coluna” obtidos de superfícies fractais e os avanços realizados por Lacasa e Iacovacci [118, 119] que analisaram grafos de visibilidade mapeados de dados bidimensionais caóticos e aleatórios e abordaram problemas de processamento de imagem e classificação. Uma vez que processos para extrair informação útil de imagens são ubíquos na ciência, estender e propor abordagens para o mapeamento de imagens em redes complexas pode contribuir com novos quantificadores de imagens derivados de métricas de rede bem estabelecidas.

Neste contexto, propusemos uma extensão do formalismo de redes ordinais que permite a representação de imagens na forma dessas redes complexas [33]. Essa proposta foi descrita durante a apresentação do pacote `ordpy` (Seção 1.6) e, nas seções que se seguem, descrevemos restrições de conectividade que essas redes herdaram do processo de simbolização e

---

<sup>1</sup>Para uma visão geral dessas transformações de séries temporais em redes complexas, veja a referência [104].

determinamos a forma exata de redes ordinais mapeadas de ruído bidimensional. Mapeando imagens obtidas de ornamentos periódicos, movimento browniano fracionário bidimensional e do modelo de Ising em redes ordinais, ilustramos o uso dessa nova abordagem para identificar e descrever esses sistemas a partir de métricas de rede. Aplicamos essa transformação na caracterização de um conjunto de fotografias de texturas a fim de mostrar que as redes ordinais são capazes de distinguir entre diferentes tipos de texturas e identificar simetrias nas imagens. Ainda, em duas tarefas de classificação de imagens, demonstramos que métricas de redes ordinais, quando utilizadas como preditores, são robustas em relação à adição de ruído e levam a acurácias mais altas que outras estatísticas tradicionalmente empregadas ao analisar imagens.

## 2.2 Restrições de conectividade em redes ordinais

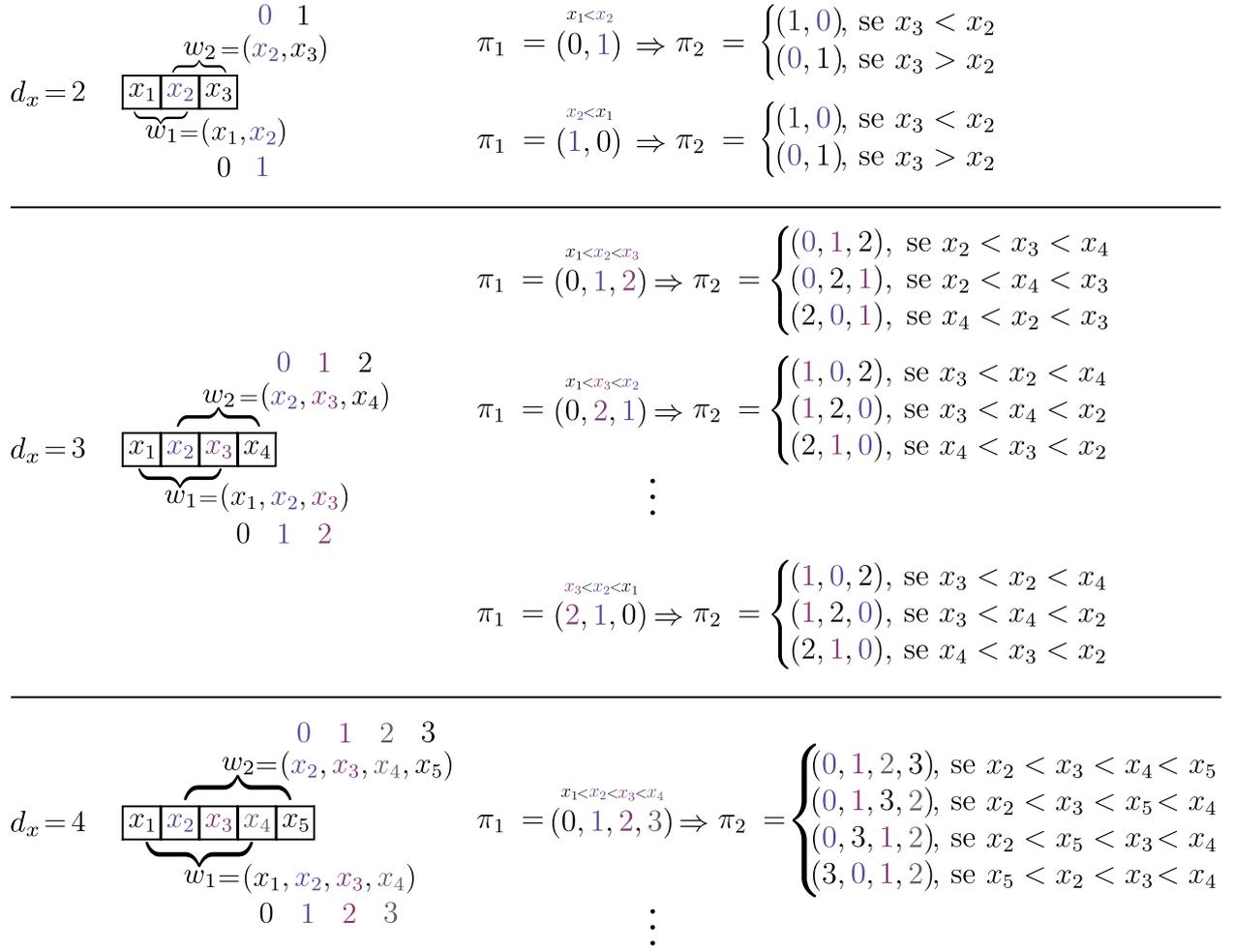
Uma vez que uma série temporal ou imagem é mapeada em uma rede ordinal, os nós e a estrutura das ligações da rede, sua topologia, passa a ser fundamental para a discriminação entre diferentes tipos de dados. Para isso, faz-se necessário conhecermos características fundamentais dessas redes que são consequências do procedimento de simbolização de Bandt e Pompe e que restringem consideravelmente a estrutura das redes ordinais (como mencionamos na Seção 1.6).

No caso de redes ordinais mapeadas de séries temporais, essas restrições se mostram nos graus de entrada e saída dos nós (restritos entre 0 e  $d_x$  devido a limitações de sucessão entre padrões ordinais) [32]. A Figura 2.1 ilustra como o ordenamento de partições adjacentes é limitado por seus elementos comuns, restringindo, por sua vez, as transições possíveis entre padrões ordinais.

Similarmente ao caso de séries temporais [32], redes ordinais mapeadas a partir de imagens (matrizes) apresentam restrições que proíbem a existência de inúmeras arestas entre os nós (permutações) de uma tal rede. Nesse caso, as restrições também são consequência do procedimento de simbolização e do particionamento dos dados e emergem mesmo em casos de matrizes constituídas de números aleatórios.

Para esclarecer esse fato, consideramos uma matriz  $\{y_i^j\}_{i=1,\dots,N_x}^{j=1,\dots,N_y}$  e os parâmetros de *embedding*  $d_x = d_y = 2$  e  $\tau_x = \tau_y = 1$  e supomos que a primeira submatriz obtida de  $\{y_i^j\}$ ,

$$w_1^1 = \begin{pmatrix} y_1^1 & y_1^2 \\ y_2^1 & y_2^2 \end{pmatrix},$$



**Figura 2.1: Transições permitidas em redes ordinais mapeadas de séries temporais.** Exemplos de possíveis sucessões entre permutações (ligações em redes ordinais) para as *embedding dimensions*  $d_x = 2, 3$  e  $4$  (com  $\tau_x = 1$ ) são mostrados. As observações de uma série temporal  $x_t = \{x_1, x_2, \dots\}$  que integram duas partições consecutivas ( $w_p$  e  $w_{p+1}$ ) da série são destacadas em cor. Além disso, os números de ordem associados a essas observações (em cada uma das janelas) são correspondentemente coloridos. Notamos que os índices de posição de cada observação da série decrescem com o avanço da janela móvel.

é caracterizada pela permutação  $\pi_1^1 = (2, 1, 3, 0)$ . Sua partição horizontalmente adjacente é

$$w_2^1 = \begin{pmatrix} y_1^2 & y_1^3 \\ y_2^2 & y_2^3 \end{pmatrix}.$$

Notamos que ambas partições compartilham os elementos  $y_1^2$  e  $y_2^2$ . Assim, uma vez que  $w_1^1$  é associada a  $\pi_1^1 = (2, 1, 3, 0)$  (isto é,  $y_2^1 < y_1^2 < y_2^2 < y_1^1$ ), a permutação  $\pi_2^1$  associada a  $w_2^1$  deve refletir a desigualdade  $y_1^2 < y_2^2$  manifesta em  $\pi_1^1$ . Decorre da restrição anterior que de todas as  $(d_x d_y)! = 24$  permutações possíveis,  $\pi_2^1$  é restringida a um conjunto de 12 permutações nas quais o número 0 (que correspondente à posição de  $y_1^2$  em  $w_2^1$ ) precede o

número 2 (correspondente à posição de  $y_2^2$ ). Partições verticalmente adjacentes apresentam restrições similares. Por exemplo,

$$w_1^2 = \begin{pmatrix} y_2^1 & y_2^2 \\ y_3^1 & y_3^2 \end{pmatrix}$$

compartilha os elementos  $y_2^1$  e  $y_2^2$  com  $w_1^1$  e o índice 0 (posição de  $y_2^1$  em  $w_1^2$ ) tem de preceder o índice 1 (posição de  $y_2^2$  em  $w_1^2$ ) em  $\pi_1^2$  pois a desigualdade  $y_2^1 < y_2^2$  é manifestada em  $\pi_1^1$ . Assim,  $\pi_1^2$  é também restringida por  $\pi_1^1$  a uma entre 12 permutações. Essas restrições são válidas independentemente da permutação correspondente a  $\pi_1^1$  e, para cada permutação, existem apenas 12 outras que podem ser imediatamente adjacentes (horizontal ou verticalmente) quando  $d_x = d_y = 2$  e  $\tau_x = \tau_y = 1$  e as partições se sobrepõem. O mesmo raciocínio aplica-se a sucessões entre permutações envolvendo valores arbitrários de  $d_x$  e  $d_y$ . Entretanto, é importante notar que o número de restrições em sucessões de permutações aumenta conforme as partições adjacentes compartilham mais elementos. Por exemplo, quando  $d_x = 3$  e  $d_y = 2$ , partições horizontalmente adjacentes compartilham quatro elementos, enquanto partições verticalmente adjacentes compartilham três. Para esses valores de *embedding dimensions*, de todas as  $(3 \times 2)! = 720$  permutações possíveis, uma vez que a permutação  $\pi_s^t$  é definida, existem apenas 30 padrões ordinais permitidos como vizinho horizontal ( $\pi_{s+1}^t$ ) e 120 padrões ordinais permitidos como vizinho vertical ( $\pi_s^{t+1}$ ). Considerando uma permutação e conhecendo o conjunto de vizinhos horizontais e verticais permitidos, o maior número possível de arestas de saída do vértice associado a essa permutação é dado pela interseção desses dois conjuntos. Interessantemente, encontramos que o maior número de arestas de saída depende do padrão ordinal. Por exemplo, o padrão ordinal  $\Pi_1 = (0, 1, 2, 3)$  pode ter no máximo 16 arestas de saída, enquanto  $\Pi_2 = (0, 1, 3, 2)$  pode ter até 20 ligações direcionadas em uma rede ordinal.

Investigando o número máximo de arestas possíveis para cada permutação, encontramos o maior número de arestas possível para toda a rede ordinal. Para os parâmetros de *embedding*  $d_x = d_y = 2$  e  $\tau_x = \tau_y = 1$ , encontramos que uma rede ordinal pode ter ao máximo 24 vértices e 416 arestas. Assim como no caso unidimensional [32], o número de nós e arestas em redes ordinais aumenta significativamente com as *embedding dimensions*. Por exemplo, uma rede ordinal pode ser formada por até 720 nós e 104184 arestas quando  $d_x = 3$  e  $d_y = 2$ . Isso impõe limitações importantes na investigação de dados relativamente pequenos, visto que a condição  $(d_x d_y)! \ll N_x N_y$  é necessária para uma estimativa confiável das probabilidades de transição (pesos das arestas) entre permutações. A maioria das investigações é, portanto, limitada a valores de *embedding dimensions* iguais a dois ou três.

## 2.3 Redes ordinais aleatórias

As restrições discutidas na seção anterior indicam que matrizes aleatórias mapeadas em redes ordinais não podem originar grafos aleatórios<sup>2</sup>. Conforme já discutimos, redes ordinais mapeadas de séries temporais aleatórias suficientemente longas (chamadas redes ordinais aleatórias) têm uma estrutura de ligações não trivial que pode ser determinada a partir das restrições de transição entre permutações e considerações probabilísticas acerca da amplitude das observações dentro das partições, independentemente do valor de  $d_x$  (contanto que  $\tau_x = 1$ ) [32]. Por exemplo, caso  $d_x = 2$ , a matriz de adjacência esperada para uma rede ordinal aleatória é dada por

$$\mathbf{A}_{d_x=2} = \begin{pmatrix} \frac{1}{(2+1)!} & \frac{2}{(2+1)!} \\ \frac{2}{(2+1)!} & \frac{1}{(2+1)!} \end{pmatrix} = \begin{pmatrix} 1/6 & 2/6 \\ 2/6 & 1/6 \end{pmatrix}, \quad (2.1)$$

com os pesos correspondendo, linha por linha, às seguintes ligações:  $p_{01 \rightarrow 01} = 1/6$ ,  $p_{10 \rightarrow 01} = 2/6$ ,  $p_{01 \rightarrow 10} = 2/6$ , e  $p_{10 \rightarrow 10} = 1/6$ .

Passamos agora a investigar como esses resultados são generalizados para dados bidimensionais. Para tanto, consideramos uma matriz de dados  $\{y_i^j\}_{i=1, \dots, N_x}^{j=1, \dots, N_y}$  cujos elementos são amostrados de uma distribuição de probabilidade de suporte contínuo e *embedding dimensions*  $d_x = d_y = 2$  e  $\tau_x = \tau_y = 1$ . As duas primeiras partições horizontalmente adjacentes são

$$w_1^1 = \begin{pmatrix} y_1^1 & y_1^2 \\ y_2^1 & y_2^2 \end{pmatrix}$$

e

$$w_2^1 = \begin{pmatrix} y_1^2 & y_1^3 \\ y_2^2 & y_2^3 \end{pmatrix},$$

e supomos que  $y_2^1 < y_1^2 < y_2^2 < y_1^1$ , isto é,  $\pi_1^1 = (2, 1, 3, 0)$ . Para determinarmos as probabilidades de transição (pesos das arestas, como definidos na Equação 1.22) de  $(2, 1, 3, 0)$  para outras permutações, precisamos encontrar todas as permutações que podem estar associadas a  $\pi_2^1$  avaliando as desigualdades que envolvem os valores de  $w_1^1$  e  $w_2^1$ . Começamos analisando todas as possíveis relações de amplitude entre  $y_1^3$  e a desigualdade já estabelecida

---

<sup>2</sup>Por grafos aleatórios, referimo-nos a grafos cujo processo de formação envolve alguma aleatoriedade [90]. No modelo mais simples de grafos aleatórios, devido a Paul Erdős e Alfred Rényi, uma vez fixado o número de vértices do grafo, arestas são alocadas entre pares de vértices com probabilidade  $p$  [90].

$y_2^1 < y_1^2 < y_2^2 < y_1^1$ , ou seja:

$$\begin{aligned}
i) & y_2^1 < y_1^2 < y_2^2 < y_1^1 < \mathbf{y}_1^3; \\
ii) & y_2^1 < y_1^2 < y_2^2 < \mathbf{y}_1^3 < y_1^1; \\
iii) & y_2^1 < y_1^2 < \mathbf{y}_1^3 < y_2^2 < y_1^1; \\
iv) & y_2^1 < \mathbf{y}_1^3 < y_1^2 < y_2^2 < y_1^1; \\
v) & \mathbf{y}_1^3 < y_2^1 < y_1^2 < y_2^2 < y_1^1.
\end{aligned} \tag{2.2}$$

Em seguida, incluímos  $y_2^3$  e analisamos cada relação de amplitude possível para cada uma das cinco condições anteriores. Esse procedimento leva-nos a um total de 30 relações de amplitude possíveis entre os seis elementos contidos nas partições  $w_1^1$  e  $w_2^1$ :

$$\begin{aligned}
i) & y_2^1 < y_1^2 < y_2^2 < y_1^1 < y_1^3 < \mathbf{y}_2^3; \\
ii) & y_2^1 < y_1^2 < y_2^2 < y_1^1 < \mathbf{y}_2^3 < y_1^3; \\
& \vdots \\
vii) & y_2^1 < y_1^2 < y_2^2 < y_1^3 < y_1^1 < \mathbf{y}_2^3; \\
& \vdots \\
xxx) & \mathbf{y}_2^3 < y_1^3 < y_2^1 < y_1^2 < y_2^2 < y_1^1.
\end{aligned} \tag{2.3}$$

Examinando as posições relativas de  $y_1^2, y_1^3, y_2^2$  e  $y_2^3$  (elementos que constituem  $w_2^1$ ) em cada uma das 30 desigualdades anteriores, podemos associar um padrão ordinal a cada uma dessas desigualdades:

$$\begin{aligned}
i) & \Pi_3 = (0, 2, 1, 3); \\
ii) & \Pi_4 = (0, 2, 3, 1); \\
& \vdots \\
vii) & \Pi_3 = (0, 2, 1, 3); \\
& \vdots \\
xxx) & \Pi_{21} = (3, 1, 0, 2).
\end{aligned} \tag{2.4}$$

Concluimos então que  $\pi_2^1$  pode estar associada a qualquer uma das permutações listadas acima. Conforme discutimos, existem apenas 12 permutações que podem seguir  $\pi_1^1$  horizontalmente, o que significa que algumas das permutações que decorrem das 30 desigualdades anteriores aparecem mais de uma vez (como vemos em *i* e *vii*) para diferentes relações de amplitude entre os seis elementos de  $w_1^1$  e  $w_2^1$ . Essas diferentes frequências de ocorrência implicarão na existência de diferentes pesos nas arestas de uma rede ordinal aleatória.

Repetindo o mesmo procedimento para

$$w_1^1 = \begin{pmatrix} y_1^1 & y_1^2 \\ y_2^1 & y_2^2 \end{pmatrix}$$

e

$$w_1^2 = \begin{pmatrix} y_2^1 & y_2^2 \\ y_3^1 & y_3^2 \end{pmatrix},$$

encontramos outro conjunto de 30 desigualdades e suas correspondentes permutações. Assim, a partir da análise das relações de amplitude entre dados em  $w_1^1$  e seus vizinhos  $w_2^1$  e  $w_1^2$ , encontramos um total de 60 desigualdades correspondentes a 60 permutações (que podem se repetir). Como todas essas desigualdades envolvem dados aleatórios, todas elas são equiprováveis e podemos contar as ocorrências dessas permutações diferentes para definir suas frequências relativas (em relação às 60 desigualdades anteriores). Essas frequências relativas indicam as probabilidades de transição entre  $(2, 1, 3, 0)$  e outras permutações (sendo sua soma igual a um). Para finalmente encontrarmos os pesos das arestas (Equação 1.22) que compõem uma rede ordinal aleatória bidimensional, normalizamos essas frequências relativas de sucessão entre padrões ordinais dividindo-as por  $(d_x d_y)!$ . Esse passo é necessário para garantir que a soma de todos os pesos das arestas da rede seja igual a um e a interpretação probabilística das mesmas fique evidente.

Automatizando todo esse intrincado procedimento, de modo a considerar explicitamente cada padrão ordinal possível de ocorrer em  $w_1^1$  [isto é,  $\pi_1^1 = (0, 1, 2, 3)$ ,  $(0, 1, 3, 2)$ ,  $(0, 2, 1, 3)$ , e assim por diante], podemos estimar os pesos de todas as ligações que podem ocorrer em uma rede ordinal mapeada de um dado bidimensional aleatório suficientemente longo<sup>3</sup>. Essa abordagem permite-nos especificar completamente a estrutura esperada de uma rede ordinal aleatória para valores arbitrários das *embedding dimensions*  $d_x$  e  $d_y$  e estimar todas as métricas relevantes da rede (incluindo a entropia global dos vértices definida na Equação 1.24).

## 2.4 Redes ordinais de ornamentos periódicos ruidosos

Uma aplicação que nos permite prontamente observar a emergência da estrutura de ligações de uma rede ordinal aleatória é o processo de randomização de um ornamento geométrico<sup>4</sup> [28]. Com essa aplicação, pretendemos generalizar algumas análises realizadas com séries temporais regulares e estocásticas [32] e observar o comportamento de algumas métricas das redes ordinais.

A Figura 2.2(a) ilustra o processo de randomização do ornamento, no qual a probabilidade

---

<sup>3</sup>A função `random_ordinal_network` do pacote `ordpy` implementa essas ideias para construir a forma esperada de uma rede ordinal aleatória mapeada a partir de um dado bidimensional.

<sup>4</sup>Esse ornamento e seu processo de randomização são definidos no Apêndice C.

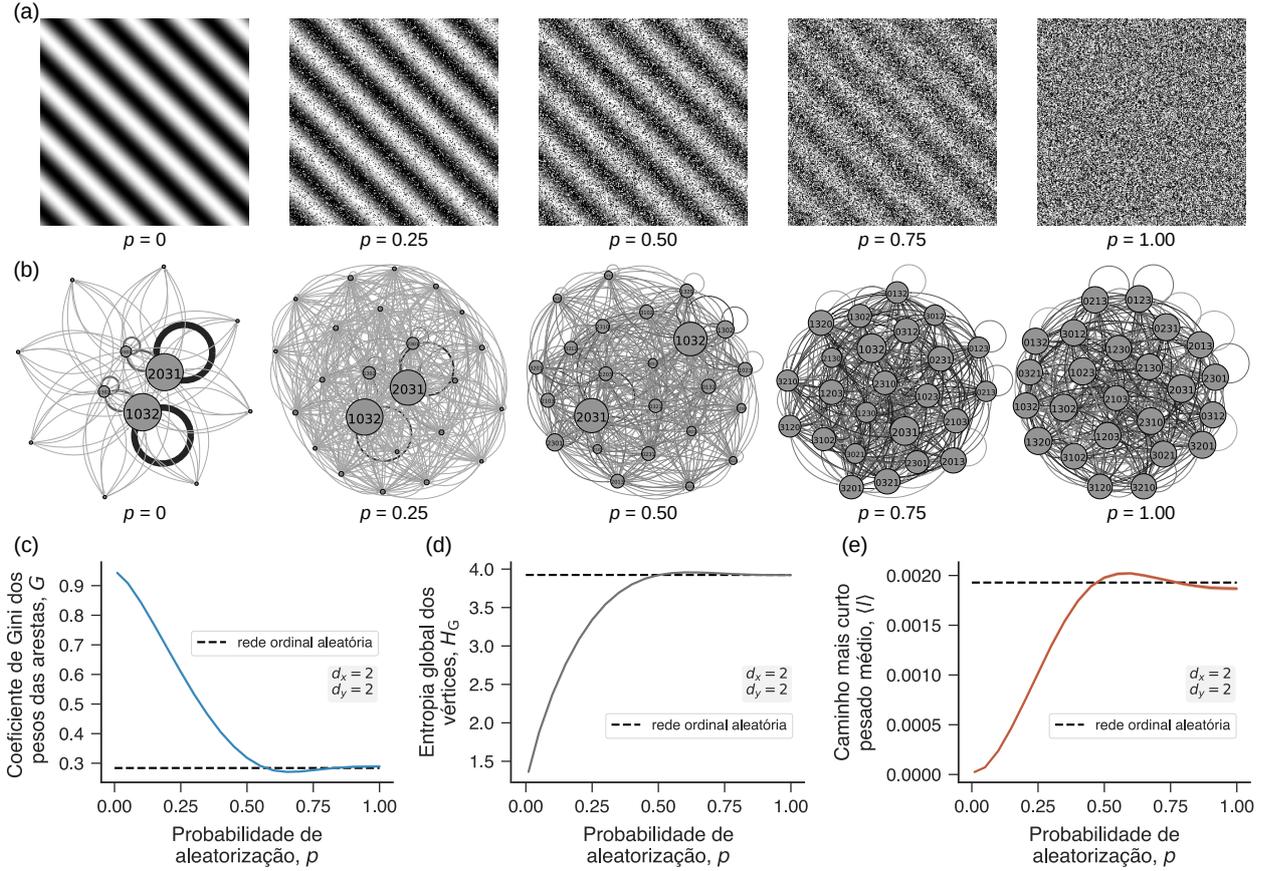
$p$  de embaralhar os pixels controla a transição de uma imagem periódica ( $p = 0$ ) para uma imagem aleatória ( $p = 1$ ). Redes ordinais correspondentes a essas imagens ( $250 \times 250$  pixels) são mostradas na Figura 2.2(b). Percebemos que o aspecto visual dessas redes ordinais alude às características desses ornamentos periódicos ruidosos. Para valores pequenos de  $p$ , observamos que as redes ordinais são marcadas por conexões intensas (que concentram grande parte da probabilidade de transição na rede) entre os nós  $(1, 0, 3, 2)$  e  $(1, 3, 0, 2)$ , assim como entre  $(2, 0, 3, 1)$  e  $(2, 3, 0, 1)$ . Esses quatro vértices apresentam autoarestas ainda mais intensas. Com o aumento dos valores de  $p$ , a força dessas ligações desvanece e dá origem a uma distribuição muito mais uniforme de pesos.

A fim de investigarmos sistematicamente as redes ordinais desses ornamentos, geramos um *ensemble* contendo 100 amostras para cada probabilidade de embaralhamento  $p \in \{0,01; 0,05; 0,10; \dots; 1,00\}$  e as transformamos em redes ordinais usando os parâmetros de *embedding*  $d_x = d_y = 2$  e  $\tau_x = \tau_y = 1$ . A partir dessas redes, estimamos o valor médio do coeficiente de Gini dos pesos das arestas  $G$ , a entropia global dos vértices normalizada  $H_G$  e o caminho mais curto pesado médio  $\langle l \rangle$ , como mostram as Figuras 2.2(c)-(e). O coeficiente de Gini pode ser definido como [120]

$$G = \frac{\sum_i^{N_e} (2i - N_e - 1)x_i}{N_e \sum_i^{N_e} x_i}, \quad (2.5)$$

com  $x_i$  representando os pesos das arestas em ordem ascendente,  $i$  o ranking do peso e  $N_e$  o número de arestas na rede. Em redes pesadas, o caminho mais curto médio é definido como o valor médio dos caminhos de menor soma de pesos que ligam todos os pares de vértices. Além disso, estimamos os valores dessas três métricas de redes a partir da forma exata das redes ordinais aleatórias de mesmo parâmetros de *embedding* [linhas tracejadas nas Figuras 2.2(c)-(e)].

Conforme esperado, observamos que as três medidas aproximam-se do valor exato para a rede ordinal aleatória com o aumento de  $p$ . É interessante perceber que a entropia global dos vértices  $H_G$  ultrapassa o valor esperado para o caso aleatório quando  $p \cong 0,5$  e, em seguida, converge para esse valor quando  $p \rightarrow 1$ . Isso acontece porque redes ordinais aleatórias não são as redes ordinais mais entrópicas [32], uma característica explicada pelo fato dos pesos das arestas não serem uniformes (como argumentado na Seção 2.3). Também observamos que o caminho mais curto pesado médio  $\langle l \rangle$  parece convergir para um valor ligeiramente menor que o esperado teoricamente. Essa aparente discrepância acontece porque as imagens são relativamente pequenas, impedindo uma estimativa mais precisa das probabilidades de transição entre permutações. Verificamos, entretanto, que essa diferença desvanece com o aumento do tamanho da imagem.



**Figura 2.2: Emergência de redes ordinais aleatórias em ornamentos periódicos ruidosos.** (a) Visualizações de imagens de um ornamento geométrico para diferentes probabilidades de randomização dos pixels (mostradas logo abaixo das imagens). (b) Redes ordinais ( $d_x = d_y = 2$  e  $\tau_x = \tau_y = 1$ ) mapeadas das imagens mostradas no painel anterior. A espessura e intensidade de cor de cada aresta são proporcionais ao seu peso (probabilidade associada a uma transição). O tamanho dos vértices reflete a soma dos pesos das arestas de entrada, de modo que vértices (permutações) maiores são mais prováveis de ocorrerem na sequência simbólica. Dependência do (c) coeficiente de Gini  $G$ , (d) entropia global dos vértices  $H_G$  e (e) caminho mais curto pesado médio  $\langle l \rangle$  com a probabilidade de aleatorização  $p$  dos pixels do ornamento. Nos últimos três painéis, as linhas contínuas indicam os valores médios para as métricas das redes e as regiões sombreadas indicam uma banda de um desvio padrão estimada a partir de um *ensemble* de 100 amostras do ornamento para cada valor de  $p$ . As linhas pretas tracejadas indicam os valores exatos (teoricamente esperados) das respectivas métricas para redes ordinais aleatórias quando  $d_x = d_y = 2$ .

## 2.5 Redes ordinais de imagens brownianas fracionárias

Em outra aplicação, investigamos redes ordinais mapeadas de imagens de movimentos brownianos fracionários bidimensionais [121] visando estender resultados previamente obtidos para o caso unidimensional [32]. Essa classe de processos estocásticos foi proposta como um modelo teórico de paisagens e relevos naturais e é caracterizada pelo expoente de Hurst

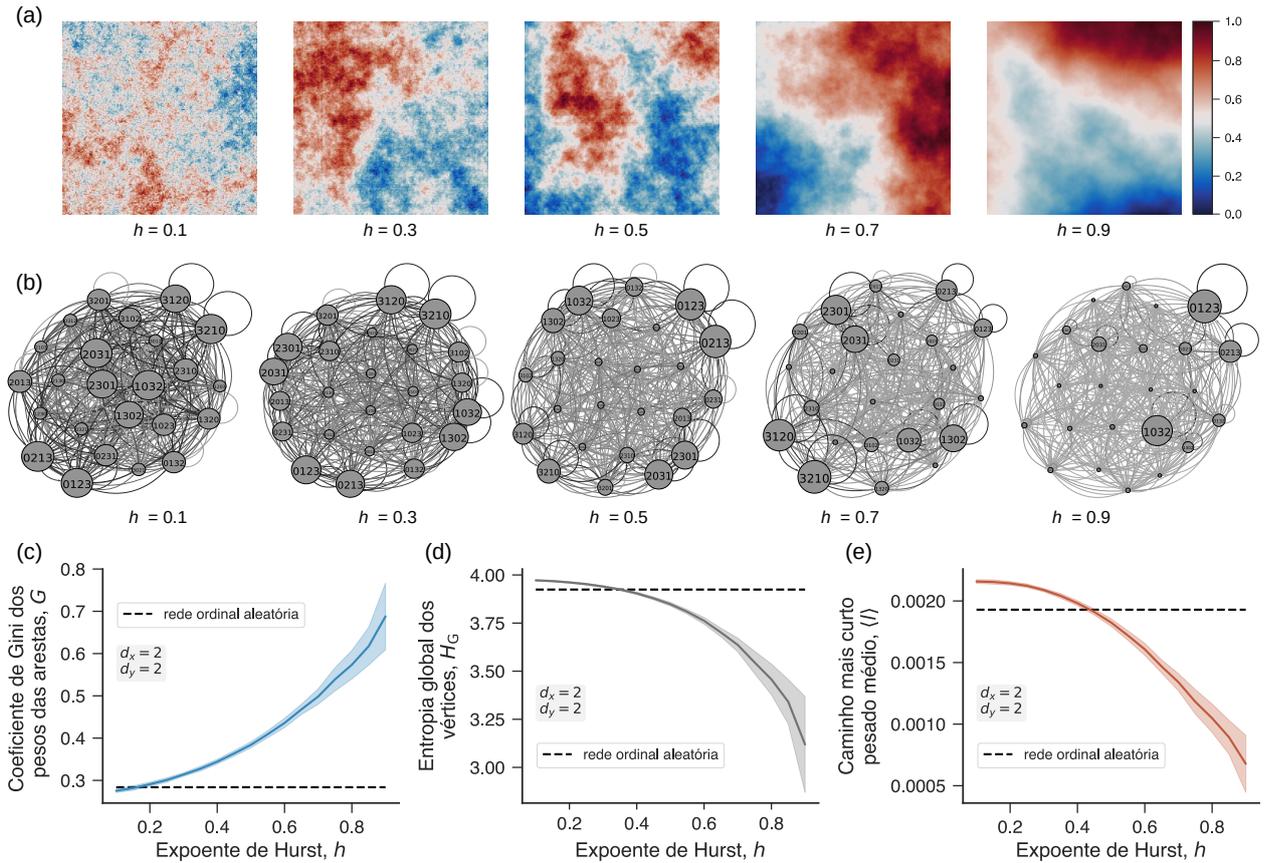
$h \in (0, 1)$ , um parâmetro que controla a rugosidade da superfície. Superfícies geradas com valores pequenos de  $h$  ( $h \rightarrow 0$ ) são rugosas enquanto valores altos de  $h$  ( $h \rightarrow 1$ ) produzem superfícies suaves. Cortes transversais de imagens brownianas fracionárias com  $h = 1/2$  representam caminhadas aleatórias usuais (movimento browniano). A Figura 2.3(a) mostra exemplos de imagens brownianas fracionárias geradas pelo método *turning bands*<sup>5</sup> [122] para diferentes valores do expoente de Hurst. Essencialmente, essas imagens são matrizes cujos elementos representam a altura de um relevo hipotético em um dado ponto.

Geramos um *ensemble* contendo 100 imagens brownianas fracionárias de tamanho  $256 \times 256$  para cada valor de  $h \in \{0,10; 0,15; 0,20; \dots; 0,90\}$  e mapeamos cada amostra em uma rede ordinal com parâmetros de *embedding*  $d_x = d_y = 2$  e  $\tau_x = \tau_y = 1$ . A Figura 2.3(b) apresenta alguns exemplos dessas redes ordinais mapeadas a partir das imagens da Figura 2.3(a). Observamos que mudanças na rugosidade das superfícies afetam os padrões de conectividade das redes resultantes. Superfícies mais rugosas ( $h \rightarrow 0$ ) produzem redes ordinais com distribuições menos desiguais dos pesos das arestas que se assemelham a redes ordinais aleatórias [último painel na Figura 2.2(b)]. Conforme as superfícies brownianas fracionárias tornam-se mais suaves, observamos uma concentração de pesos em algumas arestas entre vértices particulares, enquanto a intensidade da maioria das ligações diminui. Essa concentração da probabilidade de transição entre poucos nós reflete a ocorrência predominante de apenas algumas permutações nas sequências simbólicas relacionadas a imagens menos rugosas.

Além dessas observações mais qualitativas, calculamos os valores médios do coeficiente de Gini dos pesos das arestas  $G$ , a entropia global dos vértices normalizada  $H_G$  e o caminho mais curto pesado médio  $\langle l \rangle$  em função do expoente de Hurst  $h$  usando nosso *ensemble* de superfícies brownianas fracionárias. As Figuras 2.3(c)-(e) mostram o comportamento dessas três métricas de rede com o aumento do parâmetro  $h$ . Esses resultados estão de acordo com as nossas conclusões anteriores. Especificamente, o coeficiente de Gini cresce juntamente com o expoente de Hurst, refletindo a concentração de peso em algumas poucas arestas. Por outro lado, a entropia global dos vértices  $H_G$  e o caminho mais curto médio pesado  $\langle l \rangle$  diminuem com o aumento de  $h$ . Os relacionamentos monotônicos dessas três métricas com o expoente de Hurst indicam que seus valores podem ser usados como preditores da rugosidade dessas superfícies, assim como acontece no caso unidimensional [32]. Também verificamos que o comportamento dessas métricas obtidas com  $d_x = d_y = 2$  é similar àquele obtido para  $d_x = 3$  e  $d_y = 2$  (ou  $d_x = 2$  e  $d_y = 3$ ) com imagens (superfícies) maiores.

---

<sup>5</sup>Esse método é descrito mais detalhadamente no Apêndice C.



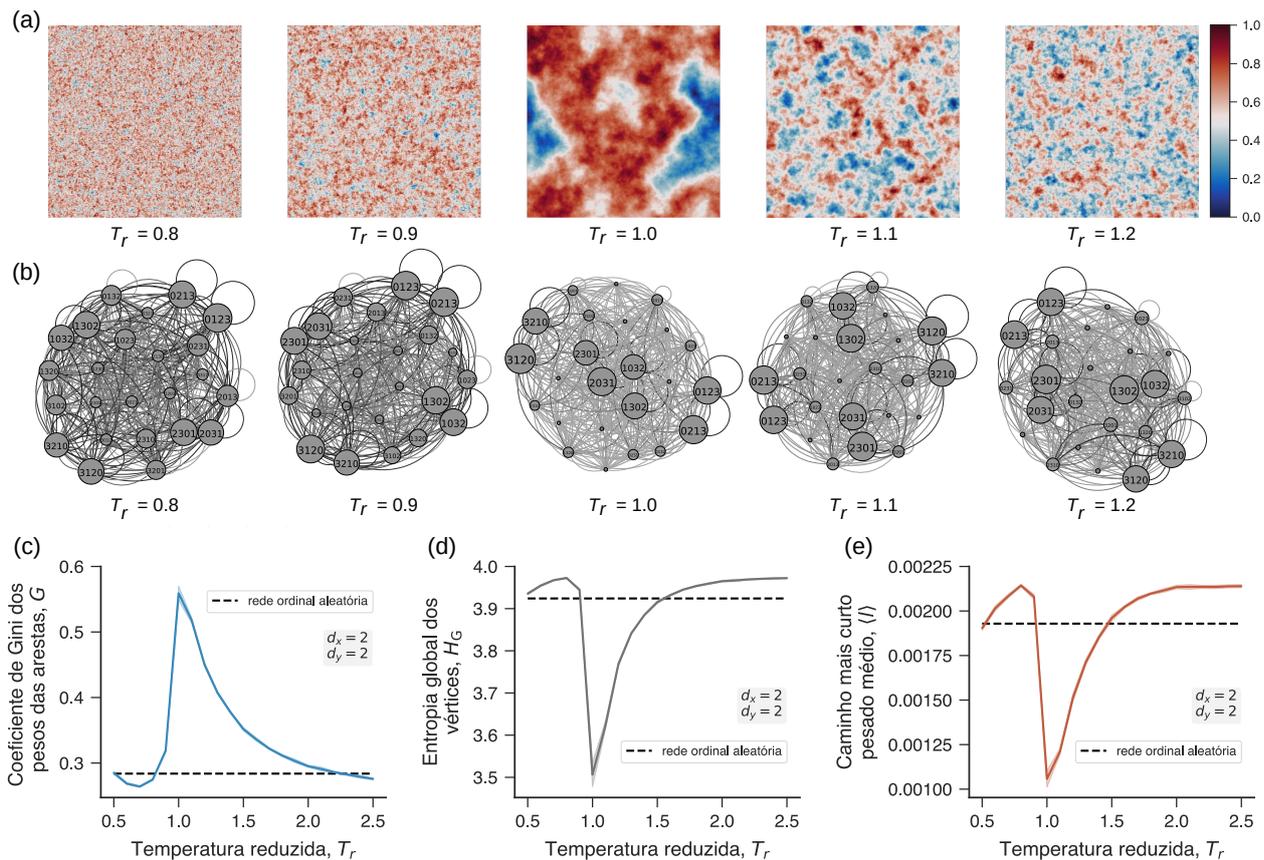
**Figura 2.3: Redes ordinais de superfícies brownianas fracionárias.** (a) Exemplos de superfícies brownianas fracionárias para alguns valores do expoente de Hurst (mostrados abaixo de cada imagem). Normalizamos as superfícies no intervalo  $[0,1]$  de modo que tons de azul indicam regiões de baixa amplitude (relevo de baixa altitude) e tons de vermelho indicam o contrário. (b) Redes ordinais mapeadas de superfícies fractais mostradas no painel anterior. Dependência do (c) coeficiente de Gini dos pesos das arestas  $G$ , (d) entropia global dos vértices  $H_G$  e (e) caminho mais curto pesado médio  $\langle l \rangle$  com o expoente de Hurst  $h$  do movimento browniano fracionário bidimensional. Nesses três últimos painéis, as linhas contínuas representam valores médios (calculados de um *ensemble* de 100 superfícies para cada valor de  $h$ ) para cada métrica de rede e as regiões sombreadas indicam uma banda de um desvio padrão. As linhas pretas tracejadas representam os valores para as respectivas métricas estimadas para redes ordinais aleatórias de mesmos parâmetros de *embedding*.

## 2.6 Redes ordinais de superfícies de Ising

Aplicamos também o formalismo de redes ordinais às superfícies de Ising<sup>6</sup> [123, 124] para verificar se métricas de rede são capazes de identificar transições de fase nesse sistema simulado. Essas superfícies são obtidas ao acumular variáveis de *spin*  $\sigma(t)$  do modelo bidimensional canônico de Ising em uma simulação de Monte Carlo [70].

A Figura 2.4(a) mostra exemplos de superfícies de Ising de tamanho  $250 \times 250$  pixels obti-

<sup>6</sup>Esse sistema é definido no Apêndice C.



**Figura 2.4: Redes ordinais de superfícies de Ising.** (a) Ilustração de superfícies de Ising para valores diferentes da temperatura reduzida  $T_r$  (indicados abaixo das imagens). Notamos que padrões mais complexos emergem perto da criticidade ( $T_r = 1$ ). (b) Exemplos de redes ordinais mapeadas a partir das imagens no painel anterior. Dependência do (c) coeficiente de Gini dos pesos das arestas  $G$ , (d) entropia global dos vértices normalizada  $H_G$  e (e) caminho mais curto pesado médio  $\langle l \rangle$  com a temperatura reduzida  $T_r$ . As linhas tracejadas nos três últimos painéis indicam valores da métrica correspondente estimados de redes ordinais aleatórias.

das de diferentes temperaturas reduzidas  $T_r = T/T_c$ , sendo  $T_c = 2/\ln(1+\sqrt{2})$  a temperatura crítica na qual o modelo sofre uma transição de fase. Superfícies geradas a temperaturas distantes do valor crítico ( $T_r = 1$ ) parecem aleatórias, sem qualquer tipo de padrão ou estrutura, da mesma forma que o ornamento da Figura 2.2(a) quando  $p = 1$ . Entretanto, começamos a observar padrões mais complexos conforme a temperatura reduzida se aproxima e, especialmente, quando alcança o valor crítico.

Geramos um *ensemble* contendo 10 superfícies de Ising de tamanho  $250 \times 250$  pixels para cada valor de  $T_r \in \{0,5; 0,6; \dots; 2,5\}$ . Cada superfície é obtida da soma acumulada de variáveis de *spin* durante 30000 passos de Monte Carlo para que o equilíbrio seja atingido [27, 125]. Em seguida, mapeamos todas as superfícies em redes ordinais com parâmetros de *embedding*  $d_x = d_y = 2$  e  $\tau_x = \tau_y = 1$ . A Figura 2.4(b) mostra exemplos de redes mapeadas das imagens na Figura 2.4(a). Uma simples inspeção visual dessas redes ordinais sugere que

suas propriedades mudam com a temperatura reduzida.

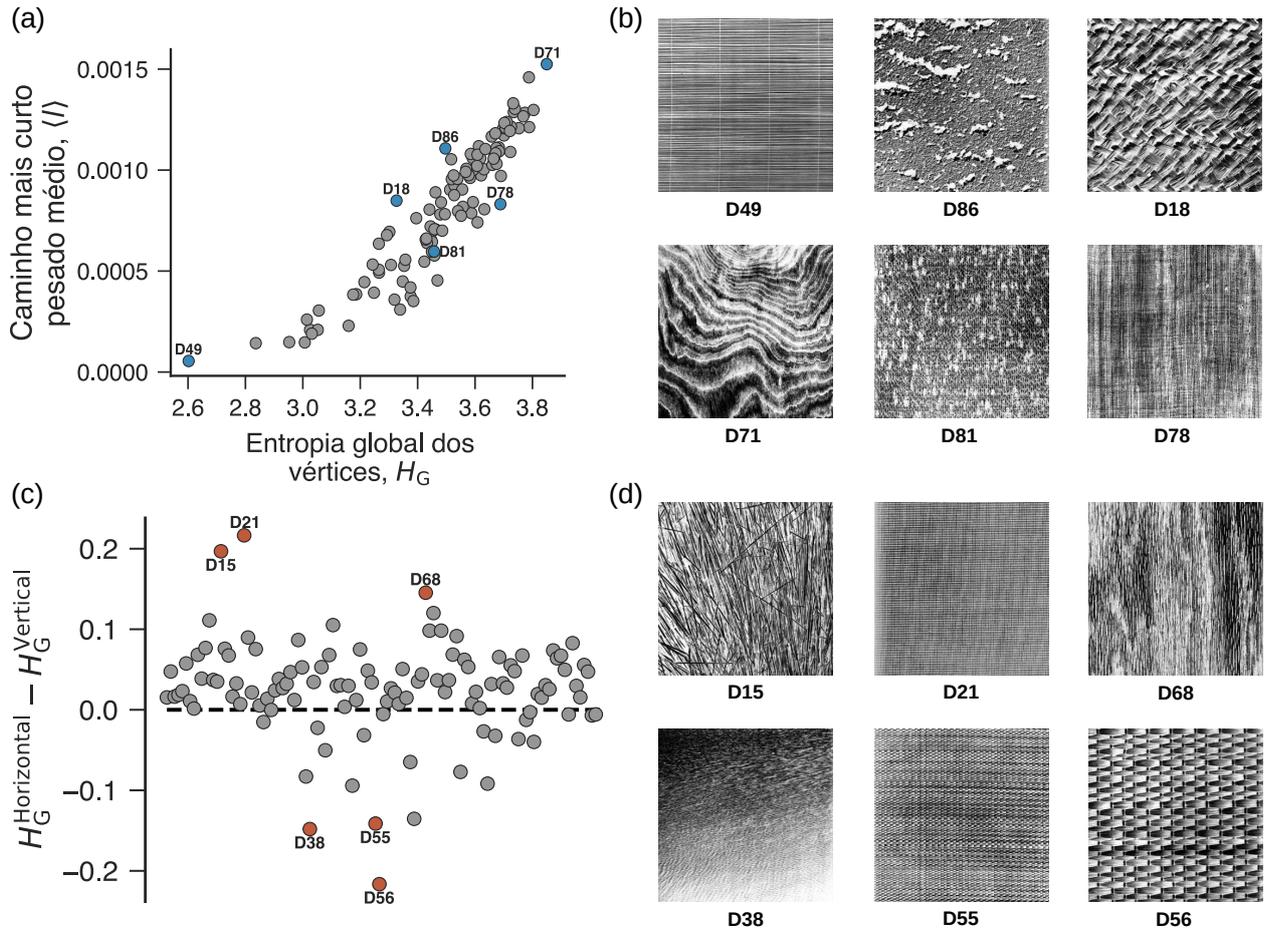
Similarmente às aplicações anteriores, calculamos o coeficiente de Gini dos pesos das arestas  $G$ , a entropia global dos vértices normalizada  $H_G$  e o caminho mais curto pesado médio  $\langle l \rangle$  em função da temperatura reduzida  $T_r$ . Os resultados apresentados nas Figuras 2.4(c)-(e) mostram que as três métricas apresentam variações abruptas ao redor da transição de fase em  $T_r = 1$ , assumindo valores extremos na temperatura crítica. Interessantemente, os valores dessas métricas na temperatura crítica são os mais distantes daqueles estimados para redes ordinais aleatórias [linhas tracejadas nas Figuras 2.4(c)-(e)]. Além disso, notamos que variações dessas métricas são mais pronunciadas antes da criticalidade, indicando que correlações espaciais são lentamente quebradas com o aumento da temperatura. Vale a pena observar que a altas temperaturas essas redes tornam-se ligeiramente mais entrópicas que redes ordinais aleatórias.

## 2.7 Redes ordinais de texturas naturais

Continuando a aplicar nosso método de redes ordinais em um cenário de testes, utilizamos esse algoritmo para mapear imagens de texturas naturais em redes ordinais. Para tanto, consideramos 112 fotografias de texturas que compõem o conjunto de texturas normalizadas de Brodatz [97], um grupo de imagens 8 bits (256 níveis de cinza) de tamanho  $640 \times 640$  pixels que abrange uma grande variedade de texturas naturais frequentemente utilizadas na validação de métodos de análise de texturas [97]. Mapeamos essas imagens em redes ordinais usando os parâmetros de *embedding*  $d_x = d_y = 2$  e  $\tau_x = \tau_y = 1$ . Em seguida, estimamos o coeficiente de Gini dos pesos das arestas  $G$ , a entropia global dos vértices normalizada  $H_G$ , e o caminho mais curto pesado médio  $\langle l \rangle$  de cada rede.

Essas medidas de rede são correlacionadas entre si com valores do coeficiente de correlação de Pearson variando entre 0,92 (para  $\langle l \rangle$  versus  $H_G$ ) até  $-0,97$  (para  $G$  versus  $H_G$ ). A Figura 2.5(a) mostra a relação entre  $\langle l \rangle$  e  $H_G$  (a associação de variáveis menos correlacionada) para todas as 112 texturas. O padrão de dispersão sugere que ambas as medidas são relacionadas não linearmente. A Figura 2.5(b) mostra seis imagens diferentes que são realçadas na Figura 2.5(a). Observamos que as duas texturas caracterizadas pelos valores mais extremos de  $\langle l \rangle$  e  $H_G$  (D49 e D71) são bastante diferentes: enquanto faixas horizontais regulares marcam a textura D49, a textura D71 mostra estruturas muito mais complexas.

Notamos a existência de texturas com valores similares de  $H_G$  e valores distintos de  $\langle l \rangle$  e vice-versa. Esses resultados sugerem que ambas medidas podem quantificar aspectos diferentes de imagens. Por exemplo, as texturas D86 e D81 [ressaltadas na Figura 2.5(a)] têm aproximadamente o mesmo valor de entropia global dos vértices, mas o valor de  $\langle l \rangle$  é consideravelmente maior para D86 do que para D81. Inspeccionando essas duas texturas na Figura 2.5(b), notamos que D86 parece mais rugosa que D81. Similarmente, as texturas



**Figura 2.5:** Redes ordinais mapeadas de texturas de Brodatz. (a) Localização das texturas de Brodatz em um plano determinado pela entropia global dos vértices  $H_G$  e pelo caminho mais curto pesado médio  $\langle l \rangle$ . (b) Seis imagens diferentes correspondendo às texturas ressaltadas (marcadores azuis) no painel anterior. (c) Diferença entre a entropia global dos vértices estimados das redes ordinais horizontais e verticais ( $H_G^{\text{Horizontal}} - H_G^{\text{Vertical}}$ ). (d) Seis imagens correspondentes às texturas realçadas (marcadores vermelhos) no painel anterior.

D18 e D78 têm valores comparáveis de  $\langle l \rangle$ , mas valores de  $H_G$  razoavelmente diferentes. A inspeção visual dessas texturas sugere que D18 é mais regular e estruturada do que D78. Apesar de ser difícil generalizar essas interpretações a outras imagens, acreditamos que os valores de  $H_G$  quantificam padrões em um nível local enquanto  $\langle l \rangle$  e  $G$  são medidas mais globais. Essa ideia concorda com a definição dessas medidas, uma vez que  $H_G$  é baseada em relações envolvendo primeiros vizinhos, enquanto que  $\langle l \rangle$  e  $G$  envolvem toda a rede ordinal.

Ainda investigamos a possibilidade de explorar simetrias nas texturas de Brodatz. Para tanto, implementamos uma pequena modificação em nosso algoritmo original para criar duas redes ordinais de uma única imagem (como mencionado na Seção 1.6). Uma dessas redes considera apenas transições horizontais entre permutações (redes ordinais horizontais) e a outra utiliza apenas transições verticais entre permutações (redes ordinais verticais). Assim, mapeamos cada textura de Brodatz em uma rede ordinal horizontal e uma rede

ordinal vertical com parâmetros de *embedding*  $d_x = d_y = 2$  e  $\tau_x = \tau_y = 1$ . Em seguida, estimamos a entropia global dos vértices das redes ordinais horizontal ( $H_G^{\text{Horizontal}}$ ) e vertical ( $H_G^{\text{Vertical}}$ ). A Figura 2.5(c) mostra a diferença entre essas quantidades ( $H_G^{\text{Horizontal}} - H_G^{\text{Vertical}}$ ) para cada textura. Observamos algumas texturas com valores extremos para essa diferença e realçamos seis delas, as quais são apresentadas na Figura 2.5(d). A maioria dessas imagens é caracterizada por faixas predominantemente orientadas nas direções vertical ou horizontal, indicando que redes ordinais verticais e horizontais são capazes de detectar essas simetrias simples.

## 2.8 Comparação entre redes ordinais e métodos tradicionais de análise de imagem

Como última aplicação de redes ordinais a imagens neste capítulo, comparamos variáveis descritivas de imagens derivadas de redes ordinais com outras estatísticas. Especificamente, utilizamos as chamadas matrizes de coocorrência de nível de cinza (GLCMs)<sup>7</sup> [127, 128], uma das abordagens mais conhecidas e utilizadas para extrair variáveis descritivas e analisar texturas [129, 130]. Os elementos de uma GLCM representam o número (ou probabilidade) de coocorrências de níveis de cinza a uma distância  $D$  e ângulo  $\theta$  entre os pixels de uma imagem. Nessa abordagem, obtemos variáveis descritivas calculando diferentes estatísticas dessas matrizes de coocorrência. Aqui, utilizamos cinco das propriedades mais comuns de uma GLCM: contraste, dissimilaridade, homogeneidade, energia e correlação. Para as redes ordinais, além do coeficiente de Gini dos pesos das arestas, da entropia global dos vértices e do caminho mais curto pesado médio, consideramos três outros quantificadores adicionais das imagens: variância da força dos nós, curtose da força e transitividade pesada<sup>8</sup>.

Tendo como base os experimentos numéricos anteriormente descritos, propomos duas tarefas de classificação<sup>9</sup> para comparar as redes ordinais e as GLCMs. Na primeira tarefa, utilizamos as estatísticas mencionadas acima como variáveis preditoras (*features*) do expoente de Hurst  $h$  em imagens brownianas fracionárias. Para tanto, mais uma vez, geramos um conjunto de dados composto de 100 superfícies fractais de tamanho  $256 \times 256$  pixels para cada “classe” de  $h \in \{0,10; 0,15; \dots; 0,90\}$  e calculamos as *features* das imagens com as

<sup>7</sup>Do inglês, *gray-level co-occurrence matrices* (GLCMs). Como o nome sugere, essas matrizes foram propostas para serem aplicadas a imagens em nível de cinza. Sendo assim, antes de aplicarmos essa técnica, transformamos as imagens analisadas em imagens de 8 bits em nível de cinza. Nessas imagens, as intensidades de cada pixel (elementos da matriz) variam entre 0 (preto) e 255 (branco). Essas matrizes e suas estatísticas mais comuns são definidas no Apêndice D. Para os cálculos envolvendo as GLCMs e suas estatísticas, empregamos o pacote Python `scikit-image` [126].

<sup>8</sup>A força de um nó é dada pela soma dos pesos das arestas ligadas a ele [90]. A transitividade pesada é uma medida da tendência de formação de triângulos na rede que leva em conta os pesos das arestas [90].

<sup>9</sup>O Apêndice F traz uma breve apresentação de conceitos fundamentais relacionados à teoria de aprendizagem de máquina e aos algoritmos utilizados ao longo deste trabalho. Na aplicação desses algoritmos aos conjuntos de dados, utilizamos as implementações contidas no pacote Python `scikit-learn` [131].

GLCMs e redes ordinais. As métricas de rede foram calculadas a partir de redes ordinais mapeadas com parâmetros de *embedding*  $d_x = d_y = 2$  e  $\tau_x = \tau_y = 1$ . Para as matrizes de ocorrência, após investigarmos combinações entre  $D \in \{1, 2, 3, 4, 5\}$  e  $\theta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$  (frequentemente considerados os melhores valores [130]), decidimos por calcular o valor médio de cada uma das cinco estatísticas das matrizes usando três pares de valores para os parâmetros  $(D, \theta)$ :  $\{(2, 0^\circ), (2, 45^\circ), (2, 90^\circ)\}$ .

Em seguida, utilizando, uma a uma, as variáveis obtidas das redes ordinais e das GLCMs, treinamos algoritmos de  $k$ -primeiros vizinhos<sup>10</sup> [132, 133] para classificar os expoentes de Hurst com estratégias de validação cruzada em 10 camadas para encontrar o melhor modelo em cada caso. Os resultados na Figura 2.6(a) mostram que *features* derivadas de redes ordinais apresentam as quatro melhores acurácias, com a entropia global dos vértices produzindo o melhor resultado entre todas as *features* utilizadas. As acurácias obtidas das melhores *features* de redes ordinais ( $\sim 70\%$ ) também são consideravelmente mais altas que as melhores *features* das GLCMs ( $\sim 40\%$ ). A Figura 2.6(b) mostra as matrizes de confusão para os modelos de primeiros vizinhos treinados com a entropia global dos vértices (redes ordinais) e com a homogeneidade (GLCMs). As bandas diagonais nessas matrizes indicam que quando os modelos classificam incorretamente o expoente de Hurst  $h$ , o valor predito é próximo do valor verdadeiro. Ainda, a banda diagonal observada para a entropia global dos vértices é muito mais estreita que aquela observada para a homogeneidade, particularmente para valores intermediários de  $h$ .

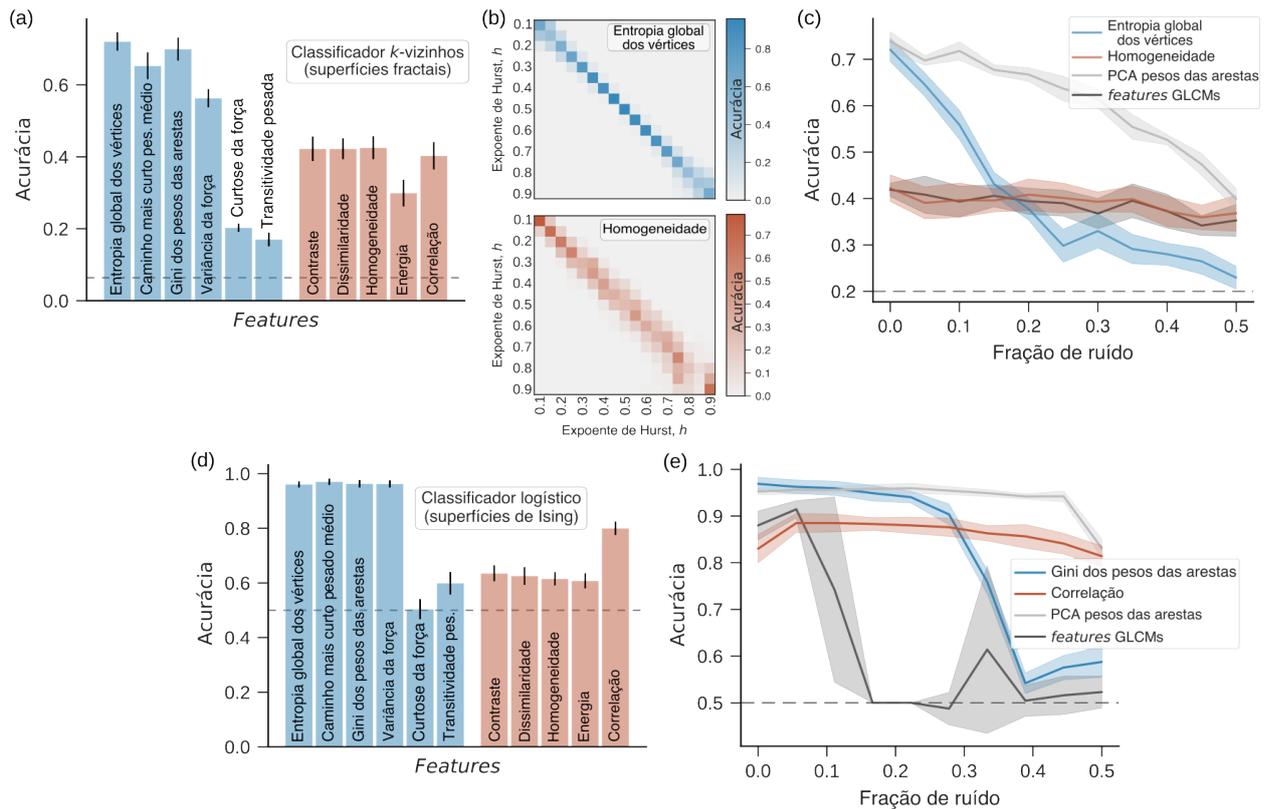
Além de precisos, quantificadores de imagens extraídos de redes ordinais também precisam ser robustos à adição de ruído, visto que imagens em aplicações práticas podem apresentar algum grau de ruído. Para investigar essa questão, consideramos, mais uma vez, a classificação das mesmas imagens fractais, mas dessa vez adicionamos ruído escolhendo uma fração de pixels e substituindo seus valores pelas intensidades máxima e mínima da imagem<sup>11</sup>. Usando esses dados, estimamos a acurácia de classificadores de primeiros vizinhos em função da fração de ruído  $\xi \in \{0, 0,05; 0,10; \dots; 0,50\}$ . A Figura 2.6(c) mostra essa análise quando utilizamos a entropia global dos vértices e a homogeneidade como variáveis preditivas. Observamos que a entropia global dos vértices tem melhor desempenho médio ( $\sim 50\%$ ) até 15% de ruído. Entretanto, a homogeneidade é mais estável em relação à adição de ruído e tem desempenho superior à entropia global dos vértices a partir de  $\sim 20\%$  de ruído. Também consideramos todas as cinco *features* das GLCM, conjuntamente, como preditores. Os resultados na Figura 2.6(c) indicam que a adição de preditores não melhora a acurácia ou a robustez à adição de ruído dos classificadores. Contrariamente, se utilizarmos cinco variáveis obtidas da aplicação do algoritmo de análise de componentes principais<sup>12</sup> (PCA<sup>13</sup>) [132, 133]

<sup>10</sup>O Apêndice F traz uma breve apresentação desse algoritmo.

<sup>11</sup>Esse tipo de ruído é conhecido como *salt-and-pepper noise*.

<sup>12</sup>O Apêndice F traz alguns detalhes sobre esse algoritmo de aprendizagem não supervisionada.

<sup>13</sup>Do inglês, *principal component analysis* (PCA).



**Figura 2.6: Comparação entre variáveis de redes ordinais e GLCMs em tarefas de classificação e robustez à adição de ruído.** (a) Acurácia (fração de classificações corretas) na determinação do expoente de Hurst de superfícies fractais com preditores obtidos de redes ordinais (em azul) e matrizes de coocorrência de nível de cinza (GLCMs – em vermelho). (b) Matrizes de confusão detalham o desempenho das entropia global dos vértices e homogeneidade como variáveis preditivas do expoente de Hurst de amostras de movimento browniano fracionário bidimensional. (c) Acurácia de classificadores de  $k$ -primeiros vizinhos que utilizam o coeficiente de Gini dos pesos das arestas, a correlação das GLCM, cinco componentes do PCA dos pesos das arestas e todas as cinco estatísticas das GLCM para prever o expoente de Hurst em função da fração de ruído adicionado a imagens fractais. (d) Acurácias de classificadores logísticos na classificação binária das superfícies de Ising acima ( $T_r = 1,01$ ) e abaixo ( $T_r = 0,99$ ) da temperatura crítica a partir de estatísticas de redes ordinais (em azul) e GLCMs (em vermelho). (e) Acurácia em tarefas de classificação relacionadas às superfícies de Ising em função das frações de ruído adicionado a essas imagens quando o coeficiente de Gini dos pesos das arestas, a correlação das GLCMs, a primeira componente principal dos pesos das arestas e todas as cinco estatísticas das GLCMs são utilizadas como preditores. Em todos os painéis, as linhas tracejadas indicam a acurácia de classificadores de linha de base que aplicam uma estratégia de previsão de acordo com a distribuição de classes (estratificado). Em todos os painéis, as barras de erro e as regiões sombreadas representam um desvio padrão estimado a partir de validações cruzadas em 10 camadas.

a todos os 416 pesos das arestas<sup>14</sup> das redes ordinais, observamos uma ligeira melhora na

<sup>14</sup>Uma aresta recebe peso zero caso não ocorra em uma rede ordinal.

acurácia das previsões na ausência de ruído e uma melhora considerável na estabilidade do classificador quando ruído é adicionado às imagens. De fato, as cinco *features* obtidas do PCA dos pesos das arestas levam a um desempenho melhor que todas as outras combinações de variáveis, mesmo com níveis de ruído de 50%.

Em uma segunda tarefa de classificação, utilizamos *features* extraídas de redes ordinais e GLCMs para prever a fase de superfícies de Ising com temperaturas abaixo ( $T_r < 1$ ) ou acima ( $T_r > 1$ ) da temperatura crítica ( $T_r = 1$ ). Para isso, geramos um conjunto de dados contendo 1000 superfícies de Ising de tamanho  $150 \times 150$  e temperaturas reduzidas iguais a  $T_r = 0,99$  (fase ordenada) e  $T_r = 1,01$  (fase desordenada). Para essa tarefa de classificação binária, treinamos um classificador logístico<sup>15</sup> e estimamos sua acurácia ao utilizar cada uma das *features* de redes ordinais ( $d_x = d_y = 2$  e  $\tau_x = \tau_y = 1$ ) e GLCMs ( $D = 5$  e  $\theta = 0^\circ$ ). A Figura 2.6(d) mostra que as quatro melhores acurácias ( $\sim 95\%$ ) são obtidas de quantificadores de imagens extraídos de redes ordinais, enquanto a estatística de correlação das GLCMs é a quinta melhor *feature*, com acurácia significativamente mais baixa ( $\sim 80\%$ ).

Estudamos também como as acurácias dessas classificações binárias são afetadas pela adição de ruído às imagens. A Figura 2.6(e) mostra a acurácia quando o coeficiente de Gini do peso das arestas e a correlação das GLCMs são utilizados como preditores em função da fração de ruído  $\xi$  adicionado às superfícies de Ising. Os resultados são similares àqueles que encontramos para as superfícies brownianas. O índice de Gini tem melhor desempenho que um preditor de linha de base que sempre imputa uma das classes às imagens (linha tracejada na Figura 2.6), mas a correlação das GLCMs é o preditor mais estável quanto à adição de ruído e tem melhor desempenho que o Gini quando a fração de ruído atinge  $\sim 30\%$  dos pixels das imagens. Incluindo todas as estatísticas das GLCMs como preditores, observamos um ligeiro aumento na acurácia (para pequenos valores da fração de ruído) comparada ao caso no qual somente a correlação das GLCMs é usada. Entretanto, isso torna essa abordagem um tanto instável em relação à adição de ruído [curva em cinza escuro na Figura 2.6(e)]. Por outro lado, uma única *feature* obtida do PCA dos pesos das arestas das redes ordinais melhora significativamente a estabilidade dos classificadores logísticos em relação ao ruído.

---

<sup>15</sup>Esse algoritmo é apresentado no Apêndice F.

---

# Aplicação de redes ordinais à predição de propriedades físicas de cristais líquidos

---

Neste terceiro capítulo, apresentamos uma aplicação da transformação de redes ordinais a texturas obtidas experimentalmente de amostras de cristais líquidos [40]. Essas texturas são mapeadas em redes ordinais que, juntamente de algoritmos de aprendizagem estatística, são utilizadas para investigar transições de fase e determinar propriedades físico-químicas desses materiais.

### 3.1 Aprendizagem de máquina, redes ordinais e cristais líquidos

O uso de métodos de aprendizagem de máquina para explorar propriedades físicas e químicas de materiais tem se tornado cada vez mais popular entre pesquisadores da Física [134–137], Química [138, 139] e Ciência dos Materiais [140–144]. Esses desenvolvimentos recentes estão intimamente relacionados a avanços importantes em computação, novos métodos de aprendizagem de máquina e a disponibilidade de grandes bases de dados. A combinação desses desenvolvimentos tem grande potencial para revolucionar o papel de métodos computacionais em pesquisa aplicada e tem sido considerada um novo paradigma em Ciência dos Materiais [145].

Uma maneira de combinar aprendizagem de máquina ao estudo de materiais consiste em investigar um conjunto de imagens extraídas de um material [30, 146, 147]. Muitos trabalhos, particularmente aqueles envolvendo materiais biológicos e complexos, utilizam técnicas ópticas para a obtenção de imagens em virtude de suas características não destrutivas. Cristais

líquidos<sup>1</sup> são um exemplo típico de material que tem sido extensivamente estudado usando métodos ópticos.

Isso acontece porque cristais líquidos são materiais birrefringentes e o uso de microscopia de luz polarizada é frequentemente suficiente para determinar diferentes propriedades desses materiais [148, 149]. Apesar disso, a aplicação de métodos de aprendizagem de máquina no estudo de cristais líquidos permanece surpreendentemente limitada e, de fato, poucos trabalhos tentaram associar propriedades físicas desses materiais a suas texturas [30, 147]. Assim como em outros problemas de aprendizagem de máquina (regressões ou classificações) envolvendo imagens, é possível explorar a física subjacente aos cristais líquidos extraindo variáveis preditivas (*features*) de suas texturas e treinando algoritmos a partir de um conjunto de exemplos de imagens desses materiais e suas propriedades físicas de interesse. O processo de extração de variáveis preditivas em problemas de regressão ou classificação comumente demanda conhecimento técnico acerca da estrutura do material [140, 150, 151] e métodos capazes de gerar essas *features* independentemente das particularidades do material são frequentemente úteis.

Nesse sentido, estatísticas derivadas do formalismo de Bandt e Pompe [6] (Seção 1.2), constituem uma interessante possibilidade. Desenvolvimentos recentes atrelados ao procedimento de simbolização de Bandt e Pompe (como aqueles mostrados no capítulo anterior), mostram que a estrutura de sucessão entre as permutações, representada na forma de uma rede ordinal, pode ser bastante efetiva na caracterização de conjuntos de imagens. Sendo assim, no restante deste capítulo, após mapear texturas de cristais líquidos em redes ordinais, utilizamos essas redes juntamente de algoritmos de aprendizagem de máquina para investigar amostras experimentais de cristais líquidos. Nossa pesquisa mostra que a combinação dessas ferramentas é capaz de identificar efetivamente transições de fase, distinguir entre transições de primeira e segunda ordem e prever concentrações de dopantes e temperaturas de amostras.

## 3.2 Apresentação dos dados e procedimentos experimentais

Para a obtenção de imagens de cristais líquidos, foram realizados experimentos<sup>2</sup> de aquecimento e resfriamento de amostras de cristais líquidos 8CB<sup>3</sup> (5 amostras) e E7<sup>4</sup> (26 amostras).

---

<sup>1</sup>O Apêndice E traz uma breve introdução aos cristais líquidos.

<sup>2</sup>Esses experimentos foram realizados em colaboração com o Prof. Rafael S. Zola, membro do Programa de Pós-Graduação em Física da Universidade Estadual de Maringá.

<sup>3</sup>4'-octil-4-bifenilcarbonitrila [152].

<sup>4</sup>O cristal líquido E7 é uma mistura de quatro cristais líquidos nas seguintes proporções: 51% 4-ciano-4'-n-pentilbifenila (5CB); 25% 4-ciano-4'-n-heptilbifenila (7CB); 36% 4-ciano-4'-n-octiloxibifenila (8OCB); 8% 4-ciano-4-n-pentil-p-terfenila (5CT) [153].

Dentre as 5 amostras de 8CB, 2 amostras eram puras e as 3 restantes foram misturadas ao dopante quiral<sup>5</sup> R811<sup>6</sup> (enantiômero dextrogiro) a uma concentração de 3,26% do peso total da amostra. As 26 amostras de E7 foram divididas em dois grandes grupos. O primeiro grupo consiste em 2 amostras puras e 10 amostras dopadas com R811 em concentrações variando entre 0,75% e 23,20% do peso total da amostra. O segundo grupo consiste em 16 amostras dopadas com R811 e S811<sup>7</sup> (enantiômero levogiro), tais que as concentrações de R811 são sempre menores que as de S811<sup>8</sup>, enquanto a concentração total dos dois dopantes é mantida fixa em 10% do peso total da amostra. A Tabela 3.1 detalha as concentrações desses dopantes em cada amostra. À temperatura ambiente, as amostras puras de 8CB encontram-se na mesofase esméctica A e na mesofase esméctica A quiral quando dopadas com R811. As amostras puras de E7 encontram-se na mesofase nemática e tornam-se cristais líquidos colestéricos quando dopadas com um aditivo quiral<sup>9</sup>. As diferentes concentrações de dopantes nas amostras de E7 implicam em diferentes passos colestéricos. Ainda, o segundo grupo de amostras de E7, dopadas com R811 e S811, permite que o passo da fase colestérica mude sem que outras propriedades físicas da mesofase sejam alteradas.

A fim de se obter texturas típicas de cada mesofase, cada amostra de cristal líquido foi introduzida em um capilar retangular de vidro de borossilicato ( $100 \mu\text{m} \times 1,0 \text{ mm}$ ) a  $60^\circ\text{C}$ . As amostras foram resfriadas até a temperatura ambiente e colocadas em um controlador de temperatura (Instec MK2000) sob um microscópio de luz polarizada (Leica). Em seguida, as amostras foram aquecidas a uma taxa de  $0,1^\circ\text{C}/\text{min}$  em intervalos de temperatura envolvendo transições de fase de primeira e segunda ordem, enquanto os cristais líquidos eram fotografados (a cada 10 segundos) por uma câmera (Leica ICC50W) acoplada ao microscópio. Após alcançar temperaturas nas quais as amostras passavam à mesofase isotrópica, as amostras foram resfriadas à taxa de  $-0,1^\circ\text{C}/\text{min}$ , tendo imagens capturadas a cada 10 segundos. Realizada a captura dessas imagens, elas foram todas cortadas no tamanho de  $800 \times 1070$  pixels e salvas em formato JPEG com 24 bits por pixel (8 bits para cada uma das cores primárias – vermelho, verde, e azul – no espaço de cores RGB).

Com o aumento da temperatura, as amostras de 8CB (puras e dopadas) passam primeiramente por transições de segunda ordem (esméctica A  $\rightarrow$  nemática, em amostras puras, e esméctica A quiral  $\rightarrow$  colestérica, em amostras dopadas) antes de passarem por transições de

---

<sup>5</sup>Um composto quiral é um composto que não pode ser perfeitamente sobreposto à sua imagem especular (assim como não podemos sobrepor nossas mãos esquerda e direita) [154]. Esses compostos estão geralmente associados à existência de um carbono quiral (átomo de carbono ligado a quatro grupos químicos distintos) e são compostos opticamente ativos, isto é, são capazes de rotacionar o plano de polarização de um feixe de luz incidente [154].

<sup>6</sup>(R)-2-Octil 4-[4-(Hexiloxi)benzoiloxi]benzoato [155].

<sup>7</sup>S-(+)-2-Octil 4-(4-hexiloxibenzoiloxi)benzoato [156].

<sup>8</sup>Os compostos R811 e S811 formam um par de moléculas de mesma composição mas de atividade óptica distinta (chamados enantiômeros ou isômeros ópticos). Incidindo-se um feixe de luz polarizada sobre um dos isômeros, aquele que desvia o plano da luz polarizada para a direita é o enantiômero dextrogiro enquanto o enantiômero levogiro desvia esse plano para a esquerda [154].

<sup>9</sup>O Apêndice E traz alguns detalhes acerca dessas diferentes mesofases.

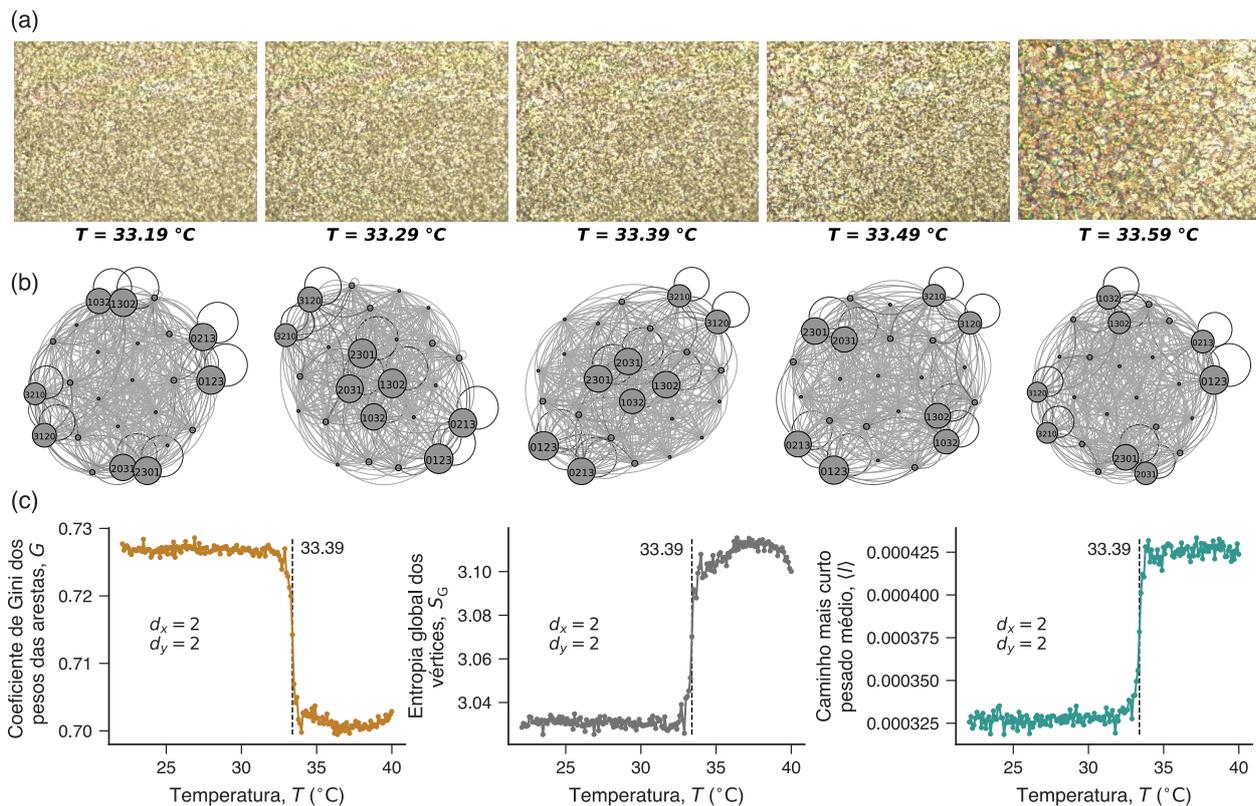
Hospedeiro	R811	S811	# Amostras	# Imagens	Mesofase
8CB	–	–	2	323	Esmética A
8CB	3,26%	–	3	495	Esmética A quiral
E7	–	–	2	316	Nemática
E7	0,75%	–	2	316	Colestérica
E7	3,14%	–	2	309	Colestérica
E7	5,50%	–	2	301	Colestérica
E7	8,00%	–	2	282	Colestérica
E7	23,20%	–	2	186	Colestérica
E7	–	10,00%	3	451	Colestérica
E7	1,00%	9,00%	3	470	Colestérica
E7	2,00%	8,00%	3	473	Colestérica
E7	3,00%	7,00%	1	150	Colestérica
E7	4,00%	6,00%	2	331	Colestérica
E7	4,85%	5,15%	2	331	Colestérica

**Tabela 3.1: Resumo das amostras de cristais líquidos 8CB e E7.** Percentuais de dopantes quirais (R811 e S811) em relação ao peso total das amostras. A ausência de um dopante é denotada por um traço. O número de imagens refere-se a todas as amostras com a mesma concentração de dopante. As fases referem-se ao estado da amostra em temperatura ambiente.

primeira ordem (nemática  $\rightarrow$  isotrópica, para amostras puras, e colestérica  $\rightarrow$  isotrópica, em amostras dopadas), enquanto amostras de E7 apresentam uma transição de primeira ordem (nemática  $\rightarrow$  isotrópica, em amostras puras, e colestérica  $\rightarrow$  isotrópica, em amostras dopadas). A Figura 3.1(a) mostra exemplos de texturas obtidas de uma amostra de 8CB dopada com 3,26% de R811 em um pequeno intervalo de temperatura englobando a transição de fase de segunda ordem. Observamos que as texturas imediatamente antes e depois da transição de fase são bastante similares, de modo que mesmo um físico experimental experiente pode ter dificuldade em identificar precisamente a temperatura crítica somente pela inspeção visual dessas imagens. Texturas obtidas de amostras de E7 e 8CB imediatamente antes de transições de primeira ordem são igualmente difíceis de distinguir visualmente.

### 3.3 Classificação de transições de fase em cristais líquidos usando redes ordinais

Apresentada a base de dados, nosso objetivo principal consiste na investigação de amostras de cristais líquidos utilizando uma estratégia que mapeia imagens dessas amostras em redes ordinais com parâmetros de *embedding*  $d_x = d_y = 2$  e  $\tau_x = \tau_y = 1$  [como mostra a Figura 3.1(b)]. Seguindo a literatura, calculamos a média das intensidades dos pixels das três



**Figura 3.1: Identificação de transições entre mesofases em amostras de cristais líquidos usando redes ordinais.** (a) Cinco texturas obtidas de uma amostra de cristal líquido 8CB dopada com 3,26% de R811 próximas à temperatura crítica  $T_c = 33,39^\circ\text{C}$  associada à transição de fase de segunda ordem entre as mesofases esméctica A quiral e colestérica [157, 158]. (b) Representações das imagens no painel (a) como redes ordinais. Os tamanhos dos nós e a intensidade de cor das arestas são proporcionais à frequência relativa das permutações e transições entre as mesmas, respectivamente. (c) Dependência de três propriedades de rede com a temperatura para redes ordinais mapeadas de uma amostra de cristal líquido 8CB dopada com 3,26% de R811. Temperaturas candidatas a temperatura crítica da amostra sugeridas pelas métricas das redes (maior mudança nessas métricas entre duas imagens consecutivas) são indicadas por uma linha vertical tracejada. Como os três valores sugeridos são o mesmo, uma estratégia de voto da maioria decide por  $T_c = 33,39^\circ\text{C}$  como temperatura crítica da amostra.

camadas RGB para representar cada imagem como uma matriz [30, 33]. Notamos que nossa escolha de parâmetros de *embedding* restringe a estrutura da rede ordinal a um máximo de 24 vértices e 416 arestas (Seção 2.2) [33].

Usando essas redes ordinais associadas às texturas, examinamos diretamente se as métricas de rede são capazes de capturar e revelar informação relacionada a transições de fase em uma situação experimental, uma vez que já demonstramos resultados positivos em sistemas simulados computacionalmente (Seção 2.6). Para tanto, procuramos identificar as temperaturas críticas em cada amostra calculando o coeficiente de Gini ( $G$ ) dos pesos das arestas, a entropia global dos vértices ( $S_G$ ) e o caminho mais curto médio pesado ( $\langle l \rangle$ ) e,

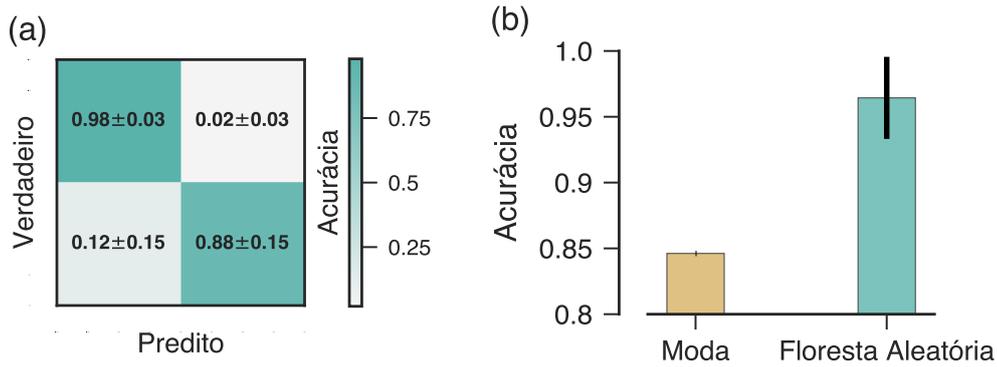
em seguida, avaliamos a variação de cada métrica entre imagens consecutivas [veja a Figura 3.1(c)]. Atribuímos às temperaturas associadas às maiores mudanças em cada uma das três métricas (para cada amostra) a condição de candidatas a temperatura crítica da amostra e usamos uma estratégia de voto da maioria para inferir a temperatura crítica de cada amostra. A Figura 3.1 ilustra esse procedimento com uma transição de segunda ordem entre as fases esmética A quiral e colestérica com o aquecimento de uma amostra dopada de 8CB (3,26% de R811). De modo geral, o procedimento previamente descrito funciona bem e é capaz de estimar temperaturas críticas bastante próximas aos valores reportados na literatura [157, 158].

Em uma sequência natural, investigamos se a informação capturada por nossas redes ordinais é capaz de discriminar entre transições de primeira e segunda ordem entre mesofases líquido cristalinas nas nossas amostras. Para isso, estabelecemos uma tarefa de classificação binária na qual texturas de cristais líquidos em temperaturas próximas à temperatura crítica são utilizadas para o treinamento de um algoritmo de floresta aleatória<sup>10</sup> para decidir se as imagens estão associadas a transições de fase de primeira ou segunda ordem. Os pesos de todas as 416 arestas<sup>11</sup> de redes ordinais com  $d_x = d_y = 2$  e  $\tau_x = \tau_y = 1$  são utilizadas como variáveis preditivas nessa tarefa de classificação. Para compor os dados analisados, selecionamos cinco imagens de cada amostra na Tabela 3.1: duas anteriores, duas posteriores e uma à temperatura crítica. Relembramos que nossas amostras apresentam quatro diferentes transições, duas de primeira ordem (nemática  $\rightarrow$  isotrópica e colestérica  $\rightarrow$  isotrópica) e duas de segunda ordem (esmética A  $\rightarrow$  nemática e esmética A quiral  $\rightarrow$  colestérica).

Como resultado, obtemos 155 imagens mapeadas em redes ordinais de cristais líquidos próximos a transições de fase, 25 representando transições de segunda ordem e 130 representando transições de primeira ordem. Separamos essas imagens em um conjunto de treinamento (75% dos dados) e um conjunto de teste (25% dos dados). Em seguida, procuramos pelo melhor modelo realizando a otimização do número de árvores de decisão (principal parâmetro desse algoritmo de aprendizagem estatística) no conjunto de treinamento via método de *grid-search* aliado a uma validação cruzada em cinco camadas [131–133]. Considerando a quantidade limitada de imagens no nosso conjunto de dados, amostramos aleatoriamente os conjuntos de treinamento e teste por 100 vezes para estimar o desempenho médio dos melhores modelos de classificação. A Figura 3.2(a) retrata a matriz de confusão associada a esse problema de classificação binária. Os resultados indicam que as redes ordinais contém informação capaz de discriminar a ordem das transições de fase. A Figura 3.2(b) mostra que a acurácia média dos classificadores de floresta aleatória é de cerca de 96%, um resultado melhor que aquele obtido de um preditor de linha de base (*dummy*) que atribui a classe mais comum no conjunto de treinamento (transição de primeira ordem, nesse caso) a todas as

<sup>10</sup>Esse algoritmo é descrito no Apêndice F.

<sup>11</sup>Uma aresta recebe peso zero caso não ocorra em uma rede ordinal.



**Figura 3.2: Classificação de transições de fase em imagens de cristais líquidos.** (a) Matriz de confusão associada à tarefa de distinguir entre transições de fase de primeira e segunda ordem em amostras de cristal líquido. Os elementos da matriz representam médias calculadas de um *ensemble* de 100 classificadores de floresta aleatória treinados e avaliados em diferentes conjuntos ( $\pm$  indica um desvio padrão). A primeira linha da matriz de confusão corresponde a transições de primeira ordem. (b) Acurácia média do *ensemble* de modelos de floresta aleatória ( $\cong 96\%$  com a barra de erro representando um desvio padrão) comparada à acurácia de um classificador de linha de base ou *dummy* ( $\cong 85\%$ ) que classifica qualquer imagem como a moda do conjunto de treinamento (transição de primeira ordem).

imagens do conjunto de teste.

### 3.4 Determinação de concentrações de dopantes em cristais líquidos usando redes ordinais

Continuando nossas investigações, focamos nos dois grupos de amostras de cristais líquidos E7 (segundo e terceiro grupos de amostras na Tabela 3.1) com o objetivo de explorar o uso de redes ordinais na classificação de amostras com diferentes concentrações de dopante, independentemente da temperatura da amostra. Todas essas amostras apresentam uma transição para a fase isotrópica e são, portanto, indistinguíveis quando alcançam essa mesofase (texturas isotrópicas são inteiramente pretas). Por isso, restringimos nossa análise a texturas de temperaturas até  $1^\circ\text{C}$  abaixo da temperatura crítica de cada amostra. Também notamos que as temperaturas críticas de cristais líquidos E7 dopados somente com R811 (segundo grupo na Tabela 3.1) diminuem com o aumento da concentração de dopante e por isso os intervalos de temperatura variam entre as amostras. Contrariamente, amostras de E7 com uma quantidade fixa de dopante (10% do peso total da amostra correspondente a R811 e S811) apresentam as mesmas temperaturas críticas e estão todas restritas a aproximadamente o mesmo intervalo de temperatura. Assim, a princípio, diferenciar entre as amostras do segundo conjunto pode ser considerado mais desafiador, uma vez que os diferentes intervalos de temperatura de amostras de E7 dopadas apenas com R811 conferem informação

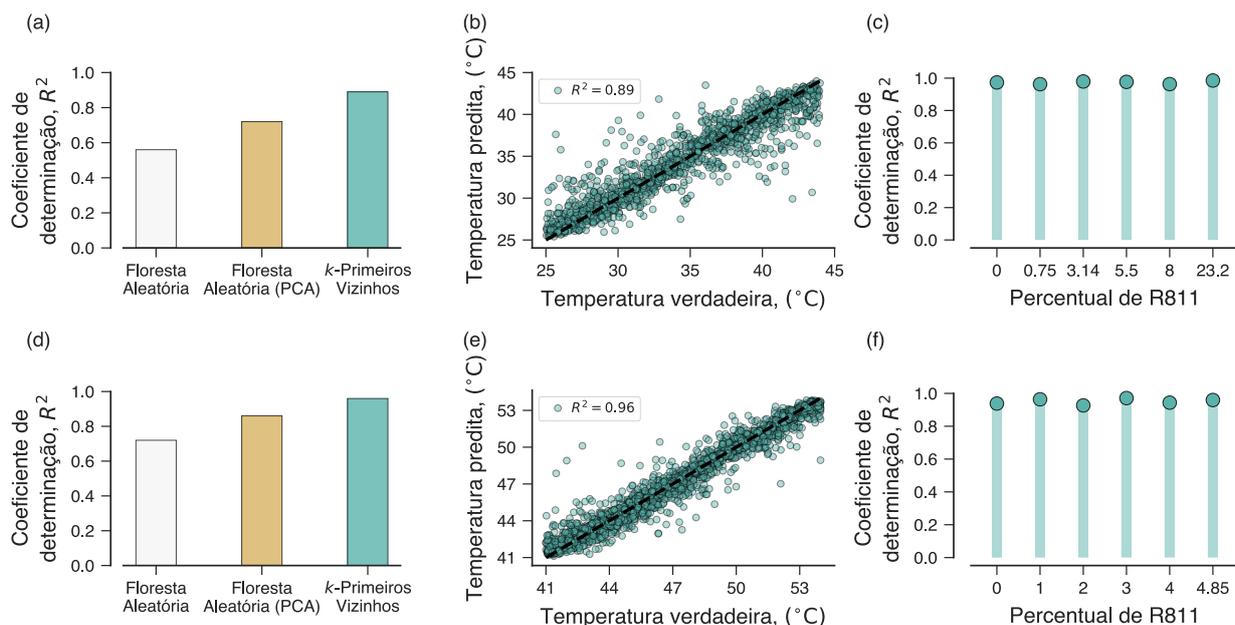


mento (75% dos dados) e teste (25% dos dados), na mesma proporção que as concentrações ocorrem no dado como um todo. Mais uma vez, usamos todos os 416 pesos das arestas das redes ordinais mapeadas dessas imagens (com  $d_x = d_y = 2$  e  $\tau_x = \tau_y = 1$ ) como *features* para treinar o algoritmo e procuramos pelo melhor modelo realizando um *grid-search* e validação cruzada em cinco camadas no conjunto de treinamento para determinar o melhor número de árvores de decisão. As Figuras 3.3(b) e 3.3(e) mostram as matrizes de confusão obtidas a partir da aplicação dos algoritmos treinados aos conjuntos de teste. Observamos que os elementos diagonais das matrizes são próximos a um, indicando que nossa abordagem classifica as amostras quase perfeitamente quanto à concentração. Ainda verificamos que a acurácia dos classificadores de floresta aleatória treinados com os pesos das arestas de redes ordinais é consideravelmente maior que as acurácias de classificadores de linha de base amparados na frequência de cada concentração ou na concentração que ocorre mais frequentemente no conjunto de treinamento, conforme mostram as Figuras 3.3(c) e 3.3(f). Assim, esses resultados mostram que as redes ordinais são igualmente capazes de prever concentrações de dopante em cristais líquidos E7 dopados apenas com R811 ou com misturas de R811 e S811.

### 3.5 Previsão da temperatura de cristais líquidos usando redes ordinais

Para continuar demonstrando a versatilidade da nossa abordagem, construímos modelos preditivos para a temperatura das amostras independentemente da concentração de dopante. Mais uma vez, consideramos separadamente as amostras de E7 dopadas apenas com R811 e aquelas dopadas com misturas de R811 e S811. Para o primeiro conjunto de amostras de E7, selecionamos todas as imagens no intervalo de temperatura entre 25°C e 44°C, um intervalo no qual nenhuma amostra passa por qualquer transição de fase. Para o segundo grupo de amostras de E7, o intervalo de temperatura é mais homogêneo (todas as amostras transitam de fase aproximadamente à mesma temperatura) e as imagens são restringidas a temperaturas entre 41°C e 54°C. Impostas essas restrições, ficamos com 898 imagens de amostras de E7 dopadas com R811 e 1454 texturas relacionadas ao segundo grupo de amostras de E7. Para aumentar o número de exemplos aos quais o algoritmo é exposto durante o processo de treinamento, dividimos cada imagem inicial em seis imagens menores de tamanho  $800 \times 175$  pixels. Em seguida, mapeamos essas imagens menores em redes ordinais com parâmetros de *embedding*  $d_x = d_y = 2$  e  $\tau_x = \tau_y = 1$ .

Separamos as imagens dos dois conjuntos de amostras em conjuntos de treinamento (75% dos dados) e teste (25% dos dados) para treinar modelos de floresta aleatória usando os pesos das arestas das redes ordinais como *features* e determinamos o melhor número de árvores de decisão usando um *grid-search* aliado a uma validação cruzada em cinco camadas. En-



**Figura 3.4: Predição da temperatura de amostras de cristais líquidos E7.** (a) Coeficiente de determinação ( $R^2$ ) de três modelos de regressão treinados em amostras de cristal líquido E7 dopadas apenas com R811: floresta aleatória ( $R^2 \cong 0,56$ ); floresta aleatória com *features* obtidas após a aplicação do método PCA às redes ordinais ( $R^2 \cong 0,72$ );  $k$ -primeiros vizinhos ( $R^2 \cong 0,89$ ). (b) Relação entre as temperaturas verdadeiras e preditas para amostras de cristal líquido E7 aplicando o modelo de  $k$ -primeiros vizinhos mostrado em (a) ao conjunto de teste. (c) Coeficiente de determinação ( $R^2$ ) para seis modelos preditivos de  $k$ -primeiros vizinhos treinados para cada uma das diferentes concentrações de dopagem no conjunto de imagens de amostras de E7 dopadas apenas com R811. (d) Coeficiente de determinação ( $R^2$ ) para três modelos de regressão treinados em amostras de E7 dopadas com misturas de R811 e S811: floresta aleatória ( $R^2 \cong 0,72$ ); floresta aleatória com *features* obtidas após a aplicação do método PCA às redes ordinais ( $R^2 \cong 0,86$ );  $k$ -primeiros vizinhos ( $R^2 \cong 0,96$ ). (e) Relação entre as temperaturas verdadeiras e preditas para amostras de cristal líquido E7 aplicando o modelo de  $k$ -primeiros vizinhos mostrado em (d) ao conjunto de teste. (f) Coeficiente de determinação ( $R^2$ ) para seis modelos preditivos de  $k$ -primeiros vizinhos treinados para cada uma das diferentes concentrações de dopagem no conjunto de imagens de amostras de E7 dopadas com misturas de R811 e S811.

tretanto, após essa etapa de treinamento, verificamos que essa abordagem não leva a bons resultados. O algoritmo de floresta aleatória “sobreajusta-se” (*overfits*) aos conjuntos de treino, comprometendo a generalização dos modelos a imagens às quais o algoritmo nunca foi exposto. Para o primeiro grupo de amostras de E7, o valor obtido para o coeficiente de determinação ( $R^2$ ) no conjunto de teste é  $R^2 \cong 0,56$  e, para o segundo grupo de amostras,  $R^2 \cong 0,72$ , conforme mostram as Figuras 3.4(a) e 3.4(d). Procurando melhorar a performance do nosso modelo, usamos a técnica de análise de componentes principais<sup>12</sup> (PCA<sup>13</sup>)

<sup>12</sup>O Apêndice F traz alguns detalhes sobre essa técnica.

<sup>13</sup>Do inglês, *principal component analysis* (PCA).

para reduzir a dimensionalidade de nossas *features* [131–133]. Reduzindo os 416 pesos das arestas a 15 novas *features* obtidas via redução de dimensionalidade (PCA *features*), repetimos o processo de treinamento do algoritmo de floresta aleatória e obtemos melhores resultados para ambos os grupos de amostras, a saber,  $R^2 \cong 0,72$  e  $R^2 \cong 0,86$ , respectivamente [Figuras 3.4(a) e 3.4(d)]. Tendo em vista o sucesso prévio em tarefas de classificação, consideramos esses resultados ainda insatisfatórios. Como o problema de sobreajuste não foi solucionado pela redução de dimensionalidade, optamos por substituir o algoritmo de floresta aleatória pelo algoritmo de  $k$ -primeiros vizinhos<sup>14</sup>, um algoritmo bastante simples e intuitivo [131–133]. Sendo assim, retornamos à nossa abordagem anterior de utilizar os pesos das arestas das redes ordinais como variáveis preditivas e treinamos o algoritmo como discutido anteriormente, com a única diferença que o parâmetro a ser otimizado passa a ser o número de primeiros vizinhos. Os resultados obtidos indicam uma melhora significativa, de modo que encontramos  $R^2 \cong 0,89$  e  $R^2 \cong 0,96$  para os primeiro e segundo grupos de amostras de cristais líquidos E7, respectivamente [Figuras 3.4(a) e 3.4(d)]. As relações entre os valores verdadeiros e preditos para a temperatura das amostras nos conjuntos de teste, estimadas com os modelos de primeiros vizinhos, são apresentadas nas Figuras 3.4(b) e 3.4(e).

Como última aplicação, propomos prever a temperatura das amostras de E7 agrupadas por concentração de R811 ou de R811 e S811 (totalizando 12 grupos, como mostrado na Tabela 3.1). Nesse caso, usamos o maior intervalo de temperatura para cada grupo de amostras. As imagens são mais uma vez subdivididas em seis imagens menores e mapeadas em redes ordinais com parâmetros de *embedding*  $d_x = d_y = 2$  e  $\tau_x = \tau_y = 1$ . Em seguida, essas mesmas imagens (redes ordinais) são separadas em conjuntos de treinamento (75% dos dados) e teste (25% dos dados) para cada um dos 12 grupos de amostras e treinamos 12 modelos de  $k$ -primeiros vizinhos com os pesos das arestas das redes. As Figuras 3.4(c) e 3.4(f) mostram que os valores de  $R^2$  (avaliados nos conjuntos de teste) são maiores que 0,92 para todos os tipos de amostras de E7 e são ainda maiores ( $R^2 > 0,96$ ) para os grupos de amostras dopados com R811 e S811.

---

<sup>14</sup>O Apêndice F traz alguns detalhes sobre esse algoritmo.

---

## Agrupamento de movimentos de papéis em queda livre usando entropia e complexidade

---

Neste último capítulo, apresentamos uma aplicação do plano complexidade-entropia a séries temporais obtidas do movimento de queda livre de pequenos pedaços de papel [159]. Aliadas a um algoritmo de agrupamento, entropia e complexidade são utilizadas para classificar diferentes tipos de movimento de queda de maneira automática e consistente.

### 4.1 O problema dos papéis em queda livre

Ao menos desde os trabalhos de Aristóteles, o movimento de queda livre tem fascinado estudiosos [160] e desempenhando um papel fundamental no desenvolvimento da Mecânica Clássica e da Gravitação, bem como de toda a Física [160, 161]. Primariamente conhecido por sua regularidade e descrição física simples, o movimento de queda livre pode tornar-se notavelmente complexo em muitas situações comuns. Esse é o caso do movimento de um simples pedaço de papel caindo pelo ar, cuja completa descrição matemática ainda permanece um grande desafio, apesar da longa história de pesquisas sobre o assunto [162–165].

A abordagem tradicional para o enfrentamento dessa complexa forma de movimento remonta ao trabalho seminal de Maxwell acerca do movimento de queda de pedaços de papel em tombos sucessivos (*tumbling*) [162], e vale-se de modelagem matemática via física básica [163, 164, 166] ou de representações desses movimentos em um espaço bidimensional cujas variáveis independentes são o momento de inércia adimensional e o número de Reynolds<sup>1</sup> para distinguir entre movimentos de queda estacionários, em tombos sucessivos (*tumbling*)

---

<sup>1</sup>O número de Reynolds  $Re$  associado ao movimento de queda livre de um pedaço de papel é definido

e caóticos [168, 169]. Tendo isso em conta, investigações mais recentes têm mostrado que o problema da queda de pequenos pedaços de papel pode ser abordado empiricamente, aliando visão computacional e métodos de aprendizagem de máquina a experimentos automatizados de larga escala [167, 170].

Os trabalhos de Howison e colaboradores [167, 170] são pioneiros em demonstrar a utilidade de um sistema automatizado de experimentação física para avançar a compreensão do movimento de queda livre de pequenos pedaços de papel. Em nosso trabalho, utilizamos seus dados experimentais para investigar e classificar séries temporais associadas a pequenos pedaços de papel em queda livre. Diferentemente de trabalhos pregressos, usamos os padrões ordinais obtidos do processo de simbolização de Bandt e Pompe para investigar o movimento de queda desses papéis. Nossa abordagem abdica da descrição matemática do movimento de queda do papel pela estimativa de duas medidas de complexidade, a saber, a entropia de permutação [6] e a complexidade estatística [51, 62], ambas associadas às distribuições de probabilidade das permutações obtidas de séries temporais da área observável dos pedaços de papel capturadas por câmeras de vídeo. O uso conjunto dessas medidas de complexidade dá origem ao chamado plano complexidade-entropia, uma ferramenta que já encontramos à Seção 1.2. Para além de uma descrição simples e direta, o uso de medidas de teoria de informação pode ser ainda motivado por conexões entre o movimento de queda de pedaços de papel e teoria do caos [163, 169].

## 4.2 Apresentação dos dados

Os dados analisados neste trabalho foram obtidos do artigo de Howison *et al.* [167] e estão disponíveis no repositório <https://github.com/th533/Falling-Paper>. Os autores da referência [167] combinam robótica e visão computacional para criar um sistema automatizado de experimentação capaz de cortar, soltar e monitorar o movimento de queda de pequenos pedaços de papel com diferentes formatos. Para o monitoramento, eles utilizam duas câmeras de alta velocidade que permitem a reconstrução da trajetória dos centros de

---

por [167]

$$Re = \frac{v_z L}{\nu},$$

com  $v_z$  a componente vertical da velocidade,  $L$  uma medida característica do comprimento do papel e  $\nu$  a viscosidade dinâmica do ar. No caso de papéis com formato circular, o diâmetro  $d$  é a medida de comprimento utilizada, assim como o lado  $l$  é utilizado para papéis quadrados. O momento de inércia adimensional  $I^*$  depende do formato do papel e é escrito como

$$I_{\text{formato}}^* = \frac{I_{\text{formato}}}{\rho_{\text{ar}} d^5}.$$

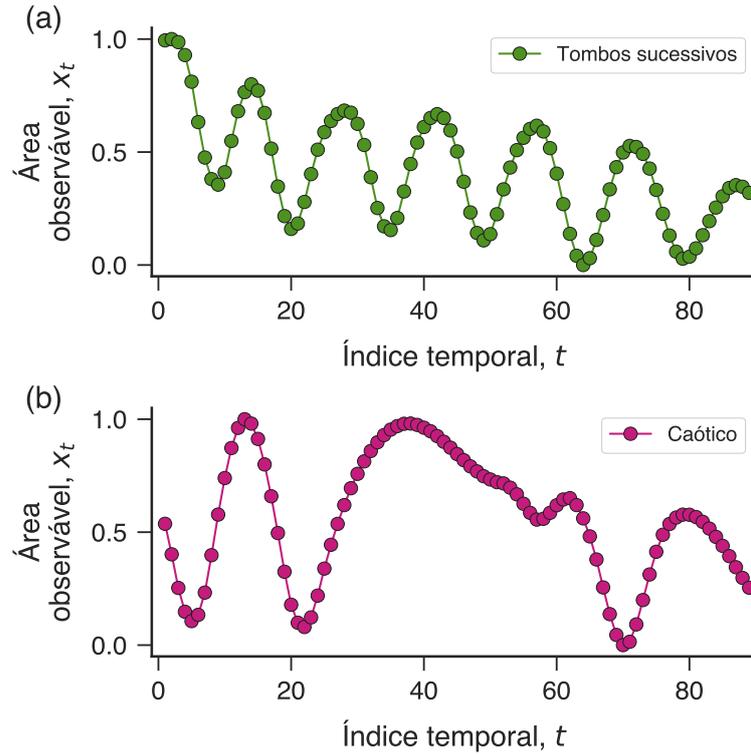
No caso de papéis em forma de círculo, o momento de inércia utilizado é aquele do disco,  $I_{\text{disco}} = \frac{mr^2}{2}$ . Ainda, na definição acima, o denominador representa o momento de inércia de uma massa esférica de ar de diâmetro próximo ao diâmetro do pedaço papel [167, 168].

Formato	# Séries temporais	Tombos sucessivos	Caótico
Círculo	170	49%	51%
Hexágono	120	39%	61%
Quadrado	92	30%	70%
Cruz	59	31%	69%

**Tabela 4.1: Resumo das séries de papéis em queda livre.** Número de séries temporais utilizadas e abundância relativa de séries exibindo movimentos em tombos sucessivos (*tumbling*) e caóticos para cada formato de papel.

massa dos papéis e também geram séries temporais de área observável, isto é, séries temporais da fração da área dos pedaços de papel que é visível (em relação à área total do papel) por cada uma das câmeras. Uma vez que o processo é inteiramente automatizado, os autores puderam monitorar e analisar o movimento de queda de centenas de papéis com formato circular, hexagonal, quadrado e de cruz, como apresentados na Tabela 4.1. Além disso, Howison *et al.* [167] oferecem uma classificação dos comportamentos de queda realizada visualmente por dois especialistas (Tabela 4.1). Os movimentos de queda são divididos em três categorias: estacionário, tombos sucessivos (*tumbling*) e caótico. Quedas estacionárias são as mais simples e diretas, sendo caracterizadas por um movimento vertical do centro de massa combinado a pequenas oscilações horizontais. Quedas em forma de tombos sucessivos (*tumbling*) são marcadas por um movimento contínuo de rotação em torno do próprio eixo combinado a um movimento lateral de translação. Em contraste, quedas caóticas alternam entre giros em torno do próprio eixo e movimentos de translação longos, rápidos e suaves, sem qualquer regularidade aparente.

Concentramos nossas investigações na distinção entre os movimentos caótico e em forma de tombos sucessivos, uma vez que o movimento estacionário é facilmente distinguível desses comportamentos mais complexos [167]. Além disso, utilizamos séries de área observável  $\{x_t\}_{t=1,\dots,N_x}$  ( $N_x$  é o número de observações da série) obtidas de apenas uma das câmeras para classificar o tipo do movimento como caótico ou em forma de tombos sucessivos. A Figura 4.1 mostra séries temporais típicas para cada uma dessas duas categorias de movimento de queda. Notamos o comportamento aproximadamente periódico da série de queda em tombos sucessivos em contraposição ao comportamento errático da série caótica. Valores de  $x_t \approx 1$  indicam que o pedaço de papel está paralelo ao plano de visão da câmera, enquanto  $x_t \approx 0$  indica que o papel está perpendicular a esse plano. As observações dessas séries temporais foram gravadas com uma frequência amostral de 98Hz (o intervalo de tempo entre observações consecutivas é de  $\approx 0,01$ s). A Tabela 4.1 mostra o número total de séries temporais usadas em nosso trabalho e os percentuais de quedas exibindo comportamentos em forma de tombos sucessivos e caóticos para cada formato de papel. Os pedaços de papel utilizados durante os experimentos têm dimensões típicas entre 5cm e 10cm [167].

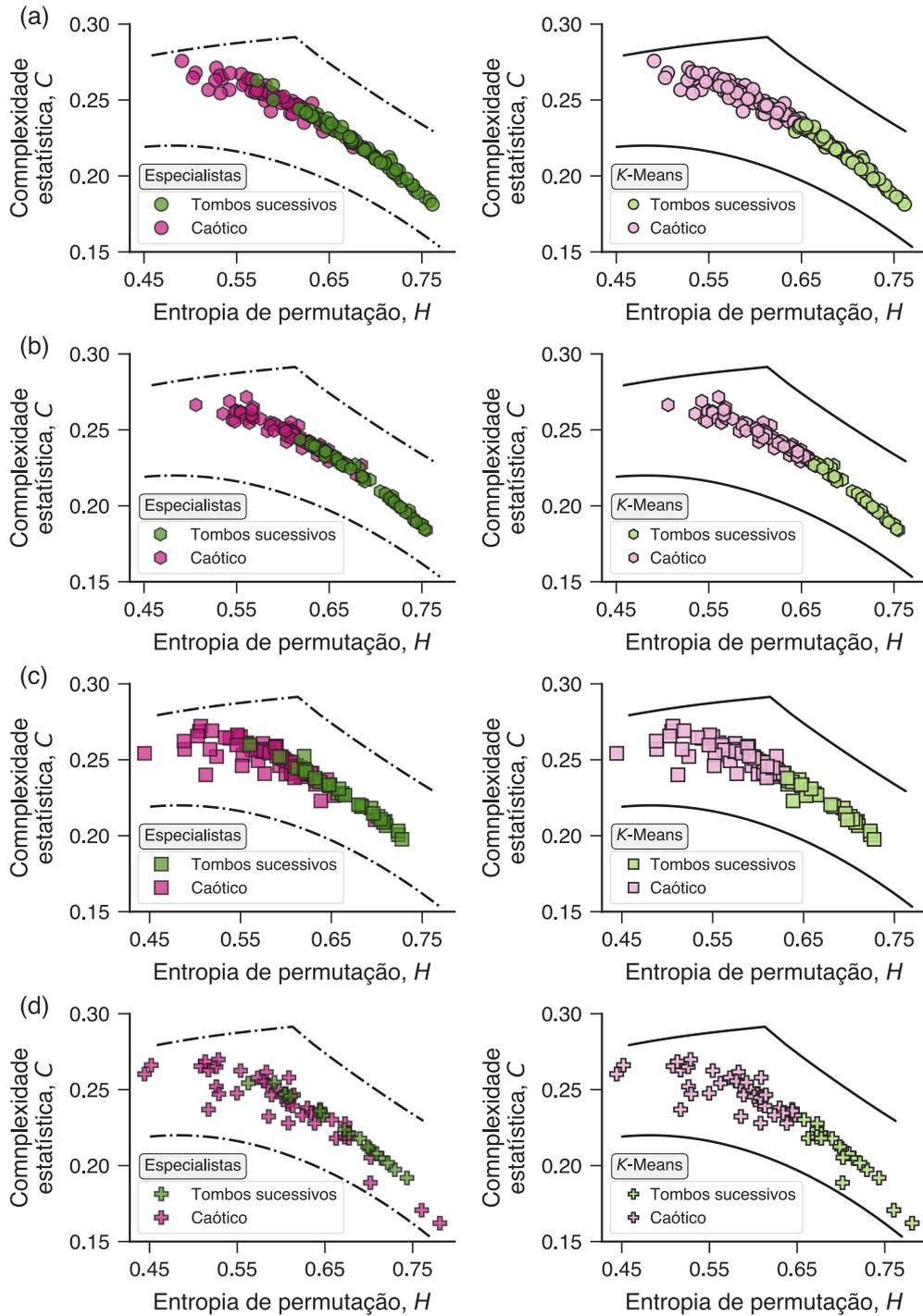


**Figura 4.1: Séries de áreas observáveis de papéis em queda livre.** Séries temporais de áreas observáveis de pequenos pedaços circulares de papel em queda livre exibindo movimentos (a) em tombos sucessivos (*tumbling*) e (b) caóticos.

### 4.3 Classificação de movimentos de queda livre

Apresentados os dados, relembramos que os parâmetros de *embedding*  $d_x$  e  $\tau_x$  precisam ser adequadamente escolhidos para que a distribuição ordinal seja apropriadamente estimada (Seção 1.2). Como as nossas séries temporais têm um comprimento médio de 110 observações, fixamos  $d_x = 3$  e  $\tau_x = 1$  em todas as nossas análises subsequentes. Para a implementação numérica do plano complexidade-entropia, utilizamos a função `complexity_entropy` do módulo `ordpy`.

Assim, calculamos os valores de entropia de permutação e complexidade estatística para todas as séries temporais de área observável após agrupá-las pelo formato do papel. A primeira coluna da Figura 4.2 apresenta a disposição dessas séries no plano complexidade-entropia para todos os formatos de papel e mostra os movimentos de queda classificados visualmente por especialistas como caóticos e em tombos sucessivos, coloridos em rosa e verde, respectivamente. Esses painéis também mostram os valores máximos (linha superior) e mínimos (linha inferior) possíveis para a complexidade em função dos valores da entropia. Observamos que séries temporais relacionadas a quedas que se dão em tombos sucessivos localizam-se em uma região do plano caracterizada por alta entropia e mais baixa complexidade quando comparadas a séries de quedas caóticas. Além disso, séries temporais caóticas



**Figura 4.2: Localização e classificação de movimentos de queda livre de pedaços de papel no plano complexidade-entropia.** Os painéis (a)-(d) mostram os valores de complexidade e entropia das séries temporais de área observável de pedaços de papel em forma de círculo, hexágono, quadrado e cruz, respectivamente. Na primeira coluna, são mostradas as classificações realizadas por dois especialistas, enquanto na coluna à direita temos classificações automáticas obtidas a partir da aplicação do algoritmo *K*-means (com  $K = 2$  grupos). Em todos os painéis, as curvas em preto indicam os valores máximos e mínimos de complexidade.

localizam-se próximas aos valores máximos de complexidade, enquanto o movimento em tomos sucessivos é praticamente equidistante dos valores máximos e mínimos de complexidade. Esses resultados assemelham-se, de certa maneira, à distinção entre mapas caóticos e processos estocásticos reportada no trabalho seminal de Rosso *et al.* [51]. Nesse trabalho, os autores encontram que séries temporais de vários mapas caóticos localizam-se próximas aos valores máximos de complexidade, enquanto séries de processos estocásticos apresentam valores intermediários de complexidade<sup>2</sup>. A Figura 4.2 mostra que esses padrões são válidos não apenas para papéis circulares mas também para os hexagonais, quadrados e em forma de cruz.

Esses resultados demonstram que séries da área observável de papéis em diferentes tipos de queda têm graus distintos de entropia e complexidade, o que pode permitir agrupá-las automaticamente com o uso de métodos de aprendizagem estatística não supervisionados. Para testar essa possibilidade, aplicamos o algoritmo de agrupamento  $K$ -means<sup>3</sup> [132, 133] aos pares de valores de entropia e complexidade referentes às séries temporais, separadas pelo formato de papel. Este método divide um conjunto de dados em  $K$  grupos por meio da otimização dos centros dos grupos para que cada observação torne-se ao mesmo tempo próxima ao centro de seu grupo e afastada dos centros dos outros grupos. A segunda coluna da Figura 4.2 mostra as separações das séries no plano complexidade-entropia em dois grupos ( $K = 2$ ) realizadas pelo algoritmo para todos os formatos de papéis. Ainda mais, uma vez que o número de grupos é pré-definido no algoritmo  $K$ -means, investigamos se a separação dos dados em dois grupos representa um particionamento ótimo. Para tanto, estimamos o índice de silhueta<sup>4</sup> [133, 171] para diferentes particionamentos realizados pelo algoritmo  $K$ -means para diferentes valores de  $K$ . Podendo variar entre  $-1$  e  $1$ , o índice de silhueta quantifica a coesão e a separação dos agrupamentos de um dado de tal forma que quanto mais alto seu valor, melhor a configuração dos grupos. Os resultados na Figura 4.3 mostram que  $K = 2$  maximiza o valor do índice de silhueta para todos os formatos de papel e, assim, o particionamento em dois grupos pode ser considerado ótimo em termos dessa métrica. Chegamos à mesma conclusão para todos os formatos de papel quando usamos o “método do cotovelo”<sup>5</sup> [133] para avaliar alterações no coeficiente de inércia (soma de quadrados das distâncias ao centros dos grupos) dos agrupamentos (veja a coluna à direita na Figura 4.3).

Podemos ainda quantificar o desempenho da nossa abordagem de agrupamento automático comparando os grupos obtidos a partir do plano complexidade-entropia com a classificação realizada pelos especialistas [167]. Essa concordância é de 87% para os pedaços quadrados de papel, 84% para pedaços circulares e hexagonais e de 69% para aqueles em

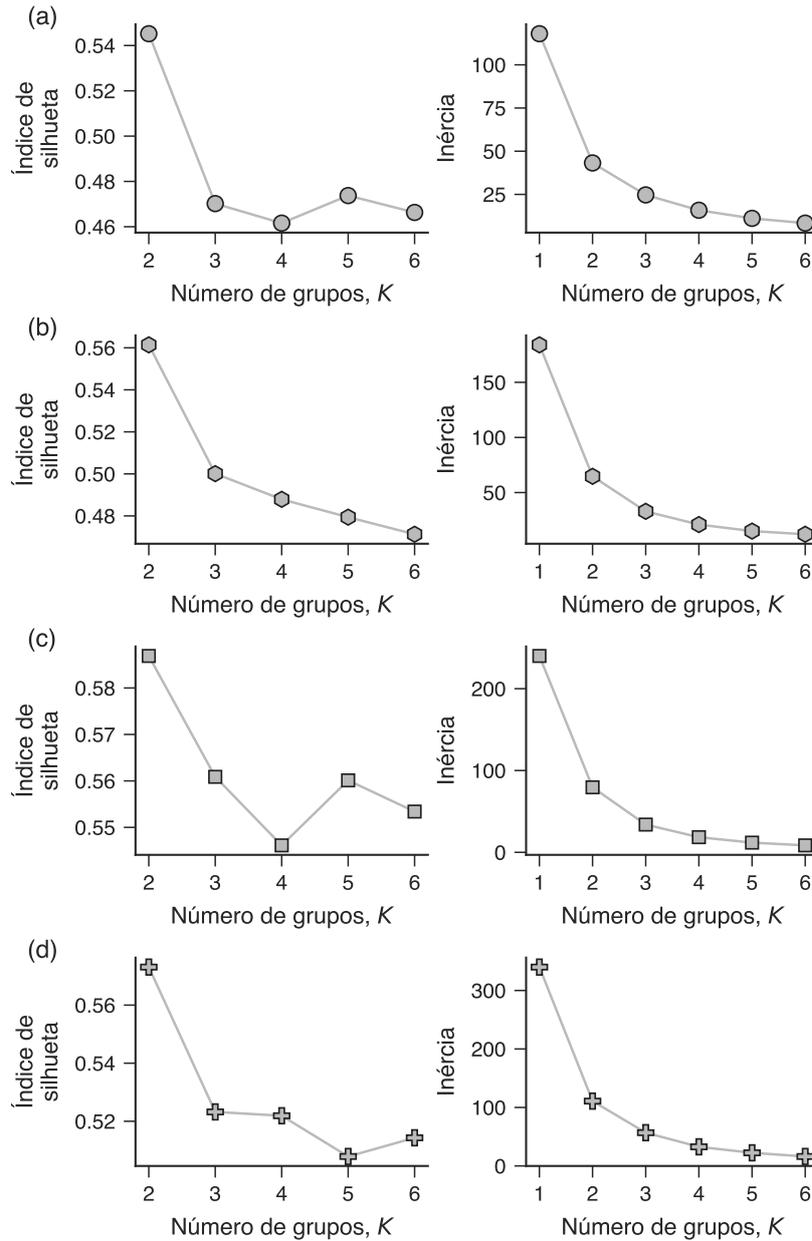
---

<sup>2</sup>Replicamos esses resultados na Figura 1.4 ao apresentarmos o módulo `ordpy`.

<sup>3</sup>O Apêndice F traz uma descrição desse algoritmo de aprendizagem não supervisionada. Aqui, utilizamos a implementação desse algoritmo contida no módulo Python `scikit-learn` [131].

<sup>4</sup>O índice de silhueta é definido no Apêndice F.

<sup>5</sup>Em inglês, *elbow method*. Esse método é descrito no Apêndice F.



**Figura 4.3: Avaliação dos agrupamentos do algoritmo  $K$ -means.** Índice de silhueta (à esquerda) e inércia (à direita) das separações das séries temporais de área observável em  $K$  grupos. Os painéis (a)-(d) mostram os resultados para o agrupamentos de séries de pedaços de papel em forma de círculo, hexágono, quadrado e cruz, respectivamente. Observamos que  $K = 2$  resulta no melhor particionamento de acordo com o coeficiente de silhueta e também com o “método do cotovelo”.

formato de cruz. Esses percentuais de concordância são significativamente mais altos que aqueles obtidos ao aplicar o algoritmo  $K$ -means a versões embaralhadas das séries temporais de áreas observáveis (concordância média próxima a escolhas aleatórias), o que confirma que as medidas de complexidade e entropia de permutação são úteis para distinguir entre esses dois tipos de movimento. Nossos resultados também são melhores que a simples estratégia de

imputar a todas as séries temporais, agrupadas pelo formato do papel, seu comportamento mais comum (Tabela 4.1) e bastante similares àqueles reportados por Howison *et al.* [167], que agruparam os movimentos de queda a partir de características como a velocidade vertical do papel  $v_z$  e sua frequência de rotação ao cair  $\omega$ . Todavia, vale notar que o cálculo da velocidade de queda depende do processamento da trajetória tridimensional dos pedaços de papel [167], enquanto valores de entropia e complexidade são estimados diretamente das áreas observáveis dos papéis filmados durante as quedas (uma série unidimensional obtida mais facilmente que a trajetória completa do centro de massa). Finalmente, como também defendido por Howison *et al.* [167], vale ressaltar que a distinção visual entre movimentos de queda caóticos e em tombos sucessivos é, por vezes, ambígua. Especialistas nem sempre concordam acerca dos comportamentos que observam, o que justifica nossa investigação não supervisionada da localização de séries de queda em tombos sucessivos e caóticas no plano complexidade-entropia. Assim, nossa classificação baseada no plano complexidade-entropia é menos subjetiva e pode ajudar a reduzir a ambiguidade nas classificações de especialistas.

---

## Conclusões e Perspectivas

---

Iniciamos este trabalho realizando uma revisão da literatura relacionada à entropia de permutação e a todo o formalismo que se seguiu desta medida de complexidade. Conjuntamente a essa revisão, apresentamos o pacote Python `ordpy` [39] por meio da replicação de resultados importantes da literatura de métodos ordinais. Tendo em mente um novo paradigma que avança na ciência, decorrente do uso e disponibilidade de grandes bases de dados [172–175], características como a simplicidade e escalabilidade dos métodos ordinais tornam-os ferramentas bastante atrativas. Assim, acreditamos que o pacote `ordpy` pode colaborar com uma maior popularização dos métodos ordinais, principalmente em áreas de pesquisa com menos tradição em computação científica. Esperamos também que esse pacote fomente a reprodutibilidade e acessibilidade desses métodos, da mesma forma que outros *softwares* de código aberto, tais como `tisean` [176] (análise de séries temporais não lineares), `pyunicorn` [177] (redes de séries temporais e análise de recorrência) e `powerlaw` [178] (ajuste e análise de distribuições de cauda pesada) têm feito.

No restante deste trabalho, dirigimos grande parte de nossa atenção às redes ordinais, um desenvolvimento mais recente no contexto dos métodos ordinais. Começamos nossas investigações com essas redes apresentando uma nova abordagem para mapear dados bidimensionais (imagens) em redes ordinais [33]. Além disso, apresentamos uma análise sobre as restrições de conectividade em redes ordinais mapeadas de dados bidimensionais e desenvolvemos um método numérico para encontrar a estrutura de redes ordinais mapeadas de dados aleatórios. Avaliamos a aplicação dessa transformação a imagens de um ornamento geométrico periódico, de movimento browniano fracionário bidimensional e de superfícies de Ising geradas via simulações computacionais. Em todos os casos, a observação do comportamento de algumas propriedades das redes ordinais mostrou-se muito eficaz em identificar alterações nas dinâmicas dos sistemas. Quando aplicadas a um conjunto de texturas reais, as redes ordinais foram capazes de identificar simetrias simples nessas imagens. Em dois cenários de classificação, com superfícies brownianas fracionárias e superfícies de Ising, esta-

tísticas obtidas de redes ordinais demonstraram maior capacidade preditiva em comparação a estatísticas obtidas de matrizes de coocorrência de nível de cinza (GLCMs), um método tradicional de análise de imagens. A adição de ruído a essas superfícies testou a resiliência das redes ordinais e revelou a possibilidade de combinação dos pesos das arestas à análise de componentes principais na geração de novos preditores. Enfim, ressaltamos que mapeamentos de séries temporais [34] ou imagens [117, 118] em redes complexas para fins de caracterização representam um desenvolvimento recente dentro da ciência de redes e nossos resultados contribuem para o avanço dessas abordagens.

Progredimos ainda mais com aplicações de redes ordinais a imagens, mais especificamente, a texturas de cristais líquidos obtidas experimentalmente [40]. Verificamos que os pesos das arestas de redes ordinais representando imagens de amostras de cristais líquidos aliados a algoritmos de aprendizagem de máquina são uma combinação capaz de classificar transições de fase e concentrações de dopantes, além de prever a temperatura dessas amostras com boa precisão. Comparando nossos resultados àqueles obtidos da aplicação de redes neurais convolucionais a imagens de cristais líquidos [147], concluímos que nossa abordagem produz acurácias comparáveis em classificações e números ligeiramente piores em regressões. De modo geral, o sucesso que alcançamos em diferentes problemas e cenários demonstra a versatilidade da transformação de redes ordinais como ferramenta de extração de *features* de imagens, especialmente no que toca imagens de materiais físicos complexos.

Em uma última aplicação do formalismo de Bandt e Pompe, séries temporais de área observável obtidas de movimentos de queda livre de pequenos pedaços de papel foram mapeadas em pontos no plano complexidade-entropia. Em seguida, aplicamos o algoritmo de agrupamento  $K$ -means a esses valores de complexidade e entropia a fim de separar quedas caóticas de quedas que se dão em tombos sucessivos. Os grupos encontrados pelo algoritmo apresentaram boa concordância com classificações realizadas por um par de especialistas e mostram que o plano complexidade-entropia funciona como um espaço de representação no qual movimentos de queda em meios viscosos podem ser convenientemente discriminados. Notamos ainda que a substituição da complexidade estatística pela medida de informação de Fisher [179] ou pela recém-proposta divergência de permutação de Jensen-Shannon [180], pode levar a resultados bastante similares aos que apresentamos. Tudo isso demonstra a versatilidade e robustez do formalismo de Bandt e Pompe, uma vez que diferentes medidas de complexidade são capazes de produzir análises apropriadas dessas séries temporais empíricas, curtas e não lineares.

Finalmente, este trabalho proporciona uma perspectiva geral do formalismo de Bandt e Pompe, o que facilita a identificação de novas oportunidades de investigação e aplicação de métodos ordinais. Sendo assim, pretendemos continuar aumentando o número de métodos ordinais implementados no pacote `ordpy`, bem como continuar aplicando métodos ordinais a novos conjuntos de dados, especialmente de imagens.

---

## Seleção de parâmetros de *embedding* na análise de séries temporais

---

Os parâmetros de *embedding* ( $d_x$  e  $\tau_x$ ) são importantes em várias aplicações relacionadas ao formalismo de Bandt e Pompe, de modo que escolhas equivocadas desses parâmetros podem levar a conclusões incorretas. Ao mesmo tempo, não existe um único procedimento para a seleção de valores ótimos para esses parâmetros e essa escolha depende da natureza da série temporal e da motivação da pesquisa. No contexto da entropia de permutação, Myers *et al.* [181] sugerem três estratégias principais: *i)* seguir as sugestões de especialistas; *ii)* tentativa e erro; *iii)* utilizar métodos de reconstrução do espaço de fase relacionados à análise de séries temporais não lineares.

A primeira estratégia, consistindo em seguir boas práticas previamente estabelecidas na literatura, tem como bons pontos de partida os artigos de revisão sobre entropia de permutação e métodos afins [182–185]. O trabalho de Riedl *et al.* [183] é particularmente interessante visto que os autores compilam diferentes escolhas de parâmetros de *embedding* de acordo com as características da série temporal e área de pesquisa. Entre outras proposições, esses autores sugerem a utilização de  $\tau_x = 1$  e da maior *embedding dimension* que possibilita uma avaliação apropriada da distribuição ordinal para dados mais facilmente descritos por modelos discretos.

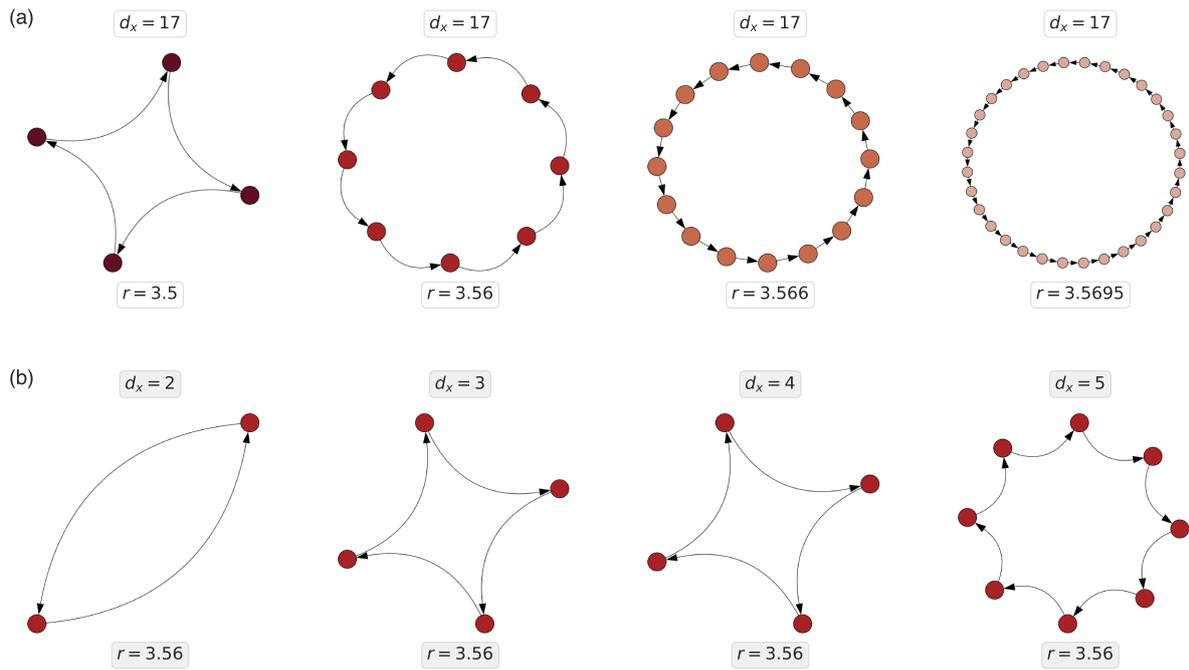
A segunda estratégia vale-se da origem computacional do formalismo de Bandt e Pompe e está em acordo com as estratégias de otimização de parâmetros em aprendizagem de máquina [132, 133], nas quais a seleção de parâmetros ótimos é frequentemente alcançada via experimentação (tentativa e erro), heurística e validação usando modelos nulos. Acreditamos que essa estratégia seja fundamental em aplicações envolvendo tarefas de regressão e classificação, nas quais os parâmetros de *embedding* ótimos podem ser encontrados minimizando funções de perda usando validação cruzada e conjuntos de treino e teste [26]. Uma outra

heurística, proposta por Kulp *et al.* [186], sugere o uso de *ensembles* de séries temporais aleatórias com o mesmo comprimento da série temporal sob análise e selecionar a maior *embedding dimension* para a qual o número de padrões ordinais faltantes é zero.

A terceira estratégia para a seleção de parâmetros de *embedding* refere-se ao uso de métodos derivados ou relacionados à análise de séries temporais não lineares [7, 8, 187]. Técnicas comuns, tais como procurar pelo primeiro zero da função de autocorrelação ou o primeiro mínimo da informação mútua, podem ser especialmente interessantes na escolha de  $\tau_x$  [7, 187]. Vale a pena lembrar que os parâmetros de *embedding* do procedimento de simbolização de Bandt e Pompe estão intimamente relacionados à ideia de *embedding* e reconstrução de espaço de fase no contexto de sistemas dinâmicos [188–191]. De fato, investigações baseadas em métodos ordinais no contexto de dinâmica caótica são parte instrumental do desenvolvimento do formalismo de Bandt e Pompe [6, 31, 51]. Nesse contexto, uma representação simples e interessante de como a *embedding dimension* se relaciona ao espaço de fase subjacente é apresentada por Groth [192].

Além dessas três estratégias, existem tentativas de desenvolvimento de procedimentos automáticos para a seleção de parâmetros de *embedding* em aplicações da entropia de permutação a séries temporais [181, 183, 193]. Ressaltamos a importante comparação entre recomendações de especialistas e recomendações automatizadas apresentada por Myers *et al.* [181].

Considerando os desenvolvimentos mais recentes relacionados ao mapeamento de séries temporais em redes ordinais, vários trabalhos investigaram a seleção de parâmetros de *embedding* ótimos [31, 86, 89, 194]. Nesse contexto, propriedades topológicas das redes tornam-se critérios importantes para a seleção apropriada de parâmetros de *embedding* capazes de capturar características da dinâmica do sistema. Para ilustrar essa abordagem, reproduzimos parcialmente os resultados de Sakellariou *et al.* [89] (veja a Figura 8 desta referência) acerca de redes ordinais mapeadas de séries periódicas do mapa logístico com período  $2^k$  para  $k \in \{2, 3, 4, 5\}$ . A Figura A.1(a) mostra uma representação dessas redes mapeadas com  $d_x = 17$  e  $\tau_x = 1$ . Observamos que esse valor de  $d_x$  é suficientemente grande para mapear o comportamento periódico dessas séries temporais em redes regulares (de estrutura circular, semelhantes a anéis) com o número de nós precisamente igual ao período da série temporal. Conforme discutido por Sakellariou *et al.* [89], a *embedding dimension* precisa ser maior que  $2^{k-1}$  para que a topologia da rede represente explicitamente o período  $2^k$  da série temporal. A Figura A.1(b) ilustra o que acontece com redes ordinais mapeadas de uma série temporal com período de 8 observações ( $k = 3$ ) para diferentes valores de  $d_x$ . Os resultados claramente confirmam que a topologia da rede somente alude corretamente ao período quando  $d_x > 4$ . Problemas similares envolvendo escolhas não ótimas de  $d_x$  emergem quando usamos redes ordinais para estimar invariantes dinâmicos como a entropia topológica [195].



**Figura A.1: Redes ordinais mapeadas de séries logísticas periódicas.** (a) Redes ordinais mapeadas a partir de séries logísticas periódicas ( $N_x = 100$ ) com parâmetros  $r = 3,5$  (período 4),  $r = 3,56$  (período 8),  $r = 3,566$  (período 16) e  $r = 3,5695$  (período 32). Todas as quatro redes são obtidas usando parâmetros de *embedding*  $d_x = 17$  e  $\tau_x = 1$ . (b) Redes ordinais mapeadas a partir de uma série do mapa logístico de período 8 ( $r = 3,56$ ) com *embedding dimensions*  $d_x \in \{2, 3, 4, 5\}$  e  $\tau_x = 1$ . Notamos que  $d_x$  deve ser maior que 4 para que o número de nós seja igual ao período da série. Ainda mais, a estrutura dessa rede não é modificada quando consideramos valores de  $d_x$  maiores que 4 (como mostrado pela última rede do painel (b) e pela segunda rede do painel (a)).

## APÊNDICE B

---

### Sistemas dinâmicos

---

Neste apêndice, apresentamos brevemente as definições dos sistemas dinâmicos<sup>1</sup> mencionados ao longo deste trabalho.

1. O mapa<sup>2</sup> logístico é definido pela seguinte equação de diferença [197]:

$$x_{t+1} = rx_t(1 - x_t), \quad (\text{B.1})$$

com  $r$  um parâmetro adimensional.

2. O mapa logístico transiente é definido como [47, 198]

$$x_{t+1} = r(t)x_t(1 - x_t), \quad (\text{B.2})$$

com o parâmetro  $r(t)$  sendo incrementado a cada iteração.

3. O *skew tent map* é dado por  $x_{t+1} = F(x_t)$  sendo [199]

$$F(x) = \begin{cases} x/\omega & \text{para } x \in [0, \omega] \\ (1-x)/(1-\omega) & \text{para } x \in [\omega, 1] \end{cases} \quad (\text{B.3})$$

---

<sup>1</sup>Um sistema dinâmico é um sistema definido por algum tipo de regra (comumente, equações diferenciais ou de recorrência) e que, estando disponível informação suficiente de seu estado atual, podemos aplicar aquela regra para prever seus estados futuros [187]. Outra característica desses sistemas é que sua evolução temporal é descrita em um espaço de fase definido pelas variáveis que caracterizam o estado do sistema [7].

<sup>2</sup>Um mapa (ou equação de diferença) é um exemplo de sistema dinâmico no qual o tempo é uma variável discreta [7, 196]. De um modo geral, um mapa é escrito na forma

$$\mathbf{x}_{t+1} = \mathbf{F}(\mathbf{x}_t),$$

com  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  e  $\mathbf{F}$  um campo vetorial de mesma dimensão [7, 196].

e  $\omega$  um parâmetro.

4. O mapa de Hénon é definido como [200]

$$\begin{cases} x_{t+1} = 1 - ax_t^2 + y_t \\ y_{t+1} = bx_t \end{cases}, \quad (\text{B.4})$$

sendo  $a$  e  $|b| < 1$  seus parâmetros. Esse mapa pode ser pensado como uma extensão bidimensional do mapa logístico [200].

5. O mapa de Schuster é definido por [200]

$$x_{t+1} = (x_t + x_t^z) \bmod 1, \quad (\text{B.5})$$

com  $z$  um parâmetro. A operação “mod” retorna a parte fracionária (decimal) de um número.

6. O sistema de Rössler é um sistema dinâmico no qual o tempo é uma variável contínua<sup>3</sup> sendo definido pelo conjunto de equações diferenciais acopladas [201, 202]

$$\begin{aligned} \frac{dx}{dt} &= -y - z \\ \frac{dy}{dt} &= x + ay \\ \frac{dz}{dt} &= b + z(x - c) \end{aligned}, \quad (\text{B.6})$$

no qual  $a$ ,  $b$ , e  $c$  são parâmetros. Esse sistema de equações foi resolvido numericamente usando o pacote Python `scipy` [203].

---

<sup>3</sup>Sistemas dinâmicos nos quais o tempo é uma variável contínua são descritos por conjuntos de equações diferenciais de primeira ordem do tipo

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{F}(\mathbf{x}(t)).$$

Esses sistemas dinâmicos também são chamados de fluxos [7, 196]. Na equação acima,  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  e  $\mathbf{F}$  é um campo vetorial de mesma dimensão. No caso do sistema de Rössler, em que o campo vetorial não depende explicitamente do tempo, o sistema é ainda chamado de autônomo [7, 196].

---

## Processos estocásticos

---

Neste apêndice apresentamos brevemente as definições dos processos estocásticos<sup>1</sup> mencionados ao longo deste trabalho.

1. Uma superfície de Ising [123, 124] é uma rede quadrada na qual a altura de cada sítio da rede representa a soma acumulada de variáveis de *spin* em uma simulação de Monte Carlo [70]. Assumindo que  $\sigma_i \in \{-1, 1\}$  representa a variável de *spin* no sítio  $i$ , podemos escrever a hamiltoniana do sistema como

$$\mathcal{H} = - \sum_{\langle i,j \rangle} \sigma_i \sigma_j, \quad (\text{C.1})$$

com o somatório envolvendo apenas os primeiros vizinhos. A amplitude  $S_i$  no sítio  $i$  da superfície de Ising correspondente é definida por

$$S_i = \sum_t \sigma_i(t), \quad (\text{C.2})$$

sendo  $\sigma_i(t)$  o valor assumido pela variável de *spin* no passo  $t$  da simulação de Monte Carlo. Para cada superfície, definimos a temperatura reduzida  $T_r$  como sendo a razão entre a temperatura  $T$  e a temperatura crítica  $T_c$  do sistema de Ising [ $T_c = 2/\ln(1 + \sqrt{2})$ ]. Utilizamos condições periódicas de contorno em nossas simulações.

---

<sup>1</sup>Um processo estocástico (ou processo aleatório) é uma coleção de variáveis aleatórias  $\{X(t), t \in T\}$ , indexadas por uma variável  $t$  [204]. De maneira mais geral, nos casos em que as variáveis  $t$  e/ou  $X$  são multidimensionais, o processo estocástico é dito ser um campo aleatório [205]. Quando o conjunto  $T$  dos índices é um conjunto enumerável, o processo é dito discreto [204]. Caso  $T$  seja um intervalo da reta real (ou uma região de um espaço multidimensional) o processo é contínuo [204].

2. O movimento browniano fracionário é um processo estocástico contínuo, autossimilar e não estacionário introduzido por Mandelbrot e Van Ness [206]. O expoente de Hurst  $h \in (0, 1)$ , seu único parâmetro, controla a rugosidade observada em séries temporais desse processo, de modo que quanto menor o valor de  $h$  mais rugosa é a série temporal. O caso em que  $h = 1/2$  corresponde ao movimento browniano usual (ruído branco gaussiano integrado). Para gerar séries temporais desse processo estocástico, empregamos o método de Hosking [207].

No caso bidimensional, e seguindo o método *turning bands* [122], amostras de superfícies brownianas fracionárias são geradas, a partir de amostras unidimensionais, da seguinte maneira. Primeiramente, o espaço bidimensional é dividido por  $L$  retas que formam um ângulo  $\theta$  entre si (bem como com o eixo  $x$ ), sendo o vetor diretor da  $i$ -ésima reta denotado por  $\vec{u}_i$ . Em seguida, na direção determinada pela  $i$ -ésima reta, uma série temporal browniana fracionária  $Z_i$  é colocada na origem do plano cartesiano. Essa série é ainda duplicada e revertida de modo a ocupar a outra metade da  $i$ -ésima reta (na direção do vetor  $-\vec{u}_i$ ). Considerando um vetor  $\vec{x}$  que localiza um ponto  $X$  no plano  $xy$ , a amplitude  $B_h(\vec{x})$  de uma superfície browniana fracionária é dada por

$$B_H(\vec{x}) = \frac{1}{\sqrt{L}} \sum_{i=1}^L Z_i(\vec{x} \cdot \vec{u}_i). \quad (\text{C.3})$$

Para gerar as superfícies brownianas fracionárias utilizadas neste trabalho, usamos o método *turning bands* conforme implementado na referência [208].

3. Um ruído gaussiano fracionário é um processo estocástico estacionário que representa os incrementos do movimento browniano fracionário. Para esse processo gaussiano, o parâmetro de Hurst  $h \in (0, 1)$  controla o alcance das correlações temporais do processo. Para  $h > 1/2$ , o processo apresenta memória de longo alcance marcada por um comportamento persistente. Para  $h < 1/2$ , o processo apresenta correlações de curto alcance marcadas por um comportamento antipersistente. Caso  $h = 1/2$ , recuperamos o ruído branco gaussiano. Para gerar séries temporais de ruído gaussiano fracionário, também usamos o método de Hosking [207, 209]. Informações detalhadas acerca de toda a teoria envolvida em simulações de ruído fracionário gaussiano e movimento browniano fracionário a partir do método de Hosking podem ser encontradas na referência [104]. O código fonte, escrito na linguagem programação C, que foi usado nesse trabalho, está publicamente disponível na referência [210].

4. Ruídos  $1/f$  [211, 212] são uma classe de processos estocásticos que tem por característica uma densidade espectral que decai como lei de potência na forma  $1/f^k$  [211–213], sendo  $k \geq 0$  um parâmetro. O caso  $k = 0$  corresponde ao ruído branco, enquanto  $k = 2$  corresponde ao ruído marrom (caminhada aleatória ou ruído branco integrado). Geramos ruídos  $1/f^k$  gaussianos para  $k \in \{0; 0,25; 0,50; \dots; 3,0\}$  usando o algoritmo proposto por Timmer e König [213] conforme implementado na referência [214].

5. O ornamento geométrico periódico usado neste trabalho pode ser representado como uma matriz quadrada  $\mathbf{Z}$ . Para defini-lo, primeiramente construímos duas matrizes  $\mathbf{X}$  e  $\mathbf{Y}$  cujos elementos são dados por

$$X_{ij} = \frac{2\pi(j-1)}{(n-1)} \quad \text{e} \quad Y_{ij} = \frac{2\pi(i-1)}{(n-1)}. \quad (\text{C.4})$$

A partir dessas definições, os elementos  $Z_{ij}$  da matriz que representa o ornamento são dados por

$$Z_{ij} = \text{sen} \left( \frac{\omega}{2\pi} X_{ij} \cos \theta - \frac{\omega}{2\pi} Y_{ij} \text{sen} \theta \right), \quad (\text{C.5})$$

com  $i, j = 1, \dots, N$  ( $N$  o tamanho da matriz),  $\theta$  definindo o ângulo entre as listras do ornamento e a horizontal e  $\omega \in \mathbb{Z}_+$  a frequência dessas listras. O ornamento mostrado na Figura 1.9(e) é obtido com os parâmetros  $N = 250$ ,  $\omega = 9$  e  $\theta = 135^\circ$ . Apesar de ser completamente determinístico, podemos introduzir aleatoriedade nesse ornamento [28, 33, 125]. Nesse caso, introduzimos um parâmetro  $p$  para controlar a probabilidade de que elementos da matriz sejam embaralhados aleatoriamente. Uma função implementando esse ornamento geométrico está disponível no *notebook* de exemplos do pacote `ordpy` [39].

---

Matrizes de coocorrência de nível de cinza

---

As matrizes de coocorrência de nível de cinza (GLCMs) [127, 128] constituem uma abordagem muito popular de análise de texturas. Essa abordagem consiste do cálculo de matrizes  $m_{\theta,D}(i, j)$  que representam a frequência relativa de coocorrência de intensidades de pixels  $i$  e  $j$  [com  $0 \leq (i, j) \leq N_{\text{cinza}}$  e  $N_{\text{cinza}} = 255$  para imagens 8 bits] à distância  $D$  e ângulo  $\theta$ . A partir das GLCMs, podemos calcular várias estatísticas descritivas de uma textura [128]. Neste trabalho, utilizamos cinco medidas comumente usadas e que estão implementadas no pacote Python `scikit-image` [126]. Essas medidas são:

$$\text{contraste}(\theta, D) = \sum_{i,j}^{N_g} (i - j)^2 m_{\theta,D}(i, j), \quad (\text{D.1})$$

$$\text{dissimilaridade}(\theta, D) = \sum_{i,j}^{N_g} |i - j| m_{\theta,D}(i, j), \quad (\text{D.2})$$

$$\text{homogeneidade}(\theta, D) = \sum_{i,j}^{N_g} \frac{m_{\theta,D}(i, j)}{1 + (i - j)^2}, \quad (\text{D.3})$$

$$\text{energia}(\theta, D) = \sqrt{\sum_{i,j}^{N_g} m_{\theta,D}(i, j)^2}, \quad (\text{D.4})$$

e

$$\text{correlação}(\theta, D) = \sum_{i,j}^{N_g} m_{\theta,D}(i, j) \frac{(i - \mu_x)(j - \mu_y)}{\sigma_x \sigma_y}, \quad (\text{D.5})$$

sendo

$$\mu_x = \sum_{i,j}^{N_g} i m_{\theta,D}(i, j),$$

$$\mu_y = \sum_{i,j}^{N_g} j m_{\theta,D}(i, j),$$

$$\sigma_x^2 = \sum_{i,j}^{N_g} (i - \mu_x)^2 m_{\theta,D}(i, j),$$

$$\sigma_y^2 = \sum_{i,j}^{N_g} (j - \mu_y)^2 m_{\theta,D}(i, j).$$

---

### Cristais líquidos

---

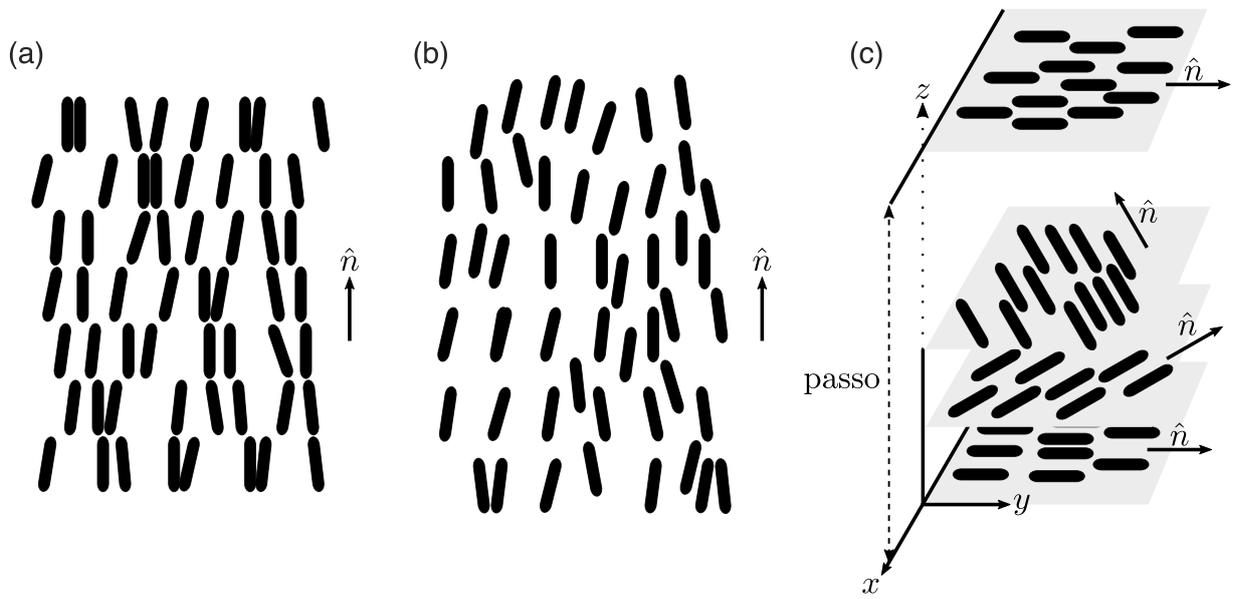
É comumente considerado que a matéria apresenta três fases ou estados: sólido, líquido e gasoso. Esses estados diferem-se pelo ordenamento dos átomos (sólidos) e propriedades físicas como a fluidez (líquidos e gases) [148]. Verifica-se, entretanto, que alguns compostos orgânicos apresentam fases intermediárias da matéria cujas propriedades mecânicas e de simetria (arranjo molecular) estão entre aquelas de sólidos cristalinos e líquidos isotrópicos [148, 215]. Essas fases recebem coletivamente o nome de cristais líquidos<sup>1</sup> ou fases mesomórficas ou, ainda, mesofases [148].

Os cristais líquidos dividem-se em dois grandes grupos quanto ao mecanismo responsável pelas transições para fases líquido cristalinas (ou entre essas fases): liotrópicos e termotrópicos [148, 215]. Transições de fase envolvendo cristais líquidos liotrópicos podem ser induzidas por meio da variação de concentração de determinados tipos de moléculas químicas caracteristicamente alongadas e rígidas (como polímeros) em um solvente apropriado [148, 215]. Transições de fase envolvendo cristais líquidos termotrópicos são obtidas por meio da alteração da temperatura do material [148, 215]. As características físicas que marcam os cristais líquidos tem sua origem, em grande parte, na forma de suas moléculas constituintes. No caso dos cristais líquidos termotrópicos, moléculas em forma de disco (discóticas) ou moléculas alongadas, em forma de bastão (calamíticas), são os tipos mais comuns [148, 215].

Entre as muitas mesofases líquido cristalinas, restringimos nossa atenção a apenas três, relacionadas a cristais líquidos termotrópicos formados por moléculas alongadas: esmética, nemática e colestérica [148, 215].

---

<sup>1</sup>Compostos químicos encontrados nesses estados também são denominados cristais líquidos. Isso acontece da mesma forma que sólidos, líquidos e gases recebem seus nomes em clara alusão aos seus estados físicos.



**Figura E.1:** Representação esquemática de mesofases líquido cristalinas. (a) Fase esmética A. (b) Fase nemática. (c) Fase colestérica.

Dentre a classe de fases esméticas temos, por exemplo, as fases esméticas A, B e C [148]. Para este trabalho, em especial, são importantes a fase esmética A e a fase esmética A quiral [148]. A fase esmética A [Figura E.1(a)] tem como principal característica a disposição de suas moléculas em camadas de tamanho bem definido [148]. Além disso, as moléculas desses planos estão orientadas (em média) na direção de um vetor diretor  $\hat{n}$  que é perpendicular às camadas das moléculas. A fase esmética A de um composto quiral apresenta características de simetria distintas à fase esmética A e é denominada esmética A quiral [148].

A mesofase nemática [Figura E.1(b)] tem como principal característica o alinhamento médio das moléculas na direção de um vetor diretor  $\hat{n}$ , a despeito da falta de ordenamento dos centros de massa das moléculas (inexistência de qualquer simetria translacional) [148, 215]. Comparativamente à fase esmética, a fase nemática é mais desordenada, apresenta menor viscosidade e ocorre a temperaturas mais altas em um mesmo composto [148, 215].

Por fim, cristais líquidos colestéricos [Figura E.1(c)] podem ser sintetizados a partir de moléculas quirais ou obtidos a partir da dopagem de cristais líquidos nemáticos com moléculas quirais [148, 215]. Essa mesofase apresenta características locais de ordenamento similares à mesofase nemática, com a distinção de que a orientação do vetor diretor realiza um movimento de rotação no espaço. A distância associada a um giro completo do vetor diretor é chamada de passo do colestérico [148].

---

## Aprendizagem estatística

---

Algoritmos de aprendizagem estatística têm como principal objetivo automatizar a procura por padrões em conjuntos de dados [216]. Mais comumente, esses conjuntos de dados são do tipo  $\{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ , sendo cada par  $(x_i, y_i)$  uma observação das variáveis  $X$  (variável independente ou preditora) e  $Y$  (variável dependente ou resposta). Em geral, estamos interessados em estabelecer uma relação do tipo

$$\hat{y} = \hat{f}(X = x^*), \quad (\text{F.1})$$

na qual  $\hat{y}$  representa uma estimativa da variável  $Y$  obtida a partir de  $\hat{f}$  quando  $X$  assume o valor  $x^*$  [132]. A estimativa da função  $\hat{f}$  (essência dos métodos de aprendizagem estatística) é realizada a partir da exposição do algoritmo a um subconjunto de observações, o conjunto de treinamento [132, 133]. A qualidade do ajuste de  $\hat{f}$  é medida em outro subconjunto de observações, às quais o algoritmo nunca foi exposto, o conjunto de teste [132, 133]. É comum que durante a etapa de aprendizagem (estimativa de  $\hat{f}$ ) os parâmetros do algoritmo sejam exaustivamente variados na procura pelo melhor modelo em um procedimento denominado *grid-search* [132, 133]. Normalmente, essa procura pelos melhores parâmetros é aliada à realização de uma validação cruzada em  $k$  camadas [132, 133]. Em uma validação cruzada, o conjunto de treinamento é novamente subdividido, de modo que fixados os parâmetros, o algoritmo é repetidamente treinado em  $k - 1$  camadas, enquanto uma delas é usada como conjunto de validação, uma espécie de conjunto de teste dentro do conjunto de treinamento [132, 133]. Estimando o desempenho do algoritmo no conjunto de validação, podemos avaliar a capacidade de generalização do modelo treinado antes de propriamente

testá-lo nas observações do conjunto de teste.

Em casos nos quais os valores da variável dependente  $Y$  são conhecidos, a avaliação da precisão das predições é feita pela comparação direta entre os valores preditos e observados. Nesse caso, o processo de aprendizagem é dito supervisionado. Contrariamente, caso os valores de  $Y$  não sejam conhecidos, o processo de aprendizado é dito não supervisionado. Problemas de agrupamento (*clustering*), estimativa de distribuição de probabilidade ou redução de dimensionalidade (para fins de visualização, por exemplo) constituem problemas não supervisionados [132, 133].

Outra importante distinção entre esses métodos de aprendizagem estatística passa pelo tipo da variável  $Y$ . Caso a variável  $Y$  assuma valores dentro de um conjunto finito de classes ou categorias, dizemos que a tarefa de previsão de  $Y$  é uma tarefa de classificação [132, 133]. Caso  $Y$  assuma valores numéricos em um intervalo contínuo, dizemos que a tarefa é uma regressão [132, 133].

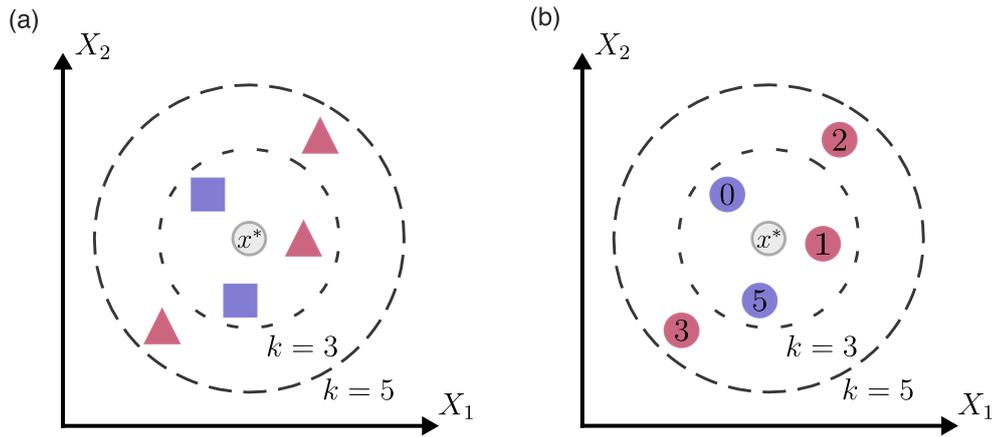
Ao longo deste trabalho, utilizamos vários algoritmos de aprendizagem de máquina e em contextos diversos. Em tarefas supervisionadas de regressão e classificação, utilizamos os algoritmos  $k$ -primeiros vizinhos, logístico e floresta aleatória. Dos métodos não supervisionados, a análise de componentes principais (PCA) foi utilizada como ferramenta de redução de dimensionalidade (e pré-processamento de variáveis preditivas) e o algoritmo  $K$ -means nos ajudou a explorar um conjunto de dados acerca dos quais não existia consenso sobre as verdadeiras classes.

O algoritmo classificador de  $k$ -primeiros vizinhos prediz uma classe ou categoria  $\hat{y}$  para a resposta  $Y$  de uma observação  $X = x^*$  usando um número  $k$  de observações vizinhas [132, 133]. Esses vizinhos são os pontos mais próximos (segundo alguma medida de distância, como a euclidiana) em um espaço abstrato de mesma dimensão que a variável preditora<sup>1</sup>  $X$  [132, 133]. A partir da distribuição de probabilidade das classes na vizinhança de  $x^*$ , o algoritmo prediz como resposta  $\hat{y}$  a classe mais provável dessa distribuição [132, 133]. Posto de outra forma, esse algoritmo funciona segundo uma espécie de “voto da maioria”, em que a classe mais comum entre os vizinhos de  $x^*$  é atribuída a  $\hat{y}$  [Figura F.1(a)]. Assim como acontece em tarefas de classificação em geral, o desempenho do classificador de  $k$ -primeiros vizinhos pode ser medido pela fração de classificações corretas [132].

Quando usado em tarefas de regressão, o método de  $k$ -primeiros vizinhos atribui um valor numérico  $\hat{y}$  a uma observação  $x^*$  seguindo a mesma estratégia de avaliar os valores de sua vizinhança. Denotando o conjunto dos  $k$  primeiros vizinhos de  $x^*$  por  $\mathcal{N}^*$ , o valor estimado

---

<sup>1</sup>A variável preditora não é necessariamente restringida a um escalar simples. É comum que tenhamos um vetor  $X = (X_1, X_2, \dots, X_n)$  com  $n$  características preditivas da variável dependente  $Y$ . Essas características formam um espaço  $n$  dimensional no qual a localização de uma observação é determinada pelas componentes de  $x^* = (x_1^*, x_2^*, \dots, x_n^*)$ .



**Figura F.1: Classificação e regressão de  $k$ -primeiros vizinhos.** Uma observação  $x^* = (x_1^*, x_2^*)$  e seus primeiros vizinhos são localizados em um espaço bidimensional associado às variáveis independentes  $X_1$  e  $X_2$ . (a) Em uma tarefa de classificação, quando a resposta de  $x^*$  pertence a uma das classes do conjunto {"quadrado", "triângulo"}, a predição de primeiros vizinhos resulta em  $\hat{y}$  igual a "quadrado" quando  $k = 3$  e igual a "triângulo" para  $k = 5$ . (b) Em uma tarefa de regressão, quando os vizinhos de  $x^*$  assumem valores numéricos (mostrados na figura), obtemos  $\hat{y} = 2$  quando  $k = 3$  e  $\hat{y} = 2,2$  para  $k = 5$ .

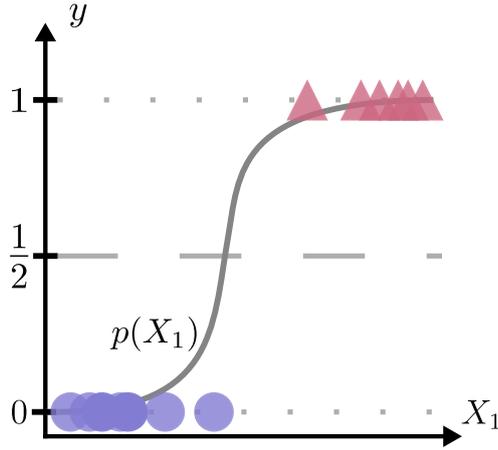
para a resposta  $\hat{y}$  dessa observação é

$$\hat{y} = \hat{f}(x^*) = \frac{1}{k} \sum_{x_i \in \mathcal{N}^*} y_i, \quad (\text{F.2})$$

na qual  $x_i$  e  $y_i$  representam os valores observados para os preditores e resposta do  $i$ -ésimo vizinho de  $x^*$ . Sendo assim, é atribuído a  $\hat{y}$  o valor médio das respostas dos vizinhos de  $x^*$  [Figura F.1(b)]. A avaliação das predições e, conseqüentemente, da qualidade do ajuste do algoritmo aos dados, pode ser feita utilizando o coeficiente de determinação  $R^2$  definido pela expressão [132]

$$R^2 = 1 - \frac{SQ_{\text{res}}}{SQ_{\text{tot}}}, \quad (\text{F.3})$$

sendo  $SQ_{\text{res}} = \sum_{i=1}^n (y_i - \hat{y}_i)^2$  a chamada soma residual de quadrados e  $SQ_{\text{tot}} = \sum_{i=1}^n (y_i - \bar{y})^2$  a soma total de quadrados. A soma total de quadrados quantifica a variabilidade intrínseca ao conjunto de dados, enquanto a soma residual de quadrados representa uma medida da variabilidade que permanece não explicada mesmo após a regressão. Desse modo, o coeficiente de determinação mede a quantidade de variação da resposta que é explicada pelos preditores [132]. Esse coeficiente tem seu máximo igual a 1 quando a regressão explica a dependência entre  $Y$  e  $X$  perfeitamente e é igual a 0 caso a predição da variável independente seja sempre igual ao valor médio do conjunto de dados ( $\hat{y}_i = \bar{y}$ ). O coeficiente de determinação também pode assumir valores negativos, indicando uma regressão arbitrariamente ruim.



**Figura F.2: Regressão logística.** Dependência de um conjunto de observações binárias, representadas por círculos (classe negativa,  $y = 0$ ) e triângulos (classe positiva,  $y = 1$ ), em relação à variável independente  $X_1$ . A curva logística [ $p(X_1)$ , curva contínua em cinza escuro] dá a probabilidade de pertencimento à classe positiva em função de  $X_1$ . O ponto de interseção entre essa curva e a reta  $y = 1/2$  marca a fronteira de decisão entre as classes.

O modelo de regressão logística é um modelo de aprendizagem supervisionada frequentemente aplicado a situações de classificação em que temos apenas duas classes (classificação binária), como mostrado na Figura F.2 [132, 133]. Nesse modelo, as classes negativa e positiva são usualmente representadas pelos números 0 e 1 e as classificações são feitas a partir da probabilidade  $p$  de pertencimento à classe positiva [132, 133]. Concretamente, realizada uma observação  $x^*$ , a classe  $\hat{y}$  atribuída a essa observação é tal que [132, 133]

$$\hat{y} = \begin{cases} 0, & \text{se } p(x^*) < 0,5 \\ 1, & \text{se } p(x^*) \geq 0,5 \end{cases}. \quad (\text{F.4})$$

Esse modelo recebe seu nome em razão da função  $p(x)$  que é utilizada para estimar a probabilidade da classe positiva ser chamada de função logística, a qual é dada por

$$p(x) = \frac{e^{ax+b}}{1 + e^{ax+b}}. \quad (\text{F.5})$$

Os parâmetros  $a$  e  $b$  na Equação F.5 são obtidos do ajuste do modelo a um conjunto de observações e passa pela minimização da função custo  $J$  definida por

$$J(a, b) = \frac{1}{N} \sum_{i=1}^N \{y_i \log [p_{a,b}(x_i)] + (1 - y_i) \log [1 - p_{a,b}(x_i)]\}. \quad (\text{F.6})$$

Um procedimento numérico relativamente simples chamado de método do gradiente<sup>2</sup> altera

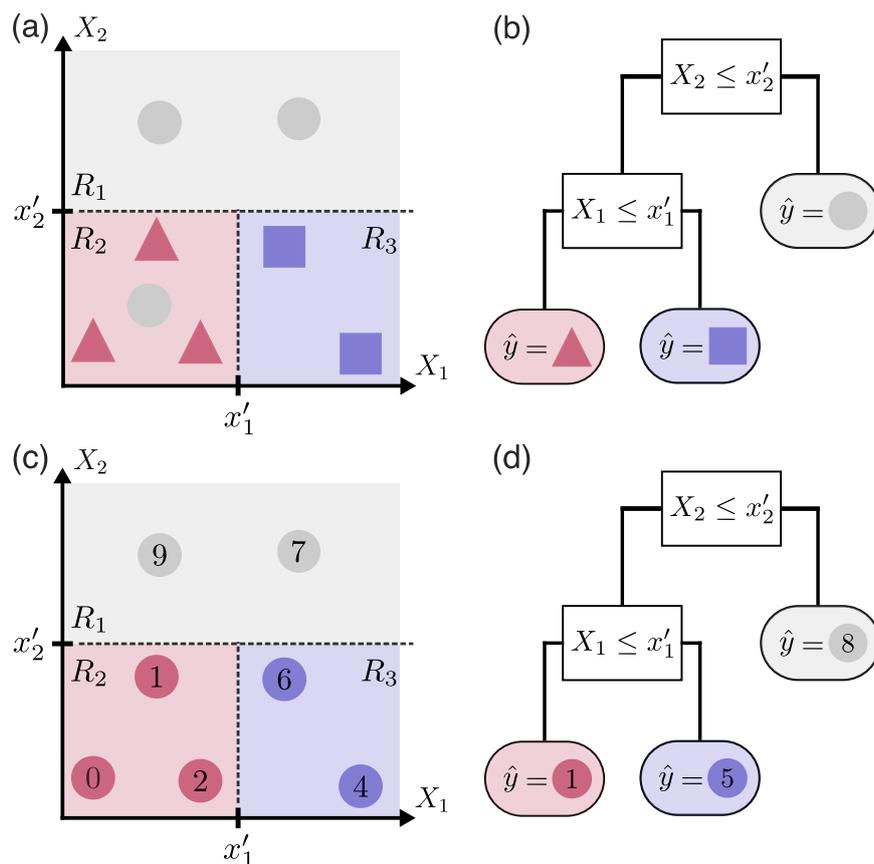
<sup>2</sup>*Gradient descent*, em inglês.

recursivamente os valores de  $a$  e  $b$  de modo a minimizar a soma na Equação F.6 [131–133].

Além dos modelos de primeiros vizinhos e logístico, também usamos o algoritmo de floresta aleatória, que recebe esse nome por ser um modelo que faz previsões a partir de um *ensemble* de árvores de decisão [132, 133]. Árvores de decisão podem ser utilizadas tanto em tarefas de regressão quanto de classificação e têm como principal característica a segmentação (ou estratificação) do espaço de variáveis preditivas. Uma árvore de decisão é constituída de nós internos, nos quais condições binárias implementam a segmentação do espaço de variáveis independentes em diversas regiões  $R_1, R_2, \dots, R_N$  [132, 133]. Esses nós são ligados pelos chamados ramos da árvore (possivelmente passando por outros nós) e terminam em folhas, nós terminais da árvore que atribuem um valor à resposta de uma observação (Figura F.3). Em tarefas de regressão ou classificação, a segmentação do espaço de variáveis (que conseqüentemente determina a estrutura da árvore) é realizada de modo a minimizar uma função de perda como o índice de Gini (em classificações) ou o erro médio quadrático (em regressões) [131, 132]. A previsão da resposta  $\hat{y}$  de uma observação  $x^*$ , encontrada na região  $R^*$ , dá-se da seguinte maneira. Em tarefas de regressão, o valor médio das respostas dentro dessa região é atribuído à resposta  $\hat{y}$ . Em situações de classificação, a classe mais frequente é atribuída à resposta (Figura F.3).

Árvores de decisão são bastante versáteis e de interpretabilidade bastante simples. Entretanto, é uma característica do método a propensão a sobreajustar-se ao conjunto de treinamento (*overfitting*), afetando a generalização do modelo treinado [132]. Sendo assim, a ideia fundamental do algoritmo de floresta aleatória é explorar essa característica das árvores de decisão para construir modelos melhores. Para tanto, em uma floresta aleatória, várias árvores de decisão (um *ensemble*) são treinadas a partir de uma amostragem com reposição do conjunto de treinamento e impondo uma limitação ao número de variáveis preditivas que são consideradas ao longo da estruturação dos nós dessas árvores [132]. Essas duas mudanças geralmente garantem uma melhora substancial no desempenho do método ao custo da redução de interpretabilidade e aumento da complexidade do modelo [132]. Finalmente, o *ensemble* de modelos de árvores aleatórias realiza previsões por meio de um voto da maioria ou soma de probabilidades em classificações, ou por meio da média das previsões em regressões [131, 132].

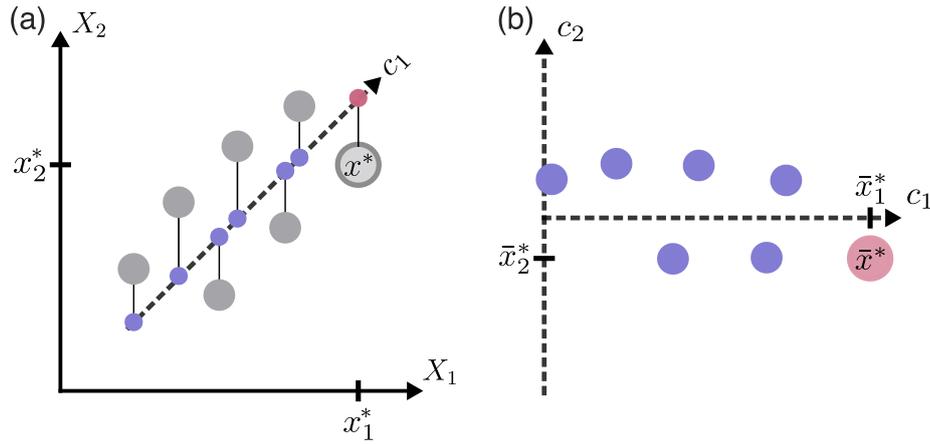
Diferentemente dos algoritmos  $k$ -primeiros vizinhos e floresta aleatória, o método de análise de componentes principais (PCA) é um método não supervisionado [132, 133]. Essencialmente, dado um conjunto de observações, o método PCA é um método de redução de dimensionalidade que lineariza o espaço de variáveis preditivas projetando essas observações em um hiperplano de menor dimensão [Figura F.4(a)] [132, 133]. O hiperplano no qual esses dados são projetados é aquele mais próximo às observações e que, ao mesmo tempo, maximiza a variância remanescente no dado projetado [132, 133]. Os vetores diretores desse hiperplano são os chamados componentes principais, sendo elencados em ordem decrescente da variân-



**Figura F.3: Classificação e regressão com árvores de decisão** (a) As variáveis independentes  $X_1$  e  $X_2$  determinam um espaço bidimensional que é segmentado nas regiões  $R_1$ ,  $R_2$  e  $R_3$ , delimitadas por  $X_1 = x'_1$  e  $X_2 = x'_2$ . (b) A árvore de decisão correspondente à segmentação do espaço nas três regiões mostradas em (a) é constituída de dois nós internos e três folhas. Os ramos da árvore são tais que à esquerda seguem as observações que satisfazem a condição imposta por um nó interno e vice-versa. Em uma tarefa de classificação, na qual a variável dependente pertence a uma das classes em {"círculo", "triângulo", "quadrado"}, a árvore de decisão prediz a classe mais frequente em cada uma das regiões como resposta  $\hat{y}$  de uma observação  $x^* = (x_1^*, x_2^*)$ . (c) Mesmo espaço de variáveis preditivas apresentado em (a), exceto pelo caráter contínuo da variável dependente. (d) Árvore de decisão correspondente à segmentação do espaço das variáveis preditivas em (c). Em uma tarefa de regressão, a árvore de decisão prediz o valor médio em cada uma das regiões como a resposta  $\hat{y}$  de uma observação  $x^* = (x_1^*, x_2^*)$ .

cia concentrada em cada direção [132, 133]. Quando o número de componentes principais se iguala à dimensionalidade do espaço original, obtemos uma mudança de base na representação das observações [como mostrado na Figura F.4(b)]. A projeção das observações sobre um hiperplano gera um novo conjunto de variáveis preditivas, diferentes das originais e de menor dimensão, que pode ser utilizado para fins de visualização ou pré-processamento de variáveis independentes antes da etapa de aprendizagem [132, 133].

O algoritmo de agrupamento  $K$ -means é um algoritmo de aprendizagem não supervisionada utilizado para encontrar grupos em conjuntos de observações [132, 133]. Denotando



**Figura F.4: Análise de componentes principais.** (a) Um conjunto de observações (em cinza) é localizado em um espaço bidimensional associado às variáveis independentes  $X_1$  e  $X_2$ . Uma reta (hiperplano unidimensional) tracejada indica a direção do primeiro componente principal  $c_1$ . As projeções das observações sobre a reta (espaço unidimensional) são ressaltadas em cor. (b) Representação das observações em (a) em um espaço bidimensional cuja base são os dois componentes principais  $c_1$  e  $c_2$ .

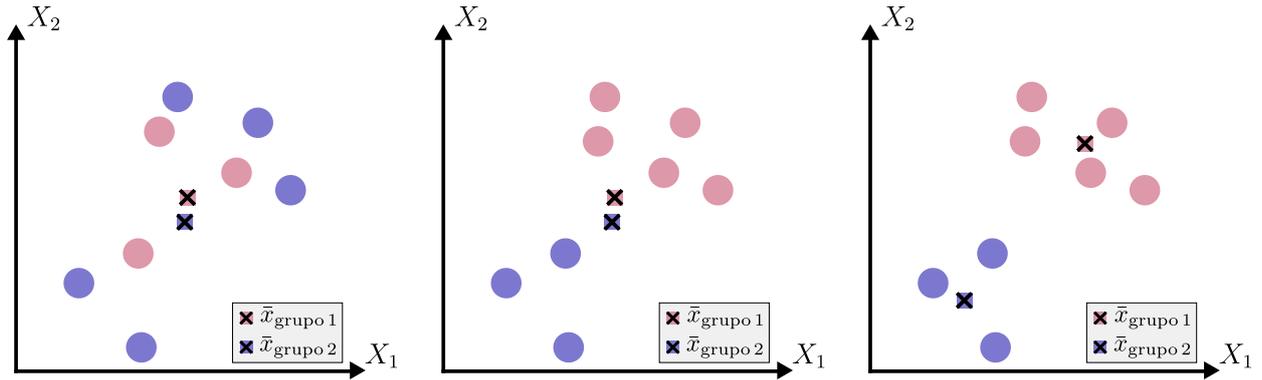
por  $G_k$  o  $k$ -ésimo desses grupos, o algoritmo determina o melhor agrupamento por meio da minimização da inércia  $I$  definida como [132]

$$I = \sum_{k=1}^K \frac{2}{|G_k|} \sum_{i \in G_k} (x_i - \bar{x}_k)^2, \quad (\text{F.7})$$

com  $x_i$  indicando a  $i$ -ésima observação contida no grupo  $G_k$  e o termo  $\bar{x}_k$  representando o valor médio das variáveis independentes desse mesmo grupo. Para minimizar a Equação F.7, o algoritmo  $K$ -means implementa uma rotina simples [132,133]: em um primeiro passo, cada observação é aleatoriamente atribuída a um entre os  $K$  grupos; em seguida, o algoritmo calcula iterativamente o centroide (centro geométrico) de cada grupo e reatribui os grupos às observações com base em suas proximidades aos centroides até que os centros dos grupos tornem-se estáveis (Figura F.5).

O algoritmo  $K$ -means, entretanto, não avalia a pertinência do número de grupos nos quais as observações são segregadas. Usualmente, algumas estratégias de avaliação dos agrupamentos são empregadas conjuntamente. Por exemplo, o chamado “método do cotovelo”<sup>3</sup> analisa a alteração da inércia  $I$  com o aumento do número de grupos e considera o ponto a partir do qual não ocorrem grandes alterações (o cotovelo) como o número ótimo de grupos que devem separar as observações [133]. Outra maneira de avaliar os agrupamentos consiste em comparar seus índices de silhueta [133,171]. Essa estatística representa o valor médio do

<sup>3</sup>Do inglês, *elbow method*.



**Figura F.5: Agrupamento de observações com o algoritmo  $K$ -means.** Um conjunto de observações, localizado em um espaço bidimensional (variáveis independentes  $X_1$  e  $X_2$ ), é apresentado em todos os três painéis. O primeiro painel à esquerda ilustra as posições dos centros dos grupos vermelho (grupo 1) e roxo (grupo 2) após uma atribuição aleatória das observações aos dois grupos ( $K = 2$ ). No painel central, as observações são realocadas aos dois grupos com base em suas proximidades aos centroides. O processo de realocação das observações aos grupos é repetido até que a posição dos centroides permaneça estável e a configuração final dos grupos (mostrada no último painel) seja atingida.

coeficiente de silhueta de cada uma das observações, dado pela expressão [133, 171]

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \quad (\text{F.8})$$

na qual  $a(i)$  é a distância média entre a  $i$ -ésima observação e outras observações de seu grupo e  $b(i)$  a distância média entre a  $i$ -ésima observação e as observações do grupo mais próximo a ela (excluindo o grupo ao qual a observação  $i$  de fato pertence). O coeficiente de silhueta (bem como o índice) pode variar entre 1 e -1, com valores próximos a 1 indicando grupos mais coesos [133, 171].

---

## Referências Bibliográficas

---

- [1] Bridgman, P. W. *The Logic of Modern Physics* (The Macmillan Company, 1927).
- [2] Mitchell, M. *Complexity: A Guided Tour* (Oxford University Press, New York, 2011).
- [3] Crutchfield, J. P. & Young, K. Inferring statistical complexity. *Physical Review Letters* **63**, 105–108 (1989).
- [4] Gell-Mann, M. & Lloyd, S. Information measures, effective complexity, and total information. *Complexity* **2**, 44–52 (1996).
- [5] Lloyd, S. Measures of complexity: A nonexhaustive list. *IEEE Control Systems Magazine* **21**, 7–8 (2001).
- [6] Bandt, C. & Pompe, B. Permutation entropy: A natural complexity measure for time series. *Physical Review Letters* **88**, 174102 (2002).
- [7] Kantz, H. & Schreiber, T. *Nonlinear Time Series Analysis* (Cambridge University Press, New York, 2004).
- [8] Bradley, E. & Kantz, H. Nonlinear time-series analysis revisited. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **25**, 097610 (2015).
- [9] Shannon, C. E. A mathematical theory of communication. *The Bell System Technical Journal* **27**, 379–423 (1948).
- [10] Stone, J. *Information Theory: A Tutorial Introduction* (Sebtel Press, 2016).
- [11] Nicolaou, N. & Georgiou, J. Detection of epileptic electroencephalogram based on permutation entropy and support vector machines. *Expert Systems with Applications* **39**, 202–209 (2012).

- [12] Zunino, L., Zanin, M., Tabak, B. M., Pérez, D. G. & Rosso, O. A. Forbidden patterns, permutation entropy and stock market inefficiency. *Physica A* **388**, 2854–2864 (2009).
- [13] Garland, J., Jones, T., Neuder, M., Morris, V., White, J. & Bradley, E. Anomaly detection in paleoclimate records using permutation entropy. *Entropy* **20**, 931 (2018).
- [14] Yan, R., Liu, Y. & Gao, R. X. Permutation entropy: A nonlinear statistical measure for status characterization of rotary machines. *Mechanical Systems and Signal Processing* **29**, 474–484 (2012).
- [15] Matilla-García, M. & Ruiz Marín, M. A non-parametric independence test using permutation entropy. *Journal of Econometrics* **144**, 139–155 (2008).
- [16] Zunino, L., Pérez, D., Kowalski, A., Martín, M., Garavaglia, M., Plastino, A. & Rosso, O. Fractional Brownian motion, fractional Gaussian noise, and Tsallis permutation entropy. *Physica A* **387**, 6057–6068 (2008).
- [17] Liang, Z., Wang, Y., Sun, X., Li, D., Voss, L. J., Sleight, J. W., Hagihira, S. & Li, X. EEG entropy measures in anesthesia. *Frontiers in Computational Neuroscience* **9**, 16 (2015).
- [18] Aziz, W. & Arif, M. Multiscale permutation entropy of physiological time series. In *2005 Pakistan Section Multitopic Conference*, 1–6 (IEEE, 2005).
- [19] Zunino, L., Soriano, M. C., Fischer, I., Rosso, O. A. & Mirasso, C. R. Permutation-information-theory approach to unveil delay dynamics from time-series analysis. *Physical Review E* **82**, 046212 (2010).
- [20] Zunino, L., Soriano, M. C. & Rosso, O. A. Distinguishing chaotic and stochastic dynamics from time series by using a multiscale symbolic approach. *Physical Review E* **86**, 046210 (2012).
- [21] Fadlallah, B., Chen, B., Keil, A. & Príncipe, J. Weighted-permutation entropy: A complexity measure for time series incorporating amplitude information. *Physical Review E* **87**, 022911 (2013).
- [22] Xia, J., Shang, P., Wang, J. & Shi, W. Permutation and weighted-permutation entropy analysis for the complexity of nonlinear time series. *Communications in Nonlinear Science and Numerical Simulation* **31**, 60–68 (2016).
- [23] Azami, H. & Escudero, J. Amplitude-aware permutation entropy: Illustration in spike detection and signal segmentation. *Computer Methods and Programs in Biomedicine* **128**, 40–51 (2016).

- [24] Chen, S., Shang, P. & Wu, Y. Weighted multiscale Rényi permutation entropy of nonlinear time series. *Physica A* **496**, 548–570 (2018).
- [25] Bian, C., Qin, C., Ma, Q. D. Y. & Shen, Q. Modified permutation-entropy analysis of heartbeat dynamics. *Physical Review E* **85**, 021906 (2012).
- [26] Cuesta-Frau, D., Varela-Entrecanales, M., Molina-Picó, A. & Vargas, B. Patterns with equal values in permutation entropy: Do they really matter for biosignal classification? *Complexity* **2018**, 1324696 (2018).
- [27] Ribeiro, H. V., Zunino, L., Lenzi, E. K., Santoro, P. A. & Mendes, R. S. Complexity-entropy causality plane as a complexity measure for two-dimensional patterns. *PLOS ONE* **7**, 1–9 (2012).
- [28] Zunino, L. & Ribeiro, H. V. Discriminating image textures with the multiscale two-dimensional complexity-entropy causality plane. *Chaos, Solitons & Fractals* **91**, 679–688 (2016).
- [29] Sigaki, H. Y. D., Perc, M. & Ribeiro, H. V. History of art paintings through the lens of entropy and complexity. *Proceedings of the National Academy of Sciences* **115**, E8585–E8594 (2018).
- [30] Sigaki, H. Y. D., de Souza, R. F., de Souza, R. T., Zola, R. S. & Ribeiro, H. V. Estimating physical properties from liquid crystal textures via machine learning and complexity-entropy methods. *Physical Review E* **99**, 013311 (2019).
- [31] Small, M. Complex networks from time series: Capturing dynamics. In *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013)*, 2509–2512 (2013).
- [32] Pessa, A. A. B. & Ribeiro, H. V. Characterizing stochastic time series with ordinal networks. *Physical Review E* **100**, 042304 (2019).
- [33] Pessa, A. A. B. & Ribeiro, H. V. Mapping images into ordinal networks. *Physical Review E* **102**, 052312 (2020).
- [34] Zou, Y., Donner, R. V., Marwan, N., Donges, J. F. & Kurths, J. Complex network approaches to nonlinear time series analysis. *Physics Reports* **787**, 1–97 (2019).
- [35] The data deluge. Disponível: <https://www.economist.com/leaders/2010/02/25/the-data-deluge> (2010). Acessado: 25 Out 2021.
- [36] Bell, G., Hey, T. & Szalay, A. Beyond the data deluge. *Science* **323**, 1297–1298 (2009).
- [37] Mattmann, C. A. A vision for data science. *Nature* **493**, 473–475 (2013).

- [38] Blei, D. M. & Smyth, P. Science and data science. *Proceedings of the National Academy of Sciences* **114**, 8689–8692 (2017).
- [39] Pessa, A. A. B. & Ribeiro, H. V. ordpy: A Python package for data analysis with permutation entropy and ordinal network methods. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **31**, 063110 (2021).
- [40] Pessa, A. A., Zola, R. S., Perc, M. & Ribeiro, H. V. Determining liquid crystal properties with ordinal networks and machine learning. *Chaos, Solitons & Fractals* **154**, 111607 (2022).
- [41] Harris, C. R. *et al.* Array programming with NumPy. *Nature* **585**, 357–362 (2020).
- [42] Perkel, J. M. Pick up Python. *Nature* **518**, 125–126 (2015).
- [43] Shen, H. Interactive notebooks: Sharing the code. *Nature* **515**, 151–152 (2014).
- [44] Kluyver, T. *et al.* Jupyter notebooks – A publishing format for reproducible computational workflows. In Loizides, F. & Schmidt, B. (eds.) *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, 87–90 (IOS Press, 2016).
- [45] Daw, C. S., Finney, C. E. A. & Tracy, E. R. A review of symbolic analysis of experimental data. *Review of Scientific Instruments* **74**, 915–930 (2003).
- [46] Chatfield, C. *The Analysis of Time Series: An Introduction* (CRC Press, London, 2016).
- [47] Cao, Y., wen Tung, W., Gao, J. B., Protopopescu, V. A. & Hively, L. M. Detecting dynamical changes in time series using the permutation entropy. *Physical Review E* **70**, 046217 (2004).
- [48] Zunino, L., Olivares, F., Scholkmann, F. & Rosso, O. A. Permutation entropy based time series analysis: Equalities in the input signal can lead to false conclusions. *Physics Letters A* **381**, 1883–1892 (2017).
- [49] Amigó, J., Zambrano, S. & Sanjuán, M. A. Combinatorial detection of determinism in noisy time series. *EPL (Europhysics Letters)* **83**, 60005 (2008).
- [50] Cuesta-Frau, D., Murillo-Escobar, J. P., Orrego, D. A. & Delgado-Trejos, E. Embedded dimension and time series length. Practical influence on permutation entropy and its applications. *Entropy* **21**, 385 (2019).
- [51] Rosso, O. A., Larrondo, H. A., Martin, M. T., Plastino, A. & Fuentes, M. A. Distinguishing noise from chaos. *Physical Review Letters* **99**, 154102 (2007).

- [52] Rosso, O. A. & Masoller, C. Detecting and quantifying temporal correlations in stochastic resonance via information theory measures. *The European Physical Journal B* **69**, 37–43 (2009).
- [53] Zunino, L., Zanin, M., Tabak, B. M., Pérez, D. G. & Rosso, O. A. Complexity-entropy causality plane: A useful approach to quantify the stock market inefficiency. *Physica A* **389**, 1891–1901 (2010).
- [54] Zunino, L., Fernández Bariviera, A., Guercio, M. B., Martinez, L. B. & Rosso, O. A. On the efficiency of sovereign bond markets. *Physica A* **391**, 4342–4349 (2012).
- [55] Ribeiro, H. V., Zunino, L., Mendes, R. S. & Lenzi, E. K. Complexity–entropy causality plane: A useful approach for distinguishing songs. *Physica A* **391**, 2421–2428 (2012).
- [56] Alves, L. G., Sigaki, H. Y., Perc, M. & Ribeiro, H. V. Collective dynamics of stock market efficiency. *Scientific Reports* **10**, 21992 (2020).
- [57] Ayyad, M. & Coulibaly, S. The cellular automata inside optical chimera states. *Chaos, Solitons & Fractals* **153**, 111524 (2021).
- [58] Zanin, M. & Olivares, F. Ordinal patterns-based methodologies for distinguishing chaos from noise in discrete time series. *Communications Physics* **4**, 1–14 (2021).
- [59] Jara, J., Morales-Rojas, C., Fernández-Muñoz, J., Haunton, V. J. & Chacón, M. Using complexity–entropy planes to detect Parkinson’s disease from short segments of haemodynamic signals. *Physiological Measurement* **42**, 084002 (2021).
- [60] Valensise, C. M., Serra, A., Galeazzi, A., Etta, G., Cinelli, M. & Quattrociochi, W. Entropy and complexity unveil the landscape of memes evolution. *Scientific Reports* **11**, 20022 (2021).
- [61] Ma, F., Fan, Q. & Ling, G. Complexity-entropy causality plane analysis of air pollution series. *Fluctuation and Noise Letters* **21**, 2250011 (2022).
- [62] López-Ruiz, R., Mancini, H. L. & Calbet, X. A statistical measure of complexity. *Physics Letters A* **209**, 321–326 (1995).
- [63] Lin, J. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory* **37**, 145–151 (1991).
- [64] Lamberti, P. W., Martin, M. T., Plastino, A. & Rosso, O. A. Intensive entropic non-triviality measure. *Physica A* **334**, 119–131 (2004).
- [65] Martin, M. T., Plastino, A. & Rosso, O. A. Generalized statistical complexity measures: Geometrical and analytical properties. *Physica A* **369**, 439–462 (2006).

- [66] Feldman, D. P. & Crutchfield, J. P. Measures of statistical complexity: Why? *Physics Letters A* **238**, 244–252 (1998).
- [67] Bandt, C. & Shiha, F. Order patterns in time series. *Journal of Time Series Analysis* **28**, 646–665 (2007).
- [68] Amigó, J. M., Kocarev, L. & Szczepanski, J. Order patterns and chaos. *Physics Letters A* **355**, 27–31 (2006).
- [69] Amigó, J. M., Zambrano, S. & Sanjuán, M. A. True and false forbidden patterns in deterministic and random dynamics. *EPL (Europhysics Letters)* **79**, 50001 (2007).
- [70] Landau, D. P. & Binder, K. *A Guide to Monte Carlo Simulations in Statistical Physics* (Cambridge University Press, New York, 2009).
- [71] Abstract Painting: Blue. Disponível: <https://www.wikiart.org/en/ad-reinhardt/abstract-painting-blue-1953>. Acessado: 24 Nov 2020.
- [72] Abaporu. Disponível: <https://www.wikiart.org/en/tarsila-do-amaral/abaporu-1928>. Acessado: 24 Nov 2020.
- [73] Number 1 (Lavender Mist). Disponível: <https://www.wikiart.org/en/jackson-pollock/number-1-lavender-mist-1950-1>. Acessado: 24 Nov 2020.
- [74] Zanin, M. Forbidden patterns in financial time series. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **18**, 013119 (2008).
- [75] Sakellariou, K., McCullough, M., Stemler, T. & Small, M. Counting forbidden patterns in irregularly sampled time series. II. Reliability in the presence of highly irregular sampling. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **26**, 123104 (2016).
- [76] McCullough, M., Sakellariou, K., Stemler, T. & Small, M. Counting forbidden patterns in irregularly sampled time series. I. The effects of under-sampling, random depletion, and timing jitter. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **26**, 123103 (2016).
- [77] Kulp, C. W., Chobot, J. M., Niskala, B. J. & Needhammer, C. J. Using forbidden ordinal patterns to detect determinism in irregularly sampled time series. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **26**, 023107 (2016).
- [78] Bandt, C. A new kind of permutation entropy used to classify sleep stages from invisible EEG microstructure. *Entropy* **19**, 197 (2017).

- [79] Small, M., McCullough, M. & Sakellariou, K. Ordinal network measures – Quantifying determinism in data. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, 1–5 (2018).
- [80] Tsallis, C. Possible generalization of Boltzmann-Gibbs statistics. *Journal of Statistical Physics* **52**, 479–487 (1988).
- [81] Rényi, A. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, 547–561 (University of California Press, Berkeley, 1961).
- [82] Ribeiro, H. V., Jauregui, M., Zunino, L. & Lenzi, E. K. Characterizing time series via complexity-entropy curves. *Physical Review E* **95**, 062106 (2017).
- [83] Jauregui, M., Zunino, L., Lenzi, E. K., Mendes, R. S. & Ribeiro, H. V. Characterization of time series via Rényi complexity-entropy curves. *Physica A* **498**, 74–85 (2018).
- [84] Tsallis, C. *Introduction to Nonextensive Statistical Mechanics: Approaching a Complex World* (Springer, New York, 2009).
- [85] van Erven, T. & Harremoës, P. Rényi divergence and Kullback-Leibler divergence. *IEEE Transactions on Information Theory* **60**, 3797–3820 (2014).
- [86] McCullough, M., Small, M., Stemler, T. & Iu, H. H.-C. Time lagged ordinal partition networks for capturing dynamics of continuous dynamical systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **25**, 053101 (2015).
- [87] McCullough, M., Small, M., Iu, H. H. C. & Stemler, T. Multiscale ordinal network analysis of human cardiac dynamics. *Philosophical Transactions of the Royal Society A* **375**, 20160292 (2017).
- [88] Sun, X., Small, M., Zhao, Y. & Xue, X. Characterizing system dynamics with a weighted and directed network constructed from time series data. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **24**, 024402 (2014).
- [89] Sakellariou, K., Stemler, T. & Small, M. Markov modeling via ordinal partitions: An alternative paradigm for network-based time-series analysis. *Physical Review E* **100**, 062307 (2019).
- [90] Newman, M. *Networks: An Introduction* (Oxford University Press, New York, 2010).
- [91] Unakafov, A. M. & Keller, K. Conditional entropy of ordinal patterns. *Physica D* **269**, 94–102 (2014).

- [92] Peixoto, T. P. The graph-tool Python library. *figshare* (2014).
- [93] Hagberg, A. A., Schult, D. A. & Swart, P. J. Exploring network structure, dynamics, and function using networkX. In Varoquaux, G., Vaught, T. & Millman, J. (eds.) *Proceedings of the 7th Python in Science Conference*, 11–15 (Pasadena, 2008).
- [94] Csardi, G. & Nepusz, T. The igraph software package for complex network research. *InterJournal Complex Systems*, 1695 (2006).
- [95] Peng, C.-K., Buldyrev, S. V., Havlin, S., Simons, M., Stanley, H. E. & Goldberger, A. L. Mosaic organization of DNA nucleotides. *Physical Review E* **49**, 1685–1689 (1994).
- [96] Centre for Research and Applications in Remote Sensing (CARTEL, University of Sherbrooke). Multiband texture database. Disponível: [https://multibandtexture.recherche.usherbrooke.ca/normalized\\_brodatz.html](https://multibandtexture.recherche.usherbrooke.ca/normalized_brodatz.html). Acessado: 18 Dez 2020.
- [97] Safia, A. & He, D.-C. New Brodatz-based image databases for grayscale color and multiband texture analysis. *ISRN Machine Vision* **2013**, 14 (2013).
- [98] Vespignani, A. Twenty years of network science. *Nature* **558**, 528–529 (2018).
- [99] Dorogovtsev, S. *Lectures on Complex Networks*. Oxford Master Series in Physics (Oxford University Press, 2010).
- [100] Albert, R. & Barabási, A.-L. Statistical mechanics of complex networks. *Reviews of Modern Physics* **74**, 47–97 (2002).
- [101] Zanin, M., Papo, D., Sousa, P. A., Menasalvas, E., Nicchi, A., Kubik, E. & Boccaletti, S. Combining complex networks and data mining: Why and how. *Physics Reports* **635**, 1–44 (2016).
- [102] Lacasa, L., Luque, B., Ballesteros, F., Luque, J. & Nuño, J. C. From time series to complex networks: The visibility graph. *Proceedings of the National Academy of Sciences* **105**, 4972–4975 (2008).
- [103] Marwan, N., Donges, J. F., Zou, Y., Donner, R. V. & Kurths, J. Complex network approach for recurrence analysis of time series. *Physics Letters A* **373**, 4246–4254 (2009).
- [104] Pessa, A. A. B. *Análise de Séries Temporais via Redes Ordinais*. Dissertação de Mestrado, Universidade Estadual de Maringá (2019).

- [105] Luque, B., Lacasa, L., Ballesteros, F. & Luque, J. Horizontal visibility graphs: Exact results for random time series. *Physical Review E* **80**, 046103 (2009).
- [106] Lacasa, L., Nuñez, A., Roldán, É., Parrondo, J. M. R. & Luque, B. Time series irreversibility: A visibility graph approach. *The European Physical Journal B* **85**, 217 (2012).
- [107] Bezsudnov, I. V. & Snarskii, A. A. From the time series to the complex networks: The parametric natural visibility graph. *Physica A* **414**, 53–60 (2014).
- [108] Bianchi, F. M., Livi, L., Alippi, C. & Jenssen, R. Multiplex visibility graphs to investigate recurrent neural network dynamics. *Scientific Reports* **7**, 44037 (2017).
- [109] Eckmann, J.-P., Kamphorst, S. O. & Ruelle, D. Recurrence plots of dynamical systems. *EPL (Europhysics Letters)* **4**, 973–977 (1987).
- [110] Olivares, F., Zanin, M., Zunino, L. & Pérez, D. Contrasting chaotic with stochastic dynamics via ordinal transition networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **30**, 063101 (2020).
- [111] Wu, Z., Lu, X. & Deng, Y. Image edge detection based on local dimension: A complex networks approach. *Physica A* **440**, 9–18 (2015).
- [112] Xu, D., Chen, X., Xie, Y., Yang, C. & Gui, W. Complex networks-based texture extraction and classification method for mineral flotation froth images. *Minerals Engineering* **83**, 105–116 (2015).
- [113] Cuadros, O., Botelho, G., Rodrigues, F. & Neto, J. B. Segmentation of large images with complex networks. In *2012 25th SIBGRAPI Conference on Graphics, Patterns and Images*, 24–31 (IEEE, 2012).
- [114] Backes, A. R., Casanova, D. & Bruno, O. M. A complex network-based approach for boundary shape analysis. *Pattern Recognition* **42**, 54–67 (2009).
- [115] Backes, A. R., Casanova, D. & Bruno, O. M. Texture analysis and classification: A complex network-based approach. *Information Sciences* **219**, 168–180 (2013).
- [116] Gonçalves, W. N., Machado, B. B. & Bruno, O. M. A complex network approach for dynamic texture recognition. *Neurocomputing* **153**, 211–220 (2015).
- [117] Xiao, Q., Pan, X., Li, X.-L., Stephen, M., Yang, H.-J., Jiang, Y., Wang, J.-Y. & Zhang, Q.-J. Row-column visibility graph approach to two-dimensional landscapes. *Chinese Physics B* **23**, 078904 (2014).

- [118] Lacasa, L. & Iacovacci, J. Visibility graphs of random scalar fields and spatial data. *Physical Review E* **96**, 012318 (2017).
- [119] Iacovacci, J. & Lacasa, L. Visibility graphs for image processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **42**, 974–987 (2019).
- [120] Cowell, F. *Measuring Inequality* (Oxford University Press, New York, 2011).
- [121] Mandelbrot, B. B. *The Fractal Geometry of Nature* (Freeman, New York, 1982).
- [122] Yin, Z.-M. New methods for simulation of fractional brownian motion. *Journal of Computational Physics* **127**, 66–72 (1996).
- [123] Brito, A. F., Redinz, J. A. & Plascak, J. A. Dynamics of rough surfaces generated by two-dimensional lattice spin models. *Physical Review E* **75**, 046106 (2007).
- [124] Brito, A. F., Redinz, J. A. & Plascak, J. A. Two-dimensional *XY* and clock models studied via the dynamics generated by rough surfaces. *Physical Review E* **81**, 031130 (2010).
- [125] Brazhe, A. Shearlet-based measures of entropy and complexity for two-dimensional patterns. *Physical Review E* **97**, 061301 (2018).
- [126] Van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E. & Yu, T. scikit-image: Image processing in Python. *PeerJ* **2**, e453 (2014).
- [127] Haralick, R. M., Shanmugam, K. & Dinstein, I. H. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics* **3**, 610–621 (1973).
- [128] Haralick, R. M. Statistical and structural approaches to texture. *Proceedings of the IEEE* **67**, 786–804 (1979).
- [129] Nanni, L., Brahnam, S., Ghidoni, S., Menegatti, E. & Barrier, T. Different approaches for extracting information from the co-occurrence matrix. *PLOS ONE* **8**, e83554 (2013).
- [130] Cavalin, P. & Oliveira, L. S. A review of texture classification methods and databases. In *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images Tutoriais (SIBGRAPI-T)*, 1–8 (IEEE, 2017).
- [131] Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* **12**, 2825–2830 (2011).

- [132] James, G., Witten, D., Hastie, T. & Tibshirani, R. *An Introduction to Statistical Learning with Applications in R*. Springer Texts in Statistics (Springer, New York, 2014).
- [133] Géron, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* (O’Reilly Media, 2019).
- [134] Baldi, P., Sadowski, P. & Whiteson, D. Searching for exotic particles in high-energy physics with deep learning. *Nature Communications* **5**, 4308 (2014).
- [135] Mukund, N., Abraham, S., Kandhasamy, S., Mitra, S. & Philip, N. S. Transient classification in LIGO data using difference boosting neural network. *Physical Review D* **95**, 104059 (2017).
- [136] Carleo, G., Cirac, I., Cranmer, K., Daudet, L., Schuld, M., Tishby, N., Vogt-Maranto, L. & Zdeborová, L. Machine learning and the physical sciences. *Reviews of Modern Physics* **91**, 045002 (2019).
- [137] Dreissigacker, C., Sharma, R., Messenger, C., Zhao, R. & Prix, R. Deep-learning continuous gravitational waves. *Physical Review D* **100**, 044009 (2019).
- [138] Ma, W., Cheng, F. & Liu, Y. Deep-learning-enabled on-demand design of chiral metamaterials. *ACS Nano* **12**, 6326–6334 (2018).
- [139] Zhang, Z., Schott, J. A., Liu, M., Chen, H., Lu, X., Sumpter, B. G., Fu, J. & Dai, S. Prediction of carbon dioxide adsorption via deep learning. *Angewandte Chemie International Edition* **58**, 259–263 (2019).
- [140] Butler, K. T., Davies, D. W., Cartwright, H., Isayev, O. & Walsh, A. Machine learning for molecular and materials science. *Nature* **559**, 547–555 (2018).
- [141] Jha, D., Ward, L., Paul, A., Liao, W.-k., Choudhary, A., Wolverton, C. & Agrawal, A. Elemnet: Deep learning the chemistry of materials from only elemental composition. *Scientific Reports* **8**, 1–13 (2018).
- [142] Ziletti, A., Kumar, D., Scheffler, M. & Ghiringhelli, L. M. Insightful classification of crystal structures using deep learning. *Nature Communications* **9**, 2775 (2018).
- [143] Wei, J., Chu, X., Sun, X.-Y., Xu, K., Deng, H.-X., Chen, J., Wei, Z. & Lei, M. Machine learning in materials science. *InfoMat* **1**, 338–358 (2019).
- [144] Tshitoyan, V., Dagdelen, J., Weston, L., Dunn, A., Rong, Z., Kononova, O., Persson, K. A., Ceder, G. & Jain, A. Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature* **571**, 95 (2019).

- [145] Agrawal, A. & Choudhary, A. Perspective: Materials informatics and big data: Realization of the “fourth paradigm” of science in materials science. *APL Materials* **4**, 053208 (2016).
- [146] Stein, H. S., Guevarra, D., Newhouse, P. F., Soedarmadji, E. & Gregoire, J. M. Machine learning of optical properties of materials – Predicting spectra from images and images from spectra. *Chemical Science* **10**, 47–55 (2019).
- [147] Sigaki, H. Y. D., Lenzi, E. K., Zola, R. S., Perc, M. & Ribeiro, H. V. Learning physical properties of liquid crystals with deep convolutional neural networks. *Scientific Reports* **10**, 7664 (2020).
- [148] de Gennes, P. & Prost, J. *The Physics of Liquid Crystals*. International Series of Monographs on Physics (Clarendon Press, Oxford, 1993).
- [149] Lee, T., Senyuk, B., Trivedi, R. P. & Smalyukh, I. I. *Optical Microscopy of Soft Matter Systems*, chap. 10, 165–185 (John Wiley & Sons, 2016).
- [150] Ramprasad, R., Batra, R., Pilia, G., Mannodi-Kanakkithodi, A. & Kim, C. Machine learning in materials informatics: recent applications and prospects. *npj Computational Materials* **3**, 54– (2017).
- [151] Murdock, R. J., Kauwe, S. K., Wang, A. Y.-T. & Sparks, T. D. Is domain knowledge necessary for machine learning materials properties? *Integrating Materials and Manufacturing Innovation* **9**, 221–227 (2020).
- [152] Merck. 4'-Octyl-4-biphenylcarbonitrile. Disponível: <https://www.sigmaaldrich.com/BR/pt/product/aldrich/338680>. Acessado: 02 Nov 2021.
- [153] Wu, S.-T. & Yang, D.-K. *Reflective Liquid Crystal Displays* (Wiley, Hoboken, 2001).
- [154] Feltre, R. *Química*, vol. 3 (Editora Moderna, São Paulo, 2004).
- [155] Chemical, D. CAS NO.133676-09-2 R811. Disponível: <https://www.dakenchem.com/product/cas-no-133676-09-2/>. Acessado: 24 Out 2021.
- [156] Daken Chemical. CAS NO.87321-20-8 S811. Disponível: <https://www.dakenchem.com/product/cas-no-87321-20-8/>. Acessado: 24 Out 2021.
- [157] Lin, Y., Douali, R., Dubois, F., Segovia-Mera, A. & Daoudi, A. On the phase transitions of 8cb/sn<sub>2</sub>p<sub>2</sub>s<sub>6</sub> liquid crystal nanocolloids. *The European Physical Journal E* **38**, 1–8 (2015).

- [158] Zola, R. S., Evangelista, L., Yang, Y.-C. & Yang, D.-K. Surface induced phase separation and pattern formation at the isotropic interface in chiral nematic liquid crystals. *Physical Review Letters* **110**, 057801 (2013).
- [159] Pessa, A. A. B., Perc, M. & Ribeiro, H. V. Clustering free-falling paper motion with complexity and entropy. *EPL (Europhysics Letters)* **138**, 30003 (2022).
- [160] Dugas, R. *A History of Mechanics* (Routledge & Kegan Paul, London, 1955).
- [161] Rovelli, C. Aristotle’s physics: A physicist’s look. *Journal of the American Philosophical Association* **1**, 23–40 (2015).
- [162] Maxwell, J. C. *On a particular case of the descent of a heavy body in a resisting medium*, vol. 1 of *Cambridge Library Collection - Physical Sciences*, 115–118 (Cambridge University Press, 2011).
- [163] Tanabe, Y. & Kaneko, K. Behavior of a falling paper. *Physical Review Letters* **73**, 1372–1375 (1994).
- [164] Belmonte, A., Eisenberg, H. & Moses, E. From flutter to tumble: Inertial drag and froude similarity in falling paper. *Physical Review Letters* **81**, 345–348 (1998).
- [165] Mahadevan, L., Ryu, W. S. & Samuel, A. D. T. Tumbling cards. *Physics of Fluids* **11**, 1–3 (1999).
- [166] Andersen, A., Pesavento, U. & Wang, Z. J. Analysis of transitions between fluttering, tumbling and steady descent of falling cards. *Journal of Fluid Mechanics* **541**, 91–104 (2005).
- [167] Howison, T., Hughes, J. & Iida, F. Large-scale automated investigation of free-falling paper shapes via iterative physical experimentation. *Nature Machine Intelligence* **2**, 68–75 (2020).
- [168] Willmarth, W. W., Hawk, N. E. & Harvey, R. L. Steady and unsteady motions and wakes of freely falling disks. *The Physics of Fluids* **7**, 197–208 (1964).
- [169] Field, S. B., Klaus, M., Moore, M. G. & Nori, F. Chaotic dynamics of falling disks. *Nature* **388**, 252–254 (1997).
- [170] Howison, T., Hughes, J., Giardina, F. & Iida, F. Physics driven behavioural clustering of free-falling paper shapes. *PLOS ONE* **14**, e0217997 (2019).
- [171] Rousseeuw, P. J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics* **20**, 53–65 (1987).

- [172] Kitchin, R. Big data, new epistemologies and paradigm shifts. *Big Data & Society* **1**, 1 (2014).
- [173] Anderson, C. The End of Theory: The Data Deluge Makes the Scientific Method Obsolete. Disponível: <https://www.wired.com/2008/06/pb-theory/> (2008). Acessado: 25 Out 2021.
- [174] Hey, T., Tansley, S. & Tolle, K. Jim Gray on eScience: A Transformed Scientific Method. In *The Fourth Paradigm: Data-Intensive Scientific Discovery*, xvii–xxxi (Redmond: Microsoft Research, 2009).
- [175] Thorp, H. H. & Yaffe, M. B. Seeing is great, understanding is better. *Science Signaling* **16**, eadg2868 (2023).
- [176] Hegger, R., Kantz, H. & Schreiber, T. Practical implementation of nonlinear time series methods: The TISEAN package. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **9**, 413–435 (1999).
- [177] Donges, J. F. *et al.* Unified functional network and nonlinear time series analysis for complex systems science: The pyunicorn package. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **25**, 113101 (2015).
- [178] Alstott, J., Bullmore, E. & Plenz, D. powerlaw: A Python package for analysis of heavy-tailed distributions. *PLOS ONE* **9**, 1–11 (2014).
- [179] Olivares, F., Plastino, A. & Rosso, O. A. Contrasting chaos with noise via local versus global information quantifiers. *Physics Letters A* **376**, 1577–1583 (2012).
- [180] Zunino, L., Olivares, F., Ribeiro, H. V. & Rosso, O. A. Permutation Jensen-Shannon distance: A versatile and fast symbolic tool for complex time-series analysis. *Physical Review E* **105** (2022).
- [181] Myers, A. & Khasawneh, F. A. On the automatic parameter selection for permutation entropy. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **30**, 033130 (2020).
- [182] Zanin, M., Zunino, L., Rosso, O. A. & Papo, D. Permutation entropy and its main biomedical and econophysics applications: A review. *Entropy* **14**, 1553–1577 (2012).
- [183] Riedl, M., Müller, A. & Wessel, N. Practical considerations of permutation entropy. *The European Physical Journal Special Topics* **222**, 249–262 (2013).
- [184] Amigó, J. M., Keller, K. & Unakafova, V. A. Ordinal symbolic analysis and its application to biomedical recordings. *Philosophical Transactions of the Royal Society A* **373**, 20140091 (2015).

- [185] Keller, K., Mangold, T., Stolz, I. & Werner, J. Permutation entropy: New ideas and challenges. *Entropy* **19**, 134 (2017).
- [186] Kulp, C. W., Chobot, J. M., Freitas, H. R. & Sprechini, G. D. Using ordinal partition transition networks to analyze ECG data. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **26**, 073114 (2016).
- [187] Small, M. *Applied Nonlinear Time Series Analysis: Applications in Physics, Physiology and Finance*. World Scientific series in nonlinear science, Series A (World Scientific, Singapore, 2005).
- [188] Packard, N. H., Crutchfield, J. P., Farmer, J. D. & Shaw, R. S. Geometry from a time series. *Physical Review Letters* **45**, 712–716 (1980).
- [189] Kennel, M. B., Brown, R. & Abarbanel, H. D. I. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Physical Review A* **45**, 3403–3411 (1992).
- [190] Cao, L. Practical method for determining the minimum embedding dimension of a scalar time series. *Physica D* **110**, 43–50 (1997).
- [191] Fraser, A. M. & Swinney, H. L. Independent coordinates for strange attractors from mutual information. *Physical Review A* **33**, 1134–1140 (1986).
- [192] Groth, A. Visualization of coupling in time series by order recurrence plots. *Physical Review E* **72**, 046220 (2005).
- [193] Wang, X., Si, S., Wei, Y. & Li, Y. The optimized multi-scale permutation entropy and its application in compound fault diagnosis of rotating machinery. *Entropy* **21**, 170 (2019).
- [194] McCullough, M., Sakellariou, K., Stemler, T. & Small, M. Regenerating time series from ordinal networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **27**, 035814 (2017).
- [195] Sakellariou, K., Stemler, T. & Small, M. Estimating topological entropy using ordinal partition networks. *Physical Review E* **103**, 022214 (2021).
- [196] Ott, E. *Chaos in Dynamical Systems* (Cambridge University Press, New York, 2002).
- [197] May, R. M. Simple mathematical models with very complicated dynamics. *Nature* **261**, 459–467 (1976).
- [198] Trulla, L. L., Giuliani, A., Zbilut, J. P. & Webber, C. L. Recurrence quantification analysis of the logistic equation with transients. *Physics Letters A* **223**, 255–260 (1996).

- [199] Sakai, H. & Tokumaru, H. Autocorrelations of a certain chaos. *IEEE Transactions on Acoustics, Speech, and Signal Processing* **28**, 588–590 (1980).
- [200] Schuster, H. G. & Just, W. *Deterministic Chaos: An Introduction* (Wiley, Weinheim, 2005).
- [201] Rössler, O. An equation for continuous chaos. *Physics Letters A* **57**, 397–398 (1976).
- [202] Strogatz, S. H. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering* (Perseus Books, New York, 1994).
- [203] Virtanen, P. *et al.* Scipy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods* **17**, 261–272 (2020).
- [204] Ross, S. M. *Introduction to Probability Models* (Elsevier, 2006).
- [205] Taylor, J. Random fields. Disponível: [https://statweb.stanford.edu/~jtaylo/courses/stats352/notes/random\\_fields.pdf](https://statweb.stanford.edu/~jtaylo/courses/stats352/notes/random_fields.pdf). Acessado: 20 Out 2021.
- [206] Mandelbrot, B. B. & Ness, J. W. V. Fractional Brownian motions, fractional noises and applications. *SIAM Review* **10**, 422–437 (1968).
- [207] Hosking, J. R. M. Modeling persistence in hydrological time series using fractional differencing. *Water Resources Research* **20**, 1898–1908 (1984).
- [208] Gallon, S. fbm2d. Disponível: <https://github.com/sgallon/fbm2d>. Acessado: 20 Out 2021.
- [209] Diecker, T. *Simulation of fractional Brownian motion*. Ph.D. thesis, University of Twente (2004). URL <http://www.columbia.edu/~ad3217/fbm/thesis.pdf>.
- [210] Diecker, T. hosking.c. Disponível: <http://www.columbia.edu/~ad3217/fbm/hosking.c>. Acessado: 26 Nov 2020.
- [211] Voss, R. F. 1/f (flicker) noise: A brief review. In *33rd Annual Symposium on Frequency Control*, 40–46 (1979).
- [212] Kasdin, N. J. Discrete simulation of colored noise and stochastic processes and  $1/f^\alpha$  power law noise generation. *Proceedings of the IEEE* **83**, 802–827 (1995).
- [213] Timmer, J. & König, M. On generating power law noise. *Astronomy and Astrophysics* **300**, 707–710 (1995).

- [214] Patzelt, F. colorednoise.py. Disponível: <https://github.com/felixpatzelt/colorednoise>. Acessado: 26 Nov 2020.
- [215] Priestley, E. B. *Introduction to Liquid Crystals* (Springer, New York, 1976).
- [216] Bishop, C. M. *Pattern Recognition and Machine Learning*. Information Science and Statistics (Springer, New York, 2006).