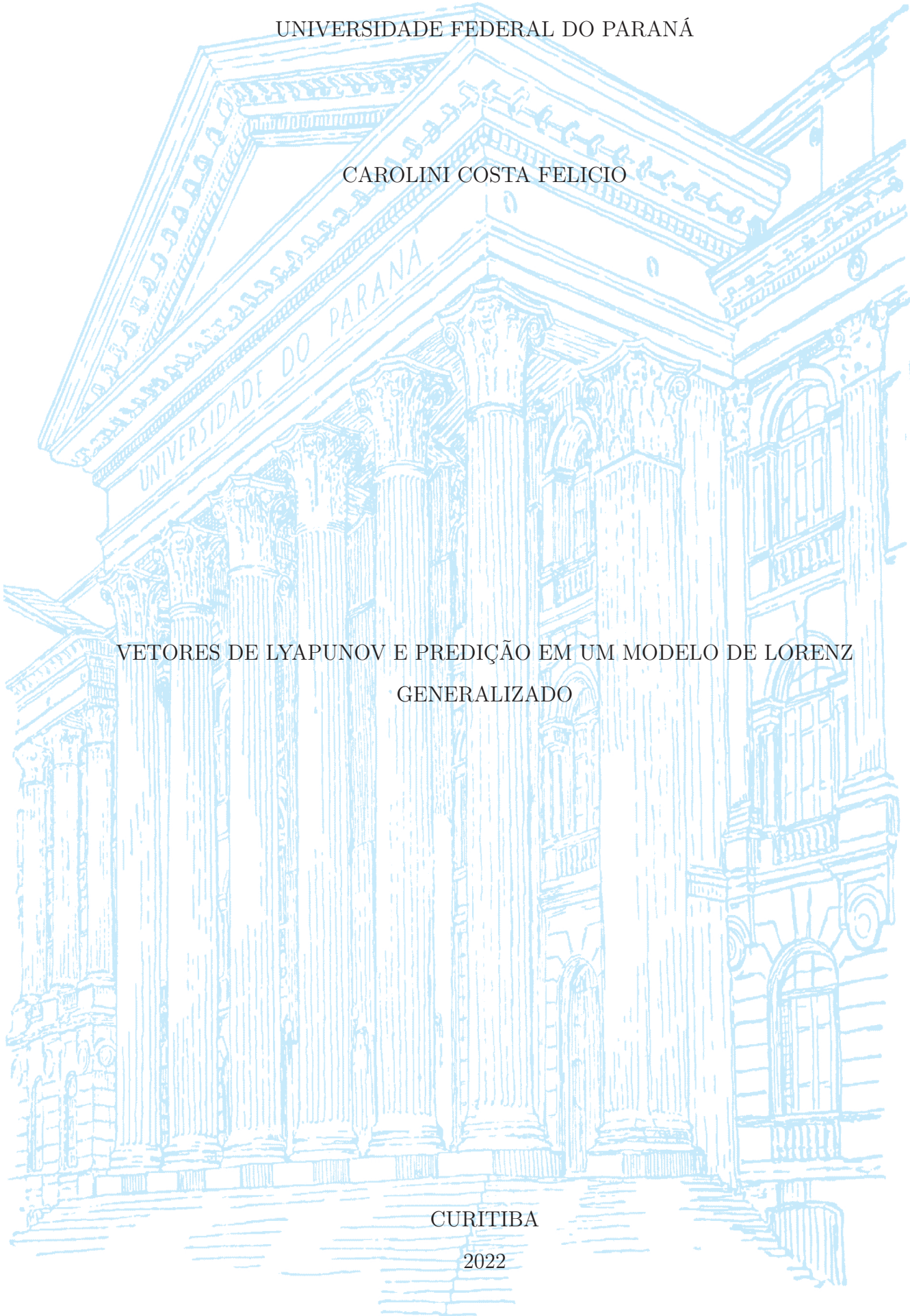




UNIVERSIDADE FEDERAL DO PARANÁ

CAROLINI COSTA FELICIO

VETORES DE LYAPUNOV E PREDIÇÃO EM UM MODELO DE LORENZ
GENERALIZADO

CURITIBA

2022

CAROLINI COSTA FELICIO

VETORES DE LYAPUNOV E PREDIÇÃO EM UM MODELO DE LORENZ
GENERALIZADO

Tese apresentada ao Programa de Pós-Graduação em Física, Setor de Ciências Exatas da Universidade Federal do Paraná, como requisito parcial para a obtenção do título de Doutora em Física.

Orientador: Prof. Dr. Marcus Werner Beims

CURITIBA

2022

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)
UNIVERSIDADE FEDERAL DO PARANÁ
SISTEMA DE BIBLIOTECAS – BIBLIOTECA CIÊNCIA E TECNOLOGIA

Felicio, Carolini Costa.

Vetores de Lyapunov e predição em um modelo de Lorenz
generalizado. / Carolini Costa Felicio. – Curitiba, 2022.

1 recurso on-line : PDF.

Tese (Doutorado) – Universidade Federal do Paraná, Setor de
Ciências Exatas, Programa de Pós-Graduação em Física.

Orientadora: Prof. Dr. Marcus Werner Beims.

1. Física. 2. *Machine learning*. 3. Lorenz, Modelos de. 4. Simulação
(Computadores). I. Beims, Marcus Werner. II. Universidade Federal do
Paraná. Programa de Pós-Graduação em Física. III. Título.

Bibliotecário: Nilson Carlos Vieira Junior CRB-9/1797

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação FÍSICA da Universidade Federal do Paraná foram convocados para realizar a arguição da tese de Doutorado de **CAROLINI COSTA FELICIO** intitulada: **"Vetores de Lyapunov e predição em um modelo de Lorenz generalizado"**, sob orientação do Prof. Dr. MARCUS WERNER BEIMS, que após terem inquirido a aluna e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa. A outorga do título de doutora está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 26 de Agosto de 2022.

Assinatura Eletrônica
26/08/2022 12:24:15.0
MARCUS WERNER BEIMS
Presidente da Banca Examinadora

Assinatura Eletrônica
26/08/2022 11:55:08.0
CÉLIA BEATRIZ ANTENEODO DE PORTO
Avaliador Externo (PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO)

Assinatura Eletrônica
26/08/2022 12:42:20.0
RICARDO LUIZ VIANA
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica
26/08/2022 11:56:49.0
CARLOS EDUARDO FIORE DOS SANTOS
Avaliador Interno (UNIVERSIDADE DE SÃO PAULO)

A Deus e ao amor.

AGRADECIMENTOS

A Deus, por sempre estar ao meu lado, atendendo minhas orações e olhando por mim. Seu amor e presença são constantes.

À minha mãe Neide Costa Felicio, pela vida, pelo exemplo de coragem, altruísmo e humildade que é e por tudo que fez e faz por mim.

Ao meu noivo Max Ruy de Souza Neto, pela compreensão, amor e apoio ao longo desses anos.

Aos meus avós Euclides Costa e Bernardina Tomasi Costa, por toda educação, cuidados e dedicação.

Ao meu orientador professor Marcus Werner Beims, pelo tempo disponibilizado, paciência, pelos ensinamentos, por me inspirar e por entender os problemas que tive.

Ao meu orientador de mestrado e iniciação científica professor Paulo Cesar Rech, pelos anos de orientação, conhecimento transmitido e por ser um exemplo para mim.

Aos amigos e familiares que estiveram ao meu lado.

Aos colegas de grupo, pelas contribuições dadas e ao amigo de trabalho Eduardo Luís Brugnago.

À PPGF-UFPR, pelo acolhimento e sensibilidade que tiveram comigo.

À CAPES, pelo suporte financeiro.

"O Senhor me ouviu quando o invoquei."

Sl 4,4

RESUMO

Neste trabalho investigamos a predição da duração de regimes utilizando um modelo de Lorenz generalizado, chamado de proto-Lorenz. O modelo de Lorenz clássico apresenta um atrator caótico que podemos dividi-lo em duas regiões mantendo cada uma das asas dentro de cada região. Os modelos aqui estudados tem atrator caótico para os mesmos parâmetros do modelo de Lorenz mas com uma complexidade maior, sendo essa o número de asas a mais. Para esses modelos encontramos as séries das variáveis, os vetores covariantes de Lyapunov, bem como os ângulos entre as variedades instável, estável e central. O alinhamento entre os vetores covariantes de Lyapunov precede a ocorrência de picos e troca de região em alguns sistemas. Com essas informações definimos as durações (ou classes) e o par de ângulos associados a ela com o intuito de aplicarmos técnicas de *machine learning* e o método de primeiros vizinhos *KNN* (*K-nearest neighbours*) para predição da duração dos regimes. Apresentamos figuras para o atrator, uma combinação de ângulos que predizem as durações (classes) e também figuras para analisarmos os métodos de aprendizagem de máquina utilizados. Acurácias maiores que 0,90 foram obtidas para predição de classes menores que três regiões subsequentes e acurácias maiores que 0,50 para predições de classes menores que sete, que significa mais que duas voltas completas no atrator no modelo com três asas. Para as durações do modelo com quatro asas obtemos previsões para até quatro regiões a frente, ou seja tomando dados de uma região no atrator estamos predizendo a duração para uma região que está uma volta completa a frente no atrator.

Palavras-chaves: Vetores de Lyapunov; Lorenz; *KNN*.

ABSTRACT

In this work we investigate the prediction of regime duration using a generalized Lorenz model, called proto-Lorenz. The classic Lorenz model presents a chaotic attractor that we can divide it into two regions keeping each of the wings inside each region. The models studied here have a chaotic attractor for the same parameters as the Lorenz model but with a greater complexity, which is the greater number of wings. For these models we find the series of variables, the Lyapunov covariant vectors, as well as the angles between the unstable, stable and central manifolds. Alignment between the covariant Lyapunov vectors precedes the occurrence of peaks and region switching in some systems. With this information we define the durations (or classes) and the pair of angles associated with it in order to apply *machine learning* techniques and the first neighbors method *KNN* (*K-nearest neighbors*) to predict the duration of the regimes. We present figures for the attractor, a combination of angles that predict durations (classes) and also figures to analyze the machine learning methods used. Accuracies greater than 0.90 were obtained for prediction of classes smaller than three subsequent regions and accuracies greater than 0.50 for predictions of classes smaller than seven, which means more than two complete turns of the attractor in the model with three wings. For the durations of the model with four wings, we get predictions for up to four regions ahead, that is, taking data from a region in the attractor we are predicting the duration for a region that is one complete revolution ahead in the attractor.

Key words: Lyapunov vectors; Lorenz; *KNN*.

LISTA DE FIGURAS

Figura 2.1	Uma trajetória em um espaço de fase tridimensional. Figura retirada da referência [29].	23
Figura 2.2	Evolução temporal de duas trajetórias de um sistema genérico a partir de condições iniciais infinitesimalmente próximas.	24
Figura 2.3	Esboço dos rolos de convecção, sem levar em conta efeitos de borda. Figura baseada na referência [54].	27
Figura 2.4	Atrator caótico do sistema de Lorenz 2.11 utilizando os valores dos parâmetros como $\sigma = 10,0$, $r = 28,0$ e $b = 8/3$	28
Figura 2.5	Atrator de Lorenz no plano $y \times x$. O atrator possui duas asas as quais estão separadas por regiões A e B conforme as cores na legenda.	29
Figura 2.6	Regime dentro de uma asa. O regime é considerado no momento que o atrator entra em uma região até o instante que sai dela.	30
Figura 2.7	Mesmo regime da Fig.2.6 mas ilustramos alguns vetores de forma didática. Os vetores em verde indicam o sentido do fluxo, em vermelho temos o vetor covariante de Lyapunov mais instável, em amarelo o VCL na direção do fluxo e em azul o VCL mais estável. O ângulo entre eles está apresentado de forma ilustrativa.	31
Figura 2.8	Ilustração para o método <i>KNN</i> . O ponto de interrogação sinaliza o elemento a ser classificado. O círculo tracejado indica o limite da vizinhança do ponto. As cores amarelo, verde e azul representam as classes dos pontos. Para esse exemplo, a classe do ponto é verde.	38

Figura 2.9	Diagrama esquemático para a MLP usada. Os retângulos pretos representam os neurônios na camada de entrada, os círculos na área cinza representam os neurônios nas camadas ocultas e o quadrado rosa é a camada de saída. As $W_{a,b}$ representam as matrizes de pesos e suas conexões entre os neurônios das diferentes camadas. Do ponto de vista de implementação, a camada de entrada de neurônios distribui informações dos ângulos extremos $\varphi_n^{i,j}$ para a primeira camada oculta, sem fazer transformações na entrada de informação. O processo de aprendizado de redes neurais artificiais ocorre atualizando os pesos das conexões entre as sucessivas camadas de neurônios. Os pesos são atualizados durante as temporadas de treinamento usando um algoritmo de retropropagação.	40
Figura 2.10	Atrator caótico do sistema de Lorenz 2.23.	42
Figura 2.11	Atrator caótico do sistema 2.24. Os parâmetros foram mantidos em $\sigma = 10,0$, $r = 28,0$ e $b = 8/3$	44
Figura 2.12	Atrator ilustrativo projetado no plano $q \times p$ com 81600 pontos e cores delimitando as asas localizadas nas regiões A, B e C. A direção do movimento segue a ordem $A \rightarrow B \rightarrow C \rightarrow A \dots$. Círculos azuis são as localizações dos pontos fixos instáveis relacionados a cada asa. A trajetória caótica começa na parte mais interna da asa, próxima ao ponto fixo (exemplificada pela linha verde dentro da região A do anel), move-se para a parte mais externa de cada asa (linha amarela para vermelha), até mudar de asa (linha tracejada em vermelho com a seta).	45

- Figura 3.1 O mesmo atrator da Fig. 2.12 mas com 10^5 pontos e cores adicionais (veja a paleta de cores) indicando os ângulos entre as variedades (a) instável e central, (b) instável e estável e (c) central e estável no espaço tangente. Os ângulos $\theta^{(2,3)}$ em (c) e $\theta^{(1,3)}$ em (b) tendem a $\pi/2$ próximos da origem (0,0), e a 0 nas partes mais externas (pontas das asas), antes de mudarem de asa. Isso ocorre para as três asas. O alinhamento da variedade instável e central ao longo da variedade estável precede uma mudança de asa como mostra a cor azul clara nas partes mais externas das asas em (b) e (c). Note que tal alinhamento precedendo mudanças de asas são mais fortes em (a), reconhecidos pela cor azul escuro. 49
- Figura 3.2 Janela temporal mostrando (a) (p,q) , (b) variável z e (c) o ângulo $\theta^{(2,3)}$. As cores de fundo identificam as regiões A, B e C da Fig. 2.12. Como um exemplo, em (b) nós mostramos as classes $\Omega_n = 4$ (máximo de z) dentro da região A. O menor valor de $\theta^{(2,3)} (= \varphi_n^{(2,3)})$ dentro de qualquer região ocorre antes da troca de região (veja os destaques em vermelho). Consequentemente, coletando todos os destaques em vermelho ao longo da série temporal, dados de $\varphi_n^{(2,3)}$ (o alinhamento mais forte de $\theta^{(2,3)}$) contém informação relevante quando a troca de região ocorre. 50
- Figura 3.3 Duração Δt_{n+1} do regime $n + 1$ como função do alinhamento extremo (a) $\varphi_n^{(1,2)}$; (b) $\varphi_n^{(1,3)}$; e (c) $\varphi_n^{(2,3)}$ ao longo do regime n . As cores são as classes da asa $n + 1$. Resultados em (a) demonstram claramente a projeção ao longo do eixo de ângulo são únicas, fazendo previsões de Δt_{n+1} , para um dado $\varphi_n^{(1,2)}$, muito impreciso. Tais imprecisões tendem a desaparecer para os ângulos $\varphi_n^{(1,3)}$ em (b), e $\varphi_n^{(2,3)}$ em (c). 52

- Figura 3.4 Classes Ω_{n+1} (veja a legenda de cores) como função dos ângulos $\varphi_n^{(1,3)}$ e $\varphi_n^{(2,3)}$. São 573111 pontos obtidos de 100 séries temporais. As classes são bem particionadas em regiões, o quadro interno mostra o padrão crescente das classes conforme o alinhamento aumenta. Em outras palavras, como os ângulos $\varphi_n^{(1,3)}$ e $\varphi_n^{(2,3)}$ diminuem, as classes aumentam, significando que maiores durações Δt_{n+1} são esperadas. 53
- Figura 3.5 Ocorrência de cada classes (pontos coloridos conforme legenda de cores da Fig. 3.4.) A ocorrência decai exponencialmente, como indicado pela linha lilás. Destacamos que a classe 20 inclui casos com 20 ou mais voltas em torno da mesma asa. 54
- Figura 3.6 Classes (a) Ω_{n+2} , (b) Ω_{n+3} , (c) Ω_{n+4} , (d) Ω_{n+5} , (e) Ω_{n+7} e (f) Ω_{n+10} são mostradas como função dos ângulos $\varphi_n^{(1,3)}$ e $\varphi_n^{(2,3)}$. As cores das classes seguem a legenda. O fundo branco é a região da classe $\Omega_{n+1} = 1$ da Fig. 3.4 e a cor de fundo bege é a região de classe $\Omega_{n+1} = 2$ da Fig. 3.4. Em (a) nós observamos que dentro da classe $\Omega_{n+1} = 1$ e $\Omega_{n+1} = 2$ da Fig. 3.4, novas sequências de altas-ordens de classes surgem. Esse surgimento de novas classes ocorre mais e mais, visivelmente em (b) e (c), e as cores se tornam mais e mais misturadas em (d)-(f), tornando a tarefa de previsão quase impossível para tempos longos. 55

- Figura 3.7 Comparação entre os resultados para MLP e método KNN para avanços $a = 1$, detalhados por classes. KNN com vizinhança $K = 1$. Em geral o método MLP (quadrados pretos) mostra uma menor acurácia comparada aos dois procedimentos KNN ; A métrica euclidiana ($KNNE$ em rosa) dá melhores resultados para classes $\Omega_{n+1} \leq 9$ do que a mínima distância ($KNNM$ em verde). Das classes $\Omega_{n+1} \geq 13$ ambas aplicações do KNN apresentam acurácias muito próximas. Classes 15 e 16 são altamente previsíveis pela MLP, também as classes $\Omega_{n+1} \geq 18$ tem resultado similar para o método KNN 58
- Figura 3.8 Taxas de acerto usando o método KNN para a asa Ω_{n+a} como função do inteiro $a = 1, 2, \dots, 10$. O número K de vizinhos é identificado pelas cores, e a distância medida entre os vizinhos é (a) euclidiana e (b) mínima distância. Enquanto em (a) dois vizinhos são suficientes para alcançar o efeito de muitos vizinhos, em (b) a acurácia é mais dependente do número de vizinhos usados. 59
- Figura 3.9 Acurácia em cada classe. (a) Melhores resultados para $K = 1$ vizinho com $a \leq 4$. (b) Melhores resultados para $K = 5$ vizinhos com $5 \leq a \leq 10$. Enquanto somente a acurácia para $a = 1$ não decai para classes altas, para todos os outros a ela decai. 60
- Figura 3.10 Acurácia em cada classe. Comparação entre diferentes métricas e a classificação com probabilidade Bayesiana. $KNNE$ para $K = 1$, $KNNM$ para $K = 5$ e $KNNMB$ para $K = 5$. A região em bege destaca os melhores resultados para $KNNMB$. Avanços (a) $a = 2$, (b) $a = 3$, (c) $a = 4$, (d) $a = 5$ e (e) $a = 6$. A região destacada em bege caminha a esquerda conforme o avanço aumenta, como podemos ver para $a > 4$, com exceção da classe $\Omega = 1$, $KNNMB$ tem acurácia maior para classes menores. 62

Figura 4.1	Atrator do sistema proto-Lorenz com 4 asas no plano $t \times s$. Foram utilizados 10^5 pontos e as cores (veja a paleta de cores) indicam os ângulos entre as variedades (a) instável e central, (b) instável e estável e (c) central e estável no espaço tangente. Todos os três ângulos tendem a zero nas extremidades das asas e a $\pi/2$ próximos da origem $(0,0)$	65
Figura 4.2	Janela temporal mostrando (a) (s,t) , (b) o ângulo $\theta^{(2,3)}$. As cores de fundo identificam as regiões A, B, C e D da Fig. 4.1. O menor valor de $\theta^{(2,3)} (= \varphi_n^{(2,3)})$ em cada região está destacado em vermelho ao longo da série.	66
Figura 4.3	Durações Δt_{n+1} (veja a legenda de cores) como função dos ângulos $\varphi_n^{(1,2)}$ e $\varphi_n^{(2,3)}$. São 571488 pontos obtidos de 100 séries temporais. As durações são bem divididas em regiões, o quadro interno mostra o padrão crescente da duração conforme o alinhamento aumenta. Em outras palavras, como os ângulos $\varphi_n^{(1,2)}$ e $\varphi_n^{(2,3)}$ diminuem, as durações aumentam.	67
Figura 4.4	Ocorrência de cada duração. A ocorrência decai exponencialmente, como indicado pela linha vermelha.	68
Figura 4.5	Durações (a) Δt_{n+1} , (b) Δt_{n+1} , (c) Δt_{n+1} são mostradas como função dos ângulos $\varphi_n^{(1,2)}$ e $\varphi_n^{(2,3)}$. As cores seguem a legenda. Em (a) notamos faixas nas quais as durações decaem conforme ambos os ângulos se aproximam de 0. Em (b) essas faixas são mais dificilmente observáveis e em (c) já temos uma mistura grande, o que torna a previsão complicada.	69
Figura 4.6	Erro relativo médio $\bar{\varepsilon}_{rel}$ em função das durações $\Delta t_{n+1}(t_L)$. $\Delta t_{n+1}(t_L)$ são em função do tempo de Lyapunov.	70
Figura 4.7	Quadro $\Delta t_{n+1}(t_L) \times \bar{\varepsilon}_{rel}$ onde em cores temos ρ que é quantidade de durações dentro de dum intervalo do erro relativo médio.	71

- Figura 4.8 Erro relativo médio $\bar{\varepsilon}_{rel}$ em função das durações $\Delta t_{n+a}(t_L)$. Em cores temos os valores dos avanços conforme a legenda. Para avanços menores temos $\bar{\varepsilon}_{rel}$ menores, assim como para durações $\Delta t_{n+a}(t_L)$ menores os erros também são menores. 72
- Figura 4.9 Quadro $\Delta t_{n+a}(t_L) \times \bar{\varepsilon}_{rel}$ onde em cores temos ρ que é quantidade de durações dentro de um intervalo de erro relativo médio. Em (a) $\Delta t_{n+2}(t_L)$, (b) $\Delta t_{n+2}(t_L)$ e em (c) $\Delta t_{n+3}(t_L)$ 73

LISTA DE SÍMBOLOS E SIGLAS

$\theta_n^{(i,j)}$	Ângulo entre os vetores covariantes de Lyapunov
$\varphi_n^{(i,j)}$	Ângulo extremo (mínimo de $\theta_n^{(i,j)}$ em um intervalo)
a	Avanço
Ω_{n+a}	Classe
$\Phi_i^{(a)}(\varphi_n^{(i,j)}, \varphi_n^{(i,j)}; \Omega_{i+a})$	Rótulo com ângulos extremos e classe
$\Gamma_i^{(a)}(\varphi_n^{(i,j)}, \varphi_n^{(i,j)}; \Delta t_{i+a}(t_L))$	Rótulo com ângulos extremos e duração em tempos de Lyapunov
$\bar{\varepsilon}_{rel}$	Erro relativo médio
Δt_{n+a}	Duração
$\Delta t_{n+a}(t_L)$	Duração em tempos de Lyapunov
ρ	Quantidade de durações dentro de um intervalo de $\bar{\varepsilon}_{rel}$
KNN	K -nearest neighbours ou K vizinhos mais próximos
KNNE	KNN com distância euclidiana
KNNM	KNN com mínima distância
KNNMB	KNN com mínima distância e probabilidade bayesiana
VCL	Vetores covariantes de Lyapunov

SUMÁRIO

1	INTRODUÇÃO	19
2	FUNDAMENTAÇÃO TEÓRICA	22
2.1	SISTEMAS DINÂMICOS	22
2.1.1	Definindo termos	27
2.2	VETORES DE LYAPUNOV	30
2.3	APRENDIZADO DE MÁQUINA	36
2.3.1	KNN	36
2.3.2	MLP	39
2.4	MODELO DE LORENZ GENERALIZADO	41
2.5	UTILIZANDO OS CONCEITOS	46
3	RESULTADOS - 3 ASAS	48
3.1	ÂNGULOS ENTRE OS VCLs	48
3.2	CLASSES E PREDIÇÃO	52
3.3	MLP e KNN	56
3.3.1	KNN com probabilidade Bayesiana	61
4	RESULTADOS - 4 ASAS	64
4.1	ÂNGULOS ENTRE OS VCLs	64
4.2	DURAÇÕES E PREDIÇÃO	66
4.3	KNN	69
5	CONCLUSÃO	74
	REFERÊNCIAS	78
	APÊNDICE A	85

Capítulo 1

INTRODUÇÃO

A capacidade de prever o futuro com base em dados passados e presentes é de grande relevância em áreas distintas, como mudanças climáticas, bolsas de valores, transições críticas e eventos extremos e raros [1]. Até agora, não há quantidade física adequada capaz de fazer previsões precisas sobre o futuro em sistemas caóticos [2]. Assim, qualquer desenvolvimento adicional na tarefa de predição com base em dados passados e presentes em sistemas físicos específicos é altamente desejável.

Vetores covariantes de Lyapunov contêm um registro do que exatamente acontece com os ângulos entre variedades estáveis e instáveis durante toda a evolução do sistema, uma informação apagada permanentemente por ortogonalizações repetidas [3]. O alinhamento entre os vetores ao longo de direções específicas permite antecipar a ocorrência de grandes picos em séries temporais caóticas [4, 5] e também transições críticas [6]. O cálculo para os ângulos foi proposto por Ginelli *et al* [7] e são obtidos integrando as equações de movimento para frente e para trás ao longo da mesma trajetória [7–11] e foram determinados em muitos sistemas realistas [3, 12–20]. Previsões de transições entre asas do atrator de Lorenz, e sua duração, foi estudado usando o conceito de *bred-vectors* [21], o máximo de um pico (para uma determinada variável) logo antes da mudança de região [22] e também usando técnicas de aprendizado de máquina [1]. Aqui nos propomos utilizar o alinhamento entre os vetores covariantes de Lyapunov combinando com procedimentos de classificação e aprendizado de máquina aplicados a um modelo generalizado do sistema de Lorenz [23] a fim de prever a duração de regimes a frente.

O sistema apresentado por Lorenz em seu artigo compunha três equações diferenciais

ordinárias autônomas de primeira ordem, as quais continham três variáveis, $x(t)$, $y(t)$ e $z(t)$, e três parâmetros: σ , r e b [23], este sistema possui um atrator caótico para uma combinação de valores dos parâmetros e esse atrator pode ser dividido em regiões. Este modelo foi amplamente estudado e vários resultados foram publicados sobre o mesmo. Em 1993, Miranda e Stone publicaram o artigo *The proto-Lorenz System* [24] onde apresentaram um modelo generalizado do modelo de Lorenz contendo os mesmos parâmetros do modelo de Lorenz e chamado de *proto-Lorenz*. O modelo *proto-Lorenz* possui duas versões que foram estudadas aqui, uma contendo as variáveis $p(t)$, $q(t)$ e $z(t)$ que vamos chamar de sistema com 3 asas, e outra com as variáveis $s(t)$, $t(t)$ e $z(t)$ que nomeamos de modelo 4 asas.

A informação do alinhamento entre os VCLs para o modelo generalizado de Lorenz foi combinada com o máximo da variável z do sistema e a essa demos o nome de *classe* para o caso 3 asas. O rótulo gerado pela combinação de ângulos mais a classe nos permitiu aplicar métodos de aprendizado de máquina. Analisando os dados que obtemos, optamos pelo uso de dois métodos de aprendizado, ambos supervisionados: o *KNN* [25] e redes do tipo MLP [26]. O *KNN* faz uma associação das classes com relação aos vizinhos do elemento teste a fim de determinar a classe desse, já a MLP é uma rede neural com camada de entrada, camadas ocultas e camada de saída que trabalha com pesos e atualizações para o aprendizado ocorrer. Combinamos ainda a informação do alinhamento entre os VCLs para o caso 4 asas mas utilizando diretamente a informação da duração do regime. O rótulo que geramos aqui também nos permitiu a aplicação do método *KNN*.

Os resultados obtidos para as técnicas de aprendizado de máquina combinadas com o alinhamento dos VCLs são mostrados nos capítulos *Resultados*. Com os dados para a região n do atrator, encontramos as durações (ou classes) dos regimes a frente. Em outras palavras, conseguimos usar dados de uma região atual n e obter as durações (ou classes) regiões bem a frente no atrator. Utilizando esses dados, aplicamos a MLP e versões diferentes do *KNN* no caso 3 asas e alcançamos acurácias superiores a 0,90 para até três asas visitadas à frente e também, acurácias superiores a 0,50 para até seis avanços. Os resultados são comparados e mostrados nas últimas figuras. No caso 4 asas optamos pelo método *KNN* e obtemos resultados para previsão de até quatro asas visitadas a frente.

Este trabalho compreende cinco capítulos. No capítulo 2, nomeado *Fundamentação Teórica* apresentamos de forma sucinta aspectos teóricos da área de sistemas dinâmicos os quais utilizamos para construção e compreensão dos resultados obtidos. Ainda, apresentamos os termos *região*, *regime* e *asa* que são frequentemente usados para auxiliar na compreensão. Também expomos as técnicas de aprendizado de máquina aplicadas ao sistema estudado. O capítulo *Resultados - 3 asas* inicia apresentando o atrator do sistema 3 asas com os ângulos, na sequência apresentamos as classes e a combinação de ângulo, bem como as figuras para os avanços no atrator. Concluimos o capítulo *Resultados - 3 asas* trazendo as comparações entre as acurácias dos métodos de aprendizagem de máquina aplicados. Já o capítulo *Resultados - 4 asas* apresenta os resultados obtidos para o caso 4 asas, sendo mostrado inicialmente o atrator desse sistema juntamente com os ângulos entre os VCLs, depois apresentamos uma combinação dos ângulos extremos com as durações dos regimes a frente e encerramos o capítulo analisando os resultados para o método *KNN* aplicado. Por fim, apresentamos as conclusões e as referências utilizadas neste estudo.

Capítulo 2

FUNDAMENTAÇÃO TEÓRICA

A fim de compreendermos as discussões e resultados apresentados no capítulo seguinte, faz-se necessária a apresentação de alguns conceitos. Os pontos abordados na seção de *Sistemas Dinâmicos* foram amplamente discutidos em bibliografias [27–30] ao longo dos anos, e portanto aqui expomos resumidamente aqueles pertinentes ao trabalho. Também é importante o conhecimento do modelo generalizado de Lorenz utilizado. A seguir, abordaremos sobre sistemas dinâmicos, vetores de Lyapunov, aprendizado de máquina e sobre o modelo utilizado neste estudo.

2.1 SISTEMAS DINÂMICOS

Sistemas cujos estados futuros evoluem de seus estados atuais de acordo com leis físicas ou equações matemáticas são conhecidos como *sistemas dinâmicos* [31], como exemplo temos aqueles que caracterizam circuitos elétricos [32] e modelo de neurônios [33]. O tempo desta evolução pode ser considerado como uma variável contínua ou como uma variável discreta, sendo o último caso representado por *mapas*. Comumente chamados de *fluxo*, os sistemas a tempo contínuo são descritos por equações diferenciais, como no

exemplo:

$$\begin{aligned}\frac{dx_1}{dt} &= f_1(x_1, x_2, \dots, x_n, t), \\ \frac{dx_2}{dt} &= f_2(x_1, x_2, \dots, x_n, t), \\ &\vdots \\ \frac{dx_n}{dt} &= f_n(x_1, x_2, \dots, x_n, t),\end{aligned}\tag{2.1}$$

que apresenta um sistema de equações diferenciais de primeira ordem. Ou na forma vetorial, representamos genericamente um sistema a tempo contínuo por:

$$\frac{d\vec{X}}{dt} = \vec{F}[\vec{X}(t)],\tag{2.2}$$

onde $\vec{X}$ é um vetor n -dimensional e $\vec{F}$ é uma função vetorial desta variável [29]. Para nossos estudos, vamos considerar um modelo a tempo contínuo.

Agora, considerando um espaço de fase n -dimensional, no qual os eixos coordenados são as variáveis dinâmicas do sistema e onde um estado do sistema é representado por um ponto com coordenadas $x_1(t), x_2(t), \dots, x_n(t)$, este se moverá conforme o sistema é evoluído. A medida que o tempo passa, o ponto descreve um caminho e este é chamado de *trajetória*.

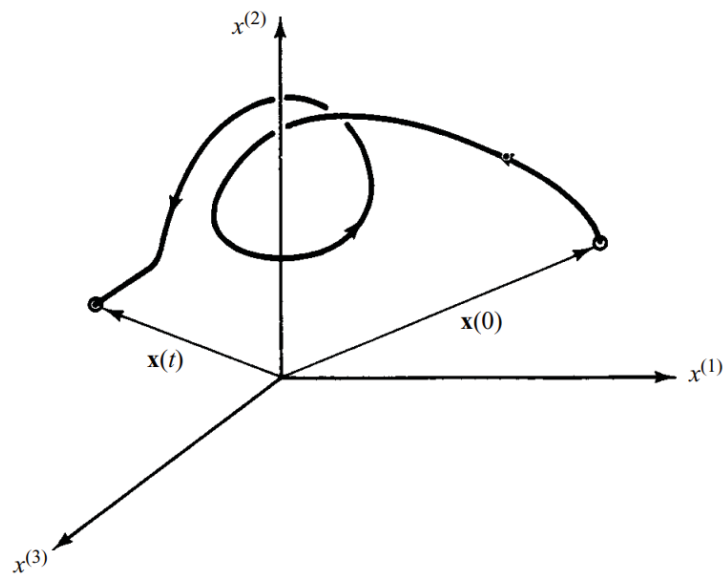


Figura 2.1 – Uma trajetória em um espaço de fase tridimensional. Figura retirada da referência [29].

Olhando para o espaço de fase do sistemas podemos diferenciá-lo entre dissipativo e conservativo. Sistemas *dissipativos* são caracterizados pela contração no volume do espaço de fase conforme o tempo passa [34], enquanto que para sistemas *conservativos* ocorre a conservação deste volume. O modelo estudado é definido como dissipativo. Para diferenciarmos um sistema dissipativo de um conservativo, vamos começar definindo uma trajetória. Considerando um espaço de fase n -dimensional, no qual os eixos coordenados são as variáveis dinâmicas do sistema e onde um estado do sistema é representado por um ponto com coordenadas $x_1(t), x_2(t), \dots, x_n(t)$, este se moverá conforme o sistema é evoluído. A medida que o tempo passa, o ponto descreve um caminho e este é chamado de *órbita* (ou *trajetória*). A Figura 2.1 mostra o espaço de fase de um sistema tridimensional e a trajetória seguida pelo sistema.

Agora, vamos imaginar um sistema de n equações ordinárias onde duas trajetórias, $\vec{x}_1(t_0)$ e $\vec{x}_2(t_0)$ são iniciadas infinitesimalmente próximas, conforme está representado na Figura 2.2. À medida que o sistema evolui, as trajetórias se afastam exponencialmente, ou

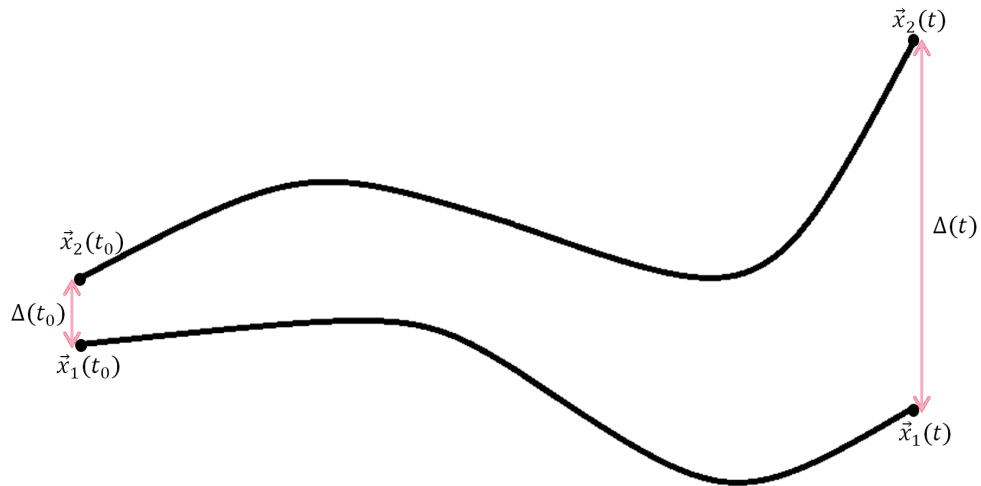


Figura 2.2 – Evolução temporal de duas trajetórias de um sistema genérico a partir de condições iniciais infinitesimalmente próximas.

seja, duas condições iniciais escolhidas arbitrariamente próximas (após passado um tempo adequadamente longo) levaram a soluções bem diferentes. Isto nos leva a considerar uma dependência sensível às condições iniciais e o *expoente de Lyapunov* quantificará a taxa de divergência exponencial entre trajetórias de um mesmo sistema, obtidas a partir de condições iniciais ligeiramente diferentes [35]. Para melhor entendermos, vamos considerar uma hipersfera de condições iniciais no ponto $\vec{x}(t_0)$. Conforme o tempo evolui, o volume

desta hipersfera se deformará. Sendo que ao longo de cada dimensão n , o raio inicial $d_j(t_0)$ varia exponencialmente no tempo, tal que a relação entre $d_j(t_0)$ e o valor correspondente em t , dado por $d_j(t)$, pode ser escrito como:

$$d_j(t) = d_j(t_0)e^{\lambda_j(t-t_0)}, \quad (2.3)$$

com $j = 1, 2, \dots, n$. Teremos então:

$$\lambda_j = \lim_{t \rightarrow \infty} \frac{\ln[d_j(t)/d_j(t_0)]}{(t - t_0)}. \quad (2.4)$$

Sendo os λ_j os expoentes de Lyapunov. Podemos observar que para o caso de comportamento periódico, onde a distância entre duas condições iniciais vizinhas se mantém constante na média, o expoente associado a esta direção será nulo e nas demais direções haverá contração do volume no espaço de fase. Para um comportamento caótico, o qual apresenta divergência exponencial de trajetórias vizinhas, temos pelo menos um expoente positivo. Isso implica na sensibilidade nas condições iniciais. Vale salientar que o número de expoentes de Lyapunov está associado a dimensão n do sistema. Considerando que o volume da hipersfera no tempo t é $V(t)$, podemos escrever:

$$V(t) \propto \prod_{j=1}^n d_j(t). \quad (2.5)$$

Ou, utilizando 2.3, temos:

$$V(t) \propto V(t_0)e^{(t-t_0)\sum_{j=1}^n \lambda_j}. \quad (2.6)$$

Agora podemos utilizar a equação (2.6) para distinguirmos um sistema conservativo de um dissipativo. Pois, se o sistema é conservativo, teremos:

$$V(t) = V(t_0). \quad (2.7)$$

Portanto,

$$\sum_{j=1}^n \lambda_j = 0. \quad (2.8)$$

E, se um sistema é dissipativo então

$$V(t) < V(t_0), \quad (2.9)$$

Assim:

$$\sum_{j=1}^n \lambda_j < 0. \quad (2.10)$$

Isto verifica-se pelo teorema de Liouville [36, 37]. Sistemas dissipativos são tipicamente caracterizados pela presença de *atratores* no espaço de fase [29]. Este é um conjunto limitado numa região no espaço de fase para o qual um volume de condições iniciais convergem a medida que o tempo evolui. Podemos classificar os tipos de atratores utilizando o expoente de Lyapunov, assim para um sistema com três equações diferenciais, teremos:

- *ponto de equilíbrio*: $\lambda_1 < 0, \lambda_2 < 0, \lambda_3 < 0$, aqui a trajetória convergirá para um ponto no espaço de fase;
- *ciclo-limite ou atrator periódico*: $\lambda_1 = 0, \lambda_2 < 0, \lambda_3 < 0$, com o expoente nulo correspondendo à direção ao longo da trajetória fechada;
- *superfície toroidal ou atrator quase-periódico*: $\lambda_1 = 0, \lambda_2 = 0, \lambda_3 < 0$, onde as trajetórias atratoras se encontram sobre uma superfície;
- *atrator caótico*: $\lambda_1 > 0, \lambda_2 = 0, \lambda_3 < 0$, onde o expoente positivo está associado à direção na qual a trajetória diverge no espaço de fase.

Vamos utilizar o conceito de atrator nos resultados apresentados no capítulo seguinte, bem como o expoente de Lyapunov.

2.1.1 Definindo termos

Região, *regime* e *asa* são termos que utilizamos com frequência neste trabalho e assim, cabe defini-los aqui. Com o intuito de facilitar o entendimento, vamos tomar o sistema de Lorenz [23] como modelo. Lorenz apresentou em 1963 [23] um sistema simplificado do modelo de Saltzman [51]. Ele notou que os valores de quatro, das sete variáveis deste sistema de Saltzman, tornavam-se muito pequenas rapidamente. O sistema (2.11) é autônomo de primeira ordem com três equações diferenciais ordinárias as quais formam um modelo matemático para a convecção de Rayleigh-Bénard em um fluido [52, 53]. O fluido está contido entre duas placas paralelas colocadas na direção perpendicular à força gravitacional da Terra. As placas são aquecidas por baixo e refrigeradas por cima, como esboça a Figura 2.3.

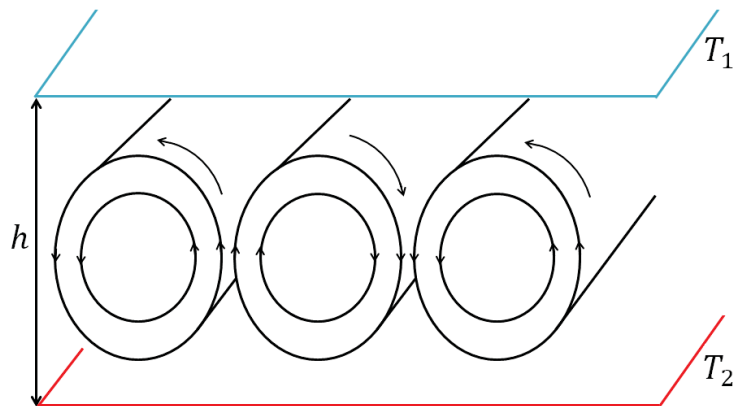


Figura 2.3 – Esboço dos rolos de convecção, sem levar em conta efeitos de borda. Figura baseada na referência [54].

O sistema possui três variáveis, onde a variável $x(t)$ é proporcional à intensidade da convecção, a variável $y(t)$ é proporcional à diferença de temperatura entre as correntes ascendentes e descendentes do fluido e a variável $z(t)$ é proporcional à distorção do perfil vertical de temperatura em relação ao perfil linear e, ainda, esse sistema contém σ, r, b como parâmetros. Podemos escrever o sistema de Lorenz como:

$$\dot{x} = \sigma y - \sigma x,$$

$$\dot{y} = rx - y - xz, \quad (2.11)$$

$$\dot{z} = xy - bz.$$

Para um estudo aprofundado sobre o sistema (2.11) temos o livro de Sparrow [55], e mais resumidamente encontramos em Dullin [56].

O modelo de Lorenz apresenta um atrator caótico para os parâmetros $\sigma = 10,0$, $r = 28,0$ e $b = 8/3$, como mostra a Fig. 2.4. Olhando para o atrator da Fig. 2.4 e associando sua forma com uma borboleta, temos o termo *asa* como sendo uma parte do atrator circunda um ponto fixo. Para o modelo 2.11 temos 2 asas.

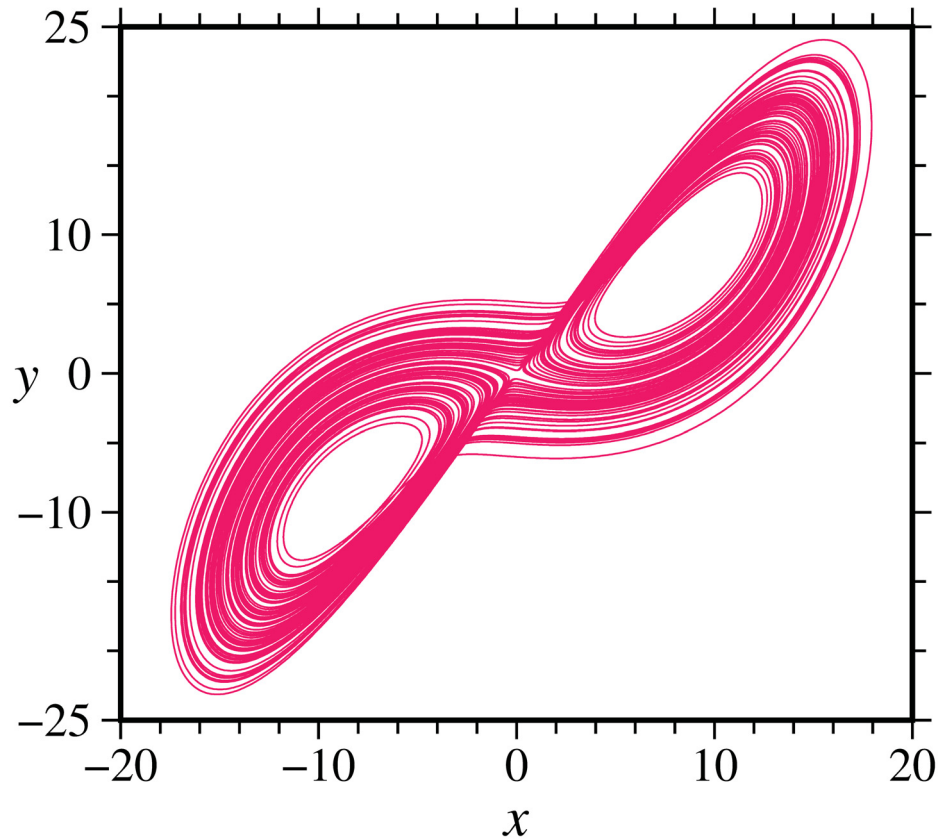


Figura 2.4 – Atrator caótico do sistema de Lorenz 2.11 utilizando os valores dos parâmetros como $\sigma = 10,0$, $r = 28,0$ e $b = 8/3$.

Agora, vamos imaginar uma separação no atrator para $x > 0$ e $x < 0$. Chamaremos essas de *regiões* e elas são definidas como divisões no espaço de fases. A Fig. 2.5 mostra a divisão feita com as duas regiões coloridas e chamadas de A e B . Note que poderíamos ter

separado dividido o espaço em diferentes formas ou até em outras quantidades de regiões. Aqui escolhemos essa maneira pois nos asseguramos que dentro de cada região temos uma asa do atrator.

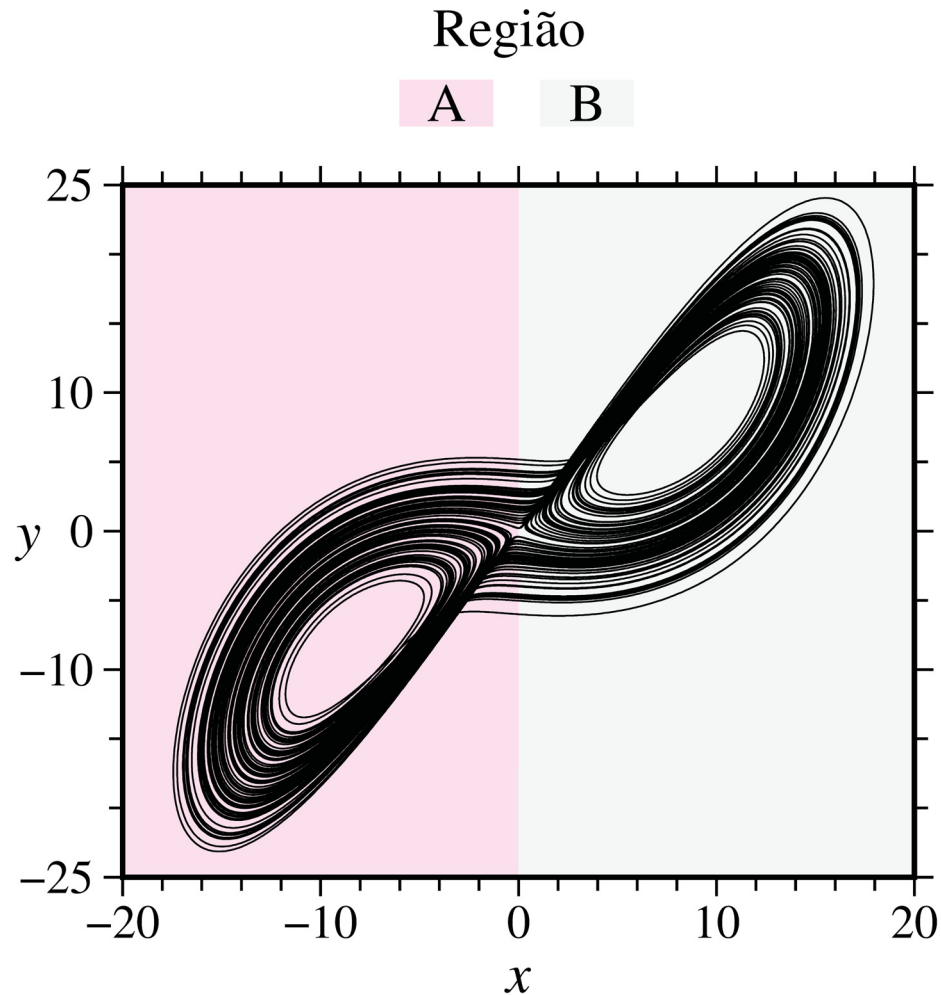


Figura 2.5 – Atrator de Lorenz no plano $y \times x$. O atrator possui duas asas as quais estão separadas por regiões A e B conforme as cores na legenda.

Cada asa do atrator pode ser visitada muitas vezes. O *regime* é definido como movimento descrito pela partícula quando ela entra em uma asa até o momento que ela sai. Escolhemos um trecho arbitrário no tempo para exemplificar um regime na Fig. 2.6.

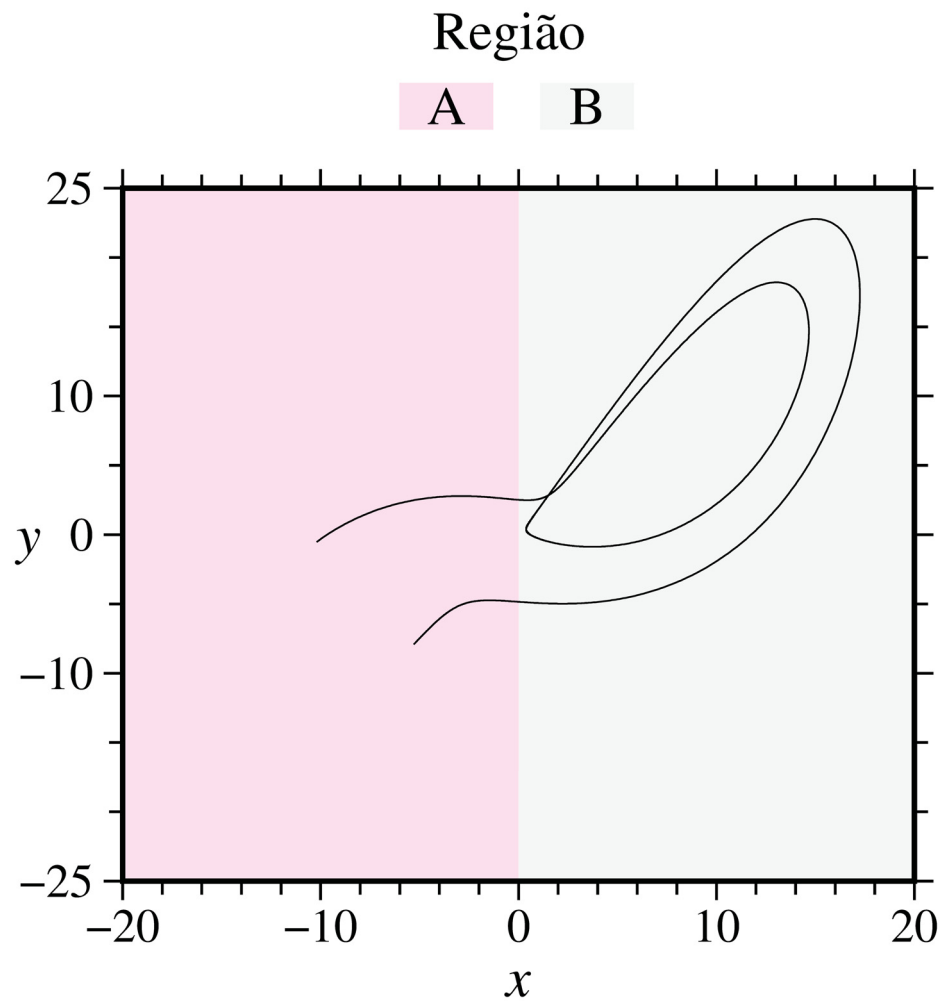


Figura 2.6 – Regime dentro de uma asa. O regime é considerado no momento que o atrator entra em uma região até o instante que sai dela.

2.2 VETORES DE LYAPUNOV

A dinâmica de sistemas pode ser investigada utilizando os expoentes de Lyapunov vistos na seção anterior, como também os correspondentes vetores de Lyapunov, conhecidos na literatura por vetores covariantes de Lyapunov [7,8]. Estes têm atraído interesse pelas diferentes informações que podemos obter de sistemas complexos. Para introduzirmos os expoentes de Lyapunov, imaginamos uma hipersfera de condições iniciais. Agora, consideremos que um conjunto infinito de estados iniciais preencha a superfície de uma esfera infinitesimal no espaço de fase e que evoluímos o nosso sistema, tal que os estados evoluídos recairão sob a superfície de um elipsóide infinitesimal. Os expoentes de Lyapu-

nov são as médias, em um tempo longo, de quanto os eixos aumentaram ou diminuíram por unidade de tempo [31]. Usamos estes expoentes para diferenciar caos e regularidade, sendo eles obtidos através da norma dos *vetores de Lyapunov* ortogonalizados, propagados durante a evolução temporal da dinâmica [3]. Estes vetores fornecem um registro detalhado do que exatamente acontece com os ângulos entre as variedades estáveis e instáveis durante a evolução de um processo dinâmico [5]. Faz-se interessante pois poderemos verificar se existe alinhamento entre os vetores e se isto ocorre precedendo um máximo (ou um mínimo) do sistema, conforme resultados apresentados em [4].

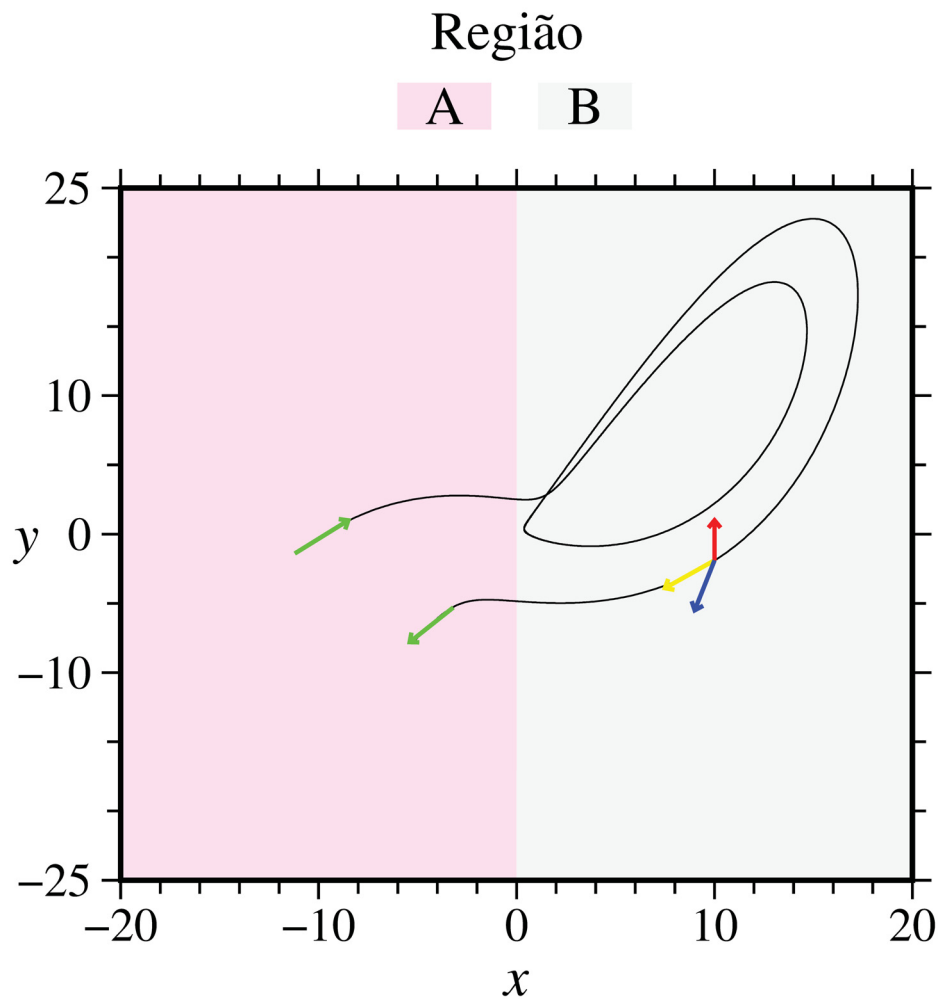


Figura 2.7 – Mesmo regime da Fig.2.6 mas ilustramos alguns vetores de forma didática. Os vetores em verde indicam o sentido do fluxo, em vermelho temos o vetor covariante de Lyapunov mais instável, em amarelo o VCL na direção do fluxo e em azul o VCL mais estável. O ângulo entre eles está apresentado de forma ilustrativa.

O cálculo para determinar o expoente de Lyapunov é bastante utilizado e pode ser

visto em [38–40], com este método determinamos a direção mais instável de uma trajetória caótica, sendo esta associada ao maior expoente de Lyapunov, mas perdemos as informações das demais direções não vinculadas ao maior expoente, uma vez que este processo utiliza da ortonormalização de Gram-Schmidt. Como precisamos destas informações para encontrarmos os ângulos entre os vetores, vamos utilizar o método descrito em [7,8] o qual evolui o sistema para frente e para trás no tempo recuperando as direções das variedades instáveis (nestes casos temos somente uma variedade instável) e estáveis. Para um exemplo usando um sistema tridimensional, na evolução para "frente" no tempo temos três vetores de Lyapunov: um associado à direção instável representado pela cor vermelha na Fig. 2.7, outro na direção do fluxo colorido de amarelo e o terceiro que está associado a direção estável e escolhemos a cor verde. Um artifício utilizado aqui é que o vetor mais instável na ida será o estável na evolução para "trás" no tempo. Seguindo [5] vamos apresentar brevemente o procedimento para obtenção dos ângulos de um sistema tridimensional, este começa com a evolução do sistema para frente no tempo e deixa-se passar um transiente suficientemente longo. Para a interação utilizamos o método de Runge-Kutta de quarta ordem [41] e após o transiente, novamente evoluímos o sistema integrando as equações do sistema, juntamente com as equações linearizadas:

$$\begin{pmatrix} \Delta x(t+h) \\ \Delta y(t+h) \\ \Delta z(t+h) \end{pmatrix} = \mathbf{J}(t) \begin{pmatrix} \Delta x(t) \\ \Delta y(t) \\ \Delta z(t) \end{pmatrix}, \quad (2.12)$$

sendo esta a equação que governa a evolução temporal do sistema e $\mathbf{J}(t)$ é a matriz

Jacobiana do mesmo, escrita como:

$$\mathbf{J}(t) = \begin{pmatrix} \frac{\partial \dot{x}}{\partial x} & \frac{\partial \dot{x}}{\partial y} & \frac{\partial \dot{x}}{\partial z} \\ \frac{\partial \dot{y}}{\partial x} & \frac{\partial \dot{y}}{\partial y} & \frac{\partial \dot{y}}{\partial z} \\ \frac{\partial \dot{z}}{\partial x} & \frac{\partial \dot{z}}{\partial y} & \frac{\partial \dot{z}}{\partial z} \end{pmatrix}. \quad (2.13)$$

Escolhendo $\Delta x(0) = \mathbf{a}_0^{(1)}$, $\Delta y(0) = \mathbf{a}_0^{(2)}$ e $\Delta z(0) = \mathbf{a}_0^{(3)}$ como uma base ortonormal inicial $\mathbf{G}_0 = (\mathbf{a}_0^{(1)}, \mathbf{a}_0^{(2)}, \mathbf{a}_0^{(3)})$, a evolução no espaço tangente é obtida por:

$$\tilde{\mathbf{G}}_n = \mathbf{J}_n \mathbf{G}_0, \quad (2.14)$$

com $\mathbf{J}_n = \mathbf{J}_{n-1} \times \mathbf{J}_n, \dots, \times \mathbf{J}_0$ sendo o produto da matriz Jacobiana do sistema evoluído para cada ponto da órbita. Para que não haja divergência, a matriz $\tilde{\mathbf{G}}_n$ é renormalizada a cada passo e isso pode ser feito através de uma decomposição QR, como apresentada no *Apêndice A*:

$$\tilde{\mathbf{G}}_n = \mathbf{G}_n \mathbf{R}_n. \quad (2.15)$$

A matriz $\mathbf{R}_n$ será uma triangular superior e contém as informações obtidas no processo de ortonormalização de Gram-Schmidt de $\tilde{\mathbf{G}}_n$:

$$\mathbf{R}_n = \begin{pmatrix} \mathbf{g}_n^{(1)} \cdot \mathbf{a}_n^{(1)} & \mathbf{g}_n^{(1)} \cdot \mathbf{a}_n^{(2)} & \mathbf{g}_n^{(1)} \cdot \mathbf{a}_n^{(3)} \\ 0 & \mathbf{g}_n^{(2)} \cdot \mathbf{a}_n^{(2)} & \mathbf{g}_n^{(2)} \cdot \mathbf{a}_n^{(3)} \\ 0 & 0 & \mathbf{g}_n^{(3)} \cdot \mathbf{a}_n^{(3)} \end{pmatrix}, \quad (2.16)$$

onde $\mathbf{G}_n = (\mathbf{a}_n^{(1)}, \mathbf{a}_n^{(2)}, \mathbf{a}_n^{(3)})$ são os vetores antes da ortonormalização em um tempo n e os $(\mathbf{g}_n^{(1)}, \mathbf{g}_n^{(2)}, \mathbf{g}_n^{(3)})$ são os vetores já ortonormais. Desta forma só obtemos as taxas de

expansão (ou contração) local dos vetores, sendo estes fornecidos pela diagonal principal de $\mathbf{R}_n$. Devido à perda da informação sobre a direção dos autovetores da Jacobiana, utilizaremos a evolução para trás no tempo como proposto em [8]. Aqui nos valem do fato que para a evolução para trás (volta) no tempo a direção instável é a direção estável da evolução para frente (ida) no tempo. Então, o vetor de Lyapunov relacionado ao maior expoente de Lyapunov na evolução para trás no tempo aponta ao longo da direção estável na evolução para frente. Assim podemos escrever para os vetores:

$$\mathbf{v}_n^{(1)} = c_n^{(1,1)} \mathbf{g}_n^{(1)}, \quad (2.17)$$

$$\mathbf{v}_n^{(2)} = c_n^{(1,2)} \mathbf{g}_n^{(1)} + c_n^{(2,2)} \mathbf{g}_n^{(2)}, \quad (2.18)$$

$$\mathbf{v}_n^{(3)} = c_n^{(1,3)} \mathbf{g}_n^{(1)} + c_n^{(2,3)} \mathbf{g}_n^{(2)} + c_n^{(3,3)} \mathbf{g}_n^{(3)}. \quad (2.19)$$

Conforme já calculamos os $\mathbf{g}_n^{(1)}, \mathbf{g}_n^{(2)}$ e $\mathbf{g}_n^{(3)}$ na evolução da ida e armazenamos, os coeficientes de $c_n^{(j,i)}$ podem ser determinados pela dinâmica no espaço tangente na direção da volta, fazendo:

$$\mathbf{C}_{j-1} = \mathbf{R}_j^{-1} \mathbf{C}_j, \quad (2.20)$$

onde $\mathbf{R}_j^{-1}$ é a inversa da matriz (2.16) e o tempo j inicia após passarmos um transiente suficientemente longo. Como o comprimento dos vetores de Lyapunov são normalizados, as colunas de $\mathbf{C}_j$ devem ser normalizadas para 1. Desta forma, os ângulos entre os vetores de Lyapunov podem ser calculados por:

$$\theta^{(i,j)} = \cos^{-1}(\mathbf{v}_n^{(i)} \cdot \mathbf{v}_n^{(j)}). \quad (2.21)$$

Um pseudocódigo foi proposto por Beims e Gallas [5] para obtenção dos vetores e ângulos e este será apresentado com o intuito de resumir os principais passos necessários.

PROGRAM Vetores de Lyapunov

Define variables $Y(12)$

comentário $Y(1) = p, Y(2) = q, Y(3) = z$ **variáveis de estado**

comentário $Y(4) \dots Y(12)$ **3 conjuntos de variáveis linearizadas**

comentário **ODE (Integrador de equações diferenciais ordinárias)**

comentário **GS (determina os expoentes de Lyapunov)**

do loop *tempo para frente - transiente*

call *ODE*

call *GS*

end loop *tempo - transiente*

do loop *tempo para frente*

call *ODE*

call *GS*

save $Y(1), \dots, Y(12)$ *para todos os tempos*

save R_n *para todos os tempos*

end loop *tempo para frente*

do loop *tempo para trás - transiente*

determine $c^{(i,j)}$ *de* $C_{j-1} = R_j^{-1} C_j$ *para cada tempo*

normalize C_{j-1} *para cada tempo*

end loop *tempo - transiente*

do loop *tempo para trás*

determine $c^{(i,j)}$ *de* $C_{j-1} = R_j^{-1} C_j$ *para cada tempo*

normalize C_{j-1} *para cada tempo*

calculate $v^{(1)}, v^{(2)}, v^{(3)}$ *para cada tempo*

determine $\theta^{(i,j)}$ *e escreve* $Y(1), Y(2), Y(3), \theta^{(i,j)}$ *para cada tempo*

end loop *tempo*

end PROGRAM

2.3 APRENDIZADO DE MÁQUINA

Definimos o aprendizado de máquina, também chamado de *Machine Learning*, como um conjunto de métodos que podem detectar automaticamente padrões nos dados e, em seguida, usar os padrões descobertos para prever dados futuros ou para realizar outros tipos de tomada de decisão sob incerteza [42]. Um algoritmo de aprendizado de máquina é um algoritmo capaz de aprender a partir de dados [43]. Uma definição para o que queremos dizer com aprender é quando um programa de computador aprende com a experiência E em relação a alguma classe de tarefas T e a medida de desempenho P , se seu desempenho em T , medido por P , melhora com a experiência E [44].

No aprendizado de máquina temos o passo de treinamento, o qual podemos dividir em *supervisionado* ou *não supervisionado*. Na abordagem *supervisionada*, o objetivo é aprender um mapeamento das entradas x para as saídas y , dado um conjunto rotulado de pares de entrada-saída $\mathcal{D} = (x_i, y_i)_{i=1}^N$. Aqui $\mathcal{D}$ é chamado de conjunto de treinamento e N é o número de exemplos de treinamento [42].

Podemos usar aprendizado de máquina para problemas de classificação e neste caso, o programa de computador é solicitado a especificar a qual das k categorias uma entrada pertence [26]. Aqui o algoritmo de aprendizado geralmente é solicitado a produzir uma função $f : \mathcal{X} \rightarrow 1, \dots, k$. Quando $y = f(x)$, o modelo atribui uma entrada descrita pelo vetor x a uma categoria identificada pelo código numérico y . Optamos pela utilização de aprendizado de máquina supervisionado, e vamos apresentar as duas técnicas usadas: o método KNN e as redes neurais do tipo MLP .

2.3.1 KNN

A busca por uma abordagem diferente para tratar o problema nos levou à classificação por *primeiros vizinhos* ou vizinhos mais próximos, chamada de KNN , do inglês *K-nearest neighbours* [25, 45, 46]. O KNN é um método popular para aprendizagem supervisionada [47]. Optamos por este algoritmo pois trabalhamos com durações e classes, sendo a saída do KNN uma associação de classe (ou duração) com relação aos vizinhos do elemento em questão.

Vamos usar duas versões do KNN , para o caso com classes e para o caso com durações. Primeiro vamos descrever como utilizamos aqui este método para o caso com classes: encontramos inicialmente a série para z e com ela conseguimos associar uma classe à região do atrator que estamos considerando, uma vez que a classe é o número de máximos da variável z dentro da n -ésima asa. Depois com a informação sobre os ângulos encontrada, podemos associar um ponto o par $\Phi_i^{(a)}(\varphi_n^{(i,j)}, \varphi_n^{(i,j)}; \Omega_{i+a})$, sendo este então o *rótulo* do ponto e isso será apresentado na Fig. 3.4. Com essas informações podemos construir dois conjuntos de dados: um para referência e outro para teste. O próximo passo foi definir a quantidade de vizinhos escolhidos e o cálculo atribuído à distância desta vizinhança. Aqui optamos por $1 \leq K \leq 5$ e usamos a distância euclidiana e a mínima distância para encontrar os vizinhos mais próximos. A análise feita para obtenção dos melhores resultados com o KNN deve levar em conta a escolha da quantidade de vizinhos e da métrica apropriada, por exemplo, para nosso caso, um dos melhores resultados foi obtido com apenas $K = 1$ vizinho e a distância Euclidiana.

A Figura 2.8 ilustra o método e ela apresenta um ponto em vermelho como elemento a ser classificado. Os demais pontos apresentam cores amarelo, azul e verde que podemos definir as cores como a classe de cada um, este é nosso grupo de referência. Nesse exemplo escolhemos $K = 3$ vizinhos e então a vizinhança do elemento de teste vermelho está definida como aquela dentro do círculo tracejado. Desta forma, a classe atribuída ao elemento será aquela que é a mais comum entre seus 3 vizinhos e nesse exemplo, o elemento terá classe verde. Em algum caso de empate, por exemplo usando um número par de vizinhos, optamos por manter a classe mais baixa, porque para nosso estudo as classes menores tem mais ocorrência, veja a Fig. 3.5.

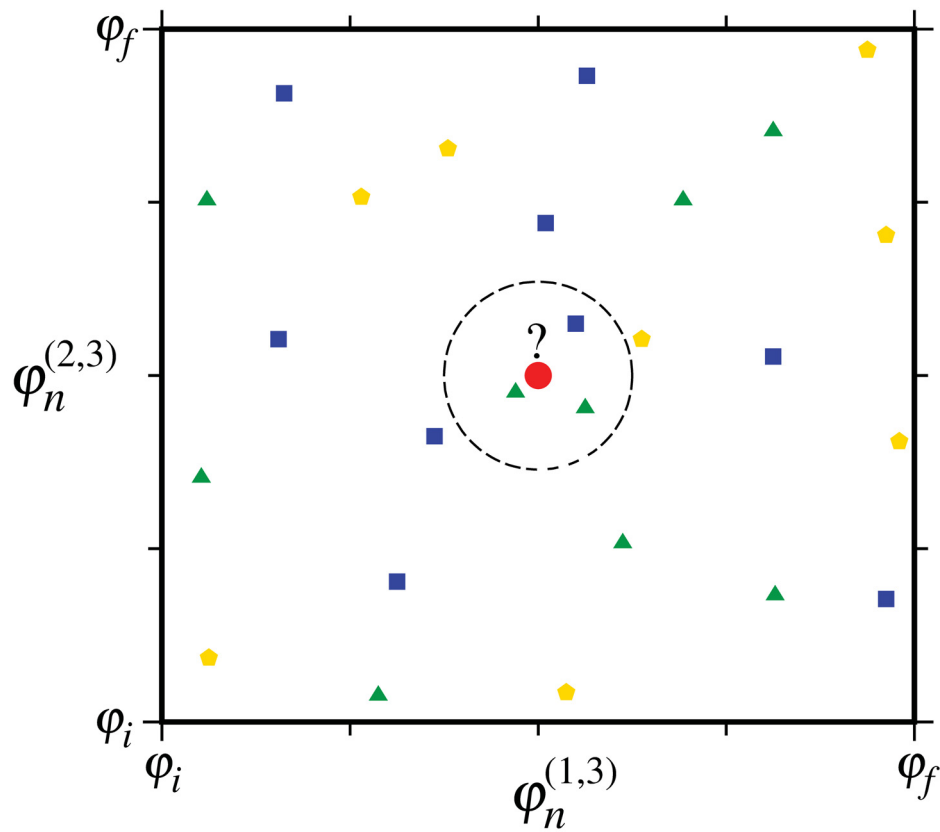


Figura 2.8 – Ilustração para o método KNN . O ponto de interrogação sinaliza o elemento a ser classificado. O círculo tracejado indica o limite da vizinhança do ponto. As cores amarelo, verde e azul representam as classes dos pontos. Para esse exemplo, a classe do ponto é verde.

O método KNN foi aplicado utilizando a distância euclidiana, a mínima distância e ainda, implementamos a probabilidade bayesiana buscando aprimorar o método. Na probabilidade bayesiana precisamos saber alguma informação anterior e a probabilidade dessa, fazendo:

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}, \quad (2.22)$$

onde $P(A|B)$ é a probabilidade do evento A acontecer, $P(B|A)$ é a probabilidade de B acontecer sendo que A já aconteceu, $P(A)$ é a probabilidade de A ocorrer e por fim, $P(B)$ é a probabilidade de B ocorrer.

Comparamos as três versões do KNN estabelecidas aqui na Fig. 3.10. Nessa figura apresentamos resultados para KNN utilizando a distância euclidiana, chamado de $KNNE$,

também para a mínima distância $KNNM$ e para o método com a probabilidade bayesiana, nomeado de $KNNMB$.

Para o método KNN também aplicamos ao caso que trabalhamos com durações, essas foram contadas como o intervalo de cada regime, ou seja, desde o momento que a trajetória entra em uma região até o momento que ela sai. Criamos um rótulo com os ângulos extremos e as durações $\Gamma_i^{(a)}(\varphi_n^{(i,j)}, \varphi_n^{(i,j)}; \Omega_{i+a})$ e novamente obtemos dois conjuntos de dados e precisamos escolher a forma de calcular a distância entre os vizinhos, bem como a quantidade de vizinhos. Nesse caso, a duração do ponto em análise será uma média da duração dos vizinhos escolhidos.

2.3.2 MLP

As redes neurais artificiais são construídas a partir de um conjunto densamente interconectado de unidades simples, onde cada unidade recebe um número de entradas de valor real e produz uma única saída de valor real [44]. Um tipo de rede neural é baseado em uma unidade chamada *perceptron*, que é um sistema nervoso hipotético, ou máquina segundo Rosenblatt [48]. O perceptron é projetado para ilustrar algumas das propriedades fundamentais dos sistemas inteligentes em geral, sem considerar os aspectos físicos e químicos reais. Uma rede neural do tipo *multi-layer perceptron* (MLP), é uma série de modelos de regressão logística empilhados uns sobre os outros, com a camada de saída sendo outra regressão logística ou um modelo de regressão linear, dependendo se estamos resolvendo um problema de classificação ou regressão [42]. Ainda, MLP é uma função matemática que mapeia algum conjunto de valores de entrada para valores de saída. A função é formada pela composição de muitas funções mais simples. Podemos pensar em cada aplicação de uma função matemática diferente como fornecendo uma nova representação da entrada [26], é uma rede neural artificial formada por conexões entre *perceptrons*.

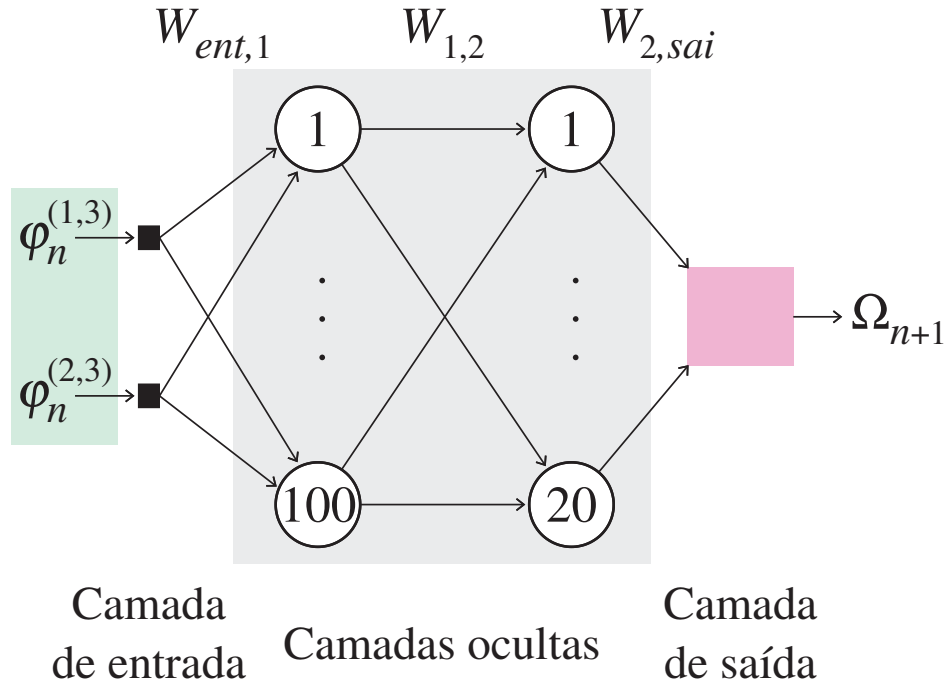


Figura 2.9 – Diagrama esquemático para a MLP usada. Os retângulos pretos representam os neurônios na camada de entrada, os círculos na área cinza representam os neurônios nas camadas ocultas e o quadrado rosa é a camada de saída. As $W_{a,b}$ representam as matrizes de pesos e suas conexões entre os neurônios das diferentes camadas. Do ponto de vista de implementação, a camada de entrada distribui informações dos ângulos extremos $\varphi_n^{i,j}$ para a primeira camada oculta, sem fazer transformações na entrada de informação. O processo de aprendizado de redes neurais artificiais ocorre atualizando os pesos das conexões entre as sucessivas camadas de neurônios. Os pesos são atualizados durante as temporadas de treinamento usando um algoritmo de retropropagação.

Uma ilustração é apresentada na Fig. 2.9 para uma configuração da MLP. Nela mostramos a camada de entrada, as camadas ocultas (no emprego da MLP atribuímos duas camadas ocultas compostas por 100 e 20 neurônios) e a camada de saída. A camada de entrada representada pelos retângulos pretos distribui a informação para as camadas ocultas (aqui utilizamos 100 neurônios), destacadas em cinza, sem sofrer transformações na informação da entrada. A última camada oculta (composta por 20 neurônios) se conecta à camada de saída (rosa). O aprendizado de redes neurais artificiais ocorre atualizando os pesos das conexões entre as sucessivas camadas de neurônios. As matrizes dos pesos dessas conexões estão indicadas na figura por $W_{ent,1}$ sendo da camada de entrada para a primeira camada oculta; $W_{1,2}$ da primeira camada oculta para a segunda camada oculta e $W_{2,sai}$ da segunda camada oculta para a camada de saída. Os pesos são atualizados

durante as temporadas de treinamento a cada n épocas, ou seja, a rede atualiza seus pesos do conjunto de treinamento a cada n repetições.

Definir as configurações da MLP é papel importante pois a escolha, por exemplo, da quantidade de camadas e a da quantidade de neurônios em cada camada interfere nos resultados obtidos. Aqui optamos por uma rede com duas camadas ocultas, com 100 e 20 neurônios do tipo ReLU [49]. Os pesos são atualizados pelo algoritmo de retropropagação (ou *backpropagation*) envolvendo o *gradiente descendente* [50]. Dadas essas configurações, obtemos resultados superiores a 98%, como mostraremos nos capítulos *Resultados*.

2.4 MODELO DE LORENZ GENERALIZADO

Existem modelos generalizados de Lorenz [23], por exemplo aqueles apresentados por Roy [57, 58] e Shein [59, 60]. Aqui estudamos os sistemas chamados de *proto-Lorenz* que foram apresentados por Miranda e Stone [24] em 1993 e são modelos generalizados do sistema de Lorenz. Como já apresentado antes, o sistema de Lorenz possui um atrator caótico para os parâmetros $\sigma = 10,0$, $r = 28,0$ e $b = 8/3$ com duas asas, mostrado na Fig. 2.4. No artigo de Miranda e Stone vemos atratores com outros números de asas. O sistema 2.23 com quatro asas possui três variáveis (s, t, z) , com as correspondentes derivadas no tempo $(\dot{s}, \dot{t}, \dot{z})$, descrito como

$$\begin{aligned}\dot{s} &= \frac{1}{2(s^2 + t^2)} \left[-\sigma s^3 + (2\sigma + r - z)s^2t + (\sigma - 2)st^2 - (r - z)t^3 \right], \\ \dot{t} &= \frac{1}{2(s^2 + t^2)} \left[(r - z)s^3 + (\sigma - 2)s^2t + (-2\sigma - r + z)st^2 - \sigma t^3 \right], \\ \dot{z} &= 2s^3t - 2st^3 - bz.\end{aligned}\tag{2.23}$$

Esse sistema 2.23 possui os mesmos três parâmetros do sistema 2.11 (σ, r, b) e fixando seus valores em $\sigma = 10,0$, $r = 28,0$ e $b = 8/3$, obtemos a Fig. 2.10.

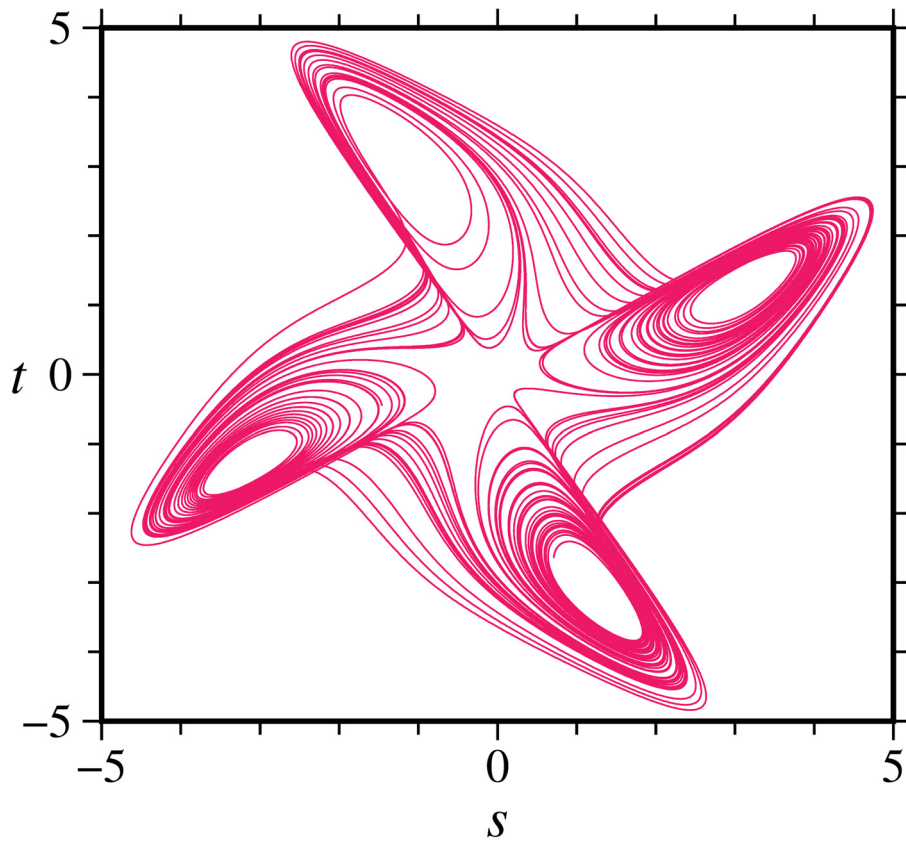


Figura 2.10 – Atrator caótico do sistema de Lorenz 2.23.

Já para o sistema com três asas, temos as variáveis (p, q, z) com os mesmos três parâmetros do Lorenz e esse modelo de três asas é escrito como

$$\begin{aligned}\dot{p} &= \frac{1}{3} [-(\sigma + 1)p + (\sigma - r + z)q] + \frac{1}{3M} [-(1 - \sigma)(p^2 - q^2) + 2(r + \sigma - r - z)pq], \\ \dot{q} &= \frac{1}{3} [(r - \sigma - z)p - (\sigma + 1)q] + \frac{1}{3M} [2(\sigma - 1)pq + (r + \sigma - z)(p^2 - q^2)], \quad (2.24) \\ \dot{z} &= \frac{1}{2} (3p^2q - q^3) - bz, \\ M &= \sqrt{p^2 + q^2}.\end{aligned}$$

Tanto o sistema 2.11 quanto os sistemas 2.24 e 2.23 têm expoente de Lyapunov $\lambda = 1,306$. Segundo Miranda e Stone os sistema de Lorenz e proto-Lorenz 2.24 são relacionados

pela transformação

$$u = x^2 - y^2,$$

$$v = 2xy,$$

$$z = z,$$

onde temos

$$u = p^3 - 3pq^2,$$

$$v = 3p^2q - q^3,$$

$$z = z.$$

No atrator de Lorenz da Fig. 2.4 vemos duas asas somente, já no sistema 2.24 temos 3 asas como vemos na Fig. 2.11. Utilizamos os mesmos valores de parâmetros para ambos os sistemas.

A predição da duração em uma asa e também a troca de asa é interesse neste estudo, logo a opção por dividir o atrator em regiões é um passo importante para os passos realizados na sequência. Ainda, precisamos ressaltar que devemos escolher apropriadamente as divisões a fim de contemplar a trajetória do atrator dentro da asa, por exemplo, para a Fig. 2.4 poderíamos escolher, como já mencionado, dividir o atrator para x positivo e negativo, tendo assim duas regiões cada uma delas contendo uma das asas. Para o atrator da Fig. 2.11 optamos por dividi-lo em três regiões.

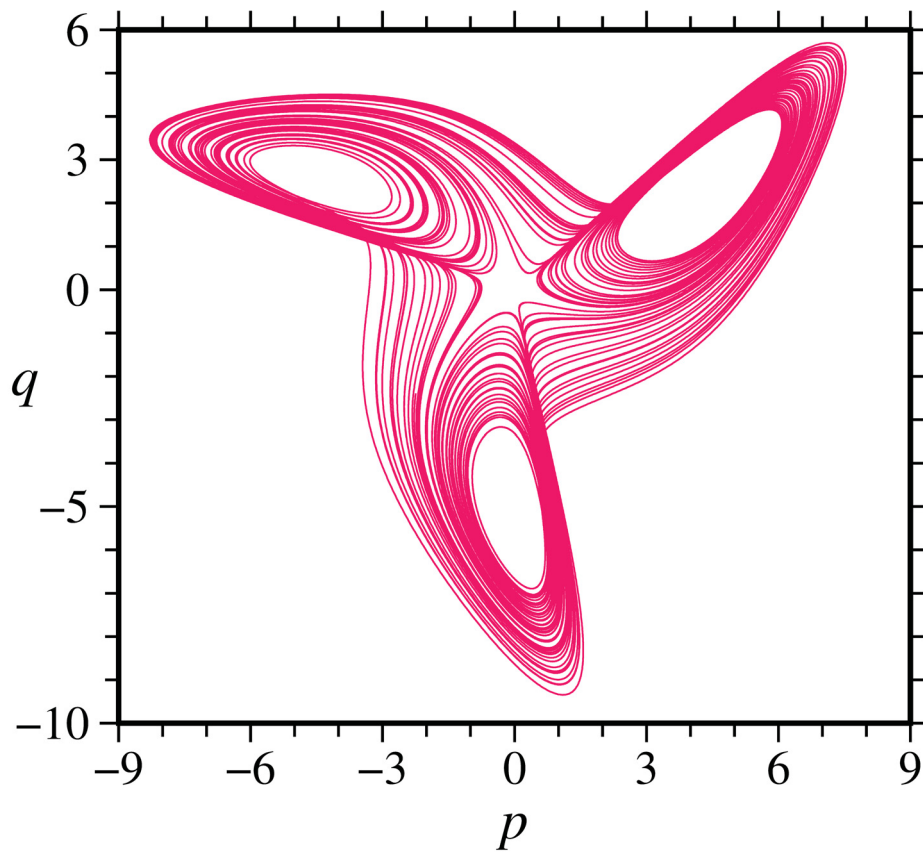


Figura 2.11 – Atrator caótico do sistema 2.24. Os parâmetros foram mantidos em $\sigma = 10,0$, $r = 28,0$ e $b = 8/3$.

Na Figura 2.12 temos novamente o atrator caótico para o sistema 2.24, mas com informações adicionais. A primeira informação é que dividimos o atrator em regiões com 120° gerando assim a região A em rosa, a B em verde e a C em azul, como identificado pela paleta de cores na figura. Nomeamos as regiões dessa forma pois a direção do movimento segue a ordem $A \rightarrow B \rightarrow C \rightarrow A \dots$. Adicionalmente incluímos os pontos fixos instáveis de cada asa, os quais destacamos em azul. Selecionamos um trecho da trajetória na região A e colorimos para indicar o movimento. Ela começa na parte mais interna da asa, a qual escolhemos a cor verde e localiza-se próxima ao ponto fixo, depois move-se para a parte mais externa da asa (também chamaremos de borda) que indicamos pela linha amarela para vermelha até por fim mudar de asa, sendo esse trecho indicado pela linha tracejada em vermelho com a seta.

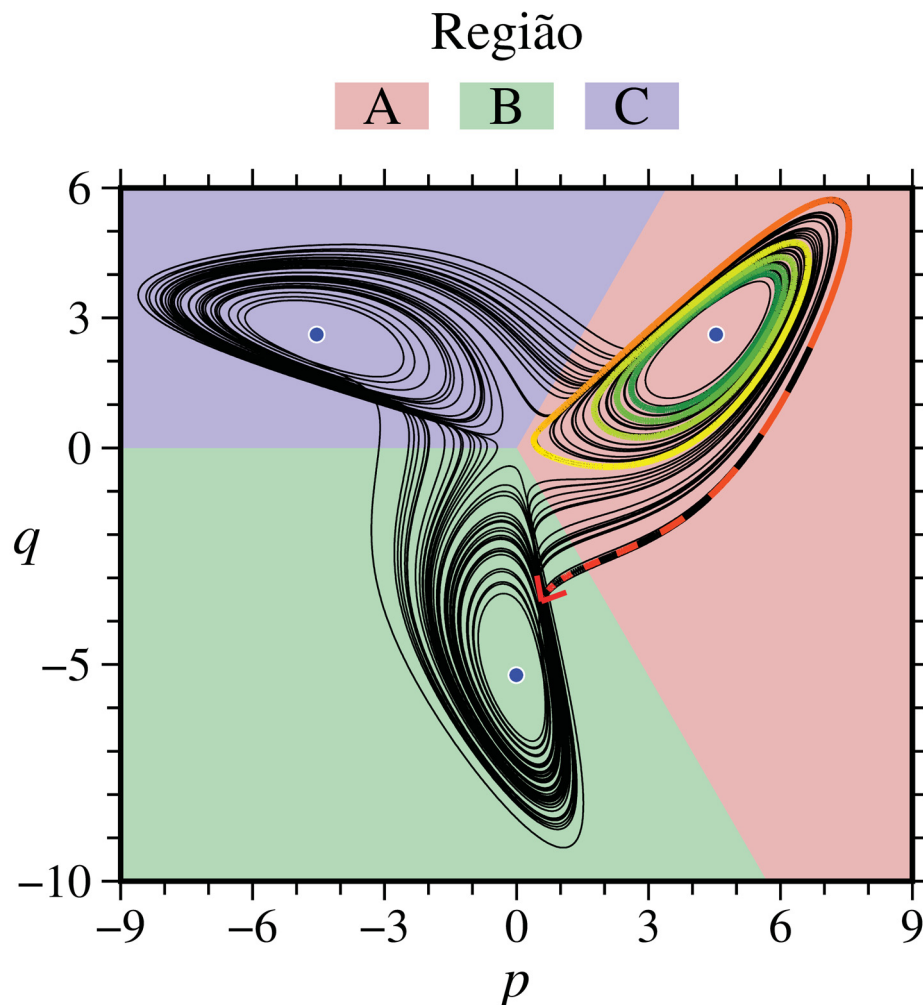


Figura 2.12 – Atrator ilustrativo projetado no plano $q \times p$ com 81600 pontos e cores delimitando as asas localizadas nas regiões A, B e C. A direção do movimento segue a ordem $A \rightarrow B \rightarrow C \rightarrow A \dots$. Círculos azuis são as localizações dos pontos fixos instáveis relacionados a cada asa. A trajetória caótica começa na parte mais interna da asa, próxima ao ponto fixo (exemplificada pela linha verde dentro da região A do anel), move-se para a parte mais externa de cada asa (linha amarela para vermelha), até mudar de asa (linha tracejada em vermelho com a seta).

Essa divisão em regiões também foi feita para o caso com quatro asas e vamos apresentá-lo mais a frente. Para o caso com quatro asas, dividimos o espaço de fases em ângulos de 90° e assim também obtemos regiões onde encontramos somente uma asa dentro.

2.5 UTILIZANDO OS CONCEITOS

Nosso objetivo é a previsão da duração de regimes utilizando a informação dos ângulos entre os VCLs. Uma vez apresentados os conceitos, vamos agora apresentar a maneira como foram utilizados aqui para obtenção dos resultados apresentados nos próximos capítulos.

O modelo generalizado com três asas foi inicialmente integrado e assim obtivemos as séries de p, q e z . Paralelamente, usando o método descrito para cálculo dos VCLs, obtemos os ângulos entre eles. Associamos o símbolo $\theta_n^{i,j}$ para os ângulos, onde i e j são as variedades e identificamos a variedade instável por 1, a central tem índice 2 e a estável numeramos 3. Com a série para z definimos as classes Ω_{n+a} sendo essa o número de máximos em z dentro de uma região n . Ainda, olhamos para os ângulos também dentro de cada região e extraímos os valores extremos de cada uma delas e identificamos esse ângulos extremos por $\varphi_n^{i,j}$. Com uma combinação de ângulos, conseguimos olhar para resultados onde tomamos valores dos ângulos em uma região n e conseguimos as classes das regiões $n + 1$. Esse $+1$ nomeamos de *avanço* e ele pode ter outros valores, de forma que esse avanço identifica o regime ao qual estamos falando. A informação dos ângulos e as classes nos mostraram que podemos encontrar a duração de regimes a frente usando informações do regime atual.

Dadas as informações de ângulos e classes podemos criar um rótulo para cada ponto nomeado $\Phi_i^{(a)}(\varphi_n^{(i,j)}, \varphi_n^{(i,j)}; \Omega_{i+a})$ para aplicarmos as técnicas de aprendizado de máquina. Note que para esses processos não vamos mais integrar o sistema, vamos utilizar o rótulo gerado como nosso dado principal. Assim, obtemos dois conjuntos de dados, sendo um para teste e outro para treino ou ainda podemos chamar de conjunto referência para o caso do KNN . Com os resultados obtidos, comparamos os métodos para avaliar onde cada um foi melhor.

O procedimento adotado para obter as séries, agora para s, t e z , os ângulos e ângulos extremos para o modelo com quatro asas foi o mesmo feito para o de três asas. A diferença é que para o modelo com quatro asas, decidimos trabalhar diretamente com as durações e não com as classes. Observamos novamente a relação entre uma combinação de ângulos e a possibilidade de prever a duração de regimes a frente com dados do regime atual.

Para esse caso temos o rótulo $\Gamma_i^{(a)}(\varphi_n^{(i,j)}, \varphi_n^{(i,j)}; \Delta t_{i+a}(t_L))$ e aplicamos somente o método do KNN . Esses resultados obtidos são apresentados após o caso para três asas.

Capítulo 3

RESULTADOS - 3 ASAS

Aqui expomos os resultados obtidos para o sistema proto-Lorenz 2.24. Com os parâmetros fixos $(b, r, \sigma) = (8/3; 28; 10)$, usamos o método Runge-Kutta de quarta ordem [41] com passo fixo igual a 10^{-3} para integrar o sistema e um transiente de 10^6 . Para as condições iniciais usamos aleatoriamente valores dentro do intervalo $0 < p_0, q_0, z_0 < 1$. Obtivemos o maior expoente de Lyapunov $\lambda \approx 1,306$ ao longo de 10^7 pontos do atrator com o algoritmo de Wolf [38]. Nosso objetivo é prever a troca de região (asa) e a sua duração, assim começamos apresentando o alinhamento entre os vetores covariantes de Lyapunov, a definição das classes e por sequência os métodos aplicados.

3.1 ÂNGULOS ENTRE OS VCLs

A Figura 3.1 apresenta o mesmo atrator caótico, uma vez que temos o maior expoente de Lyapunov $\lambda \approx 1,306$, da Fig. 2.12 mas com cores indicando o valor do ângulo entre as variedades instável e central em 3.1 (a), instável e estável em 3.1 (b) e em 3.1 (c) entre as variedades central e estável. Para $\theta^{(i,j)} \approx 0$ temos a cor azul escura, para $\theta^{(i,j)} \approx \pi/4$ a cor preta e para $\theta^{(i,j)} \approx \pi/2$ a cor magenta. Como estamos interessados no alinhamento, estabelecemos para $\theta^{(i,j)} > \pi/2$ a transformação $\theta^{(i,j)} \mapsto \pi - \theta^{(i,j)}$.

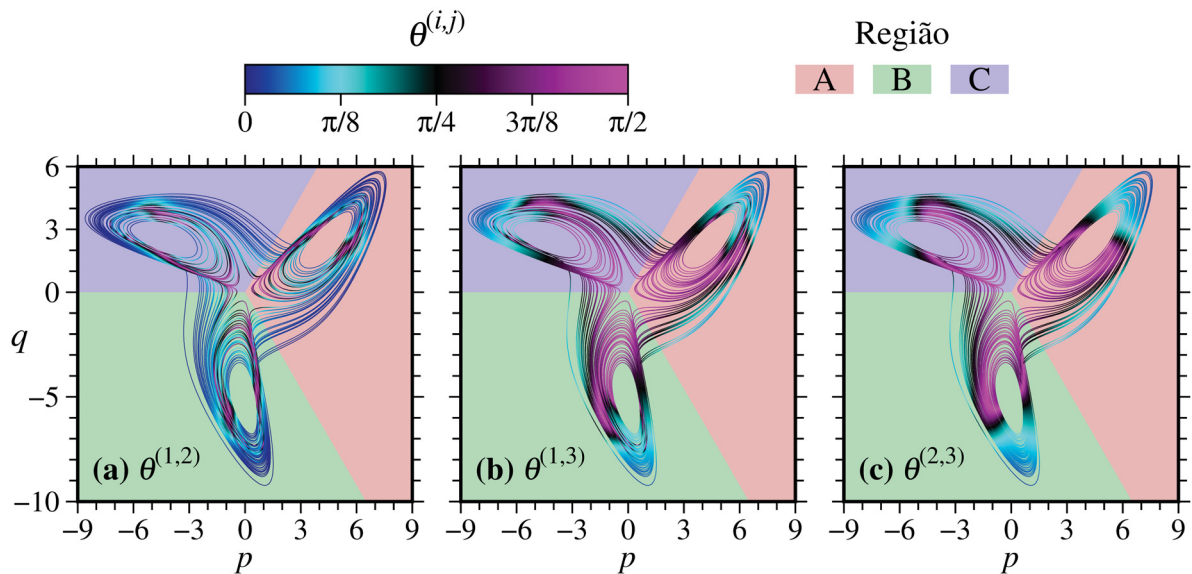


Figura 3.1 – O mesmo atrator da Fig. 2.12 mas com 10^5 pontos e cores adicionais (veja a paleta de cores) indicando os ângulos entre as variedades (a) instável e central, (b) instável e estável e (c) central e estável no espaço tangente. Os ângulos $\theta^{(2,3)}$ em (c) e $\theta^{(1,3)}$ em (b) tendem a $\pi/2$ próximos da origem $(0,0)$, e a 0 nas partes mais externas (pontas das asas), antes de mudarem de asa. Isso ocorre para as três asas. O alinhamento da variedade instável e central ao longo da variedade estável precede uma mudança de asa como mostra a cor azul clara nas partes mais externas das asas em (b) e (c). Note que tal alinhamento precedendo mudanças de asas são mais fortes em (a), reconhecidos pela cor azul escuro.

Notamos que independe da região que observamos, o ângulo $\theta^{(i,j)}$ se comporta de maneira similar. Por exemplo, olhando para a Fig. 3.1(b) na região B (verde) temos o ângulo $\theta^{(1,3)}$ tendendo a $\pi/2$ próximo da origem $(0,0)$, na parte intermediária da asa o ângulo é próximo de $\pi/4$ e nas bordas (parte próxima ao eixo p) temos valores entre $\pi/8 < \theta^{(1,3)} < 0$. Como apresentado na Fig. 2.12, o movimento de saída da asa ocorre pela borda externa e, acompanhando nas Figs. 3.1(b) e (c) tal movimento, vemos que o alinhamento (em azul claro) precede uma troca de região. Isso é mais claramente identificado na Fig. 3.2. Na Fig. 3.2(a) temos as projeções (p,q) com as cores de fundo indicando as regiões da Fig. 3.1. Já em 3.2(b) z é plotada, essa variável definirá a *classe* Ω_n , uma vez que a classe é o *número de máximos da variável z dentro da n -ésima asa*. No exemplo da Fig. 3.2(b) a classe é $\Omega_n = 4$, que é o número de máximos em z dentro da região A (rosa).

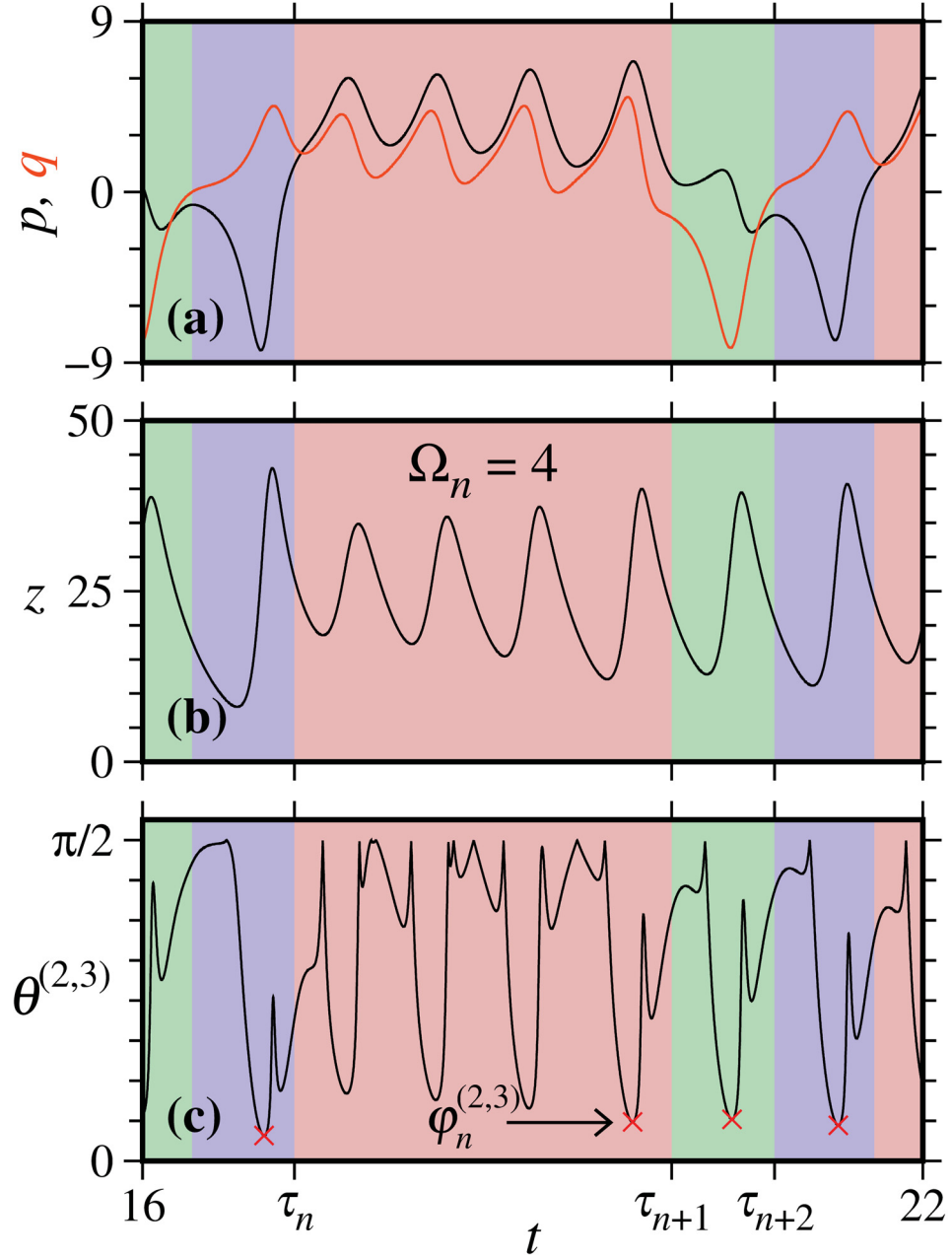


Figura 3.2 – Janela temporal mostrando (a) (p, q) , (b) variável z e (c) o ângulo $\theta^{(2,3)}$. As cores de fundo identificam as regiões A, B e C da Fig. 2.12. Como um exemplo, em (b) nós mostramos as classes $\Omega_n = 4$ (máximo de z) dentro da região A. O menor valor de $\theta^{(2,3)} (= \varphi_n^{(2,3)})$ dentro de qualquer região ocorre antes da troca de região (veja os destaques em vermelho). Consequentemente, coletando todos os destaques em vermelho ao longo da série temporal, dados de $\varphi_n^{(2,3)}$ (o alinhamento mais forte de $\theta^{(2,3)}$) contém informação relevante quando a troca de região ocorre.

A Figura 3.2(c) mostra o ângulo $\theta^{(2,3)}$. Destacamos em vermelho os menores valores de $\theta^{(2,3)}$ em cada região e observamos que esse ocorre antes da troca de asa, a esses

identificaremos como $\varphi^{(2,3)}$. Definimos então $\varphi^{(i,j)}$ como

$$\varphi^{(i,j)} = \min[\theta^{(i,j)}(t), \tau_{n'} \leq t < \tau_{n+1}], \quad (3.1)$$

sendo $\tau_{n'}$ o tempo em que a n -ésima troca de asa ocorre e $\tau_{n'} = \tau_n$ para $\Omega_n = 1$. Ainda, τ_{n+1} é o tempo do último mínimo de z para $\Omega_n > 1$. Adotamos o alinhamento somente depois do último mínimo de z para considerarmos só o alinhamento extremo perto da troca de asa. Dessa forma, $\varphi^{(2,3)}$ é o menor valor de $\theta^{(2,3)}$ dentro de uma região durante a última volta na asa e o alinhamento mais forte (mais próximo de 0) de $\theta^{(2,3)}$ ocorre logo antes da troca de asa.

A Figura 3.3 apresenta a duração $\delta t_{n+1} = \tau_{n+2} - \tau_{n+1}$ da asa $n + 1$ como função do alinhamento extremo $\varphi_n^{(i,j)}$. Em Fig. 3.3(a) o ângulo $\varphi_n^{(1,2)}$, em (b) $\varphi_n^{(1,3)}$ e em (c) $\varphi_n^{(2,3)}$, as cores indicam os valores das classes $\Omega_n = 1, 2, 3, 4, 5$ como indicado da paleta. Na Fig. 3.3(a) há mais valores de δt_{n+1} relacionados ao ângulo extremo $\varphi_n^{(1,2)}$ do que aos ângulos extremos $\varphi_n^{(1,3)}$ e $\varphi_n^{(2,3)}$ nas Figs. 3.3(b) e (c), respectivamente. Para predição buscamos valores mais precisos de δt_{n+1} , assim resultados como o da Fig. 3.3(a) não são desejados, mas os das Figs. 3.3(b) e (c) já nos mostram que o número permitido de ângulos extremos ($\varphi_n^{(1,3)}$ e $\varphi_n^{(2,3)}$) para um dado δt_{n+1} é menor, sendo assim mais preciso que para $\varphi_n^{(1,2)}$. A duração é fortemente relacionada às classes, assim utilizamos como alternativa prever a classe Ω_n ao invés da duração δt_n , pois o número de voltas dado em uma asa antes de trocar a região e a classe atribuída são as mesmas. Para as médias da duração, usando regressão linear a partir dos dados, temos a relação

$$\bar{\Delta}t_n = 0,638\Omega_n + 0,388, \quad (3.2)$$

como uma boa aproximação. Destarte usaremos somente Ω_n a partir daqui, mantendo em mente a relação desse com a duração.

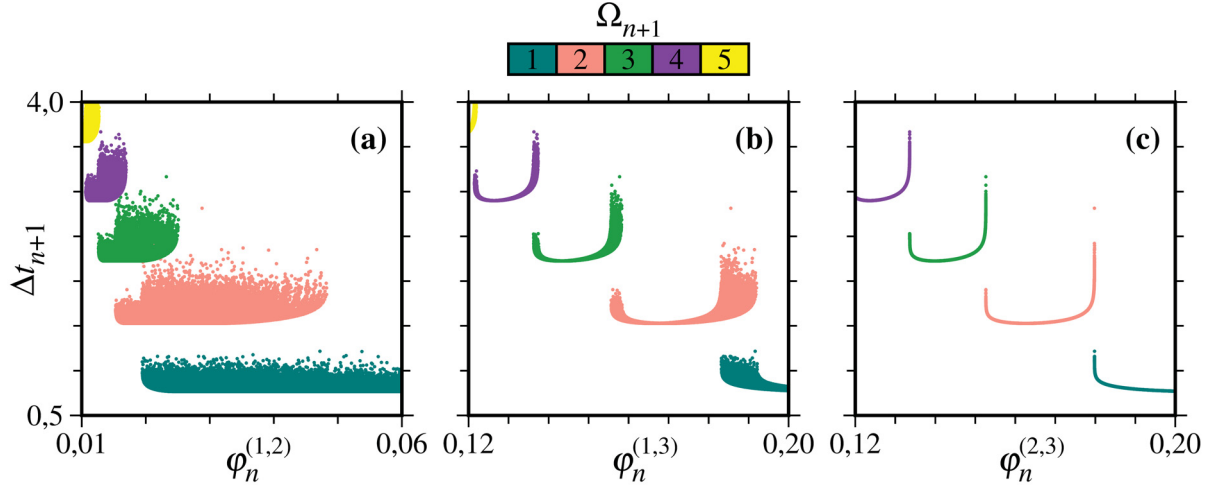


Figura 3.3 – Duração Δt_{n+1} do regime $n + 1$ como função do alinhamento extremo (a) $\varphi_n^{(1,2)}$; (b) $\varphi_n^{(1,3)}$; e (c) $\varphi_n^{(2,3)}$ ao longo do regime n . As cores são as classes da asa $n + 1$. Resultados em (a) demonstram claramente a projeção ao longo do eixo de ângulo são únicas, fazendo previsões de Δt_{n+1} , para um dado $\varphi_n^{(1,2)}$, muito impreciso. Tais imprecisões tendem a desaparecer para os ângulos $\varphi_n^{(1,3)}$ em (b), e $\varphi_n^{(2,3)}$ em (c).

3.2 CLASSES E PREDIÇÃO

A Figura 3.4 exibe as classes Ω_{n+1} como função dos ângulos $\varphi_n^{(1,3)}$ e $\varphi_n^{(2,3)}$. Usaremos agora o termo Ω_{n+a} , onde a é a localização da asa em questão, por exemplo se estivermos olhando para os ângulos na região A , Ω_{n+1} é a classe da asa em B , já se tivéssemos Ω_{n+2} , estaríamos falando da asa na região C e assim por diante. A paleta de cores identifica as classes na figura. Para classes $\Omega_{n+1} \geq 20$ atribuímos a cor preta. Na Fig. 3.4 vemos uma divisão clara entre as classes e também que o número da classe aumenta conforme os ângulos $\varphi_n^{(1,3)}$ e $\varphi_n^{(2,3)}$ se aproximam de 0, sendo isso melhor observado na ampliação feita dentro da figura. Dela podemos dizer que menores valores de $\varphi_n^{(1,3)}$ e $\varphi_n^{(2,3)}$ podem prever classes de maiores valores, o que leva a relação do forte alinhamento das variedades instável e central ao longo da direção estável.

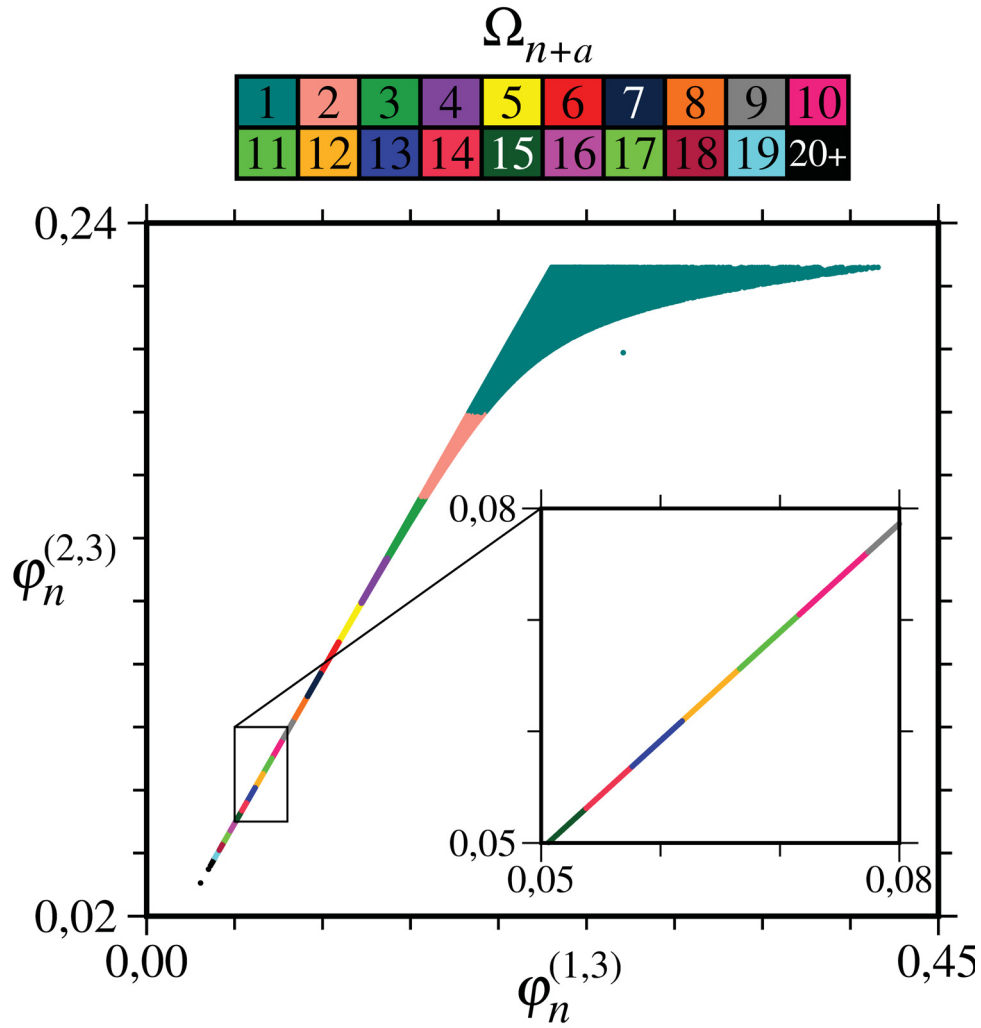


Figura 3.4 – Classes Ω_{n+1} (veja a legenda de cores) como função dos ângulos $\varphi_n^{(1,3)}$ e $\varphi_n^{(2,3)}$. São 573111 pontos obtidos de 100 séries temporais. As classes são bem particionadas em regiões, o quadro interno mostra o padrão crescente das classes conforme o alinhamento aumenta. Em outras palavras, como os ângulos $\varphi_n^{(1,3)}$ e $\varphi_n^{(2,3)}$ diminuem, as classes aumentam, significando que maiores durações Δt_{n+1} são esperadas.

Classes maiores ocupam regiões menores na Fig. 3.4 e na Fig. 3.5 distinguimos a ocorrência de cada classe, tal qual decai exponencialmente como mostrado pela reta lilás $f(x)$. Os pontos estão coloridos com a cor respectiva de cada classe conforme a paleta de cores da Fig. 3.4.

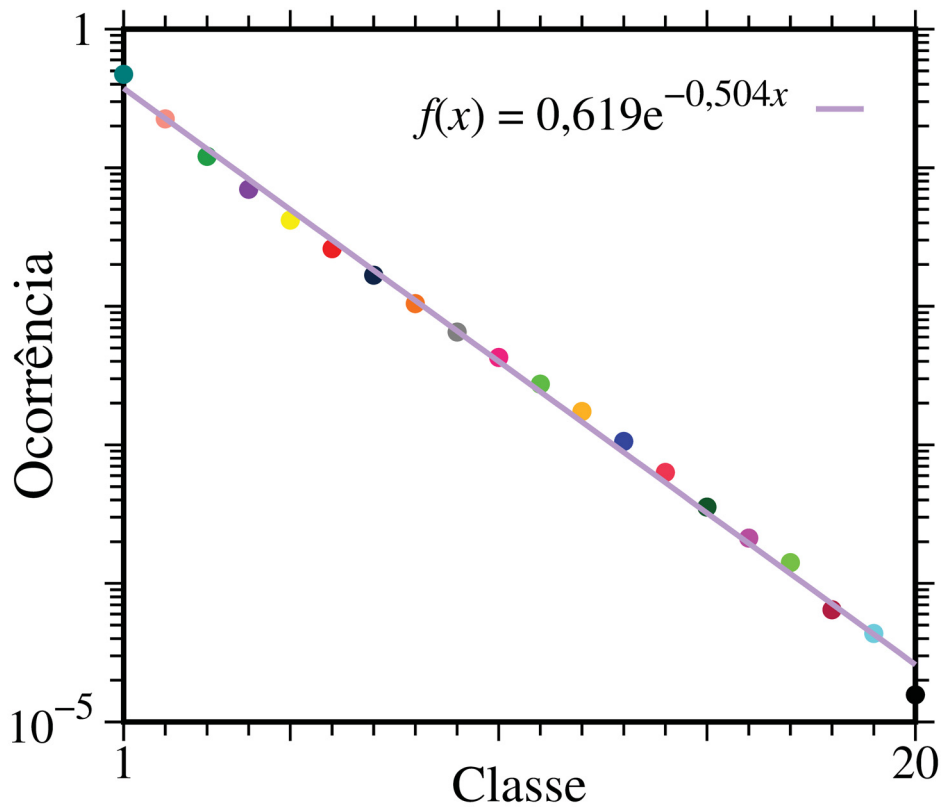


Figura 3.5 – Ocorrência de cada classes (pontos coloridos conforme legenda de cores da Fig. 3.4.) A ocorrência decai exponencialmente, como indicado pela linha lilás. Destacamos que a classe 20 inclui casos com 20 ou mais voltas em torno da mesma asa.

A classe Ω_{n+1} presente na Fig. 3.4 refere-se a classe da asa seguinte, pois $a = 1$ como já mencionado antes. Para melhor compreensão, podemos pensar no a como o *avanço* feito. Por exemplo, se tomarmos o alinhamento de $\varphi_n^{(1,3)}$ na região A , esse pode prever a duração (relacionada a classe) de B . Igualmente teremos para o alinhamento de $\varphi_n^{(1,3)}$ na região B prevendo a C . Desta forma poderíamos pensar que o alinhamento de $\varphi_n^{(1,3)}$ na região A resultaria na previsão da classe (duração) da asa C , ou seja, a classe Ω_{n+2} , onde $a = 2$. Ainda, ampliando a ideia, o alinhamento de $\varphi_n^{(1,3)}$ em A conseguiria prever a classe Ω_{n+3} que olhando para a Fig. 3.1(b) é como se déssemos uma volta no atrator. O pensamento segue para classes Ω_{n+4} , Ω_{n+5} e por diante.

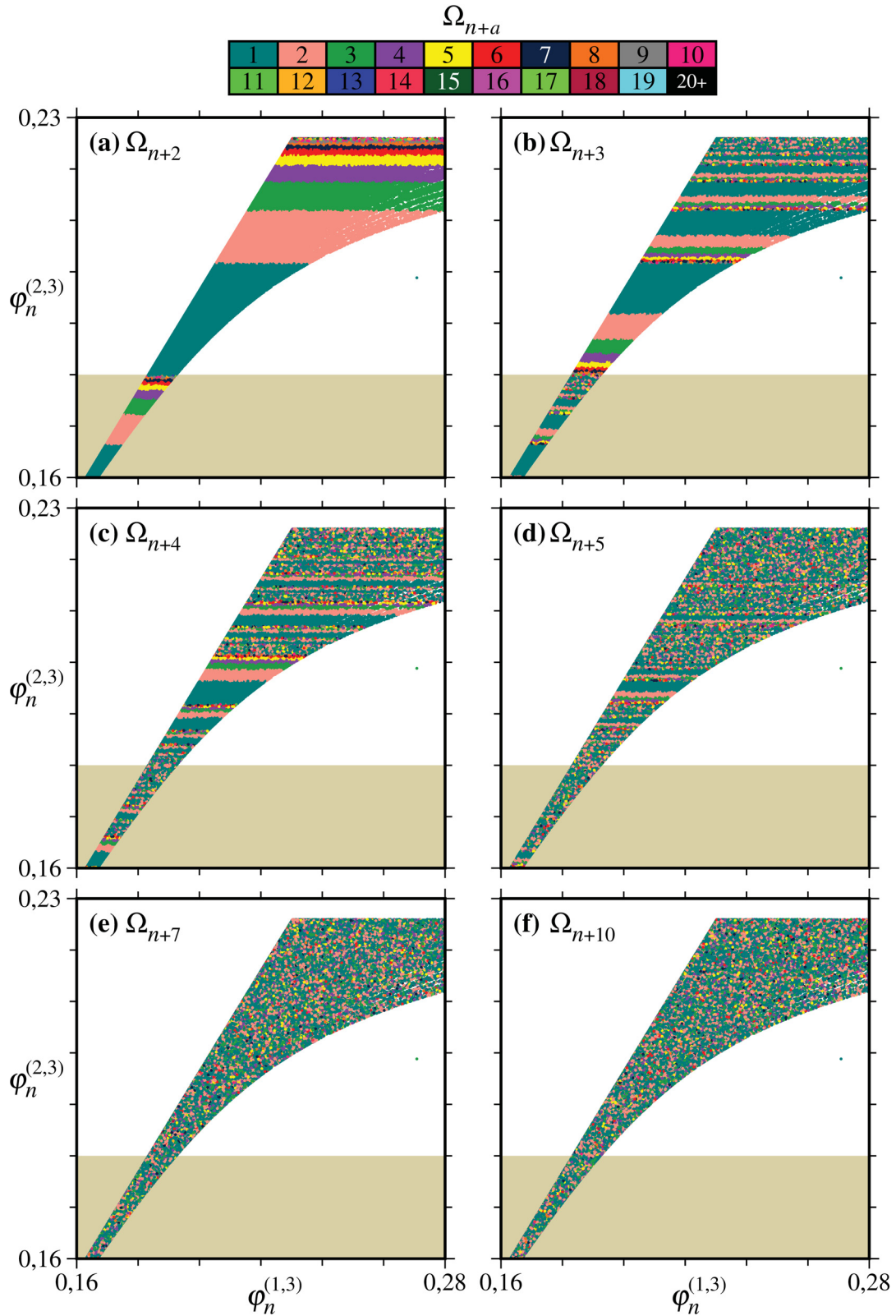


Figura 3.6 – Classes (a) Ω_{n+2} , (b) Ω_{n+3} , (c) Ω_{n+4} , (d) Ω_{n+5} , (e) Ω_{n+7} e (f) Ω_{n+10} são mostradas como função dos ângulos $\varphi_n^{(1,3)}$ e $\varphi_n^{(2,3)}$. As cores das classes seguem a legenda. O fundo branco é a região da classe $\Omega_{n+1} = 1$ da Fig. 3.4 e a cor de fundo bege é a região de classe $\Omega_{n+1} = 2$ da Fig. 3.4. Em (a) nós observamos que dentro da classe $\Omega_{n+1} = 1$ e $\Omega_{n+1} = 2$ da Fig. 3.4, novas sequências de altas-ordens de classes surgem. Esse surgimento de novas classes ocorre mais e mais, visivelmente em (b) e (c), e as cores se tornam mais e mais misturadas em (d)-(f), tornando a tarefa de previsão quase impossível para tempos longos.

A fim de investigarmos a previsão das avanços à frente $a > 1$, temos a Fig. 3.6, a qual mostra as classes (a) Ω_{n+2} , (b) Ω_{n+3} , (c) Ω_{n+4} , (d) Ω_{n+5} , (e) Ω_{n+7} e (f) Ω_{n+10} como função dos ângulos $\varphi_n^{(1,3)}$ e $\varphi_n^{(2,3)}$. O branco no fundo da imagem indica a região de classe $\Omega_{n+1} = 1$ da Fig. 3.4 e o fundo bege a região $\Omega_{n+1} = 2$ da mesma figura. A Fig. 3.6 é como uma ampliação da Fig. 3.4 para um intervalo menor de $\varphi_n^{(1,3)}$ e $\varphi_n^{(2,3)}$ (contemplando apenas as classes $\Omega_{n+1} = 1$ e 2) mas para outros avanços a (asas seguintes).

Interessante notar que diferentemente da Fig. 3.4, onde as classes aumentam conforme os ângulos se aproximam de 0, aqui na Fig. 3.6(a) as classes diminuem à medida que os ângulos tendem a 0 dentro das duas regiões de fundo (cores branca e bege). Diferentemente, na Fig. 3.6(b) ocorre que dentro de uma região, por exemplo de fundo branco, se olharmos para a região na qual tínhamos $\Omega_{n+2} = 1$ na Fig. 3.6(a) vemos que os valores de Ω_{n+3} diminuem para ângulo tendendo a 0. Isso muda para cada um dos avanços apresentados na Fig. 3.6. Percebemos que as cores se misturam segundo o avanço aumenta e a precisão na previsão é então perdida.

3.3 MLP e KNN

As classes (cores) são facilmente identificadas no plano $\varphi_n^{(2,3)} \times \varphi_n^{(1,3)}$ da Fig. 3.4 pois temos uma divisão clara das cores e então identificamos valores para os pares $(\varphi_n^{(1,3)}, \varphi_n^{(2,3)})$, atribuindo uma cor a cada par. Como nosso objetivo é prever a troca de asa e a duração, podemos utilizar a classificação aplicando técnicas de *machine learning* e *método de primeiros vizinhos*. Para isso, a partir dos 573111 pontos do conjunto de dados, geramos 2 novos conjuntos com 250000 pontos cada, sendo um para o treinamento e outro para teste. Nos dois conjuntos mantivemos a mesma distribuição exponencial de classe mostrada na Fig. 3.5. Para MLP, o estágio de treinamento consiste em 200 épocas, ou seja a rede atualiza os seus pesos do conjunto de treinamento a cada 200 repetições. Desta forma, conseguimos uma acurácia de 0,9821 na predição.

A Tabela 3.1 mostra a acurácia da predição em cada classe. Destacamos que existem classes com 100% de acurácia. Na tabela temos as colunas -1 que são desvio da precisão para uma classe abaixo e as colunas $+1$ são para os desvios de uma classe acima. Isto significa que para a classe 8, por exemplo, nós obtemos acurácia de 0,9852 e na coluna -1

temos 0,0148, informando assim que 0,0148 deveria ser de classe 8 mas foi reconhecida como 7.

Tabela 3.1 – Acurácia da predição (AP) em cada classe da MLP. As colunas -1 são os desvios na precisão para uma classe abaixo e as colunas $+1$ são os desvios na precisão para uma classe acima.

Classe	AP	- 1	+ 1	Classe	AP	- 1	+ 1
1	0,9983	-	0,0017	11	0,9642	0,0089	0,0268
2	0,9595	0,0291	0,0114	12	0,9823	0,0000	0,0177
3	0,9526	0,0017	0,0458	13	0,9884	0,0000	0,0116
4	1,0000	0,0000	0,0000	14	0,9620	0,0000	0,0380
5	0,9641	0,0253	0,0106	15	1,0000	0,0000	0,0000
6	1,0000	0,0000	0,0000	16	1,0000	0,0000	0,0000
7	0,9875	0,0125	0,0000	17	0,9318	0,0000	0,0682
8	0,9852	0,0148	0,0000	18	1,0000	0,0000	0,0000
9	0,9969	0,0031	0,0000	19	1,0000	0,0000	0,0000
10	0,9715	0,0285	0,0000	20	1,0000	0,0000	-

Para compararmos este resultado com o uso de outro procedimento, escolhemos o método de primeiros vizinhos KNN , do inglês *K nearest neighbours* (K vizinhos mais próximos). Esse método usa um conjunto de pontos de referência previamente rotulados e atribui rótulos aos pontos de teste de acordo com os valores dos K vizinhos mais próximos. Dado o estudo envolvendo classes, adotamos o KNN como um algoritmo classificador. O conjunto de referência é formado por pontos rotulados como $\Phi_i^{(a)}(\varphi_n^{(1,3)}, \varphi_n^{(2,3)}; \Omega_{i+a})$ e esse é o mesmo conjunto com 250000 pontos usados no treinamento da MLP. A classe $\Omega_i^{(a)}$ atribuída por este algoritmo a um ponto de teste $(\varphi_n^{(1,3)}, \varphi_n^{(2,3)})$ é dada pela classe mais frequente observada nos K pontos de referência mais próximos do ponto de teste. No caso de um empate, atribuímos a classe mais baixa. Para calcular as distâncias entre os pontos de referência e de teste, diferentes métricas podem ser adotadas, incluindo medidas que não são métricas.

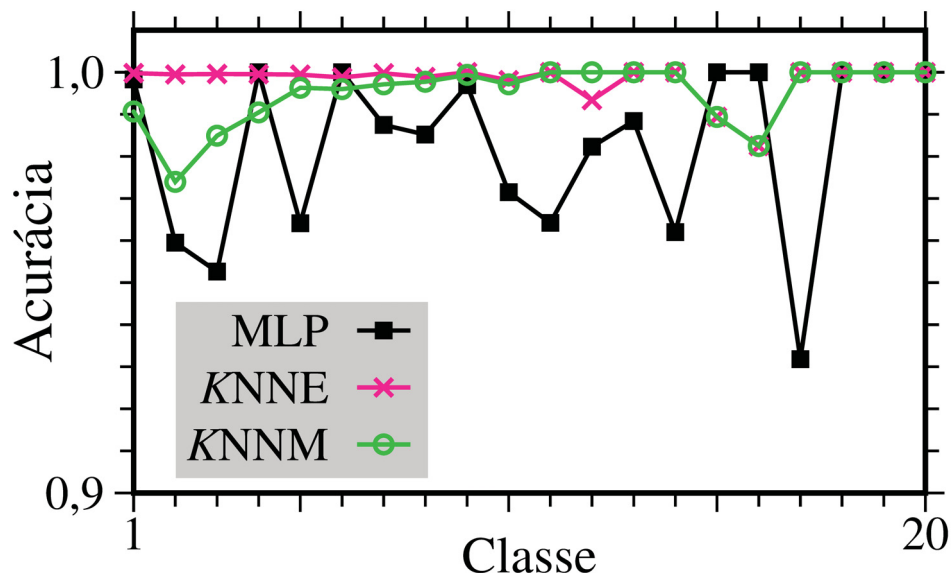


Figura 3.7 – Comparação entre os resultados para MLP e método KNN para avanços $a = 1$, detalhados por classes. KNN com vizinhança $K = 1$. Em geral o método MLP (quadrados pretos) mostra uma menor acurácia comparada aos dois procedimentos KNN; A métrica euclidiana (KNNE em rosa) dá melhores resultados para classes $\Omega_{n+1} \leq 9$ do que a mínima distância (KNNM em verde). Das classes $\Omega_{n+1} \geq 13$ ambas aplicações do KNN apresentam acurácias muito próximas. Classes 15 e 16 são altamente previsíveis pela MLP, também as classes $\Omega_{n+1} \geq 18$ tem resultado similar para o método KNN.

A Figura 3.7 mostra a comparação entre a acurácia da predição feita pela MLP e pelo método KNN. A acurácia é calculada olhando para os valores das classes obtidas e para os valores das classes do conjunto de teste. Como usamos a distância entre vizinhos no KNN precisamos de algo que calcule essa distância para determinar aqueles mais próximos, assim optamos pela *métrica Euclidiana* no plano, a qual chamaremos de KNNE. Também usaremos a *mínima distância* em uma das duas direções no plano e a sigla dessa será KNNM. A mínima distância não consiste de uma métrica e pode ser escrita como

$$d(x,y) = \min(|x_1 - y_1|, |x_2 - y_2|). \quad (3.3)$$

Para Figura 3.7 selecionamos o conjunto de treinamento da MLP como os dados de referência para a classificação do conjunto de teste nos dois casos do KNN. Ainda, estabelecemos $K = 1$ para os métodos KNNE e KNNM. Na Fig. 3.6 vimos que para avanços $a > 1$ as cores se misturam, por isso não conseguimos identificar facilmente classes $\Omega_{n+a} > 6$ e isso dificulta a tomada de decisão da MLP sobre os limites, desta forma para

comparação escolhemos $a = 1$. A acurácia para a MLP está marcada pelos quadrados pretos, para $KNNE$ marcamos em rosa e para $KNNM$ usamos círculos vazados verdes. Notamos que a acurácia é superior a 0.90 para os três casos e que a MLP apresenta valores menores se comparados aos dois métodos KNN , com exceção das classes 15 e 16.

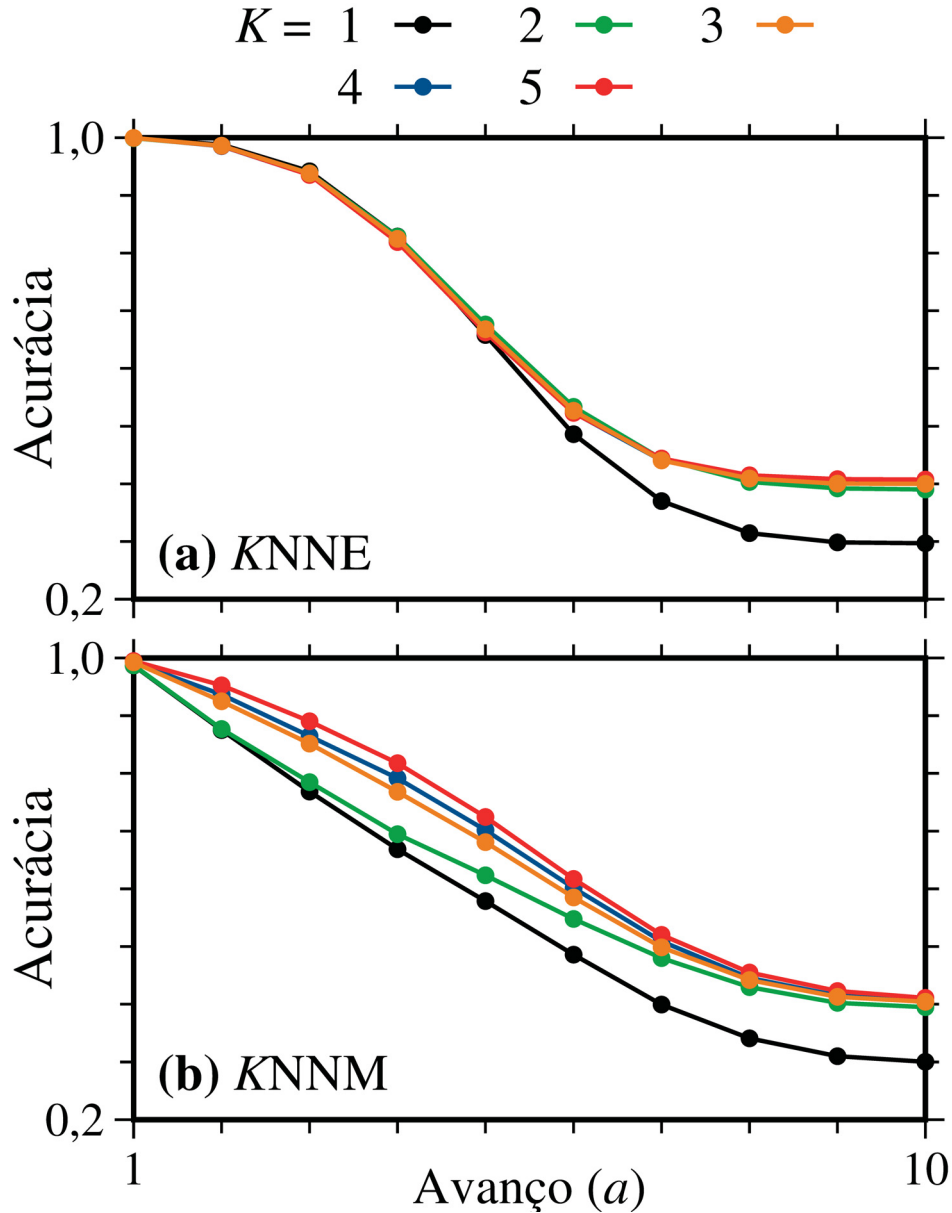


Figura 3.8 – Taxas de acerto usando o método KNN para a asa Ω_{n+a} como função do inteiro $a = 1, 2, \dots, 10$. O número K de vizinhos é identificado pelas cores, e a distância medida entre os vizinhos é (a) euclidiana e (b) mínima distância. Enquanto em (a) dois vizinhos são suficientes para alcançar o efeito de muitos vizinhos, em (b) a acurácia é mais dependente do número de vizinhos usados.

Para avanços $a \geq 1$ temos a comparação entre $KNNE$ e $KNNM$ na Fig. 3.8. Para obtenção destes resultados separamos dois conjuntos, um de referência e um de teste,

ambos com 250000 elementos e mantendo a mesma distribuição exponencial apresentada na Fig. 3.5. Na Fig. 3.8 temos as acurácias usando $KNNE$ e $KNNM$ como função das classes Ω_{n+a} a serem preditas. Calculamos ambos KNN para até 5 vizinhos, sendo estes resultados apresentados por diferentes cores, como mostra a legenda acima da figura. Na Fig. 3.8(a) os valores são para a distância Euclidiana entre os vizinhos e na (b) usando a mínima distância 3.3. Ambas apresentam decaimento na acurácia conforme o avanço a aumenta. Além disso, notamos que na Fig. 3.8(a) as curvas apresentam comportamento

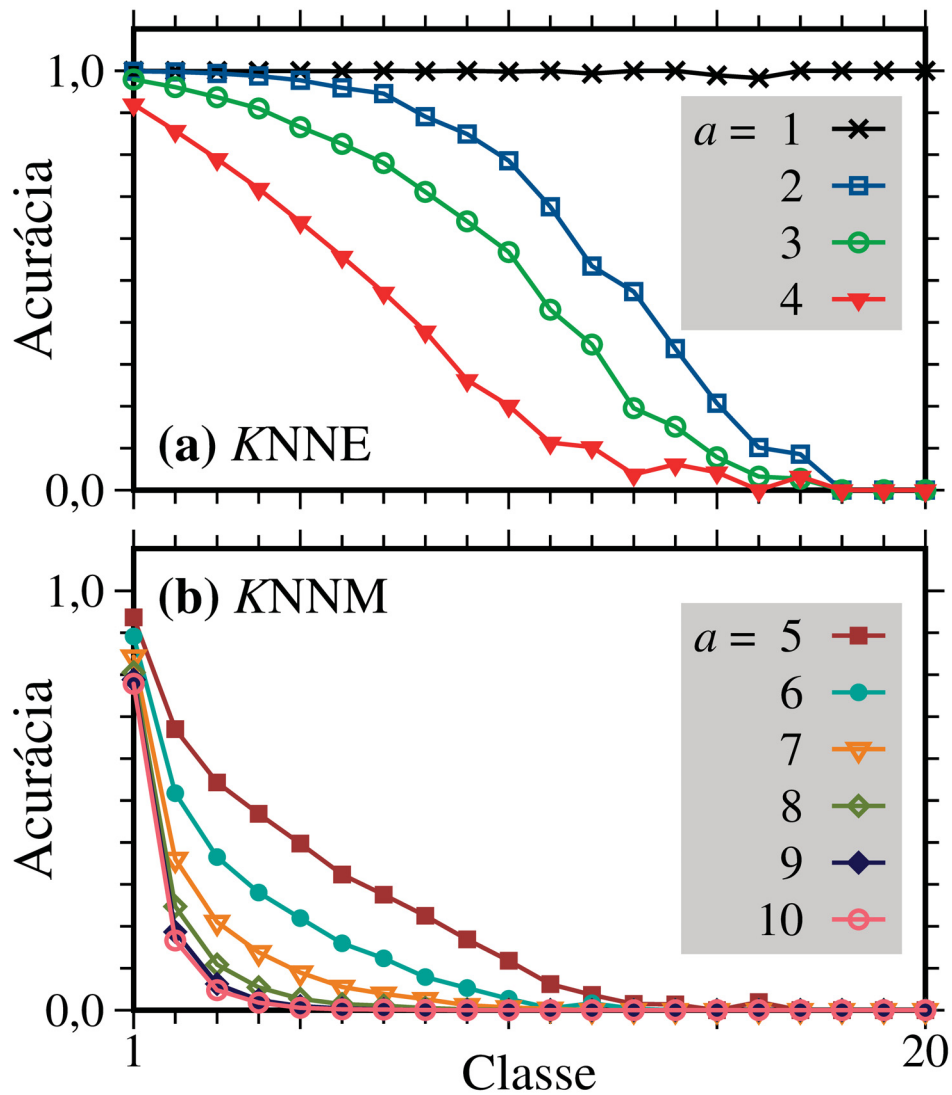


Figura 3.9 – Acurácia em cada classe. (a) Melhores resultados para $K = 1$ vizinho com $a \leq 4$. (b) Melhores resultados para $K = 5$ vizinhos com $5 \leq a \leq 10$. Enquanto somente a acurácia para $a = 1$ não decai para classes altas, para todos os outros a ela decai.

parecido, com exceção daquela para $K = 1$, diferentemente do que ocorre em (b). Ainda, vemos que a curva vermelha está mais acima o que indica que a acurácia é maior se

incluirmos mais vizinhos no método, pelo menos para o caso $KNNM$.

Agora na Fig. 3.9 temos a acurácia para cada classe e em cores o avanço a , ainda optamos por apresentar os melhores resultados obtidos, por isso há diferença na quantidade de vizinhos. Na Fig. 3.9(a) selecionamos o avanço no intervalo $1 \leq a \leq 4$ com $K = 1$ usando a distância Euclidiana. Para a Fig. 3.9(b) temos o avanço $a \geq 5$ com $K = 5$ usando a mínima distância. Destacamos que os resultados para $a = 1$ com $K = 1$ na Fig. 3.9(a) se mantêm muito próximo de 1, até mesmo para classes maiores. Já olhando para os demais avanços, podemos dizer que a predição de classes maiores decai conforme o avanço aumenta. Podemos relacionar isso com a mistura das cores na Fig. 3.6 segundo o aumento de a .

3.3.1 KNN com probabilidade Bayesiana

Buscamos métodos que resultem em acurácias para predição ainda maiores do que as encontradas. Por consequência, investindo no método KNN , incluímos a probabilidade bayesiana [61–63]. Ela nos permite associar uma probabilidade *a posteriori* como função de uma probabilidade *a priori*

$$P(\Omega|\mathcal{C}\Omega') = \frac{P(\mathcal{C}\Omega'|\Omega)P(\Omega)}{P(\mathcal{C}\Omega')}, \quad (3.4)$$

onde $P(\Omega|\mathcal{C}\Omega')$ é a probabilidade condicional da classe Ω , sendo que o método KNN a classificou como Ω' ($\mathcal{C}\Omega'$ significa classificar como Ω'). A probabilidade condicional 3.4 é uma função das ocorrências de Ω ($P(\Omega)$) e $\mathcal{C}\Omega'$ ($P(\mathcal{C}\Omega')$) e também da probabilidade $P(\mathcal{C}\Omega'|\Omega)$, que é a probabilidade de um ponto de classe Ω ser classificado como Ω' . Portanto, a classe predita é assumida como a mais provável de ser a probabilidade *a posteriori*. Nomeamos $KNNMB$ como o método KNN usando distância mínima 3.3 adicionado a probabilidade calculada a partir da Eq. 3.4.

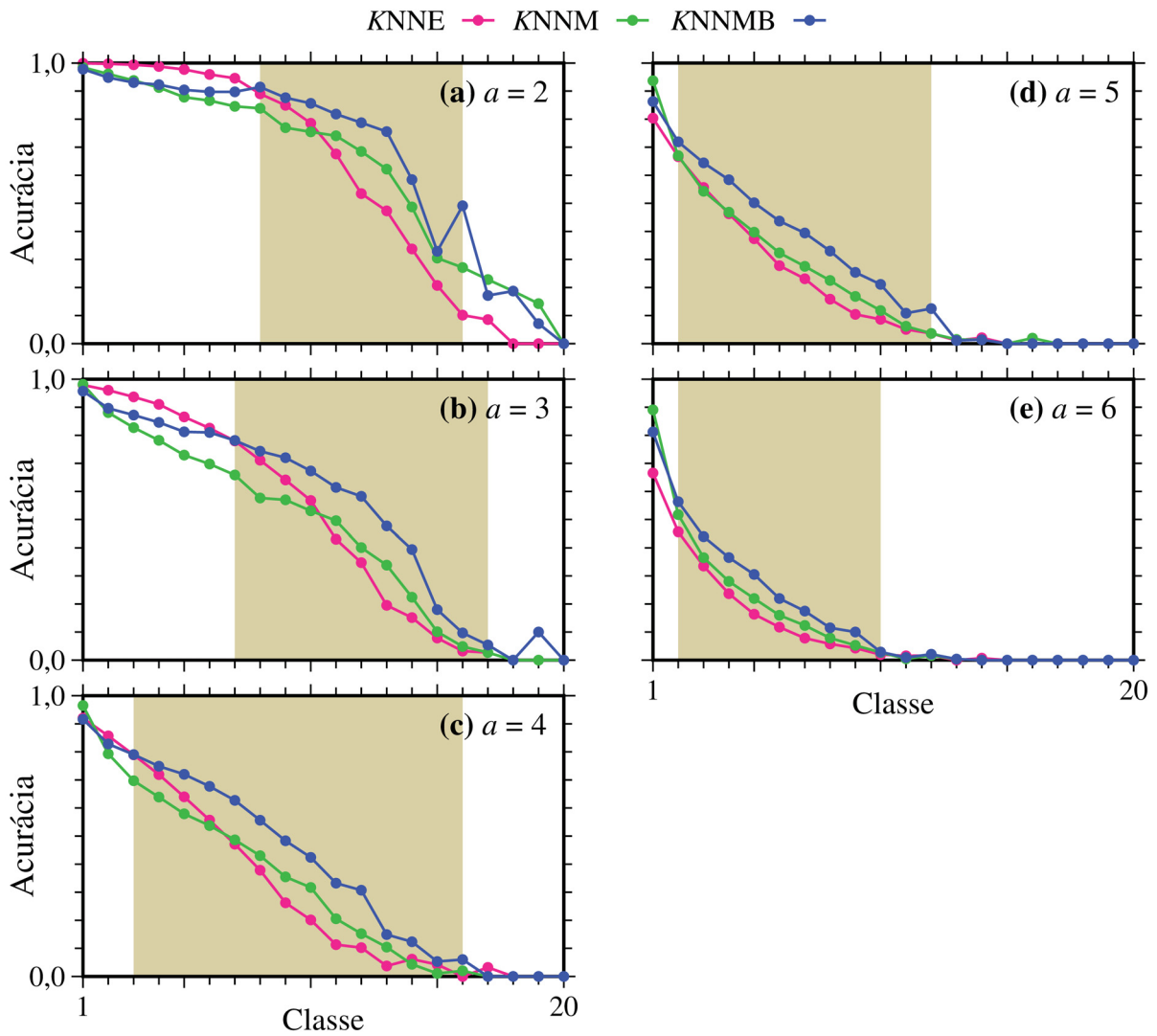


Figura 3.10 – Acurácia em cada classe. Comparação entre diferentes métricas e a classificação com probabilidade Bayesiana. K NNE para $K = 1$, K NNM para $K = 5$ e K NNMB para $K = 5$. A região em bege destaca os melhores resultados para K NNMB. Avanços (a) $a = 2$, (b) $a = 3$, (c) $a = 4$, (d) $a = 5$ e (e) $a = 6$. A região destacada em bege caminha a esquerda conforme o avanço aumenta, como podemos ver para $a > 4$, com exceção da classe $\Omega = 1$, K NNMB tem acurácia maior para classes menores.

Aplicamos o K NN nos mesmos conjuntos de referência e teste usados anteriormente e podemos agora comparar os resultados para os três casos do K NN. A Fig. 3.10 mostra a acurácia em cada classe para as versões do método K NN. Os resultados para K NNE com $K = 1$ estão em rosa, em verde são os valores para K NNM com $K = 5$ e para K NNMB usamos a cor azul com $K = 5$. Na figura também vemos os resultados para diferentes avanços, em (a) temos $a = 2$, em (b) $a = 3$, na (c) temos $a = 4$, em (d) $a = 5$ e em (e) temos o avanço $a = 6$. Destacamos pela região com fundo bege as classes com

os melhores resultados para $KNNMB$. Notamos que houve melhora para $8 \leq \Omega_{n+2} \leq 14$, $7 \leq \Omega_{n+3} \leq 12$, $3 \leq \Omega_{n+4} \leq 8$, $2 \leq \Omega_{n+5} \leq 5$ e $\Omega_{n+6} = 3$ usando $KNNMB$. Assim, acurácias maiores nas classes maiores foram encontradas para avanços menores e também vemos melhores resultados de $KNNMB$ para classes menores (porém acima de 2) para avanços maiores. Visualmente é como se a região de cor bege se deslocasse para a esquerda conforme o avanço aumenta. Para maiores valores de a não vimos melhora significativa.

Capítulo 4

RESULTADOS - 4 ASAS

O sistema com quatro asas 2.23 foi integrado usando o método Runge-Kutta de quarta ordem [41] com passo fixo de igual a 10^{-3} e um transiente de 10^6 . Mantivemos os parâmetros fixos em $(b, r, \sigma) = (8/3; 28; 10)$ como feito para o sistema anterior e também usamos valores aleatórios dentro do intervalo $0 < s_0, t_0, z_0 < 1$ para as condições iniciais. O maior expoente de Lyapunov obtido foi $\lambda \approx 1,306$ ao longo de 10^7 pontos do atrator com o algoritmo de Wolf [38]. Vamos apresentar nas seções seguintes os resultados obtidos para o sistema com quatro asas.

4.1 ÂNGULOS ENTRE OS VCLs

O atrator para o sistema com quatro asas já foi apresentado na Fig. 2.10. Agora mostramos o mesmo atrator dividido em quatro regiões coloridas com tons de azul conforme mostra a Fig. 4.1. Além da divisão em regiões, temos os ângulos entre os VCLs no atrator conforme a escala de cores. Nas extremidades das asas temos valores para os três ângulos próximos de zero (vermelho) e na parte interna do atrator, ou seja em $(s, t) = (0, 0)$ temos valores próximos a $\pi/2$ (preto). Novamente estabelecemos para $\theta^{(i,j)} > \pi/2$ a transformação $\theta^{(i,j)} \mapsto \pi - \theta^{(i,j)}$.

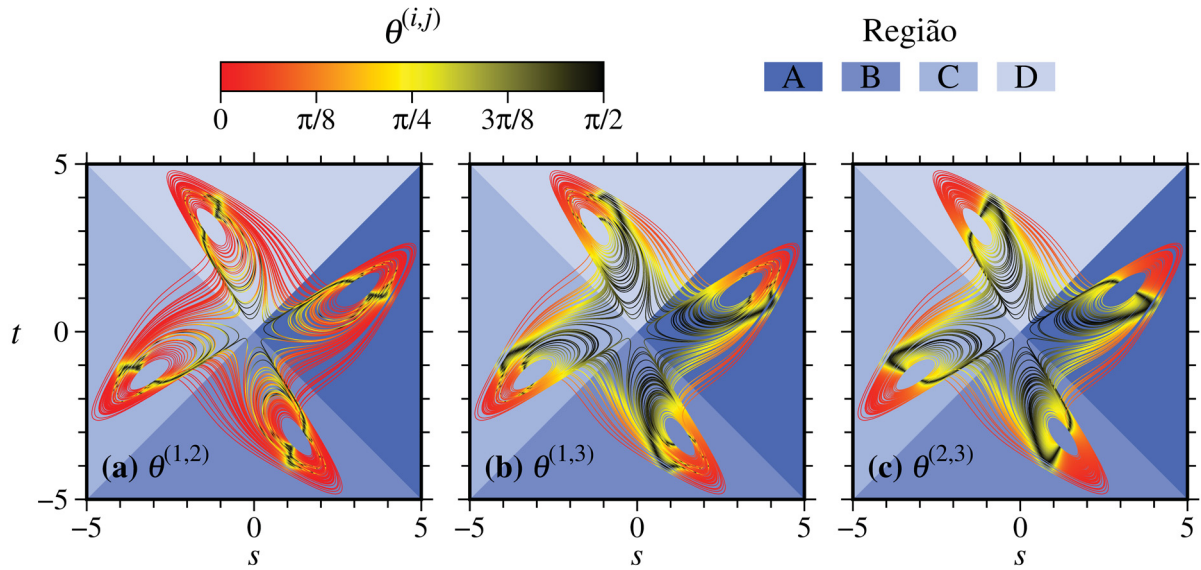


Figura 4.1 – Atrator do sistema proto-Lorenz com 4 asas no plano $t \times s$. Foram utilizados 10^5 pontos e as cores (veja a paleta de cores) indicam os ângulos entre as variedades (a) instável e central, (b) instável e estável e (c) central e estável no espaço tangente. Todos os três ângulos tendem a zero nas extremidades das asas e a $\pi/2$ próximos da origem $(0,0)$.

A Figura 4.2 mostra em (a) as séries para s e t e em (b) o ângulo $\theta^{(2,3)}$, ambos com as cores de fundo delimitando as mesmas regiões da Fig. 4.1. Aqui notemos no eixo t os tempos τ_n , τ_{n+1} e assim por diante que são os tempos que utilizamos para calcular a duração dos regimes. Destacamos em vermelho na Fig. 4.2(b) os ângulos extremos de cada região, os quais também chamamos de $\varphi^{(i,j)}$ para esse caso, assim como fizemos para o caso com três asas. Novamente, adotamos o alinhamento somente depois do último mínimo de z para considerarmos só o alinhamento extremo perto da troca de asa. Dessa forma, $\varphi^{(2,3)}$ é o menor valor de $\theta^{(2,3)}$ dentro de uma região durante a última volta na asa. Como já fizemos para o caso com três asas, vamos olhar para uma combinação de ângulos extremos $\varphi^{(i,j)}$, porém aqui juntaremos essa informação com a duração para analisarmos se há alguma relação entre elas.

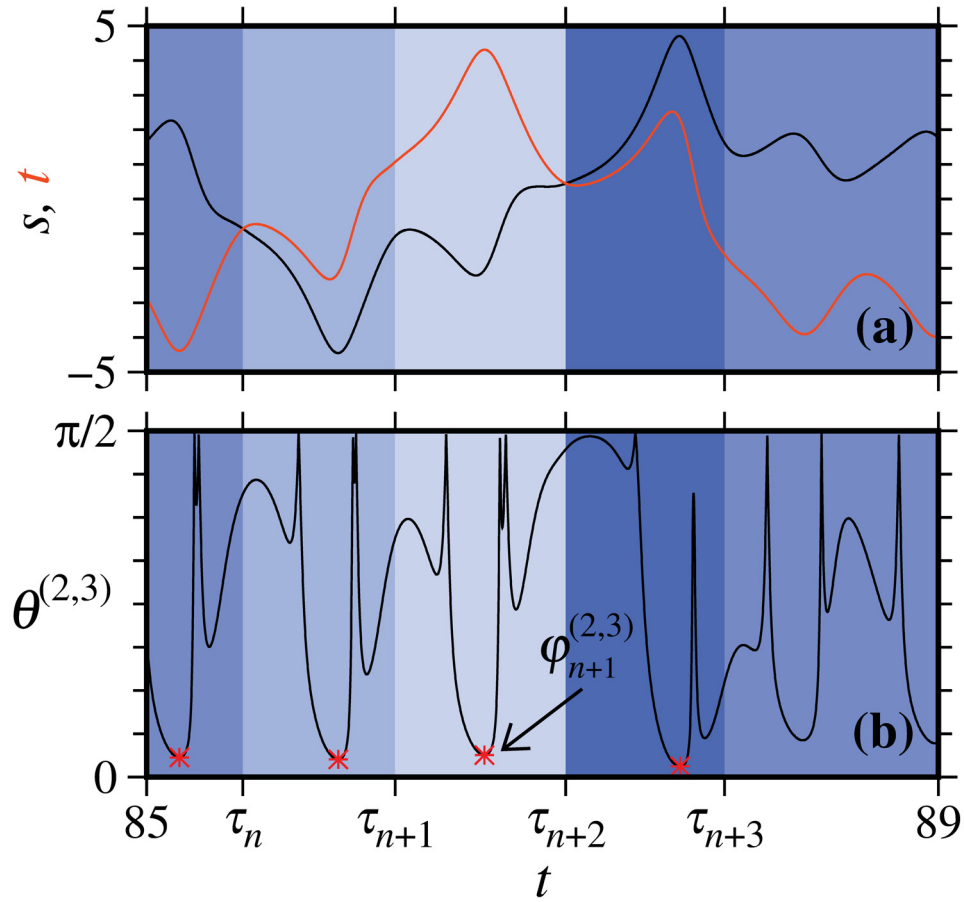


Figura 4.2 – Janela temporal mostrando (a) (s, t) , (b) o ângulo $\theta^{(2,3)}$. As cores de fundo identificam as regiões A, B, C e D da Fig. 4.1. O menor valor de $\theta^{(2,3)} (= \varphi_n^{(2,3)})$ em cada região está destacado em vermelho ao longo da série.

4.2 DURAÇÕES E PREDIÇÃO

A Figura 4.3 apresenta a duração Δt_{n+1} como função dos ângulos $\varphi_n^{(1,2)}$ e $\varphi_n^{(2,3)}$. Relembrando, o índice a em Δt_{n+a} se refere ao regime a frente que estamos considerando, como para o caso de três asas. A paleta de cores indica a duração na figura. É interessante ressaltar que na Fig. 4.3 estamos olhando para os ângulos extremos em uma região n e para a duração do regime na região $n + 1$. Destacamos também que é facilmente observável uma separação entre os valores da duração e ainda, que conforme ambos os ângulos $\varphi_n^{(1,2)}$ e $\varphi_n^{(2,3)}$ se aproximam de 0, a Δt_{n+1} aumenta, ou então, podemos dizer que alinhamentos maiores predizem durações maiores.

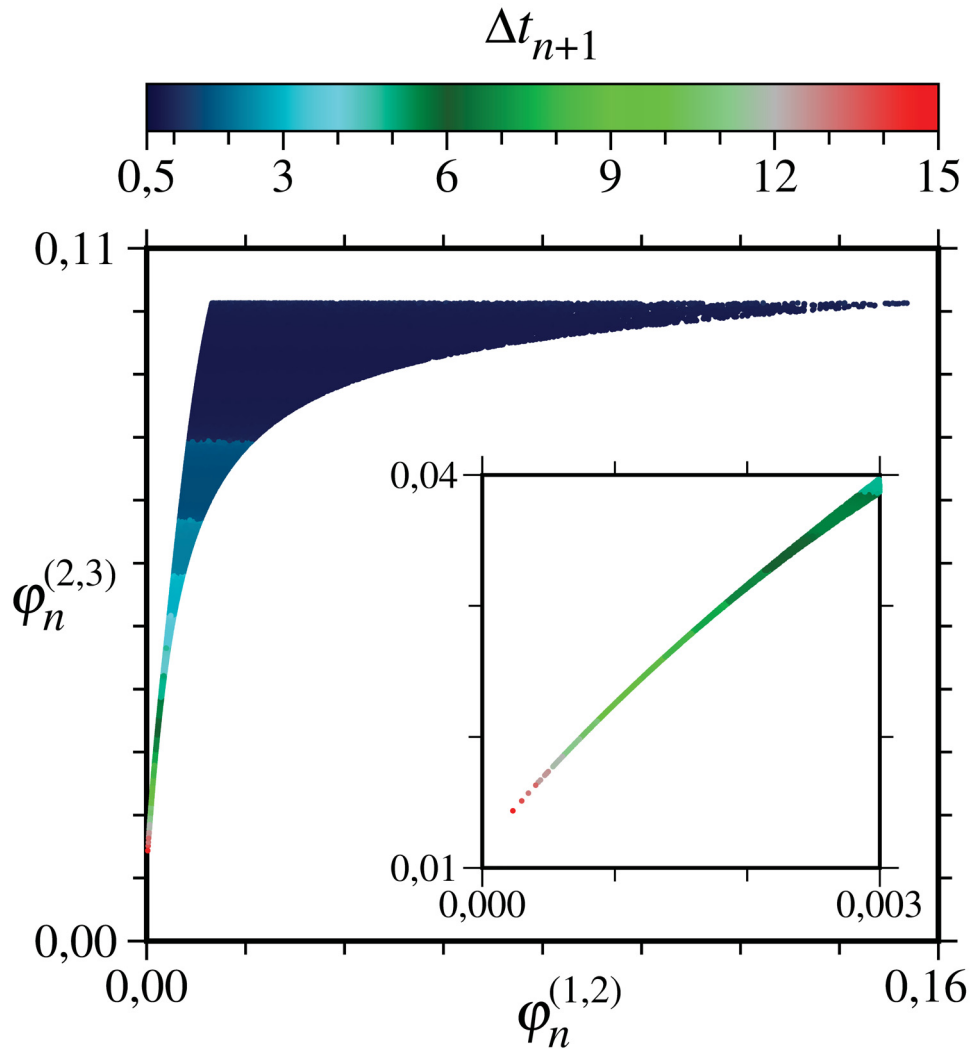


Figura 4.3 – Durações Δt_{n+1} (veja a legenda de cores) como função dos ângulos $\varphi_n^{(1,2)}$ e $\varphi_n^{(2,3)}$. São 571488 pontos obtidos de 100 séries temporais. As durações são bem divididas em regiões, o quadro interno mostra o padrão crescente da duração conforme o alinhamento aumenta. Em outras palavras, como os ângulos $\varphi_n^{(1,2)}$ e $\varphi_n^{(2,3)}$ diminuem, as durações aumentam.

Olhando para a Fig. 4.3 percebemos que durações menores ocupam espaços maiores no plano $\varphi_n^{(2,3)} \times \varphi_n^{(1,2)}$. Na Fig. 4.4 apresentamos a ocorrência das durações em tempos de Lyapunov (t_L). Para os tempos de Lyapunov usamos o valor do maior expoente de Lyapunov encontrado e calculamos

$$t_L = \frac{1}{\lambda_{max}}. \quad (4.1)$$

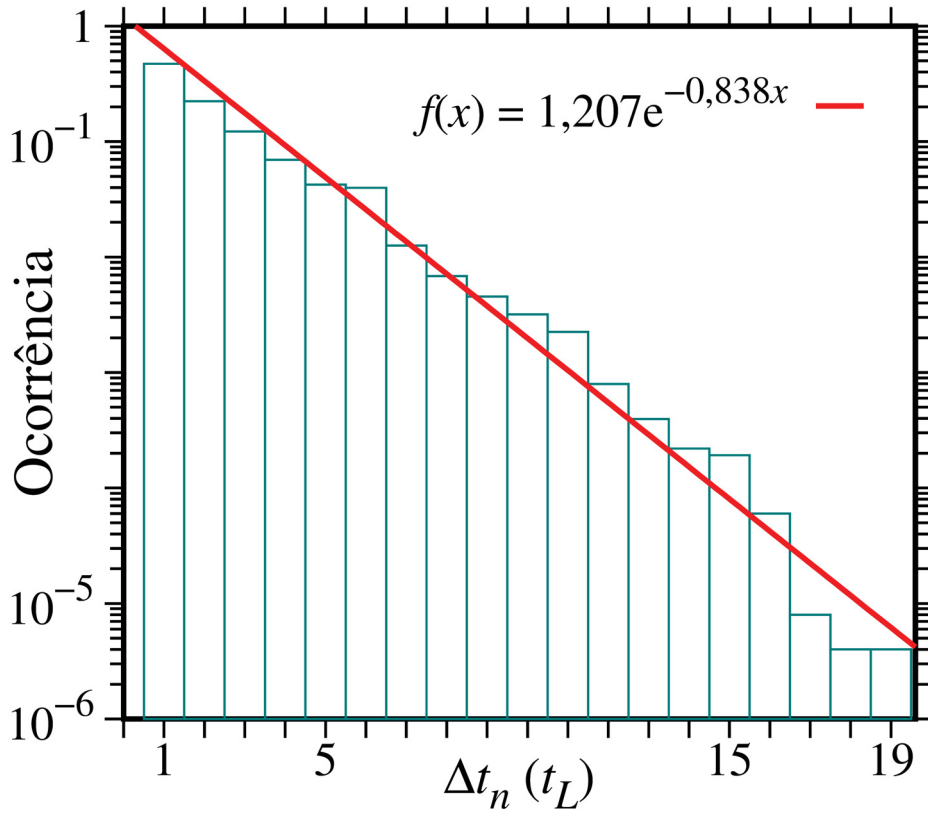


Figura 4.4 – Ocorrência de cada duração. A ocorrência decai exponencialmente, como indicado pela linha vermelha.

Desta forma, na Fig. 4.4 caso queiramos retomar a duração, basta multiplicarmos $\Delta t_n(t_L)$ por t_L e retomamos Δt_n . Notamos que durações menores tem ocorrência maior, conforme já tínhamos observado na Fig. 4.3.

A combinação de dois ângulos extremos no regime n e a duração no regime $n + 1$, ou seja para o avanço $a = 1$ foi apresentada na Fig. 4.3 e agora vamos olhar para outros valores de avanço, como mostrado na Fig. 4.5. Na Fig. 4.5(a) temos Δt_{n+2} , em (b) Δt_{n+3} e em (c) Δt_{n+4} . Vemos que as faixas que antes eram bem delimitadas na Fig. 4.3 perdem a homogeneidade conforme o avanço aumenta. Testamos até 8 avanços mas já percebemos que para $a \geq 4$ há uma mistura muito grande nos valores de duração e torna a previsão difícil.

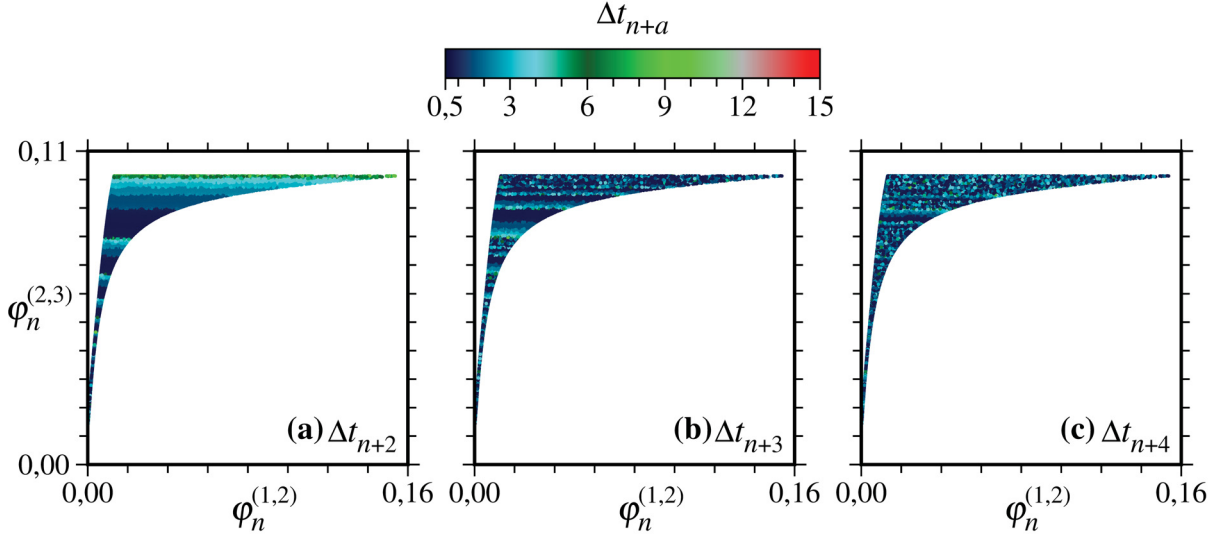


Figura 4.5 – Durações (a) Δt_{n+1} , (b) Δt_{n+1} , (c) Δt_{n+1} são mostradas como função dos ângulos $\varphi_n^{(1,2)}$ e $\varphi_n^{(2,3)}$. As cores seguem a legenda. Em (a) notamos faixas nas quais as durações decaem conforme ambos os ângulos se aproximam de 0. Em (b) essas faixas são mais dificilmente observáveis e em (c) já temos uma mistura grande, o que torna a previsão complicada.

4.3 KNN

Os dados obtidos para previsão dos regimes na seção anterior necessitam dos procedimentos já descritos para encontrar os valores dos ângulos e as durações. Conseguimos obter as durações dos regimes a frente olhando para valores dos ângulos extremos no regime atual. Agora, vamos criar um rótulo para cada ponto chamado de $\Gamma_i^{(a)}(\varphi_n^{(i,j)}, \varphi_n^{(i,j)}; \Delta t_{n+a}(t_L))$ e a partir dele vamos utilizar o *KNN* para prever as durações. Dos 571.488 pontos do conjuntos de dados, geramos dois conjuntos com 250000 pontos cada, sendo um conjunto de referência e um de teste. Mantivemos a mesma distribuição exponencial apresentada na Fig. 4.4 em ambos os conjuntos.

Para o método *KNN* é importante definir o cálculo da distância e a quantidade de vizinhos considerados e aqui utilizamos a distância euclidiana e obtivemos resultados variando a quantidade de vizinhos entre $1 < K < 5$, porém os melhores resultados e os que estão aqui apresentados foram considerando somente 1 vizinho. O primeiro resultado para o *KNN* está apresentado na Fig. 4.6.

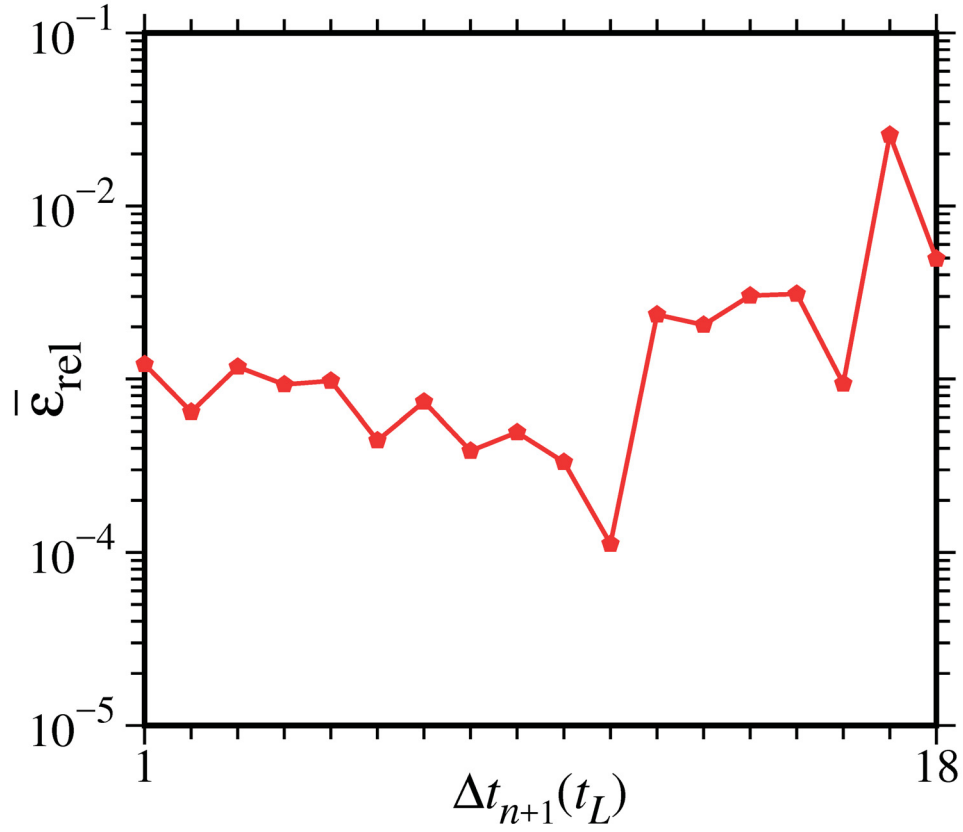


Figura 4.6 – Erro relativo médio $\bar{\varepsilon}_{rel}$ em função das durações $\Delta t_{n+1}(t_L)$. $\Delta t_{n+1}(t_L)$ são em função do tempo de Lyapunov.

Na Figura 4.6 apresentamos o *erro relativo médio* $\bar{\varepsilon}_{rel}$ em função das durações $\Delta t_{n+1}(t_L)$. Para encontrarmos esse erro relativo médio adotamos o seguinte procedimento: do conjunto de referência calculamos a duração prevista $\Delta t_{n+a}^{prevista}$ e então o erro relativo é obtido fazendo

$$\varepsilon_{rel} = \left| \frac{\Delta t_{n+a}^{prevista} - \Delta t_{n+a}^{teste}}{\Delta t_{n+a}^{teste}} \right|, \quad (4.2)$$

onde Δt_{n+a}^{teste} é a duração do conjunto de teste. Em seguida, definimos intervalos, por exemplo entre 1×10^3 e 5×10^3 ou então 1×10^2 e 5×10^2 e, dentro desse intervalo somamos os ε_{rel} e dividimos pela ocorrência do intervalo:

$$\bar{\varepsilon}_{rel} = \frac{1}{\text{ocorrência}} \sum_{\text{esquerda}}^{\text{direita}} \varepsilon_{rel}, \quad (4.3)$$

sendo *esquerda* o limite inferior do intervalo e *direita* o limite superior. Na Fig. 4.6 notamos que somente para $\Delta t_{n+a}(t_L) = 17$ tivemos $\bar{\varepsilon}_{rel} > 10^{-2}$ e ainda, para $\Delta t_{n+a}(t_L) < 12$ temos erros próximos ou menores que 10^{-3} .

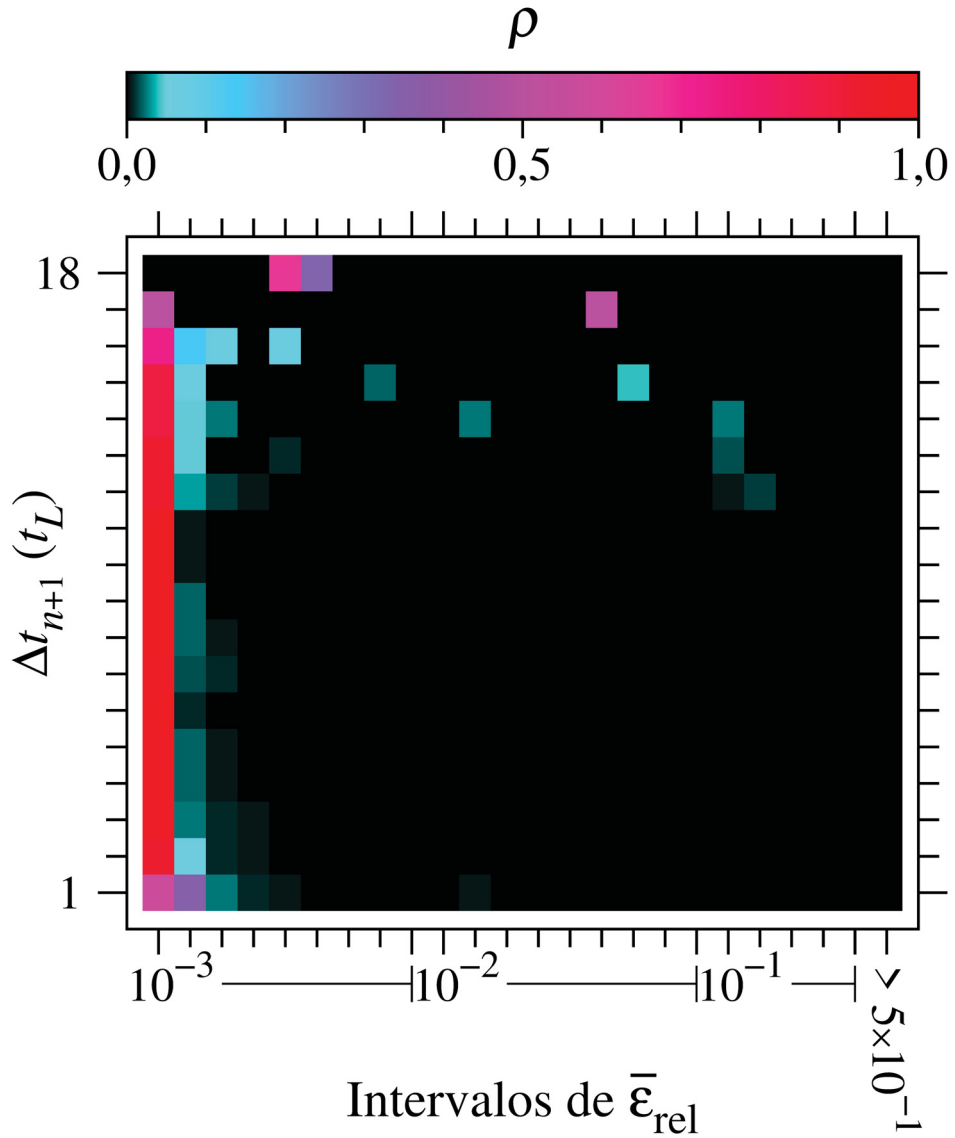


Figura 4.7 – Quadro $\Delta t_{n+1}(t_L) \times \bar{\varepsilon}_{rel}$ onde em cores temos ρ que é quantidade de durações dentro de um intervalo do erro relativo médio.

A Figura 4.7 nos mostra uma relação entre as durações $\Delta t_{n+1}(t_L)$ e os intervalos de $\bar{\varepsilon}_{rel}$. O eixo dos intervalos de $\bar{\varepsilon}_{rel}$ se assemelha a uma escala logarítmica, para interpretá-

la temos 10 divisões para os intervalos 10^{-3} e 10^{-2} , já para 10^{-1} temos 5 e os valores mais altos de $\bar{\varepsilon}_{rel}$ ficam marcados em $> 5 \times 10^{-1}$. A cor na Fig. 4.7 indica o valor de ρ , que é a quantidade de durações que tem dentro de um intervalo de $\bar{\varepsilon}_{rel}$. Assim, se olharmos uma linha a soma dos valores de ρ deve resultar 1. Conseguimos ver que durações $\Delta t_{n+1}(t_L) \geq 12$ apresentam ρ distribuídos em intervalos de $\bar{\varepsilon}_{rel}$ maiores que as demais durações. Também podemos pensar que encontramos uma maior quantidade de erros relativos médios em na ordem de 10^{-3} para durações menores.

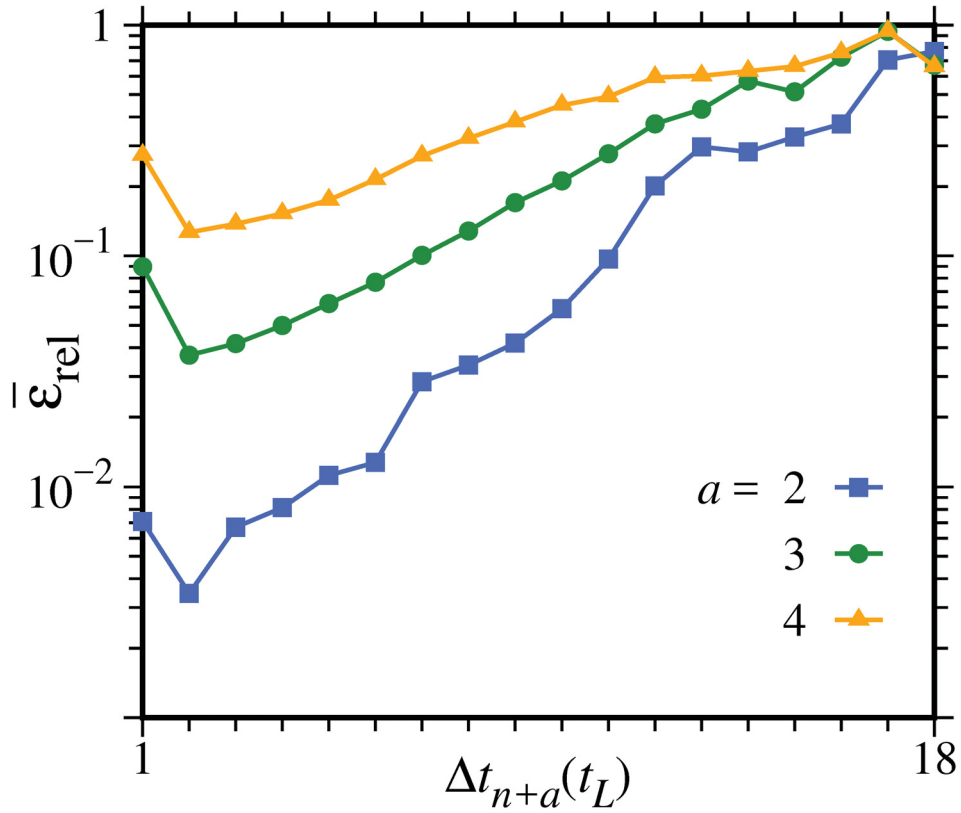


Figura 4.8 – Erro relativo médio $\bar{\varepsilon}_{rel}$ em função das durações $\Delta t_{n+a}(t_L)$. Em cores temos os valores dos avanços conforme a legenda. Para avanços menores temos $\bar{\varepsilon}_{rel}$ menores, assim como para durações $\Delta t_{n+a}(t_L)$ menores os erros também são menores.

A Figura 4.8 mostra o $\bar{\varepsilon}_{rel}$ para diferentes avanços. Em azul temos $a = 2$, em verde $a = 3$ e em laranja $a = 4$. Para avanços maiores obtemos erros maiores e por isso optamos por mostrar até 4 avanços. Observamos na figura que somente para $a = 2$ temos erros da ordem de 10^{-2} e para as demais, vemos que erros menores estão ligados a $\Delta t_{n+a}(t_L)$

menores, bem como avanços menores também.

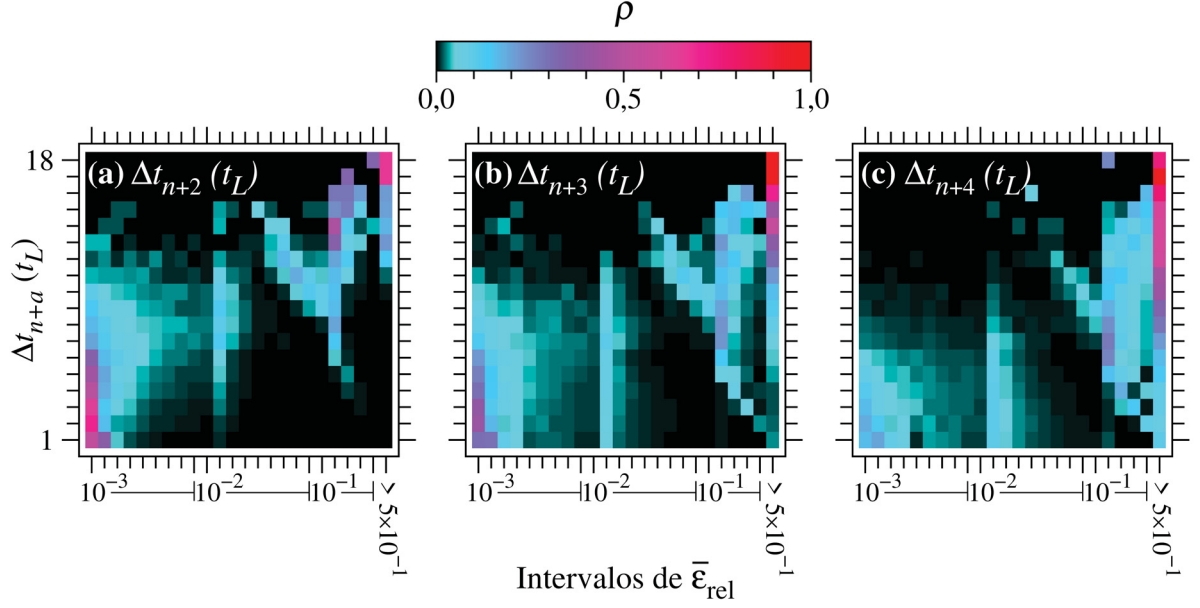


Figura 4.9 – Quadro $\Delta t_{n+a}(t_L) \times \bar{\epsilon}_{rel}$ onde em cores temos ρ que é quantidade de durações dentro de um intervalo de erro relativo médio. Em (a) $\Delta t_{n+2}(t_L)$, (b) $\Delta t_{n+2}(t_L)$ e em (c) $\Delta t_{n+3}(t_L)$

Apresentamos a Fig. 4.7 para durações $\Delta t_{n+1}(t_L)$, agora na Fig. 4.9 temos o mesmo quadro mas para durações em outros avanços: em (a) $\Delta t_{n+2}(t_L)$, em (b) $\Delta t_{n+2}(t_L)$ e em (c) $\Delta t_{n+3}(t_L)$. De forma mais detalhada que na Fig. 4.8, na Fig. 4.9 conseguimos ver os a concentração dos valores para $\bar{\epsilon}_{rel}$. Percebemos para os três avanços que temos erros $> 5 \times 10^{-1}$ para alguns casos. Para o caso de $a = 2$ vemos os erros para durações menores concentrados em intervalos de $\bar{\epsilon}_{rel}$ pequenos.

Capítulo 5

CONCLUSÃO

Encontrar a duração dos regimes no sistema proto-Lorenz foi o nosso objetivo. Iniciamos escolhendo este sistema que é um modelo generalizado de Lorenz. Para o Lorenz clássico temos duas asas, enquanto que nos modelos aqui estudados vemos três e quatro asas. Nosso interesse foi em prever a duração de um regime a frente usando os dados da atual região. Consideramos até dez avanços para frente para o caso com três asas, e assim podemos pensar em uma região e tais resultados (para o décimo avanço) estão predizendo a duração da décima asa a frente, ou então, é como se percorrêssemos mais de duas voltas no atrator. Já para o caso com quatro asas, escolhemos olhar para até oito avanços, que aqui podemos pensar que estamos tomando os dados da região A e olhando para duas voltas completas no atrator, porém os resultados com alguma informação relevante se encontram para até quatro avanços, ou uma volta completa no atrator.

Para ambos os sistemas, integramos e paralelamente calculamos o expoente de Lyapunov e investigamos seus vetores covariantes de Lyapunov, bem como o ângulo entre as variedades instável, estável e central. No caso com três asas optamos trabalhar com classes e então, encontramos para cada região a classe Ω_n onde essa é o máximo de z dentro da n -ésima asa e as classes de cada região estão diretamente relacionadas com a duração das mesmas, como mostrado pela equação para $\bar{\Delta}t_n$. Combinamos com o alinhamento entre os VCLs, uma vez que esses precedem a ocorrência de picos e a troca de regiões e assim formamos o rótulo $\Phi_i^{(a)}(\varphi_n^{(1,3)}, \varphi_n^{(2,3)}; \Omega_{i+a})$. No caso com quatro asas trabalhamos diretamente com as durações de cada regime e essas também foram combinadas com os ângulos extremos para formar o rótulo $\Gamma_i^{(a)}(\varphi_n^{(1,3)}, \varphi_n^{(2,3)}; \Delta t_{i+a}(t_L))$.

No caso com três asas temos a Fig. 3.4 que apresenta o rótulo mencionado e mostra regiões bem delimitadas para cada classe e ainda, que conforme ambos os ângulos extremos se aproximam de zero, a classe prevista é maior. Já para a Fig. 3.6 não encontramos regiões tão bem delimitadas quanto antes, especialmente para avanços maiores de quatro. As cores se misturam muito e acabamos perdendo a precisão nos limites das regiões de cada cor (classe).

Técnicas de aprendizado de máquinas foram aplicadas ao estudo. A rede MLP obteve uma acurácia superior a 95% para a maioria das classes, mas o método utilizando *KNN* foi ainda melhor. Esse método olha para a classes dos vizinhos mais próximos ao elemento teste e atribui a ele um rótulo. Dois critérios importantes influenciam nos resultados obtidos: o número de vizinhos e a distância para a vizinhança. Escolhemos estudar de um a cinco vizinhos para analisar os melhores resultados e para o cálculo da distância entre vizinhos usamos a distância euclidiana e a mínima distância.

As três técnicas citadas acima foram inicialmente aplicadas para um avanço $a = 1$ pois para mais avanços as cores se misturam e a decisão por limites é difícil para a MLP. Mantendo $K = 1$ e para $a = 1$ alcançamos acurácias superiores a 0,90 para todos os casos, sendo o método *KNN* superior na maioria das classes.

Nossa intenção foi investigar a predição para os avanços a frente também, assim aplicamos o método *KNN* para até dez avanços e analisamos os resultados para $1 \leq K \leq 5$ tanto para o caso com a distância euclidiana quanto para a mínima distância. Para *KNNE* conseguimos melhores resultados utilizando um vizinho, quanto que para *KNNM* as acurácias mais altas foram para cinco vizinhos. Desta forma, mantivemos as melhores configurações das versões do *KNN* e analisamos por classe. A acurácia decai conforme a classe aumenta para ambos os casos, *KNNE* e *KNNM*, com exceção do $a = 1$ para *KNNE*. Ainda assim, precisões maiores que 0,9 são obtidas para predições de classes em $a \leq 3$ asas visitadas subsequentemente.

Buscando incrementar o método *KNN* adicionamos a probabilidade bayesiana. Ela descreve a probabilidade de um evento, baseado em um conhecimento *a priori* que pode estar relacionado ao evento. Com este conseguimos acurácia maiores que 0,50 e melhores que *KNNE* e *KNNM* para as classes $8 \leq \Omega_{n+2} \leq 14$, $7 \leq \Omega_{n+3} \leq 12$, $3 \leq \Omega_{n+4} \leq 8$,

$2 \leq \Omega_{n+5} \leq 5$ e $\Omega_{n+6} = 3$. O intervalo de classes que $KNNMB$ é superior se desloca para as classes mais baixas conforme o avanço aumenta. Ainda que não tenhamos alcançado acurácias altas para avanços próximos a dez, os resultados aqui encontrados mostram sucesso quanto aos métodos aplicados para predição da duração nas asas do sistema.

A Figura 4.3 para o caso com quatro asas apresenta o plano $\varphi_n^{2,3} \times \varphi_n^{1,2}$ com cores relacionadas as durações Δt_{n+1} e nela vemos regiões bem delimitadas. Para regimes a frente, ou seja outros valores do avanço a obtemos a Fig. 4.5 e nela ainda conseguimos ver alguma divisão entre as regiões para $a = 2$ e 3 , mas para $a = 4$ vemos muita mistura entre os pontos, o que nos mostra que a previsão é perdida para avanços muito altos.

O método KNN também foi aplicado ao caso com quatro asas e considerando a distância euclidiana, os melhores resultados foram para 1 vizinho. Apresentamos dois tipos de figura para expor os resultados. O primeiro tipo nos mostra o plano $\bar{\varepsilon}_{rel} \times \Delta t_{n+a}(t_L)$ e temos duas figuras para mostrar essa relação para diferentes avanços que são as Fig. 4.6 e 4.8. Em ambas figuras vemos que para durações mais altas temos erros relativos médios maiores, ainda assim, para $a = 1$ a ordem dos erros é menor que 4×10^{-2} , com exceção do caso $\Delta t_{n+a}(t_L) = 17$. Já para os demais avanços, somente o caso $a = 2$ para $1 \geq \Delta t_{n+a}(t_L) \geq 4$ vemos erros até $\approx 0,008$, para as outras durações e avanços encontramos valores maiores.

As Figuras 4.7 e 4.9 nos apresentam a informação do ρ que é a quantidade de durações dentro de um intervalo de $\bar{\varepsilon}_{rel}$. Para $a = 1$ os valores de $\bar{\varepsilon}_{rel}$ se concentram nos intervalos da ordem de 10^{-3} , sendo que para durações maiores obtemos erros maiores também. Já na Fig. 4.9 temos olhando os valores de ρ que durações menores tem erros concentrados no intervalo 10^{-3} e as durações maiores ficam com erros acima de 10^{-1} . Ainda assim, para alguns avanços e a frente e durações intermediárias conseguimos resultados onde os erros relativos médios são baixos.

O nosso objetivo era encontrar a duração dos regimes nos sistemas generalizados de Lorenz propostos por Miranda e Stone [24] com três e quatro asas, uma vez que esses fornecem uma complexidade maior que aquela apresentada no sistema de Lorenz [23]. Olhando para a informação dos ângulos entre os VCLs juntamente com a classe ou duração, conseguimos obter a previsão dessa duração (classe) para regimes a frente, em ambos

os casos para até uma volta completa no atrator e para o caso com três asas, em até duas voltas completas no atrator. O artigo *Forecasting the duration of three connected wings in a generalizes Lorenz model* [65] apresenta os resultados aqui elaborados para o modelo com três asas. Assim, o objetivo foi alcançado e vemos que para durações (classes) menores temos previsões com menores erros em regimes a frente.

REFERÊNCIAS

- 1 BRUGNAGO, E. L.; HILD, T. A.; WEINGÄRTNET, D.; BEIMS, M. W. Classification strategies in machine learning techniques predicting regime changes and durations in the Lorenz system. **Chaos**, v. 30, 2020.
- 2 BRUGNAGO, E. L.; GALAAS, J. A.; BEIMS, M. W. Predicting regime changes and durations in Lorenz's atmospheric convection model. **Chaos**, v. 30, 2020.
- 3 BEIMS, M. W.; GALLAS, J. A. C. Manifold angles, the concept of self-similarity, and angle-enhanced bifurcation diagrams. **Scientific Reports**, v. 6, 2016.
- 4 BEIMS, M. W.; GALLAS, J. A. C. Alignment of Lyapunov vector: a quantitative criterion to predict catastrophes?. **Scientific Reports**, v. 6, 2016.
- 5 BEIMS, M. W.; GALLAS, J. A. C. Predictability of the onset of spiking and bursting in complex chemical reactions. **Physical Chemistry Chemical Physics**, v. 20, 2018.
- 6 SHARAFI, N.; TIMME, M.; HALLERBERG, S. Critical transitions and perturbation growth directions. **Physical Review E**, v. 96, 2017.
- 7 GINELLI, F.; et al. Characterizing Dynamics with Covariant Lyapunov Vector. **Physical Review Letters**, v. 99, 2007.
- 8 GINELLI, F.; et al. Covariant Lyapunov Vector. **Journal of Physics A: Mathematical and Theoretical**, v. 46, 2013.

-
- 9 LAI, Y. C.; YORKE, J. A.; KAN, I. How often are chaotic saddles nonhyperbolic?. **Nonlinearity**, v. 6, 1993.
 - 10 NORWOOD, A.; KALNAY, E.; IDE, K.; YANG, S. C.; WOLFE, C. Lyapunov, singular and bred vectors in a multi-scale system: an empirical exploration of vectors related to instabilities. **Journal of Physics A: Mathematical and Theoretical**, v. 46 (254021), 2013.
 - 11 KUPTSOV, P. V.; PARLITZ, U. Theory and computation of covariant Lyapunov vectors. **Journal of Nonlinear Science**, v. 22 (727), 2012.
 - 12 TRUANT, D. P.; MORRISS, G. P. Backward and covariant Lyapunov vectors and exponents for hard-disk systems with a steady heat current. **Physical Review E**, v. 90 (052907), 2014.
 - 13 KUPTSOV, P. V.; KUPTSOVA, A. V. Predictable nonwandering localization of covariant Lyapunov vectors and cluster synchronization in scale-free networks of chaotic maps. **Physical Review E**, v. 90 (032901), 2014.
 - 14 INUBUSHI, M.; TAKEHIRO, S.; YAMADA, M. Regeneration cycle and the covariant Lyapunov vectors in a minimal wall turbulence. **Physical Review E**, v. 92 (023022), 2015.
 - 15 PALATELLA, L.; TREVISAN, A.; RAMBALDI, S. Nonlinear stability of traffic models and the use of Lyapunov vectors for estimating the traffic state. **Physical Review E**, v. 88 (022901), 2013.
 - 16 KRÜGER, T. S.; GALUZIO, P. P.; PRADO, T. L.; VIANA, R. L.; SZEZECH, J. D.; LOPES, S. R. Mechanism for stickiness suppression during extreme events in hamiltonian systems. **Physical Review E**, v. 91 (062903), 2015.
 - 17 JOUSSEPH, C. A.; KRÜGER, T. S.; MANCHEIN, C.; LOPES, S. R.; BEIMS, M. W. Weak dissipative effects on trajectories from the edge of basins of attraction. **Physica A**, v. 456 (68), 2016.

-
- 18 HANSEN, R. S.; HANNESTAD, S. Chaotic flavor evolution in an interacting neutrino gas. **Physical Review D**, v. 90 (025009), 2014.
 - 19 TREVISAN, A. ; PANCOTTI, F. Periodic orbits, Lyapunov vectors, and singular vectors in the Lorenz system. **Journal of the Atmospheric Sciences**, v. 55 (390), 1998.
 - 20 SAIKI, Y.; KOBAYASHI, M. U. Numerical identification of nonhyperbolicity of the Lorenz system through Lyapunov vectors. **JSIAM Letters**, v. 2 (107), 2010.
 - 21 EVANS, E.; BHATTI, N.; KINNEY, J.; PANN, L.; PEÑA, M.; YANG, S. C.; KALNAY, E.; HANSEN, J. Rise undergraduates find that regime changes in Lorenz's model are predictable. **Bulletin of the American Meteorological Society**, v. 85 (520), 2004.
 - 22 YADAV, R. S.; DWIVEDI, S.; MITTAL, A. K. Prediction rules for regime changes and length in a new regime for the Lorenz model. **Journal of the Atmospheric Sciences**, v. 62 (2316), 2005.
 - 23 LORENZ, E.N. Deterministic Nonperiodic Flow. **Journal of the Atmospheric Sciences** v. 20, 1963.
 - 24 MIRANDA, R.; STONE, E. The Proto-Lorenz System. **Physics Letters A**, v. 178, 1993.
 - 25 COVER, T.; HART, P.; Nearest neighbor pattern classification. **IEEE Transactions on Information Theory**, v. 13 (1), 1967.
 - 26 GOODFELLOW, Y.; COUVILLE, A. **Deep Learning**. Courville: MIT Press, 2016.
 - 27 HASSELBLATT, B.; KATOK, A. **A first course in dynamics: with a panorama of recent developments**. Cambridge University Press, 2003.
 - 28 WIGGINS, S. **Introduction to applied nonlinear dynamical systems and chaos**, volume 2. Springer Science & Business Media, 2003.

-
- 29 OTT, E. **Chaos in Dynamical Systems**. 1a. ed. Nova York: Cambridge University Press, 1993.
- 30 ALLIGOOD, K.T.; SAUER, T.D.; YORKE, J.A. **Chaos An Introduction to Dynamical Systems**. Nova York: Springer, 1996.
- 31 LORENZ, E.N. **Predictability - a problem partly solved**. Predictability of Weather and Climate, ed Tim Palmer and Renate Hagedorn. Cambridge: Cambridge University Press, 2006.
- 32 GALLAS, J.A.C. The Structure of Infinite Periodic and Chaotic Hub Cascades in Phase Diagrams of Simple Autonomous Flows. **International Journal of Bifurcation and Chaos**, v. 20, 2010.
- 33 FELICIO, C.C.; RECH, P.C. Arnold Tongues and the Devil's Staircase in a Discrete-Time Hindmarsh-Rose Neuron Model. **Physics Letters A**, v. 379, 2015.
- 34 LICHTENBERG, A.J.; LIEBERMAN, M.A. **Regular and chaotic dynamics**, volume 38. Springer Science & Business Media, 1992.
- 35 ECKMANN, J. P.; RUELLE, D. Ergodic theory of chaos and strange attractors. **Reviews of Modern Physics**, v. 57, p. 617, 1985.
- 36 GOLDSTEIN, H.; POOLE, C. P.; SAFKO, J. **Classical Mechanics**. 3ª edição. San Francisco: Addison-Wesley Publishing Company, 2002.
- 37 MARION, J. B.; THORNTON, S.T. **Classical Dynamics of Particles and Systems**. 4ª edição. Pacific Grove: Brooks/Cole, 2005.
- 38 WOLF, A.; et al. Determining Lyapunov exponents from a time series. **Physica D**, v. 16, 1985.
- 39 BENETTIN, L.G.; GIORGILLI, A.; STRELCYN, J.-M. Lyapunov characteristic exponents for smooth dynamical systems and for Hamilton systems; A method for computing all of them. **Meccanica**, v. 15, 1980.

-
- 40 SCHIMADA, I.; NAGASHIMA, T. A Numerical Approach to Ergodic Problem of Dissipative Dynamical Systems. **Progress of Theoretical Physics**, v. 61, 1979.
- 41 BUTCHER, J.C. **Numerical Methods for Ordinary Differential Equations**. 2ª edição. New Zealand: University of Auckland, 2008.
- 42 MURPHY, K. P. **Machine Learning: a Probabilistic Perspective**. Cambridge: MIT Press, 2012.
- 43 BISHOP, C. M. **Pattern Recognition and Machine Learning**. Nova York: Springer, 2006.
- 44 MITCHELL, T. M. **Machine Learning**. Nova York: McGraw Hill, 1997.
- 45 HASTIE, T.; TIBSHIRANI, R. Discriminant adaptive nearest neighbor classification and regression. **In Advances in Neural Information Processing Systems**, v. 8, 1995.
- 46 WANG, J.; NESKOVIC, P.; COOPER, N. Improving nearest neighbor rule with a simple adaptive distance measure. **Pattern Recognition Letters**, v. 28, 2007.
- 47 ALTMAN, N. S. An Introduction to Kernel an Nearest-Neighbor Nonparametric Regression. **The American Statistician**, v. 46, 1992.
- 48 ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological Review**, v. 65, 1958.
- 49 YAROTSKY, D. Error bounds for approximations with deep ReLU networks. **Neural Networks**, v. 94, 2017.
- 50 HAYKIN, S. S. **Neural networks: a comprehensive foundation**. 2a edição. Prentice Hall PTR, 1999.

-
- 51 SALTZMAN, B. Finite Amplitude Free Convection as an Initial Value Problem - I. **Journal of the Atmospheric Sciences**, v. 19, 1962.
- 52 RAYLEIGH, L. On the Convective Currents in a Horizontal Layer of Fluid When the Higher Temperatures is on the Under Side. **Philosophical Magazine**, v. 32, 1916.
- 53 BÉNARD, H. Les Tourbillons Cellulaires dans une nappe liquide. **Revue Générale des Sciences Pures et Appliquées**, v. 11, 1900.
- 54 MONTEIRO, L.H.A. **Sistemas Dinâmicos**. 3ª edição. São Paulo: Livraria da Física, 2011.
- 55 SPARROW, C. **The Lorenz Equations: Bifrcations, Chaos, and Strange Attractors**. New York: Springer Verlag, 1982.
- 56 DULLIN, H.R.; et al. Extended Phase Diagram of the Lorenz Model. **International Journal of Bifurcation and Chaos**, v. 17, 2007.
- 57 ROY, D.; MUSIELAK, Z.E. Generalized Lorenz Models and their Routes to Chaos. I. Energy-conserving Vertical Mode Truncations. **Chaos, Solions and Fractals**, v. 32, 2007.
- 58 ROY, D.; MUSIELAK, Z.E. Generalized Lorenz Models and their Routes to Chaos. II. Energy-conserving Horizontal Mode Truncations. **Chaos, Solions and Fractals**, v. 31, 2007.
- 59 SHEN, B.W. Nonlinear Feedback in a Five-Dimensional Lorenz Model. **Journal of the Atmospheric Sciences**, v. 71, 2013.
- 60 SHEN, B.W. Nonlinear Feedback in a Six-Dimensional Lorenz Model: Impact of an additional heating term. **Nonlinear Processes in Geophysics**, v. 22, 2015.
- 61 BERRAR, D. Bayes' Theorem and the Naive Bayes Classifier. **Encyclopedia of Bioinformatics and Computational Biology**. v. 403, 2018.

-
- 62 OLSHAUSEN, B. A. Bayesian probability theory. **The Redwood Center for Theoretical Neuroscience**. 2004.
- 63 LINDLEY, D. V. **Bayesian statistics: A review**. Society for Industrial and Applied Mathematics, 1972.
- 64 BOLDRINI, J. L.; COSTA, S. I.; FIGUEREDO, V. L.; WETZLER, H. G. **Álgebra Linear**. Harper & Row 1972.
- 65 BRUGNAGO, E. L.; FELICIO, C. C.; BEIMS, M. W. Forecasting the duration of three connected wings in a generalized Lorenz model. **Internationl Journal of Bifurcation and Chaos**, aceito para publicação.

APÊNDICE A

Decomposição QR e Ortogonalização de Gram-Schmidt

O Processo de Gram-Schmidt é um algoritmo para obter uma base ortogonal (ou ortonormal) a partir de uma base qualquer, ou ainda, este método permite transformar um conjunto de vetores linearmente independentes em um conjunto ortogonal que gera o mesmo espaço vetorial [64].

Seja $A \in R^n \times R^n$ uma matriz com colunas linearmente independentes. Vamos aplicar o processo de ortogonalização de Gram-Schmidt no conjunto, obtendo os elementos não nulos e ortogonais, fazendo:

$$\hat{q}_k = a_k - \sum_{j=1}^{k-1} r_{jk} q_j, k = 1, \dots, n, \quad (1)$$

onde $r_{jk} = \langle v_k, q_j \rangle$, para $j = 1, \dots, k$. Para os vetores ortonormais q_k , normalizamos os vetores $\hat{q}_k$:

$$q_k = \frac{\hat{q}_k}{r_{kk}}, k = 1, \dots, n, \quad (2)$$

com $r_{kk} = \|\hat{q}_k\|_2 > 0$. Unindo Eq. 1 e 2, temos:

$$a_k = \sum_{j=1}^{k-1} r_{jk}q_j + r_{kk}q_k, k = 1, \dots, n, \quad (3)$$

ou,

$$\begin{aligned} a_1 &= q_1 r_{11}, \\ a_2 &= q_1 r_{12} + q_2 r_{22}, \\ &\vdots \\ a_n &= q_1 r_{1n} + q_2 r_{2n} + \dots + q_n r_{nn}. \end{aligned} \quad (4)$$

Podemos escrever na forma de um produto matricial, sendo assim

$$\begin{bmatrix} a_1 & a_1 & \dots & a_n \end{bmatrix} = \begin{bmatrix} q_1 & q_2 & \dots & q_n \end{bmatrix} \times \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ 0 & r_{22} & \dots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & r_{nn} \end{bmatrix} \Leftrightarrow A = QR. \quad (5)$$

A matriz Q é ortogonal, uma vez que suas colunas são vetores ortonormais e R é uma matriz triangular superior com diagonal positiva. Com isso temos a fatoração ortogonal $A = QR$ da matriz A .