

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Geradores de homologia persistente e aplicações

Carlos Henrique Venturi Ronchi

Dissertação de Mestrado do Programa de Pós-Graduação em
Matemática (PPG-Mat)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Carlos Henrique Venturi Ronchi

Geradores de homologia persistente e aplicações

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências – Matemática. *EXEMPLAR DE DEFESA*

Área de Concentração: Matemática

Orientador: Prof. Dr. Marcio Fuzeto Gameiro

USP – São Carlos
Outubro de 2019

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

V469g Venturi Ronchi, Carlos Henrique
 Geradores de homologia persistente e aplicações /
 Carlos Henrique Venturi Ronchi; orientador Marcio
 Fuzeto Gameiro. -- São Carlos, 2019.
 118 p.

 Dissertação (Mestrado - Programa de Pós-Graduação
 em Matemática) -- Instituto de Ciências Matemáticas
 e de Computação, Universidade de São Paulo, 2019.

 1. topologia algébrica. 2. homologia
 persistente. 3. enovelamento. 4. proteína. I. Fuzeto
 Gameiro, Marcio, orient. II. Título.

Carlos Henrique Venturi Ronchi

Persistent homology generators and applications

Dissertation submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP – in accordance with the requirements of the Mathematics Graduate Program, for the degree of Master in Science.
EXAMINATION BOARD PRESENTATION COPY

Concentration Area: Mathematics

Advisor: Prof. Dr. Marcio Fuzeto Gameiro

USP – São Carlos
October 2019

AGRADECIMENTOS

Primeiramente, gostaria de agradecer ao meu orientador Marcio Gameiro, que me deu a oportunidade de explorar as mais diversas ideias que tive ao longo desses anos, por todo apoio e discussões que tivemos. Também gostaria de agradecer por todas as portas que ele me abriu para fazer ciência e seguir com a minha carreira.

Eu também gostaria de agradecer Konstantin Mischaikow por ter aberto as portas para eu poder realizar o BEPE sob sua orientação. Os encontros e discussões foram muito proveitosos, me ajudando e ensinando muito do que é ciência e como fazer-la. Do mesmo lugar, gostaria de agradecer Lun Zhang pelas discussões que tivemos sobre homologia persistente.

Meus agradecimentos mais que especiais para minha esposa Priscila, que me acompanha nessa jornada a mais de 7 anos, através do seu apoio e conselhos que são imensuráveis. Suas palavras foram fundamentais e ajudaram muito durante essa caminhada que foi o mestrado.

Agradeço também aos meus pais por proporcionarem todo o apoio emocional e financeiro para que eu esteja onde cheguei. Sou muito privilegiado e agradeço por tudo que eles fizeram para eu chegar até ponto.

Sou muito grato a todos os amigos que fiz durante o mestrado, por terem me ajudado nas disciplinas através de discussões e conselhos.

Agradeço também a Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) pela bolsa de pesquisa concedida, por meio do processo nº 2017/14678-0, Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) e também pela bolsa de pesquisa do BEPE, processo nº 2018/20659-0, Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).

RESUMO

RONCHI, C. H. V. **Geradores de homologia persistente e aplicações**. 2019. 116 p. Dissertação (Mestrado em Ciências – Matemática) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2019.

Nos últimos anos dados são produzidos em uma taxa sem precedentes. Analisar todo o conhecimento obtido através de dados é cada vez mais difícil, devido a proporção das informações. Por isso, é necessário o desenvolvimento de novos métodos de análise.

A análise topológica de dados é uma nova área da matemática computacional que procura estudar e entender as propriedades topológicas de dados através de ferramentas como a homologia persistente. Este método estuda de forma geral as componentes conexas, buracos e cavidades dos conjuntos de dados. Este trabalho apresenta os princípios básicos da homologia persistente, sua fundamentação teórica e assim como algumas ferramentas relacionadas, como os ciclos ótimos e imagens de persistência.

Com estas ferramentas, propomos alguns modelos para o problema de enovelamento de proteínas. O primeiro é para a predição do score de estabilidade de um banco de proteínas. Alcançamos resultados próximos aos do estado da arte e apresentamos novas perspectivas para o desenvolvimento de proteínas. Já o segundo método estuda o panorama de energias de proteínas simuladas. Mostramos como a homologia persistente pode auxiliar softwares de modelagem para o desenvolvimento de proteínas mais estáveis.

Palavras-chave: Homologia, análise topológica de dados, proteína, enovelamento.

ABSTRACT

RONCHI, C. H. V. **Persistent homology generators and applications**. 2019. 116 p. Dissertação (Mestrado em Ciências – Matemática) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2019.

In the recent year data has been produced in a large scale and rapidly. It is becoming ever difficult to analyse it all through the current methods. Thus, it is necessary to develop and apply new methods.

Topological data analysis is a whole new field in computational mathematics/ algebraic topology that studies the topological properties of data through tools like persistent homology. This tool searches for components, holes and cavities in the data. In this dissertation we show the basic ideas of persistent homology, its theory as well as some related tools, such as optimal cycles and persistence images.

We propose using these tools models to the protein folding problem. The first one is a stability score predictor for a protein dataset. We show some results close to the state of art and new perspectives to the design of new proteins. In the second method we study the designed protein landscape energy. We show how persistent homology can be used as an aid to macromolecular designing softwares to get more stable proteins.

Keywords: homology, topological data analysis, protein, folding.

LISTA DE ILUSTRAÇÕES

Figura 1 – Representação do pipeline para a utilização da homologia persistente com um conjunto de dados.	24
Figura 2 – Exemplos de k -simplexos para $k \in \{0, 1, 2, 3\}$	25
Figura 3 – Exemplo em que a interseção de dois simplexos não é um simplexo. . .	26
Figura 4 – Exemplo de filtração para um complexo simplicial K	26
Figura 5 – Exemplo de um complexo simplicial abstrato e sua realização geométrica	28
Figura 6 – Exemplo de um complexo de Čech para um raio r fixado.	28
Figura 7 – Exemplo do complexo de Vietoris-Rips com os mesmos pontos utiliza- dos para a construção na Figura 6.	29
Figura 8 – Diagrama de Voronoi de três pontos no plano.	30
Figura 9 – Complexo Alpha para um conjunto de pontos no plano.	31
Figura 10 – Exemplo da filtração de um complexo simplicial e o barcode e diagramas de persistência associados.	33
Figura 11 – 0- e 1- Diagramas de Persistência da Figura 4.	35
Figura 12 – Filtração de um complexo simplicial.	36
Figura 13 – 0- e 1- diagramas de persistência da filtração mostrada na Figura 12.	36
Figura 14 – Exemplo de um diagrama de persistência de um módulo de persistência q -tame com um quadrante em destaque.	41
Figura 15 – Representação por intervalo (esquerda), pela função rank (meio) e pelo ponto decorado (direita) do módulo intervalar $\mathbf{k}[1, 3) = \mathbf{k}(1^-, 3^-)$	44
Figura 16 – Pontos decorados que são detectados pela medida aplicada no retângulo R	50
Figura 17 – Representação gráfica da Proposição 3.19	51
Figura 18 – Possíveis casos dos pontos decorados (r^*, s^*) na interseção $\cap_i R_i$	56
Figura 19 – Os morfismos Φ, Ψ recuperados do módulo de persistência \mathfrak{W} sobre $\Delta_x \cup \Delta_y$	62
Figura 20 – Dois ciclos homólogos que representam o buraco. Note que o ciclo que representa o buraco não é ótimo no número de simplexos. O ciclo ótimo é o com linhas pontilhadas.	74
Figura 21 – Pontos extraídos de um círculo com ruídos.	85
Figura 22 – Diagramas de persistência do círculo X . Em laranja o diagrama de persistência de dimensão 1, em azul o de dimensão 0. A filtração de Vietoris-Rips foi usada para calcular o complexo simplicial.	86

Figura 23 – Seis imagens de persistência do diagrama de dimensão 1 da Figura 22.	86
Figura 24 – Complexo simplicial associado a X , Z e f do Exemplo 4.8	88
Figura 25 – Exemplo do Mapper sendo aplicado em um círculo com ruído.	89
Figura 26 – Processo de quebra de uma cópia da proteína sobre a superfície de uma célula de levedura.	93
Figura 27 – Pipeline da metodologia utilizada para a predição do score de estabilidade.	95
Figura 28 – Logaritmo da soma acumulada em relação à importância das propriedades dado pelo algoritmo GBoost.	98
Figura 29 – Heatmap das regiões dos diagramas de persistência de dimensão 1 que aparecem nas primeiras 50 propriedades. Eixo x representa nascimento, enquanto que o eixo y representa a persistência.	99
Figura 30 – Heatmap das regiões dos diagramas de persistência de dimensão 2 que aparecem nas primeiras 50 propriedades. Eixo x representa nascimento, enquanto que o eixo y representa a persistência.	99
Figura 31 – Número de ciclos associados para as top 50 propriedades e seus respectivos diagramas de persistência.	99
Figura 32 – Panorama de energia para decoys modeladas em relação à proteína 1T2I.	100
Figura 33 – Soma dos átomos de carbono que compõem os ciclos do 1º diagrama de persistência das decoys da 2QY7.	102
Figura 34 – Soma dos átomos de nitrogênio (esquerda) e oxigênio (direita) que compõe os ciclos do 1º diagrama de persistência das decoys da proteína 1T2I.	103
Figura 35 – Valor do RMSD para cada decoy no top 10. Não existem falsos mínimos para a proteína 1T2I, enquanto isso existem 7 falsos mínimos para a proteína 2NQW.	104
Figura 36 – Proteína 1T2I. RMSD previsto x RMSD verdadeiro para o top 10 dados os regressores treinados em outras proteínas.	105
Figura 37 – Proteína 2NQW. RMSD previsto x RMSD verdadeiro para o top 10 dados os regressores treinados em outras proteínas.	105
Figura 38 – RMSD previsto x RMSD verdadeiro para o top 10 decoys da proteína 1T2I dados os regressores com a melhor acurácia binária no conjunto de validação.	107
Figura 39 – RMSD previsto x RMSD verdadeiro para o top 10 decoys da proteína 2NQW dados os regressores com a melhor acurácia binária no conjunto de validação.	107

LISTA DE ALGORITMOS

Algoritmo 1 – Redução da matriz bordo ∂	34
Algoritmo 2 – Procedimento de otimização dos geradores.	77
Algoritmo 3 – Algoritmo para calcular o diagrama de persistência de uma filtração.	78
Algoritmo 4 – Algoritmo para calcular o diagrama de persistência de uma filtração e os ciclos ótimos.	79
Algoritmo 5 – Procedimento para encontrar o ciclo ótimo.	80

LISTA DE CÓDIGOS-FONTE

algoritmos/std_alg.jl	115
---	-----

LISTA DE TABELAS

Tabela 1 – Resultados do algoritmo treinado pelos autores de (ROCKLIN <i>et al.</i> , 2017).	94
Tabela 2 – Resultados para variância igual a 0,1.	95
Tabela 3 – Resultados para variância igual a 0,3.	95
Tabela 4 – Resultados para variância igual a 0,5.	96
Tabela 5 – Resultados para variância igual a 0,7.	96
Tabela 6 – Resultados dos modelos treinados utilizando apenas as imagens de persistência com variância 0.1	97
Tabela 7 – Resultados dos modelos treinados utilizando apenas as imagens de persistência com variância 0.3	97
Tabela 8 – Resultados dos modelos treinados utilizando apenas as imagens de persistência com variância 0.5	97
Tabela 9 – Resultados dos modelos treinados utilizando apenas as imagens de persistência com variância 0.7	97
Tabela 10 – Rank mostrando as top 5 decoys em relação a 1T2I.	101
Tabela 11 – Alguns dos testes feitos para obter as imagens de persistência e usa-las para o treinamento.	105
Tabela 12 – Melhores parâmetros para cada métrica para os regressores treinados nas imagens de persistência.	106
Tabela 13 – Melhores regressores treinados com as propriedades das proteínas	106

SUMÁRIO

1	INTRODUÇÃO	21
2	INTRODUÇÃO A HOMOLOGIA PERSISTENTE	23
2.1	Filtrações	24
2.1.1	<i>Complexo de Čech</i>	27
2.1.2	<i>Complexo de Vietoris-Rips</i>	29
2.1.3	<i>Complexo Alpha</i>	30
2.2	A matriz de bordo ∂	32
2.3	Redução da matriz ∂	34
2.4	Calculando a homologia persistente	35
3	MÓDULOS DE PERSISTÊNCIA	39
3.1	Módulos de persistência e decomposições	39
3.1.1	<i>Índices e posets</i>	41
3.1.2	<i>Categoria de módulos</i>	42
3.1.3	<i>Módulos Intervalares</i>	42
3.1.4	<i>Decomposição em módulos intervalares</i>	44
3.1.5	<i>Cálculos com quivers</i>	46
3.2	Medidas retangulares	49
3.2.1	<i>A medida de persistência</i>	49
3.2.2	<i>r-medidas abstratas</i>	52
3.2.3	<i>Equivalência de medidas e diagramas</i>	53
3.3	Comportamento de módulos e exemplos	57
3.4	<i>Intercalação</i>	59
3.4.1	<i>Homomorfismos e módulos de persistência</i>	59
3.4.2	<i>O lema de interpolação</i>	62
3.5	O teorema de isometria	63
3.5.1	<i>A distância de interlçamento</i>	63
3.5.2	<i>A distância bottleneck</i>	65
3.5.3	<i>O teorema de isometria</i>	68
3.5.4	<i>A volta do teorema de estabilidade</i>	69
3.5.5	<i>O teorema de estabilidade</i>	72
4	GERADORES ÓTIMOS E OUTROS CONCEITOS	73

4.1	Geradores ótimos	73
4.1.1	Único Gerador	74
4.1.2	Múltiplos geradores	76
4.1.3	Geradores ótimos em homologia persistente	77
4.2	Vetorização do diagrama de persistência	81
4.2.1	Estabilidade da Imagem de Persistência	82
4.2.2	Exemplos de Imagens de Persistência	85
4.3	Mapper	87
4.3.1	Mapper topológico	87
4.3.2	Mapper Estatístico	87
4.3.3	Funções filtro	89
4.3.3.1	Ecentricidade	89
4.3.3.2	Densidade	90
4.3.4	Implementação	90
5	ESTABILIDADE DE PROTEÍNAS	91
5.1	Estudando a estabilidade - Proteínas I	92
5.1.1	A estabilidade da proteína	92
5.1.2	Prevendo a estabilidade	93
5.1.3	Metodologia	94
5.1.4	Resultados e análises	95
5.1.4.1	Primeiro método	95
5.1.4.2	Segundo método	96
5.2	Analisando a energia total - Proteínas II	100
5.2.1	Análise de falso mínimos	100
5.2.1.1	VAE e ciclos ótimos	101
5.2.1.2	Resultados	102
5.2.1.3	Parâmetros	103
5.2.2	Prevendo o RMSD	103
5.2.2.1	Resultados	104
6	CONCLUSÃO	109
	REFERÊNCIAS	111
	APÊNDICE A ALGORITMO STANDARD E FUNÇÕES AUXILIARES	115

INTRODUÇÃO

A topologia é a área da matemática que estuda formas e geometrias dos objetos através de funções contínuas. Um exemplo clássico é o da rosquinha e a caneca. Esses dois objetos tão distintos na vida real se tornam iguais do ponto de vista topológico, já que existem transformações que os torcem e entortam até o outro, mas não cortam e colam. Para um estudo mais sistemático e algébrico, a topologia algébrica foi desenvolvida. Com ela associamos grupos ou espaços vetoriais a espaços topológicos e assim podemos extrair informações como componentes conexas, buracos e cavidades, além de outros buracos n -dimensionais.

O estudo topológico dos objetos matemáticos com a álgebra é estudado desde o início do século XX com Poincaré ([POINCARÉ, 1895](#)). Mas apenas recentemente começou a se desenvolver algoritmos, devido ao advento dos dados e computadores para cálculos eficientes. Essa nova área, chamada de análise topológica de dados, possui várias ferramentas, como a homologia persistente ([EDELSBRUNNER, 2010](#)) e o mapper ([SINGH; MEMOLI; CARLSSON, 2007](#)).

A homologia persistente estuda os invariantes topológicos de dados através da topologia algébrica. Para cada conjunto de dados, podemos construir um complexo simplicial, objeto combinatório da matemática que codifica informações do conjunto geometricamente. Com esse complexo, podemos construir uma filtração e então obter informações dos buracos n -dimensionais do conjunto de dados através dos grupos de homologia, que geralmente são espaços vetoriais, e que por fim são codificados no diagrama de persistência. Essa técnica pode ser considerada uma ferramenta de redução de dimensão, assim como para a extração de propriedades geométricas de dados em baixa dimensão, como no plano ou espaço 3D.

Devido a natureza da ferramenta, nesta dissertação mostramos algumas aplicações em biologia. O problema de enovelamento de proteína pode ser resumido com a seguinte

expressão:

Sequência de aminoácidos \rightarrow Estrutura \rightarrow Função.

A questão fundamental é como encontrar uma estrutura estável com uma específica função a partir de uma sequência de aminoácidos específica.(DILL *et al.*, 2008) Vários métodos foram desenvolvidos envolvendo propriedades físicas e estatísticas das proteínas para prever a estrutura de uma molécula dada uma sequência de aminoácidos. O mais conhecido é o software Rosetta, desenvolvido por um grupo da Universidade de Washington em 1997.(SIMONS *et al.*, 1997)

Este software tem tido muito sucesso na predição de proteínas pequenas, mas não é perfeito. Várias vezes ele apresenta proteínas não estáveis, o que em desenvolvimento de proteínas em larga escala pode acarretar em custos mais altos quando testes são realizados no laboratório. Em (ROCKLIN *et al.*, 2017) apenas algumas proteínas são escolhidas para a fase de testes em laboratório após seu modelamento no Rosetta. O método de escolha é dado por um algoritmo de machine learning, que prevê a estabilidade das moléculas baseado em 110 propriedades por proteína.

Como aplicação nós desenvolvemos um novo método para a escolha de proteínas para a fase de testes que não apenas prevê a estabilidade em nível do estado da arte, mas também nos dá novas informações. Para cada proteína, temos alguns diagramas de persistência. Cada ponto desse diagrama de persistência está relacionado com alguma cadeia na classe de homologia que pode ser visualizada na molécula. Temos então essa nova informação geométrica que pode auxiliar no design de novas proteínas.

Uma outra aplicação é o desenvolvimento de uma função de energia para o software Rosetta baseado em outras proteínas. Essa nova função de energia foi treinada para prever a distância da molécula simulada em relação a original. Diversas proteínas do PDB (Protein Data Bank) foram usadas no treinamento.

A dissertação está dividida nos seguintes capítulos. No Capítulo 2 temos uma introdução computacional de homologia persistente, apresentando as principais filtrações e como obter as informações topológicas através do algoritmo de redução. No Capítulo 3 temos a fundamentação teórica de homologia persistente, através de ideias inspiradas na teoria de medida e categorias. O Capítulo 4 contém algoritmos decorrentes da homologia persistente para a utilização no aprendizado de máquinas e também há uma introdução do mapper. Já no Capítulo 5 apresentamos duas aplicações voltadas ao desenvolvimento de proteínas, mostrando resultados novos e similares aos do estado da arte. E o Capítulo 6 fecha a dissertação com algumas conclusões.

INTRODUÇÃO A HOMOLOGIA PERSISTENTE

A topologia sempre foi vista como uma área de abstração da matemática, sem espaço para aplicações. Ela é usada para o estudo de diversos espaços em sua forma abstrata, auxiliando matemáticos em diversas demonstrações de teoremas e dando uma base fundamental para grande parte da teoria matemática usada no dia a dia ([POINCARÉ, 1895](#)).

Certas propriedades dos espaços topológicos são estudadas através da topologia algébrica, dando algumas informações, como o número de componentes conexas por caminhos de um espaço e buracos. A princípio esta é uma área altamente abstrata da matemática, nos últimos anos esta visão foi mudando, com o desenvolvimento da Homologia Persistente e Análise Topológica de Dados.

Um conjunto de dados, geralmente um subconjunto finito de algum espaço métrico, pode ser estudado através da homologia persistente e assim obtemos informações topológicas do objeto em estudo.

O pipeline da análise topológica de dados pode ser dividido nos seguintes passos:

1. A entrada do algoritmo pode ser um conjunto de pontos ou alguma matriz de distância/similaridade do conjunto de dados.
2. A construção de um objeto combinatorial em cima do conjunto de dados ou da matriz de distância. Geralmente uma filtração ou um complexo simplicial.
3. A partir da filtração ou do complexo simplicial é possível extrair informações topológicas e geométricas do conjunto de dados, por exemplo o número de componentes conexas, como um algoritmo de Clustering.

4. Por fim a interpretação dos dados obtidos e possível pós processamento para a utilização em outros algoritmos, como os de classificação ou regressão.

Neste capítulo descrevemos de forma ingênua a homologia persistente, começando com filtrações, passando pelos espaços vetoriais associados aos complexos simpliciais e chegando ao algoritmo de homologia persistente. Mostraremos também como interpretar os resultados obtidos. A Figura 1 mostra os passos para utilizar esta ferramenta em um conjunto de dados.

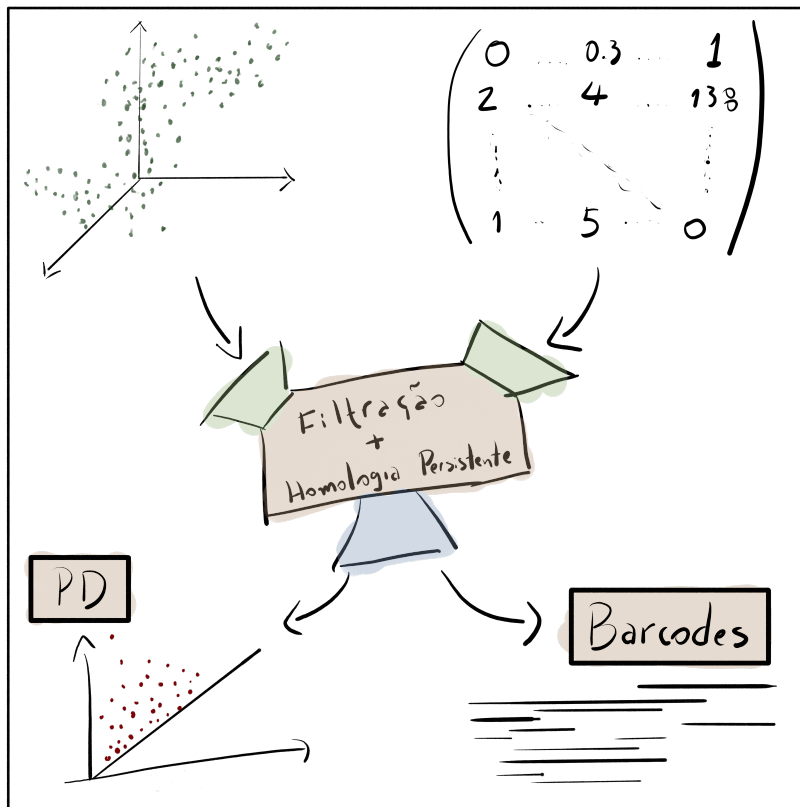


Figura 1 – Representação do pipeline para a utilização da homologia persistente com um conjunto de dados.

Fonte: Elaborada pelo autor.

2.1 Filtrações

A filtração de um conjunto de dados é o primeiro passo na nossa sequência apresentada na Figura 1. Dado um conjunto de dados precisamos construir um objeto combinatorial de forma que possa ser analisado do ponto de vista da topologia assim como computacionalmente. A filtração é este objeto que captura as mudanças do conjunto dada uma escala.

Algumas definições se fazem necessárias para entendermos o que é a filtração e qual o seu papel na análise topológica de dados. Começamos definindo um simplexo, primeiro objeto combinatorial que é a base da filtração.

Definição 2.1. Sejam $v_0, v_1, \dots, v_k \in \mathbb{R}^n$ linearmente afins, ou seja $\{v_1 - v_0, \dots, v_k - v_0\}$ é um conjunto linearmente independente. O k -simplexo definido pelos pontos acima, chamados de vértices, é a envoltória convexa, definida na abaixo.

$$\left\{ \sum_{i=0}^k \lambda_i v_i \mid \sum_{i=0}^k \lambda_i = 1 \text{ e } \lambda_i \geq 0, \forall i \right\}. \quad (2.1)$$

Denotamos o k -simplexo por $\langle v_0, \dots, v_k \rangle$.

Note que para $k = 0$, temos um único vértice. Para $k = 1$, temos uma reta, já para $k = 2$ temos um triângulo preenchido. E no caso $k = 3$, um tetraedro. Os simplexos podem ser vistos na Figura 2. Além disso, dizemos que a dimensão do k -simplexo é k . A envoltória convexa de qualquer subconjunto dos vértices de um simplexo σ é chamado de face de σ .

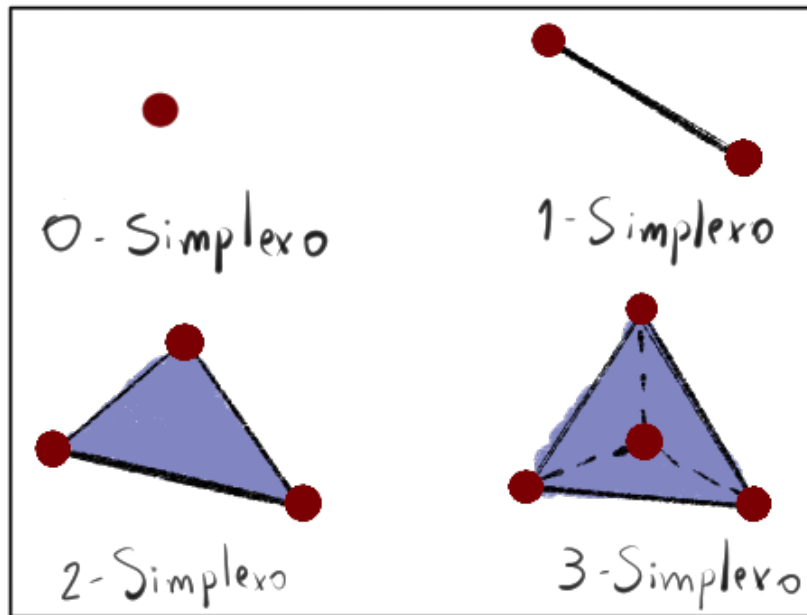


Figura 2 – Exemplos de k -simplexos para $k \in \{0, 1, 2, 3\}$.

Fonte: Elaborada pelo autor.

Tendo definido os k -simplexos, podemos definir o complexo simplicial.

Definição 2.2. Um complexo simplicial K é uma coleção de simplexos satisfazendo as seguintes relações:

- Dado $\sigma \in K$, temos que para toda face $\tau \subset \sigma$ vale $\tau \in K$.
- A interseção de dois simplexos é face de ambos os simplexos, em outras palavras, $\sigma, \tau \in K$ implica que $\sigma \cap \tau \subset \sigma$ e $\sigma \cap \tau \subset \tau$.

Nessa definição utilizamos o símbolo \subset para indicar que uma face. Usaremos esse símbolo com essa denotação quando falarmos sobre simplexes e faces. A segunda condição é necessária para evitar casos patológicos como mostrado na Figura 3. Dizemos que a dimensão do complexo simplicial K é a maior dimensão dentre os simplexes em K . Podemos definir agora a filtração de um complexo simplicial.

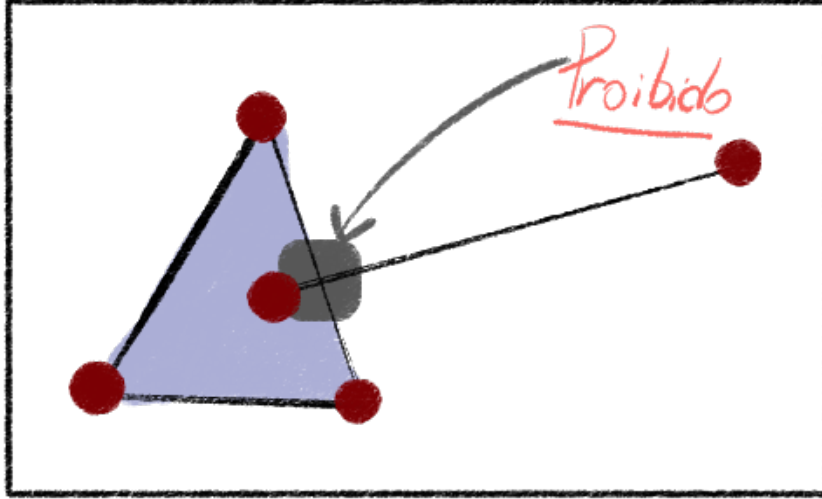


Figura 3 – Exemplo em que a interseção de dois simplexes não é um simplexo.

Fonte: Elaborada pelo autor.

Definição 2.3. Seja K um complexo simplicial. Definimos uma filtração de K sendo uma sequência de subconjuntos $K_i \subset K$, com $i \in \{1, \dots, n\}$, de tal forma que K_i é um complexo simplicial para todo i e vale que

$$K_1 \subset \dots \subset K_{n-1} \subset K_n = K.$$

Na Figura 4 temos um exemplo de filtração para um complexo simplicial.

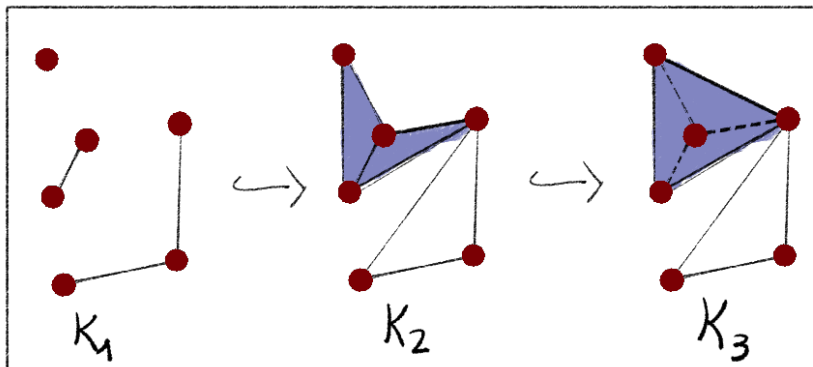


Figura 4 – Exemplo de filtração para um complexo simplicial K .

Fonte: Elaborada pelo autor.

2.1.1 Complexo de Čech

Para construir complexos simpliciais a partir dos dados, precisamos abstrair a noção de um complexo simplicial. Na definição dada anteriormente, temos uma representação geométrica do que é um simplexo, mas podemos abstrair tal noção dando origem aos *complexos simpliciais abstratos*. As definições para os complexos definidos nesta seção e nas próximas foram retiradas de (EDELBRUNNER, 2010).

Definição 2.4. Seja X um conjunto finito com pontos quaisquer. Seja F um conjunto de subconjuntos não-vazios de X . Dizemos que F é um complexo simplicial abstrato de X se a seguinte condição é satisfeita.

- Se para todo $\sigma \in F$, temos que todo subconjunto $\sigma' \subset \sigma$ está em F também.

Cada elemento $\sigma \in F$ é chamado de simplexo. Denotamos um k -simplexo σ por $\langle x_{i_0}, \dots, x_{i_k} \rangle$, onde x_{i_j} são elementos de X .

Exemplo 2.5. Seja $X = \{a, b, c\}$ e considere $F = \{\{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}\}$. Precisamos mostrar que F é um complexo simplicial abstrato. Seja $\sigma = \{a, c\}$. Note que seus subconjuntos são $\{a\}$ e $\{c\}$, além disso ambos pertencem a F . De forma análoga, mostramos que para qualquer outro simplexo, suas faces (subconjuntos) estão em F .

Podemos realizar os complexos simpliciais abstratos geometricamente, ou seja, apesar de trabalharmos com conjuntos de elementos quaisquer, podemos incluir esses complexos em algum \mathbb{R}^n e assim visualiza-los. Para obtermos o complexo simplicial *geométrico*, associamos a cada simplexo abstrato σ um simplexo geométrico. Por exemplo, se adotarmos o complexo simplicial abstrato F acima mostrado, teríamos que sua realização geométrica seria um triângulo sem preenchimento, como é mostrado na Figura 5.

Observe que se o nosso conjunto X for um subconjunto finito de \mathbb{R}^d , podemos ter simplexos de dimensão maiores do que d , ou seja, não podem ser realizados (ou visualizados) em \mathbb{R}^d necessariamente. Um exemplo dessa situação pode ser visto no complexo simplicial final da Figura 4, considerando que os pontos vermelhos são a realização geométrica dos pontos de X , onde X é um subconjunto do \mathbb{R}^2 .

Essa é uma grande diferença entre os complexos simpliciais geométricos e abstratos. Uma vez tendo definido os complexos simpliciais abstratos, podemos definir o *complexo de Čech*.

Definição 2.6. Seja X um conjunto de pontos $\{x_1, \dots, x_n\}$ em \mathbb{R}^d . O complexo de Čech de X para um valor real $r > 0$ é o conjunto $C^r(X)$, onde $\sigma = \langle x_{i_1}, \dots, x_{i_k} \rangle \in C^r(X)$ se, e somente se vale a seguinte condição

$$\bigcap_{j=1}^k B(x_{i_j}, r) \neq \emptyset.$$

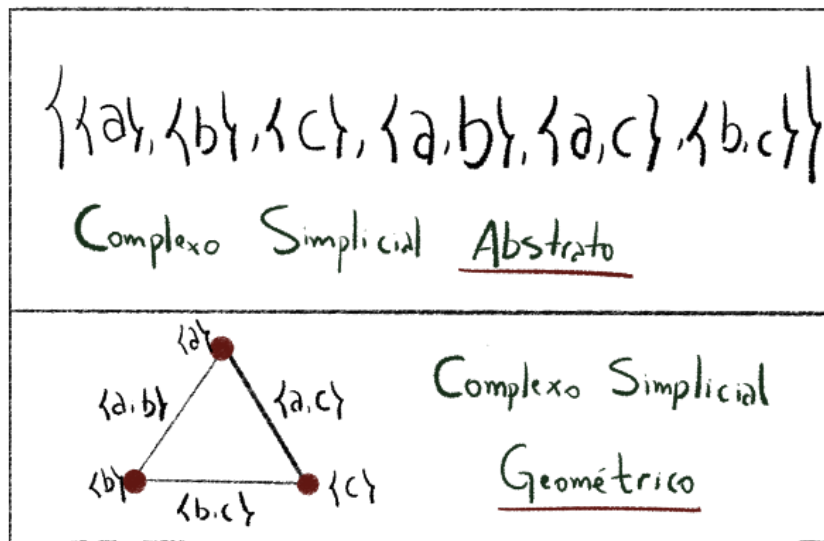


Figura 5 – Exemplo de um complexo simplicial abstrato e sua realização geométrica

Fonte: Elaborada pelo autor.

A definição acima nos diz que quando temos k pontos cujas bolas de raio r centradas neles se intersectam, adicionamos um k simplexo no complexo simplicial abstrato, o que seria apenas o conjunto desses pontos. Geometricamente falando, se duas bolas se intersectam, adicionamos uma aresta. Se três bolas se intersectam, adicionamos um triângulo preenchido, e assim por diante. Na Figura 6 temos um exemplo do complexo simplicial de Čech.

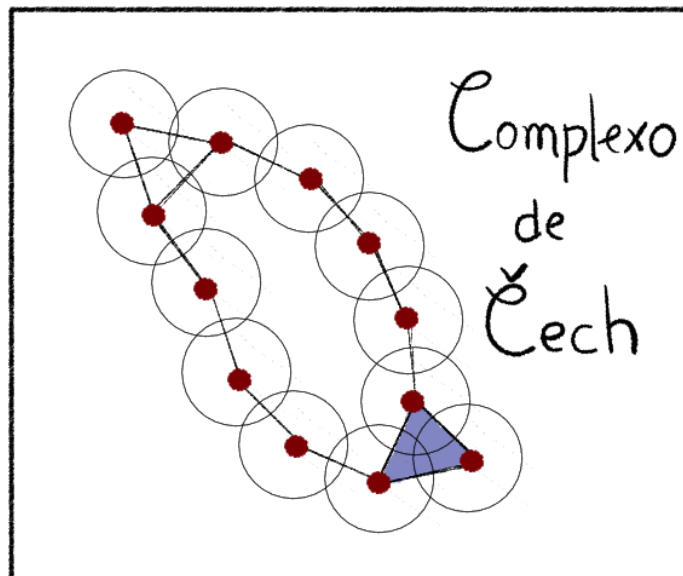


Figura 6 – Exemplo de um complexo de Čech para um raio r fixado.

Fonte: Elaborada pelo autor.

Da mesma forma que definimos a filtração para um complexo simplicial geométrico, o mesmo vale para o caso abstrato.

2.1.2 Complexo de Vietoris-Rips

O complexo de Vietoris-Rips possui uma construção similar ao complexo de Čech, porém computacionalmente é um método mais barato, já que analisa apenas distância entre pontos dois a dois.

Definição 2.7. Seja X um conjunto de pontos $\{x_1, \dots, x_n\}$ em \mathbb{R}^d . O complexo de Vietoris-Rips de X para um valor real $r > 0$ é o conjunto $V^r(X)$, onde o simplexo $\sigma = \langle x_{i_1}, \dots, x_{i_k} \rangle \in V^r(X)$ se, e somente se vale a seguinte condição

$$d(x_{i_k}, x_{i_j}) < r \quad \forall j, l \in 1, \dots, k.$$

A Figura 7 é um exemplo do complexo de Vietoris-Rips. Uma das diferenças que a construção dos dois complexos já definidos nos dá é que no caso do complexo de Čech não temos triângulos preenchidos, e isso ocorre para Vietoris-Rips.

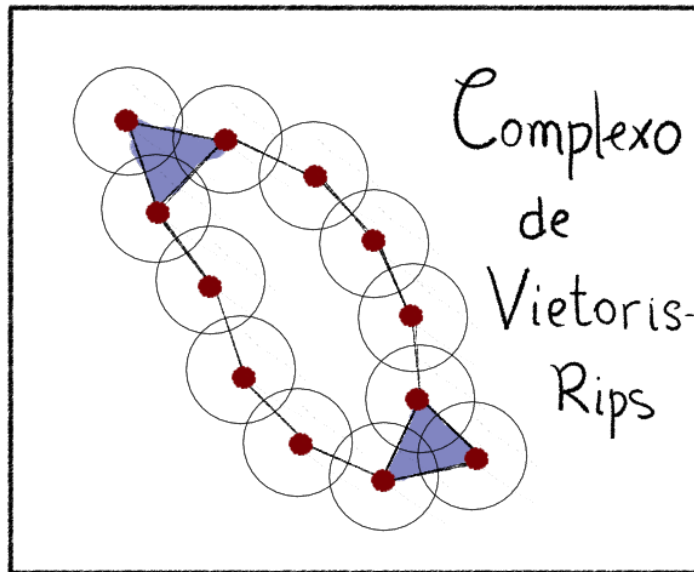


Figura 7 – Exemplo do complexo de Vietoris-Rips com os mesmos pontos utilizados para a construção na Figura 6.

Fonte: Elaborada pelo autor.

Mesmo com as regras diferentes para a construção de complexos, temos a seguinte relação entre os dois complexos.

$$C^r(X) \subset V^r(X) \subset C^{2r}(X) \quad (2.2)$$

A primeira inclusão segue do fato que se k bolas se intersectam então elas se intersectam dois a dois com a mesma distância. A segunda inclusão segue da desigualdade triangular da métrica sendo usada e o fato que as bolas se intersectam duas a duas.

2.1.3 Complexo Alpha

E como uma terceira opção para a construção de um complexo simplicial através de pontos no \mathbb{R}^n , temos o complexo Alpha. A construção é similar ao complexo de Čech, porém os conjuntos centrados nos pontos são uma interseção de bolas no \mathbb{R}^n com conjuntos convexos especiais, as células de Voronoi. Nesta subseção utilizaremos o \mathbb{R}^n para as definições, porém elas podem ser generalizadas para qualquer espaço métrico.

O diagrama de Voronoi é um tipo especial de decomposição de um espaço métrico, um conjunto que possui uma distância associada a ele. Dado um subconjunto $X \subset \mathbb{R}^n$ finito, onde $X = \{x_1, \dots, x_k\}$, definimos a célula de Voronoi associada ao ponto x_i sendo o seguinte conjunto

$$V_i = \{x \in \mathbb{R}^n \mid d(x_i, x) \leq d(x_j, x), \forall j \in 1, \dots, k\},$$

em que d é a distância euclidiana usual. A Figura 8 mostra um exemplo de diagram de Voronoi para três pontos no \mathbb{R}^2 . Podemos agora definir o complexo simplicial Alpha.

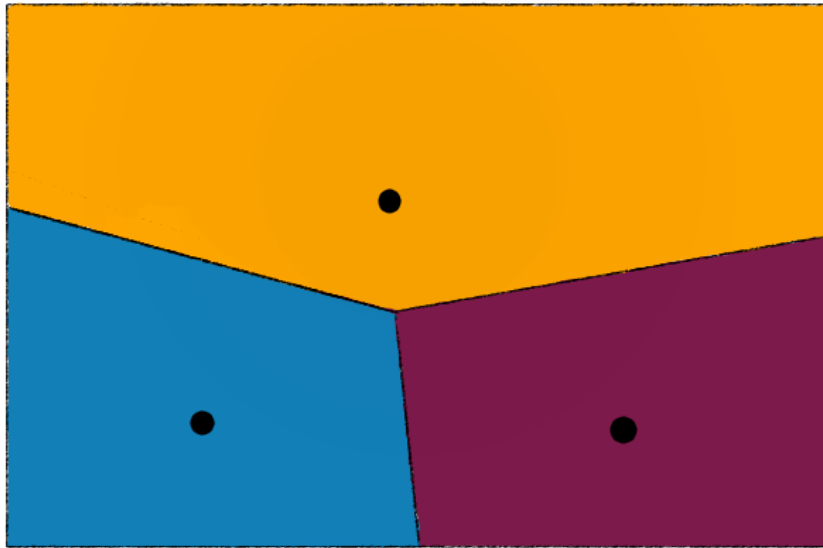


Figura 8 – Diagrama de Voronoi de três pontos no plano.

Fonte: Elaborada pelo autor.

Definição 2.8. Seja X um conjunto de pontos $\{x_1, \dots, x_n\}$ em \mathbb{R}^d . O complexo Alpha de X para um valor real $r > 0$ é o conjunto $A^r(X)$, onde o simplexo $\sigma = \langle x_{i_1}, \dots, x_{i_k} \rangle \in A^r(X)$ se, e somente se vale a seguinte condição

$$\bigcap_{j=1}^k R(x_{i_j}, r) \neq \emptyset,$$

onde $R(x_{ij}, r) = B(x_{ij}, r) \cap V_{ij}$, para todo $j \in \{1, \dots, k\}$.

Na Figura 9 temos o exemplo de um complexo Alpha. É interessante notar que o Alpha é um subcomplexo do complexo de Čech, ou seja, para $r > 0$, $A^r(X) \subset C^r(X)$. Além disso esse complexo herda uma propriedade importante dos diagramas de Voronoi, a realização geométrica no espaço em que os pontos se encontram, isto é, se os pontos em \mathbb{R}^d satisfazem a condição de posição geral, então o complexo simplicial abstrato Alpha pode ser realizado geometricamente no \mathbb{R}^d , ou seja o complexo simplicial geométrico pode ser construído no \mathbb{R}^d ! Isso é fundamental computacionalmente, já que diminui a complexidade dos cálculos e aumenta a velocidade para obtenção do complexo.

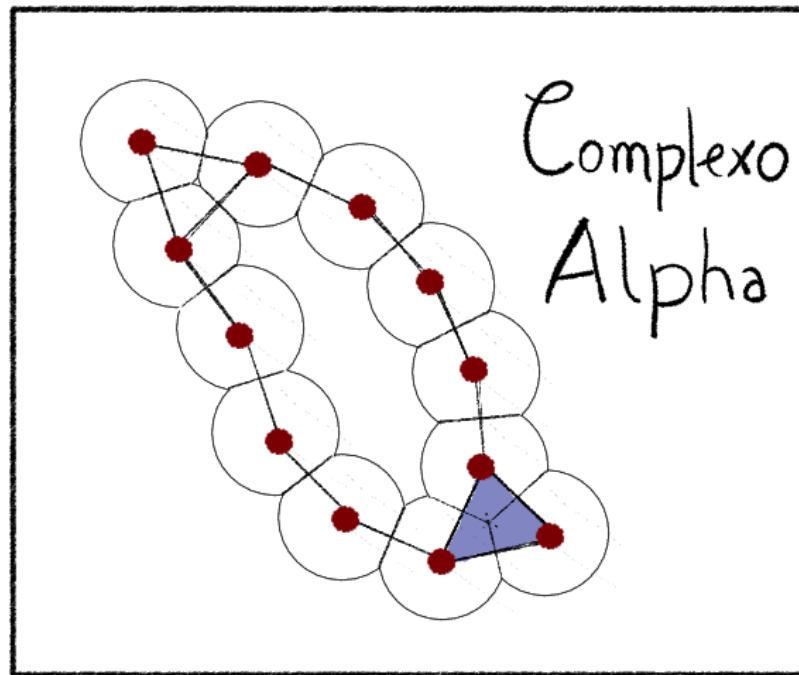


Figura 9 – Complexo Alpha para um conjunto de pontos no plano.

Fonte: Elaborada pelo autor.

Uma variação muito importante do complexo Alpha é a versão com peso. Ao invés de considerar um raio fixo para cada bola ao redor de um ponto, podemos dar um *peso* para cada ponto. Seja $X = \{x_1, \dots, x_n\}$ o nosso conjunto de pontos finito e $\{w_1, \dots, w_n\}$ conjunto de valores maiores ou iguais a zero, que serão os pesos associados a cada ponto. Para cada x_i , ao invés de associar a bola usual do complexo Alpha, associamos a seguinte bola.

$$R_{w_i}(x_i, r) = B(x_i, r + w_i^2) \cap V_i$$

Esse é um complexo muito usado em aplicações biomoleculares, em que o conjunto de pontos são átomos de uma molécula e o peso para cada átomo é o seu respectivo raio de Van der Waals.

2.2 A matriz de bordo ∂

Agora vamos para o terceiro passo descrito na lista anteriormente. Uma vez com os dados, podemos construir uma filtração de um complexo simplicial criado a partir deles que irá capturar diversas informações, como os buracos que um conjunto de dados tem e o quanto eles persistem na nossa filtração.

A ferramenta matemática utilizada para extrair essas informações da filtração são os grupos de homologia. Para uma filtração $K_1 \subset \dots \subset K_m = K$ e um p fixo, a p -ésima homologia persistente de K é o par

$$(\{H_p(K_i)\}_{1 \leq i \leq m}, \{f_{i,j}\}_{1 \leq i \leq j \leq m}), \quad (2.3)$$

em que para todo $i, j \in \{1, \dots, m\}$, $f_{i,j}$ são aplicações lineares entre os espaços vetoriais $H_p(K_i)$ e $H_p(K_j)$. Mais especificamente, os espaços vetoriais $H_p(K_i)$ são grupos de homologia com coeficientes em um corpo. No nosso caso usamos o \mathbb{Z}_2 . Consulte (EDELBRUNNER, 2010) para uma introdução à teoria de homologia nesse contexto.

A homologia persistente dá informações topológicas sobre a filtração do complexo simplicial. Os elementos das bases de cada $H_p(K_i)$ correspondem a ciclos p -dimensionais, podendo ser buracos. Ciclos são os nomes dados aos representantes dos elementos da base do espaço vetorial em questão. No caso $p = 0$, temos que cada elemento da base corresponde à uma componente conexa, $p = 1$ cada elemento corresponde a um buraco. Portanto, considere os elementos da base de $H_p(K_i)$. Para cada um deles, desenhe um ponto em um papel. Para u elemento da base de $H_p(K_i)$, se $f_{i,i+1}(u) = 0$, então desenhe um intervalo que começa no ponto anterior e termina em $i + 1$. Se $f_{i,i+1}(u) = v$, onde v é um elemento da base de $H_p(K_{i+1})$, então desenhe uma reta que liga u ao ponto que representa v no próximo passo da filtração. Dessa forma vamos anotando os ciclos, que são os elementos da base, ao longo da filtração. Na Figura 10 temos um exemplo para uma filtração.

Podemos falar também que $u \in H_p(K_i)$ nasceu no tempo i da filtração se u não é imagem de nenhum elemento de $H_p(K_{i-1})$ sobre $f_{i-1,i}$. Dizemos também que $u \in H_p(K_j)$ morreu em j se j é o menor índice tal que $f_{i,j}(u) = 0$, onde $j > i$. A persistência do ponto u pode ser representada pelo intervalo $[i, j)$. Além disso, se u nasce no tempo i e nunca morre, denotamos o intervalo associado à essa informação como $[i, +\infty)$.

Existem duas formas de visualizar esses intervalos, através dos *barcodes* ou dos *diagramas de persistência* (PD). No barcode desenhamos uma barra do comprimento do intervalo $[i, j)$. Já no diagrama de persistência representamos com um ponto (i, j) no plano. A Figura 10 possui o barcode e o diagrama de persistência para o conjunto de dados associado.

Tendo essa ferramenta, como podemos traduzi-la para o contexto dos dados, e como calcular os pares dos diagramas de persistência? Abaixo segue uma lista dos primeiros

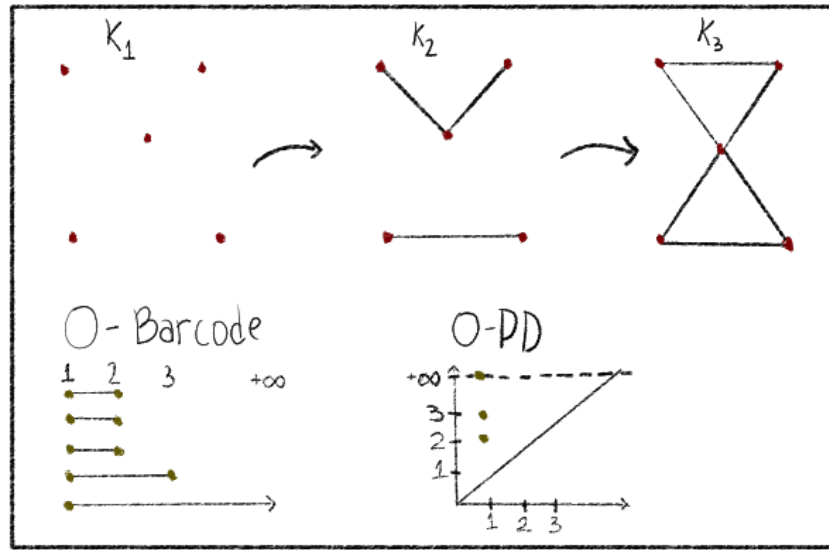


Figura 10 – Exemplo da filtração de um complexo simplicial e o barcode e diagramas de persistência associados.

Fonte: Elaborada pelo autor.

passos que devem ser feitos para a obtenção do diagrama de persistência.

1. Dado um conjunto de dados, determinar alguma filtração;
2. Listar todos os simplexes na filtração;
3. Ordenar os simplexes satisfazendo duas regras:
 - a) A face um simplexo o precede na ordenação;
 - b) Um simplexo no complexo K_i precede os simplexes em K_j , $j > i$, que não pertencem a K_i ;
4. Construir a matriz de bordo.

A matriz de bordo é quem vai armazenar as informações topológicas importantes das quais iremos extrair mais tarde.

Definição 2.9. Seja K um complexo simplicial, $K_1 \subset \dots \subset K_m$ uma filtração e $\sigma_1, \dots, \sigma_n$ uma ordenação dos simplexes de K satisfazendo as regras acima mencionadas. A *matriz de bordo* de K , denotada por ∂ , é uma matriz de tamanho $n \times n$, em que cada entrada tem o seguinte valor

$$\delta(i, j) = \begin{cases} 1, & \text{se o simplexo } \sigma_i \text{ é face de } \sigma_j \text{ e } \dim(\sigma_j) = \dim(\sigma_i) + 1 \\ 0, & \text{caso contrário.} \end{cases}$$

Com a matriz construída, podemos utilizar um método de eliminação de Gauss para a redução da matriz.

2.3 Redução da matriz ∂

O algoritmo que será descrito aqui é conhecido como algoritmo *standard* para a redução da matriz ∂ (EDELSBRUNNER; LETSCHER; ZOMORODIAN, 2000). Estamos trabalhando sobre \mathbb{Z}_2 , ou seja, $1 + 1 = 0$. Durante o processo de redução da matriz será apenas necessário somar colunas.

Dado $j \in \{1, \dots, n\}$, denotamos por $low(j)$ o maior inteiro $i \in \{1, \dots, n\}$ tal que $\delta(i, j) = 1$. Note que $i < j$, pois segundo as regras de construção da matriz de bordo, temos que $\delta(i, j) = 1$ só quando σ_i é face de codimensão 1 de σ_j . Assim temos o Algoritmo 1 para reduzir a matriz de bordo.

Algoritmo 1 – Redução da matriz bordo ∂ .

- 1: Dados os simplexes $\sigma_1, \dots, \sigma_n$ e a matriz de bordo ∂ correspondente.
 - 2: **para** $j \in \{1, \dots, n\}$ **faça**
 - 3: **enquanto** existe i tal que $low(i) = low(j)$ **faça**
 - 4: Some a coluna i à coluna j .
 - 5: **fim enquanto**
 - 6: **fim para**
-

Dizemos que a matriz está reduzida quando $low(j) \neq low(j_0)$ para quaisquer colunas j, j_0 não nulas. Observe que uma coluna j pode ser zerada, dizemos então que $low(j)$ é indefinido. Além disso, a matriz reduzida, denotada por R , é escrita como a multiplicação de duas matrizes.

$$R = \partial \cdot V \tag{2.4}$$

A matriz V é uma matriz triangular superior que acumula a informação dos ciclos. Uma vez com a matriz ∂ reduzida a R , podemos interpretar as colunas de R da seguinte forma.

1. A coluna j é nula. Dizemos que o simplexo σ_j é *positivo*, pois dá vida a um ciclo.
2. A coluna j é não-nula. Seja i tal que $low(j) = i$. Dizemos então que σ_j é um simplexo *negativo*, pois quando ele é adicionado temos a morte de um ciclo. Ainda mais, esse ciclo nasceu com a adição do simplexo σ_i .

A nomenclatura de simplexes positivos e negativos vêm da teoria clássica de homologia que estuda propriedades homológicas de um complexo simplicial ao invés de toda a filtração. Para mais detalhes, consulte (EDELSBRUNNER, 2010).

Agora podemos construir o diagrama de persistência para a filtração dada utilizando a matriz de redução. Denotamos por Dgm_p o p -ésimo diagrama de persistência, com $p \in \{0, \dots, k-1\}$ onde k é a maior dimensão dentre os simplexes σ_i . Cada p representa a dimensão dos grupos de homologia descritos anteriormente. Se $p = 0$, então o diagrama

de persistência nos dirá quais componentes conexas apareceram na filtração e o quão persistente elas são. Para $p = 1$, teremos buracos 1-dimensionais, por exemplo, um círculo vazado tem um buraco. Já um toro, tem dois buracos, um visível e outro por dentro da superfície.

Seja p fixado e σ_j um simplexo de dimensão $p+1$ tal que $\text{low}(j) = i$. Dessa forma, adicionamos o ponto (a_i, a_j) ao multiconjunto Dgm_p , em que a_i e a_j são os menores índices tais que $\sigma_i \in K_{a_i}$ e $\sigma_j \in K_{a_j}$, por exemplo, se σ_i é adicionado na filtração em K_l e σ_j é adicionado em K_q , então $a_i = l$ e $a_j = q$. Se tivermos um simplexo σ_i de dimensão p tal que $\text{low}(i)$ é indefinido, então adicionamos o ponto $(a_i, +\infty)$ à Dgm_p . Observe na Figura 11 os diagramas de persistência de dimensão 0 e 1 da filtração da Figura 4.

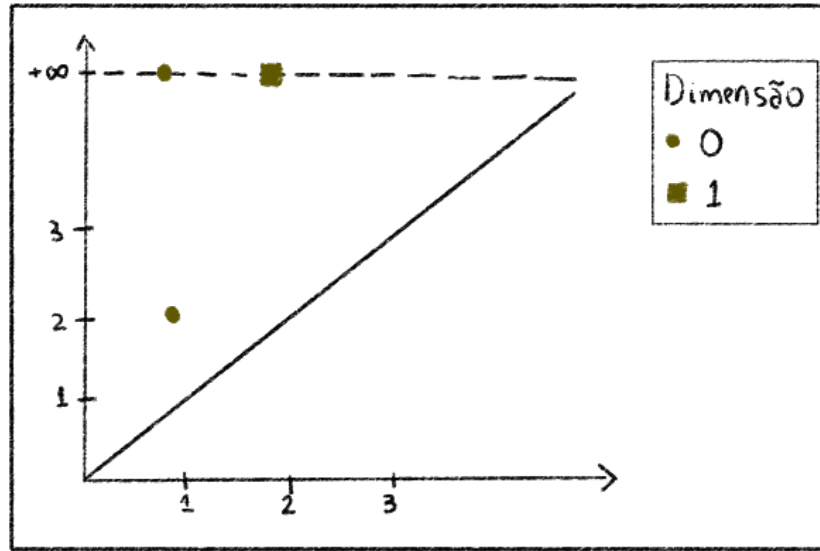


Figura 11 – 0- e 1- Diagramas de Persistência da Figura 4.

Fonte: Elaborada pelo autor.

2.4 Calculando a homologia persistente

Nesta seção iremos calcular a homologia persistente de uma filtração já dada, além disso apresentaremos uma implementação para redução da matriz ∂ em Julia.

Considere a filtração da Figura 12. Observe que temos 4 vértices ($\sigma_1, \dots, \sigma_4$), 5 arestas ($\sigma_5, \dots, \sigma_9$) e 1 triângulo (σ_{10}) ao total, temos 10 simplexos. Diretamente da figura já podemos extrair os diagrams de persistência de dimensão 0 e 1. Note que no primeiro passo da filtração temos 2 componentes conexas, sendo que uma delas morre no terceiro passo e a outra sobrevive até o final. Temos portanto dois intervalos e logo dois pontos no 0-diagrama de persistência: $[1, +\infty)$ e $[1, 3)$.

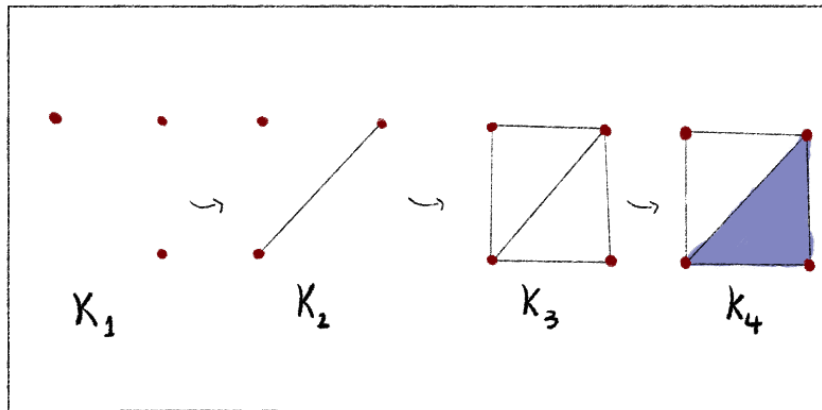


Figura 12 – Filtração de um complexo simplicial.

Fonte: Elaborada pelo autor.

Quando temos um intervalo infinito, geralmente se representa o acima dos índices da filtração no momento em que ele nasceu, como pode ser visto na Figura 13. Já para $p = 1$, constatamos dois intervalos, que representam os dois buracos unidimensionais que surgiram. O primeiro buraco é o que aparece no passo 2 da filtração com a introdução dos simplexes σ_3 e σ_6 e não morre, ou seja, se mantém até o final da filtração, enquanto o segundo buraco surge no passo 3 da filtração com a introdução dos simplexes σ_8 e σ_9 e morre no passo 4 com o nascimento do triângulo σ_{10} . Logo, nossos intervalos são $[2, +\infty)$ e $[3, 4)$.

Sendo assim, podemos construir os dois diagramas de persistência, que podem ser vistos na Figura 13.

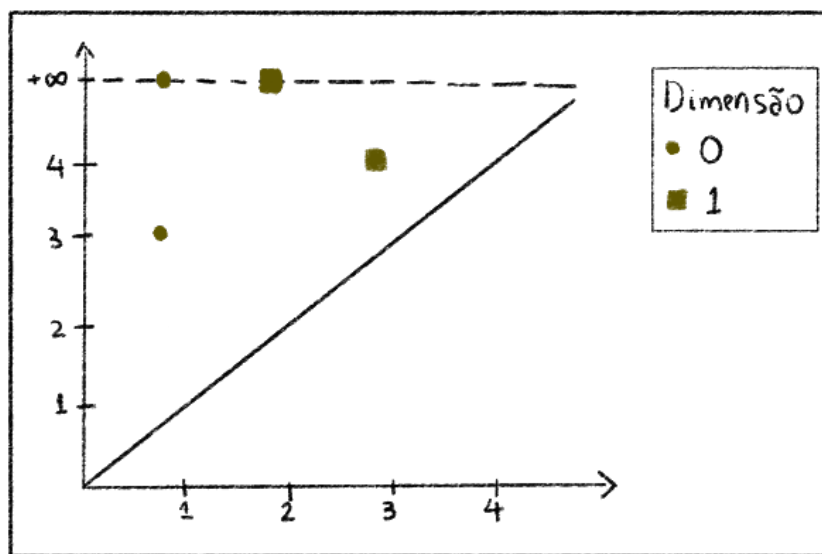


Figura 13 – 0- e 1- diagramas de persistência da filtração mostrada na Figura 12.

Fonte: Elaborada pelo autor.

Agora que calculamos intuitivamente os diagramas de persistência, vamos construir a matriz de bordo ∂ da filtração mostrada na Figura 12 e utilizar implementações no *Julia* para verificar os resultados. A matriz de bordo pode ser visualizada abaixo, note que as regras para construção da matriz são satisfeitas.

$$\partial = \begin{matrix} & \sigma_1 & \sigma_2 & \sigma_3 & \sigma_4 & \sigma_5 & \sigma_6 & \sigma_7 & \sigma_8 & \sigma_9 & \sigma_{10} \\ \begin{matrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \\ \sigma_7 \\ \sigma_8 \\ \sigma_9 \\ \sigma_{10} \end{matrix} & \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix} \quad (2.5)$$

Para reduzir vamos realizar as seguintes operações:

1. somar a coluna 7 com a coluna 6,
2. somar a coluna 7 com a coluna 5, assim zerando a coluna 7,
3. somar a coluna 9 com a coluna 6,
4. somar a coluna 9 com a coluna 8,
5. somar a coluna 9 com a coluna 5, assim zerando a coluna 9.

Note então que após esses passos a matriz ∂ está reduzida e com a seguinte forma.

$$R = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (2.6)$$

Vamos interpretar a matriz agora, para isso temos que parear os simplexos. Utilizando as regras de pareamento descritas anteriormente, temos os pares:

- σ_5 com σ_2 ,
- σ_6 com σ_4 ,
- σ_8 com σ_3 ,
- σ_{10} com σ_9 .

Além disso, existem simplexos que não foram pareados, como os simplexos σ_1 e σ_7 . Note que eles representam o nascimento de p -ciclos que não morrem ao longo da filtração, ou seja, σ_1 corresponde à componente conexa que nasce no primeiro passo da filtração e não morre e σ_7 corresponde ao buraco que nasce no segundo passo e não morre até o final da filtração. Portanto, temos os seguintes intervalos:

- $[a_2, a_5) = [1, 1)$, que não seria adicionado ao diagrama,
- $[a_4, a_6) = [2, 2)$, que não seria adicionado ao diagrama,
- $[a_3, a_8) = [1, 3)$,
- $[a_9, a_{10}) = [3, 4)$,
- $[a_1, +\infty) = [1, +\infty)$,
- $[a_7, +\infty) = [2, +\infty)$,

Logo, $Dgm_0 = \{(1, 3), (1, +\infty)\} \cup \Delta$ e $Dgm_1 = \{(3, 4), (2, +\infty)\} \cup \Delta$, onde $\Delta = \{(x, x) \mid x \in \mathbb{R}^+\}$. Note que obtemos o mesmo resultado, como esperado! O algoritmo *standard* está implementado e pode ser visto no Apêndice [A](#).

MÓDULOS DE PERSISTÊNCIA

A homologia persistente teve seu início em uma intersecção entre as ciências da computação e a matemática. Os primeiros artigos mostravam algoritmos sobre espaços topológicos simples, como esferas (EDELSBRUNNER; LETSCHER; ZOMORODIAN, 2000). No entanto, a teoria foi se desenvolvendo ao longo dos anos ao ponto em que as linguagens utilizadas para tratar da homologia persistente é a teoria de categorias conjuntamente com a teoria de representações (CHAZAL *et al.*, 2016).

Neste capítulo tratamos do desenvolvimento da homologia persistente sob a luz dessas linguagens. Na primeira seção definimos o que são os módulos de persistência e suas relações com os diagramas de persistência. Na segunda seção descrevemos a medida retangular, usada para abstrair o conceito de diagrama de persistência e poder estudar o quão *tame* ele o é. Apresentamos na terceira seção alguns exemplos do comportamento dos módulos de persistência e exemplos. A quarta seção é fundamental, pois mostramos como comparar dois módulos de persistência, através do *interleaving*. E finalmente, apresentamos a teoria de isometria e mostramos uma das implicações com a teoria desenvolvida neste capítulo.

3.1 Módulos de persistência e decomposições

Nesta seção iremos definir os módulos de persistência, apresentar teoremas de decomposição dos módulos e introduzir a notação de quiver, que será utilizada para as próximas seções e demonstrações de outros resultados.

Fixaremos aqui a notação \mathbf{k} como corpo para todos os espaços vetoriais apresentados neste texto, sem especificar necessariamente qual é o corpo.

Definição 3.1. Um módulo de persistência \mathfrak{M} sobre os números reais \mathbb{R} é uma família

indexada sobre \mathbb{R} de espaços vetoriais

$$(V_t \mid t \in \mathbb{R}),$$

e uma família de aplicações lineares duplamente indexadas

$$(v_t^s: V_s \rightarrow V_t \mid s \leq t)$$

que satisfazem a seguinte relação de composição

$$v_t^s \circ v_s^r = v_t^r,$$

em que a função v_r^r é considerada a função identidade.

O módulo de persistência pode ser visto como um funtor entre a categoria dos números reais com o morfismo $s \rightarrow t$, em que $s \leq t$ e a categoria de espaços vetoriais.

Vamos dar um exemplo de módulo de persistência que se encontra no contexto de análise topológica de dados. Seja X um espaço vetorial e $f: X \rightarrow \mathbb{R}$ uma função, não necessariamente contínua e considere os conjuntos de nível

$$X^t = (X, f)^t = \{x \in X \mid f(x) \leq t\}.$$

Temos uma sequência de conjuntos encaixados, X^t com $t \in \mathbb{R}$, ou seja, existe uma função inclusão $\iota_t^s: X^s \hookrightarrow X^t$ que satisfaz trivialmente a lei de composição e existe uma função identidade. Chamamos esta sequência de conjuntos e funções de filtração de subníveis de (X, f) , denotada por \mathfrak{X}_{sub} ou \mathfrak{X}_{sub}^f .

Dada a sequência acima, podemos transforma-la em um módulo de persistência utilizando qualquer funtor da categoria de espaços topológicos para a categoria de espaços vetoriais. Neste caso utilizamos o funtor de homologia $H = H_k(-, \mathbf{k})$ de dimensão k com coeficientes em \mathbf{k} . Assim, podemos definir o seguinte módulo de persistência \mathfrak{V}

$$V_t = H(X^t) \quad v_t^s = H(\iota_t^s): V_s \rightarrow V_t.$$

Podemos também escrever $\mathfrak{V} = H(\mathfrak{X}_{sub})$.

Um exemplo na análise topológica de dados é quando X é um complexo simplicial finito e X^t é um subcomplexo. Devido às propriedades dos complexos, existem finitos valores críticos onde há mudanças em X . Suponha que os valores sejam $a_1 < \dots < a_n$. Então toda a informação do módulo de persistência é dada pela seguinte sequência de espaços vetoriais de dimensão finita

$$H(X^{a_1}) \rightarrow \dots \rightarrow H(X^{a_n}).$$

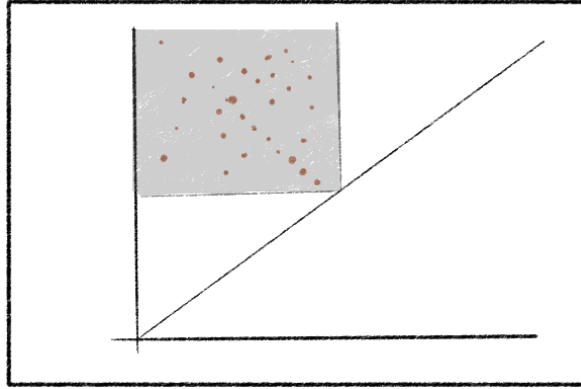


Figura 14 – Exemplo de um diagrama de persistência de um módulo de persistência q -tame com um quadrante em destaque.

Fonte: Elaborada pelo autor.

Neste caso, $H(\mathfrak{X}_{sub})$ admite uma descrição compacta, existe um algoritmo eficiente para o seu cálculo e por último, a descrição é contínua com relação a f , ou seja, é estável sob uma métrica.

A descrição mencionada acima é o diagrama de persistência ou barcode. A estrutura é dada por uma lista de intervalos da forma $[b, d) = [a_i, a_j)$ ou $[a_i, +\infty)$. Cada intervalo representa um ciclo, uma propriedade, que nasce em b e morre em d .

Iremos mostrar aqui que é possível associar um diagrama de persistência para módulos de persistência \mathfrak{V} q -tame. Um módulo de persistência é q -tame se

$$r_t^s = \text{rank}(v_t^s) < \infty \text{ para } s < t.$$

Intuitivamente falando, um módulo é q -tame se para todo quadrante que pegamos com a origem na diagonal existem finitos pontos do diagrama de persistência neste quadrante, como pode ser visto na Figura 14.

3.1.1 Índices e posets

No início desta seção definimos o módulo de persistência com o conjunto de índices sendo os reais. No entanto, é possível definir utilizando quaisquer conjuntos parcialmente ordenados da mesma forma que com os reais. Seja \mathbf{T} um poset, a coleção de espaços vetoriais e aplicações lineares que satisfazem as leis de composição e identidade é chamada de \mathbf{T} -módulo de persistência, ou módulo de persistência sobre \mathbf{T} .

Além disso, podemos restringir o poset \mathbf{T} para um subconjunto $\mathbf{S} \subset \mathbf{T}$ de forma a obter o \mathbf{S} -módulo de persistência, que são os espaços vetoriais e aplicações lineares cujos índices são elementos de \mathbf{S} . Esta é a restrição de \mathfrak{V} em \mathbf{S} e pode ser denotada por $\mathfrak{V}_{\mathbf{S}}$ ou $\mathfrak{V}|_{\mathbf{S}}$.

3.1.2 Categoria de módulos

Com a definição de módulos de persistência sobre um poset \mathbf{T} qualquer, podemos definir homomorfismos entre módulos. Sejam $\mathfrak{U}, \mathfrak{V}$ \mathbf{T} -módulos de persistência. Um homomorfismo Φ entre \mathfrak{U} e \mathfrak{V} é uma família de aplicações lineares $(\phi_t: U_t \rightarrow V_t \mid t \in \mathbf{T})$ tal que o seguinte diagrama comuta para todo $s \leq t$.

$$\begin{array}{ccc} U_s & \xrightarrow{u_t^s} & U_t \\ \phi_s \downarrow & & \downarrow \phi_t \\ V_s & \xrightarrow{v_t^s} & V_t \end{array}$$

A composição de dois homomorfismos Φ, Ψ é dada por cada índice $t \in \mathbf{T}$, ou seja, $\Phi \circ \Psi$ é a coleção de aplicações lineares $(\phi_t \circ \psi_t: U_t \rightarrow W_t \mid t \in \mathbf{T})$, onde Φ é homomorfismo entre \mathfrak{U} e \mathfrak{V} e Ψ entre \mathfrak{V} e \mathfrak{W} . A identidade é definida de forma trivial. Portanto, temos a categoria dos módulos. Definamos os seguintes conjuntos

$$\begin{aligned} \text{Hom}(\mathfrak{U}, \mathfrak{V}) &= \{ \text{homomorfismos } \mathfrak{U} \rightarrow \mathfrak{V} \}, \\ \text{End}(\mathfrak{V}) &= \{ \text{homomorfismos } \mathfrak{V} \rightarrow \mathfrak{V} \}. \end{aligned}$$

3.1.3 Módulos Intervalares

A relação entre os diagramas de persistência e módulos de persistência são fundamentadas pelos módulos intervalares. Eles são a base da teoria de homologia persistente.

Um intervalo em um conjunto totalmente ordenado \mathbf{T} é um subconjunto $J \subset \mathbf{T}$ tal que se $r \in J$ e $t \in J$ tal que $r < s < t$, então $s \in J$. Portanto, para qualquer intervalo $J \subset \mathbf{T}$, o módulo intervalar $\mathfrak{I} = \mathbf{k}^J$ é definido como o \mathbf{T} -módulo de persistência com a seguinte família de espaços vetoriais

$$I_t = \begin{cases} \mathbf{k} & \text{se } t \in J \\ 0 & \text{caso contrário,} \end{cases}$$

e as aplicações lineares

$$\tilde{i}_t^s = \begin{cases} id & \text{se } s, t \in J \\ 0 & \text{caso contrário.} \end{cases}$$

Como mencionado anteriormente, os intervalos seriam as propriedades representadas no diagrama de persistência, ou seja, o módulo intervalar \mathbf{k}^J representa uma propriedade que persiste por todo intervalo J .

Devida a sua importância, módulos intervalares com índices em subconjuntos de \mathbb{R} possuem uma notação especial. Para distinguir os vários casos de intervalos, usamos uma supernotação: $+$ e $-$, a decoração dos pontos. Para intervalos finitos adota-se o seguinte dicionário

$$(p^-, q^-) = [p, q)$$

$$(p^-, q^+) = [p, q]$$

$$(p^+, q^-) = (p, q)$$

$$(p^+, q^+) = (p, q]$$

O dicionário acima vale para $p < q$. No caso em que $p = q$, representamos o intervalo por $(p^-, p^+) = [p, p]$. Para intervalos infinitos, usamos o símbolo $-\infty^+$ e $+\infty^-$ com definição similar à acima e com a adição do seguinte intervalo

$$(-\infty^+, +\infty^-) = (-\infty, +\infty).$$

Quando queremos referenciar um ponto decorado mas não sabemos sua decoração, denotamos por p^* , podendo ser p^- ou p^+ .

Podemos estender os reais para os reais decorados, um conjunto totalmente ordenado com as seguintes relações

$$p^- < p < p^+ < q^- < q < q^+,$$

para todo $p < q$. Definimos o semiplano diagonal superior em \mathbb{R}^2 como

$$\mathcal{H} = \{ (p, q) \mid p \leq q \}.$$

O semiplano diagonal superior $\tilde{\mathcal{H}}$ é a união de \mathcal{H} com os pontos no infinito.

Portanto, um módulo intervalar pode ser representado de diversas formas, visualizados também na Figura 15.

- Como um intervalo na reta real;
- como uma função $\mathcal{H} \rightarrow \{0, 1\}$ definida por $(s, t) \mapsto \text{rank}(i_t^s)$;
- como um ponto $(p, q) \in \mathcal{H}$ e um traço representando a respectiva decoração.

Os traços representando a decoração são dados por

$$(p^-, q^-): \swarrow$$

$$(p^-, q^+): \searrow$$

$$(p^+, q^-): \swarrow$$

$$(p^+, q^+): \searrow$$

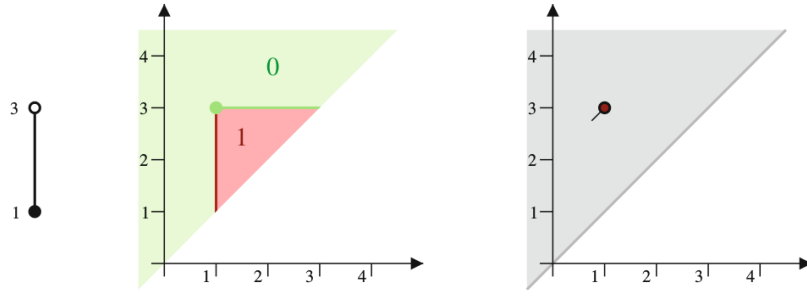


Figura 15 – Representação por intervalo (esquerda), pela função rank (meio) e pelo ponto decorado (direita) do módulo intervalar $\mathbf{k}[1, 3) = \mathbf{k}(1^-, 3^-)$.

Fonte: Chazal *et al.* (2016).

3.1.4 Decomposição em módulos intervalares

Definição 3.2. A soma direta $\mathfrak{W} = \mathfrak{U} \oplus \mathfrak{V}$ de dois módulos de persistência \mathfrak{U} e \mathfrak{V} é definida por

$$W_t = U_t \oplus V_t, \quad w_t^s = u_t^s \oplus v_t^s.$$

Esta definição generaliza-se para somas arbitrárias, tanto finitas como infinitas. Vamos agora definir a indecomponibilidade de um módulo de persistência.

Definição 3.3. Um módulo de persistência \mathfrak{W} é indecomponível se dada uma decomposição $\mathfrak{U} \oplus \mathfrak{V}$, então $\mathfrak{U} = 0$ ou \mathfrak{W} e $\mathfrak{V} = 0$ ou \mathfrak{W} .

Podemos estudar os módulos de persistência através de sua decomposição por módulos intervalares. Dado uma sequência de intervalos $(J_l \mid l \in L)$,

$$\mathfrak{W} \cong \bigoplus_{l \in L} \mathbf{k}^{J_l}.$$

Neste caso, podemos pensar que cada intervalo J_l representa uma propriedade. Esta decomposição acaba sendo muito importante por este motivo. Mas a questão que fica é: quais módulos são decomponíveis em intervalos? E porque decompõe-se em módulos intervalares?

A resposta para a primeira pergunta é o Teorema 3.7. Já para a segunda questão, os módulos intervalares são indecomponíveis, como mostramos na Proposição 3.5.

Proposição 3.4. Seja $\mathfrak{J} = \mathbf{k}_T^J$ um módulo intervalar sobre $\mathbf{T} \subset \mathbb{R}$. Então $\text{End}(\mathfrak{J}) = \mathbf{k}$.

Demonstração. Vamos definir uma função Φ entre $\text{End}(\mathfrak{J})$ e \mathbf{k} que será um isomorfismo de anéis. Seja $\Phi: \mathbf{k} \rightarrow \text{End}(\mathfrak{J})$ definida por

$$\alpha \mapsto \varphi^\alpha$$

onde φ^α é um endomorfismo de \mathfrak{J} tal que $\varphi_t^\alpha: I_t \rightarrow I_t$ e $\varphi_t^\alpha(x) = \alpha x$. É fácil ver que a aplicação é um homomorfismo de anéis. Vamos definir a inversa de Φ . Para isso, note primeiro que qualquer endomorfismo de \mathfrak{J} age como multiplicação por escalar em qualquer I_t não nulo. Precisamos mostrar que dados s, t , temos que o escalar definido é o mesmo para ambos os casos:

$$\begin{aligned}\Psi^{-1}: \mathfrak{J} &\rightarrow \mathbf{k} \\ \varphi &\mapsto \alpha.\end{aligned}$$

Vamos mostrar que a aplicação está bem definida.

Primeiro, pela observação acima, dados s, t tais que $I_s, I_t \neq 0$, temos que vale o seguinte para $\varphi \in \mathfrak{J}$.

$$\begin{aligned}\varphi_s: \mathbf{k} &\rightarrow \mathbf{k} \\ x &\mapsto \alpha x\end{aligned}$$

e

$$\begin{aligned}\varphi_t: \mathbf{k} &\rightarrow \mathbf{k} \\ x &\mapsto \beta x\end{aligned}$$

Precisamos mostrar que $\alpha = \beta$, demonstrando a proposição. Mas isso segue pelo diagrama comutativo dos homomorfismos entre módulos de persistência, como podemos ver na Eq. (3.1), assumindo que $s \leq t$.

$$\begin{array}{ccc} I_s & \xrightarrow{id} & I_t \\ \varphi_s \downarrow & & \downarrow \varphi_t \\ I_s & \xrightarrow{id} & I_t \end{array} \quad (3.1)$$

No caso acima temos a identidade entre I_s e I_t , já que ambos são \mathbf{k} . Logo, segue que $\alpha = \beta$, provando a Proposição. \square

Proposição 3.5. Módulos intervalares são indecomponíveis.

Demonstração. Suponha que exista uma decomposição $\mathfrak{J} = \mathfrak{U} \oplus \mathfrak{V}$. Considere agora as projeções sobre \mathfrak{U} e \mathfrak{V} . Ambas são homomorfismos idempotentes. Mas como $\text{End}(\mathfrak{J})$ é isomorfo a \mathbf{k} e os únicos idempotentes de \mathbf{k} são 0 e 1, segue que \mathfrak{J} é indecomponível. \square

Teorema 3.6. (Krull-Remak-Schmidt-Azumaya) Suponha que um módulo de persistência $\mathbf{T} \subset \mathbb{R}$ pode ser escrito como soma direta de módulos intervalares de duas formas diferentes

$$\mathfrak{V} \cong \bigoplus_{l \in L} \mathbf{k}^{J_l} \cong \bigoplus_{m \in M} \mathbf{k}^{K_m},$$

então existe uma bijeção $\sigma: L \rightarrow M$ tal que $J_l = K_{\sigma(l)}$ para todo $l \in L$.

Demonstração. A demonstração segue do Teorema 1 (AZUMAYA, 1950) com a observação de que se $\mathbf{k}^J \cong \mathbf{k}^L$, então $J = L$. Só é necessário verificar uma condição de localidade para aplicarmos o teorema: se $\psi, \phi \in \text{End}(\mathcal{J})$ não são isomorfismos, então $\psi + \phi$ não é isomorfismo. Mas pela proposição anterior, isso segue do fato que a única aplicação que não é isomorfismo em $\text{End}(\mathcal{J})$ é a aplicação nula. \square

Teorema 3.7. (Gabriel, Auslander, Ringel-Tachikawa, Webb, Crawley-Boevey) Seja \mathfrak{V} um módulo de persistência sobre $\mathbf{T} \subset \mathbb{R}$. Então \mathfrak{V} pode ser decomposto como uma soma direta de módulos intervalares sob as seguintes condições:

- \mathbf{T} é um conjunto finito;
- cada V_t é um espaço vetorial de dimensão finita.

Por outro lado, existe um módulo de persistência sob \mathbb{Z} que não admite uma decomposição intervalar.

Demonstração. Detalhes podem ser vistos em (CHAZAL et al., 2016), página 22, **Teorema 2.8.** \square

Se um módulo de persistência indexado sobre \mathbb{R} pode ser decomposto

$$\mathfrak{V} \cong \bigoplus_{l \in L} (p_l^*, q_l^*),$$

então o diagrama de persistência decorado é definido pelo multiconjunto

$$\text{Dgm}(\mathfrak{V}) = \text{Int}(\mathfrak{V}) = \{ (p_l^*, q_l^*) \mid l \in L \}$$

e o diagrama de persistência não decorado é o multiconjunto

$$\text{dgm}(\mathfrak{V}) = \text{int}(\mathfrak{V}) = \{ (p_l, q_l) \mid l \in L \} - \Delta,$$

onde Δ é a diagonal no plano.

Note que ambos os diagramas definidos não dependem da escolha da decomposição, devido ao Teorema 3.6. Além disso, o diagrama dgm é o diagrama de pontos não decorados e sem a diagonal, sendo encontrado com frequência em exemplos práticos de análise de dados. Para a definição da distância *bottleneck*, acaba sendo mais importante.

3.1.5 Cálculos com quivers

Vamos agora definir uma notação para trabalhar com módulos de persistência sobre conjuntos de índices finitos. Um módulo de persistência \mathfrak{V} sobre um conjunto finito de índices

$$\mathbf{T}: \quad a_1 < \dots < a_n$$

da reta real pode ser visto como um diagrama de n espaços vetoriais e $n - 1$ aplicações lineares, como mostrado abaixo

$$\mathfrak{V}: V_{a_1} \rightarrow \cdots \rightarrow V_{a_n}.$$

O diagrama acima é a representação do seguinte **quiver**:

$$\bullet \longrightarrow \bullet \longrightarrow \cdots \longrightarrow \bullet$$

Vimos que podemos decompor alguns módulos de persistência em módulos intervalares. Para estes podemos representa-los com quivers da seguinte forma.

Exemplo 3.8. Seja $a < b < c$. Existem 6 módulos intervalares diferentes sobre este intervalo.

$$\begin{aligned} \mathbf{k}[a, a] &= \bullet_a \text{---} \circ_b \text{---} \circ_c & \mathbf{k}[a, b] &= \bullet_a \text{---} \bullet_b \text{---} \circ_c & \mathbf{k}[a, c] &= \bullet_a \text{---} \circ_b \text{---} \bullet_c \\ \mathbf{k}[b, b] &= \circ_a \text{---} \bullet_b \text{---} \circ_c & \mathbf{k}[b, c] &= \circ_a \text{---} \bullet_b \text{---} \bullet_c \\ \mathbf{k}[c, c] &= \circ_a \text{---} \circ_b \text{---} \bullet_c \end{aligned}$$

Os círculos \bullet representam uma cópia do espaço vetorial \mathbf{k} unidimensional. O círculo \circ representa o espaço vetorial nulo. A aplicação linear entre dois \bullet é a identidade e qualquer aplicação contendo \circ é a nula.

Esta notação pode ser usada para representar a multiplicidade dos módulos intervalares da decomposição de um módulo de persistência sobre um conjunto de índices finito, essa quando existe. Seja \mathfrak{V} um módulo de persistência sobre o conjunto $\mathbf{T} = \{a_1, \dots, a_n\}$. Definimos a multiplicidade de $[a_i, a_j] \subseteq \mathbf{T}$ em $\mathfrak{V}_{\mathbf{T}}$ como o número de cópias do módulo $\mathbf{k}[a_i, a_j]$ na decomposição de $\mathfrak{V}_{\mathbf{T}}$.

Exemplo 3.9. Seja \mathfrak{V} módulo de persistência sobre $\mathbf{T} = \{a, b, c\}$. Escrevemos

$$\langle [b, c] | \mathfrak{V}_{\mathbf{T}} \rangle \text{ ou } \langle \circ_a \text{---} \bullet_b \text{---} \bullet_c | \mathfrak{V} \rangle$$

para representar a multiplicidade de $\circ_a \text{---} \bullet_b \text{---} \bullet_c$ no seguinte módulo de 3 termos

$$\mathfrak{V}: V_a \rightarrow V_b \rightarrow V_c.$$

Exemplo 3.10. Considere o módulo com dois espaços vetoriais e uma única aplicação linear $\mathfrak{V}: V_a \xrightarrow{\mu} V_b$. Então os invariantes de μ são

$$\begin{aligned} \text{rank}(\mu) &= \langle \bullet_a \text{---} \bullet_b | \mathfrak{V} \rangle, \\ \text{nulidade}(\mu) &= \langle \bullet_a \text{---} \circ_b | \mathfrak{V} \rangle, \\ \text{conulidade}(\mu) &= \langle \circ_a \text{---} \bullet_b | \mathfrak{V} \rangle. \end{aligned}$$

Basta notar que para V_a, V_b espaços de dimensão finita, existe uma decomposição das suas bases

$$e_1, \dots, e_r, f_1, \dots, f_n \text{ e } g_1, \dots, g_r, h_1, \dots, h_m$$

tais que $\mu(e_i) = g_i$, $\mu(f_j) = 0$ para todo i, j . Assim, os espaços vetoriais unidimensionais gerados pelos elementos das bases geram uma decomposição do módulo \mathfrak{V} nos seguintes intervalos

$$\begin{aligned} (\text{span}(e_i) &\rightarrow \text{span}(g_i)) \\ (\text{span}(f_j) &\rightarrow 0) \\ (0 &\rightarrow \text{span}(h_k)) \end{aligned}$$

que são isomorfos respectivamente à $\bullet_a \text{---} \bullet_b$, $\bullet_a \text{---} \circ_b$ e $\circ_a \text{---} \bullet_b$.

Proposição 3.11. Suponha que podemos decompor um módulo de persistência \mathfrak{V} como uma soma direta

$$\mathfrak{V} = \bigoplus_{l \in L} \mathfrak{V}^l,$$

então

$$\langle [a_i, a_j] \mid \mathfrak{V}_{\mathbf{T}} \rangle = \sum_{l \in L} \langle [a_i, a_j] \mid \mathfrak{V}_{\mathbf{T}}^l \rangle$$

para qualquer conjunto de índices $\mathbf{T} = \{a_1, \dots, a_n\}$ e intervalos $[a_i, a_j] \subseteq \mathbf{T}$.

Demonstração. Segue do fato que a decomposição intervalar de $\mathfrak{V}_{\mathbf{T}}$ é a soma direta das decomposições intervalares de cada $\mathfrak{V}_{\mathbf{T}}^l$ para todo $l \in L$. \square

Proposição 3.12. (Princípio da restrição) Sejam \mathbf{S}, \mathbf{T} conjuntos de índices com $\mathbf{S} \subset \mathbf{T}$. Então

$$\langle I \mid \mathfrak{V}_{\mathbf{S}} \rangle = \sum_J \langle J \mid \mathfrak{V}_{\mathbf{T}} \rangle,$$

onde a soma é sobre todos os intervalos $J \subseteq \mathbf{T}$ que se restringe a I sobre \mathbf{S} .

Demonstração. Tome uma decomposição intervalar arbitrária de $\mathfrak{V}_{\mathbf{T}}$. Então uma decomposição intervalar é induzida em $\mathfrak{V}_{\mathbf{S}}$. Note agora que para $I \subseteq \mathbf{S}$, temos diversos intervalos $J \subseteq \mathbf{T}$ tais que $J \cap \mathbf{S} = I$. Devido a linearidade da soma direta, temos que os intervalos de $\mathfrak{V}_{\mathbf{S}}$ do tipo I são os intervalos de $\mathfrak{V}_{\mathbf{T}}$ do tipo J acima. \square

Exemplo 3.13. Seja $a < p < b < q < c$. Então temos os seguintes exemplos para os conjuntos de índices:

- $\mathbf{T}_1 = \{a, b, q, c\}$, $S_1 = \{a, b, c\}$, $I_1 = [b, c]$.

$$\langle \circ_a \text{---} \bullet_b \text{---} \bullet_c \rangle = \langle \circ_a \text{---} \bullet_b \text{---} \bullet_q \text{---} \bullet_c \rangle$$

- $\mathbf{T}_2 = \{a, p, b, c\}$, $S_2 = \{a, b, c\}$, $I_2 = [b, c]$.

$$\langle \circ_a \text{---} \bullet_b \text{---} \bullet_c \rangle = \langle \circ_a \text{---} \circ_p \text{---} \bullet_b \text{---} \bullet_c \rangle + \langle \circ_a \text{---} \bullet_p \text{---} \bullet_b \text{---} \bullet_c \rangle$$

- $\mathbf{T}_3 = \{a, b, q, c\}$, $S_2 = \{a, b, c\}$, $I_2 = [c, c]$.

$$\langle \circ_a \text{---} \circ_b \text{---} \bullet_c \rangle = \langle \circ_a \text{---} \circ_b \text{---} \circ_q \text{---} \bullet_c \rangle + \langle \circ_a \text{---} \circ_b \text{---} \bullet_q \text{---} \bullet_c \rangle$$

3.2 Medidas retangulares

Na seção anterior discutimos módulos de persistência decomponíveis e seus diagramas de persistência, Dgm e dgm. No entanto, nem sempre os módulos são decomponíveis, não sendo possível definir os diagramas de persistência. Para os definir, podemos nos guiar pela seguinte ideia: se soubermos contar o número de pontos do Dgm pertencente em cada retângulo do semiplano, então conhecemos Dgm.

Iremos nos inspirar na teoria da medida para construir uma função que nos dá um valor inteiro ou infinito e que podemos associar um diagrama de persistência com ela. A ideia é que para módulos bem comportados podemos avaliar esta função em retângulos e extrair um conjunto discreto de pontos, que juntamente com sua multiplicidade gerará o diagrama de persistência. No caso que o módulo de persistência for decomponível, as definições são iguais. Caso contrário seguimos com a teoria normalmente.

No resto deste capítulo iremos tratar apenas de medidas finitas no semiplano diagonal sem considerar os pontos no infinito. Os argumentos usados podem ser facilmente estendidos para o caso de medidas infinitas através de um processo de limite e para o semiplano diagonal com os pontos no infinito podemos usar o truque de colocar tudo dentro de um retângulo com a função \arctan .

3.2.1 A medida de persistência

Definição 3.14. Seja \mathfrak{V} um módulo de persistência. Então a medida de persistência de \mathfrak{V} é a função

$$\mu_{\mathfrak{V}}(R) = \langle \circ_a \text{---} \bullet_b \text{---} \bullet_c \text{---} \circ_d \mid \mathfrak{V} \rangle$$

definida no retângulo $R = [a, b] \times [c, d]$ no plano com $a < b \leq c < d$.

Veremos como a medida tem uma relação com módulos decomponíveis. Abaixo um resultado para módulos intervalares.

Proposição 3.15. Seja $\mathfrak{V} = \mathbf{k}^J$, em que $J = (p^*, q^*)$ é um intervalo real. Seja $R = [a, b] \times [c, d]$ tal que $a < b \leq c < d$. Então

$$\mu_{\mathfrak{V}}(R) = \begin{cases} 1 & \text{se } [b, c] \subseteq J \subseteq (a, d) \\ 0 & \text{caso contrário.} \end{cases}$$

Demonstração. Como \mathbf{k}^J restrito a $\{a, b, c, d\}$ é só um intervalo ou o módulo nulo, temos que $\mu_{\mathfrak{V}}(R) \leq 1$, pois apenas teríamos a função identidade, cujo rank é um, ou a função nula, cujo rank é zero.

Como a medida tem valores em $\{0, 1, \dots\} \cup \infty$, vamos averiguar quando acontece $\mu_{\mathfrak{M}}(R) = 1$. Note que $\mu_{\mathfrak{M}}(R) = 1$ quando

$$\mathbf{k}_{\{a,b,c,d\}}^J = \circ_a \text{---} \bullet_b \text{---} \bullet_c \text{---} \circ_d.$$

E esta restrição vale se, e somente se, $b, c \in J$ e $a, d \notin J$. \square

A Proposição 3.15 pode ser representada visualmente. Considere o módulo intervalo como um ponto decorado (p^*, q^*) no semiplano diagonal superior. Então se o ponto estiver no interior do retângulo R , ele será detectado independente da decoração. Mas se estiver na borda, apenas aqueles cuja decoração apontem para dentro do retângulo serão detectados, como pode ser visto na Figura 16.

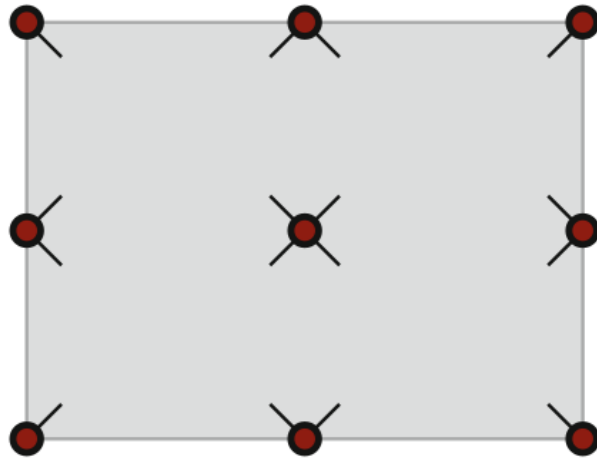


Figura 16 – Pontos decorados que são detectados pela medida aplicada no retângulo R .

Fonte: Chazal *et al.* (2016).

Definição 3.16. Seja $R = [a, b] \times [c, d]$, em que $a < b \leq c < d$ e considere o ponto decorado (p^*, q^*) com $p^* < q^*$. Definimos $(p^*, q^*) \in R$ se alguma das condições equivalentes é verdade:

- Se $p^* \in [a, b]$ e $q^* \in [c, d]$;
- Se $a < p^* < b$ e $c < q^* < d$ na ordem total definida anteriormente;
- Se $a^+ \leq p^* \leq b^-$ e $c^+ \leq q^* \leq d^-$;
- Se o intervalo satisfaz $[b, c] \subseteq (p^*, q^*) \subseteq (a, d)$;
- O ponto com o traço (p^*, q^*) está dentro do retângulo R .

Definição 3.17. Definimos como **r -interior** do retângulo R o conjunto

$$R^\times = \{(p^*, q^*) \mid (p^*, q^*) \in R\}.$$

Também podemos definir o **interior** de R como o conjunto

$$R^\circ = (a, b) \times (c, d).$$

A expressão $|_R$ indica a restrição de um multiconjunto de pontos decorados no retângulo R .

Corolário 3.18. Suponha que \mathfrak{V} seja um módulo de persistência decomponível sobre R .

$$\mathfrak{V} = \bigoplus_{l \in L} k(p_l^*, q_l^*).$$

Então

$$\mu_{\mathfrak{R}}(R) = \text{card}(\text{Dgm}(\mathfrak{V})|_R)$$

para todo retângulo $R = [a, b] \times [c, d]$ com $a < b \leq c < d$.

Demonstração. A demonstração segue direto das Proposições 3.15 e 3.11. \square

A função μ é chamada de medida pois é aditiva em relação a divisão dos retângulos. Vamos provar este fato agora.

Proposição 3.19. $\mu_{\mathfrak{V}}$ é aditiva sobre divisão vertical e horizontal dos retângulos:

$$\mu_{\mathfrak{V}}([a, b] \times [c, d]) = \mu_{\mathfrak{V}}([a, p] \times [c, d]) + \mu_{\mathfrak{V}}([p, b] \times [c, d])$$

$$\mu_{\mathfrak{V}}([a, b] \times [c, d]) = \mu_{\mathfrak{V}}([a, b] \times [c, q]) + \mu_{\mathfrak{V}}([a, b] \times [q, d])$$

para todo $a < p < b \leq c < q < d$. Esta propriedade pode ser visualizada na Figura 17.

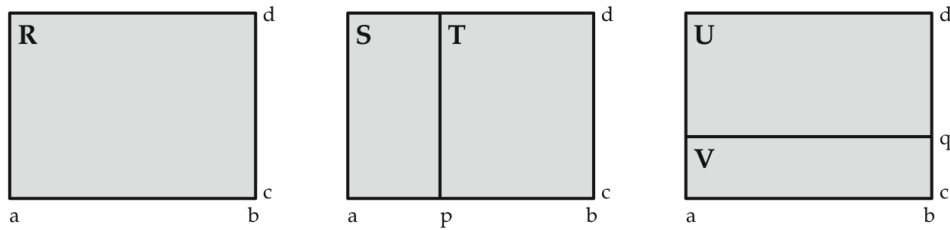


Figura 17 – Representação gráfica da Proposição 3.19

Fonte: Chazal *et al.* (2016).

Demonstração. A demonstração segue direto da Proposição 3.12: para a aditividade na divisão horizontal temos que

$$\begin{aligned} \mu_{\mathfrak{V}}([a, b] \times [c, d]) &= \langle \circ_a \text{---} \bullet_b \text{---} \bullet_c \text{---} \circ_d \rangle \\ &= \langle \circ_a \text{---} \bullet_p \text{---} \bullet_b \text{---} \bullet_c \text{---} \circ_d \rangle + \langle \circ_a \text{---} \circ_p \text{---} \bullet_b \text{---} \bullet_c \text{---} \circ_d \rangle \\ &= \langle \circ_a \text{---} \bullet_p \text{---} \bullet_c \text{---} \circ_d \rangle + \langle \text{---} \circ_p \text{---} \bullet_b \text{---} \bullet_c \text{---} \circ_d \rangle \\ &= \mu_{\mathfrak{V}}([a, p] \times [c, d]) + \mu_{\mathfrak{V}}([p, b] \times [c, d]). \end{aligned}$$

De forma análoga vale o resultado para a divisão vertical. \square

3.2.2 *r-medidas abstratas*

Até agora trabalhamos com módulos de persistência e uma medida associada. Porém, podemos trabalhar de forma mais abstrata, sem mencionar os módulos.

Definição 3.20. Seja $\mathcal{D} \subseteq \mathbb{R}^2$. Defina

$$\text{Rect}(\mathcal{D}) = \{ [a, b] \times [c, d] \subset \mathcal{D} \mid a < b \text{ e } c < d \}.$$

A **r-medida** ou **medida retangular** em \mathcal{D} é uma função

$$\mu: \text{Rect}(\mathcal{D}) \rightarrow \{0, 1, \dots\} \cup \{\infty\},$$

que também é aditiva na divisão horizontal e vertical dos retângulos.

Proposição 3.21. Seja μ uma **r-medida** em $\mathcal{D} \subseteq \mathbb{R}^2$. Então

- Se $R \in \text{Rect}(\mathcal{D})$ pode ser escrito como a união de retângulos com interior disjuntos, $R = R_1 \cup \dots \cup R_n$, então $\mu(R) = \sum \mu(R_i)$;
- Se $R \subseteq S$, então $\mu(R) \leq \mu(S)$.

Demonstração. (*Finitamente aditiva*) Seja $R = [a, b] \times [c, d]$. Por indução e pela propriedade de divisão vertical, temos que a aditividade finita vale para decomposições da forma

$$R = \bigcup_i R_i,$$

em que cada $R_i = [a_i, a_{i+1}] \times [c, d]$, com $a = a_1 < \dots < a_m = b$. De forma análoga, vale para divisões horizontais e portanto a aditividade vale para decomposições da forma

$$R = [a, b] \times [c, d] = \bigcup_{i,j} R_{ij},$$

onde $R_{ij} = [a_i, a_{i+1}] \times [c_j, c_{j+1}]$ com $a = a_1 < \dots < a_m = b$ e $c = c_1 < \dots < c_n = d$. Para uma decomposição arbitrária $R = R_1 \cup \dots \cup R_k$ o resultado segue considerando uma decomposição em que cada R_i é decomposto em intervalos da forma acima.

(*Monotonicidade*) Decomponha S em uma coleção de retângulos R e R_1, \dots, R_k que possuem interiores disjuntos. Portanto, da propriedade de aditividade e que $\mu \geq 0$

$$\begin{aligned} \mu(S) &= \mu(R) + \mu(R_1) + \dots + \mu(R_k) \\ &\geq \mu(R). \end{aligned}$$

□

Proposição 3.22. (Subaditividade) Seja μ uma **r-medida** em $\mathcal{D} \subseteq \mathbb{R}^2$. Se um retângulo $R \in \text{Rect}(\mathcal{D})$ está contido numa união finita de retângulos $R_i \in \text{Rect}(\mathcal{D})$

$$R \subseteq R_1 \cup \dots \cup R_k,$$

então

$$\mu(R) \leq \mu(R_1) + \dots + \mu(R_k).$$

Demonstração. Seja $a_1 < \dots < a_m$ sequência de todos as coordenadas x de cada vértice dos retângulos. Considere também $c_1 < \dots < c_n$ sequência de todos os valores do eixo y de cada vértice dos retângulos. Então cada retângulo R_k pode ser decomposto como união dos seguintes subretângulos

$$[a_i, a_{i+1}] \times [c_j, c_{j+1}],$$

que possuem interiores disjuntos por construção e sua medida é a soma das medidas de cada subretângulo. Como cada subretângulo de R pertence a um ou mais retângulos R_i , segue da aditividade o resultado. \square

3.2.3 Equivalência de medidas e diagramas

As r -medidas de persistência permitem o estudo dos diagramas de persistência de maneira mais analítica, facilitando o desenvolvimento da teoria. Nesta seção iremos demonstrar a equivalência entre medidas abstratas e multiconjuntos localmente finitos. Para isso, assumiremos que a medida é finita, como dito anteriormente.

O **r-interior** de uma região $\mathcal{D} \subseteq \mathbb{R}^2$ é o conjunto definido abaixo

$$\mathcal{D}^\times = \{ (p^*, q^*) \mid \exists R \in \text{Rect}(\mathcal{D}) \text{ tal que } (p^*, q^*) \in R \}.$$

A definição acima pode ser vista com o seguinte significado: o conjunto dos ponto decorados pode ser determinado por algum retângulo em \mathcal{D} .

O **interior** de \mathcal{D} é dado por

$$\mathcal{D}^\circ = \{ (p, q) \mid \exists R \in \text{Rect}(\mathcal{D}) \text{ tal que } (p, q) \in R^\circ \},$$

onde para um retângulo $R = [a, b] \times [c, d]$, $R^\circ = (a, b) \times (c, d)$.

Teorema 3.23. (O teorema da equivalência) Seja $\mathcal{D} \subseteq \mathbb{R}^2$. Então existe uma correspondência bijetiva entre

1. r -medidas μ finitas em \mathcal{D} . Finito neste caso significa que $\mu(R) < \infty$ para todo $R \in \text{Rect}(\mathcal{D})$.
2. Multiconjuntos A em \mathcal{D} localmente finitos. Localmente finito significa $\text{card}(A|_R) < \infty$ para todo $R \in \text{Rect}(\mathcal{D})$.

A medida μ correspondente ao multiconjunto A é relacionada pela fórmula

$$\mu(R) = \text{card}(A|_R) \quad (3.2)$$

para todo $R \in \text{Rect}(\mathcal{D})$, ou equivalentemente

$$\mu(R) = \sum_{(p^*, q^*) \in R} m(p^*, q^*), \quad (3.3)$$

em que

$$m: \mathcal{D}^\times \rightarrow \{0, 1, 2, \dots\}$$

é a função multiplicidade de A .

Demonstração. $(2) \rightarrow (1)$: para este passo, basta provar que a medida definida na Eq. (3.2) é uma r -medida. De fato, é finita pois A é localmente finito. Para verificar a aditividade, suponha que para um retângulo R qualquer, ele se divida horizontalmente ou verticalmente em R_1 e R_2 . Note então que (p^*, q^*) pertence a exatamente R_1 ou R_2 . Portanto,

$$\mu(R) = \text{card}(A|_R) = \text{card}(A|_{R_1}) + \text{card}(A|_{R_2}) = \mu(R_1) + \mu(R_2),$$

provando a primeira implicação.

$(1) \rightarrow (2)$: Dada uma r -medida, iremos (1) construir o multiconjunto A em \mathcal{D}^\times , (2) mostrar que μ e A estão relacionadas pela Eq. (3.2) e (3) mostrar que A é único. Na prática, iremos construir a função de multiplicidade m e definir a Eq. (3.3) ao invés de A diretamente.

Passo 1. (Fórmula da multiplicidade) Seja μ uma r -medida finita em \mathcal{D} . Defina

$$m(p^*, q^*) = \min \{ \mu(R) \mid R \in \text{Rect}(\mathcal{D}), (p^*, q^*) \in R \} \quad (3.4)$$

para $(p^*, q^*) \in \mathcal{D}^\times$. Observe que o mínimo é atingido, já que tomamos $(p^*, q^*) \in \mathcal{D}^\times$ e μ é uma medida que assume valores inteiros. Utilizaremos uma definição alternativa, ao invés de minimizar sob todos os retângulos, tomamos o limite de uma sequência decrescente de retângulos.

Lema 3.24. Sejam (ξ_i) e (η_i) duas sequências não crescentes de números reais positivos que tendem a zero quando $i \rightarrow \infty$. Então

$$m(p^+, q^+) = \lim_{i \rightarrow \infty} \mu([p, p + \xi_i] \times [q, q + \eta_i]),$$

e similarmente

$$m(p^+, q^-) = \lim_{i \rightarrow \infty} \mu([p, p + \xi_i] \times [q - \eta_i, q]),$$

$$m(p^-, q^+) = \lim_{i \rightarrow \infty} \mu([p - \xi_i, p] \times [q, q + \eta_i]),$$

$$m(p^-, q^-) = \lim_{i \rightarrow \infty} \mu([p - \xi_i, p] \times [q - \eta_i, q]).$$

Demonstração. Note primeiro que a sequência de retângulos $R_i = [p, p + \xi_i] \times [q, q + \eta_i]$ é cofinal no conjunto de retângulos R contendo (p^+, q^+) , ou seja, para todo R deste tipo, $R_i \subseteq R$ para i suficientemente grande.

Pela monotonicidade de μ e como a sequência de inteiros não negativos $\mu(R_i)$ é não crescente, ela estabiliza no seu limite em algum momento. Portanto

$$m(p^+, q^+) \leq \min_i \mu(R_i) = \lim_{i \rightarrow \infty} \mu(R_i) \leq \mu(R)$$

para todo R contendo (p^+, q^+) . Tomando o mínimo de ambos os lados da desigualdade acima sob todos os R , o lado direito se torna $m(p^+, q^+)$, logo

$$m(p^+, q^+) = \lim_{i \rightarrow \infty} \mu(R_i).$$

Os outros três casos são similares. □

Passo 2. Uma vez com a função multiplicidade definida, mostraremos que ela está de acordo com a Eq. (3.3). Podemos definir uma medida baseada na função multiplicidade

$$\nu(R) = \sum_{(p^*, q^*) \in R} m(p^*, q^*).$$

Nos resta mostrar então que $\mu = \nu$. Vamos utilizar indução sobre $k = \mu(R)$.

Caso base. Se $\mu(R) = 0$, então para todo $(p^*, q^*) \in R$ temos

$$0 \leq m(p^*, q^*) \leq \mu(R) = 0.$$

Portanto, $\nu(R) = 0$.

Passo indutivo. Suponha $\mu(R) = \nu(R)$ para todo retângulo R com $\mu(R) < k$. Seja agora um retângulo R_0 tal que $\mu(R_0) = k$. Vamos mostrar que $\nu(R_0) = 0$. A ideia para este passo é construir uma sequência decrescente de retângulos fechados de forma que haverá apenas um ponto na interseção destes retângulos (Teorema de Cantor).

Divida o retângulo em quatro quadrantes iguais, S_1, \dots, S_4 . Pela aditividade finita

$$\begin{aligned} \mu(R_0) &= \mu(S_1) + \dots + \mu(S_4), \\ \nu(R_0) &= \nu(S_1) + \dots + \nu(S_4). \end{aligned}$$

Se todo quadrante satisfaz $\mu(S_i) < k$, segue então que $\mu(R_0) = \nu(R_0)$, pelo passo indutivo, concluindo a demonstração. Caso contrário, um dos quadrantes tem medida igual a k , digamos R_1 , enquanto o resto será 0. Então $\mu(R_1) = k$. Resta mostrar que $\nu(R_1) = k$.

Repetindo o argumento acima, obtemos um retângulo R_i tal que $\mu(R_i) = k$. Divida o retângulo R_i em quatro quadrantes iguais. Temos dois casos: todos os quadrantes satisfazem a hipótese indutiva $\mu < k$ e teríamos terminado a demonstração. Ou existe um quadrante R_{i+1} com $\mu(R_{i+1}) = k$ e temos que mostrar que $\nu(R_{i+1}) = k$.

No pior caso, temos uma sequência decrescente de retângulos fechados

$$R_0 \supset R_1 \supset R_2 \supset \dots$$

com cada quadrante sendo do mesmo tipo anterior, $\mu(R_i) = k$. Pelo teorema de Cantor e o fato de o diâmetro dos conjuntos tender a 0, temos que a interseção dos intervalos fechados possuem apenas um ponto (r, s) .

Vamos mostrar agora que $v(R_0) = k$ avaliando a soma explicitamente sob todos os pontos decorados em R_0 .

Note primeiro que pontos decorados em R_0 que saem da sequência de retângulos em algum momento não contribuem para $v(R_0)$, já que se $(p^*, q^*) \in R_0$ e $(p^*, q^*) \in R_{i-1} - R_i$ para algum i , então o ponto pertence a algum dos outros três quadrantes, sendo assim $\mu = 0$. Portanto, pela fórmula de multiplicidade, $m(p^*, q^*) = 0$.

Assim, os únicos pontos que contribuem para $v(R_0)$ são as variações decoradas de (r, s) , já que é o único ponto não decorado na interseção. Agora a avaliação de $v(R_0)$ depende de como este ponto decorado se encontra na interseção. Existem 3 possíveis casos: as 4, 2, 1 decorações estão contidas nos retângulos, como podemos ver na Figura 18

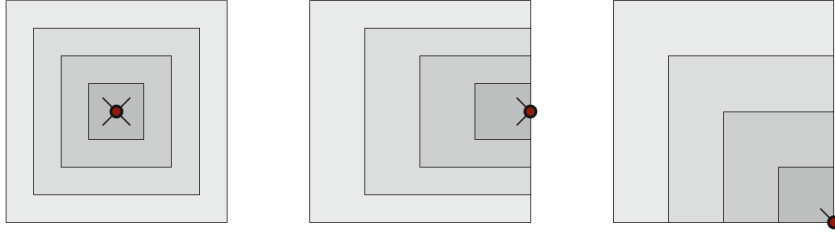


Figura 18 – Possíveis casos dos pontos decorados (r^*, s^*) na interseção $\cap_i R_i$.

Fonte: Chazal *et al.* (2016).

Vamos mostrar apenas para o caso em que as 4 decorações estão em todos os retângulos R_i . Os outros dois casos são análogos.

Suponha que todas as decorações $(r^*, s^*) \in R_i$ para todo i . Agora divida cada retângulo R_i em quatro partes, de forma que cada parte contenha apenas uma decoração: $R_i^{++}, R_i^{+-}, R_i^{-+}, R_i^{--}$. A divisão ocorre de forma que o ponto (r, s) fique num vértice dividido pelos quatro retângulos. Pelo Lema 3.24,

$$\begin{aligned} m(r^+, s^+) &= \lim_{i \rightarrow \infty} \mu(R_i^{++}), & m(r^+, s^-) &= \lim_{i \rightarrow \infty} \mu(R_i^{+-}), \\ m(r^-, s^+) &= \lim_{i \rightarrow \infty} \mu(R_i^{-+}), & m(r^-, s^-) &= \lim_{i \rightarrow \infty} \mu(R_i^{--}). \end{aligned}$$

Além disso, cada uma dessas sequências decrescentes de inteiros estabilizam no seu limite. Portanto, para valores de i suficientemente grandes

$$\begin{aligned} v(R_0) &= m(r^+, s^+) + m(r^+, s^-) + m(r^-, s^+) + m(r^-, s^-) \\ &= \mu(R_i^{++}) + \mu(R_i^{+-}) + \mu(R_i^{-+}) + \mu(R_i^{--}) = \mu(R_i) = k. \end{aligned}$$

Passo 3. (Unicidade) Suponha que $m'(p^*, q^*)$ é uma outra função multiplicidade em \mathcal{D}^\times cuja r -medida associada

$$v'(R) = \sum_{(p^*, q^*) \in R} m'(p^*, q^*)$$

satisfaz $\mu = v'$. Vamos mostrar que $m = m'$.

Seja $(p^*, q^*) \in \mathcal{D}^\times$ e R um retângulo que contém o ponto (p^*, q^*) em um dos seus vértices. Como $v(R) = v'(R) = \mu(R) < \infty$, existem apenas finitos pontos decorados em R . Além disso, podemos diminuir R de forma que (p^*, q^*) seja o único ponto decorado em R com multiplicidade positiva em ambas as medidas. Portanto,

$$m(p^*, q^*) = v(R) = \mu(R) = v'(R) = m'(p^*, q^*).$$

Como (p^*, q^*) era um ponto qualquer, segue o resultado. \square

Definição 3.25. Seja μ uma medida finita em $\mathcal{D} \subseteq \mathbb{R}^2$. Então

- o **diagrama decorado** de μ é o único multiconjunto localmente finito $\text{Dgm}(\mu)$ em \mathcal{D}^\times tal que

$$\mu(R) = \text{card}(\text{Dgm}(\mu)|_R)$$

para todo retângulo $R \in \text{Rect}(\mathcal{D})$;

- o **diagrama não decorado** de μ é o multiconjunto localmente finito em \mathcal{D}°

$$\text{dgm}(\mu) = \{ (p, q) \mid (p^*, q^*) \in \text{Dgm}(\mu) \} \cap \mathcal{D}^\times$$

obtido esquecendo a decoração dos pontos e restringindo ao interior.

3.3 Comportamento de módulos e exemplos

Quando estende-se as medidas e diagramas de persistência para o semiplano diagonal superior com os pontos no infinito e para medidas que assumem valor infinito também, os módulos de persistência não são sempre bem comportados, necessitando diferenciar entre as diversas situações. Abaixo estão três diferentes noções e em seguida apresentaremos apenas mais uma, já que estamos trabalhando apenas com medidas finitas.

- Um módulo de persistência é do **tipo finito** se é uma soma direta finita de módulos intervalares;
- Um módulo de persistência é **localmente finito** se é uma soma direta de módulos intervalares e satisfaz a seguinte propriedade: qualquer subconjunto de \mathbb{R} intersecta um número finito de módulos intervalares;

- Um módulo de persistência \mathfrak{V} é **pontualmente de dimensão finita (pdf)** se cada espaço vetorial V_t tiver dimensão finita.

Como estamos trabalhando medidas finitas apenas, vamos definir o módulo q -tame. Seja \mathfrak{V} um módulo de persistência. Dizemos que \mathfrak{V} é q -tame se $\mu_{\mathfrak{V}}(Q) < \infty$ para todo quadrante Q que não toca a diagonal. Em outras palavras

$$\langle \bullet_b \text{---} \bullet_c \mid \mathfrak{V} \rangle < \infty$$

para todo $b < c$. O diagrama de persistência $\text{Dgm}(\mu_{\mathfrak{V}})$ é definido sobre o conjunto

$$\{(p^*, q^*) \mid -\infty \leq p < q \leq +\infty\}.$$

Abaixo mostramos alguns exemplos que encontra-se na teoria para módulos de persistência q -tame.

Teorema 3.26. Seja X um poliedro compacto e $f: X \rightarrow \mathbb{R}$ uma função contínua. Então a homologia persistente $H(\mathfrak{X}_{\text{sub}})$ da filtração de subnível de (X, f) é q -tame.

Demonstração. Precisamos mostrar que

$$H(X^b) \rightarrow H(X^c)$$

tem rank finito para qualquer $b < c$, já como visto anteriormente, o rank da função acima é igual a

$$\langle \bullet_b \text{---} \bullet_c \rangle.$$

Considere uma triangulação de X e a subdivida de forma que nenhum simplexo intersecte $f^{-1}(b)$ e $f^{-1}(c)$ ao mesmo tempo. Defina agora Y como a união de todos os simplexos que intersectam X^b . Portanto,

$$X^b \subseteq Y \subseteq X^c$$

e aplicando o funtor de homologia

$$H(X^b) \longrightarrow H(Y) \longrightarrow H(X^c).$$

Como Y é um poliedro compacto, $H(Y)$ tem dimensão finita, portanto a aplicação $H(X^b) \rightarrow H(X^c)$ tem rank finito. \square

Corolário 3.27. Seja X um poliedro localmente compacto e $f: X \rightarrow \mathbb{R}$ uma função propriamente contínua que seja limitada inferiormente. Então $H(\mathfrak{X}_{\text{sub}})$ é q -tame.

Demonstração. Novamente, precisamos mostrar que $H(X^b) \rightarrow H(X^c)$ tem rank finito. Mas neste caso iremos aplicar o Teorema 3.26, para tanto precisamos de um subpoliedro compacto de X que contenha X^c . Seja uma triangulação localmente finita de X e considere os

simplexos fechados que intersectam X^c . Como X^c é compacto, já que f é uma função própria, no sentido que a pre-imagem de compacto é compacto, temos que existe um número finito de conjuntos fechados que intersecta X^c . Considere agora a união desses conjuntos fechados como o subpoliedro desejado. \square

Corolário 3.28. Seja A um subconjunto não vazio compacto de $X = \mathbb{R}^n$ e $f: X \rightarrow \mathbb{R}$, $f(x) = \min_{a \in A} \|x - a\|$, para qualquer norma $\|\cdot\|$. Segue então do Corolário 3.27 que $H(\mathfrak{X}_{\text{sub}})$ é q -tame.

3.4 Intercalação

A intercalação é um modo de comparar dois módulos de persistência. Dizemos que os módulos \mathfrak{U} e \mathfrak{V} são isomorfos se existem homomorfismos

$$\Psi \in \text{Hom}(\mathfrak{U}, \mathfrak{V}), \quad \Phi \in \text{Hom}(\mathfrak{V}, \mathfrak{U}),$$

tais que

$$\Psi\Phi = 1_{\mathfrak{U}}, \quad \Phi\Psi = 1_{\mathfrak{V}}.$$

No entanto, para trabalhar com módulos de persistência, a noção de isomorfismo é muito forte. Para isso, podemos enfraquecê-la definindo a δ -intercalação entre dois módulos. Nesta subseção iremos definir o interlaçamento e demonstrar o lema de interpolação, como se define um *caminho contínuo* entre dois módulos de persistência.

3.4.1 Homomorfismos e módulos de persistência

Primeiro vamos considerar homomorfismos que mudam o índice de persistência dos módulos. Sejam $\mathfrak{U}, \mathfrak{V}$ módulos de persistência sobre \mathbb{R} e $\delta \in \mathbb{R}$ qualquer. Então um homomorfismo de grau δ é uma coleção Φ de aplicações lineares

$$\phi_t: U_t \rightarrow V_{t+\delta}$$

para todo $t \in \mathbb{R}$ tal que o diagrama

$$\begin{array}{ccc} U_s & \xrightarrow{u_t^\delta} & U_t \\ \phi_s \downarrow & & \downarrow \phi_t \\ V_{s+\delta} & \xrightarrow{v_{t+\delta}^{s+\delta}} & V_{t+\delta} \end{array}$$

comuta para todo $s \leq t$. Escrevemos

$$\begin{aligned} \text{Hom}^\delta(\mathfrak{U}, \mathfrak{V}) &= \{ \text{homomorfismos } \mathfrak{U} \rightarrow \mathfrak{V} \text{ de grau } \delta, \\ \text{End}^\delta(\mathfrak{V}) &= \{ \text{homomorfismos } \mathfrak{V} \rightarrow \mathfrak{V} \text{ de grau } \delta \}. \end{aligned}$$

A composição é definida de forma natural, nos dando a aplicação

$$\mathrm{Hom}^{\delta_2}(\mathfrak{V}, \mathfrak{W}) \times \mathrm{Hom}^{\delta_1}(\mathfrak{U}, \mathfrak{V}) \rightarrow \mathrm{Hom}^{\delta_1 + \delta_2}(\mathfrak{U}, \mathfrak{W}).$$

Para $\delta \geq 0$, a aplicação de grau δ mais importante é a aplicação *shift*

$$1_{\mathfrak{V}}^{\delta} \in \mathrm{End}^{\delta}(\mathfrak{V}),$$

que é a coleção de aplicações lineares $(v_{t+\delta}^t)$ da estrutura de \mathfrak{V} . Se Φ é um homomorfismo $\mathfrak{U} \rightarrow \mathfrak{V}$ de grau qualquer, então por definição

$$\Phi 1_{\mathfrak{U}}^{\delta} = 1_{\mathfrak{V}}^{\delta} \Phi,$$

para todo $\delta \geq 0$.

Observação 3.29. Podemos escrever os morfismos de grau não nulos de forma diferente. Seja \mathfrak{V} um módulo de persistência e $\delta \in \mathbb{R}$ qualquer, denotamos por $\mathfrak{V}[\delta]$ o **módulo transladado**

$$(V[\delta])_t = V_{t+\delta}, \quad (v[\delta])_t^s = v_{t+\delta}^{s+\delta},$$

ou seja, $V[\delta]$ é o módulo \mathfrak{V} transladado em δ para baixo. Além disso, temos identificações com os homomorfismos

$$\mathrm{Hom}^{\delta}(\mathfrak{U}, \mathfrak{V}) = \mathrm{Hom}(\mathfrak{U}, \mathfrak{V}[\delta]) = \mathrm{Hom}(\mathfrak{U}[a], \mathfrak{V}[a + \delta])$$

para todo $a \in \mathbb{R}$.

Definição 3.30. Seja $\delta \geq 0$. Dizemos que $\mathfrak{U}, \mathfrak{V}$ são δ -intercalados se existem aplicações

$$\Phi \in \mathrm{Hom}^{\delta}(\mathfrak{U}, \mathfrak{V}), \Psi \in \mathrm{Hom}^{\delta}(\mathfrak{V}, \mathfrak{U})$$

tais que

$$\Psi\Phi = 1_{\mathfrak{U}}^{2\delta}, \Phi\Psi = 1_{\mathfrak{V}}^{2\delta}.$$

Exemplo 3.31. Seja X um espaço topológico e $f, g: X \rightarrow \mathbb{R}$. Suponha que $\|f - g\|_{\infty} < \delta$. Então os módulos de persistência $H(\mathfrak{X}_{\mathrm{sub}}^f), H(\mathfrak{X}_{\mathrm{sub}}^g)$ são δ -intercalados.

De fato, note primeiro que

$$\begin{aligned} (X, f)^t &\subseteq (X, g)^{t+\delta} \\ (X, g)^t &\subseteq (X, f)^{t+\delta}, \end{aligned}$$

já que para $x \in (X, f)^t$ temos que $f(x) \leq t$ e como $\|f - g\|_{\infty} < \delta$, $|g(x) - f(x)| < \delta$ e logo $g(x) < f(x) + \delta \leq t + \delta$, implicando $x \in (X, g)^{t+\delta}$. De forma análoga temos a segunda inclusão.

Pela functorialidade de H , temos que para todo t , existem aplicações lineares induzidas das inclusões acima

$$\begin{aligned} \Phi: H(\mathfrak{X}_{\mathrm{sub}}^f) &\rightarrow H(\mathfrak{X}_{\mathrm{sub}}^g) \\ \Psi: H(\mathfrak{X}_{\mathrm{sub}}^g) &\rightarrow H(\mathfrak{X}_{\mathrm{sub}}^f) \end{aligned}$$

de grau δ . Como as aplicações são decorrentes de um funtor, as condições de interlaçamento são satisfeitas.

Da mesma forma que módulos de persistência podem ser vistos sobre posets, uma relação de interlaçamento entre dois módulos pode ser vista como um módulo de persistência sobre um poset. A seguir iremos desenvolver esta ideia.

Considere a seguinte ordem parcial no plano

$$(p_1, q_1) \leq (p_2, q_2) \iff p_1 \leq p_2, q_1 \leq q_2.$$

Agora, para todo número real x , defina a faixa diagonal transladada no plano

$$\Delta_x = \{ (p, q) \mid q - p = 2x \} = \{ (t - x, t + x) \mid t \in \mathbb{R} \}. \quad (3.5)$$

Δ_x visto como um poset é isomorfo à reta real. Iremos usar o isomorfismo de (3.5) como uma identificação canônica entre módulos de persistência sobre \mathbb{R} e sobre Δ_x .

Proposição 3.32. Sejam x, y números reais. Os módulos de persistência \mathfrak{U} e \mathfrak{V} são $|y - x|$ -intercalados se, e somente se, existe um módulo de persistência \mathfrak{W} sobre $\Delta_x \cup \Delta_y$ tal que $\mathfrak{W}|_{\Delta_x} = \mathfrak{U}$ e $\mathfrak{W}|_{\Delta_y} = \mathfrak{V}$. O conjunto $\Delta_x \cup \Delta_y$ é um subposet de \mathbb{R}^2 .

Demonstração. Suponha que $x < y$ sem perda de generalidade. Vamos mostrar que a informação 1 abaixo dada pelas aplicações de $|y - x|$ -interlaçamento são equivalentes às informações abaixo (2) do módulo de persistência \mathfrak{W} .

1. Note que como $\mathfrak{U}, \mathfrak{V}$ são módulos $|y - x|$ -intercalados, existem morfismos Φ, Ψ tais que

$$\phi_t: U_t \rightarrow V_{t+y-x}, \quad \psi_t: V_t \rightarrow U_{t+y-x},$$

que satisfazem as seguintes relações para todo $\eta \geq 0$:

$$\begin{aligned} \Phi 1_{\mathfrak{U}}^{\eta} &= 1_{\mathfrak{V}}^{\eta} \Phi, & \Psi 1_{\mathfrak{V}}^{\eta} &= 1_{\mathfrak{U}}^{\eta} \Psi, \\ \Psi \Phi &= 1_{\mathfrak{U}}^{2y-2x}, & \Phi \Psi &= 1_{\mathfrak{V}}^{2y-2x}. \end{aligned} \quad (3.6)$$

2. Sejam $R, S, T \in \Delta_x \cup \Delta_y$ tais que $R \leq S \leq T$. Temos então as aplicações lineares entre os espaços vetoriais de \mathfrak{W} que satisfazem a lei de composição

$$w_T^R = w_T^S \circ w_S^R.$$

Vamos mostrar que cada aplicação w_T^S está relacionada com algum dos mapas a seguir

$$\begin{aligned} 1_{\mathfrak{U}}^{\eta} & \text{ de } \Delta_x \text{ para } \Delta_y \\ 1_{\mathfrak{V}}^{\eta} & \text{ de } \Delta_y \text{ para } \Delta_x \\ 1_{\mathfrak{U}}^{\eta} \Phi & \text{ de } \Delta_x \text{ para } \Delta_y \\ 1_{\mathfrak{U}}^{\eta} \Psi & \text{ de } \Delta_y \text{ para } \Delta_x. \end{aligned}$$

Note primeiro que podemos recuperar as aplicações ϕ_t, ψ_t como mapas verticais de Δ_x para Δ_y e mapas horizontais de Δ_y para Δ_x respectivamente, como pode ser visto na Figura 19.

$$\begin{aligned} U_t &\cong W_{(t-x, t+x)} \rightarrow W_{(t-x, t+2y-x)} \cong V_{t+y-x} \\ V_t &\cong W_{(t-y, t+y)} \rightarrow W_{(t+y-2x, t+y)} \cong U_{t+y-x} \end{aligned}$$

Portanto, a lei de composição das aplicações de \mathfrak{W} implicam nas relações dadas

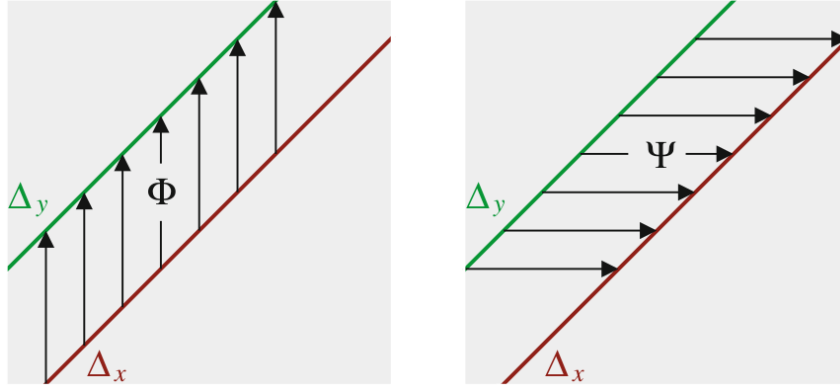


Figura 19 – Os morfismos Φ, Ψ recuperados do módulo de persistência \mathfrak{W} sobre $\Delta_x \cup \Delta_y$.

Fonte: Chazal *et al.* (2016).

em (3.6). Tendo mostrado que 2 implica em 1, note que as aplicações lineares w_T^S , $S \leq T$ podem ser fatoradas da seguinte forma:

$$\begin{aligned} w_T^S &= v_t^{s+y-x} \circ \phi_s, \text{ se } S \in \Delta_x \text{ e } T \in \Delta_y, \\ w_T^S &= u_t^{s+y-x} \circ \psi_s, \text{ se } S \in \Delta_y \text{ e } T \in \Delta_x. \end{aligned} \tag{3.7}$$

Assim, a lei de composição é satisfeita para cada par de aplicações apenas usando as relações dadas em (3.7):

$$(1_{\mathfrak{W}}^{\eta} \Phi)(1_{\mathfrak{U}}^{\xi} \Psi) = 1_{\mathfrak{W}}^{\eta} \Phi 1_{\mathfrak{U}}^{\xi} \Psi = 1_{\mathfrak{W}}^{\eta} 1_{\mathfrak{W}}^{\xi} \Phi \Psi = 1^{\eta+\xi} 1_{\mathfrak{W}}^{2y-2x} = 1_{\mathfrak{W}}^{\eta+\xi+2y-2x}.$$

□

3.4.2 O lema de interpolação

Vamos anunciar o lema da interpolação, que será utilizado na demonstração do teorema da estabilidade dos módulos de persistência.

Lema 3.33. (Lema de interpolação) Suponha que \mathfrak{U} e \mathfrak{V} são módulos de persistência δ -intercalados. Então existe uma família de módulos de persistência $(\mathfrak{U}_x \mid x \in [0, \delta])$ tais que U_0 e U_{δ} são iguais a \mathfrak{U} e \mathfrak{V} respectivamente e $\mathfrak{U}_x, \mathfrak{U}_y$ são $|y-x|$ -intercalados para todo $x, y \in [0, \delta]$. Além disso, se \mathfrak{U} e \mathfrak{V} são q -tames, então (\mathfrak{U}_x) são q -tames.

O Teorema 3.34 é uma versão mais forte do Lema 3.33.

Teorema 3.34. (Lema de interpolação, versão 2) Todo módulo de persistência, \mathfrak{W} , sobre $\Delta_0 \cup \Delta_\delta$, se estende a um módulo de persistência $\tilde{\mathfrak{W}}$ sobre a faixa diagonal

$$\Delta_{[0,\delta]} = \{ (p, q) \mid 0 \leq q - p \leq 2\delta \} \subset \mathbb{R}^2.$$

Se $\mathfrak{W}|_{\Delta_0}, \mathfrak{W}|_{\Delta_\delta}$ são q -tames, então a extensão pode ser escolhida de forma que cada $\tilde{\mathfrak{W}}|_{\Delta_x}$ é q -tame.

Demonstração. Sejam $\Delta_0 \cup \Delta_\delta$ e $\Delta_{[0,\delta]}$ duas categorias no mesmo sentido definido no início deste capítulo para o conjunto dos reais \mathbb{R} ($s \rightarrow t$ se, e somente se, $s \leq t$). Então os módulos de persistência sobre esses posets podem ser vistos como funtores para a categoria dos espaços vetoriais. O teorema de extensão de Kan (LANE, 1978) afirma que existe uma extensão $\tilde{\mathfrak{W}}$

$$\begin{array}{ccc} & \Delta_{[x_0, x_1]} & \\ \uparrow & \text{---} \tilde{\mathfrak{W}} \text{---} & \\ \Delta_{x_0} \cup \Delta_{x_1} & \xrightarrow{\mathfrak{W}} & \text{Vect} \end{array}$$

para qualquer funtor \mathfrak{W} , completando a demonstração. \square

Se \mathfrak{U} e \mathfrak{V} são δ -intercalados, então pela Proposição 3.32 existe um módulo de persistência \mathfrak{W} sobre $\Delta_0 \cup \Delta_\delta$ tal que $\mathfrak{W}|_{\Delta_0} = \mathfrak{U}$ e $\mathfrak{W}|_{\Delta_\delta} = \mathfrak{V}$. Pelo Teorema 3.34, este módulo se estende a $\tilde{\mathfrak{W}}$ sobre a faixa $\Delta_{[0,\delta]}$. Defina então $\mathfrak{U}_x = \tilde{\mathfrak{W}}|_{\Delta_x}$, logo \mathfrak{U}_x e \mathfrak{U}_y são $|y-x|$ -intercalados para todo $x, y \in [0, \delta]$.

3.5 O teorema de isometria

Existe uma relação entre os módulos de persistência e seus respectivos diagramas de persistência. Para módulos de persistência bem comportados, como os q -tames, existe uma isometria entre módulos e diagramas. Para mostrar parcialmente este resultado, iremos definir a distância de interlaçamento entre módulos de persistência e também a distância *bottleneck*, que é calculada em multiconjuntos.

3.5.1 A distância de interlaçamento

Para o primeiro passo, iremos definir a distância de interlaçamento entre módulos de persistência. Primeiro note que para módulos $\mathfrak{U}, \mathfrak{V}$ que são δ -intercalados, então eles são $(\delta + \varepsilon)$ -intercalados para todo $\varepsilon > 0$. Basta tomar os morfismos

$$\begin{aligned} \Phi' &= \Phi 1_{\mathfrak{U}}^\varepsilon = 1_{\mathfrak{V}}^\varepsilon \Phi \\ \Psi' &= \Psi 1_{\mathfrak{V}}^\varepsilon = 1_{\mathfrak{U}}^\varepsilon \Psi \end{aligned}$$

que satisfazem a condição de interlaçamento.

Portanto, o desafio é obter o menor parâmetro possível de forma que módulos de persistência sejam intercalados. O problema é que nem sempre é obtido, sendo assim definimos o conceito de δ^+ -interlaçamento. Dizemos que $\mathfrak{U}, \mathfrak{V}$ são δ^+ -intercalados se eles são $(\delta + \varepsilon)$ -intercalados para todo $\varepsilon > 0$.

Definição 3.35. Sejam $\mathfrak{U}, \mathfrak{V}$ dois módulos de persistência. A **distância de interlaçamento** entre $\mathfrak{U}, \mathfrak{V}$ é dada por

$$\begin{aligned} d_i(\mathfrak{U}, \mathfrak{V}) &= \inf \{ \delta \mid \mathfrak{U}, \mathfrak{V} \text{ são } \delta\text{-intercalados} \} \\ &= \min \{ \delta \mid \mathfrak{U}, \mathfrak{V} \text{ são } \delta^+\text{-intercalados} \}. \end{aligned}$$

Se não existe nenhum δ -interlaçamento entre $\mathfrak{U}, \mathfrak{V}$ para qualquer δ , então $d_i(\mathfrak{U}, \mathfrak{V}) = \infty$.

Proposição 3.36. A distância de interlaçamento satisfaz a desigualdade triangular:

$$d_i(\mathfrak{U}, \mathfrak{W}) \leq d_i(\mathfrak{U}, \mathfrak{V}) + d_i(\mathfrak{V}, \mathfrak{W})$$

para quaisquer módulos de persistência $\mathfrak{U}, \mathfrak{V}$ e \mathfrak{W} .

Demonstração. Dados δ_1 -interlaçamento entre $\mathfrak{U}, \mathfrak{V}$ e δ_2 -interlaçamento entre $\mathfrak{V}, \mathfrak{W}$, então podemos construir um $(\delta_1 + \delta_2)$ -interlaçamento entre $\mathfrak{U}, \mathfrak{W}$ simplesmente compondo as outras duas aplicações de interlaçamento

$$\begin{array}{ccccc} \mathfrak{U} & \xrightarrow{\Phi_1} & \mathfrak{V} & \xrightarrow{\Phi_2} & \mathfrak{W} \\ \mathfrak{U} & \xleftarrow{\Psi_1} & \mathfrak{W} & \xleftarrow{\Psi_2} & \mathfrak{W}. \end{array}$$

Vamos mostrar que $\Phi = \Phi_2 \Phi_1$ e $\Psi = \Psi_1 \Psi_2$. De fato

$$\begin{aligned} \Psi \Phi &= \Psi_1 \Psi_2 \Phi_2 \Phi_1 = \Psi_1 1_{\mathfrak{W}}^{2\delta_2} \Phi_1 = \Psi_1 \Phi_1 1_{\mathfrak{U}}^{2\delta_2} = 1_{\mathfrak{U}}^{\delta_1} 1_{\mathfrak{U}}^{\delta_2} = 1_{\mathfrak{U}}^{2\delta} \\ \Phi \Psi &= \Phi_2 \Phi_1 \Psi_1 \Psi_2 = \Phi_2 1_{\mathfrak{V}}^{2\delta_1} \Psi_2 = \Phi_2 \Psi_2 1_{\mathfrak{W}}^{2\delta_1} = 1_{\mathfrak{W}}^{\delta_2} 1_{\mathfrak{W}}^{\delta_1} = 1_{\mathfrak{W}}^{2\delta} \end{aligned}$$

Agora basta tomar o ínfimo entre δ_1, δ_2 . □

A proposição nos diz que d_i é uma pseudométrica. Não chega a ser uma métrica, pois é possível construir exemplos tais que $d_i(\mathfrak{U}, \mathfrak{V}) = 0$ não implica que $\mathfrak{U} \cong \mathfrak{V}$. Só teremos um isomorfismo entre dois módulos q -tames se os diagrams de persistência não decorados forem iguais.

Exemplo 3.37. Os módulos intervalares

$$\mathbf{k}[p, q], \mathbf{k}[p, q), \mathbf{k}(p, q], \mathbf{k}(p, q)$$

são 0^+ -intercalados, mas não isomórfos.

Proposição 3.38. Sejam $\mathfrak{U}_1, \mathfrak{U}_2, \mathfrak{V}_1, \mathfrak{V}_2$ módulos de persistência. Então

$$d_i(\mathfrak{U}_1 \oplus \mathfrak{U}_2, \mathfrak{V}_1 \oplus \mathfrak{V}_2) \leq \max(d_i(\mathfrak{U}_1, \mathfrak{V}_1), d_i(\mathfrak{U}_2, \mathfrak{V}_2)).$$

De forma geral a equação acima vale para somas diretas arbitrárias. Sejam $(\mathfrak{U}_l \mid l \in L)$ e $(\mathfrak{V}_l \mid l \in L)$ famílias de módulos de persistência indexadas em algum conjunto L . Seja

$$\mathfrak{U} = \bigoplus_{l \in L} \mathfrak{U}_l, \quad \mathfrak{V} = \bigoplus_{l \in L} \mathfrak{V}_l.$$

Então

$$d_i(\mathfrak{U}, \mathfrak{V}) \leq \sup(d_i(\mathfrak{U}_l, \mathfrak{V}_l) \mid l \in L).$$

Demonstração. Dados δ -interlaçamentos Φ_l, Ψ_l para cada par $\mathfrak{U}_l, \mathfrak{V}_l$, as aplicações $\Phi = \oplus \Phi_l, \Psi = \oplus \Psi_l$ constituem um δ -interlaçamento entre $\mathfrak{U}, \mathfrak{V}$. Portanto, qualquer cota superior de $d_i(\mathfrak{U}_l, \mathfrak{V}_l)$ é uma cota superior de $d_i(\mathfrak{U}, \mathfrak{V})$, em particular, para o \sup . \square

3.5.2 A distância bottleneck

Com a primeira distância definida, vamos definir a segunda, a *bottleneck*. Denote por

$$\tilde{\mathcal{H}}^\circ = \{ (p, q) \mid -\infty \leq p < q \leq +\infty \}$$

o semiplano estendido aberto (sem a diagonal). Para definirmos a distância *bottleneck*, precisamos definir certas relações entre os pontos em $\tilde{\mathcal{H}}^\circ$.

- **ponto a ponto:** A primeira ideia é que dois diagramas não decorados estão próximos um do outro se existe uma bijeção entre eles que não leva um ponto muito longe. Para definir uma distância entre esses pontos, usamos a métrica l^∞ no plano:

$$d^\infty((p, q), (r, s)) = \max(|p - r|, |q - s|).$$

Pontos no infinito são calculados da seguinte forma:

$$d^\infty((-\infty, q), (-\infty, s)) = |q - s|$$

$$d^\infty((p, +\infty), (r, +\infty)) = |p - r|$$

$$d^\infty((-\infty, +\infty), (-\infty, +\infty)).$$

Pontos em regiões diferentes, como (p, q) e $(-\infty, s)$, possuem distância igual a ∞ .

- **ponto a diagonal:** A outra ideia é que pontos próximos da diagonal podem ser absorvidos pela diagonal. Usando novamente a métrica l^∞ :

$$d^\infty((p, q), \Delta) = \frac{1}{2}(q - p).$$

Vamos mostrar agora duas proposições relacionadas com os dois itens acima.

Proposição 3.39. Sejam (p^*, q^*) e (r^*, s^*) intervalos e

$$\mathfrak{U} = \mathbf{k}(p^*, q^*) \text{ e } \mathfrak{V} = \mathbf{k}(r^*, s^*)$$

os módulos intervalares correspondentes. Então

$$d_i(\mathfrak{U}, \mathfrak{V}) \leq d^\infty((p, q), (r, s)).$$

Demonstração. A demonstração pode ser encontrada na página 87 em (CHAZAL *et al.*, 2016). \square

Proposição 3.40. Seja (p^*, q^*) um intervalo e

$$\mathfrak{U} = \mathbf{k}(p^*, q^*)$$

seu módulo intervalar correspondente. Denote por 0 o módulo de persistência nulo. Então

$$d_i(\mathfrak{U}, 0) = \frac{1}{2}(q - p)$$

Demonstração. Seja $\delta \geq 0$. Observe que como tomamos o módulo de persistência nulo, os únicos morfismos saindo e chegando nele são os morfismos nulos. Portanto, a única condição que precisamos checar é quando dados $\Psi\Phi = 1_{\mathfrak{U}}^{2\delta}$, temos $0 = 1_{\mathfrak{U}}^{2\delta}$. Isso vale quando $\delta > \frac{1}{2}(q - p)$ e falha para $\delta < \frac{1}{2}(q - p)$. De fato, note que as aplicações $u_{t+2\delta}^t: U_t \rightarrow U_{t+2\delta}$ são nulas precisamente quando $t < p$ ou $t + 2\delta > q$, ou seja, $\delta < \frac{1}{2}(q - t)$ e pela primeira condição $(-t > -p)$, teríamos a desigualdade desejada. \square

Utilizando os resultados e definições acima, podemos definir a distância *bottleneck* entre multiconjuntos no semiplano estendido. Neste contexto, iremos adaptar os multiconjuntos a conjuntos onde cada elemento possui um índice, ou seja, se α é um elemento de multiplicidade k , teremos os elementos $\alpha_1, \dots, \alpha_k$.

Definição 3.41. Um **emparelhamento parcial** entre os multiconjuntos A e B é uma coleção de pares

$$M \subset A \times B$$

tal que

- para todo $\alpha \in A$ existe no máximo um $\beta \in B$ tal que $(\alpha, \beta) \in M$;
- para todo $\beta \in B$ existe no máximo um $\alpha \in A$ tal que $(\alpha, \beta) \in M$.

Além disso, dizemos que um emparelhamento parcial M é um δ -emparelhamento se as seguintes condições são verdadeiras

- se $(\alpha, \beta) \in M$, então $d^\infty(\alpha, \beta) \leq \delta$;
- se $\alpha \in A$ não está emparelhado, então $d^\infty(\alpha, \Delta) \leq \delta$;
- se $\beta \in B$ não está emparelhado, então $d^\infty(\beta, \Delta) \leq \delta$.

Definição 3.42. A distância *bottleneck* entre dois multiconjuntos A, B em $\tilde{\mathcal{H}}^\circ$ é

$$d_b(A, B) = \inf \{ \delta \mid \text{Existe um } \delta\text{-emparelhamento entre } A \text{ e } B \}.$$

Proposição 3.43. A distância *bottleneck* satisfaz a desigualdade triangular

$$d_b(A, C) \leq d_b(A, B) + d_b(B, C)$$

para quaisquer multiconjuntos A, B, C .

Demonstração. Sejam M_1 um δ_1 -emparelhamento entre A, B e M_2 um δ_2 -emparelhamento entre B, C . Defina agora $\delta = \delta_1 + \delta_2$. Vamos definir agora um δ -emparelhamento M entre A e C .

Defina M como a composição de M_1 e M_2 , ou seja,

$$M = \{ (\alpha, \gamma) \mid \text{Existe um } \beta \in B \text{ tal que } (\alpha, \beta) \in M_1 \text{ e } (\beta, \gamma) \in M_2 \}.$$

Observe que M é um emparelhamento parcial pois M_1, M_2 são. Resta mostrar agora que M é o *delta*-emparelhamento que procuramos. De fato

- Se $(\alpha, \gamma) \in M$, então

$$d^\infty(\alpha, \gamma) \leq d^\infty(\alpha, \beta) + d^\infty(\beta, \gamma) \leq \delta_1 + \delta_2 = \delta,$$

em que β é o elemento da definição de M que une α a γ .

- Se α não está emparelhado em M então há duas possibilidades: α não está emparelhado em M_1 , então por definição

$$d^\infty(\alpha, \Delta) \leq \delta_1 \leq \delta,$$

e caso α está emparelhado em M_1 , $(\alpha, \beta) \in M_1$. Portanto, β não está emparelhado em M_2 , caso contrário α estaria emparelhado em M . Logo,

$$d^\infty(\alpha, \Delta) \leq d^\infty(\alpha, \beta) + d^\infty(\beta, \Delta) \leq \delta_1 + \delta_2 = \delta.$$

- Se γ não está emparelhado em M , de maneira análoga temos que

$$d^\infty(\gamma, \Delta) \leq \delta.$$

Assim mostramos que M é o δ -emparelhamento requerido.

□

Por fim, finalizamos a subseção com seu teorema mais importante:

Teorema 3.44. Sejam $\mathfrak{U}, \mathfrak{V}$ módulos de persistência decomponíveis. Então

$$d_i(\mathfrak{U}, \mathfrak{V}) \leq d_b(\text{dgm}(\mathfrak{U}), \text{dgm}(\mathfrak{V})).$$

Demonstração. Vamos mostrar que para todo δ -emparelhamento entre $\text{dgm}(\mathfrak{U})$ e $\text{dgm}(\mathfrak{V})$, temos $d_i(\mathfrak{U}, \mathfrak{V}) \leq \delta$. Após isso tomamos o infimum sobre todo δ acima e obtemos o resultado.

Seja M um δ -emparelhamento entre $\text{dgm}(\mathfrak{U})$ e $\text{dgm}(\mathfrak{V})$. Como os módulos são decomponíveis, podemos construir M a partir de um emparelhamento parcial entre os módulos intervalares de \mathfrak{U} e \mathfrak{V} .

Reescrevendo \mathfrak{U} e \mathfrak{V}

$$\mathfrak{U} = \bigoplus_{l \in L} \mathfrak{U}_l, \quad \mathfrak{V} = \bigoplus_{l \in L} \mathfrak{V}_l$$

de forma que cada par é um dos seguintes

1. um par de intervalos emparelhados;
2. \mathfrak{U}_l não está emparelhado e $\mathfrak{V}_l = 0$;
3. \mathfrak{V}_l não está emparelhado e $\mathfrak{U}_l = 0$.

Pela Proposição 3.39 e 3.40, temos $d_i(\mathfrak{U}_l, \mathfrak{V}_l) \leq \delta$. Pela Proposição 3.38, segue que $d_i(\mathfrak{U}, \mathfrak{V}) \leq \delta$. □

3.5.3 O teorema de isometria

Com as duas distâncias definidas, podemos apresentar o teorema de estabilidade dos módulos de persistência.

Teorema 3.45. Sejam $\mathfrak{U}, \mathfrak{V}$ módulos de persistência q -tame. Então

$$d_i(\mathfrak{U}, \mathfrak{V}) = d_b(\text{dgm}(\mathfrak{U}), \text{dgm}(\mathfrak{V}))$$

O resultado pode ser dividido em duas partes. A primeira parte seria o teorema de estabilidade conhecido na literatura (COHEN-STEINER; EDELSBRUNNER; HARER, 2006; CHAZAL *et al.*, 2009)

$$d_i(\mathfrak{U}, \mathfrak{V}) \geq d_b(\text{dgm}(\mathfrak{U}), \text{dgm}(\mathfrak{V})), \tag{3.8}$$

e a volta do teorema de estabilidade

$$d_i(\mathfrak{U}, \mathfrak{V}) \leq d_b(\text{dgm}(\mathfrak{U}), \text{dgm}(\mathfrak{V})). \quad (3.9)$$

Nesta dissertação iremos mostrar apenas a volta, já que a volta envolve argumentos muito complexos que fogem do escopo da exposição. Anteriormente mostramos o resultado para módulos decomponíveis. Agora vamos mostrar para módulos q -tame que não sabemos ser decomponíveis.

3.5.4 A volta do teorema de estabilidade

Agora vamos deduzir a volta do teorema de estabilidade, dado pela Inequação 3.9. A ideia principal é aproximar um módulo de persistência por um outro módulo "suave".

Definição 3.46. Seja \mathfrak{V} um módulo de persistência e $\varepsilon > 0$. A ε -suavização de \mathfrak{V} é o módulo de persistência \mathfrak{V}^ε definido como a imagem da aplicação

$$1_{\mathfrak{V}}^{2\varepsilon}: V[-\varepsilon] \rightarrow \mathfrak{V}[\varepsilon].$$

Ou seja, $(V^\varepsilon)_t$ é a imagem da aplicação

$$v_{t+\varepsilon}^{t-\varepsilon}: V_{t-\varepsilon} \rightarrow V_{t+\varepsilon},$$

e $(v^\varepsilon)_t^s$ é a restrição de $v_{t+\varepsilon}^{s+\varepsilon}$. Portanto, temos a fatorização de $1_{\mathfrak{V}}^{2\varepsilon}$ dado por

$$\mathfrak{V}[-\varepsilon] \longrightarrow V^\varepsilon \longrightarrow \mathfrak{V}[\varepsilon], \quad (3.10)$$

em que a primeira aplicação é sobrejetora e a segunda é injetora. Dado um índice t , tem-se que

$$V_{t-\varepsilon} \xrightarrow{v_{t+\varepsilon}^{t-\varepsilon}} V_t^\varepsilon \xrightarrow{1} V_{t+\varepsilon}$$

Proposição 3.47. Seja \mathfrak{V} um módulo de persistência. Então $d_i(\mathfrak{V}, \mathfrak{V}^\varepsilon) \leq \varepsilon$.

Demonstração. O morfismo dado em 3.10 é um ε -interlçamento. □

Exemplo 3.48. Seja $\mathfrak{V} = \mathbf{k}(p^*, q^*)$. Então a ε -suavização de \mathfrak{V} é dada por

$$\mathfrak{V}^\varepsilon = \begin{cases} \mathbf{k}((p+\varepsilon)^*, (q-\varepsilon)^*), & \text{se } (p+\varepsilon)^* < (q-\varepsilon)^* \\ 0, & \text{caso contrário.} \end{cases}$$

Em outras palavras, a ε -suavização diminui o intervalo em ambos os lados.

Proposição 3.49. O diagrama de persistência de \mathfrak{V}^ε é obtido através do diagrama de persistência de \mathfrak{V} pela translação $T_\varepsilon: (p, q) \mapsto (p+\varepsilon, q-\varepsilon)$ para a parte do semiplano acima da linha $\Delta_\varepsilon = \{ (t-\varepsilon, t+\varepsilon) \mid t \in \mathbb{R} \}$.

Demonstração. Considere os dois casos: quando \mathfrak{V} é decomponível e o caso geral.

Primeiro caso: Suponha que \mathfrak{V} seja decomponível, então seja $\mathfrak{V} = \oplus \mathfrak{V}_l$ a decomposição de \mathfrak{V} em módulos intervalares. Como soma direta é linear sob aplicações lineares, segue que a ε -suavização de \mathfrak{V} é a soma direta das ε -suavizações de cada \mathfrak{V}_l

$$\left[\bigoplus_{l \in L} \mathfrak{V}_l \right]^\varepsilon = \bigoplus_{l \in L} \mathfrak{V}_l^\varepsilon.$$

Logo, pelo Exemplo 3.48, segue o resultado.

Segundo caso: Vamos mostrar que a medida de persistência de \mathfrak{V}^ε é igual a medida de persistência de \mathfrak{V} transladada por T_ε . Defina

$$A = a - \varepsilon, \quad B = b - \varepsilon, \quad C = c + \varepsilon, \quad D = d + \varepsilon.$$

Queremos mostrar então que

$$\langle \circ_A \text{---} \bullet_B \text{---} \bullet_C \text{---} \circ_D \mid \mathfrak{V} \rangle = \langle \circ_a \text{---} \bullet_b \text{---} \bullet_c \text{---} \circ_d \mid \mathfrak{V}^\varepsilon \rangle,$$

para todo $a < b \leq c < d$.

O resto da demonstração é baseado no seguinte diagrama comutativo:

$$\begin{array}{ccccccc} & & (V^\varepsilon)_a & \longrightarrow & (V^\varepsilon)_b & \longrightarrow & (V^\varepsilon)_c & \longrightarrow & (V^\varepsilon)_d \\ & \nearrow & & & \nearrow & & \searrow & & \searrow \\ V_A & \longrightarrow & V_B & \longrightarrow & V_C & \longrightarrow & V_D \end{array}$$

onde as aplicações \nearrow são sobrejetoras e \searrow são injetoras. As setas induzem um poset, então o diagrama pode ser visto como um módulo de persistência sobre esse poset com 8 elementos.

Para as aplicações sobrejetoras \nearrow , temos que

$$\langle \circ_A \text{---} \bullet_a \rangle = 0 \text{ e } \langle \circ_B \text{---} \bullet_b \rangle = 0, \quad (3.11)$$

já para as aplicações injetoras \searrow

$$\langle \bullet_c \text{---} \circ_C \rangle = 0 \text{ e } \langle \bullet_d \text{---} \circ_D \rangle = 0. \quad (3.12)$$

Além disso, os módulos intervalares contendo alguma das configurações dada pelas equações acima possuem multiplicidade zero pelo princípio de restrição. Portanto

$$\begin{aligned} \langle \circ_A \text{---} \bullet_b \text{---} \bullet_c \text{---} \circ_D \rangle &= \langle \circ_A \text{---} \bullet_B \text{---} \bullet_b \text{---} \bullet_c \text{---} \bullet_C \text{---} \circ_D \rangle + \text{outros três termos} \\ &= \langle \circ_A \text{---} \bullet_B \text{---} \bullet_b \text{---} \bullet_c \text{---} \bullet_C \text{---} \circ_D \rangle \\ &= \langle \circ_A \text{---} \bullet_B \text{---} \bullet_C \text{---} \circ_D \rangle \\ &= \langle \circ_A \text{---} \bullet_B \text{---} \bullet_C \text{---} \circ_D \mid \mathfrak{V} \rangle, \end{aligned}$$

e analogamente

$$\begin{aligned}
\langle \circ_A \text{---} \bullet_b \text{---} \bullet_c \text{---} \circ_D \rangle &= \langle \circ_A \text{---} \circ_a \text{---} \bullet_b \text{---} \bullet_c \text{---} \circ_d \text{---} \circ_D \rangle + \text{outros tr\^es termos} \\
&= \langle \circ_A \text{---} \circ_a \text{---} \bullet_b \text{---} \bullet_c \text{---} \circ_d \text{---} \circ_D \rangle \\
&= \langle \text{---} \circ_a \text{---} \bullet_b \text{---} \circ_c \text{---} \circ_d \text{---} \rangle \\
&= \langle \circ_a \text{---} \bullet_b \text{---} \bullet_c \text{---} \circ_d \mid \mathfrak{V}^\varepsilon \rangle.
\end{aligned}$$

Logo temos a igualdade. Os seis termos extras s\~ao todos nulos pois eles cont\~em as configura\~oes (3.11) e (3.12). \square

Corol\'ario 3.50. Seja \mathfrak{V} q -tame, ent\~ao $d_b(\text{dgm}(\mathfrak{V}), \text{dgm}(\mathfrak{V}^\varepsilon)) \leq \varepsilon$.

Demonstra\~ao. Seja o ε -emparelhamento definido a seguir

$$(p, q) \in \text{dgm}(\mathfrak{V}^\varepsilon) \iff (p - \varepsilon, q + \varepsilon) \in \text{dgm}(\mathfrak{V}).$$

Note que isso \'{e} uma bije\~cao para todos os pontos, exceto para aquele que n\~ao foram emparelhados em $\text{dgm}(\mathfrak{V})$, mas estes s\~ao justamente os pontos que ficam na linha Δ_ε ou abaixo dela. Portanto, possuem uma dist\~ancia de no m\~aximo ε da diagonal. \square

Proposi\~ao 3.51. Se \mathfrak{V} \'{e} q -tame, ent\~ao \mathfrak{V}^ε \'{e} localmente finito.

Demonstra\~ao. Como $\dim((V^\varepsilon)_t) = \text{rank}[V_{t-\varepsilon} \rightarrow V_{t+\varepsilon}] < \infty$, segue do Teorema 3.7 que \mathfrak{V}^ε \'{e} decompon\'ivel em m\'odulos intervalares. Vamos mostrar agora que essa cole\~ao de intervalos \'{e} localmente finita.

Seja $t \in \mathbb{R}$ qualquer, ent\~ao

$$\begin{aligned}
&\# \left\{ \text{intervalos que intersecta } [t - \frac{1}{2}\varepsilon, t + \frac{1}{2}\varepsilon] \right\} \\
&= \# \left\{ \text{pontos em } \text{dgm}(\mathfrak{V}^\varepsilon) \text{ no quadrante superior esquerdo a partir do ponto } (t - \frac{1}{2}\varepsilon, t + \frac{1}{2}\varepsilon) \right\} \\
&\leq \# \left\{ \text{pontos em } \text{dgm}(\mathfrak{V}) \text{ no quadrante superior esquerdo a partir do ponto } (t - \frac{1}{2}\varepsilon, t + \frac{1}{2}\varepsilon) \right\} \\
&= \text{rank}[V_{t-\varepsilon} \rightarrow V_{t+\varepsilon}] < \infty
\end{aligned}$$

Temos que a inequa\~ao da terceira linha segue da Proposi\~ao 3.49. \square

Agora podemos demonstrar a inequa\~ao 3.9.

Demonstra\~ao. 3.9 Sejam $\mathfrak{U}, \mathfrak{V}$ m\'odulos de persist\~encia q -tames. Para qualquer $\varepsilon > 0$, a ε -suaviza\~ao de $\mathfrak{U}^\varepsilon, \mathfrak{V}^\varepsilon$ s\~ao decompon\'iveis, ent\~ao a volta do teorema de estabilidade se

aplica a eles. Portanto

$$\begin{aligned} d_i(\mathfrak{U}, \mathfrak{V}) &\leq d_i(\mathfrak{U}^\varepsilon, \mathfrak{V}^\varepsilon) + 2\varepsilon && \text{Proposição 3.47} \\ &\leq d_b(\text{dgm}(\mathfrak{U}^\varepsilon), \text{dgm}(\mathfrak{V}^\varepsilon)) + 2\varepsilon && \text{Teorema 3.44} \\ &\leq d_b(\text{dgm}(\mathfrak{U}), \text{dgm}(\mathfrak{V})) + \varepsilon && \text{Corolário 3.50} \end{aligned}$$

Como isso vale para todo $\varepsilon > 0$, temos que

$$d_i(\mathfrak{U}, \mathfrak{V}) \leq d_b(\text{dgm}(\mathfrak{U}), \text{dgm}(\mathfrak{V})).$$

□

3.5.5 O teorema de estabilidade

A Inequação 3.8 pode ser expressa da seguinte maneira

Teorema 3.52. Sejam $\mathfrak{U}, \mathfrak{V}$ módulos de persistência q -tames que são δ^+ -intercalados. Então existe um δ -emparelhamento entre os multiconjuntos $\text{dgm}(\mathfrak{U})$ e $\text{dgm}(\mathfrak{V})$.

Podemos, no entanto, provar a seguinte forma do teorema.

Teorema 3.53. Sejam $\mathfrak{U}, \mathfrak{V}$ módulos de persistência q -tames que são δ -intercalados. Então existe um δ -emparelhamento entre os multiconjuntos $\text{dgm}(\mathfrak{U})$ e $\text{dgm}(\mathfrak{V})$.

Podemos deduzir o Teorema 3.52 de 3.53 pois se $\mathfrak{U}, \mathfrak{V}$ são δ^+ -intercalados, então existe um η -interlaçamento entre os respectivos diagramas para todo $\eta > \delta$, então existe um δ -emparelhamento. O Teorema 3.53 pode ser provado usando os seguintes resultados:

- O Lema de Interpolação 3.33;
- As inequações do Lema 3.55, que relacionam localmente as medidas de $\mathfrak{U}, \mathfrak{V}$.

Iremos apenas apresentar o Lema 3.55. Sua demonstração se encontra em (CHAZAL *et al.*, 2016). Utiliza-se o método de continuidade apresentado em (COHEN-STEINER; EDELSBRUNNER; HARER, 2006) para concluir a demonstração do Teorema 3.53. Não iremos apresentar a demonstração por completo aqui devido a sua dificuldade.

Definição 3.54. Sejam $R = [a, b] \times [c, d]$ um retângulo em \mathbb{R}^2 . O δ -engrossamento de R é o retângulo

$$R^\delta = [a - \delta, b + \delta] \times [c - \delta, d + \delta].$$

Lema 3.55. Sejam $\mathfrak{U}, \mathfrak{V}$ módulos de persistência δ -intercalados. Seja R um retângulo cujo δ -engrossamento R^δ fique acima da diagonal. Então $\mu_{\mathfrak{U}}(R) \leq \mu_{\mathfrak{V}}(R^\delta)$ e $\mu_{\mathfrak{V}}(R) \leq \mu_{\mathfrak{U}}(R^\delta)$.

GERADORES ÓTIMOS E OUTROS CONCEITOS

Neste capítulo iremos descrever alguns métodos da análise topológica de dados para tratar os diagramas de persistência.

O primeiro dos métodos é o gerador ótimo. Para cada ponto no diagrama de persistência, temos um ciclo associado. A ideia é analisar este ciclo geometricamente e extrair informação desse maneira. O segundo é a imagem de persistência, um método de vetorização do diagrama de persistência, de forma que ele possa ser usado para prever e classificar conjuntos de dados. Por último, tratamos *mapper*, uma ferramenta da análise topológica de dados para visualização de conjuntos de alta dimensão em algum espaço de baixa dimensão.

4.1 Geradores ótimos

O diagrama de persistência nos dá informação sobre buracos e cavidades que persistentem ao longo de uma filtração. Para cada buraco temos um ciclo associado, um representante da classe homológica nos grupos de homologia da filtração. É possível visualizar esses ciclos através dos simplexos do complexo simplicial associado à filtração e ao diagrama. Geralmente eles podem nos dar informações geométricas, como a localidade de buracos, revelando informações escondidas.

Entretanto, os ciclos por várias vezes não representam de forma ótima uma propriedade do conjunto. Na Figura 20 podemos ver que um ciclo é maior do que poderia ser em relação ao número de simplexos, não representando de maneira ótima o buraco relacionado.

A seguir apresentamos a ideia de geradores ótimos, que trata de ajustar o problema

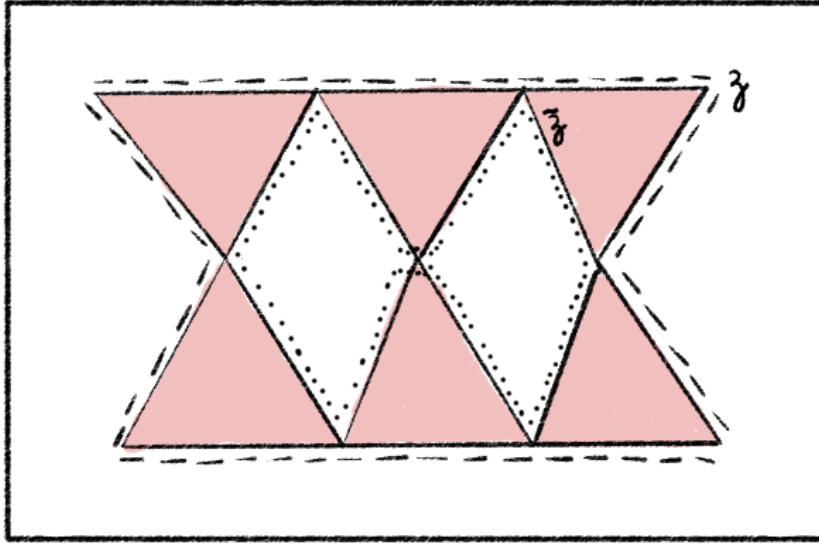


Figura 20 – Dois ciclos homólogos que representam o buraco. Note que o ciclo que representa o buraco não é ótimo no número de simplexos. O ciclo ótimo é o com linhas pontilhadas.

Fonte: Elaborada pelo autor.

de melhor gerador. Começamos com o problema de otimização de um único ciclo e depois apresentamos para múltiplos geradores. Esta seção tem como referência ([ESCOLAR; HIRAOKA, 2015](#)).

4.1.1 Único Gerador

Seja X um complexo simplicial qualquer e denote por $Z_q(X)$ o conjunto dos q -ciclos de X , com respeito ao operador bordo ∂_q e denote por $B_q(X)$ o conjunto dos q -bordos de X , em que $B_q(X) = \text{im} \partial_{q+1}$.

Para cada q , considere $\{\sigma_1, \dots, \sigma_N\}$ conjunto de q -simplexos de X como a base de $C_q(X)$, grupo de todas as q -cadeias de X . Então representamos todo $x = \sum x_i \sigma_i \in C_q(X)$ por um vetor $[x_1, \dots, x_N]^T$.

Seja agora $z \in Z_q(X)$, considere o problema

$$\begin{aligned} & \text{minimize} && \|x\|_1 \\ & \text{sujeito a} && \begin{cases} x - \partial_{q+1} y = z \\ x, y \text{ inteiros.} \end{cases} \end{aligned} \quad (4.1)$$

Aqui tentamos achar um ciclo \tilde{z} homólogo a z que possui a menor 1-norma entre todas as cadeias homólogas a z . A 1-norma é definida por $\|\sum_i x_i \sigma_i\|_1 = \sum_i |x_i|$.

Podemos no entanto alterar o problema. Ao invés de considerar x um vetor inteiro qualquer, vamos restringir x para valores em $\{-1, 0, 1\}$, facilitando a interpretação geo-

métrica de \tilde{z} . Uma consequência disso é $\|x\|_0 = \|x\|_1$, em que $\|x\|_0 = |\{x_i \neq 0\}|$ em $C_q(X)$. Adicionamos mais uma condição, de que para $\tilde{z} = \sum_{\sigma} n_{\sigma} \sigma$, $n_{\sigma} \in \{-1, 0, 1\}$. Sendo assim, podemos formular o problema anterior da seguinte maneira:

$$\begin{aligned} & \text{minimize} && \|x\|_1 \\ & \text{sujeito a} && \begin{cases} x - \partial_{q+1}y = z \\ x \text{ é um vetor com entradas em } \{-1, 0, 1\} \text{ e } y \text{ é vetor inteiro} \end{cases} \end{aligned} \quad (4.2)$$

Se z é um vetor com entradas em $\{-1, 0, 1\}$, então existe uma solução para o problema (4.2) (DEY; HIRANI; KRISHNAMOORTHY, 2010). Escreva também x como $x^+ - x^-$, $x^+, x^- \geq 0$ correspondendo as partes positiva e negativa de x espectivamente. Então podemos reescrever o problema como

$$\begin{aligned} & \text{minimize} && \|x\|_1 = \sum_{i=1}^N (x_i^+ + x_i^-) \\ & \text{sujeito a} && \begin{cases} (x^+ - x^-) - \partial_{q+1}y = z \\ x^+, x^- \text{ com entradas em } \{0, 1\} \text{ e } y \text{ é inteiro,} \end{cases} \end{aligned}$$

em que x_i^+, x_i^- são as entradas dos vetors x^+, x^- respectivamente.

A integralidade das soluções não é garantida, precisamos considerar uma restrição a mais. Uma matrix é dita unimodular se o determinante de cada submatriz for $-1, 0$ ou 1 . Então podemos garantir uma solução se não exigirmos que \tilde{z} seja inteiro e considerar o problema de programação linear sobre os reais. Então a unimodularidade total da matriz de restrição do problema garantirá que o problema possuirá solução inteira. Vamos escrever ambos os problemas de seguinte forma:

$$\begin{aligned} & \text{minimize} && c'x \\ & \text{sujeito a} && \begin{cases} Ax = b \\ x \geq 0, \text{ inteiro.} \end{cases} \end{aligned}$$

Para o Problema (4.1), temos

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^N (x_i^+ + x_i^-) \\ & \text{sujeito a} && \begin{cases} x^+ - x^- - \partial_{q+1}(y^+ - y^-) = z \\ x^+, x^-, y^+, y^- \geq 0, \text{ inteiros,} \end{cases} \end{aligned}$$

e possui uma matriz de restrição $A = \begin{bmatrix} I & -I & -\partial_{q+1} & \partial_{q+1} \end{bmatrix}$. O Problema (4.2) também pode ser escrito da forma acima, bastando colocar a restrição de que x é um vetor com entradas em $\{-1, 0, 1\}$ na matriz A acima. Então se ∂_{q+1} é unimodular, temos que a matriz A também é.

Para um $q \geq 0$ fixado, existem condições para que ∂_{q+1} seja totalmente unimodular (DEY; HIRANI; KRISHNAMOORTHY, 2010). Por exemplo, se X é um complexo simplicial finito triangulando uma variedade compacta de dimensão $q+1$ ou X é um complexo simplicial finito mergulhado em \mathbb{R}^{q+1} , então ∂_{q+1} é unimodular.

4.1.2 Múltiplos geradores

Nem sempre otimizar apenas um gerador é suficiente. Como podemos ver na Figura 20, os dois ciclos não são os ótimos, mesmo após o processo de otimização. Podemos dizer intuitivamente que o ciclo está emperrado entre os dois buracos. Vamos modificar a proposta e resolver este problema.

Seja $\{g_1, \dots, g_m\}$ um conjunto de ciclos. Considere agora o seguinte problema

$$\begin{aligned} & \text{minimize} \quad \|x\|_1 \\ & \text{sujeito a} \quad \begin{cases} x - \partial_{q+1}(y) + \sum_{j=1}^m a_j g_j = z \\ x, y \text{ e } a \text{ inteiros,} \end{cases} \end{aligned}$$

e seja $P(z; g_1, \dots, g_m)$ o conjunto de soluções ótimas projetado na variável x do problema acima. Linearizando o problema, obtemos

$$\begin{aligned} & \text{minimize} \quad \sum_{i=1}^N (x_i^+ + x_i^-) \\ & \text{sujeito a} \quad \begin{cases} x^+ - x^- - \partial_{q+1}(y) + \sum_{j=1}^m a_j g_j = z \\ x^+, x^- \geq 0, x^+, x^-, y, \text{ e } a \text{ inteiros.} \end{cases} \end{aligned}$$

Vamos agora entender os ciclos g_1, \dots, g_m . Estes são os ciclos que gostaríamos de tirar enquanto otimizando o ciclo z , em outras palavras, preencheríamos os buracos que ficam entre z e o seu ciclo ótimo, onde esses buracos são representados por g_i , para todo i . Estes ciclos são chamados de ciclos relativos e a soma $\sum a_j g_j$ pode ser entendida da seguinte forma. Seja

$$\partial'_{q+1} = \begin{bmatrix} \partial_{q+1} & g_1 & \dots & g_m \end{bmatrix}$$

a matriz de bordo com colunas a mais, g_1, \dots, g_m . Geometricamente, estamos adicionando células τ_j em X_{q+1} de forma que $\partial'_{q+1} \tau_j = g_j$ para $j = 1, \dots, m$. Como dito anteriormente, cada célula τ_j cobre uma propriedade topológica, permitindo que z passe por g_j .

Seja agora $\{z_1, \dots, z_n\}$ conjunto gerador de $H_q(X)$, grupo de homologia de X . Portanto, se trocarmos z_j por algum

$$\tilde{z}_j \in P(z_j; z_1, \dots, z_{j-1}, z_{j+1}, \dots, z_n),$$

temos que $\{z_1, \dots, \tilde{z}_j, \dots, z_n\}$ é um novo conjunto gerador de $H_q(X)$, já que

$$[\tilde{z}_j] = [z_j] + \sum_{i \neq j} a_i [z_i],$$

pelas restrições do problema de otimização. Portanto, temos o Algoritmo 2.

Algoritmo 2 – Procedimento de otimização dos geradores.**Requer:** Geradores z_1, \dots, z_m de $H_q(X)$

```

1: função OPTIMIZE_CYCLES( $\{z_1, \dots, z_n\}$ )
2:   para  $j \in \{1, \dots, n\}$  faça
3:     Escolha  $\tilde{z}_j$  de  $P(z_j; \tilde{z}_1, \dots, \tilde{z}_{j-1}, z_{j+1}, \dots, z_n)$ 
4:   fim para
5:   retorna  $\{\tilde{z}_1, \dots, \tilde{z}_n\}$ 
6: fim função

```

4.1.3 Geradores ótimos em homologia persistente

Seja $\emptyset = X_0 \subset X_1 \subset \dots \subset X_N$ uma filtração com a propriedade de que em cada índice, apenas um simplexo é adicionado, ou seja, se σ_j é um simplexo, então $X_j \setminus X_{j-1} = \sigma_j$, dessa forma cada simplexo causa o nascimento ou morte de uma classe de homologia. Também para cada simplexo em X_N , temos um índice j de nascimento único.

Vamos agora descrever os passos para otimizar os ciclos gerados utilizando o algoritmo de persistência. Denote por A_j o bordo do simplexo σ_j , $A_j = \partial \sigma_j$. O bordo do simplexo é representado como um vetor na base de todos os simplexos de X , ordenados de acordo com o índice da filtração. A i -ésima entrada de A_j é denotada por $A_j(i)$. Para qualquer coluna A_j , o seu pivot é denotado por $\text{pivot}(A_j)$ e representa o maior inteiro i tal que $A_j(i) \neq 0$.

O Algoritmo 3 calcula a persistência da filtração X . Note que neste caso específico utilizamos coeficientes em \mathbb{Q} .

Para cada p , o valor $l(p)$ registra na tabela l o índice da coluna reduzida que possui o pivot na linha p . Temos então dois casos, se A_j é reduzida a uma coluna nula, então temos o nascimento de um ciclo g_j . Caso contrário, temos a morte de um ciclo que nasceu em $i = \text{pivot}(A_j)$. Além disso, a mudança na base é atualizada e salva nos ciclos g_j . No final do algoritmo todo índice i é pareado com algum índice j ou não é. Sendo assim, denote por

$$d(i) = \begin{cases} j & \text{se } (i, j) \text{ é o par de } i, \\ \infty & \text{se } i \text{ não é pareado,} \end{cases}$$

o índice de morte do ciclo que nasce no tempo i . Defina então

$$\mathcal{L}_q(k) = \{i \mid i \leq k, A_i = 0, \dim \sigma_i = q, d(i) > k\}.$$

O conjunto de índices acima representa todos os ciclos que nascem antes de σ_k e não se tornam parte do borde de X_k . Pode-se mostrar que

$$\{[g_i] \mid i \in \mathcal{L}_q(k)\}$$

forma uma base de $H_q(X_k)$.

Algoritmo 3 – Algoritmo para calcular o diagrama de persistência de uma filtração.

```

1: procedimento COMPUTE_PERSISTENCE( $X$ )
2:   Inicialize  $g_j = \sigma_j$  para todo  $j = 1, \dots, N$ .
3:   para  $j = 1, \dots, N$  faça
4:     enquanto  $A_j \neq 0$  e  $l(\text{pivot}(A_j)) \neq 0$  faça
5:        $p \leftarrow \text{pivot}(A_j)$ 
6:        $r \leftarrow -\frac{A_j(p)}{A_{l(p)}(p)}$ 
7:        $A_j \leftarrow +r \cdot A_{l(p)}$ 
8:        $g_j \leftarrow +r \cdot g_{l(p)}$ 
9:     fim enquanto
10:    se  $A_j \neq 0$  então
11:       $i \leftarrow \text{pivot}(A_j)$ 
12:       $l(i) \leftarrow j$ 
13:      Insira o par  $(i, j)$  na matriz dgm
14:    fim se
15:  fim para
16:  para todo  $i$  tal que  $i \neq b, d$  para todo  $(b, d) \in \text{dgm}$  faça
17:    Insira  $(i, \infty)$  em dgm.
18:  fim para
19:  retorna dgm
20: fim procedimento

```

Com a dimensão q fixada, o Algoritmo 4 calcula os ciclos ótimos logo após o seu nascimento.

Quando um simplexo σ_j de dimensão q é encontrado, atualizamos o problema de otimização com uma nova variável e uma restrição correspondendo ao seu ciclo. No Problema (4.3) a variável x aumenta em um no seu tamanho e as restrições são completadas com zeros.

Se a coluna A_j é zerada, isso significa o nascimento de um novo ciclo. Se sua dimensão for q , então resolvemos o problema de otimização com o ciclo $g_j = z_j$ que acabou de nascer. Uma vez com o ciclo otimizado \tilde{z}_j , o adicionamos no problema de otimização como um ciclo relativo.

Se a coluna A_j não for zerada, isso sinaliza a morte de um ciclo e se $\dim \sigma_j = q + 1$, então adicionamos a coluna A_j no problema de otimização como uma coluna no final da matriz de bordo B . Além disso, se $\text{low}(j) = i$, isso significa que o ciclo \tilde{z}_i de dimensão q faz parte do bordo. Portanto, o ciclo \tilde{z}_i não é mais necessário no processo de otimização, e assim o removemos do problema.

No Teorema 4.1 mostramos que os ciclos otimizados forma uma base para cada grupo de homologia na filtração de X , mantendo a consistência da informação topológica dada pelo diagrama de persistência.

Algoritmo 4 – Algoritmo para calcular o diagrama de persistência de uma filtração e os ciclos ótimos.

```

1: procedimento PERSISTENCE__OPTIMAL__CYCLES( $X, q$ )
2:   Inicialize uma tabela  $l$ 
3:    $B$ , uma matriz vazia
4:   Inicialize  $g_j = \sigma_j$  para todo  $j = 1, \dots, N$ .
5:   para  $j = 1, \dots, N$  faça
6:     se  $\dim \sigma_j = q$  então
7:       Atualize o problema de otimização com o novo simplexo  $\sigma_j$ 
8:     fim se
9:     enquanto  $A_j \neq 0$  e  $l(\text{pivot}(A_j)) \neq 0$  faça
10:       $p \leftarrow \text{pivot}(A_j)$ 
11:       $r \leftarrow -\frac{A_j(p)}{A_{l(p)}(p)}$ 
12:       $A_j \leftarrow +r \cdot A_{l(p)}$ 
13:       $g_j \leftarrow +r \cdot g_{l(p)}$ 
14:    fim enquanto
15:    se  $A_j = 0$  e  $\dim \sigma_j = 0$  então
16:       $\tilde{z}_j = \text{OPTIMIZE\_CYCLE}(g_j)$ 
17:      Atualize o problema de otimização com  $\tilde{z}_j$ 
18:    fim se
19:    se  $A_j \neq 0$  então
20:       $i \leftarrow \text{pivot}(A_j)$ 
21:       $l(i) \leftarrow j$ 
22:      Insira o par  $(i, j)$  na matriz dgm
23:      se  $\dim \sigma_j = q + 1$  então
24:        Adicione  $A_j$  a matrix  $B$  (como uma coluna de bordo)
25:        Remova  $\tilde{z}_i$  do problema de otimização
26:      fim se
27:    fim se
28:  fim para
29:  para todo  $i$  tal que  $i \neq b, d$  para todo  $(b, d) \in \text{dgm}$  faça
30:    Insira  $(i, \infty)$  em dgm.
31:  fim para
32:  retorna dgm
33: fim procedimento

```

Algoritmo 5 – Procedimento para encontrar o ciclo ótimo.

- 1: **procedimento** OPTIMIZE_CYCLE($z_j = g_j$)
 2: Encontre uma solução ótima \tilde{z}_j para

$$\begin{array}{ll} \text{minimize} & \|x\|_1 \\ \text{sujeito a} & \left\{ \begin{array}{l} x + By + \sum_{i \in \mathcal{L}_q(j), i < j} a_i \tilde{z}_i = z_j \end{array} \right. \end{array} \quad (4.3)$$

- 3: **retorna** \tilde{z}_j
 4: **fim procedimento**
-

Teorema 4.1. Seja q um número inteiro não negativo. Sejam então $\tilde{z}_1, \dots, \tilde{z}_m$ os ciclos ótimos gerados pelo Algoritmo 4. Então $\{[\tilde{z}_i] \mid i \in \mathcal{L}_q(k)\}$ forma uma base de $H_q(X)$.

Demonstração. A classe $[\tilde{z}_i] \in H_q(X_k)$ satisfaz,

$$[\tilde{z}_i] = [g_i] + \sum_{h \in \mathcal{L}_q(i), h < i} a_h [\tilde{z}_h]$$

pelo Problema 4.3 e pelo fato de que $B = \text{img } \partial_{q+1}|_{X_k}$. De maneira similar para $[\tilde{z}_h]$,

$$[\tilde{z}_i] = [g_i] + \sum_{h < i, A_h = 0, \dim \sigma_h = q} c_h [g_h]. \quad (4.4)$$

O subíndice da soma em (4.4) se refere a todos os ciclos que nascem antes de i e possuem dimensão q . Agora queremos os índices h tais que $d(h) > k$, de forma que obtemos a soma

$$[\tilde{z}_i] = [g_i] + \sum_{h \in \mathcal{L}_q(k), h < i} . \quad (4.5)$$

Quando expandimos cada $[\tilde{z}_h] = [g_h] + \sum_{h' \in \mathcal{L}_q(k)} a'_{h'} [\tilde{z}'_{h'}]$, não sabemos se para h' e seus ciclos correspondentes $g_{h'}$ vivem até o índice k . Para aqueles h' com $d(h') \leq k$, $[g_{h'}]$, já que eles morrem antes ou quando entram em $H_q(X_k)$. Então, para os índices h que satisfazem essa propriedade, $[g_h]$ entra no bordo e podemos removê-los da soma em (4.4). Logo, obtemos (4.5).

Agora, a transformação induzida por (4.5) é invertível, pois a condição $h < i$ implica que a matriz de transformação é triangular com 1's na diagonal. \square

A implementação deste método foi feita pelos autores do artigo original (ESCOLAR; HIRAOKA, 2015) e pode ser encontrada no site: <https://bitbucket.org/remere/optiperslp>.

4.2 Vetorização do diagrama de persistência

Dado uma sequência de conjuntos de dados X_i e os respectivos diagramas de persistência tem-se uma variação no seus tamanhos, devido a natureza do algoritmo de homologia persistente. Além de variação entre os tamanhos, cada diagrama é um multi-conjunto, sendo mais difícil de analisa-los. Ao utilizar algoritmos de machine learning, assume-se entradas com tamanhos fixos no conjunto inteiro de dados. Portanto diagramas de persistências descrevendo uma sequência de proteínas, por exemplo, precisam ser vetorizados de alguma forma antes de podermos utilizar em conjunto com outros algoritmos, como redes neurais ou regressão linear.

Existem várias formas de vetorização de um diagrama de persistência, como *Persistence landscapes* (BUBENIK, 2015) e Imagem de persistência (*Persistence Image*) (ADAMS et al., 2017). Neste trabalho apresentamos a imagem de persistência e alguns exemplos.

4.2.1 Estabilidade da Imagem de Persistência

Imagem de persistência é uma vetorização de forma que o respectivo diagrama é representado como uma imagem de tamanho fixo (n, m) . De forma intuitiva esse método é uma forma de suavização do diagrama de persistência, em que uma gaussiana é centrada em cada ponto e depois são somadas com peso. Abaixo descrevemos formalmente o processo para obter uma imagem de persistência.

Seja $D = \{(b_i, d_i)\}_i$ um diagrama de persistência em alguma dimensão e considere $T(x, y) = (x, y - x)$ uma transformação linear em \mathbb{R}^2 . Seja $T(B)$ o multiconjunto decorrente da transformação linear T aplicada em B onde cada ponto $(x, y) \in B$ corresponde ao ponto $(x, y - x) \in T(B)$. Considere agora uma função de probabilidade diferenciável $\phi_u: \mathbb{R}^2 \rightarrow \mathbb{R}$ com média $u = (u_x, u_y)$.

Fixe agora uma função peso $f: \mathbb{R}^2 \rightarrow \mathbb{R}$ de tal forma que ela é zero no eixo horizontal, contínua e diferenciável por partes. É importante que essas condições sejam satisfeitas, pois elas garante a estabilidade da imagem de persistência sob a distância 1-Wasserstein. Dessa forma temos a seguinte definição.

Definição 4.2. Para um diagrama de persistência B , a correspondente superfície de persistência $\rho_B: \mathbb{R}^2 \rightarrow \mathbb{R}$ é a função dada por

$$\rho_B(z) = \sum_{u \in T(B)} f(u) \phi_u(z).$$

Entretanto, um computador não consegue utilizar uma função para fazer cálculos e estimativas, ela precisa ser vetorizada (ou discretizada) de alguma forma. Desta forma, vamos discretizar ρ_B em um domínio específico, que depende de $T(B)$. Em específico, fixamos um grid e o valor de cada pixel é dado pela integral nessa região.

Definição 4.3. Seja B um diagrama de persistência. A imagem de persistência de B é a coleção de pixels

$$I(\rho_B)_p = \iint_p \rho_B dy dx,$$

em que p é a região do pixel da imagem.

Na vetorização do diagrama alguns parâmetros precisam ser estabelecidos. Em (ADAMS et al., 2017) mostra-se que as imagens são robustas sob a escolha da resolução

(tamanho do grid). A outra escolha é a distribuição. Em (ADAMS *et al.*, 2017) a distribuição gaussiana é utilizada com variância dependendo do problema e sendo assim o usuário a escolhe. Por último, a escolha da função peso, que pode variar de problema pra problema. A função abaixo é um exemplo utilizado por (ADAMS *et al.*, 2017). Observe que para pontos com valores altos de persistência tem um valor maior também. Mas em alguns problemas os pontos de baixa ou média persistência são importantes, então a utilização de outras funções se faz necessária.

$$w_b(t) = \begin{cases} 0 & \text{if } t \leq 0, \\ \frac{t}{b} & \text{if } 0 < t < b, \\ 1 & \text{if } t \geq b, \end{cases} \quad (4.6)$$

onde b é considerado o valor de maior persistência em $T(B)$.

Em vários conjuntos é normal que apresentem ruídos e algumas variações, assim dando diagramas de persistência diferentes. Entretanto, há uma medida para avaliar a distância entre eles.

Definição 4.4. A distância p -Wasserstein definida entre dois diagramas de persistência B e B' é dada por

$$W_p(B, B') = \inf_{\gamma: B \rightarrow B'} \left(\sum_{u \in B} \|u - \gamma(u)\|_\infty^p \right)^{\frac{1}{p}},$$

onde $1 \leq p < \infty$ e γ é bijeção entre B e B' .

Seja $h: \mathbb{R}^2 \rightarrow \mathbb{R}$ uma função diferenciável. Denote $|\nabla h| = \sup_{z \in \mathbb{R}^2} \|\nabla h(z)\|_2$. Pelo teorema do valor médio, temos que

$$|h(u) - h(v)| \leq |\nabla h| \|u - v\|_2. \quad (4.7)$$

Sejam $u, v \in \mathbb{R}^2$ e considere as duas distribuições diferenciáveis ϕ_u, ϕ_v . Como o supremo e a derivada de direção maximal de uma distribuição de probabilidade diferenciável são invariantes por translação, podemos denotar $|\nabla \phi_u|$ por $|\nabla \phi|$ e $\|\phi_u\|_\infty$ por $\|\phi\|_\infty$. E observe ainda devido a invariância pela translação, temos que

$$\|\phi_u - \phi_v\|_\infty \leq |\nabla \phi| \|u - v\|_2. \quad (4.8)$$

Vamos enunciar um lema agora que será utilizado nas provas de estabilidade das imagens de superfície e persistência.

Lema 4.5. Sejam $u, v \in \mathbb{R}^2$, então $\|f(u)\phi_u - f(v)\phi_v\| \leq (\|f\|_\infty |\nabla \phi| + \|\phi\|_\infty |\nabla f|) \|u - v\|_2$.

Demonstração. Seja $z \in \mathbb{R}^2$ qualquer, então

$$\begin{aligned}
|f(u)\phi_u(z) - f(v)\phi_v(z)| &= |f(u)(\phi_u(z) - \phi_v(z)) + (f(u) - f(v))\phi_v(z)| \\
&\leq \|f\|_\infty \|\phi_u(z) - \phi_v(z)\| + \|\phi\|_\infty |f(u) - f(v)| \\
&\leq \|f\|_\infty \|\nabla\phi\| \|u - v\|_2 + \|\phi\|_\infty + \|\nabla f\| \|u - v\|_2 \quad \text{por 4.8 e 4.7} \\
&= (\|f\|_\infty \|\nabla\phi\| + \|\phi\|_\infty + \|\nabla f\|) \|u - v\|_2.
\end{aligned}$$

□

Teorema 4.6. A superfície de persistência é estável em relação a distância 1-Wasserstein. Dados B, B' diagramas de persistência finitos, temos que

$$\|\rho_B - \rho_{B'}\|_\infty \leq \sqrt{10} (\|f\|_\infty \|\nabla\phi\| + \|\phi\|_\infty + \|\nabla f\|) W_1(B, B')$$

Demonstração. Por hipótese, B e B' são finitos, logo existe uma bijeção entre B e B' que atinge o ínfimo da distância de Wasserstein. Portanto

$$\begin{aligned}
\|\rho_B - \rho_{B'}\|_\infty &= \left\| \sum_{u \in T(B)} f(u)\phi_u - \sum_{u \in T(B)} f(\gamma(u))\phi_{\gamma(u)} \right\|_\infty \\
&\leq \sum_{u \in T(B)} \|f(u)\phi_u - f(\gamma(u))\phi_{\gamma(u)}\| \\
&\leq (\|f\|_\infty \|\nabla\phi\| + \|\phi\|_\infty + \|\nabla f\|) \sum_{u \in T(B)} \|u - \gamma(u)\|_2 \quad \text{por 4.5} \\
&\leq \sqrt{2} (\|f\|_\infty \|\nabla\phi\| + \|\phi\|_\infty + \|\nabla f\|) \sum_{u \in T(B)} \|u - \gamma(u)\|_\infty \quad \text{já que } \|\cdot\|_2 \leq \sqrt{2} \|\cdot\|_\infty \text{ em } \mathbb{R}^2 \\
&\leq \sqrt{10} (\|f\|_\infty \|\nabla\phi\| + \|\phi\|_\infty + \|\nabla f\|) \sum_{u \in T(B)} \|u - \gamma(u)\|_\infty \quad \text{já que } \|T(\cdot)\|_2 \leq \sqrt{5} \|\cdot\|_\infty \\
&= \sqrt{10} (\|f\|_\infty \|\nabla\phi\| + \|\phi\|_\infty + \|\nabla f\|) W_1(B, B').
\end{aligned}$$

A última desigualdade é necessária, pois a distância de Wasserstein é definida sobre os pontos do diagrama de persistência que são da forma nascimento e morte, não nascimento e persistência. □

E por fim, temos que as imagens de persistência são estáveis.

Teorema 4.7. A imagem de persistência é estável em relação a distância 1-Wasserstein. Se A é o valor máximo dentre todos os pixels da imagem, então

$$\|I(\rho_B) - I(\rho_{B'})\|_\infty \leq \sqrt{10} (\|f\|_\infty \|\nabla\phi\| + \|\phi\|_\infty + \|\nabla f\|) W_1(B, B'). \quad (4.9)$$

Demonstração. A demonstração segue do Teorema 4.6 e do fato que para um pixel p qualquer

$$|I(\rho_B)_p - I(\rho_{B'})_p| \leq A(p) \|\rho_B - \rho_{B'}\|_\infty,$$

em que $A(p)$ representa a área do píxel p . □

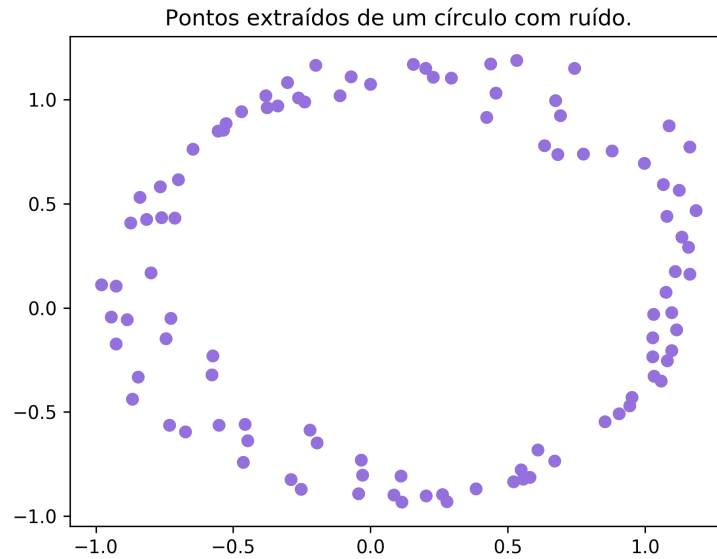


Figura 21 – Pontos extraídos de um círculo com ruídos.

Fonte: Elaborada pelo autor.

4.2.2 Exemplos de Imagens de Persistência

Considere X um conjunto de pontos extraídos de um círculo com ruído, como pode ser visto na Figura 21. Na Figura 22 tem-se os diagramas de persistência do círculo de dimensão 0 e 1, assim mostrando as componentes conexas e buracos. Note que existem dois pontos longe da diagonal, um representando a componente conexa e o outro o buraco do círculo. Vamos agora analisar as imagens de persistência para cada uma das imagens. Escolhemos a distribuição gaussiana dada por

$$g_u(x, y) = \frac{1}{2\pi\sigma^2} e^{-((x-u_x)^2 + (y-u_y)^2)/2\sigma^2}.$$

A Equação 4.6 é utilizada como função peso. Para calcular as imagens de persistência definimos três variâncias: 0.001, 0.1, 1.0, e dois tamanhos de imagem: 10×10 e 50×50 . O resultado pode ser visto na Figura 23.

Observe a diferença entre os tamanhos escolhidos para as imagens. Com um tamanho maior, a informação fica mais fina, porém a imagem fica esparsa. Além disso, com uma variância mais baixa os pontos ficam mais concentrados, enquanto para valores mais altos há uma troca contínua entre as regiões dos pontos com maior frequência.

Todo o código para gerar o círculo com ruído, calcular os diagramas e imagens se encontram no *Jupyter Notebook* no repositório da dissertação: <<https://github.com/chronchi/dissertacao>> na pasta *jupyter_notebook*.

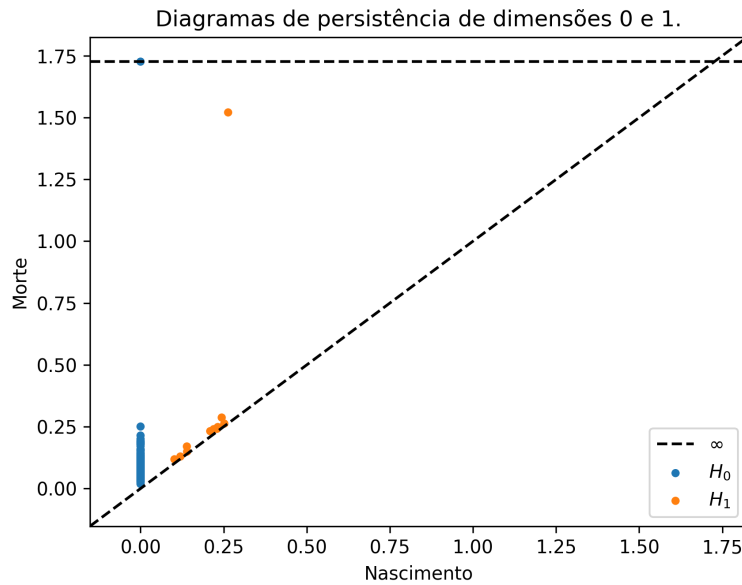


Figura 22 – Diagramas de persistência do círculo X . Em laranja o diagrama de persistência de dimensão 1, em azul o de dimensão 0. A filtração de Vietoris-Rips foi usada para calcular o complexo simplicial.

Fonte: Elaborada pelo autor.

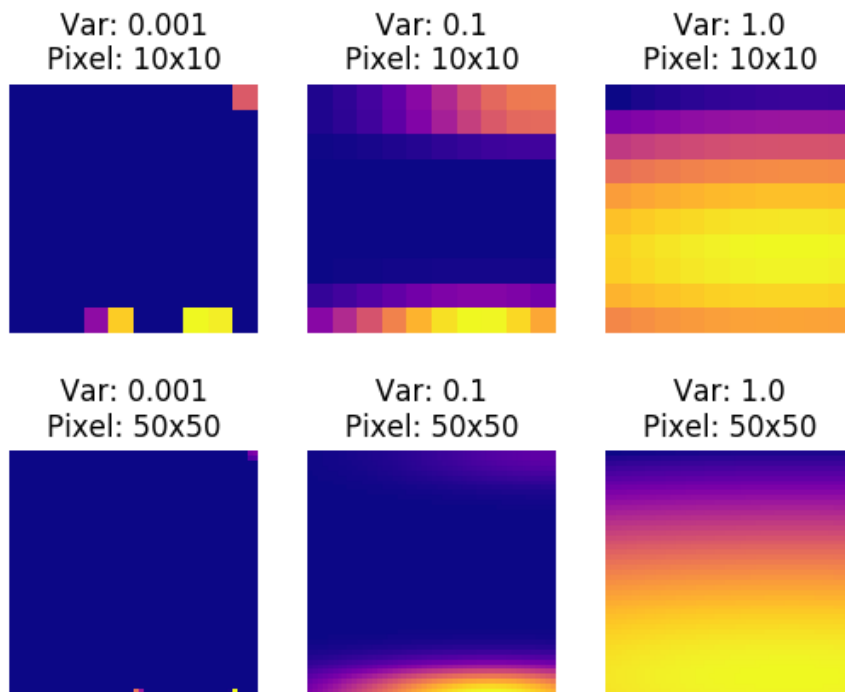


Figura 23 – Seis imagens de persistência do diagrama de dimensão 1 da [Figura 22](#).

Fonte: Elaborada pelo autor.

4.3 Mapper

Mapper é um algoritmo para visualização em \mathbb{R} ou \mathbb{R}^2 de dados com alta dimensão através do uso de grafos e complexos simpliciais (SINGH; MEMOLI; CARLSSON, 2007). Ele é aplicado em diversas áreas, sendo que uma delas é na inferência de formatos (LUM *et al.*, 2013). Nas próximas subseções iremos apresentar a motivação topológica para o desenvolvimento do algoritmo assim como sua versão estatística, usada em implementações.

4.3.1 Mapper topológico

Seja X um espaço topológico, Z um espaço de parâmetros tais que $f: X \rightarrow Z$ é uma função contínua. A função f é chamada de filtro. Considere agora uma cobertura aberta de Z , $\{U_\alpha\}_{\alpha \in A}$ para algum conjunto finito de índices A . Como f é contínua, os conjuntos $f^{-1}(U_\alpha)$ também são abertos, portanto, temos uma cobertura finita de X dada por $\{f^{-1}(U_\alpha)\}_{\alpha \in A}$.

Para cada um desses conjuntos abertos, considere suas respectivas componentes conexas por caminho. Dessa forma podemos quebrar a cobertura em conjuntos $V(\alpha, i)$, em que $U_\alpha = \cup_i V(\alpha, i)$ e cada conjunto V é aberto. Denote esta nova cobertura por \bar{U} .

Dada tal cobertura, podemos associar um complexo simplicial. Para uma cobertura \mathcal{U} qualquer com um conjunto de índices finitos A , defina o nervo $N(\mathcal{U})$ como o complexo simplicial cujo conjunto de índices é A e $\{\alpha_0, \dots, \alpha_k\} \subset A$ gera um k -simplexo em $N(\mathcal{U})$ se, e somente se, $U_{\alpha_0} \cap \dots \cap U_{\alpha_k}$ é não-vazio. O Exemplo 4.8 é um caso do mapper topológico.

Exemplo 4.8. Seja X um círculo unitário no plano com centro na origem e $Z = [-1, 1]$. defina a função filtro como a projeção no eixo y , $f(x, y) = y$. Seja \mathcal{U} a cobertura $\{[-1, -\frac{1}{3}), (-\frac{1}{2}, \frac{1}{2}), (\frac{1}{3}, 1]\}$. Observe que $f^{-1}([-1, -\frac{1}{3}))$ e $f^{-1}((\frac{1}{3}, 1])$ consistem de uma componente conexa cada, enquanto $f^{-1}((-\frac{1}{2}, \frac{1}{2}))$ de duas componentes conexas. O complexo simplicial pode ser realizado como mostra a Figura 24.

4.3.2 Mapper Estatístico

Vamos descrever agora a versão estatística do Mapper. Seja X uma nuvem de pontos com N pontos e suponha que temos uma função $f: X \rightarrow \mathbb{R}$ cujos valores sabemos para cada ponto em X . A função f é chamada de filtro, como anteriormente. Assuma também que podemos calcular a distância entre pontos de X .

O primeiro passo é achar um intervalo ($I \subset \mathbb{R}$) na imagem da função f sobre o conjunto X e uma cobertura dos dados sobre I . Após a seleção do intervalo I , temos mais

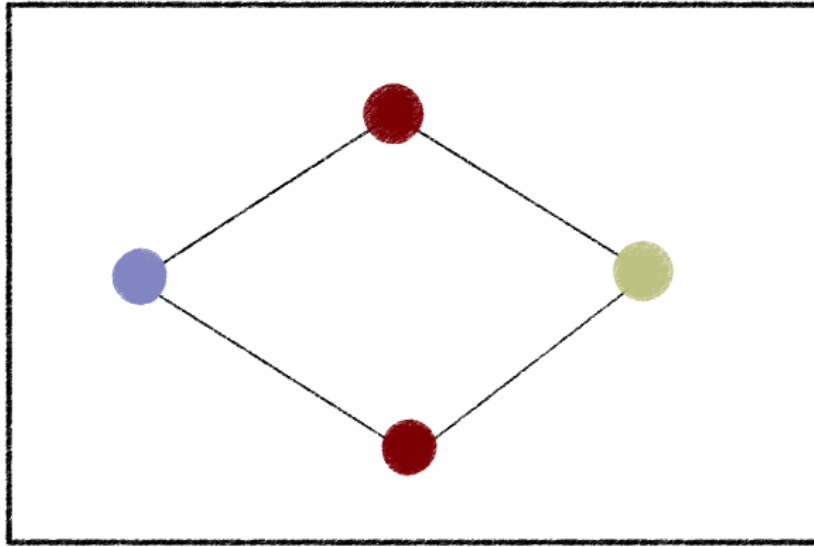


Figura 24 – Complexo simplicial associado a X , Z e f do Exemplo 4.8

Fonte: Elaborada pelo autor.

dois parâmetros para determinar: as regiões menores que dividirão I e a porcentagem de interseção entre elas.

Dados o intervalo I , cobertura S de I e p a porcentagem de interseção, tome um intervalo $I_j \in S$ e considere os conjuntos $X_j = \{x \in X \mid f(x) \in I_j\}$. Logo, temos uma cobertura natural de $X \subset \cup X_j$. Esse passo é diferente da versão topológica, já que nós trabalhamos sobre um conjunto finito, temos que agrupar os pontos em cada X_j , caso contrário dependemos da topologia, onde cada ponto seria uma componente conexa e não teríamos nenhuma informação relevante.

Portanto, devemos selecionar um algoritmo de agrupamento (Clustering), explicando o porque de precisar uma função distância entre os pontos. Então, para cada conjunto X_j aplicamos o algoritmo de agrupamento e denotamos os clusters por X_{ji} . Observe que temos uma cobertura de X_j dessa forma. Na versão topológica, cada componente conexa por caminhos era tratada como um vértice em um grafo, enquanto que a versão estatística cada cluster será tratado como um vértice. Além disso, desenhemos uma aresta entre dois vértices (clusters) X_{jk} e X_{lm} toda vez que $X_{jk} \cap X_{lm} \neq \emptyset$.

Exemplo 4.9. Este é um exemplo do mapper sendo aplicado em um círculo com ruídos. A função filtro é $f(x) = \|x - p\|_2$, em que p é o ponto mais a esquerda, em relação ao eixo y dos dados. As cores de cada vértice na Figura 25 correspondem à média do valor da função filtro em cada vértice, em que azul representa um valor baixo e vermelho um valor alto.

O intervalo do filtro é $[0, 4.2]$. O intervalo foi dividido em 5 partes menores com uma interseção de 20%. O algoritmo de agrupamento utilizado foi o *single-linkage clustering*.

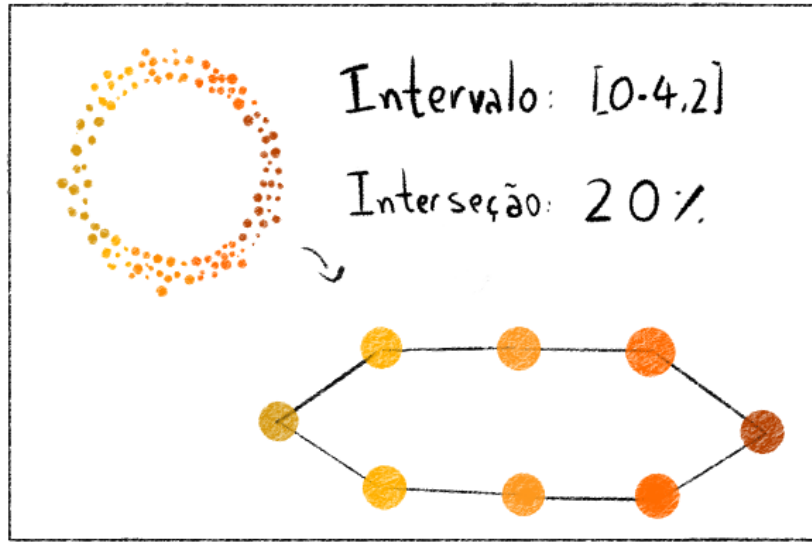


Figura 25 – Exemplo do Mapper sendo aplicado em um círculo com ruído.

Fonte: Elaborada pelo autor.

Este exemplo foi retirado de (SINGH; MEMOLI; CARLSSON, 2007).

É possível também estender esta versão do algoritmo para uma com o espaço de parâmetros de dimensão mais alta, com o \mathbb{R}^2 , mais detalhes desta versão podem ser encontrados em (SINGH; MEMOLI; CARLSSON, 2007).

4.3.3 Funções filtro

A função filtro aplicada no algoritmo Mapper precisada ser escolhida com muito cuidado, pois o grafo resultante depende da função escolhida. A seguir mostramos alguns exemplos para funções filtro que capturam algumas propriedades dos conjuntos de dados.

4.3.3.1 Ecentricidade

Essa é uma família de funções que capturam informações geométricas do conjunto de dados. Estas funções identificam pontos longe do centro, sem especificar exatamente que ponto é o centro e onde ele está.

Seja agora p com $q \leq p < \infty$, então

$$E_p(x) = \left(\frac{\sum_{y \in X} d(x, y)^p}{N} \right)^{\frac{1}{p}}, \quad (4.10)$$

e para $p = \infty$, $E_\infty(x) = \max_{x' \in X} d(x, x')$.

4.3.3.2 Densidade

Seja $\varepsilon > 0$. Então, f_ε é a estimativa do kernel gaussiano

$$f_\varepsilon(x) = C_\varepsilon \sum_y \exp\left(\frac{-d(x,y)^2}{\varepsilon}\right), \quad (4.11)$$

em que $x, y \in X$ e C_ε é uma constante tal que $\int f_\varepsilon = 1$. A ideia do kernel gaussiano é que ele suaviza o conjunto de dados de forma que o parâmetro ε controla a suavidade.

Podemos também usar funções que dependem do problema a ser trabalhado. Por exemplo, considere um conjunto de proteínas com um score de estabilidade ou energia total associado a elas. Podemos definir f em cada proteína como os valores mencionados, pois esses já carregam informações biológicas do conjunto.

4.3.4 Implementação

Uma variação da implementação do mapper pode ser encontrada em <https://github.com/chronchi/MapperMDS.jl> desenvolvida pelo estudante. O algoritmo aceita uma matriz de distância como entrada e agrupa os pontos utilizando DBScan (ESTER *et al.*, 1996) por padrão, mas aceita o algoritmo de agrupamento single linkage. Se este último for usado, recomenda-se usar o método da silhueta (ROUSSEEUW, 1987) para escolher o melhor número de cluster em cada passo de agrupamento.

ESTABILIDADE DE PROTEÍNAS

O problema de enovelamento da proteína é a questão fundamental de como a sua sequência de aminoácidos no plano se transforma em uma estrutura atômica tridimensional. Esta questão é essencial, pois um melhor entendimento dessa situação pode levar ao desenvolvimento de novos remédios e também uma melhora no combate de doenças. No entanto, continua sendo um grande desafio obter uma estrutura estável da proteína a partir da sequência de aminoácidos. Recentemente, alguns estudos ([ROCKLIN *et al.*, 2017](#)) desenvolveram novas proteínas usando o software *Rosetta*, que modela estruturas macromoleculares. Existem alguns problemas no desenvolvimento usando tal software, como proteínas modeladas que não são tão estáveis sobre o processo de proteólise, que é a quebra da proteína em pedaços menores. Pode-se contornar este problema através do uso de outras ferramentas avançadas, como as encontradas em aprendizado de máquinas e análise topológica de dados.

Neste capítulo estudamos a estabilidade de proteínas sobre um score proposto em ([ROCKLIN *et al.*, 2017](#)) utilizando imagens de persistência ([ADAMS *et al.*, 2017](#)), descritas no Capítulo 4, e vários algoritmos de aprendizado de máquinas implementados em ([PEDREGOSA *et al.*, 2011](#)). Mais detalhes são dados na Seção 5.1

Por outro lado podemos estudar a performance de algoritmos para modelagem computacional e análise estrutural de proteínas, como *Rosetta* e *Amber*. Em ([RUBENSTEIN *et al.*, 2018](#)), ambos os softwares são comparados em relação a proteína-energia. Para uma proteína específica, eles geraram milhares de moléculas similares e calcularam a raiz do erro quadrático médio em relação a proteína original. Após isso, eles analisaram as moléculas de falso mínimo. Dada uma lista de proteínas simuladas, elas são ranqueadas de acordo com suas energias normalizadas. Uma molécula simulada é uma falsa mínima se está no top 10 das moléculas no ranking e seu RMSD é maior do que 5. Eles observaram que o software *Rosetta* gerou mais proteínas de falso mínimo do que o *Amber*. Na Se-

ção 5.2 analisamos a estrutura das proteínas dadas pelo *Rosetta* utilizando ciclos ótimos (ESCOLAR; HIRAOKA, 2015), imagens de persistência (ADAMS *et al.*, 2017), VAE's (KINGMA; WELLING, 2013) e diversos algoritmos de machine learning utilizando o sklearn (PEDREGOSA *et al.*, 2011). Nós tentamos prever e apresentar uma nova função para ajudar o *Rosetta* no seu passo de otimização quando simulando novas moléculas.

5.1 Estudando a estabilidade - Proteínas I

O desenvolvimento de novas moléculas através de softwares acelerou o estudo de novas estruturas atômicas e suas propriedades. No entanto, fica cada vez mais difícil verificar experimentalmente se uma proteína modelada computacionalmente é estável ou não, pois por várias vezes existem milhares ou dezenas de milhares de moléculas, dificultando a experimentação no laboratório devido ao grande número e o seu custo relacionado.

Em (ROCKLIN *et al.*, 2017) eles apresentam um novo método de desenvolvimento de proteínas com o auxílio de algoritmos de aprendizado de máquinas. Para cada round de desenvolvimento de proteínas, eles selecionam apenas algumas para testar a estabilidade no laboratório e estudar o que precisa ser modificado na estrutura da molécula para o próximo round de experimentos computacionais. Nesta seção apresentamos um método que utiliza análise topológica de dados para selecionar as moléculas com resultados similares. Porém, utilizando homologia persistente obtemos outra informações geométricas da proteína, que pode auxiliar no desenvolvimento e aprimoramento em cada round de experimentos. Por exemplo, as informações dos buracos e cavidades, que podem ser obtidas com a homologia persistente, nos dizem o quão hidrofóbica ou hidrofílica uma proteína é. (JAMADAGNI; GODAWAT; GARDE, 2011)

5.1.1 A estabilidade da proteína

Dada uma proteína modelada, podemos medir a sua estabilidade da seguinte maneira. Primeiro, deposita-se várias cópias da proteína sobre uma célula de levedura, para várias células. Então cada célula é geneticamente fundida à uma tag de expressão que pode ser tabelada com fluorescência. Assim, as células são encubadas com diversas concentrações diferentes de enzimas proteolíticas. A quantidade de enzimas utilizadas para quebrar metade das proteínas depositadas na levedura são então gravadas e vão dar o score de estabilidade da proteína. Se um valor baseado nessa soma das concentrações for maior que 1, dizemos que a proteína é estável, caso contrário dizemos que ela é instável. Podemos ver na Figura 26 o processo de quebra das cópias da proteína. Mais detalhes podem ser vistos em (ROCKLIN *et al.*, 2017).

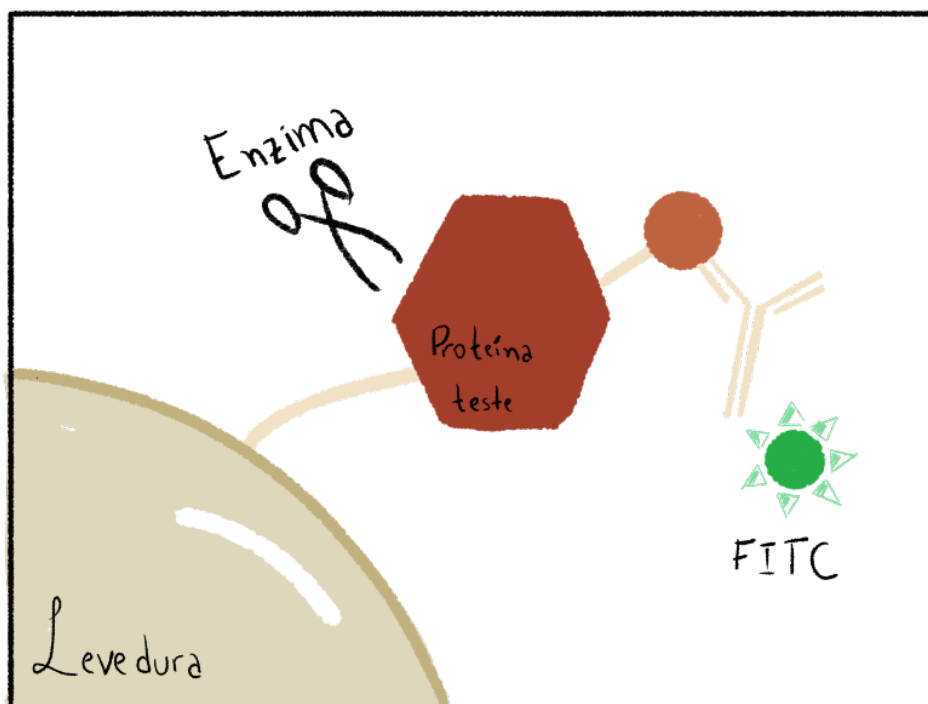


Figura 26 – Processo de quebra de uma cópia da proteína sobre a superfície de uma célula de levedura.

Fonte: Elaborada pelo autor.

5.1.2 Prevendo a estabilidade

O conjunto de proteínas utilizado contém 12927 moléculas para o treinamento e 3232 proteínas para o teste. No conjunto de treinamento existem 2210 proteínas estáveis (com score maior do que 1) enquanto que no conjunto de teste existem 305 moléculas estáveis.

Cada proteína tem 110 propriedades associadas, ou seja, valores numéricos, que vão desde área de superfície não acessível (relacionada à hidrofobicidade da proteína) a potenciais coulombianos.

Os autores de (ROCKLIN *et al.*, 2017) desenvolveram um algoritmo para prever a estabilidade da proteína e selecionar as melhores moléculas para testes em laboratório. Eles usaram o algoritmo de árvore de decisões *Random forest* para treinar mais de 12000 proteínas e prever seu score de estabilidade. Foi treinado um regressor com o erro quadrático médio (MSE em inglês) para a minimização. Os resultados são dados em relação à raiz do erro quadrático médio (RMSE em inglês), erro porcentual (RMSE dividido pela diferença entre o maior e menor score de estabilidade) e podem ser vistos na Tabela 1

A seguir apresentamos dois métodos utilizando homologia persistente para o problema de predição do score de estabilidade. O primeiro foi usando as propriedades obtidas utilizando homologia persistente e imagens de persistente além das propriedades

Tabela 1 – Resultados do algoritmo treinado pelos autores de (ROCKLIN *et al.*, 2017).

Modelo	RMSE	Erro Percentual (%)
Random Forest	0,419	11,381

das proteínas já geradas. Já o segundo foi apenas utilizando as imagens de persistência das proteínas. Nas próximas seções descrevemos a metodologia utilizada, parâmetros e resultados.

5.1.3 Metodologia

Para cada proteína construímos sete subconjuntos, um para cada conjunto em $\mathcal{A} = \{\{C\}, \{O\}, \{N\}, \{C, O\}, \{C, N\}, \{O, N\}, \{C, O, N\}\}$. Para cada subconjunto calculamos os diagramas de persistência de dimensão 1 e 2 utilizando a filtração Alpha. Os pesos utilizados para a filtração Alpha foram os raios de Van der Waals para cada átomo.

Para vetorizar os diagramas de persistência utilizamos imagem de persistência com os seguintes parâmetros:

- Tamanho da imagem: 5×4 ;
- Variância: 0,1, 0,3, 0,5, 0,7.

Então, concatenamos as imagens de persistência de forma a obter um vetor de tamanho 280. Para o primeiro método concatenamos as 110 propriedades de cada proteína no final do vetor, totalizando o seu tamanho em 390. Uma vez com os vetores podemos treinar os algoritmos de aprendizado de máquinas para prever o score de estabilidade. Abaixo temos a liste de algoritmos utilizados:

- Regressão linear;
- Regressão linear com regularização;
- Árvore de decisão;
- GBoost.

E por fim após o treinamento obtemos os scores de estabilidade dos conjuntos de teste. A Figura 27 mostra o pipeline da metodologia utilizada.

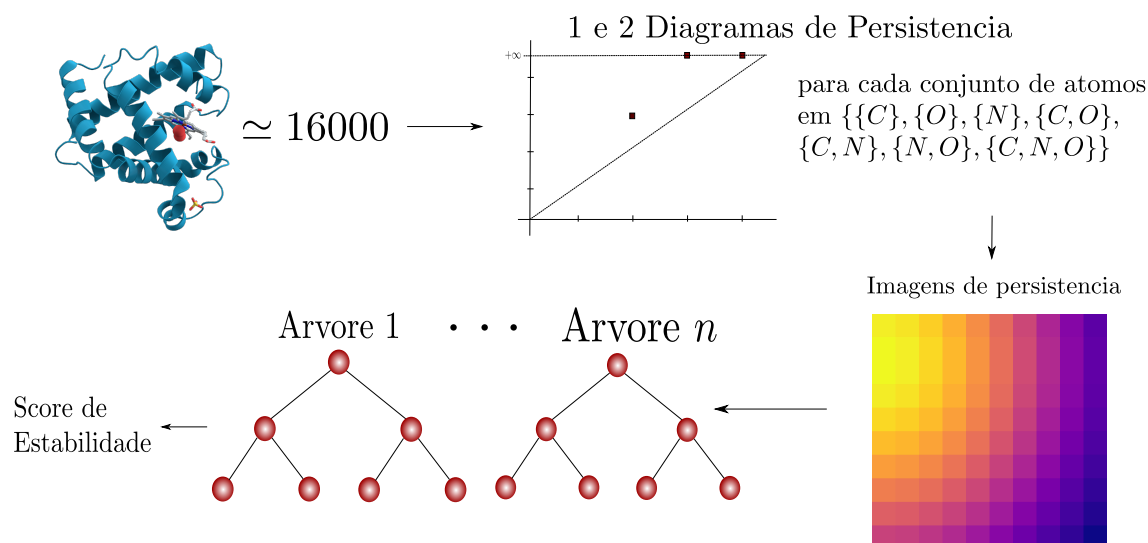


Figura 27 – Pipeline da metodologia utilizada para a predição do score de estabilidade.

Fonte: Elaborada pelo autor.

5.1.4 Resultados e análises

5.1.4.1 Primeiro método

O primeiro método consiste em utilizar as propriedades da proteínas e as imagens de persistência para prever o score de estabilidade. Temos 4 tabelas para cada uma das variâncias.

Tabela 2 – Resultados para variância igual a 0,1.

Modelo	MSE	RMSE	Erro Percentual (%)	R^2
GBoost	0,1831	0,4278	11,61	0,5529
Regressão Linear	0,2025	0,4500	12,21	0,5054
Regressão lin. c/ Reg.	0,2084	0,4565	12,39	0,4910
Árvore de decisão I	0,1771	0,4208	11,42	0,5674
Árvore de decisão II	0,1780	0,4219	11,45	0,5653

Tabela 3 – Resultados para variância igual a 0,3.

Modelo	MSE	RMSE	Erro Percentual (%)	R^2
GBoost	0,1832	0,4281	11,62	0,5525
Regressão Linear	23,6637	4,8645	132,01	-56,7950
Regressão lin. c/ Reg.	0,2075	0,4555	12,36	0,4932
Árvore de decisão I	0,1772	0,4209	11,42	0,5672
Árvore de decisão II	0,1785	0,4225	11,46	0,5641

Tabela 4 – Resultados para variância igual a 0,5.

Modelo	MSE	RMSE	Erro Percentual (%)	R^2
GBoost	0,1829	0,4277	11,61	0,5533
Regressão Linear	0,2032	0,4508	12,23	0,5036
Regressão lin. c/ Reg.	0,2068	0,4548	12,34	0,4949
Árvore de decisão I	0,1781	0,4220	11,45	0,5650
Árvore de decisão II	0,1783	0,4222	11,46	0,5646

Tabela 5 – Resultados para variância igual a 0,7.

Modelo	MSE	RMSE	Erro Percentual (%)	R^2
GBoost	0,1828	0,4275	11,60	0,5536
Regressão Linear	0,2009	0,4482	12,16	0,5093
Regressão lin. c/ Reg.	0,2069	0,4549	12,34	0,4947
Árvore de decisão I	0,1779	0,4218	11,45	0,5655
Árvore de decisão II	0,1790	0,4230	11,48	0,5629

Esperavamos obter resultados melhores quando combinássemos ambos os conjuntos de dados em um só, mas os resultados ficaram bem similares. Acreditamos que o resultado continua similar pois as propriedades dadas pelas imagens da persistência possuem uma correlação com propriedades já conhecidas de proteínas, sendo assim a adição de novas propriedades não melhora o algoritmo.

Os parâmetros para os algoritmos são os seguintes:

- GBoost: $n_estimators \in [100, 300, 500, 700]$, $min_samples_split \in [2, 5, 10, 15]$;
- Árv. Dec. I: $n_estimators = 689$, $max_features = 0.2$, $max_depth = 86$;
- Árv. Dec. II: $n_estimators = 500$, $max_depth = 100$, $max_features = 0.3$;
- Regressão Lin. c/ Reg: valor de alfa em $[0.005, 5000]$ (de 10 em 10) escolhido com cross validation de 5 folds.

5.1.4.2 Segundo método

Para o segundo método, utilizamos apenas as imagens de persistência para o treinamento. Obtivemos um resultado similar ao dos modelos treinados com as propriedades das proteínas apenas. Os resultados podem ser vistos nas tabelas a seguir. Note também como os resultados melhoram quando aumentamos a variância utilizada para as imagens de persistência.

Tabela 6 – Resultados dos modelos treinados utilizando apenas as imagens de persistência com variância 0.1

Modelo	MSE	RMSE	Erro Percentual (%)	R^2
Regressão Linear	0,2734	0,5229	14,19	0,3322
GBoost	0,2435	0,4935	13,39	0,4053
Árvore de Decisão I	0,2490	0,4990	13,54	0,3917
Árvore de Decisão II	0,2485	0,4985	13,53	0,3931
Regressão Lin. c/ Reg.	0,2723	0,5218	14,16	0,3350
GBoost Ótimo	0,2450	0,4950	13,43	0,4017

Tabela 7 – Resultados dos modelos treinados utilizando apenas as imagens de persistência com variância 0.3

Modelo	MSE	RMSE	Erro Percentual (%)	R^2
Regressão Linear	438,3102	20,9359	568,14	-1069,5076
GBoost	0,2364	0,4863	13,20	0,4225
Árvore de Decisão I	0,2417	0,4917	13,34	0,4096
Árvore de Decisão II	0,2419	0,4919	13,35	0,4091
Regressão Lin. c/ Reg.	0,2649	0,5146	13,97	0,3531
GBoost Ótimo	0,2305	0,4801	13,03	0,4371

Tabela 8 – Resultados dos modelos treinados utilizando apenas as imagens de persistência com variância 0.5

Modelo	MSE	RMSE	Erro Percentual (%)	R^2
Regressão Linear	0,2639	0,5137	13,94	0,3555
GBoost	0,2327	0,4824	13,09	0,4316
Árvore de Decisão I	0,2409	0,4908	13,32	0,4117
Árvore de Decisão II	0,2403	0,4902	13,30	0,4130
Regressão Lin. c/ Reg.	0,2594	0,5093	13,82	0,3665
GBoost Ótimo	0,2328	0,4825	13,09	0,4314

Tabela 9 – Resultados dos modelos treinados utilizando apenas as imagens de persistência com variância 0.7

Modelo	MSE	RMSE	Erro Percentual (%)	R^2
Regressão Linear	0,2546	0,5046	13,69	0,3781
GBoost	0,2342	0,4839	13,13	0,4281
Árvore de Decisão I	0,2379	0,4877	13,24	0,4190
Árvore de Decisão II	0,2376	0,4874	13,23	0,4197
Regressão Lin. c/ Reg.	0,2535	0,5035	13,66	0,3809
GBoost Ótimo	0,2276	0,4770	12,95	0,4442

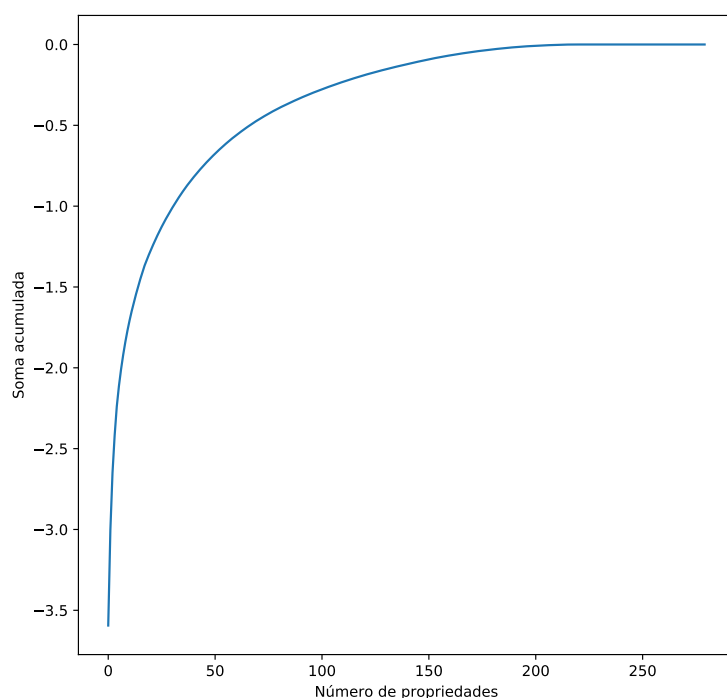


Figura 28 – Logaritmo da soma acumulada em relação à importância das propriedades dado pelo algoritmo GBoost.

Fonte: Elaborada pelo autor.

Observamos que o melhor resultado é para o GBoost ótimo com variância 0,7. Como o GBoost é um método de árvores de decisão, ele nos dá a importância de cada uma das propriedades dos vetores. Vamos fazer uma análise dessas propriedades.

Primeiro, precisamos selecionar um número significativo de propriedades, já que algumas são mais importantes que as outras. Cada propriedade tem um número associado a ela, dado pelo algoritmo GBoost. A soma de todas esses valores é 1. Na Figura 28 temos o logaritmo do valor da soma acumulada da propriedade com maior valor de importância até a última. Note como o plot dobra ao redor do número 50. Isso significa que as 50 primeiras propriedades são as mais importantes, já contribuem mais para a soma acumulada. Pela construção das imagens de persistência, temos que cada propriedade está relacionada com uma certa região da vetorização do diagrama. Portanto, podemos localizar as regiões dos diagramas de persistência que aparecem nas propriedades. Dentre as 50 propriedades mais importantes, temos que a maioria vem de pontos dos diagramas de dimensão 1 e correspondem a pontos com nascimentos logo no início da filtração e com persistência baixa, como pode ser visto nas Figuras 29 e 30

Dentre as propriedades mais importantes podemos analisar também a que conjunto



Figura 29 – Heatmap das regiões dos diagramas de persistência de dimensão 1 que aparecem nas primeiras 50 propriedades. Eixo x representa nascimento, enquanto que o eixo y representa a persistência.

Fonte: Elaborada pelo autor.



Figura 30 – Heatmap das regiões dos diagramas de persistência de dimensão 2 que aparecem nas primeiras 50 propriedades. Eixo x representa nascimento, enquanto que o eixo y representa a persistência.

Fonte: Elaborada pelo autor.

de átomos elas estão associadas quando os diagramas de persistência foram calculados. Observamos na Figura 31 que os átomos associados aos ciclos que mais aparecem são os encontrados nos diagramas de persistência calculados utilizando apenas carbono e nitrogênio. Os autores de (CANG; WEI, 2017a) afirmam que ciclos associados a esses átomos representam propriedades hidrofóbicas e hidrofílicas, características que influenciam diretamente na estabilidade da proteína.

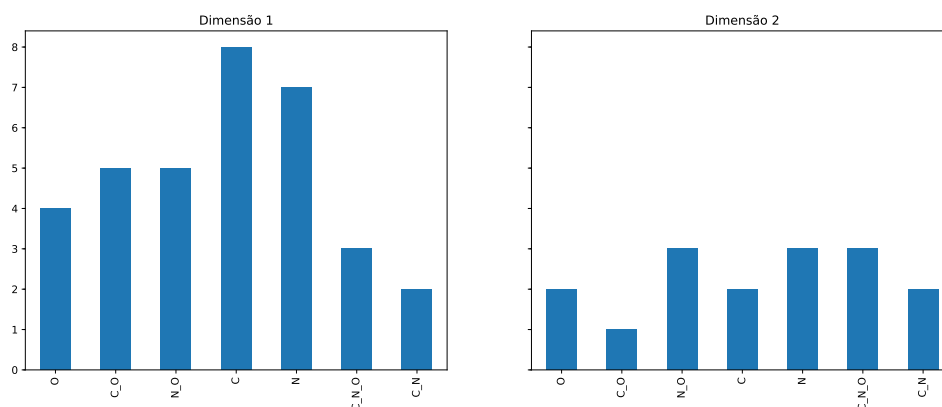


Figura 31 – Número de ciclos associados para as top 50 propriedades e seus respectivos diagramas de persistência.

Fonte: Elaborada pelo autor.

5.2 Analisando a energia total - Proteínas II

Em (RUBENSTEIN *et al.*, 2018) eles analisam a eficácia do *Rosetta* e *Amber*, dois softwares para modelagem de macromoléculas. Dado uma proteína obtida do Protein Data Bank (PDB), por exemplo a proteína de ID 1T2I, eles geraram novas moléculas usando amostragem ab-initio com viés e sem viés seguido por uma amostragem paralela loophash. Após isso, essas amostras foram sujeitas à minimização no backbone (átomos C- α) e cadeias laterais (grupo-R) usando o protocolo talaris2014 e o minimizador LBFGS. Então com os átomos C- α apenas, o RMSD foi calculado para todas as decoys (moléculas geradas pelo software).

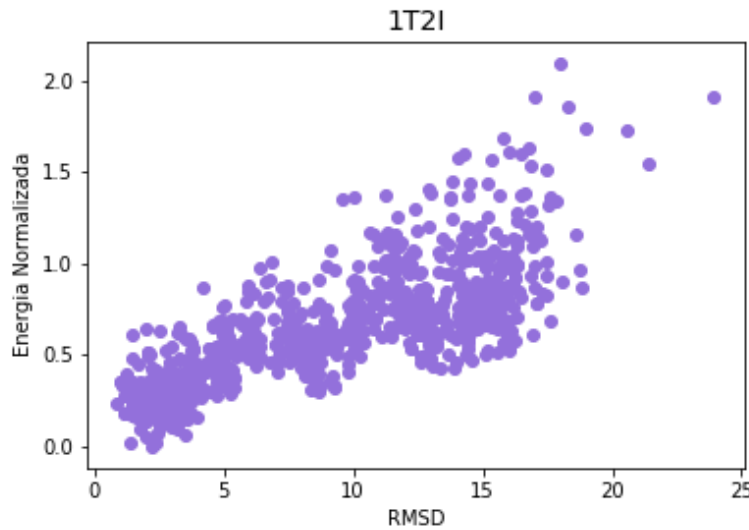


Figura 32 – Panorama de energia para decoys modeladas em relação à proteína 1T2I.

Fonte: Elaborada pelo autor.

Para cada decoy existe um score de energia associado, que é a função score minimizada pelo *Rosetta*. Com esse valor podemos plotar o panorama de energia para cada proteína, como na Figura 32. O formato ideal seria o de um túnel, já que RMSD baixo corresponderia a uma energia normalizada baixa idealmente.

O score de energia dado por *Rosetta* é normalizado usando a seguinte fórmula

$$E_{i(norm)} = \frac{E_i - E_{min}}{E_{95th} - E_{5th}}, \quad (5.1)$$

em que E_{95th} é o 95-ésimo percentil e E_{5th} é o quinto percentil.

5.2.1 Análise de falso mínimos

Dados as moléculas geradas pelo software, podemos ranquear cada decoy de acordo com sua energia normalizada e RMSD. Ranqueamos o conjunto de decoys da menor para

a maior energia, por exemplo uma molécula com energia de 0.3 está acima de outra com energia 0.5 no ranking. Na Tabela 10 temos o top 5 decoys das proteínas geradas a partir da 1T2I.

Tabela 10 – Rank mostrando as top 5 decoys em relação a 1T2I.

Rank	Energia normalizada	RMSD
1	0.000	2.233
2	0.023	1.37
3	0.025	2.395
4	0.057	2.004
5	0.061	2.356

Como mencionado anteriormente, *Rosetta* tenta minimizar uma função score de energia de forma que energia baixa corresponde a um valor baixo do RMSD. Dizemos que uma decoy é um falso mínimo se está no top 10 moléculas do ranking de acordo com a definição acima e também possui um RMSD maior do que 5.

5.2.1.1 VAE e ciclos ótimos

Para analisar os falsos mínimos, utilizamos homologia persistente (EDELSBRUNNER; LETSCHER; ZOMORODIAN, 2002) para extrair informações biológicas, como hidrofobicidade, juntamente com outras ferramentas topológicas (CANG; WEI, 2017b).

Para cada ponto no diagrama de persistência existe um ciclo, um representante para sua respectiva classe de homologia, que possui propriedades geométricas dos dados. Apesar disso, os ciclos não possuem o verdadeiro tamanho do correspondente buraco n -dimensional, com respeito ao número de arestas. Portanto, o problema de encontrar o ciclo ótimo em relação ao número de arestas é muito interessante, já que assim podemos representar as propriedades topológicas de maneira muito mais fiel (ESCOLAR; HIRAOKA, 2015).

Por um lado, ciclos ótimos codificam muita informação, por outro lado é muitas vezes difícil analisa-los de forma coesa. Portanto, nós propomos um método similar a (OBAYASHI; HIRAOKA; KIMURA, 2018). Primeiro vetorizamos os diagramas de persistência utilizando imagens de persistência (ADAMS *et al.*, 2017) e após isso treinamos um autoencoder variacional básico (KINGMA; WELLING, 2013) para extrair as regiões mais importantes da imagem de persistência, em relação a esse autoencoder. Então realizamos uma análise inversa, em que para cada região da imagem, existem pontos associados no diagrama de persistência e dessa forma seus respectivos ciclos. Então, para cada conjunto de ciclos, somamos todos os átomos correspondentes de cada ciclo, por exemplo, soma de todos os átomos de carbono de todos os ciclos.

Nas próximas subseções mostramos os resultados e parâmetros utilizados para gerar os diagramas de persistência, imagens de persistência e hiperparâmetros para o treinamento do VAE.

5.2.1.2 Resultados

Selecionamos as proteínas de ID 2QY7 e 1T2I para análise. A primeira contém vários falsos mínimos, enquanto a última não possui falso mínimo.

Para a proteína 2QY7 calculamos os diagramas de persistência para os top 100 decoys e plotamos a soma na Figura 33.

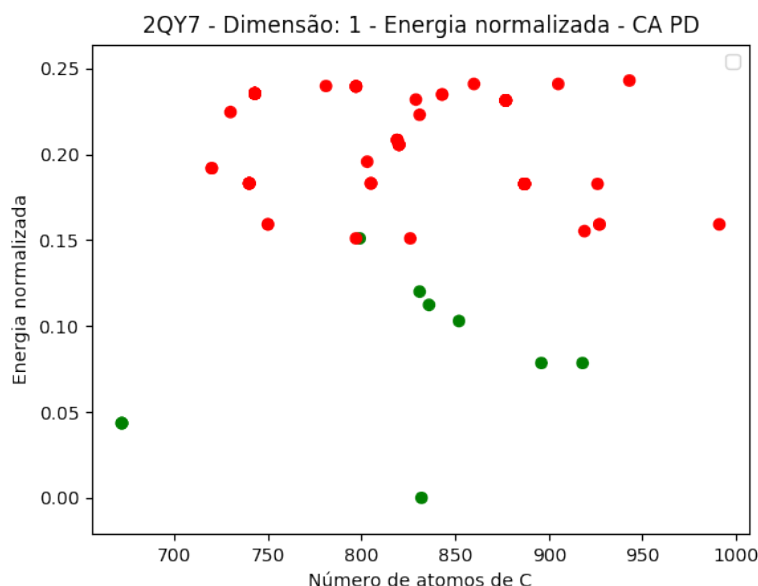


Figura 33 – Soma dos átomos de carbono que compõem os ciclos do 1º diagrama de persistência das decoys da 2QY7.

Fonte: Elaborada pelo autor.

Para cada decoy foi calculado dois diagramas de persistência, um para a dimensão 1 e outro para a dimensão 2. Em cada uma das dimensões usamos duas nuvens de pontos, a primeira composta apenas pelos átomos C- α das moléculas e a outra composta apenas pelos átomos de nitrogênio e oxigênio.

Note que quando usamos apenas os átomos de carbono, a maioria dos falsos mínimos ficam agrupados em um intervalo pequeno, como pode ser visto na Figura 33, e por outro lado com a outra nuvem de pontos os valores ficaram espalhados, indicando que para esta proteína é melhor usar os átomos C- α para análise de falsos mínimos.

Já para a proteína 1T2I um fenômeno similar acontece, como pode ser visto na Figura ???. As top decoys ficam em um intervalo menor, enquanto outras moléculas estão

espalhadas pelo intervalo.

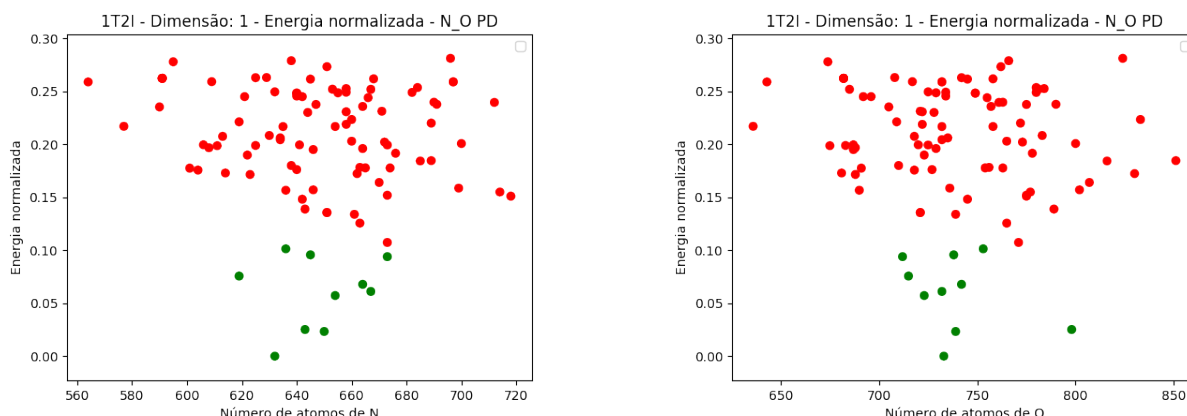


Figura 34 – Soma dos átomos de nitrogênio (esquerda) e oxigênio (direita) que compõe os ciclos do 1º diagrama de persistência das decoys da proteína 1T2I.

Fonte: Elaborada pelo autor.

É importante notar que os ciclos da proteína com um panorama de energia bom (1T2I) foram melhor caracterizados pelos átomos de nitrogênio e oxigênio, enquanto que para a outra proteína, os átomos C- α caracterizaram melhor.

5.2.1.3 Parâmetros

Calculamos os primeiro e segundo diagramas de persistência usando a filtração alpha, onde o raio de cada átomo era o raio de Van der Waals. Para os ciclos ótimos o software *optiperslp* foi utilizado. As imagens de persistência foram criadas utilizando a linguagem python e o pacote persim (SAUL; TRALIE, 2019) com os seguintes parâmetros: tamanho da imagem (pixel) = (10,10), variância = 1, e a função peso é a padrão sugerida em (ADAMS *et al.*, 2017).

Para o treinamento do VAE, 75 imagens de persistência foram utilizadas para o treinamento e 25 para o teste. O número de épocas é 300 e taxa de aprendizado igual a 0.0001. O algoritmo de otimização utilizado foi o Adam.

O número de regiões das imagens de persistência selecionadas foi 5, ou seja, 5 regiões de 100 com os maiores valores.

5.2.2 Prevendo o RMSD

Ao invés de usar a estrutura topológica dada pelos diagramas de persistência e os respectivos ciclos ótimos para estudar os falsos mínimos, utilizamos as imagens de persistência de várias decoys de diversas proteínas diferentes em algoritmos de machine

learning, como regressão linear, árvores de decisão, redes neurais e regressão linear com regularização.

Escolhemos as proteínas 1T2I e 2NQW para testar e um outro conjunto de proteínas para o treinamento (proteínas que contêm pelo menos um falso mínimo). O top 10 para ambas as proteínas pode ser visto na Figura 35.

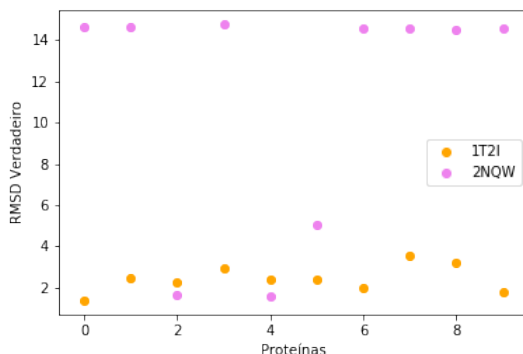


Figura 35 – Valor do RMSD para cada decoy no top 10. Não existem falsos mínimos para a proteína 1T2I, enquanto isso existem 7 falsos mínimos para a proteína 2NQW.

Fonte: Elaborada pelo autor.

5.2.2.1 Resultados

Para podermos comparar os resultados dos teste utilizando imagens de persistência, treinamos os mesmos algoritmos no conjunto de propriedades de proteínas dadas pelo *Rosetta* quando desenvolvendo um novo decoy. As propriedades são dadas por

fa_dun, fa_elec, fa_intra_rep, hbond_sc,
fa_rep, fa_sol, hbond_bb_sc, hbond_lr_bb,
hbond_sr_bb, omega, p_aa_pp, pro_close, rama.

Quando treinamos os regressores com essas propriedades, obtemos as seguintes figuras. Na Figura 36 são os resultados para a proteínas 1T2I e na Figura 37 os resultados para 2NQW.

Agora podemos analisar os modelos que utilizam imagens de persistência no treinamento. Na Tabela 11 temos os parâmetros utilizados para diversos testes. A coluna **Pixel** mostra o tamanho das imagens, n significa (n, n) . **# Teste** é uma identificação para os resultados. Para cada teste três diagramas de persistência foram calculados, somente os átomos C- α , os átomos N e O e por fim todos os átomos menos os de hidrogênio.

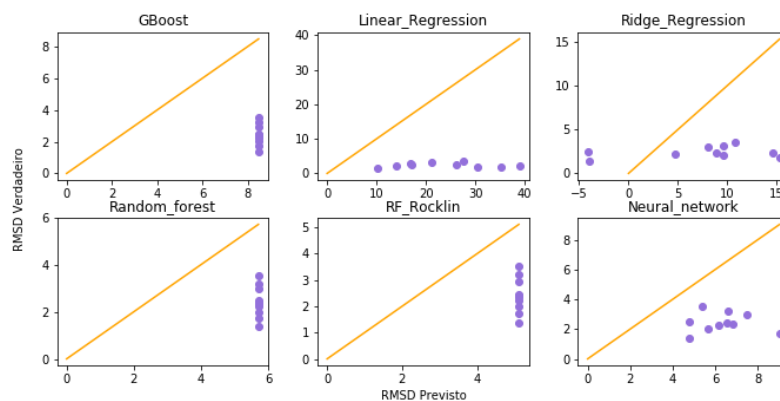


Figura 36 – Proteína 1T2I. RMSD previsto x RMSD verdadeiro para o top 10 dados os regressores treinados em outras proteínas.

Fonte: Elaborada pelo autor.

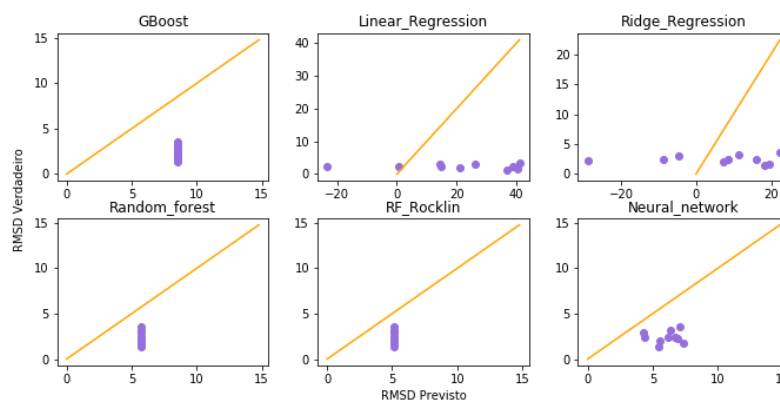


Figura 37 – Proteína 2NQW. RMSD previsto x RMSD verdadeiro para o top 10 dados os regressores treinados em outras proteínas.

Fonte: Elaborada pelo autor.

Tabela 11 – Alguns dos testes feitos para obter as imagens de persistência e usa-las para o treinamento.

#	Teste	Pixel	Variância	Dimensão PD
1		10	0,3	1
2		10	0,5	1
3		10	0,6	1
4		10	0,8	1
5		10	1,0	1
6		10	1,2	1
7		3	1,0	1
8		5	1,0	1
9		50	0,3	1
10		50	1,0	1
11		100	0,3	1
12		100	1,0	1

Treinamos os mesmos regressores como anteriormente para várias proteínas. Definimos então 4 métricas diferentes para selecionar o melhor regressor com respeito a cada uma. As métricas são:

- R^2 score: medida estatística para medir o quão perto os dados estão da linha de regressão;
- MSE: Erro quadrático médio;
- RMSE: Raíz do erro quadrático médio;
- Acurácia binária: Converte cada RMSD para 0 ou 1 usando a seguinte regra: se o RMSD é maior que 5 então 0, senão 1.

A Tabela 12 mostra os melhores experimentos para cada métrica usando imagens de persistência na hora do treinamento. Por outro lado, a Tabela 13 mostra os melhores

Tabela 12 – Melhores parâmetros para cada métrica para os regressores treinados nas imagens de persistência.

Métrica	Regressor	Pixel	Variância	Lista de átomos ¹	Score médio
R^2	Redes neurais	100	1,0	C	-5,780
MSE	Redes neurais	100	1,0	C	8,299
RMSE	Regressão lin c/ reg.	10	1,2	todo	2,599
Acurácia Binária	GBoost	10	0,6	N,O	0,657

regressores treinados nas propriedades dadas pelo *Rosetta*.

Tabela 13 – Melhores regressores treinados com as propriedades das proteínas

Métrica	Regressor	Score médio
R^2	Random Forest II	-13,706
MSE	Random Forest II	10,113
RMSE	Random Forest II	2,707
Acurácia binária	Regressão lin. c/ reg.	0,586

Os regressores treinados com imagens de persistência obtiveram melhores resultados do que os treinados utilizando as propriedades que o *Rosetta* para todas as métricas. Podemos ver na Figura 38 e 39 que os modelos baseados em imagens de persistência possuem uma maior acurácia binária. O método baseado em propriedades topológicas pode ser estendido. Devido a sua natureza e similaridade com imagens, pode-se utilizar redes neurais convolucionais para o treinamento.

Este trabalho mostra que usar imagens de persistência é melhor para as tarefas de predição do RMSD para proteínas não vistas anteriormente. Os algoritmos treinados

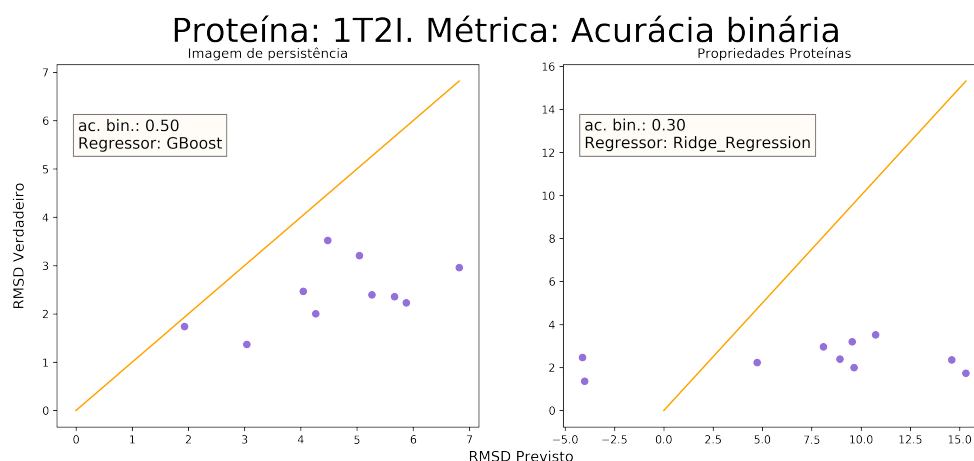


Figura 38 – RMSD previsto x RMSD verdadeiro para o top 10 decoys da proteína 1T2I dados os regressores com a melhor acurácia binária no conjunto de validação.

Fonte: Elaborada pelo autor.

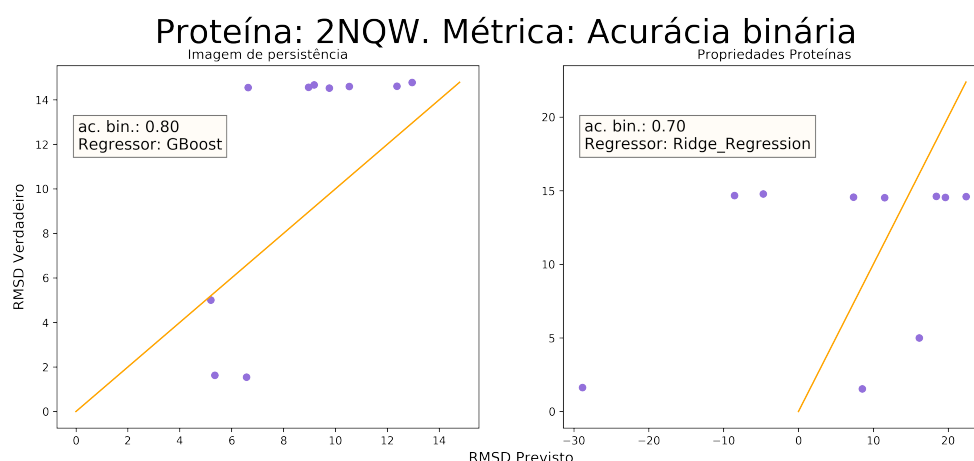


Figura 39 – RMSD previsto x RMSD verdadeiro para o top 10 decoys da proteína 2NQW dados os regressores com a melhor acurácia binária no conjunto de validação.

Fonte: Elaborada pelo autor.

podem ser usados como uma função para o *Rosetta* utilizar na hora dos passos de minimização no desenvolvimento de novas proteínas.

O Jupyter Notebook (KLUYVER *et al.*, 2016) está disponível online com a lista completa de proteínas (ID's) utilizadas no treinamento e teste, assim como com o código para a análise de resultados dos modelos. Os arquivos podem ser baixados [aqui](https://bit.ly/2XUjat2) (<<https://bit.ly/2XUjat2>>).

CONCLUSÃO

Este trabalho propôs a apresentação de homologia persistente, desde os princípios básicos a teoria, assim como aplicações diretas que produziram resultados comparáveis ao estado da arte.

O problema de enovelamento de proteína é algo que precisa ser estudado e novos métodos precisam ser discutidos. Nesta dissertação apresentamos novos métodos para o estudo do problema e obtivemos resultados similares aos de estado da arte propostos por grupos de renome internacional. O conteúdo apresentado é fruto de um trabalho interdisciplinar e mostra também o potencial da análise topológica de dados para tentar resolver outros problemas de biologia.

O aluno também desenvolveu diversos pacotes, contribuindo diretamente tanto para a comunidade de topologia aplicada como para a de bioinformática. A lista de pacotes desenvolvidos é a seguinte:

- [MapperMDS.jl](#): uma implementação do mapper em Julia.
- [PersistenceImage.jl](#): implementação da imagem de persistência em Julia.
- [ProteinPersistent.jl](#): pacote que faz chamada do Bio.PDB e ripser do python para o cálculo dos diagramas de persistência de proteínas em Julia.
- [perscode](#): pacote de vetorização de diagramas de persistência descritos em (ZIELINSKI *et al.*, 2018) na linguagem de programação python.

Todos os pacotes podem ser encontrados em <https://github.com/chronchi>. A dissertação, códigos e arquivos tex podem ser acessados em <https://github.com/chronchi/dissertacao>.

REFERÊNCIAS

ADAMS, H.; EMERSON, T.; KIRBY, M.; NEVILLE, R.; PETERSON, C.; SHIPMAN, P.; CHEPUSHTANOVA, S.; HANSON, E.; MOTTA, F.; ZIEGELMEIER, L. Persistence images: A stable vector representation of persistent homology. **Journal of Machine Learning Research**, v. 18, n. 8, p. 1–35, 2017. Disponível em: <<http://jmlr.org/papers/v18/16-337.html>>. Citado nas páginas 82, 83, 91, 92, 101 e 103.

AZUMAYA, G. Corrections and supplementaries to my paper concerning krull-remak-schmidt's theorem. **Nagoya Mathematical Journal**, Cambridge University Press (CUP), v. 1, p. 117–124, jun. 1950. Disponível em: <<https://doi.org/10.1017/s002776300002290x>>. Citado na página 46.

BUBENIK, P. Statistical topological data analysis using persistence landscapes. **Journal of Machine Learning Research**, v. 16, n. 3, p. 77–102, 2015. Disponível em: <<http://jmlr.org/papers/v16/bubenik15a.html>>. Citado na página 82.

CANG, Z.; WEI, G.-W. Integration of element specific persistent homology and machine learning for protein-ligand binding affinity prediction. **International Journal for Numerical Methods in Biomedical Engineering**, Wiley, v. 34, n. 2, p. e2914, aug 2017. Disponível em: <<https://doi.org/10.1002/cnm.2914>>. Citado na página 99.

_____. TopologyNet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions. **PLOS Computational Biology**, Public Library of Science (PLoS), v. 13, n. 7, p. e1005690, jul 2017. Disponível em: <<https://doi.org/10.1371/journal.pcbi.1005690>>. Citado na página 101.

CHAZAL, F.; COHEN-STEINER, D.; GLISSE, M.; GUIBAS, L. J.; OUDOT, S. Y. Proximity of persistence modules and their diagrams. In: **Proceedings of the Twenty-fifth Annual Symposium on Computational Geometry**. New York, NY, USA: ACM, 2009. (SCG '09), p. 237–246. ISBN 978-1-60558-501-7. Disponível em: <<http://doi.acm.org/10.1145/1542362.1542407>>. Citado na página 68.

CHAZAL, F.; SILVA, V. de; GLISSE, M.; OUDOT, S. **The Structure and Stability of Persistence Modules**. Springer International Publishing, 2016. Disponível em: <<https://doi.org/10.1007/978-3-319-42545-0>>. Citado nas páginas 39, 44, 46, 50, 51, 56, 62, 66 e 72.

COHEN-STEINER, D.; EDELSBRUNNER, H.; HARER, J. Stability of persistence diagrams. **Discrete & Computational Geometry**, Springer Science and Business Media LLC, v. 37, n. 1, p. 103–120, dez. 2006. Disponível em: <<https://doi.org/10.1007/s00454-006-1276-5>>. Citado nas páginas 68 e 72.

DEY, T. K.; HIRANI, A. N.; KRISHNAMOORTHY, B. Optimal homologous cycles, total unimodularity, and linear programming. 2010. Citado nas páginas 75 e 76.

DILL, K. A.; OZKAN, S. B.; SHELL, M. S.; WEIKL, T. R. The protein folding problem. **Annual Review of Biophysics**, Annual Reviews, v. 37, n. 1, p. 289–316, jun. 2008. Disponível em: <<https://doi.org/10.1146/annurev.biophys.37.092707.153558>>. Citado na página 22.

EDELSBRUNNER; LETSCHER; ZOMORODIAN. Topological persistence and simplification. **Discrete & Computational Geometry**, Springer Nature, v. 28, n. 4, p. 511–533, 11 2002. Disponível em: <<https://doi.org/10.1007/s00454-002-2885-2>>. Citado na página 101.

EDELSBRUNNER, H. **Computational topology : an introduction**. Providence, R.I: American Mathematical Society, 2010. ISBN 0821849255. Citado nas páginas 21, 27, 32 e 34.

EDELSBRUNNER, H.; LETSCHER, D.; ZOMORODIAN, A. Topological persistence and simplification. In: **Proceedings 41st Annual Symposium on Foundations of Computer Science**. IEEE Comput. Soc, 2000. Disponível em: <<https://doi.org/10.1109/sfcs.2000.892133>>. Citado nas páginas 34 e 39.

ESCOLAR, E. G.; HIRAOKA, Y. Optimal cycles for persistent homology via linear programming. In: **Optimization in the Real World**. [S.l.]: Springer Japan, 2015. p. 79–96. Citado nas páginas 74, 81, 92 e 101.

ESTER, M.; KRIEGEL, H.-P.; SANDER, J.; XU, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In: . [S.l.]: AAAI Press, 1996. p. 226–231. Citado na página 90.

JAMADAGNI, S. N.; GODAWAT, R.; GARDE, S. Hydrophobicity of proteins and interfaces: Insights from density fluctuations. **Annual Review of Chemical and Biomolecular Engineering**, Annual Reviews, v. 2, n. 1, p. 147–171, jul. 2011. Disponível em: <<https://doi.org/10.1146/annurev-chembioeng-061010-114156>>. Citado na página 92.

KINGMA, D. P.; WELLING, M. **Auto-Encoding Variational Bayes**. 2013. Disponível em: <<http://arxiv.org/abs/1312.6114>>. Citado nas páginas 92 e 101.

KLUYVER, T.; RAGAN-KELLEY, B.; PÉREZ, F.; GRANGER, B.; BUSSONNIER, M.; FREDERIC, J.; KELLEY, K.; HAMRICK, J.; GROUT, J.; CORLAY, S.; IVANOV, P.; AVILA, D.; ABDALLA, S.; WILLING, C. Jupyter notebooks – a publishing format for reproducible computational workflows. In: LOIZIDES, F.; SCHMIDT, B. (Ed.). **Positioning and Power in Academic Publishing: Players, Agents and Agendas**. [S.l.], 2016. p. 87 – 90. Citado na página 107.

LANE, S. M. **Categories for the Working Mathematician**. Springer New York, 1978. Disponível em: <<https://doi.org/10.1007/978-1-4757-4721-8>>. Citado na página 63.

LUM, P. Y.; SINGH, G.; LEHMAN, A.; ISHKANOV, T.; VEJDEMO-JOHANSSON, M.; ALAGAPPAN, M.; CARLSSON, J.; CARLSSON, G. Extracting insights from the shape of complex data using topology. **Scientific Reports**, The Author(s) SN -, v. 3, p. 1236 EP -, Feb 2013. Article. Disponível em: <<https://doi.org/10.1038/srep01236>>. Citado na página 87.

- OBUYASHI, I.; HIRAOKA, Y.; KIMURA, M. Persistence diagrams with linear machine learning models. **Journal of Applied and Computational Topology**, Springer Science and Business Media LLC, v. 1, n. 3-4, p. 421–449, maio 2018. Disponível em: <<https://doi.org/10.1007/s41468-018-0013-5>>. Citado na página 101.
- PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011. Citado nas páginas 91 e 92.
- POINCARÉ, H. Analysis situs. **Journal de l'École Polytechnique**, p. 1–123, 1895. Citado nas páginas 21 e 23.
- ROCKLIN, G. J.; CHIDYAUSSIKU, T. M.; GORESHNIK, I.; FORD, A.; HOULISTON, S.; LEMAK, A.; CARTER, L.; RAVICHANDRAN, R.; MULLIGAN, V. K.; CHEVALIER, A.; ARROWSMITH, C. H.; BAKER, D. Global analysis of protein folding using massively parallel design, synthesis, and testing. **Science**, American Association for the Advancement of Science (AAAS), v. 357, n. 6347, p. 168–175, jul 2017. Disponível em: <<https://doi.org/10.1126/science.aan0693>>. Citado nas páginas 17, 22, 91, 92, 93 e 94.
- ROUSSEEUW, P. J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. **Journal of Computational and Applied Mathematics**, Elsevier BV, v. 20, p. 53–65, nov. 1987. Disponível em: <[https://doi.org/10.1016/0377-0427\(87\)90125-7](https://doi.org/10.1016/0377-0427(87)90125-7)>. Citado na página 90.
- RUBENSTEIN, A.; BLACKLOCK, K.; NGUYEN, H.; CASE, D.; KHARE, S. Systematic comparison of amber and rosetta energy functions for protein structure evaluation. American Chemical Society (ACS), 2018. Citado nas páginas 91 e 100.
- SAUL, N.; TRALIE, C. **Scikit-TDA: Topological Data Analysis for Python**. 2019. Disponível em: <<https://doi.org/10.5281/zenodo.2533369>>. Citado na página 103.
- SIMONS, K. T.; KOOPERBERG, C.; HUANG, E.; BAKER, D. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and bayesian scoring functions. **Journal of Molecular Biology**, Elsevier BV, v. 268, n. 1, p. 209–225, abr. 1997. Disponível em: <<https://doi.org/10.1006/jmbi.1997.0959>>. Citado na página 22.
- SINGH, G.; MEMOLI, F.; CARLSSON, G. Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition. In: BOTSCH, M.; PAJAROLA, R.; CHEN, B.; ZWICKER, M. (Ed.). **Eurographics Symposium on Point-Based Graphics**. [S.l.]: The Eurographics Association, 2007. ISBN 978-3-905673-51-7. ISSN 1811-7813. Citado nas páginas 21, 87 e 89.
- ZIELINSKI, B.; LIPINSKI, M.; JUDA, M.; ZEPPELZAUER, M.; DLOTKO, P. **Persistence Codebooks for Topological Data Analysis**. 2018. Citado na página 109.

ALGORITMO *STANDARD* E FUNÇÕES AUXILIARES

```

1 # Retorna o valor low(column) descrito no texto.
2 # Exemplo:
3 # a = [1,1,0,0,1,0]
4 # get_low(a) retorna 5, maior índice tal que o valor de a nesse índice é 1.
5 function get_low(column)
6     a = findlast(column .== 1)
7     if a == nothing
8         return 0
9     else
10        return a
11    end
12 end
13
14 # Algoritmo Standard. Input: Matriz bordo P. Output: Matriz reduzida R.
15 function reduce_matrix(boundary)
16     # número de simplexes no complexo simplicial
17     nb_simplex = size(boundary, 2)
18     # matriz R
19     reduced = boundary
20     # matrix V
21     cycles = one(boundary)
22     # Acessa cada coluna para reduzi-la
23     lowest_ones = []
24     for col in 1:nb_simplex
25         # obtém os low(i) para i < col
26         lowest_ones = [get_low(reduced[:,k]) for k in 1:(col-1)]
27         # enquanto tiver algum j_0 tal que low(j_0) = low(col) e low(col) != 0,
28         # reduza a matriz
29         while sum(get_low(reduced[:,col]) .== lowest_ones) != 0 && get_low(
30             reduced[:,col]) != 0
31             # opera a soma nas colunas que possuem o mesmo valor low.
32             for k in 1:length(lowest_ones)
33                 if get_low(reduced[:,k]) == get_low(reduced[:,col])
34                     reduced[:,col] = rem.(reduced[:,col] + reduced[:,k], 2)

```

```
33         cycles[:,col] = rem.(cycles[:,col] + cycles[:,k], 2)
34     end
35 end
36 end
37 end
38 # retorna a matriz reduzida R e a matriz V
39 return reduced, cycles
40 end
41
42 # E por fim a matriz do nosso exemplo
43 P = [0 0 0 0 1 1 0 0 0 0;
44      0 0 0 0 1 0 1 1 0 0;
45      0 0 0 0 0 0 0 1 1 0;
46      0 0 0 0 0 1 1 0 1 0;
47      0 0 0 0 0 0 0 0 0 0;
48      0 0 0 0 0 0 0 0 0 0;
49      0 0 0 0 0 0 0 0 0 1;
50      0 0 0 0 0 0 0 0 0 1;
51      0 0 0 0 0 0 0 0 0 1;
52      0 0 0 0 0 0 0 0 0 0]
53
54 # aplicando o algoritmo de redução
55 R, V = reduce_matrix(P)
```

