



Universidade Federal de São Carlos
Centro de Ciências Exatas e de Tecnologia
Departamento de Matemática



Merge Trees Decoradas e TDA.

Autor: *Guilherme de Carvalho Falsarella*

Orientador: *Prof. Dr. Daniel Vendruscolo*

São Carlos, 10 de Agosto de 2023.

Merge Trees Decoradas e TDA

Autor: *Guilherme de Carvalho Falsarella*

Orientador: *Prof. Dr. Daniel Ventrúscolo*

Instituição: Universidade Federal de São Carlos
Centro de Ciências Exatas e de Tecnologia
Departamento de Matemática

Dissertação apresentada ao PPGM da UFSCar como parte dos requisitos para a obtenção do Título de Mestre em Matemática.

São Carlos, 10 de Agosto de 2023.

Guilherme de Carvalho Falsarella
(aluno)

Prof. Dr. Daniel Ventrúscolo
(orientador)



UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Matemática

Folha de Aprovação

Assinatura dos membros da comissão examinadora que avaliou e aprovou a Defesa de Dissertação de Mestrado do candidato Guilherme de Carvalho Falsarella, realizada em 10/08/2023:

Prof. Dr. Daniel Ventrúscolo
UFSCar

Profa. Dra. Cynthia de Oliveira Lage Ferreira
USP

Profa. Dra. Alice Kimie Miwa Libardi
UNESP

Resumo

Neste trabalho, introduziremos primeiramente conceitos básicos da Persistência Homológica e da Teoria das Categorias. Em seguida, desenvolvemos a teoria das Merge Trees que serão úteis para o rastreamento da evolução das componentes conexas de um dataset dado. Além disso, estudamos uma teoria nova que aprimora as Merge Trees usuais, pondo uma decoração que indica como o dataset se comporta com homologias de grau maior que zero.

Palavras-chave: Análise Topológica de Dados, TDA, Teoria das Categorias, Topologia Algébrica, Persistência Homologia, Merge Trees, Merge Trees Decoradas.

Abstract

In this document, first we will introduce Persistence Homology and Category Theory basic concepts. Afterwards, we will develop the Merge Tree Theory which will be useful to keep track of connected components evolution of a given dataset. Finally, we expand this Merge Tree Theory to be able to study the Decorated Merge Trees, where this decoration correspond to higher degree homology.

Keywords: Topological Data Analysis, TDA, Category Theory, Algebraic Topology, Persistent Homology, Merge Trees, Decorated Merge Trees.

Conteúdo

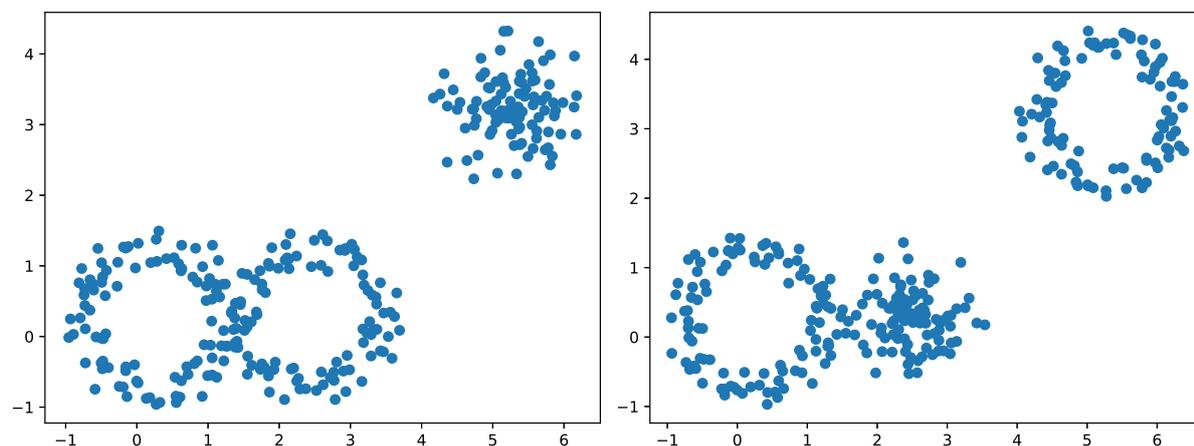
Introdução	7
1 Conceitos Preliminares	9
1.1 Persistência Homológica	9
1.2 Módulos de Persistência	19
2 Categorias	24
2.1 Teoria Básica	24
2.2 Categorificação da Homologia Persistente	38
3 Decorated Merge Trees	49
3.1 Merge Trees	49
3.2 Merge Trees Decoradas	57
3.3 Métricas Entre Merge Trees	62
3.4 Decomponibilidade das Merge Trees	73
Apêndice: Códigos Usados	86
Índice	91
Bibliografia	94

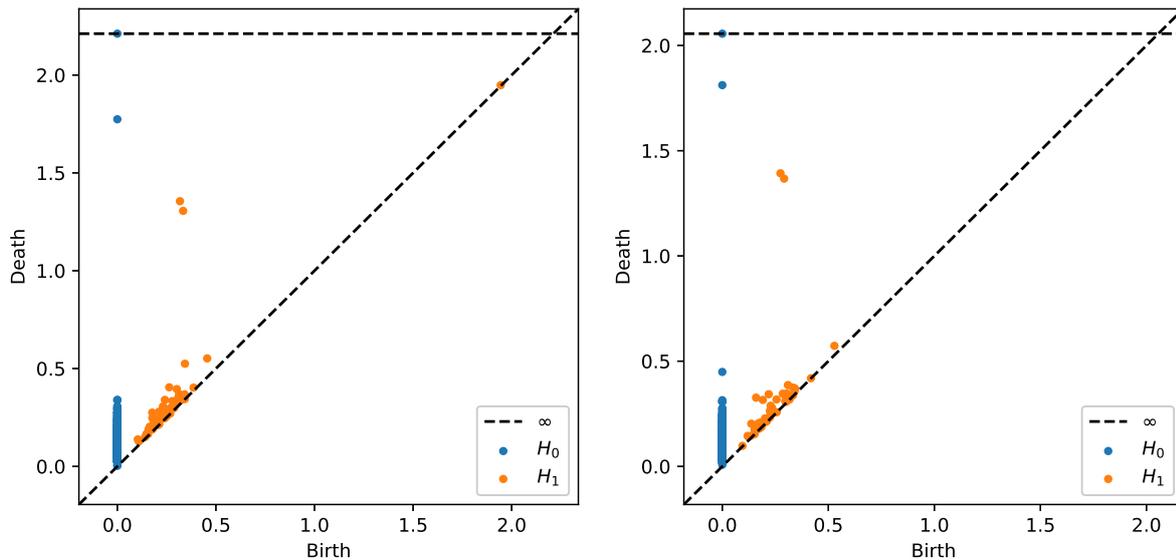
Introdução

A *Topological Data Analysis* (TDA) é uma área de pesquisa relativamente nova na matemática, que consiste em aplicar conceitos de topologia e topologia algébrica à análise de dados convencional, na tentativa de extrair mais informações dos bancos de dados.

De maneira geral, a análise de dados lida com uma quantidade finita de dados. Assim, nosso objeto de estudo não tem topologia muito interessante que possamos trabalhar. A ideia central da TDA é, de uma maneira conveniente, transformar nosso problema em um conjunto contínuo e aplicar as ferramentas de topologia algébrica nesse novo objeto. Em particular, as teorias desenvolvidas neste texto se apoiam fundamentalmente na análise vinda da Persistência Homológica. A Persistência Homológica consiste do estudo da evolução das classes de homologia de um conjunto de dados no decorrer do tempo.

Temos o seguinte problema como o motivador do estudo deste trabalho: Supondo que tenhamos dois ou mais datasets e aplicando a persistência homológica usual não obtemos diagramas de persistência que tenham diferenças significativas entre si. Como podemos trabalhar na direção de tentar criar uma ferramenta visual que nos possibilita refinar esse estudo na distinção desses objetos?





Para tentar solucionar esta questão, introduziremos as Merge Trees para complementar a persistência homológica. Isso irá nos permitir rastrear não só a evolução das componentes conexas de nosso datasets, mas também a evolução das homologias de grau maior ou igual a 1. Além disso, iremos mostrar alguns resultados que verificam que nessa nova teoria é de fato mais sensível do que a persistência homológica e estudar a estabilidade desses objetos.

Precisamos fazer sentido para o problema proposto acima, assim os dois primeiros capítulos deste texto se concentram em contextualizar a nossa questão. No Capítulo 1 apresentamos o linguajar básico de TDA e enunciamos alguns dos resultados mais relevantes da teoria. O Capítulo 2 estudamos uma introdução breve ao básico da Teoria das Categorias e como ela pode nos ser útil para aprimorar nossa teoria, nos possibilitando provar alguns resultados essenciais de maneira relativamente simples. Por fim, o Capítulo 3 desenvolvemos a teoria de Merge Trees e Merge Trees Decoradas que nos serve como uma maneira de tentar solucionar o problema proposto.

Capítulo 1

Conceitos Preliminares

Neste capítulo iremos resumir alguns fatos e proposições os quais são essenciais para desenvolver a teoria de TDA. Iremos assumir conhecido os resultados básicos de Álgebra e Topologia Algébrica como teoria de grupos, anéis, grupos fundamentais, homologia e os Axiomas de Eilenberg-Steenrod. Para uma leitura desses conceitos básicos, recomendamos [16], [19], [27], [18] e [23]. Além disso, apresentaremos de maneira superficial alguns resultados chave sobre a Teoria de TDA já consolidada na literatura, tais resultados podem ser encontrados em [12], [5], [13], [4], [7] e [34].

Destacamos que em alguns momentos iremos citar eficiência de alguns algoritmos computacionais, *Big-O notation*, e possivelmente alguns conceitos de estrutura de dados, assim recomendamos, para aqueles que não estiverem confortáveis com essa linguagem, uma breve pesquisa em [6] e [21].

1.1 Persistência Homológica

Começamos essa seção lembrando o conceito de simplexo simplicial e de complexo simplicial

Definição 1.1.1. *Em um espaço métrico, o **convex hull**, ou ainda **involatória convexa** de uma forma qualquer é o menor conjunto convexo que a contém.*

Definição 1.1.2. *Um conjunto de $k + 1$ pontos u_0, \dots, u_k em um espaço euclidiano é dito ser **independente de maneira afim** se os k vetores $u_i - u_0$, $1 \leq i \leq k$ forem linearmente independentes.*

Definição 1.1.3. *Definimos um **k -simplexo simplicial** como sendo o convex hull de $k + 1$ pontos independentes de maneira afim.*

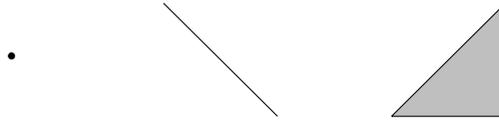


Figura 1.1: Exemplos de 0-simplexo, 1-simplexo e 2-simplexo, respectivamente. Perceba que para destacar que o 2-simplexo possui uma face, colorimos a parte interna do simplexo.

Definição 1.1.4. Um **complexo simplicial** é uma coleção de *simplexos simpliciais*, K , tal que

1. Para todo simplexo $\sigma \in K$, toda face de σ esta em K .
2. Para dois *simplexos* quaisquer $\sigma, \tau \in K$ a intersecção $\sigma \cap \tau$ é vazia ou uma face de ambos *simplexos*.

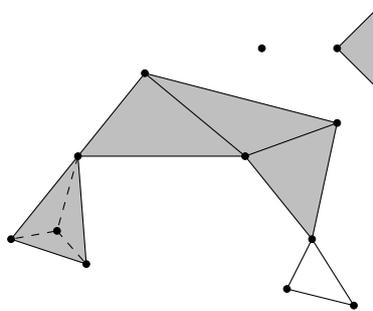


Figura 1.2: Exemplo de complexo simplicial. Perceba que não pedimos nada sobre a conexidade deste complexo, ou mesmo sobre *simplexos* de dimensões distintas.

Apresentamos aqui maneiras de se criar complexos simpliciais em nosso banco de dados. Utilizaremos alguns métodos clássicos de TDA, com exemplos computacionais realizados usando algumas bibliotecas de Python construídas especialmente para lidar com TDA, tais como Giotto-TDA, GUHDI e Ripser.py.

O resultado a seguir de topologia torna possível fazermos a discussão sobre a construção computacional dos complexos simpliciais.

Definição 1.1.5. Dada uma cobertura finita $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ de um espaço X , definimos o **nervo** da cobertura \mathcal{U} como sendo o complexo simplicial $N(\mathcal{U})$ cujo o conjunto de vértices é o conjunto de índices A , onde uma família $\{\alpha_0, \dots, \alpha_k\}$ gera um k -simplexo em $N(\mathcal{U})$ se, e somente se, $U_{\alpha_0} \cap \dots \cap U_{\alpha_k} \neq \emptyset$.

Teorema 1.1.6 (dos Nervos). Seja F uma coleção finita de convexos fechados em um espaço Euclidinao. Então o nervo de F e a união dos conjuntos em F tem a mesmo tipo homotópico.

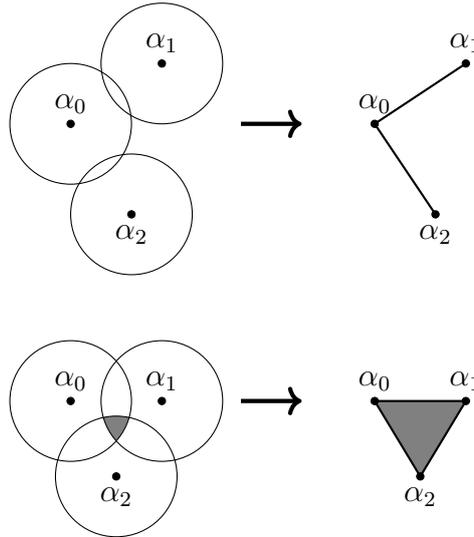


Figura 1.3: Exemplos de Nervos. Destacamos que como na imagem de baixo temos a intersecção de três bolas simultaneamente, isso nos dá uma 2-face no nosso simplexo.

Definição 1.1.7. *Seja S um conjunto finito de pontos em \mathbb{R}^d e considere*

$$\overline{B}_\varepsilon(x_0) := \{x \in \mathbb{R}^d \mid \|x - x_0\| \leq \varepsilon\}$$

a bola fechada centrada em $x_0 \in S$ e raio $\varepsilon \geq 0$. Definimos o **Complexo de Čech** de S e raio ε como sendo o nervo desta coleção de bolas, isto é,

$$\check{C}(\varepsilon) := \left\{ \sigma \subset S \mid \bigcap_{x \in \sigma} \overline{B}_\varepsilon(x) \neq \emptyset \right\}.$$

É óbvio que, do Teorema 1.1.6, $\check{C}(\varepsilon)$ tem o mesmo tipo homotópico da união das bolas fechadas de raio ε centradas em x para todo $x \in S$. Além disso, é direto verificar que $\check{C}(\varepsilon) \subset \check{C}(\varepsilon')$ para $\varepsilon \leq \varepsilon'$.

O problema desta construção é que precisamos checar as intersecções de todas as subcoleções, o que pode ser impossível de se realizar dependendo da métrica usada. Outro problema deste método é seu custo computacional para se obter um complexo simplicial. Em [9] é apresentado um algoritmo para o cálculo do complexo de Čech, que tem eficiência grosseiramente estimada em $\mathcal{O}(n^{d+3})$, lembrando que o conjunto $S \subset \mathbb{R}^d$. Isso se torna um problema para d grande e a ideia original é aplicar TDA em análise de bancos de dados com dimensão elevada, ou com grande quantidade de pontos, isto é, n grande.

Desta forma, gostaríamos que apresentar outro tipo de complexo simplicial, que seja mais viável de ser computado em grandes bancos de dados.

Definição 1.1.8. *Definimos o **complexo de Vietoris-Rips** $VR(\varepsilon)$ da seguinte maneira: um p -simplexo $\sigma = [x_0x_1 \dots x_p]$ pertence a $VR(\varepsilon)$ se, e somente se, para toda aresta $[x_i x_j]$, $0 \leq i < j \leq p$ tivermos $\|x_i - x_j\| \leq 2\varepsilon$.*

Computacionalmente, é mais simples o processo de construção do complexo de Vietoris-Rips do que o complexo de Čech, entretanto perdemos a propriedade de ter mesmo tipo homotópico da união das bolas fechadas de raio ε . Assim, podemos encontrar artefatos topológicos que não necessariamente aparecem em nossos dados, porém esses artefatos são limitados. O seguinte resultado nos assegura essa limitação dos artefatos

Teorema 1.1.9. *Seja a S um subconjunto finito de \mathbb{R}^2 . Então $\check{C}(\varepsilon) \subseteq VR(\varepsilon) \subseteq \check{C}(\varepsilon')$, onde $\varepsilon' = \frac{2\sqrt{3}}{3}\varepsilon$.*

Existem outras construções possíveis de complexos simpliciais propostos por outros pesquisadores, como os complexos Alpha e complexo Witness. Para uma leitura mais aprofundada no tema sugerimos [11]. Entretanto, tais complexos também apresentam suas peculiaridades, por exemplo o complexo Alpha depende diretamente da construção do diagrama de Voronoi. Embora para $d = 2$, tenhamos o algoritmo de Fortune, com tempo de rodagem estimado de $\mathcal{O}(n \log(n))$ (vide [10]), a determinação de tal diagrama vira um problema árduo para dimensões maiores.

Para o que nos propomos a fazer neste trabalho, os complexos de Vietoris-Rips são mais do que o suficiente, e utilizaremos a biblioteca Ripser.py para a implementação destes complexos simpliciais em exemplos quando for o caso.

Definição 1.1.10. *Uma **filtração** é uma sequência encaixante crescente de espaços topológicos. No contexto que trabalharemos aqui, consideramos K um complexo simplicial e para cada K_i um complexo simplicial intermediário, então uma filtração será uma sequência encaixante de complexos simpliciais*

$$\emptyset = K_0 \subset K_1 \subset \dots \subset K_n = K.$$

Fazer essa construção de filtração de nosso complexo simplicial nos dá a possibilidade de estudar a evolução do complexo sobre o tempo. Inicialmente temos apenas os pontos de nosso dataset e com o avançar do tempo, adicionamos simplexes em nosso espaço de maneira que obtemos complexos simpliciais que contêm todos os complexos das iterações anteriores. Por exemplo, utilizamos os algoritmos de construção de complexos simpliciais discutidos anteriormente, podemos atrelar o ε a um subconjunto da variável tempo.

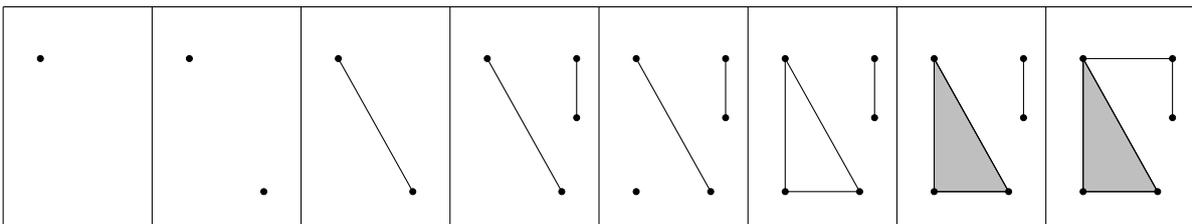


Figura 1.4: Exemplo de filtração

Fica evidente que podemos considerar a função inclusão $i : K_j \rightarrow K_l$, desta forma temos a função induzida $i_* : H_*(K_j) \rightarrow H_*(K_l)$, que nos possibilita obter uma sequência de grupos de homologia.

Quando estamos tratando de TDA, usualmente fazemos homologia sobre um corpo, tornando os grupos de homologia em espaços vetoriais. Geralmente, o corpo escolhido é \mathbb{Z}_2 , entretanto iremos informar caso usemos outro corpo nos exemplos computacionais.

Nosso intuito aqui é estudar como os grupos de homologia evoluem conforme o tempo avança. Isso motiva a seguinte definição.

Definição 1.1.11. *Considerando a função inclusão $f^{i,j} : K_i \rightarrow K_j$ que induz um homomorfismo de grupos abelianos*

$$f_p^{i,j} : H_p(K_i) \rightarrow H_p(K_j).$$

O **p-ézimo grupo de homologia persistente** $H_p^{i,j}$, $0 \leq i \leq j \leq n$, é definido pelo grupo quociente

$$H_p^{i,j} = Z_p(K_i) / B_p(K_j) \cap Z_p(K_i),$$

onde $Z_p(K_i)$ representa os p -ciclos em K_i e B_p os p -bordos em K_j .

Uma maneira alternativa para expressar o p -ezimo grupo de homologia persistente é pelo isomorfismo

$$\text{Im}(f_p^{i,j}) \cong H_p^{i,j},$$

portanto poderemos utilizar esse fato como definição quando isso for conveniente.

Definição 1.1.12. *Dizemos que uma classe de homologia $\gamma \in H_p(K_i)$ **nasce** em K_i se ela não está na imagem de $f_p^{i-1,i}$. E dizemos que γ **morre** em K_{j+1} se $f_p^{i,j}(\gamma)$ não está em $\text{Im}(f_p^{i-1,i})$ mas $f_p^{i,j+1}(\gamma)$ está em $\text{Im}(f_p^{i-1,j+1})$.*

Definição 1.1.13. *Seja γ uma classe de homologia que nasce em K_i . Se γ morrer em K_j , definimos o **índice de persistência** de γ como sendo $j - i$. Caso ocorra de que γ nunca venha a morrer, definimos o índice de persistência de γ como sendo infinito.*

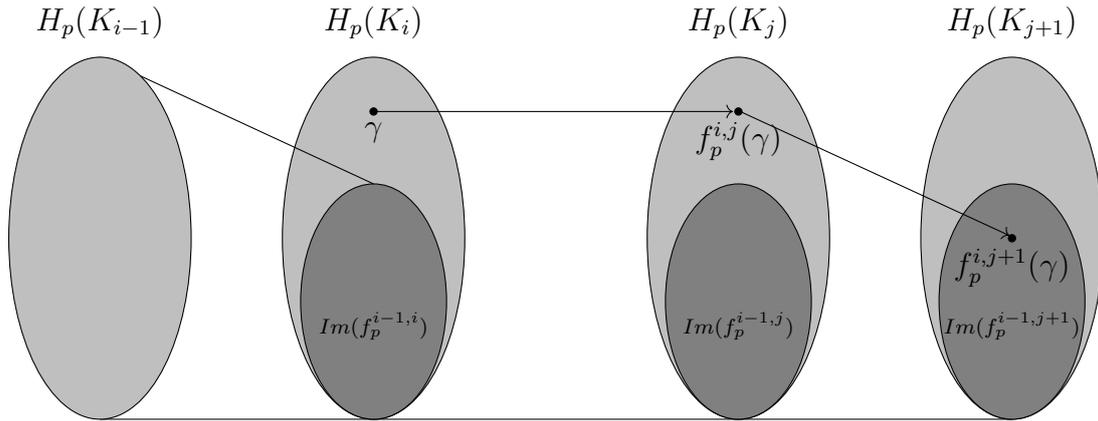


Figura 1.5: Exemplo de uma classe γ que nasce em K_i e morre ao entrar em K_{j+1} .

Apresentaremos agora duas formas de representação dessas classes de equivalências da homologia persistente.

Definição 1.1.14. *Para cada grau de homologia p , fazemos uma correspondência entre o momento de nascimento e de morte de uma classe em um eixo coordenado, formando um intervalo, ou ainda, uma **barra**. Para essa estrutura determinada pela união de todas as barras, damos o nome de **barcode**. Denotaremos por **Barcodes** o conjunto de todos os barcodes.*

A ideia desta representação gráfica é auxiliar na detecção de ruídos topológicos. Como estamos transformado um conjunto finito de pontos em um espaço topológico e estudando sua homologia persistente, podemos nos deparar com artefatos que representem classes de homologia durante a construção dos complexos simpliciais, porém elas não representam informação relevante no final do processo. Intuitivamente, esse ruído pode ser identificado com barras curtas em nosso barcode, isto é, classes de homologia persistente que nascem e morrem em um curto espaço de tempo. Já as barras longas representam classes de homologia que persistem por mais tempo, ou seja, tem informação homológica mais relevante para nós.

Exemplo 1.1.15. *Apresentamos aqui um exemplo computacional da teoria apresentada no presente momento. Faremos um dataset sintético para apresentar a ferramenta dos barcodes. Primeiramente dispomos pontos igualmente espaçados sobre duas circunferências usando o método `linspace()` da biblioteca `numpy`. Ambas as circunferências tem raio igual a 2 e elas estão centradas em $(1, 1)$ e $(3, 1)$. Em seguida adicionamos um fator de ruído, gerado por `random()`. Obtemos o dataset representado na Figura 1.6*

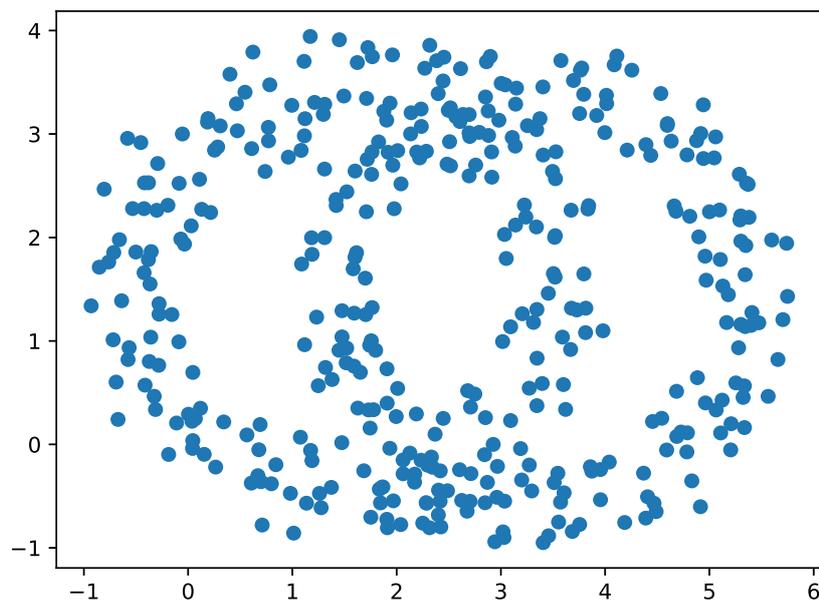


Figura 1.6: Dataset do Exemplo 1.1.15, plotado utilizando `matplotlib.pyplot`

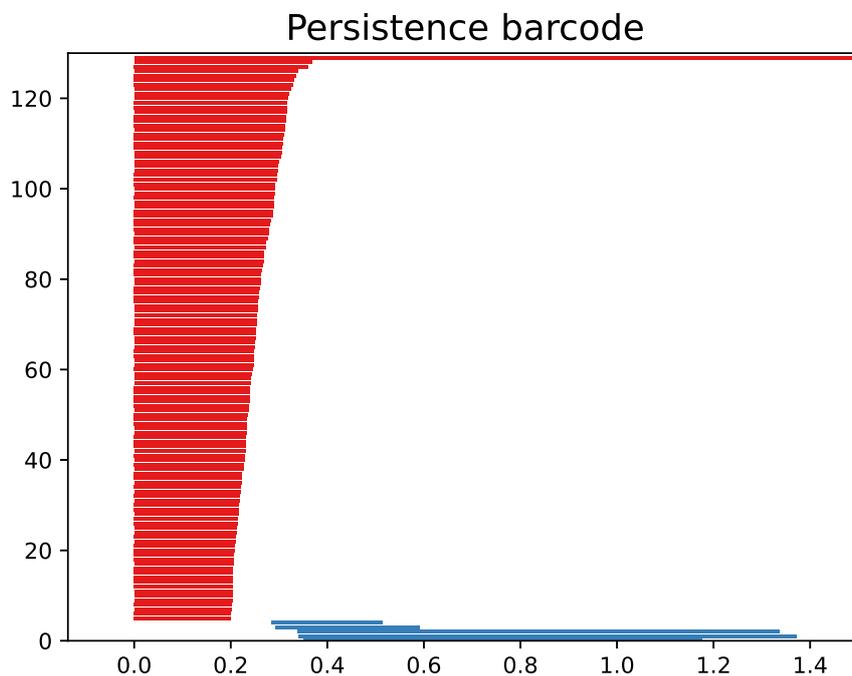
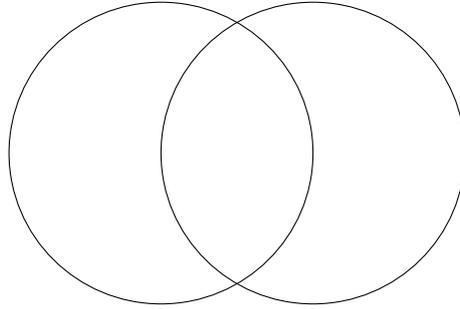


Figura 1.7: A figura representa o barcode obtido a partir do dataset da Figura 1.6. As barras na cor vermelha representam as classes de H_0 e as azuis H_1 . Escolhemos um limitante inferior para o tamanho mínimo das barras de 0.2 unidades, para tentar descartar ruídos insignificativos da homologia de grau zero e tornar a figura mais compacta. Perceba que a terceira barra que representa H_1 esta colada ao eixo horizontal, tornando difícil sua visualização.

Utilizando a biblioteca `Gudhi`, construímos o complexo de Vietoris-Rips e subsequentemente seu respectivo barcode presente na Figura 1.7. Fica evidente que o dataset apresentado foi gerado a partir da seguinte figura



portanto, esperamos que o número de Betti β_1 de nosso dataset se comporte como o β_1 da intersecção das circunferências. Escolhemos esse espaço em específico justamente porque é homotópico ao buquet de 3 círculos e sabemos que o grupo de homologia é igual a $\mathbb{Z} \oplus \mathbb{Z} \oplus \mathbb{Z}$, portanto tem número de Betti β_1 igual a 3. Perceba que olhando para as barras azuis na Figura 1.7 temos três barras que se destacam e as demais representam o ruído no sistema.

Fica evidente que dada uma quantidade de pontos grande, podemos gerar uma grande quantidade de barras em nosso barcode. Isso pode vir a atrapalhar nossa análise dos dados, assim gostaríamos de apresentar uma forma equivalente de trabalhar com uma representação gráfica das classes de homologia persistente e que seja mais compacta. Antes disso, precisamos da seguinte definição.

Definição 1.1.16. Um *multiset* é um par $A = (S, m)$, onde S é um conjunto e

$$m : S \rightarrow \mathbb{N} \cup \{\infty\}$$

é a função que mede quantas vezes um elemento em S se repete.

Obviamente, quando quisermos tratar da cardinalidade de A como multiset, devemos contar todas as vezes que um certo elemento se repete em A , isto é,

$$|A| = \sum_{s \in S} m(s).$$

Definição 1.1.17. Para cada grau de homologia persistente p , podemos considerar o subconjunto de $\mathbb{R}^2 \cup \{\infty\}$ formado pelos pontos da forma (b, d) , que representa uma classe de homologia persistente nascida em b e que persiste até d , contado suas respectivas multiplicidades. Tal conjunto chamamos de **diagrama de persistência**.

A definição do diagrama de persistência que desperta a necessidade de considerarmos os multisets. Perceba que mesmo que duas classes de persistência homológica podem

nascer e morrer no mesmo instante, elas são representadas por intervalos diferentes no barcode, já no diagrama de persistência elas são representadas pelo mesmo ponto.

Exemplo 1.1.18. *Para um exemplo de um diagrama de persistência, podemos utilizar o dataset criado no Exemplo 1.1.15.*

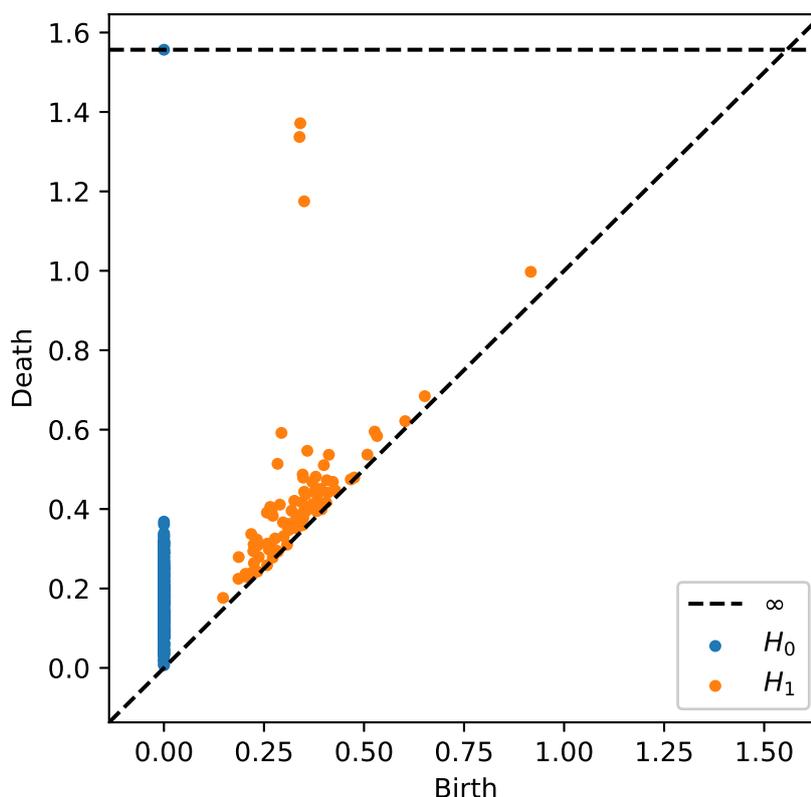


Figura 1.8: Diagrama de persistência do dataset do Exemplo 1.1.15. Tal diagrama foi criado utilizando a biblioteca Ripser.

Exemplo 1.1.19. *O Exemplo 1.1.15 nos serve para dar uma ideia de como aplicar a teoria desenvolvida. Entretanto a TDA brilha realmente com bancos de dados com dimensão alta. Assim apresentamos aqui um exemplo que não possa ser plotado em \mathbb{R}^3 . No caso, escolhemos um banco de dados artificial de pontos que estão sobre uma Garrafa de Klein. Escolhemos tal objeto pois o Teorema da Imersão de Whitney (veja Capítulo 6 em [22]) garante que a Garrafa de Klein pode ser mergulhada em \mathbb{R}^4 , porém diferente do toro \mathbb{T}^2 , não pode ser mergulhada em \mathbb{R}^3 .*

Para a confecção de tal banco de dados usando a parametrização usual

$$\begin{cases} x = (R + r \cos(\varphi)) \cos(\theta) \\ y = (R + r \cos(\varphi)) \sin(\theta) \\ z = (R + r \sin(\varphi)) \cos(\theta/2) \\ w = (R + r \sin(\varphi)) \sin(\theta/2) \end{cases}$$

onde $R, r \in \mathbb{R}$ e $\varphi, \theta \in [0, 2\pi)$. No caso, escolhemos $R = 4$ e $r = 2$ e novamente, como no Exemplo 1.1.15, utilizamos `random()` para escolher pontos aleatórios sobre a variedade. Nosso espaço amostral para este exemplo se trata de 15000 pontos.

Vale a pena destacar que para tornar a computação do diagrama de persistência dessa quantidade de pontos factível, tomamos `n_perm = 300` como parâmetro ao utilizarmos a biblioteca `ripsr`.

Além disso, a biblioteca `ripsr` usa por padrão coeficientes no corpo \mathbb{Z}_2 , para escolher coeficientes em outros corpos \mathbb{Z}_p , tome o parâmetro `coeff = p`. Para mais informações sobre, sugerimos uma leitura na documentação da biblioteca `Ripsr.py`, [32].

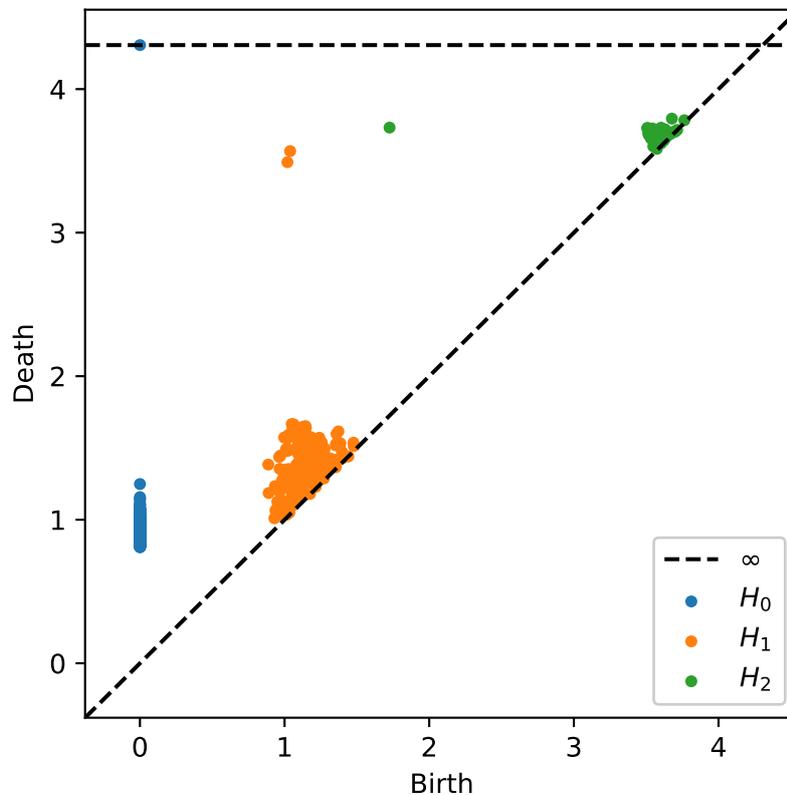


Figura 1.9: Diagrama de persistência do dataset dos pontos criados aleatoriamente sobre uma garrafa de Klein em \mathbb{R}^4 .

1.2 Módulos de Persistência

Até o final desta seção, consideraremos todos os espaços vetoriais descritos aqui sobre um corpo \mathbb{K} qualquer, previamente fixado.

A ideia desta seção é, de certa forma, generalizar alguns dos conceitos da seção anterior sobre a Persistência Homológica, na esperança de obter resultados significativos que nos assegurem aplicar esta teoria em um algoritmo robusto. Para isso, ampliaremos a definição de Persistência Homológica, para estudar quando um espaço X pode ter sua homologia persistente decomposta em barcodes.

Definição 1.2.1. *Um **módulo de persistência** sobre os números reais V , é uma família de espaços vetoriais indexada em \mathbb{R}*

$$\{V_t\}_{t \in \mathbb{R}}$$

junto com uma família duplamente indexada de transformações lineares

$$\{v_s^t : V_s \rightarrow V_t \mid s \leq t\},$$

que satisfazem a seguinte lei de composição

$$v_s^t \circ v_r^s = v_r^t,$$

sempre que $r \leq s \leq t$ e v_i^i representa a identidade em V_i .

A família duplamente indexada de transformações lineares serve para nos mostrar como os espaços vetoriais são encaixados uns nos outros. Comparando com a teoria já exposta na seção anterior, é como se tivéssemos uma filtração de homologia persistente, entretanto não estamos lidando mais com uma quantidade discreta das mesmas.

Fica evidente que não precisamos ter obrigatoriamente um módulo de persistência indexado na reta real, podemos indexar sobre qualquer conjunto parcialmente ordenado T , entretanto, costumamos usar \mathbb{R} por conveniência. Deixaremos claro quando estivermos indexando um módulo de persistência em um conjunto que não seja os números reais.

Definição 1.2.2. *Um **homomorfismo entre módulos de persistência** U e V , ϕ , é uma coleção de transformações lineares*

$$\{\phi_t : U_t \rightarrow V_t \mid t \in \mathbb{R}\}$$

tal que o seguinte diagrama comuta para $s \leq t$

$$\begin{array}{ccc}
U_s & \xrightarrow{u_s^t} & U_t \\
\phi_s \downarrow & \circlearrowleft & \downarrow \phi_t \\
V_s & \xrightarrow{v_s^t} & V_t
\end{array}$$

Se cada ϕ_t for um isomorfismo, chamamos ϕ de um **isomorfismo entre módulos de persistência**.

Definição 1.2.3. Seja $T \subseteq \mathbb{R}$ e $J \subset T$ um intervalo de T , isto é, se $r < s < t$ são elementos de T tais que $r, t \in J$, então $s \in J$. Definimos \mathbb{I}_J o T -módulo de persistência com espaços

$$I_t = \begin{cases} \mathbb{K}, & \text{se } t \in J \\ 0, & \text{caso contrário.} \end{cases}$$

e aplicações

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

O módulo \mathbb{I}_J será chamado de **módulo de intervalo**.

Essa definição de módulo de intervalo faz referência aos barcodes apresentados na seção anterior. A pergunta natural é: quando podemos decompor um espaço nessa versão mais geral de barcodes? Vamos tentar responder isso, mas antes apresentamos algumas definições e resultados essenciais.

Definição 1.2.4. Definimos a **soma direta de dois módulos de persistência** U e V , $W = U \oplus V$ como sendo

$$W_t = U_t \oplus V_t; \quad w_s^t = u_s^t \oplus v_s^t.$$

onde

$$u_s^t \oplus v_s^t(x, y) = (u_s^t(x), v_s^t(y)).$$

Definição 1.2.5. Dizemos que um módulo de persistência W é **indecomponível** se $W = U \oplus V$ implicar em $V = 0$ ou $U = 0$. Dizemos que W é **decomponível** se W não for indecomponível.

Consideramos

$$End(V) := \{\varphi : V \rightarrow V \mid \varphi \text{ é homomorfismo}\}.$$

Lema 1.2.6. Seja \mathbb{I}_J um módulo de intervalo sobre $T \subseteq \mathbb{R}$, então $End(\mathbb{I}_J) = \mathbb{K}$.

Basta apenas um breve comentário para elucidar a validade deste resultado. Qualquer endomorfismo de \mathbb{I}_J age apenas nas partes não nulas por uma multiplicação por escalar, isto é, age em $I_t = \mathbb{K}$, $t \in J$. Por outro lado, da comutatividade imposta na Definição 1.2.2 devemos ter o mesmo escalar para todo $t \in J$ e o resultado segue.

Proposição 1.2.7. *Módulos de Intervalos são indecomponíveis.*

Demonstração. Suponha que $\mathbb{I}_J = U \oplus V$ seja uma decomposição de \mathbb{I}_J . As projeções sobre U e V são elementos idempotentes no anel de endomorfismos de \mathbb{I}_J . Como \mathbb{K} é corpo, os únicos idempotentes são 0 e 1 e o resultado segue. ■

Agora, gostaríamos de estudar os módulos de persistência que podem ser decompostos em módulos de intervalos. Seja $\{J_a \mid a \in L\}$ um multiset de intervalos. Queremos determinar quando um módulo de persistência pode ser escrito da seguinte forma

$$V = \bigoplus_{a \in L} \mathbb{I}_{J_a}.$$

Perceba que o fato de $\{J_a \mid a \in L\}$ ser um multiset é devido que o mesmo intervalo pode ter multiplicidade maior que 1 quando formos escrever a decomposição de V .

Teorema 1.2.8 (Krull-Remak-Schmidt-Azumaya). *Suponha que um módulo de persistência V sobre $T \subset \mathbb{R}$ possa ser decomposto em uma soma direta de módulos de intervalos de duas maneiras diferentes*

$$V = \bigoplus_{a \in L} \mathbb{I}_{J_a} = \bigoplus_{m \in M} \mathbb{I}_{K_m}.$$

Então existe uma bijeção $\varphi : L \rightarrow M$ tal que $J_a = K_{\varphi(a)}$ para todo a .

Em outras palavras, quando um módulo de persistência for decomponível em módulos de intervalos, essa decomposição é única a menos de reordenações. A demonstração do Teorema 1.2.8 pode ser encontrada em [4].

Teorema 1.2.9 (Crawley-Boevey). *Todo módulo de persistência de dimensão pontualmente finita pode ser decomposto em módulos de intervalos.*

A demonstração deste resultado pode ser encontrada em [7].

Definição 1.2.10. *Considere B e B' dois barcodes. Um **pareamento** entre B e B' é uma bijeção ξ entre os conjuntos $\text{dom}(\xi) \subset B$ e $\text{Im}(\xi) \subset B'$. A função ξ geralmente é chamada de **pareamento parcial**. Definimos o **custo** do pareamento ξ , como sendo*

$$\max \left\{ \max_{I \in \text{dom}(\xi)} \|I - \xi(I)\|_\infty, \max_{I \in B \setminus \text{dom}(\xi)} \|I\|_\Delta, \max_{I' \in B' \setminus \text{Im}(\xi)} \|I'\|_\Delta \right\},$$

onde, para I e I' intervalos nos barcodes B e B' , respectivamente, com extremidades $b \leq d$ e $b' \leq d'$,

$$\|I - I'\|_\infty := \max\{|b - b'|, |d - d'|\} \quad e \quad \|I\|_\Delta := \frac{d - b}{2}.$$

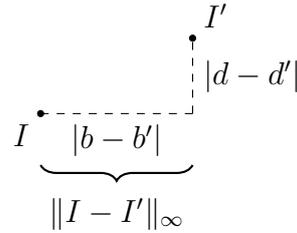
Caso um pareamento ξ tenha custo menor ou igual a ε , dizemos que ξ é um ε -pareamento.

A **distância bottleneck** é dada por

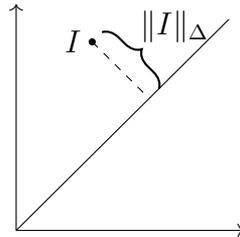
$$d_B(B, B') := \inf\{\varepsilon \geq 0 \mid \text{existe um } \varepsilon\text{-pareamento de } B \text{ e } B'\}.$$

Um pareamento que realiza a distância bottleneck é chamado de **pareamento ótimo**.

A intuição por trás dessa definição é a seguinte: gostaríamos de fazer uma bijeção entre os diagramas de persistência de B e B' , entretanto não temos necessariamente uma correspondência entre esses objetos, assim surge a importância de fazer uma bijeção entre os subconjuntos dos barcodes B e B' . Para os intervalos que estão na bijeção que criamos aplicamos $\|I - I'\|_\infty$



Para os intervalos que não foram contemplados na bijeção ξ , fazemos a distância $\|I\|_\Delta$, que nos dá o quanto o intervalo dista da diagonal do diagrama de persistência



Feito esses procedimentos, tomamos o máximo dos maiores valores encontrado nesses dois processos apresentados acima. Isso representa o custo da bijeção ξ . A distância bottleneck é aquela que minimaliza esse custo.

Gostaríamos de garantir que a teoria que estamos desenvolvendo aqui seja de certa forma contínua. Isto é, se alterarmos nosso dataset inicial de maneira pequena, nos perguntamos se essa perturbação nos retorna diagramas de persistência semelhantes no final do processo ou não. Usualmente, esse tipo propriedade é conhecida como Estabilidade. Perguntar se nossa teoria tem essa estabilidade é tentar garantir a robustez desta teoria. Antes de enunciar o Teorema de Estabilidade, daremos uma definição relevante para a compreensão do enunciado do Teorema.

Definição 1.2.11. *Sejam X um espaço topológico e uma $f : X \rightarrow \mathbb{R}$ uma função. Um **valor crítico homológico** de f é um valor real a para o qual existe um inteiro k tal que para $\varepsilon > 0$ suficientemente pequeno, a função*

$$i_* : H_k(f^{-1}(-\infty, a - \varepsilon]) \rightarrow H_k(f^{-1}(-\infty, a + \varepsilon])$$

induzida pela inclusão, não é um isomorfismo.

Perceba que pedir que i_* não seja um isomorfismo é o que nos dá um ponto de mudança nos grupos de homologia persistente, alterando seu barcode respectivamente.

Definição 1.2.12. *Uma função $f : X \rightarrow \mathbb{R}$ é dita ser **tame** se ela tiver uma quantidade finita de valores críticos homológicos e os grupos de homologia $H_k(f^{-1}(\infty, a])$ têm dimensão finita para todo $k \in \mathbb{Z}$ e $a \in \mathbb{R}$.*

Teorema 1.2.13 (Estabilidade). *Seja X um espaço topológico triangularizável com funções contínuas tame $f, g : X \rightarrow \mathbb{R}$. Então os diagramas de persistência satisfazem*

$$d_B(D(f), D(g)) \leq \|f - g\|_\infty,$$

onde $D(f)$ e $D(g)$ representam os diagramas de f e g respectivamente.¹

Esse resultado foi primeiramente enunciado e demonstrado em [5]. Perceba como esse primeiro resultado é de certa forma restritivo, pois impomos que X seja triangularizável e que f seja contínua e tame. Não iremos provar esse resultado aqui, pois no Capítulo 2, com o auxílio da Teoria das Categorias, provaremos um resultado um pouco mais geral, que não pede tanto das hipóteses de nosso espaço topológico e da função que trabalhamos.

¹No caso, um diagrama de persistência proveniente de uma função $f : X \rightarrow \mathbb{R}$ é o diagrama de persistência originado através da filtração formada pela sequência encaixante de conjuntos $f^{-1}(-\infty, a]$. Perceba como isso se relaciona com a Definição 1.2.12.

Capítulo 2

Categorias

Nosso instinto de matemático nos força a generalizar tudo que tocamos, assim o estudo da Teoria das Categorias nos vem de maneira natural. A Teoria das Categorias introduzida primeiramente por Eilenberg e MacLane em [15] surge como tentativa de generalizar certas estruturas recorrentes em toda matemática.

Como o próprio Spanier escreve em seu texto [30], a Topologia Algébrica é em si o estudo de funtores entre categorias topológicas e categorias algébricas. Desta forma, seria uma lástima não abordarmos o estudo de categorias neste trabalho.

Apresentaremos neste capítulo o mínimo da teoria básica de categorias para que o texto faça sentido num contexto de TDA e em seguida, traduziremos alguns dos conceitos apresentados no Capítulo 1 para a linguagem das categorias.

Os resultados da Teoria das Categorias que serão expostos a seguir podem ser encontrados em [1], [2], [20], [25] e [31]. Recomendamos ainda [15] para uma leitura histórica, voltada para o entendimento da motivação da criação de tal teoria, e [30] como uma referência da teoria básica de Topologia Algébrica usando a Teoria das Categorias. Por fim, para a conversão da teoria clássica de TDA para o mundo das categorias, nos apoiamos fortemente em [3] e [8].

2.1 Teoria Básica

Definição 2.1.1. *Uma **categoria** C consiste de uma classe de objetos, $Ob(C)$, e para cada par de objetos $X, Y \in Ob(C)$, temos um conjunto de **morfismos**, $Hom_C(X, Y)$, cujos elementos são usualmente expressos por $f : X \rightarrow Y$, sendo X o domínio e Y o contra-domínio. Para uma tripla de objetos $X, Y, Z \in Ob(C)$ fica definida a função **composição***

$$\begin{aligned} \circ : Hom_C(Y, Z) \times Hom_C(X, Y) &\rightarrow Hom_C(X, Z) \\ (g, f) &\mapsto g \circ f, \end{aligned}$$

que satisfaz a associatividade, isto é, $(h \circ g) \circ f = h \circ (g \circ f)$. Além disso, para todo objeto $X \in \text{Ob}(C)$ está definido o **morfismo identidade**

$$1_X : X \rightarrow X,$$

que satisfaz

$$f \circ 1_X = f \quad 1_X \circ g = g$$

para $f \in \text{Hom}_C(X, Y)$ e $g \in \text{Hom}_C(W, X)$, com $Y, W \in \text{Ob}(C)$ quaisquer.

É comum omitir tanto o subíndice C em $\text{Hom}_C(X, Y)$ quando estiver claro de qual categoria estamos tratando, quanto omitir \circ na composição de morfismos, transformando $g \circ f$ em apenas gf .

Exemplo 2.1.2.

1. **Sets** é a categoria formada por objetos sendo os conjuntos e os morfismos sendo as funções entre eles.
2. **Top** é a categoria constituída por espaços topológicos como os objetos e os morfismos sendo as funções contínuas entre eles.
3. **Vec** é a categoria com objetos sendo os espaços vetoriais e morfismos sendo as transformações lineares entre eles.
4. **vec** é a categoria com objetos sendo os espaços vetoriais de dimensão finita e morfismos sendo as transformações lineares entre eles.

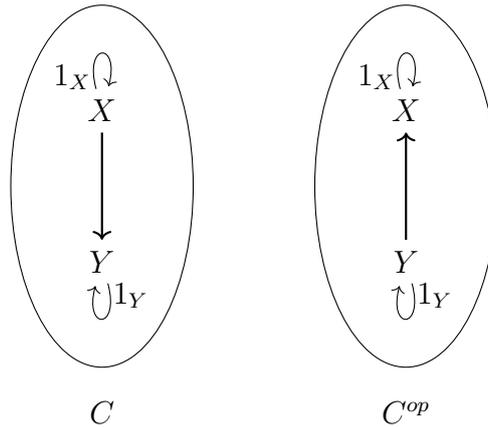
Esses exemplos apresentados acima serão usados recorrentemente durante o texto.

Definição 2.1.3. Uma categoria C é dita ser **pequena** se ao invés de uma classe de objetos, tivermos um conjunto de objetos. Caso os únicos morfismos entre os objetos de C sejam as identidades, dizemos que C é uma categoria **discreta**

Exemplo 2.1.4. A categoria pequena **sets**, diferente da categoria **Sets**, é formada por objetos sendo todos os conjuntos fixando um conjunto universo U e morfismos sendo as funções entre esses tais conjuntos.

Definição 2.1.5. Seja C uma categoria. Definimos a **categoria dual** de C , C^{op} , como sendo a categoria formada por objetos iguais aos objetos de C e para um par de objetos $X, Y \in \text{Ob}(C^{op})$ definimos $\text{Hom}_{C^{op}}(X, Y) = \text{Hom}_C(Y, X)$.

Em outras palavras, se pensarmos em uma categoria C como sendo um grafo orientado, trabalhar em C^{op} é apenas reverter a orientação do grafo C



Definição 2.1.6. Sejam C e D duas categorias. Um **funtor (covariante)** $F : C \rightarrow D$ consiste de

1. Uma aplicação

$$F : \text{Ob}(C) \rightarrow \text{Ob}(D)$$

$$X \mapsto F(X).$$

2. Para cada par de objetos $X, Y \in \text{Ob}(C)$, temos uma aplicação

$$F : \text{Hom}_C(X, Y) \rightarrow \text{Hom}_D(F(X), F(Y))$$

$$f \mapsto F(f)$$

tal que F preserva a composição de morfismos e a identidade, isto é,

$$F(g \circ f) = F(g) \circ F(f) \quad \text{e} \quad F(1_X) = 1_{F(X)},$$

quando $g \circ f$ estiver bem definido na categoria C .

Um **funtor contravariante** é definido de maneira análoga, porém utilizando C^{op} ao invés de C . Usualmente chamamos essas propriedades que um funtor satisfaz de **propriedade functorial**, ou ainda, de **functorialidade**.

Exemplo 2.1.7. Seja $F : \mathbf{Top} \rightarrow \mathbf{Sets}$ um funtor que leva $X \in \text{Ob}(\mathbf{Top})$ em $X \in \text{Ob}(\mathbf{Sets})$, isto é, F leva cada espaço topológico em seu conjunto subjacente. Além disso, para cada morfismo f em \mathbf{Top} , ou seja, f é uma função contínua entre espaços topológicos, temos $F(f)$ levado na função de conjuntos que define f . Fica evidente que a função F é de fato um funtor covariante, já que as propriedades functoriais são trivialmente satisfeitas. De certa maneira, podemos dizer que o funtor F “esquece” das propriedades da categoria \mathbf{Top} e a trata como a categoria \mathbf{Sets} . Assim, esse funtor recebe o nome sugestivo de **forgetful functor**, ou ainda, **funtor esquecimento**.

Definição 2.1.8. *Seja C uma categoria qualquer. Se C admitir um funtor $F : C \rightarrow \mathbf{Sets}$ com propriedades análogas as do forgetful functor apresentadas acima, chamaremos C de uma **categoria concreta**.*

Exemplo 2.1.9. *Seja X um espaço topológico qualquer. Sabemos que X pode ser expresso como a união de suas componentes conexas. Seja $\pi_0(X)$ o conjunto das componentes conexas de X . Com as informações dadas acima, fica direto verificar que*

$$\pi_0 : \mathbf{Top} \rightarrow \mathbf{Sets}$$

que leva cada espaço topológico no conjunto de suas componentes conexas é um funtor, uma vez que morfismos entre objetos de \mathbf{Top} também são morfismos entre objetos de \mathbf{Sets} .

Definição 2.1.10. *Sejam C uma categoria e $f \in \text{Hom}_C(X, Y)$ um morfismo. Dizemos que f é um **monomorfismo** se para morfismos $g, h \in \text{Hom}_C(W, X)$, com $W \in \text{Ob}(C)$ qualquer, tal que $fg = fh$ valer $g = h$. Analogamente, dizemos que f é um **epimorfismo** se para morfismos $g, h \in \text{Hom}_C(Y, Z)$, com $Z \in \text{Ob}(C)$ qualquer, tal que $gf = hf$ valer $g = h$.*

Definição 2.1.11. *Seja C uma categoria. Dizemos que um par de objetos $X, Y \in \text{Ob}(C)$ são **isomorfos**, se existirem morfismos $f : X \rightarrow Y$ e $g : Y \rightarrow X$ tais que*

$$fg = 1_Y \quad e \quad gf = 1_X.$$

*Quando isso acontecer, f e g são chamados de **isomorfismos**.*

Segue diretamente das definições que um isomorfismo é simultaneamente monomorfismo e epimorfismo. Consideramos a categoria \mathbf{Grps} constituída pelos grupos como objetos e morfismos sendo os homomorfismos de grupos. Usualmente, homomorfismos de grupos injetores, sobrejetores e bijetores são chamados de monomorfismo, epimorfismo e isomorfismo respectivamente.

Em \mathbf{Grps} vale a seguinte propriedade: um homomorfismo que é ao mesmo tempo monomorfismo e epimorfismo é automaticamente um isomorfismo. O nosso costume com a Teoria de Grupos e os nomes dos morfismos parecidos, nos faz acreditar que essa propriedade vale para a Teoria das Categorias em geral, entretanto destacamos que isso nem sempre é verdade.

Exemplo 2.1.12. *Seja P um conjunto qualquer munido de uma ordem parcial \preceq , geralmente chamado de **poset**. Podemos lidar com (P, \preceq) como sendo a categoria P onde $\text{Ob}(P) = P$ e os morfismos entre dois objetos $a, b \in \text{Ob}(P)$ admite no máximo um morfismo, geralmente denotado por $a \preceq b$.*

Como nessa categoria P dois objetos $a, b \in \text{Ob}(P)$ admitem no máximo um morfismo, temos que qualquer morfismo é monomorfismo e epimorfismo. Entretanto, apenas a identidade pode ser isomorfismo. De fato, sejam a, b objetos em P isomorfos, então existem morfismos $f : a \rightarrow b$, denotado por $a \preceq b$ e $g : b \rightarrow a$, denotado por $b \preceq a$ tais que

$$fg = 1_b \quad e \quad gf = 1_a.$$

Temos assim que $a \preceq b$ e $b \preceq a$, segue da propriedade antissimétrica da ordem parcial que $a = b$. Assim mostramos que para quaisquer $a, b \in \text{Ob}(P)$ distintos, temos que morfismos que são simultaneamente monomorfismo e epimorfismo mas não isomorfismo.

Proposição 2.1.13. *Seja $F : C \rightarrow D$ um funtor. Funtores preservam isomorfismos, isto é, se $X, Y \in \text{Ob}(C)$ tal que $X \cong Y$ então $F(X) \cong F(Y)$.*

Demonstração. De fato, se $X \cong Y$, então existem $f : X \rightarrow Y$ e $g : Y \rightarrow X$ tais que

$$f \circ g = 1_Y \quad e \quad g \circ f = 1_X.$$

Aplicamos o funtor F e segue da propriedade funtorial que

$$F(f) \circ F(g) = F(f \circ g) = F(1_Y) = 1_{F(Y)}$$

$$F(g) \circ F(f) = F(g \circ f) = F(1_X) = 1_{F(X)}.$$

Concluimos que existem morfismos entre $F(X)$ e $F(Y)$ que são inversos entre si, logo $F(X) \cong F(Y)$. ■

Definição 2.1.14. *Sejam C e D categorias, $F, G : C \rightarrow D$ funtores entre essas categorias. Uma **transformação natural** $\eta : F \Rightarrow G$ é um morfismo tal que para todo $X \in \text{Ob}(C)$ temos $\eta_X : F(X) \rightarrow G(X)$ em D tal que se $f : X \rightarrow Y$ for um morfismo em C , o seguinte diagrama comuta*

$$\begin{array}{ccc} F(X) & \xrightarrow{\eta_X} & G(X) \\ F(f) \downarrow & \circlearrowleft & \downarrow G(f) \\ F(Y) & \xrightarrow{\eta_Y} & G(Y) \end{array}$$

É comum nos referirmos à η_X , como sendo a **componente** X de η . Quando η_X for isomorfismo para todo $X \in \text{Ob}(C)$ dizemos que η é um **isomorfismo natural**.

Considere a categoria **CRngs**, que consiste da classe dos anéis comutativos como objetos e os homomorfismo de anéis como morfismo, para o nosso próximo exemplo de transformação natural.

Exemplo 2.1.15. *Seja M uma matriz quadrada de ordem n com coeficientes em um anel comutativo A , isto é, $M \in M_n(A)$ e considere $(A)^\times$ o grupo das unidades de A .*

Consideremos $GL_n : \mathbf{CRngs} \rightarrow \mathbf{Grps}$ o funtor que leva um anel comutativo A em $GL_n(A)$, e para cada morfismo f de \mathbf{CRngs} ele aplica f em cada entrada de $GL_n(A)$. Analogamente, podemos considerar $()^\times : \mathbf{CRngs} \rightarrow \mathbf{Grps}$ como sendo um funtor que leva cada anel $A \in \text{Ob}(\mathbf{CRngs})$ em $(A)^\times \in \text{Ob}(\mathbf{Grps})$. Assim, para um morfismo de anéis comutativos $f : A \rightarrow A'$, temos o seguinte diagrama comutativo

$$\begin{array}{ccc} GL_n(A) & \xrightarrow{\det_A} & (A)^\times \\ GL_n(f) \downarrow & \circlearrowleft & \downarrow (f)^\times \\ GL_n(A') & \xrightarrow{\det_{A'}} & (A')^\times \end{array}$$

Isso nos mostra que $\det : GL_n \Rightarrow ()^\times$ é uma transformação entre dois funtores entre as categorias \mathbf{CRngs} e \mathbf{Grps} .

Definição 2.1.16. *Sejam C e D duas categorias, $F, G, H : C \rightarrow D$ funtores entre essas duas categorias, $\varphi : F \Rightarrow G$ e $\psi : G \Rightarrow H$ transformações naturais. Definimos a **composição vertical** entre φ e ψ , denotada por $\psi \cdot \varphi$, como sendo a transformação natural onde para cada $X \in \text{Ob}(C)$ temos a componente dada por*

$$(\psi \cdot \varphi)_X := \psi_X \circ \varphi_X.$$

Definição 2.1.17. *Dadas duas categorias C e D , fica definida a categoria $\mathbf{Fun}(C, D)$, como sendo a categoria cujo objetos são os funtores de C em D e os morfismos são as transformações naturais entre os funtores. Fica claro que utilizamos a Definição 2.1.16 para definir o que vem a ser uma composição de morfismos na categoria $\mathbf{Fun}(C, D)$.*

Definição 2.1.18. *Sejam C uma categoria, $X, Y \in \text{Ob}(C)$. Definimos o **produto de X e Y** em C , $X \times Y$, como sendo um objeto em C munido de dois morfismos $\pi_1 : X \times Y \rightarrow X$, $\pi_2 : X \times Y \rightarrow Y$, usualmente chamados de projeções, que satisfazem a **Propriedade Universal**: dado um objeto Z e dois morfismos $\varphi : Z \rightarrow X$ e $\psi : Z \rightarrow Y$, existe único morfismo $f : Z \rightarrow X \times Y$ fazendo o seguinte diagrama comutar*

$$\begin{array}{ccccc} & & Z & & \\ & \swarrow \varphi & \downarrow f & \searrow \psi & \\ X & \xleftarrow{\pi_1} & X \times Y & \xrightarrow{\pi_2} & Y \end{array}$$

Na definição acima, a Propriedade Universal tem um papel importante de garantir que o produto $X \times Y$ é o objeto que se melhor encaixa para esta situação. De fato, se tivermos

um outro candidato a produto categorial, digamos $(X \times Y)'$, segue da Definição 2.1.18 que temos o seguinte diagrama comutativo

$$\begin{array}{ccccc} & & (X \times Y)' & & \\ & \swarrow \pi'_1 & & \searrow \pi'_2 & \\ X & & & & Y \\ & \swarrow \pi_1 & & \searrow \pi_2 & \\ & & X \times X & & \end{array}$$

Perceba que $f \circ f'$ é um morfismo de $X \times Y$ em $X \times Y$, temos assim

$$\pi'_1 = \pi_1 \circ f \quad \text{e} \quad \pi_1 = \pi'_1 \circ f',$$

o que implica em

$$\pi_1 = \pi_1 \circ (f \circ f').$$

Analogamente para π_2 temos que $\pi_2 = \pi_2 \circ (f \circ f')$. Assim $f \circ f'$ faz o seguinte diagrama comutar

$$\begin{array}{ccccc} & & X \times Y & & \\ & \swarrow \pi_1 & & \searrow \pi_2 & \\ X & & & & Y \\ & \swarrow \pi_1 & & \searrow \pi_2 & \\ & & X \times Y & & \end{array}$$

Por outro lado, existe outro morfismo que faz esse diagrama comutar, que é a identidade. Segue da Propriedade Universal que $f \circ f' = 1_{X \times Y}$. Com argumentos análogos, podemos mostrar que $f' \circ f = 1_{(X \times Y)'}$. Assim concluímos que $X \times Y$ e $(X \times Y)'$ são isomorfos.

Definição 2.1.19. *Sejam C uma categoria, $X, Y \in \text{Ob}(C)$, definimos o **coproduto de X e Y** , $X \sqcup Y$, como sendo a construção dual ao do produto $X \times Y$, isto é, é um objeto de C munido de morfismos $i : X \rightarrow X \sqcup Y$ e $j : Y \rightarrow X \sqcup Y$, usualmente chamados de **injeções** (mesmo que em geral não pedimos que estas funções sejam injetoras), tais que satisfazem a **Propriedade Universal**: dado um objeto Z e dois morfismos $\varphi : X \rightarrow Z$ e $\psi : Y \rightarrow Z$, existe único morfismo $f : X \sqcup Y \rightarrow Z$ fazendo o seguinte diagrama comutar*

$$\begin{array}{ccccc} & & Z & & \\ & \nearrow \varphi & & \nwarrow \psi & \\ X & \xrightarrow{i} & X \sqcup Y & \xleftarrow{j} & Y \end{array}$$

Exemplo 2.1.20. *Em **Sets**, o produto categorial corresponde ao produto cartesiano dotado das projeções canônicas e o coproduto à união disjunta com as inclusões. Em **Top** o produto corresponde ao produto de espaços topológicos com a topologia produto com as projeções canônicas e o coproduto à união disjunta dos espaços topológicos com as inclusões naturais.*

Essa noção de coproduto categorial será muito relevante para que possamos desenvolver a teoria Merge Trees, mais especificamente para definir Merge Trees Decorada Categorial, uns dos tópicos principais deste trabalho, que será abordada no Capítulo 3.

Definição 2.1.21. *Sejam C, D categorias e $F : C \rightarrow D$ um funtor qualquer. Dados dois objetos $X, Y \in Ob(C)$ consideramos a seguinte aplicação*

$$\begin{aligned} \varphi_{xy} : Hom_C(X, Y) &\rightarrow Hom_D(F(X), F(Y)) \\ f &\mapsto F(f) \end{aligned}$$

Dizemos que F é **cheio** se para cada par de objetos $X, Y \in Ob(C)$ φ_{xy} for sobrejetora. F é **fiel** se para quaisquer X, Y tivermos φ_{xy} injetora. Por fim, dizemos que F é **essencialmente sobrejetor** sobre objetos se para todo $Z \in Ob(D)$ existe algum $X \in Ob(C)$ tal que Z e $F(X)$ sejam isomorfos.

Gostaríamos de determinar quando duas categorias se comportam da mesma forma, para que possamos tratá-las como essencialmente a mesma, assim como fazemos com objetos isomorfos de uma categoria C . Entretanto, pedir isomorfismo entre duas categorias é um requisito muito forte para que possamos fazer coisas relevantes com a Teoria de Categorias, então enfraqueçamos um pouco essa noção, resultando na próxima definição.

Definição 2.1.22. *Sejam C, D duas categorias. Dizemos que C e D são **equivalentes** se existirem funtores $F : C \rightarrow D$ e $G : D \rightarrow C$ junto com isomorfismos naturais $\eta : 1_C \cong G \circ F$ e $\varepsilon : F \circ G \cong 1_D$. Quando C e D forem equivalentes, escrevemos $C \simeq D$.*

É trivial mostrar que a equivalência de categorias definida acima define uma relação de equivalência.

Lema 2.1.23. *Seja $F : C \rightarrow D$ um funtor fiel e cheio.*

- (i) *f é um morfismo em C tal que $F(f)$ é isomorfismo em D se, e somente se, f for isomorfismo.*
- (ii) *$X, Y \in Ob(C)$ tais que $F(X) \cong F(Y)$ em D se, e somente se, $X \cong Y$ em C .*

Demonstração. Perceba que segue da Proposição 2.1.13 que funtores preservam isomorfismos e assim as afirmações recíprocas seguem diretamente deste fato. Nos concentramos nas afirmações diretas dos itens deste Lema.

(i) Como F é fiel e cheio, temos então que φ_{xy} é bijetora para quaisquer $X, Y \in Ob(C)$. Por hipótese $F(f)$ é um isomorfismo em D , então existe $g : F(Y) \rightarrow F(X)$ tal que

$$F(f) \circ g = 1_{F(Y)} \quad \text{e} \quad g \circ F(f) = 1_{F(X)}.$$

Como φ_{yx} é sobrejetora, então existe $g' \in \text{Hom}_C(Y, X)$ tal que

$$\varphi_{yx}(g') = F(g') = g.$$

Perceba que $f \circ g' : Y \rightarrow Y$, então temos que

$$\varphi_{yy}(f \circ g') = F(f \circ g') = F(f) \circ F(g') = F(f) \circ g = 1_{F(Y)} = F(1_Y) = \varphi_{yy}(1_Y).$$

Como φ_{yy} é injetora por hipótese temos que $f \circ g' = 1_Y$. Um argumento análogo nos mostra que $g' \circ f = 1_X$, concluimos assim que f é isomorfismo.

(ii) É um argumento análogo ao item anterior. Se $F(X) \cong F(Y)$ então existem $f : F(X) \rightarrow F(Y)$ e $g : F(Y) \rightarrow F(X)$ tais que

$$f \circ g = 1_{F(Y)} \quad \text{e} \quad g \circ f = 1_{F(X)}.$$

Da sobrejetividade de φ_{xy} e φ_{yx} existem $f' \in \text{Hom}_C(X, Y)$ e $g' \in \text{Hom}_C(Y, X)$ tais que

$$\varphi_{xy}(f') = f = F(f') \quad \text{e} \quad \varphi_{yx}(g') = g = F(g').$$

Logo,

$$\varphi_{yy}(f' \circ g') = F(f' \circ g') = F(f') \circ F(g') = f \circ g = 1_{F(Y)} = F(1_Y) = \varphi_{yy}(1_Y).$$

Como φ_{yy} é injetora, temos que $f' \circ g' = 1_Y$. Uma argumentação análoga nos dá que $g' \circ f' = 1_X$, mostrando que de fato $X \cong Y$. ■

Lema 2.1.24. *Seja C uma categoria. Se $f : X \rightarrow Y$ for um morfismo e isomorfismo fixados $X \cong X'$ e $Y \cong Y'$, então fica determinado único morfismo $f' : X' \rightarrow Y'$ tal que o seguinte diagrama comuta*

$$\begin{array}{ccc} X & \xrightarrow{\cong} & X' \\ f \downarrow & \circlearrowleft & \downarrow f' \\ Y & \xrightarrow{\cong} & Y' \end{array}$$

Demonstração. A demonstração desse resultado é trivial. A existência de f' segue da composição adequada dos isomorfismos com o morfismo f e a unicidade segue da comutatividade do diagrama. ■

Teorema 2.1.25 (Caracterização de Equivalências de Categorias). *Um funtor que define uma equivalência de categorias é cheio, fiel e essencialmente sobrejetor sobre objetos. Assumindo o Axioma da Escolha vale a recíproca, isto é, qualquer funtor com essas propriedades define uma equivalência de categorias.*

Demonstração. (\Rightarrow) Sejam C e D categorias, $F : C \rightarrow D$, $G : D \rightarrow C$ funtores, η e ε isomorfismos naturais tal que $\eta : 1_C \cong G \circ F$ e $\varepsilon : F \circ G \cong 1_D$ definindo a equivalência das categorias C e D . Como ε é uma transformação natural, então para cada objeto $X \in Ob(D)$, devemos ter $\varepsilon_X : F \circ G(X) \cong 1_D(X) = X$, ou seja, para cada $X \in Ob(D)$ existe $G(X) \in Ob(C)$ tal que X e $F(G(X))$ são isomorfos, portanto F é essencialmente sobrejetor sobre objetos.

Nos resta mostrar que F é cheio e fiel. Consideramos dois morfismos $f, g : X \rightarrow X'$, com $X, X' \in Ob(C)$. Usando a naturalidade de η , se $F(f) = F(g)$, então f ou g fazem o seguinte diagrama comutar

$$\begin{array}{ccc} X & \xrightarrow{\eta_X} & G(F(X)) \\ f \text{ ou } g \downarrow & \circlearrowleft & \downarrow G \circ F(f) = G \circ F(g) \\ X' & \xrightarrow{\eta_{X'}} & G(F(X')) \end{array}$$

Segue do Lema 2.1.24 que $f = g$. Mostramos então que $\varphi_{xx'}$ é injetora, ou seja, F é fiel. Uma argumentação análoga pode ser feita para G e ε , mostrando que G é fiel também.

Agora, vamos verificar que para quaisquer $X, Y \in Ob(C)$ temos que φ_{xy} é sobrejetora. Dado um morfismo $k : F(X) \rightarrow F(Y)$, consideramos $G(k)$ e os isomorfismos η_X e η_Y . Como η é transformação natural, aplicando o Lema 2.1.24 sabemos que existe único $h : X \rightarrow Y$ que faz o diagrama comutar

$$\begin{array}{ccc} X & \xrightarrow{\eta_X} & G \circ F(X) \\ h \downarrow & \circlearrowleft & \downarrow G(k) \\ Y & \xrightarrow{\eta_Y} & G \circ F(Y) \end{array}$$

Tendo h garantida pelo Lema, destacamos que trocando $G(k)$ por $G \circ F(h)$ ainda temos o mesmo diagrama comutativo. Assim, obtemos da unicidade do Lema 2.1.24 que $G(k) = G \circ F(h)$. Mas argumentamos acima que G é fiel, logo temos que $F(h) = k$. Portanto, encontramos $h \in Hom_C(X, Y)$ tal que $F(h) = k$, isto é, mostramos que φ_{xy} é sobrejetora e portanto F é cheio.

(\Leftarrow) Por hipótese $F : C \rightarrow D$ é fiel, cheio e essencialmente sobrejetor. Da essencialidade sobrejetiva sobre elementos de F temos que para cada $X \in Ob(D)$ existe $Y \in Ob(C)$ tal que $X \cong F(Y)$. Aqui aplicamos o Axioma da Escolha: para cada $X \in Ob(D)$ escolhemos

$G(D) \in Ob(C)$ e um isomorfismo $\varepsilon_X : F \circ G(X) \cong X$. Então para cada morfismo $p : X \rightarrow X'$, segue do Lema 2.1.24 que temos o seguinte diagrama comutativo

$$\begin{array}{ccc} F \circ G(X) & \xrightarrow{\varepsilon_X} & X \\ \alpha \downarrow & \circlearrowleft & \downarrow p \\ F \circ G(X') & \xrightarrow{\varepsilon_{X'}} & X' \end{array}$$

onde α é único.

Como F é fiel e cheio, então para todo $A, B \in Ob(C)$ a aplicação φ_{ab} é bijetora. Logo para $G(X)$ e $G(X')$ existe único morfismo $G(X) \rightarrow G(X')$ que é levado em $F \circ G(X) \rightarrow F \circ G(X')$. Chamaremos de $G(p)$ tal morfismo.

Vamos verificar que $G : p \mapsto G(p)$ admite a propriedade funtorial, isto é, verificar que G assim definido é um funtor. De fato, perceba que $F \circ G(1_X)$ ou $F(1_{G(X)})$ fazem o seguinte diagrama comutar

$$\begin{array}{ccc} F \circ G(X) & \xrightarrow{\varepsilon_X} & X \\ F \circ G(1_X) \text{ ou } F(1_{G(X)}) \downarrow & \circlearrowleft & \downarrow 1_X \\ F \circ G(X) & \xrightarrow{\varepsilon_{X'}} & X \end{array}$$

Segue da unicidade do Lema 2.1.24 que $F \circ G(1_X) = F(1_{G(X)})$, mas por hipótese temos que F é fiel, logo temos que $G(1_X) = 1_{G(X)}$.

Além disso, dado $p' : X' \rightarrow X''$ temos que $F(G(p') \circ G(p))$ e $F \circ G(p' \circ p)$ fazem o seguinte diagrama comutar

$$\begin{array}{ccc} F \circ G(X') & \xrightarrow{\varepsilon_X} & X \\ F(G(p') \circ G(p)) \text{ ou } F \circ G(p' \circ p) \downarrow & \circlearrowleft & \downarrow p' \circ p \\ F \circ G(X'') & \xrightarrow{\varepsilon_{X'}} & X'' \end{array}$$

Segue novamente da unicidade do Lema 2.1.24 e da fidelidade de F que $G(p' \circ p) = G(p') \circ G(p)$ e G é de fato um funtor.

Só nos resta construir um isomorfismo natural $\eta : 1_C \cong G \circ F$. Fazemos o seguinte, definimos $\eta_X : X \rightarrow G \circ F(X)$ especificando quem é $F(\eta_X) : F(X) \rightarrow F \circ G \circ F(X)$. Definimos $F(\eta_X) := \varepsilon_{F(X)}^{-1}$, que é um isomorfismo, logo pelo Lema 2.1.23 temos que η_X é um isomorfismo. Para um morfismo qualquer $f : X \rightarrow X'$ consideramos o diagrama

$$\begin{array}{ccccc}
F(X) & \xrightarrow{F(\eta_X)} & F \circ G(F(X)) & \xrightarrow{\varepsilon_{F(X)}} & F(X) \\
\downarrow F(f) & & \downarrow F \circ G(F(f)) & & \downarrow F(f) \\
F(X') & \xrightarrow{F(\eta_{X'})} & F \circ G(F(X')) & \xrightarrow{\varepsilon_{F(X')}} & F(X')
\end{array}$$

Como $\varepsilon_{F(X)} : F \circ G(F(X)) \cong F(X)$ é isomorfismo, temos que o diagrama externo comuta. Além disso, construímos ε como sendo uma transformação natural, logo o diagrama da direita deve comutar por definição. Queremos mostrar que o diagrama da esquerda comuta também. De fato, como o diagrama da direita comuta e $\varepsilon_{F(X')}$ é isomorfismo, em particular um monomorfismo, usamos a comutatividade do diagrama externo

$$\varepsilon_{F(X')} \circ F(\eta_{X'}) \circ F(f) = F(f) \circ \varepsilon_{F(X)} \circ F(\eta_X) = \varepsilon_{F(X')} \circ (F \circ G(F(f))) \circ F(\eta_X).$$

Como $\varepsilon_{F(X')}$ é monomorfismo, então temos que

$$F(\eta_{X'}) \circ F(f) = F \circ G(F(f)) \circ F(\eta_X),$$

isto é, o diagrama da esquerda comuta. Como F é fiel, temos que

$$\eta_{X'} \circ f = G \circ F(f) \circ \eta_X,$$

ou seja, temos a comutatividade do seguinte diagrama

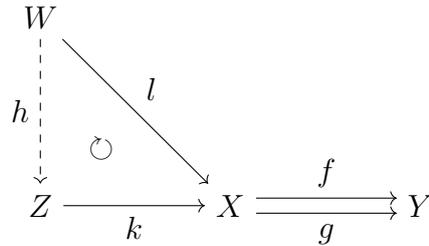
$$\begin{array}{ccc}
X & \xrightarrow{\eta_X} & G \circ F(X) \\
\downarrow f & \circlearrowleft & \downarrow G \circ F(f) \\
X' & \xrightarrow{\eta_{X'}} & G \circ F(X')
\end{array}$$

isto é, η é de fato uma transformação natural e como ε é isomorfismo natural, temos que η é um isomorfismo natural e o resultado segue. ■

Definição 2.1.26. *Sejam C uma categoria, $X, Y \in \text{Ob}(C)$, $f, g : X \rightarrow Y$ morfismos de C . Um **equalizador** de f e g é um par (Z, k) , onde $Z \in \text{Ob}(C)$ e k é um morfismo em $\text{Hom}_C(Z, X)$ tal que*

$$f \circ k = g \circ k.$$

Além disso, para qualquer outro par (W, l) , onde $W \in \text{Ob}(C)$ e $l : W \rightarrow X$ um morfismo em $\text{Hom}_C(W, X)$ tal que $f \circ l = g \circ l$, então existe único morfismo $h : W \rightarrow Z$ tal que $l = k \circ h$.



É direto ver que quando existe o equalizador de dois morfismos ele é único a menos de isomorfismo. Esse fato segue da propriedade universal apresentada acima.

Proposição 2.1.27. *Seja C uma categoria e $f, g : X \rightarrow Y$ dois morfismos que admitem um equalizador (Z, k) . Então o morfismo $k : Z \rightarrow X$ é um monomorfismo.*

Demonstração. Considere $W \in \text{Ob}(C)$ e dois morfismos $\alpha, \beta : W \rightarrow Z$ tais que $k \circ \alpha = k \circ \beta$. Queremos mostrar que $\alpha = \beta$. De fato, percebe-se que

$$f \circ k \circ \beta = f \circ k \circ \alpha = g \circ k \circ \alpha = g \circ k \circ \beta,$$

então o par $(W, k \circ \alpha) = (W, k \circ \beta)$ é um candidato a equalizador de f e g , segue da propriedade universal que existe um único morfismo entre W e Z fazendo o diagrama comutar. Portanto $\alpha = \beta$ e k é monomorfismo. ■

Exemplo 2.1.28. *Em **Sets** o equalizador de dois morfismos $f, g : X \rightarrow Y$ é dado pelo seguinte conjunto*

$$\{x \in X \mid f(x) = g(x)\}.$$

Definição 2.1.29. *O **coequalizador** é definido como a estrutura dual ao equalizador, isto é, revertemos a orientação das flechas na construção do equalizador.*

Definição 2.1.30. *Seja C uma categoria qualquer. Dizemos que $X \in \text{Ob}(C)$ é um **objeto inicial** se para todos os objetos $Y \in \text{Ob}(C)$ tivermos exatamente um morfismo $X \rightarrow Y$, isto é, a cardinalidade de $\text{Hom}_C(X, Y)$ é 1 para todo $Y \in \text{Ob}(C)$. Um **objeto terminal** é um objeto inicial na categoria C^{op} . Por fim, um **objeto nulo** é um objeto que é simultaneamente um objeto terminal e inicial.*

Exemplo 2.1.31. *Em **Sets**, o vazio é o único objeto inicial e um conjunto unitário é um objeto terminal. Em **Grps** um grupo com apenas um elemento é um objeto nulo.*

Definição 2.1.32. *Seja C uma categoria que possua um objeto nulo, que chamaremos de 0 . Definimos o **kernel** de um morfismo $f \in \text{Hom}_C(X, Y)$, com $X, Y \in \text{Ob}(C)$ quaisquer, como sendo o equalizador de f com o morfismo $0 : X \rightarrow Y$. Denotamos o kernel de f por $\mathbf{Ker}(f)$. Analogamente, definimos o **cokernel** de f como sendo o coequalizador de f e o morfismo 0 , denotado por $\mathbf{Coker}(f)$.*

Definição 2.1.33. Uma categoria C é dita ser **abeliana** se satisfaz as seguintes condições:

1. C possui um objeto nulo;
2. Todo par de objetos de C possui um produto e um coproduto;
3. Todo morfismo de C possui kernel e cokernel;
4. Todo monomorfismo de C é um kernel e todo epimorfismo de C é um cokernel.

Proposição 2.1.34. Sejam C uma categoria abeliana e A uma categoria pequena. Então $\mathbf{Fun}(A, C)$ é uma categoria abeliana.

Demonstração. A demonstração desse resultado pode ser encontrada em [2].

■

A seguinte proposição serve para expandir a discussão feita no Exemplo 2.1.12, nos dando condições para garantir quando um monomorfismo que é epimorfismo implica em isomorfismo.

Proposição 2.1.35. Seja C uma categoria abeliana. Então as seguintes condições são equivalentes:

1. f é um isomorfismo;
2. f é simultaneamente um monomorfismo e um epimorfismo.

Demonstração. (1) \Rightarrow (2) Direto.

(2) \Rightarrow (1) Perceba que em uma categoria qualquer D , se tivermos dois morfismos $\alpha, \beta : X \rightarrow Y$ e um equalizador de α e β , (Z, k) , tal que k é epimorfismo, então k é um isomorfismo. De fato, se k é epimorfismo, então segue que

$$\alpha \circ k = \beta \circ k \Rightarrow \alpha = \beta.$$

Portanto qualquer isomorfismo serve como candidato de equalizador de α e β , segue da propriedade universal que k é um isomorfismo.

Voltamos agora ao que gostaríamos de demonstrar. Como por hipótese f é um monomorfismo e C é uma categoria abeliana, então por definição existe um morfismo g tal que $f = \text{Ker}(g)$, isto é, f é o equalizador entre g e 0 . Além disso, temos também por hipótese que f é um epimorfismo e pelo o que argumentamos acima f é um isomorfismo.

■

O nome categoria abeliana vem da categoria **AbGrps** que consiste dos objetos sendo os grupos abelianos e os morfismos sendo os homomorfismos de grupos. As propriedades que pedimos que valham para uma categoria abeliana são algumas das propriedades que valem para **AbGrps**. Agora, apresentaremos um exemplo que nos será importante mais adiante no texto.

Exemplo 2.1.36. Considere a categoria \mathbf{Vec} . Obviamente o espaço vetorial nulo 0 faz o papel de um objeto nulo nessa categoria. Além disso, temos definido para todo par de objetos o produto direto e soma direta, que são respectivamente o produto categorial e coproduto. Aliás, em \mathbf{Vec} temos que o produto direto e a soma direta são isomorfos, para uma quantidade finita de objetos.

Para um morfismo $f : U \rightarrow V$ temos definido obviamente $\text{Ker}(f)$ e é fácil ver que $\text{Coker}(f)$ é o espaço vetorial quociente $V/f(U)$. Agora, perceba que todo monomorfismo é uma transformação linear injetora em \mathbf{Vec} , assim se $f : V \rightarrow U$ for um monomorfismo em \mathbf{Vec} , temos que $U \cong f(U)$ e U obviamente é o kernel da função projeção $\pi : V \rightarrow V/f(U) = \text{Coker}(f)$. Analogamente, se $f : U \rightarrow V$ for um epimorfismo, então f é sobrejetora e segue do Teorema do Isomorfismo que $U/\text{Ker}(f) \rightarrow V$ é um isomorfismo, isto é, V é o $\text{Coker}(i)$, onde $i : \text{Ker}(f) \hookrightarrow U$ é a inclusão. Portanto, concluímos que \mathbf{Vec} é de fato uma categoria abeliana.

Definição 2.1.37. Sejam C, D duas categorias abelianas e $F : C \rightarrow D$ um funtor. Se F preservar coprodutos, dizemos que F é um **funtor aditivo**.

2.2 Categorificação da Homologia Persistente

Nesse ponto do texto, temos o básico da Teoria de Categorias para tornar o estudo de TDA e de Merge Trees viável de um ponto de vista categorial, tornando a teoria mais robusta e geral. Começamos revisitando alguns conceitos para contextualizar os avanços que faremos a partir daqui.

Destacamos que o conjunto dos números reais \mathbb{R} munido da relação de ordem usual \leq é um poset, que como vimos na seção anterior, pode ser tratado como uma categoria (\mathbb{R}, \leq) , onde os objetos são os números reais e os morfismos são determinados por $a \leq b$.

Definição 2.2.1. Um **espaço de persistência**, ou ainda **espaço persistente**, é um funtor $F : (\mathbb{R}, \leq) \rightarrow \mathbf{Top}$. Para cada $s \in \mathbb{R}$ o funtor F associa um espaço topológico $F(s)$ e para um par de números reais $s \leq t$ temos uma função contínua $F(s \leq t) : F(s) \rightarrow F(t)$ que relaciona os espaços topológicos $F(s)$ e $F(t)$. Caso todas as funções $F(s \leq t)$ forem injetoras, chamamos o espaço de persistência de **filtração**.

Exemplo 2.2.2. Seja X um espaço topológico, $f : X \rightarrow \mathbb{R}$ uma função contínua a **filtração sublevel-set** é o espaço persistente

$$F(s) := f^{-1}(-\infty, s] = \{x \in X \mid f(x) \leq s\}.$$

Perceba que isso faz sentido, uma vez que $a \leq b$ implica que $f^{-1}(-\infty, a] \subseteq f^{-1}(-\infty, b]$. Logo as inclusões $F(s) \subseteq F(t)$ são contínuas e injetoras e assim definem uma aplicação $F(s \leq t)$ para todo $s \leq t$.

Outro exemplo de filtração que nos será útil mais adiante é a **filtração offset** determinada da seguinte maneira: Seja Z um subconjunto de um espaço métrico X , definimos a função offset $f_Z : X \rightarrow \mathbb{R}$, $f_Z(x) := \inf_{z \in Z} d(x, z)$, isto é, a distância de x ao conjunto Z . Definimos então a filtração offset como sendo

$$F_Z(s) := \left\{ x \in X \mid \inf_{z \in Z} d(x, z) \leq s \right\},$$

que se trata da filtração de sublevel-set da função f_Z .

Podemos também revisitar a teoria abordada no Capítulo 1 de módulos de persistência, utilizando os conceitos de categorias apresentados na seção anterior. A indexação da família de espaços vetoriais presente na definição de módulos de persistência pode ser vista como um functor $F : (\mathbb{R}, \leq) \rightarrow \mathbf{Vec}_{\mathbb{K}}$, uma vez que a imposição da lei de composição dos morfismos da Definição 1.2.1 é exatamente a propriedade functorial descrita em outras palavras.

Definição 2.2.3. *Um functor $S : (\mathbb{R}, \leq) \rightarrow \mathbf{Sets}$ é chamado de **conjunto persistente**.*

Como \mathbf{Top} é uma categoria concreta, dado um espaço de persistência $F : (\mathbb{R}, \leq) \rightarrow \mathbf{Top}$ sempre podemos obter um conjunto a partir do mesmo, basta aplicar o functor $\pi_0 : \mathbf{Top} \rightarrow \mathbf{Sets}$

$$\pi_0 \circ F : (\mathbb{R}, \leq) \rightarrow \mathbf{Sets}, \quad s \mapsto \pi_0(F(s)).$$

Recorrer a este functor $\pi_0 \circ F$ é a ideia chave para a aplicação das Merge Trees à teoria de persistência homológica. O functor $\pi_0 \circ F$ nos dá os conjuntos subjacentes das componentes conexas do espaço $F(s)$, as Merge Trees nos mostram como elas se relacionam entre si e como elas evoluem ao decorrer do tempo.

Destacamos que, do Axioma da Aditividade da Homologia, para $X \in \mathbf{Ob}(\mathbf{Top})$ um espaço topológico e X_i , $i \in I$ suas componentes conexas, temos que

$$X \approx \bigsqcup_{i \in I} X_i$$

implica em

$$H_n(X) \cong \bigoplus_{i \in I} H_n(X_i).$$

Isso é um fato extremamente importante, que nos servirá para estudar a evolução do grupo de homologia H_0 de um conjunto de dados. Veremos essas implicações no próximo capítulo, mas para isso, precisamos desenvolver primeiramente ferramentas categoriais que permitirão tal estudo.

Definição 2.2.4. *Seja C uma categoria. Definimos a **categoria dos objetos discretamente parametrizados** em C , denotada por \mathbf{pC} , tendo como objetos funtores*

$I : S \rightarrow C$, onde S é uma categoria discreta e morfismos sendo pares (m, α) , onde se $I, J \in \text{Ob}(pC)$, $I : S \rightarrow C$ e $J : T \rightarrow C$, temos a função entre conjuntos $m : S \rightarrow T$ e uma transformação natural² $\alpha : I \Rightarrow m^*J = J \circ m$. Chamamos um objeto $I : S \rightarrow C$ de **objeto S -parametrizado**.

Definição 2.2.5. Seja C uma categoria. Dois objetos parametrizados $I : S \rightarrow C$ e $J : T \rightarrow C$ são **isomorfos** se existirem $m : S \rightarrow T$, $n : T \rightarrow S$ e transformações naturais $\alpha : I \Rightarrow J \circ m$ e $\beta : J \Rightarrow I \circ n$ satisfazendo

$$m^*\beta \circ \alpha = 1_I; \quad n^*\alpha \circ \beta = 1_J.$$

Em particular, temos que $n \circ m = 1_S$ e $m \circ n = 1_T$.

Vale a pena destacar que se a categoria C admite coprodutos, então temos o seguinte diagrama envolvendo pC

$$\begin{array}{ccc} & pC & \\ \text{dom} \swarrow & & \searrow \text{cop} \\ \mathbf{Sets} & & C \end{array}$$

onde dom é o funtor que leva qualquer S -objeto parametrizado $I : S \rightarrow C$ no conjunto S e cop é o funtor que leva I em seu coproduto.

Definição 2.2.6. Vamos denotar por \mathbf{Top}_c a categoria que consiste dos objetos sendo os espaços topológicos conexos e localmente conexo e os morfismos sendo as funções contínuas. Além disso, considere \mathbf{Top}_{lc} a categoria com objetos sendo os espaços topológicos localmente conexos e morfismos sendo as funções contínuas.

Proposição 2.2.7. O funtor cop induz uma equivalência de categorias entre \mathbf{pTop}_c e \mathbf{Top}_{lc}

$$\begin{aligned} \text{cop} : \mathbf{pTop}_c &\rightarrow \mathbf{Top}_{lc} \\ I &\mapsto \bigsqcup_{s \in S} I(s), \end{aligned}$$

com I sendo um objeto S -parametrizado.

Demonstração. Pelo Teorema 2.1.25 basta mostrarmos que o funtor $\text{cop} : \mathbf{pTop}_c \rightarrow \mathbf{Top}_{lc}$ é cheio, fiel e essencialmente sobrejetor sobre objetos.

Perceba que todo espaço $X \in \text{Ob}(\mathbf{Top}_{lc})$ pode ser escrito como uma união disjunta de suas componentes conexas, que é um coproduto em \mathbf{Top} . Ou seja, $X \in \text{Ob}(\mathbf{Top}_{lc})$

²Utilizamos esta notação de m^*J para representar o pullback de J através de m , isto é, neste caso falamos precisamente da pré-composição.

é naturalmente homeomorfo ao coproduto de suas componentes conexas, o que nos dá a essencialidade sobrejetiva.

Agora vamos mostrar que para quaisquer $I : S \rightarrow \mathbf{Top}_c$ e $J : T \rightarrow \mathbf{Top}_c$, temos que a função

$$\varphi_{IJ} : Hom_{\mathbf{pTop}_c}(I, J) \rightarrow Hom_{\mathbf{Top}_{lc}} \left(\bigsqcup_{s \in S} I(s), \bigsqcup_{t \in T} J(t) \right)$$

é bijetora.

Para cada morfismo $f : \bigsqcup_{s \in S} I(s) \rightarrow \bigsqcup_{t \in T} J(t)$, como cada objeto em \mathbf{pTop}_c é um espaço conexo, então para cada $s \in S$ associamos único $t \in T$ tal que $f(I(s)) \subseteq J(t)$, uma vez que f é contínua e imagem de espaço conexo por função contínua é conexo. Desta forma, construímos um par (m, α) , onde $m : S \rightarrow T$ é uma função entre os conjuntos S e T dada por essa correspondência entre os índices e α fica definida em cada $s \in S$ da seguinte maneira $\alpha_s := f|_{I(s)}$. Assim temos o seguinte diagrama comutativo

$$\begin{array}{ccc} I(s) & \xrightarrow{\alpha_s} & J \circ m(s) \\ I(\psi) \downarrow & \circlearrowleft & \downarrow J \circ m(\psi) \\ I(s') & \xrightarrow{\alpha_{s'}} & J \circ m(s') \end{array}$$

onde $\psi : S \rightarrow S$ é uma função que atribui s a s' . Isso nos mostra que $\alpha : I \Rightarrow J \circ m$ é uma transformação natural e o par (m, α) é um morfismo de \mathbf{pTop}_c . Ou seja, mostramos que φ_{IJ} é sobrejetora.

Mostremos a injetividade de φ_{IJ} . Sejam $(m, \alpha), (n, \beta)$ morfismos de \mathbf{pTop}_c tais que $\varphi_{IJ}(m, \alpha) = \varphi_{IJ}(n, \beta)$. Então eles induzem a mesma função sobre a união disjunta das componentes conexas, desta forma, devemos ter $m = n$, pois caso contrário se $m(s) = t$ e $n(s) = t', t \neq t'$ teríamos $I(s)$ sendo mandado em $J(t)$ pela $\varphi_{IJ}(m, \alpha)$ e $I(s)$ mandado em $J(t')$ pela $\varphi_{IJ}(n, \beta)$, o que contradiz a suposição que $\varphi_{IJ}(m, \alpha) = \varphi_{IJ}(n, \beta)$. Segue desse fato e da construção das transformações naturais que $m^* \alpha = n^* \beta$ e o resultado segue. ■

A Proposição acima nos induz a definir um funtor que sirva de inverso ao funtor *cop* na equivalência de categorias entre as categorias \mathbf{pTop}_c e \mathbf{Top}_{lc}

Definição 2.2.8. *Definimos o funtor parametrização por componentes*

$$pbc : \mathbf{Top}_{lc} \rightarrow \mathbf{pTop}_c$$

que leva cada $X \in Ob(\mathbf{Top}_{lc})$ em um funtor $I : \pi_0(X) \rightarrow \mathbf{Top}_c$, que, por sua vez, para cada rótulo $i \in \pi_0(X)$ leva esse conjunto em seu espaço topológico X_i correspondente, com

a topologia de subespaço. Uma função contínua entre espaços topológicos $f : X \rightarrow Y$ é levada no morfismo (m, α) , onde $m := \pi_0(f) : \pi_0(X) \rightarrow \pi_0(Y)$ é uma função de conjuntos que registra qual componente de X é levada em sua componente correspondente de Y . Além disso, α é a transformação natural que é definida por cada componente $\alpha_i := f|_{X_i}$, tal que o seguinte diagrama comuta

$$\begin{array}{ccc}
 I(X_i) & \xrightarrow{\alpha_i} & J \circ \pi_0(f)(X_i) \\
 I(\psi) \downarrow & \circlearrowleft & \downarrow J \circ \pi_0(f)(\psi) \\
 I(X_j) & \xrightarrow{\alpha_j} & J \circ \pi_0(f)(X_j)
 \end{array}$$

onde $\psi : \pi_0(X) \rightarrow \pi_0(X)$ que leva X_i em X_j .

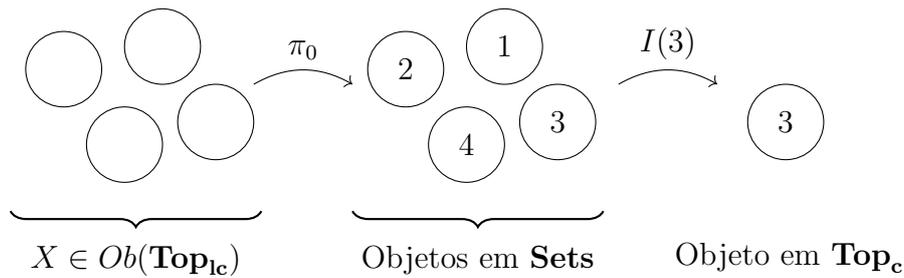


Figura 2.1: Figura ilustrativa de como podemos aplicar o funtor pb_c em um objeto de \mathbf{Top}_{1c} e construir um objeto parametrizado. Obviamente, na figura acima, estamos nos restringindo ao rótulo 3, mas esse processo se realiza para todos os rótulos de $\pi_0(X)$.

Usando como auxílio visual a Figura 2.1, fica evidente que os funtores cop e pb_c são um o inverso do outro, uma vez que o funtor pb_c apenas pega um objeto em \mathbf{Top}_{1c} e o decompõe em suas componentes conexas. Em outras palavras, pb_c faz justamente o processo reverso de cop .

Lema 2.2.9. *Todo espaço persistente $F : (\mathbb{R}, \leq) \rightarrow \mathbf{Top}_{1c}$ tem um espaço parametrizado persistente associado*

$$\tilde{F} := pb_c \circ F : (\mathbb{R}, \leq) \rightarrow \mathbf{pTop}_c.$$

Além disso, o funtor \tilde{F} nos dá o seguinte diagrama, que comuta a menos de isomorfismo natural.

$$\begin{array}{ccc}
 & (\mathbb{R}, \leq) & \\
 & \swarrow \tilde{F} \quad \searrow F & \\
 \pi_0 \circ \tilde{F} & \mathbf{pTop}_c & \\
 \swarrow \text{dom} \quad \searrow \text{cop} & & \\
 \mathbf{Sets} & \xleftarrow{\pi_0} & \mathbf{Top}_{lc}
 \end{array}$$

Demonstração. Sabemos que $\text{cop} \circ \text{pb} \cong 1_{\mathbf{Top}_{lc}}$, então temos

$$\text{cop} \circ \text{pb} \circ F \cong F \Leftrightarrow \text{cop} \circ \tilde{F} \cong F.$$

Isto significa que para cada $s \in \mathbb{R}$ os espaços $\text{cop} \circ \tilde{F}(s)$ e $F(s)$ são homeomorfos e portanto o conjunto das componentes conexas desses espaços são isomorfos, uma vez que $\pi_0 : \mathbf{Top} \rightarrow \mathbf{Sets}$ é um functor e, pela Proposição 2.1.13, π_0 preserva isomorfismos. Portanto os conjuntos persistentes $\pi_0 \circ F$ e $\text{dom} \circ \tilde{F}$ são naturalmente isomorfos. ■

Vamos focar agora em enunciar e provar uma versão um pouco mais abrangente do Teorema de Estabilidade enunciado no Capítulo 1. A Teoria de Categorias vai tornar nosso trabalho de provar tal teorema mais fácil.

Definição 2.2.10. Considerando o poset (\mathbb{R}, \leq) , para $b \geq 0$ definimos o functor

$$\begin{aligned}
 T_b : (\mathbb{R}, \leq) &\rightarrow (\mathbb{R}, \leq) \\
 a &\mapsto a + b
 \end{aligned}$$

e a transformação natural $\eta_b : 1_{(\mathbb{R}, \leq)} \Rightarrow T_b$ dada por $\eta_b(a) : a \leq a + b$. Destacamos que $T_b \circ T_c = T_{b+c}$ e $\eta_b \circ \eta_c = \eta_{b+c}$.

Definição 2.2.11. Sejam D uma categoria, $\varepsilon \geq 0$ e $F, G : (\mathbb{R}, \leq) \rightarrow D$ dois funtores. Um ε -interleaving de F e G consiste de transformações naturais $\varphi : F \Rightarrow G \circ T_\varepsilon$ e $\psi : G \Rightarrow F \circ T_\varepsilon$

$$\begin{array}{ccccc}
 (\mathbb{R}, \leq) & \xrightarrow{T_\varepsilon} & (\mathbb{R}, \leq) & \xrightarrow{T_\varepsilon} & (\mathbb{R}, \leq) \\
 F \downarrow & \varphi \Rightarrow & \downarrow G & \psi \Rightarrow & \downarrow F \\
 D & \xrightarrow{1_D} & D & \xrightarrow{1_D} & D
 \end{array}$$

tais que

$$(\varphi(T_\varepsilon))\psi = G \circ \eta_{2\varepsilon} \quad e \quad (\psi(T_\varepsilon))\varphi = F \circ \eta_{2\varepsilon}. \quad (2.1)$$

Se (F, G, φ, ψ) for um ε -interleaving, dizemos que F e G são ε -interleaved. A existência das transformações naturais φ e ψ nos dá os seguintes diagramas comutativos

$$\begin{array}{ccc} F(a) & \longrightarrow & F(b) \\ \varphi(a) \searrow & \circlearrowleft & \searrow \varphi(b) \\ & & G(a + \varepsilon) \longrightarrow G(b + \varepsilon) \end{array} \quad \begin{array}{ccc} F(a + \varepsilon) & \longrightarrow & F(b + \varepsilon) \\ \psi(a) \nearrow & \circlearrowleft & \nearrow \psi(b) \\ & & G(a) \longrightarrow G(b) \end{array}$$

De (2.1) segue os seguintes diagramas comutativos

$$\begin{array}{ccc} F(a) & \longrightarrow & F(a + 2\varepsilon) \\ \varphi(a) \searrow & \circlearrowleft & \nearrow \psi(a + \varepsilon) \\ & & G(a + \varepsilon) \end{array} \quad \begin{array}{ccc} & & F(a + \varepsilon) \\ \psi(a) \nearrow & \circlearrowleft & \searrow \varphi(a + \varepsilon) \\ & & G(a) \longrightarrow G(a + 2\varepsilon) \end{array}$$

Definição 2.2.12. Considere dois funtores $F, G : (\mathbb{R}, \leq) \rightarrow D$. Definimos a **distância interleaving**

$$d_I(F, G) := \inf \{ \varepsilon \geq 0 \mid F \text{ e } G \text{ são } \varepsilon\text{-interleaved} \},$$

se F e G forem ε -interleaved e $d_I(F, G) := \infty$ se F e G não forem ε -interleaved para todo $\varepsilon \geq 0$.

Fica claro que se F e G forem 0-interleaved temos que $F \cong G$ uma vez que temos o seguinte diagrama comutativo das transformações naturais

$$\begin{array}{ccc} F(a) & \longrightarrow & F(b) \\ \psi(a) \updownarrow & \varphi(a) \circlearrowleft \varphi(b) & \updownarrow \psi(b) \\ G(a) & \longrightarrow & G(b) \end{array}$$

Entretanto $d_I(F, G) = 0$ implica que F e G são ε -interleaved para todo $\varepsilon > 0$, não implicando que $F \cong G$.

Vamos verificar que d_I é de fato uma pseudo-métrica estendida, para isso precisamos do próximo resultado.

Lema 2.2.13. Sejam D uma categoria e $F, G : (\mathbb{R}, \leq) \rightarrow D$ funtores ε -interleaved, então F e G são ε' -interleaved para $\varepsilon' \geq \varepsilon$.

Demonstração. Se F e G são ε -interleaved, então existem transformações naturais $\varphi : F \Rightarrow G \circ T_\varepsilon$ e $\psi : G \Rightarrow F \circ T_\varepsilon$ satisfazendo as condições presentes na Definição 2.2.11. Por hipótese, temos $\varepsilon' \geq \varepsilon$, então consideramos $\bar{\varepsilon} = \varepsilon' - \varepsilon \geq 0$. Temos a transformação natural $\eta_{\bar{\varepsilon}} : 1_{(\mathbb{R}, \leq)} \Rightarrow T_{\bar{\varepsilon}}$, e aplicando $T_{\bar{\varepsilon}}$ obtemos

$$\eta_{\bar{\varepsilon}}(T_\varepsilon) : T_\varepsilon \Rightarrow T_{\bar{\varepsilon}} \circ T_\varepsilon = T_{\bar{\varepsilon} + \varepsilon} = T_{\varepsilon'}.$$

Compondo com o funtor G temos

$$G \circ \eta_{\bar{\varepsilon}}(T_{\varepsilon}) : G \circ T_{\varepsilon} \Rightarrow G \circ T_{\varepsilon'}.$$

Definimos $\hat{\varphi} := (G \circ \eta_{\bar{\varepsilon}}(T_{\varepsilon})) \circ \varphi$, temos assim

$$\hat{\varphi}(a) : F(a) \xrightarrow{\varphi_a} G(a + \varepsilon) \xrightarrow{G \circ \eta_{\bar{\varepsilon}}(T_{\varepsilon})} G(a + \varepsilon').$$

Analogamente, podemos definir $\hat{\psi} := (F \circ \eta_{\bar{\varepsilon}}(T_{\varepsilon})) \circ \psi$. Precisamos mostrar que $(\hat{\psi}(T_{\varepsilon'})) \circ \hat{\varphi} = F \circ \eta_{2\varepsilon'}$, isso segue do seguinte diagrama:

$$\begin{array}{ccccc}
 F(a) & \xrightarrow{F \circ \eta_{2\varepsilon}(a)} & F(a + 2\varepsilon) & \xrightarrow{F \circ \eta_{\bar{\varepsilon}}(T_{2\varepsilon}(a))} & F(a + \varepsilon' + \varepsilon) & \xrightarrow{F \circ \eta_{\bar{\varepsilon}}(T_{\varepsilon+\varepsilon'}(a))} & F(a + 2\varepsilon') \\
 \searrow \varphi(a) & & \nearrow \psi(a + \varepsilon) & & \nearrow \psi(a + \varepsilon') & & \\
 & & G(a + \varepsilon) & \xrightarrow{G \circ \eta_{\bar{\varepsilon}}(T_{\varepsilon}(a))} & G(a + \varepsilon') & &
 \end{array}$$

O triângulo à esquerda no diagrama superior comuta pela Definição 2.2.11. Já o paralelogramo central no digrama é comutativo pois ψ é transformação natural. Perceba que o caminho destacado em negrito no diagrama representa $(\hat{\psi}(T_{\varepsilon'})) \circ \hat{\varphi} = F \circ \eta_{2\varepsilon'}$ e além disso como $\eta_{\bar{\varepsilon}} : 1_{(\mathbb{R}, \leq)} \Rightarrow T_{\bar{\varepsilon}}$ temos que

$$\eta_{\bar{\varepsilon}} T_{\varepsilon+\varepsilon'} : T_{\varepsilon+\varepsilon'} \Rightarrow T_{2\varepsilon},$$

desta forma obtemos

$$(F \circ \eta_{\bar{\varepsilon}})(T_{\varepsilon'}) = F \circ \eta_{2\varepsilon'}.$$

De maneira análoga, podemos verificar que $(\hat{\varphi} T_{\varepsilon'}) \hat{\psi} = G \circ \eta_{2\varepsilon'}$. ■

Teorema 2.2.14. *A função d_I da Definição 2.2.12 é uma pseudo-métrica estendida em qualquer subconjunto da classe de funtores de (\mathbb{R}, \leq) em D .*

Demonstração. Consideramos os funtores $F, G, H : (\mathbb{R}, \leq) \rightarrow D$. Obviamente temos que $d_I(F, F) = 0$, pois basta usar como transformações naturais $\varphi = \psi = 1$. Além disso, segue que $d_I(F, G) = d_I(G, F)$ da simetria da própria definição de funtores ε -interleaved. Só nos resta mostrar que vale a Desigualdade Triangular.

De fato, se considerarmos $a := d_I(F, G)$ e $b := d_I(G, H)$ e $\delta > 0$. Como $a + \delta > a$ e $b + \delta > b$, então pelo Lema 2.2.13 temos que F e G são $(a + \delta)$ -interleaved e G e H $(b + \delta)$ -interleaved. Portanto, por definição existem transformações naturais

$$\varphi' : F \Rightarrow G \circ T_{a+\delta}, \quad \psi' : G \rightarrow F \circ T_{a+\delta}, \quad \varphi'' : G \rightarrow H \circ T_{b+\delta} \text{ e } \psi'' : H \rightarrow G \rightarrow T_{b+\delta}.$$

Considere

$$\varphi := (\varphi'' \circ T_{a+\delta})\varphi' : F \Rightarrow H \circ T_{a+b+2\delta} \quad \text{e} \quad \psi := (\psi' \circ T_{b+\delta})\psi'' : H \rightarrow F \circ T_{a+b+2\delta}.$$

Essas composições valem por causa dos seguintes diagramas

$$\begin{array}{ccccc} (\mathbb{R}, \leq) & \xrightarrow{T_{a+\delta}} & (\mathbb{R}, \leq) & \xrightarrow{T_{b+\delta}} & (\mathbb{R}, \leq) \\ F \downarrow & \varphi' \Downarrow & \downarrow G & \varphi'' \Downarrow & \downarrow H \\ D & \xrightarrow{1_D} & D & \xrightarrow{1_D} & D \end{array} \quad \begin{array}{ccccc} (\mathbb{R}, \leq) & \xrightarrow{T_{b+\delta}} & (\mathbb{R}, \leq) & \xrightarrow{T_{a+\delta}} & (\mathbb{R}, \leq) \\ H \downarrow & \psi'' \Downarrow & \downarrow G & \psi' \Downarrow & \downarrow F \\ D & \xrightarrow{1_D} & D & \xrightarrow{1_D} & D \end{array}$$

Juntando esse dois diagramas temos

$$\begin{array}{cccccccc} (\mathbb{R}, \leq) & \xrightarrow{T_{a+\delta}} & (\mathbb{R}, \leq) & \xrightarrow{T_{b+\delta}} & (\mathbb{R}, \leq) & \xrightarrow{T_{b+\delta}} & (\mathbb{R}, \leq) & \xrightarrow{T_{a+\delta}} & (\mathbb{R}, \leq) \\ F \downarrow & \varphi' \Downarrow & \downarrow G & \varphi'' \Downarrow & \downarrow H & \psi'' \Downarrow & \downarrow G & \psi' \Downarrow & \downarrow F \\ D & \xrightarrow{1_D} & D & \xrightarrow{1_D} & D & \xrightarrow{1_D} & D & \xrightarrow{1_D} & D \end{array}$$

Assim, concluímos que

$$(\psi \circ T_{a+b+2\delta}) \circ \varphi = F \circ \eta_{2(a+b+2\delta)} \quad \text{e} \quad (\varphi \circ T_{a+b+2\delta}) \circ \psi = H \circ \eta_{2(a+b+2\delta)}.$$

Portanto F e H são $(a + b + 2\delta)$ -interleaved para todo $\delta > 0$, logo obtemos que

$$d_I(F, H) \leq a + b. \quad \blacksquare$$

Proposição 2.2.15. *Sejam D e E categorias, $F, G : (\mathbb{R}, \leq) \rightarrow D$ e $H : D \rightarrow E$ funtores. Se F e G forem ε -interleaved então HF e HG também o são. Mais ainda, temos que*

$$d_I(HF, HG) \leq d_I(F, G).$$

Demonstração. Se F e G são ε -interleaved, então existem transformações naturais

$$\varphi : F \Rightarrow G \circ T_\varepsilon \quad \text{e} \quad \psi : G \Rightarrow F \circ T_\varepsilon.$$

Segue da functorialidade que

$$H\varphi : HF \Rightarrow H(G \circ T_\varepsilon) \quad \text{e} \quad H\psi : HG \Rightarrow H(F \circ T_\varepsilon)$$

$$\begin{array}{ccccc}
(\mathbb{R}, \leq) & \xrightarrow{T_\varepsilon} & (\mathbb{R}, \leq) & \xrightarrow{T_\varepsilon} & (\mathbb{R}, \leq) \\
F \downarrow & \cong & \downarrow G & \cong & \downarrow F \\
D & \xrightarrow{1_D} & D & \xrightarrow{1_D} & D \\
H \downarrow & & \downarrow H & & \downarrow H \\
E & \xrightarrow{1_E} & E & \xrightarrow{1_E} & E
\end{array}$$

Como já tínhamos $(\psi(T_\varepsilon))\varphi = F \circ \eta_{2\varepsilon}$ e $(\varphi(T_\varepsilon))\psi = G \circ \eta_{2\varepsilon}$, segue do diagrama acima e da functorialidade que

$$(H\psi(T_\varepsilon)) \circ (H\varphi) = HF \circ \eta_{2\varepsilon} \text{ e } (H\varphi(T_\varepsilon)) \circ (H\psi)HG \circ \eta_{2\varepsilon}.$$

Portanto HF e HG são ε -interleaved. ■

O próximo resultado é importante na teoria de TDA, pois relaciona a distância bottleneck apresentada no Capítulo 1 e a distância interleaving apresentada neste capítulo.

Teorema 2.2.16 (Isometria). *Sejam M e N módulos de dimensão pontualmente finita. Para qualquer $\varepsilon \geq 0$, M e N são ε -interleaved se, e somente se, existe um ε -pareamento entre seus respectivos barcodes. Em particular*

$$d_I(M, N) = d_B(\mathcal{B}(M), \mathcal{B}(N))$$

Demonstração. A demonstração desse resultado é omitida aqui, porém pode ser encontrada em [24]. ■

Usualmente, nos referimos a afirmação direta do Teorema da Isometria como sendo o Teorema da Estabilidade Algébrica.

Teorema 2.2.17 (Estabilidade Algébrica). *Sejam M e N módulos de dimensão pontualmente finita e $\varepsilon \geq 0$, um ε -interleaving entre esses módulos, induz um ε -pareamento entre seus respectivos barcodes. Em particular,*

$$d_B(\mathcal{B}(M), \mathcal{B}(N)) \leq d_I(M, N).$$

Com essa relação entre as duas métricas, temos as ferramentas necessárias para enunciar e demonstrar o resultado que generaliza o Teorema 1.2.13.

Teorema 2.2.18 (Estabilidade). *Sejam D uma categoria, $X \in \text{Ob}(\mathbf{Top})$, $f, g : X \rightarrow \mathbb{R}$, $F, G : (\mathbb{R}, \leq) \rightarrow \mathbf{Top}$ funtores definidos da seguinte forma $F(a) := f^{-1}(-\infty, a]$*

para $a \in \mathbb{R}$ e $F(a \leq b)$ é dada pela inclusão. G é definida de maneira análoga. Seja $H : \mathbf{Top} \rightarrow D$ um funtor qualquer. Temos então

$$d_I(HF, HG) \leq \|f - g\|_\infty.$$

Demonstração. Considere $\varepsilon := \|f - g\|_\infty$. Perceba que

$$F(a) = f^{-1}(-\infty, a] \subseteq g^{-1}(-\infty, a + \varepsilon] = G(a + \varepsilon).$$

O mesmo vale para $G(a) \subseteq F(a + \varepsilon)$. Assim temos que F e G são ε -interleaved. De fato, como $F(a) \subseteq G(a + \varepsilon)$ existe transformação natural $\varphi : F \Rightarrow G \circ T_\varepsilon$ e $G(a) \subseteq F(a + \varepsilon)$ existe transformação natural $\psi : G \Rightarrow F \circ T_\varepsilon$, ambas induzidas pela inclusão. Além disso, temos que $F(a + \varepsilon) \subseteq G(a + 2\varepsilon)$ e $G(a + \varepsilon) \subseteq F(a + 2\varepsilon)$ temos o diagrama

$$\begin{array}{ccc} F(a) & \xrightarrow{F \circ \eta_{2\varepsilon}} & F(a + 2\varepsilon) \\ \varphi(a) \searrow & \circlearrowleft & \nearrow \psi(a + \varepsilon) \\ & G(a + \varepsilon) & \end{array} \qquad \begin{array}{ccc} & & F(a + \varepsilon) \\ \psi(a) \nearrow & \circlearrowleft & \searrow \varphi(a + \varepsilon) \\ G(a) & \xrightarrow{G \circ \eta_{2\varepsilon}} & G(a + 2\varepsilon) \end{array}$$

Isto é, $(\varphi(T_\varepsilon))\psi = G \circ \eta_{2\varepsilon}$ e $(\psi(T_\varepsilon))\varphi = F \circ \eta_{2\varepsilon}$. Segue da Proposição 2.2.15 temos que HF e HG são ε -interleaved e

$$d_I(HF, HG) \leq \|f - g\|_\infty.$$

■

Capítulo 3

Decorated Merge Trees

Neste capítulo iremos seguir os resultados apresentados em [8] e [26]. Já apresentamos a teoria básica de homologia persistente no Capítulo 1, agora apresentaremos outros métodos de rastreamento da evolução das componentes conexas ao decorrer do tempo. Começamos apresentando o conceito de Merge Tree e uma topologia usual para que possamos tratar delas de maneira adequada.

Apenas o rastreamento da evolução do número de Betti β_0 , não é tão relevante quanto gostaríamos. Portanto, depois do desenvolvimento da teoria de Merge Trees, iremos trabalhar em uma decoração destas árvores, acrescentando informações sobre os outros grupos de homologia, culminando em um resultado que nos garante que essa ferramenta é mais sensível que a teoria de Homologia Persistente usual.

Além disso, ao decorrer deste capítulo, apresentaremos alguns exemplos computacionais artificiais, que nos ajudam a ver as possibilidades de aplicações desta nova ferramenta. Tais exemplos foram gerados utilizando o código apresentado por Tom Needham em seu GitHub [29].

3.1 Merge Trees

Definição 3.1.1. *Dado um espaço topológico X e uma função contínua $f : X \rightarrow \mathbb{R}$, definimos uma relação de equivalência \sim em X da seguinte maneira: dado $y \in \mathbb{R}$, consideramos $f^{-1}(y)$, dois pontos em $p, q \in X$ se relacionam $p \sim q$ se p e q pertencem a mesma componente conexa de $f^{-1}(y)$. Chamamos o espaço quociente X/\sim de **Grafo de Reeb**.*

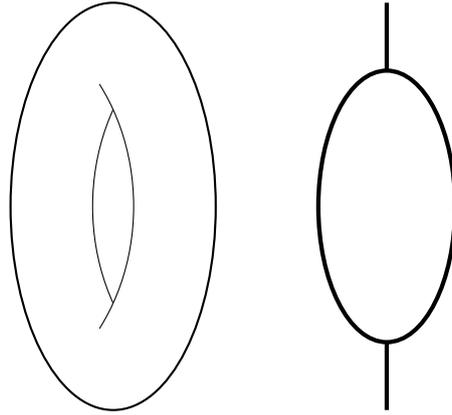


Figura 3.1: À esquerda temos um Toro e à direita temos seu respectivo Grafo de Reeb com respeito a função altura.

Definição 3.1.2. *Seja X um espaço topológico e $f : X \rightarrow \mathbb{R}$ uma função contínua. O epígrafo de f é o conjunto*

$$E_f := \{(x, r) \in X \times \mathbb{R} \mid f(x) \leq r\}.$$

A *merge tree clássica* M_f é o grafo de Reeb da projeção

$$\begin{aligned} \pi_f : E_f &\rightarrow \mathbb{R} \\ (x, r) &\mapsto r, \end{aligned}$$

isto é, $M_f := E_f / \sim$, onde \sim é a relação de equivalência definida da Definição 3.1.1.

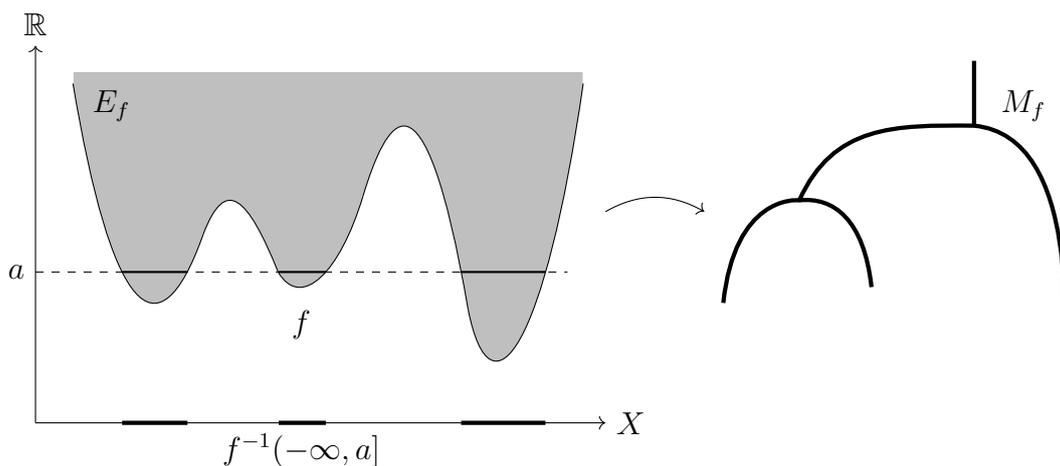
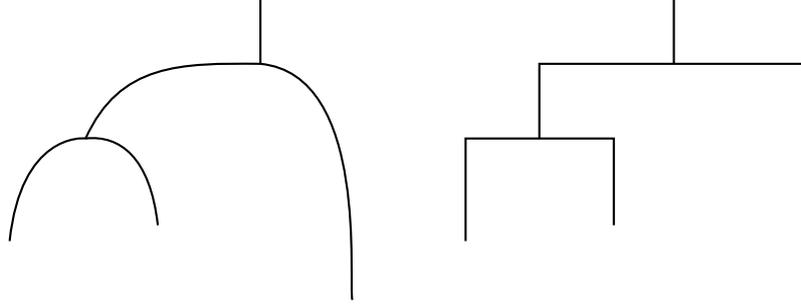


Figura 3.2: Esta figura serve de auxílio visual para a Definição 3.1.2. A região cinza acima do gráfico de função f é o epígrafo de f . A parte grifada no eixo X representa a $f^{-1}(-\infty, a]$ e os segmentos de reta em negrito no epígrafo de f corresponde à $\pi_f^{-1}(a)$. Na direita temos o Grafo de Reeb de π_f , isto é, a merge tree clássica de f .

Perceba que topologicamente não existe diferença relevante entre as seguintes merge trees, uma vez que elas são homotópicas,



Desta forma, a partir deste ponto no texto, optamos em apresentar as merge trees esboçadas como o segundo modelo acima, por uma questão de praticidade técnica.

Definição 3.1.3. *Sejam $F : (\mathbb{R}, \leq) \rightarrow \mathbf{Top}$ um espaço de persistência e $S : (\mathbb{R}, \leq) \rightarrow \mathbf{Sets}$ um conjunto persistente qualquer. Definimos o **display poset** de S como sendo o conjunto*

$$\mathcal{S} := \bigsqcup_{t \in \mathbb{R}} S(t) = \bigcup_{t \in \mathbb{R}} S(t) \times \{t\},$$

munido da seguinte ordem parcial

$$(x, s) \leq (y, t) \Leftrightarrow S(s \leq t)(x) = y.$$

A **merge tree generalizada** de F é definida como sendo o display poset (M_F, \leq) associado ao conjunto persistente $\pi_0 \circ F$. Caso em uma merge tree generalizada (M_F, \leq) toda cadeia maximal tiver elemento minimal, dizemos que (M_F, \leq) possui **folhas**.

Essa estrutura de poset atribuída a merge tree generalizada é importante para nos dar a informação de quais componentes conexas se juntaram com quais. Isto é, nos dá a possibilidade de rastreamento desses eventos de colapsamento de componentes conexas.

Definição 3.1.4. *Seja (M_F, \leq) uma merge tree generalizada e $\pi_F : M_F \rightarrow \mathbb{R}$ a função projeção. A **altura merge** dos pontos $u, v \in M_F$ é dada por*

$$\text{merge}_F(u, v) := \inf \{ \pi_F(w) \mid u, v \leq w \}.$$

*Definimos o **menor ancestral comum** de u e v como sendo o menor limitante superior comum de u e v , isto é,*

$$\text{LCA}_F(u, v) := \text{argmin} \{ \pi_F(w) \mid u, v \leq w \}.$$

Quando o LCA_F existir temos que

$$\text{merge}_F(u, v) = \pi_F(\text{LCA}_F(u, v)).$$

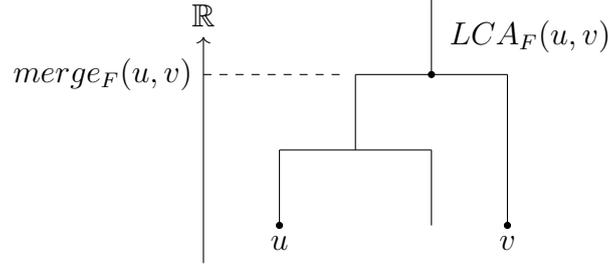
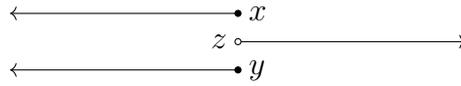


Figura 3.3: Exemplo de LCA_F e $merge_F$.

Caso todo par de pontos de uma merge tree possua menor ancestral comum, dizemos que a merge tree em questão é **conexa**.

Vale a pena destacar que nem sempre o LCA_F de dois pontos existe. No artigo [8], os autores nos dão o seguinte exemplo simples: Se $S : (\mathbb{R}, \leq) \rightarrow \mathbf{Sets}$ é um conjunto persistente tal que $S(t) = \{x, y\}$ para $t \leq 0$ e $S(t) = \{z\}$ para $t > 0$, fica evidente que x e y não admitem menor limitante superior.



Definição 3.1.5. Se a merge tree generalizada M_F for conexa, então a métrica l^p em (M_F, \leq) para $1 \leq p \leq \infty$ associa para cada par de pontos $u, v \in M_F$ o valor

$$d_{M_F}^p(u, v) = \|(merge_F(u, v) - \pi_F(u), merge_F(u, v) - \pi_F(v))\|_p$$

Estudaremos agora as merge trees um pouco mais afundo, colocando nelas uma topologia, que nos ajudará a obter resultados significativos quando avançarmos mais na teoria.

Consideremos X um espaço topológico compacto, $f : X \rightarrow \mathbb{R}$ uma função contínua e $\alpha : E_f \rightarrow T_f(X)$ a aplicação quociente de Reeb, que identifica dois pontos se eles estão na mesma componente conexa na filtração de sublevel-set.

Definimos a seguinte ordem parcial em $T_f(X)$: Dados $p, q \in T_f(X)$, temos que $p \leq q$ se existir $x \in X$ e $r, s \in \mathbb{R}$ tais que $f(x) \leq r \leq s$ e $p = \alpha(x, r)$ e $q = \alpha(x, s)$.

Lema 3.1.6. Sejam (x, r) e (y, s) pontos em E_f tal que $r \leq s$. As seguintes sentenças são equivalentes:

- i) $\alpha(x, r) \leq \alpha(y, s)$
- ii) $\alpha(x, s) = \alpha(y, s)$
- iii) $\alpha(x, t) = \alpha(y, t)$ para $t > s$.

Demonstração. $i) \Rightarrow ii)$. Seja $z \in X$ tal que $\alpha(z, r) = \alpha(x, r)$ e $\alpha(z, s) = \alpha(y, s)$, que existe pois por hipótese $\alpha(x, r) \leq \alpha(y, s)$, isto é, pertencem a mesma componente conexa. Considere C sendo a componente conexa de $\pi_f^{-1}(r)$ que contém (x, r) , D a componente conexa de $\pi_f^{-1}(s)$ que contém (y, s) e C' os pontos de C só que fazemos uma translação na segunda coordenada, transformando r em s . Como $\alpha(z, r) = \alpha(x, r)$, então (z, r) e (x, r) estão na mesma componente conexa, o que implica que $(z, r), (x, r) \in C$. Por uma argumentação semelhante temos que $(z, s), (y, s) \in D$. Do jeito que definimos C' , temos que $(z, s) \in C'$, pois $(z, r) \in C$. Como C' é conexo, necessariamente devemos ter que $C' \subseteq D$, portanto $\alpha(x, s) = \alpha(y, s)$.

$ii) \Rightarrow iii)$. Se tivermos $\alpha(x, s) = \alpha(y, s)$, com a ordem parcial de $T_f(X)$ temos que $\alpha(x, s) \leq \alpha(y, t)$ e com o argumento do item anterior, temos que $\alpha(x, t) = \alpha(y, t)$.

$iii) \Rightarrow i)$. Sejam $\{s_n\}_{n \in \mathbb{N}}$ uma sequência decrescente convergindo para s (por exemplo $s_n = s + 1/n$) e $C_n \times \{s_n\}$ a componente conexa de $\pi_f^{-1}(s_n)$ contendo ambos (x, s_n) e (y, s_n) . Perceba que os conjuntos C_n forma uma sequência de conjuntos fechados conexos encaixantes.

Seja $C := \bigcap_{n \in \mathbb{N}} C_n$, note que $C \times \{s\}$ está contido em E_f e $x, y \in C$. Vamos verificar que C é de fato conexo. Sejam V e W dois abertos de X tal que $C \subset V \sqcup W =: U$. Suponhamos que $x \in V$ e $U_n := X \setminus C_n$. Como C_n é fechado, temos que a coleção que consiste de U e dos U_n forma uma cobertura aberta de X , segue da hipótese que X é compacto que tal cobertura admite subcobertura finita. Como (U_n) é uma família crescente de abertos, isso implica que para n_0 suficientemente grande, temos que X é a união de U com U_{n_0} e portanto $C_{n_0} \subset U$.

Como C_n é conexo para todo $n \in \mathbb{N}$ e $C_n \subset V$ pois $x \in V$ então $C \subset V$ e C é conexo e o resultado segue. ■

Definição 3.1.7. Dado (X, \leq) um poset qualquer definimos o **conjunto superior** $U \subset X$ tal que para todo $u \in U$ e $x \in X$ se $u \leq x$ então $x \in U$. Podemos também considerar o conjunto superior relativo a um ponto p , denotado por U_p , que corresponde aos pontos $x \in X$ tais que $p \leq x$.

Corolário 3.1.8. 1. $(T_f(X), \leq)$ é um poset no sentido que o conjunto superior de qualquer ponto é uma cadeia, isto é, admite uma ordem total.

2. Seja C uma cadeia em $(T_f(X), \leq)$. Então C tem um ínfimo e o fecho de C na topologia quociente está contido no conjunto superior do seu ínfimo.

Demonstração. 1. Se $\alpha(x, r) \leq \alpha(y, s)$ e $\alpha(x, r) \leq \alpha(z, t)$ com $s \leq t$, então pelo Lema 3.1.6 temos que $\alpha(y, t) = \alpha(x, t) = \alpha(z, t)$ e novamente pelo lema temos $\alpha(y, s) \leq \alpha(z, t)$.

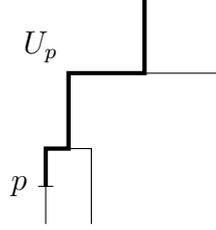


Figura 3.4: Representação gráfica do conjunto superior de um ponto p em uma merge tree qualquer.

2. Seja $(x_n, r_n)_{n \in \mathbb{N}}$ uma sequência em E_f tal que $\alpha(x_n, r_n) \in C$ para todo $n \in \mathbb{N}$,

$$\lim_{n \rightarrow \infty} r_n = r := \inf \{s \mid \alpha(x, s) \in C\}$$

e $x_n \rightarrow x \in E_f$ com $\alpha(x, t) \in C$ para $t \geq r$. Considere $(y_n, s_n)_{n \in \mathbb{N}}$ uma sequência convergente em E_f tal que $\alpha(y_n, s_n) \in C$ para todo $n \in \mathbb{N}$, $(y_n, s_n) \rightarrow (y, s)$.

Queremos mostrar que $\alpha(x, r) \leq \alpha(y, s)$. Considere $t > s$, sem perda de generalidade podemos assumir que $r_n, s_n \leq t$ para $n \in \mathbb{N}$, basta tomar t suficientemente grande para isso. Como C é uma cadeia, pelo Lema 3.1.6 temos que

$$\alpha(y_i, t) = \alpha(x_j, t), \text{ para todos } i, j.$$

Logo temos que (x_n, t) e (y_n, t) pertencem a mesma componente conexa. Assim considere D sendo a componente conexa de $\pi_f^{-1}(t)$ que contém (x_n, t) e (y_n, t) para todo $n \in \mathbb{N}$.

Agora, temos que \mathbb{R} é um espaço de Hausdorff e portanto $\{t\} \subset \mathbb{R}$ é um conjunto fechado e π_f é uma função contínua, logo $\pi_f^{-1}(t)$ é fechado em E_f e portanto D é fechado. Assim, $(x, t), (y, t) \in D$ e, como α é contínua, temos que $\alpha(x, t) = \alpha(y, t)$. Como $t > s$ arbitrário pelo Lema 3.1.6 novamente temos que $\alpha(x, r) \leq \alpha(y, s)$, o que mostra que $\alpha(x, r)$ é um limitante inferior.

Nos resta mostrar que é de fato o ínfimo que estamos tratando, isto é, a maior das cotas inferiores. Suponha que $\alpha(x', r')$ seja outra cota inferior de C . Para qualquer $t > r$, existe $n \in \mathbb{N}$ tal que $r_n \leq t$, logo teríamos $\alpha(x, t) = \alpha(x_n, t) = \alpha(x', t)$. Novamente pelo Lema 3.1.6 temos que $\alpha(x', r') \leq \alpha(x, r)$ o que mostra que é realmente o ínfimo. ■

Definição 3.1.9. *Seja $T_f(X)$ a merge tree vista como um poset. Um subconjunto $U \subseteq T_f(X)$ é aberto na **topologia de intervalo** se para cada $p \in U$ e cada $q \in X \setminus \{p\}$ comparável a p , existir $w \in U$ estritamente entre p e q tal que o intervalo entre p e w esta contido em U .*

Perceba como essa nova topologia se assemelha com a topologia da ordem em \mathbb{R} . Na verdade, o que estamos fazendo aqui é tentar argumentar que ao nos restringirmos a um

único ramo de uma merge tree, temos que esse objeto é de certa maneira correspondente a reta real. Isso ficará claro com os seguintes resultados que irão comparar as duas topologia apresentadas até aqui.

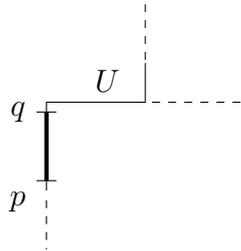


Figura 3.5: A figura acima é uma representação de um aberto da topologia de intervalo. A parte tracejada representa a merge tree subjacente, a parte com a reta cheia representa um conjunto U qualquer. Perceba que tomamos um q diferente do ponto p dado, mas que é comparável a p . O segmento de reta em negrito destacado corresponde aos possíveis w que podemos tomar de forma que U seja realmente um aberto na topologia de intervalo.

Lema 3.1.10. *O conjunto superior de qualquer ponto é fechado na topologia de intervalo.*

Demonstração. Seja C o conjunto superior de um ponto $\alpha(z, u) \in T_f(X)$ e considere $U := T_f(X) \setminus C$. Queremos mostrar que U é aberto na topologia de intervalo. Seja $\alpha(x, r) \in U$ e $\alpha(y, s)$ um ponto comparável a $\alpha(x, r)$. Temos dois casos a considerar:

Caso 1, $\alpha(y, s) < \alpha(x, r)$: Neste caso, qualquer intervalo entre eles está em U , já que U é o complementar de C e C é o conjunto superior, pois caso tivéssemos um ponto a no intervalo entre $\alpha(y, s)$ e $\alpha(x, r)$ que estivesse em C , teríamos que $\alpha(x, r) \in C$, o que seria um absurdo.

Caso 2, $\alpha(x, r) < \alpha(y, s)$: Pelo Lema 3.1.6 existe t tal que $r < t < s$ e $\alpha(x, t)$ que não está no conjunto superior de $\alpha(z, u)$. Então $\alpha(x, t)$ está estritamente entre $\alpha(x, s)$ e $\alpha(y, s)$ e o intervalo entre $\alpha(x, s)$ e $\alpha(x, t)$ está em U . Que é a definição de aberto na topologia de intervalo.

Portanto U é aberto e C é fechado. ■

Lema 3.1.11. *A função altura $h : T_f(X) \rightarrow \mathbb{R}$, $\alpha(x, r) \mapsto r$ é contínua com respeito a topologia de intervalo.*

Demonstração. Sejam $r \in \mathbb{R}$ e $\varepsilon > 0$. Queremos mostrar que $h^{-1}(r - \varepsilon, r + \varepsilon)$ é aberto em $T_f(X)$ com a topologia de intervalo. Suponha que $\alpha(x, s) \in T_f(X)$ e que $|s - r| > \varepsilon$. Seja ainda $\alpha(y, t)$ um ponto comparável a $\alpha(x, s)$. Se $\alpha(y, t) < \alpha(x, s)$, pelo Lema 3.1.6, temos que $\alpha(x, s) = \alpha(y, s)$. Tome $s' > t$ tal que $r - \varepsilon < s' < s$. Então $\alpha(y, s')$ está entre $\alpha(y, t)$ e $\alpha(x, s)$, de fato usamos novamente o Lema 3.1.6, como $s' < t$ temos que

$\alpha(y, s') < \alpha(y, t)$ e como $s' < s$ temos $\alpha(y, s') < \alpha(y, s) = \alpha(x, s)$. Portanto a imagem deste intervalo está em $[s', s] \subset (r - \varepsilon, r + \varepsilon)$.

Caso $\alpha(y, t) > \alpha(x, s)$, então $\alpha(x, t) = \alpha(y, t)$. Tome $s' < t$ tal que $s < s' < r + \varepsilon$. Usando argumentos semelhantes aos do caso anterior, temos que $\alpha(x, s')$ está estritamente entre $\alpha(x, s)$ e $\alpha(y, t)$ e a imagem deste intervalo está em $[s, s'] \subset (r - \varepsilon, r + \varepsilon)$. Portanto h é contínua. ■

Lema 3.1.12. *Para cada $x \in X$, a aplicação $\psi_x : [f(x), \infty) \rightarrow T_f(X)$, $r \mapsto \alpha(x, r)$ é contínua com respeito a topologia de intervalo.*

Demonstração. De fato, se U é um aberto na topologia de intervalo que contém $\psi_x(r) = \alpha(x, r)$, tomando $0 < \varepsilon < |f(x) - r|$, temos que $\psi_x(r)$ está no intervalo entre $\alpha(x, r - \varepsilon)$ e $\alpha(x, r + \varepsilon)$, que por sua vez tem imagem inversa em $(r - \varepsilon, r + \varepsilon)$. ■

Proposição 3.1.13. *Se $(T_f(X), \leq)$ tiver uma quantidade finita de folhas, então a topologia quociente coincide com a topologia de intervalo.*

Demonstração. Sejam C_1, \dots, C_n os conjuntos superiores das folhas de $T_f(X)$. Temos pelo Corolário 3.1.8 e Lema 3.1.10 que cada C_i é fechado em ambas topologias. Portanto temos que um conjunto U é aberto em ambas as topologias se, e somente se, sua interseção com cada C_i for aberto na topologia de subespaço.

Portanto é suficiente mostrar que a topologia de subespaço em C_i coincide na topologia de intervalo e na topologia quociente. Ambas as funções altura $\alpha(x, r) \mapsto r$ do Lema 3.1.11 e ψ_x do Lema 3.1.12 são contínuas com respeito a ambas topologias. Considerando $\alpha(x, f(x))$ denotando a folha de C_i e restringirmos h a C_i , temos que as duas aplicações são uma o inverso da outra, nos dando um isomorfismo em **Top**. ■

Proposição 3.1.14. *Seja $f : X \rightarrow \mathbb{R}$ uma função contínua e M_f sua merge tree clássica que possui uma quantidade finita de folhas. Então a topologia induzida pela métrica l^p coincide com a topologia quociente.*

Demonstração. A demonstração é semelhante a da Proposição 3.1.13, precisamos mostrar que as topologias coincidem em C_i . De fato, a interseção de uma bola da métrica l^p com cada conjunto superior C_i é relativamente aberto com respeito a topologia de intervalo, pois essa interseção é homeomorfa a um aberto pela função altura. Da mesma forma, temos que conjuntos relativamente abertos na topologia de intervalo são abertos na topologia induzida pela métrica l^p . ■

Tendo esses resultados em mãos, fica claro que quando tivermos um espaço X compacto e uma árvore associada a este espaço com uma quantidade finita de folhas, podemos ignorar as diferenças teóricas entre as merge trees clássicas e generalizadas e tratá-las como sendo um único objeto.

Isso é especialmente relevante para o que propomos a fazer, uma vez que a TDA trabalha com banco de dados finitos e computadores para fazer a análise. Desta forma, na vida real, só trabalhamos com árvores que tem uma quantidade finita de folhas e não perdemos nada em trabalhar com merge trees clássicas ou generalizadas.

3.2 Merge Trees Decoradas

Nesta seção finalmente iremos estudar como decorar as Merge Trees, para que seja possível estudar as informações vindas do funtor π_0 quanto as vindas do funtor H_n em um único objeto.

Definição 3.2.1. *Seja $F : (\mathbb{R}, \leq) \rightarrow \mathbf{Top}_{lc}$ um espaço de persistência e $\tilde{F} : (\mathbb{R}, \leq) \rightarrow \mathbf{pTop}_c$ seu espaço parametrizado persistente associado. A **merge tree decorada categorial** de grau n é o funtor*

$$\tilde{F}_n := H_n \circ \tilde{F} : (\mathbb{R}, \leq) \rightarrow \mathbf{pVec},$$

onde cada $s \in \mathbb{R}$ é levado no funtor $I(s) : \pi_0(F(s)) \rightarrow \mathbf{Vec}$, com I mandando cada índice i das componentes de $\pi_0(F(s))$ no espaço vetorial de homologia, isto é, $H_n(F(s)_i)$.

Vejamos como podemos aplicar esta definição para distinguir conjuntos nos quais a Persistência Homológica usual falha em diferenciar.

Exemplo 3.2.2. *Considere X e Y dois subespaços de \mathbb{R}^2 como na Figura 3.6 e façamos a filtração off-set.*

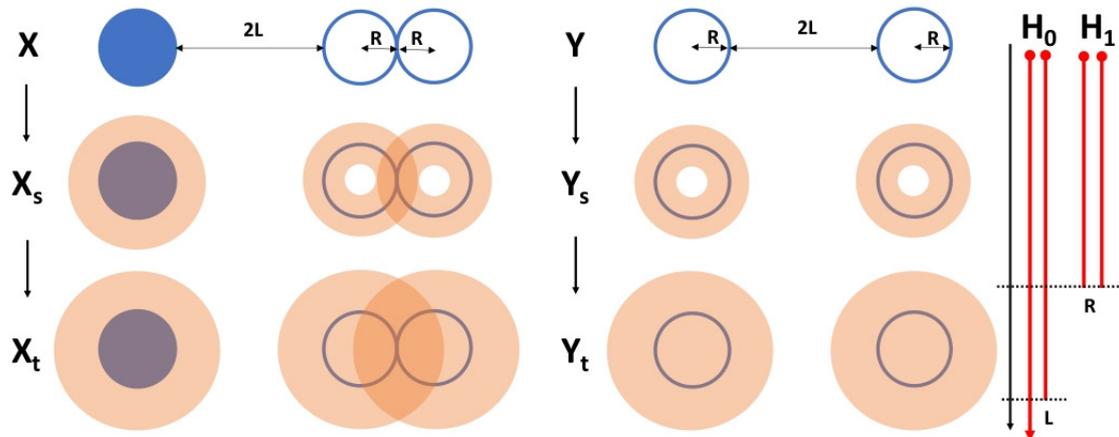


Figura 3.6: Retirado de [8]

Perceba que aplicando a teoria de homologia persistente usual obtemos os mesmos barcodes para ambos os espaços X e Y , entretanto é visível que os espaços não são equivalentes. Precisamos de mais informações para determinar a diferença entre os dois espaços, essa é a utilidade de aplicar as Merge Trees na teoria de TDA.

Fazemos o seguinte: Consideramos as filtrações off-set F e G de X e Y , respectivamente. Usando a merge tree decorada categorial de grau 1, podemos facilmente mostrar que X e Y não são homeomorfos. Sejam \tilde{F}_1 e \tilde{G}_1 as merge trees decoradas categorial de F e G , para verificar que $\tilde{F}_1 \not\cong \tilde{G}_1$, basta analisá-las em $s = 0$. Como X e Y tem a mesma quantidade de componentes conexas em $s = 0$, denotamos

$$\tilde{F}_1 := I : \{x, y\} \rightarrow \mathbf{Vec}; \quad \tilde{G}_1 := J : \{a, b\} \rightarrow \mathbf{Vec}.$$

Como uma das componentes de X tem homologia trivial e a outra tem homologia isomorfa a \mathbb{K}^2 , então podemos considerar $I(x) = \mathbb{K}^2$ e $I(y) = 0$. De maneira análoga $J(a) = J(b) = \mathbb{K}$. Assim não podemos ter um isomorfismo entre esses funtores, pois uma bijeção entre T e S iria implicar em uma transformação linear da forma

$$\mathbb{K}^2 \rightarrow \mathbb{K} \rightarrow \mathbb{K}^2,$$

que pelo Teorema do Núcleo e Imagem não pode ser isomorfismo.

Embora a merge tree decorada categorial seja eficiente para tratarmos da teoria já estabelecida de persistência homológica, gostaríamos de introduzir uma versão de merge tree decorada que lide bem com a estrutura de poset de uma merge tree generalizada. Nosso intuito é tentar ter uma forma mais computável deste objeto, nos possibilitando realmente aplicar essa teoria. A próxima definição nos indica como fazer isso.

Definição 3.2.3. *Seja $F : (\mathbb{R}, \leq) \rightarrow \mathbf{Top}_{lc}$ um espaço de persistência e considere a merge tree generalizada (M_F, \leq) . Definimos a **merge tree decorada concreta** de grau n como sendo o funtor*

$$\mathcal{F}_n : (M_F, \leq) \rightarrow \mathbf{Vec}_{\mathbb{K}}; \quad (i, s) \mapsto H_n(F(s)_i, \mathbb{K})$$

que registra o n -ésimo espaço vetorial de homologia da i -ésima componente conexa do espaço topológico localmente conexo $F(s)$.

Podemos generalizar ainda mais a definição de merge tree decorada concreta, de maneira que independa do espaço de persistência F , resultando em um funtor $\mathcal{F} : (M_F, \leq) \rightarrow \mathbf{Vec}$ definida em uma merge tree generalizada associada a um espaço persistente qualquer. Em [8], os autores frisam que chamarão essa estrutura de **módulo de árvore**, quando não quiserem dar ênfase ao espaço persistente que origina a merge tree generalizada.

Temos assim, duas maneiras de rastrear as homologias das componentes em uma filtração, entretanto essas ferramentas construídas acima não possuem uma representação gráfica de fácil compreensão como os barcodes ou os diagramas de persistências. Isso se dá pelo fato que a merge tree generalizada possui uma ordem parcial mas não total. Apresentamos uma forma de contornar esse problema.

Definição 3.2.4. *Uma **merge tree decorada por barcode** é uma aplicação de uma merge tree generalizada no conjunto de todos os barcodes*

$$\mathcal{B} : (M_F, \leq) \rightarrow \mathbf{Barcodes}.$$

Dizemos que a merge tree decorada por barcode é **determinada por restrição** se toda vez que tivermos $p := (i, s) \leq (j, t) =: q \in M_F$, tivermos

$$\mathcal{B}(q) = \mathcal{B}(p) \cap [t, \infty).$$

Se a merge tree generalizada subjacente de uma merge tree decorada por barcode determinada por restrição tiver folhas, chamamos ela de **merge tree decorada por folhas**.

Definição 3.2.5. *Suponha que (M_F, \leq) seja a merge tree generalizada associada ao conjunto persistente $\pi_0 \circ F$. Dado um módulo de árvore $\mathcal{F} : (M_F, \leq) \rightarrow \mathbf{Vec}$ e um ponto $p = (i, s) \in M_F$ definimos a **restrição de \mathcal{F} ao conjunto superior U_p** como sendo o \mathbb{R} -módulo*

$$\mathcal{F}|_{U_p} : (\mathbb{R}, \leq) \rightarrow \mathbf{Vec},$$

onde para $s \leq t$ temos

$$\mathcal{F}|_{U_p}(t) = \mathcal{F}(\pi_0 \circ F(s \leq t)(i), t)$$

e para $r < s$ definimos $\mathcal{F}|_{U_p}(r) = 0$.

Proposição 3.2.6. *Assuma que a merge tree generalizada (M_F, \leq) possua folhas. Para qualquer módulo de árvore pontualmente finito $\mathcal{F} : (M_F, \leq) \rightarrow \mathbf{vec}$, temos uma merge tree decorada por folhas*

$$\mathcal{BF} : (M_F, \leq) \rightarrow \mathbf{Barcodes}; \mathcal{BF}(p) = BC(\mathcal{F}|_{U_p}),$$

onde BC denota o Barcode do módulo de árvore e U_p o conjunto superior de p . Como (M_F, \leq) possui folhas, então é suficiente computar esse barcode para cada folha v .

Demonstração. Por hipótese temos que o módulo de árvore em questão é pontualmente finito, logo restringindo-o ao conjunto superior U_p nos dá um módulo pontualmente finito. Portanto, podemos aplicar o Teorema de Crawley-Boevey, Teorema 1.2.9, que nos garante que tal módulo tenha uma decomposição em barcodes. ■

Definição 3.2.7. A aplicação \mathcal{BF} definida na Proposição 3.2.6 é chamada de **transformada de barcode de \mathcal{F}** .

Para a definição e lema subsequentes, consideremos os barcodes como sendo um conjunto finito e cada intervalo no barcode será semi-aberto da forma $[b, d)$, onde $0 \leq b \leq d \leq \infty$.

Definição 3.2.8. Seja B um barcode e $I = [b, d) \in B$. Definimos um **truncamento de I** em uma altura h , como sendo:

- i) Igual a I se $h \leq b$;
- ii) O intervalo $[h, d)$ se $b \leq h \leq d$;
- iii) O intervalo vazio se $h > d$.

Denotaremos o truncamento de I na altura h por $\text{trunc}_h(I)$. O **truncamento de B** na altura h é o barcode obtido após realizar o truncamento na altura h de todos os intervalos que compõem B , que será denotado por $\text{trunc}_h(B)$.

Lema 3.2.9. Sejam B e B' barcodes cujos intervalos $[b, d)$ todos satisfazem $b \leq H$ para alguma constante $H \leq 0$. Suponha que exista um δ -pareamento entre B e B' , então para $\varepsilon \leq \delta$ existe um δ -pareamento entre B e $\text{trunc}_{H+\varepsilon}(B')$.

Demonstração. Seja ξ um δ -pareamento de B e B' . Defina um pareamento $\hat{\xi}$ entre B e $\text{trunc}_{H+\varepsilon}(B')$ da seguinte maneira

$$\begin{aligned} \text{dom}(\hat{\xi}) &:= \{I \in \text{dom}(\xi) \mid \text{trunc}_{H+\varepsilon}(\xi(I)) \neq \emptyset\} \\ \text{Im}(\hat{\xi}) &:= \{I' \in \text{Im}(\xi) \mid \text{trunc}_{H+\varepsilon}(I') \neq \emptyset\} \\ \hat{\xi}(I) &:= \text{trunc}_{H+\varepsilon}(\xi(I)). \end{aligned}$$

Seja $I = [b, d) \in \text{dom}(\hat{\xi})$ com $I' = [b', d') = \xi(I)$, temos que

$$b' \leq H \leq H + \varepsilon \leq d',$$

pois, caso contrário, teríamos que $\text{trunc}_{H+\varepsilon}(\xi(I)) = \emptyset$, contradizendo o que supomos acima. Mais do que isso, temos $\text{trunc}_{H+\varepsilon}(I') = [H + \varepsilon, d')$. Como ξ é um δ -pareamento, então o custo de ξ , dado por

$$c_\xi := \max \left\{ \max_{I \in \text{dom}(\xi)} \|I - \xi(I)\|_\infty, \max_{I \in B \setminus \text{dom}(\xi)} \|I\|_\Delta, \max_{I' \in B' \setminus \text{Im}(\xi)} \|I'\|_\Delta \right\},$$

então temos que

$$|d - d'| \leq c_\xi \leq \delta.$$

Analogamente, temos que $|b - b'| \leq \delta$. Perceba que

$$|b - b'| \leq \delta \Rightarrow -\delta \leq b - b' \leq \delta \Rightarrow \delta \geq b - b' \geq b - (H + \varepsilon),$$

uma vez que $\text{trunc}_{H+\varepsilon}(I')$ implica que $b' \leq H + \varepsilon$. Além disso, temos

$$b \leq H \Rightarrow H - b \geq 0 \Rightarrow \delta \geq \varepsilon \geq \varepsilon + (H - b) = -(b - (H + \varepsilon)),$$

o que implica que

$$|b - (H + \varepsilon)| \leq \delta.$$

Portanto o custo do pareamento de I com $\text{trunc}_{H+\varepsilon}(I')$ é menor que δ . Por outro lado, se $I \notin \text{dom}(\hat{\xi})$, então ou $I \notin \text{dom}(\xi)$, ou $\text{trunc}_{H+\varepsilon}(\xi(I')) = \emptyset^3$. Caso $I \notin \text{dom}(\xi)$ acabou, caso $\text{trunc}_{H+\varepsilon}(\xi(I))$, então $I = [b, d]$ e $I' = [b', d'] = \xi(I)$. Como nesse caso $\text{trunc}_{H+\varepsilon}(\xi(I)) = \emptyset$, temos que $d \leq H + \varepsilon$, logo temos que

$$d - b = d - d' + d' - b \leq \delta + d' - b \leq d' + \delta - H \leq H + \varepsilon + \delta - H = \varepsilon + \delta \leq 2\delta,$$

assim temos que

$$\frac{d - b}{2} \leq \delta.$$

Portanto, obtemos $\|[b, d]\|_{\Delta} \leq \delta$. Com argumentos análogos podemos mostrar o mesmo para $I' \in \text{Im}(\hat{\xi})$. Concluimos então que de fato existe um δ -pareamento entre B e $\text{trunc}_{H+\varepsilon}(B')$. ■

Teorema 3.2.10 (Continuidade da Merge Tree Decorada por Barcode). *Seja (M_F, \leq) uma merge tree generalizada conexa associada a um conjunto de persistência $\pi_0 \circ F$. Considere M_F munida com a métrica estendida $d_{M_F}^p$ e **Barcodes** munido com a métrica bottleneck d_B . Para qualquer módulo de árvore de dimensão pontualmente finita $\mathcal{F} : (M_F, \leq) \rightarrow \text{vec}$, a merge tree decorada associada*

$$\mathcal{BF} : M_F \rightarrow \text{Barcodes}$$

é $2^{1-1/p}$ -Lipschitz se $1 \leq p < \infty$ e 2-Lipschitz para $p = \infty$.

Demonstração. Vamos usar a ideia de truncamento apresentada acima para argumentar que a merge tree decorada por barcode é 1-Lipschitz. De fato, sejam $p = (i, s), q = (j, t) \in M_F$ e considere $r_0 := \text{merge}_F(p, q)$. Desta forma, temos

$$d_{M_F}^1(p, q) = \|(\text{merge}_F(p, q) - \pi_F(p), \text{merge}_F(p, q) - \pi_F(q))\|_1$$

³Perceba que esse “ou” é um “ou exclusivo”.

$$\begin{aligned}
&= |r_0 - s| + |r_0 - t| \\
&= (r_0 - s) + (r_0 - t),
\end{aligned}$$

lembrando que $\text{merge}_F(p, q) \geq \pi_F(p), \pi_F(q)$.

Para $\delta > 0$ qualquer, tomamos $z = (k, r) \in M_F$ tal que z é ancestral comum de p e q com $r < r_0 + \delta$. Obtemos $\mathcal{BF}(z)$ como sendo o truncamento de $\mathcal{BF}(p)$ na altura r , assim temos

$$d_B(\mathcal{BF}(z), \mathcal{BF}(p)) \leq r - s,$$

pois na definição da distância bottleneck, Definição 1.2.10, o ínfimo dos máximos das distâncias dos barcodes é no máximo $r - s$. Analogamente temos que

$$d_B(\mathcal{BF}(z), \mathcal{BF}(q)) \leq r - q.$$

Da Desigualdade Triangular, temos que

$$\begin{aligned}
d_B(\mathcal{BF}(p), \mathcal{BF}(q)) &\leq d_B(\mathcal{BF}(z), \mathcal{BF}(p)) + d_B(\mathcal{BF}(z), \mathcal{BF}(q)) \\
&= r - s + r - t \\
&< r_0 + \delta - s + r_0 + \delta - t \\
&= d_{M_F}^1 + 2\delta.
\end{aligned}$$

Como tomamos um δ qualquer, temos então que \mathcal{BF} é 1-Lipschitz. Para mostrar o resto do resultado basta lembrarmos da seguinte desigualdade de normas

$$\|x\|_p \leq \|x\|_r \leq n^{1/r-1/p} \|x\|_p, \quad \text{para } x \in \mathbb{R}^n, \quad 1 \leq r < p.$$

Desta forma, tomando $n = 2, r = 1$ temos que \mathcal{BF} é $2^{1-1/p}$ -Lipschitz, para $1 \leq p < \infty$ e no caso em que $p = \infty$ temos que \mathcal{BF} é 2-Lipschitz. ■

3.3 Métricas Entre Merge Trees

Nesta parte gostaríamos de quantificar quanto uma merge tree dista da outra, assim iremos apresentar métricas entre elas e compará-las e assim identificar possíveis situações onde essas métricas sejam intercambiáveis.

Apresentamos primeiramente a métrica interleaving entre merge trees clássicas, presente em [26].

Definição 3.3.1. *Sejam $f : X \rightarrow \mathbb{R}$ e $g : Y \rightarrow \mathbb{R}$ funções contínuas e M_f e M_g suas merge trees clássicas associadas. Para $\varepsilon > 0$ definimos uma ε -aplicação como sendo*

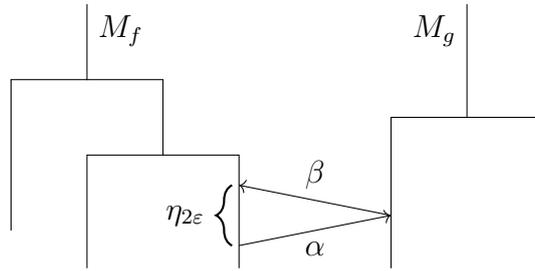
uma função contínua, com respeito a topologia quociente, $\alpha : M_f \rightarrow M_g$ tal que

$$\tilde{\pi}_g \circ \alpha(\bar{x}, t) = t + \varepsilon,$$

para todo $(\bar{x}, t) \in M_f$ e $\tilde{\pi}_g$ é a projeção induzida por $\pi_g : E_g \rightarrow \mathbb{R}$. Duas ε -aplicações $\alpha : M_f \rightarrow M_g$ e $\beta : M_g \rightarrow M_f$ são ditas serem **ε -compatíveis** se valer

$$\beta \circ \alpha = \eta_{2\varepsilon} \quad \text{e} \quad \alpha \circ \beta = \eta_{2\varepsilon},$$

onde $\eta_{2\varepsilon}$ representa apenas uma translação em 2ε .



Escolhemos essa representação para fazer uma correspondência com a distância interleaving do Capítulo 2, entretanto frisamos que $\eta_{2\varepsilon}$ não é necessariamente uma transformação natural.

Definimos a **distância interleaving Morozov-Beketayev-Weber** entre M_f e M_g como sendo

$$\theta_I^{MBW}(M_f, M_g) := \inf \{ \varepsilon \geq 0 \mid \exists \text{ aplicações } \varepsilon\text{-compatíveis } \alpha : M_f \rightarrow M_g \text{ e } \beta : M_g \rightarrow M_f \}.$$

Precisamos verificar que θ_I^{MBW} é de fato uma métrica. Que $\theta_I^{MBW}(M_f, M_f) = 0$ é trivial, basta tomar as funções identidades. Além disso, a própria definição de θ_I^{MBW} nos garante que é simétrica. Basta só verificarmos que ela satisfaz a Desigualdade Triangular.

Considere M_f, M_g e M_h merge trees clássicas, de forma que

$$\theta_I^{MBW}(M_f, M_g) = \varepsilon_1 \quad \text{e} \quad \theta_I^{MBW}(M_g, M_h) = \varepsilon_2.$$

Então, para todo $\delta > 0$, existem aplicações $\left(\varepsilon_1 + \frac{\delta}{2}\right)$ -compatíveis

$$\rho : M_f \rightarrow M_g \quad \text{e} \quad \sigma : M_g \rightarrow M_h$$

e aplicações $\left(\varepsilon_2 + \frac{\delta}{2}\right)$ -compatíveis

$$\varphi : M_g \rightarrow M_h \quad \text{e} \quad \psi : M_h \rightarrow M_g.$$

Considere $\varepsilon_3 := \varepsilon_1 + \varepsilon_2$ e defina $\alpha : M_f \rightarrow M_h$ e $\beta : M_h \rightarrow M_f$ da seguinte forma

$$\alpha := \varphi \circ \rho \quad \text{e} \quad \beta := \sigma \circ \psi.$$

Precisamos verificar que elas são realmente $(\varepsilon_3 + \delta)$ -compatíveis. De fato

$$\beta \circ \alpha = \sigma \circ \psi \circ \varphi \circ \rho = \sigma \circ \eta_{2(\varepsilon_1 + \delta/2)} \circ \rho = \eta_{2(\varepsilon_1 + \varepsilon_2 + \delta)} = \eta_{2(\varepsilon_3 + \delta)}.$$

O mesmo raciocínio pode ser usado para mostrar que $\alpha \circ \beta = \eta_{2(\varepsilon_3 + \delta)}$. Isso mostra que α e β são $(\varepsilon_3 + \delta)$ -compatíveis e como isso vale para todo $\delta > 0$ temos então que

$$\theta_I^{MBW}(M_f, M_h) \leq \varepsilon_3 = \theta_I^{MBW}(M_f, M_g) + \theta_I^{MBW}(M_g, M_h).$$

Concluimos assim que a θ_I^{MBW} é uma métrica.

Como nos Capítulos anteriores, nos perguntamos: As Merge Trees munidas com a métrica θ_I^{MBW} são, de alguma forma, estáveis? O próximo teorema nos garante que sim.

Teorema 3.3.2 (*Estabilidade das Merge Trees*). *Sejam $f, g : X \rightarrow \mathbb{R}$ duas funções quaisquer e M_f, M_g suas respectivas Merge Trees clássicas. Temos portanto que*

$$\theta_I^{MBW}(M_f, M_g) \leq \|f - g\|_\infty.$$

Demonstração. Considere F, G as filtrações de sublevel-set de f e g respectivamente. Prosseguindo de maneira análoga à demonstração do Teorema 2.2.18, tomamos $\varepsilon := \|f - g\|_\infty$ e obtemos para cada $a \in \mathbb{R}$ as seguintes inclusões

$$F(a) \subset G(a + \varepsilon) \subset F(a + 2\varepsilon).$$

Perceba que para cada $(\bar{x}, a) \in M_f$ representa uma componente conexa em $F(a) \subset G(a + \varepsilon)$, desta forma definimos da inclusão a seguinte função

$$\begin{aligned} \alpha : M_f &\rightarrow M_g \\ (\bar{x}, t) &\mapsto (\bar{x}', t + \varepsilon), \end{aligned}$$

onde \bar{x}' corresponde ao respectivo ponto que \bar{x} é mandado pela inclusão em M_g . Além disso, temos também que

$$G(a) \subset F(a + \varepsilon) \subset G(a + 2\varepsilon),$$

e podemos definir β de maneira análoga a α . Perceba que α e β são ambas ε -aplicações, uma vez que

$$\tilde{\pi}_g \circ \alpha(\bar{x}, t) = t + \varepsilon = \tilde{\pi}_f \circ \beta(\bar{y}, t).$$

Fica claro que por construção α e β são ε -compatíveis, pois estamos usando as inclusões neste caso e se \bar{x} é levado em \bar{x}' por α , aplicando β devemos ter que \bar{x}' é levado em \bar{x} , isto é,

$$\beta \circ \alpha = \eta_{2\varepsilon} \text{ e } \alpha \circ \beta = \eta_{2\varepsilon}.$$

Assim, temos que $\theta_I^{MBW}(M_f, M_g) \leq \varepsilon$. ■

Definição 3.3.3. *Sejam $f : X \rightarrow \mathbb{R}$, $g : Y \rightarrow \mathbb{R}$ funções contínuas e M_f, M_g suas respectivas merge trees clássicas associadas. Consideramos as filtrações de sublevel-set $F, G : (\mathbb{R}, \leq) \rightarrow \mathbf{Top}$ de f e g respectivamente. Definimos a **distância interleaving entre as merge trees M_f e M_g** como sendo*

$$\theta_I(M_f, M_g) := d_I(\pi_0 \circ F, \pi_0 \circ G).$$

Fica claro que uma ε -aplicação $\alpha : M_f \rightarrow M_g$ qualquer leva a componente $f^{-1}(-\infty, t]$ em $g^{-1}(-\infty, t + \varepsilon]$. Assim temos uma transformação natural $\alpha : \pi_0 \circ F \Rightarrow \pi_0 \circ G \circ T_\varepsilon$ dada pelo seguinte diagrama comutativo

$$\begin{array}{ccc} \pi_0 \circ F(s) & \xrightarrow{\alpha_s} & \pi_0 \circ G \circ T_\varepsilon(s) \\ \downarrow & \circlearrowleft & \downarrow \\ \pi_0 \circ F(s') & \xrightarrow{\alpha_{s'}} & \pi_0 \circ G \circ T_\varepsilon(s') \end{array}$$

Para comparar as duas distâncias interleavings entre merge trees apresentadas, primeiramente enunciamos o seguinte resultado que relaciona funções ε -compatíveis com funtores ε -interleaved.

Lema 3.3.4. *Sejam $\alpha : M_f \rightarrow M_g$ e $\beta : M_g \rightarrow M_f$ são ε -aplicações que são ε -compatíveis entre si. Então, α e β induzem um ε -interleaving entre $\pi_0 \circ F$ e $\pi_0 \circ G$, onde $F, G : (\mathbb{R}, \leq) \rightarrow \mathbf{Top}$ são as filtrações de sublevel-set de f e g respectivamente.*

Demonstração. Como argumentamos acima, α e β definem transformações naturais $\alpha : \pi_0 \circ F \Rightarrow \pi_0 \circ G \circ T_\varepsilon$ e $\beta : \pi_0 \circ G \Rightarrow \pi_0 \circ F \circ T_\varepsilon$. Aplicamos α e β a definição de ε -interleaving, tomando $\alpha = \varphi$ e $\beta = \psi$

$$\begin{array}{ccccc} (\mathbb{R}, \leq) & \xrightarrow{T_\varepsilon} & (\mathbb{R}, \leq) & \xrightarrow{T_\varepsilon} & (\mathbb{R}, \leq) \\ \pi_0 \circ F \downarrow & \xRightarrow{\alpha} & \downarrow \pi_0 \circ G & \xRightarrow{\beta} & \downarrow \pi_0 \circ F \\ \mathbf{Sets} & \xrightarrow{1_{\mathbf{Sets}}} & \mathbf{Sets} & \xrightarrow{1_{\mathbf{Sets}}} & \mathbf{Sets} \end{array}$$

Portanto $(\beta T_\varepsilon)\alpha = \pi_0 \circ F \circ \eta_{2\varepsilon}$ e $(\alpha T_\varepsilon)\beta = \pi_0 \circ G \circ \eta_{2\varepsilon}$, pois α e β são ε -compatíveis, e o resultado segue. ■

Infelizmente, não temos garantido ainda a continuidade das aplicações entre as merge trees M_f e M_g e observando o Lema anterior, o fato que as ε -aplicações ε -compatíveis induzem ε -interleaving nos dá um refinamento da métrica θ_I , isto é,

$$\theta_I(M_f, M_g) \leq \theta_I^{MBW}(M_f, M_g).$$

Queremos garantir condições para as quais as duas métricas coincidam.

Proposição 3.3.5. *Sejam X, Y espaços topológicos compactos, $f : X \rightarrow \mathbb{R}, g : Y \rightarrow \mathbb{R}$ funções contínuas e suas merge trees clássicas associadas M_f, M_g , respectivamente, possuindo uma quantidade finita de folhas e $F, G : (\mathbb{R}, \leq) \rightarrow \mathbf{Top}$ as filtrações de sublevel-set de f e g . Todo ε -interleaving dos conjuntos de persistência $\pi_0 \circ F$ e $\pi_0 \circ G$ definem um par de aplicações ε -compatíveis entre as merge trees M_f e M_g . Consequentemente*

$$\theta_I^{MBW}(M_f, M_g) = \theta_I(M_f, M_g) = d_I(\pi_0 \circ F, \pi_0 \circ G).$$

Demonstração. Considere $\varphi : \pi_0 \circ F \Rightarrow \pi_0 \circ G \circ T_\varepsilon$ e $\psi : \pi_0 \circ G \Rightarrow \pi_0 \circ F \circ T_\varepsilon$ as transformações naturais definindo o ε -interleaving da hipótese. Consideramos o display poset de $\pi_0 \circ F$ e $\pi_0 \circ G$, isto é, as merge trees generalizadas M_F e M_G . Segue dos argumentos da seção 3.1 que, como poset, M_f coincide com M_F e M_g coincide com M_G .

Equipando as merge trees generalizadas com a topologia de intervalo, pelo Lema 3.1.10 o conjunto superior é fechado. Então, como por hipótese temos uma quantidade finita de folhas, fazemos uma cobertura das merge trees pelos conjuntos superior das folhas de cada merge tree, como feito na demonstração da Proposição 3.1.13. Desta forma, fica evidente que a função $\varphi : M_F \rightarrow M_G$ é contínua quando restrita ao conjunto superior de uma folha, segue do Lema da Colagem que φ é contínua em M_F .

Analogamente, podemos mostrar que $\psi : M_G \rightarrow M_F$ é contínua e pela Proposição 3.1.13 temos que a topologia de quociente e a de intervalo coincidem, logo φ e ψ são contínuas na topologia quociente e portanto podem ser vistas como aplicações ε -compatíveis e o resultado segue. ■

Definição 3.3.6. *Sejam M_F e M_G duas merge trees generalizadas dos conjuntos de persistência $\pi_0 \circ F$ e $\pi_0 \circ G$, respectivamente. Dadas duas merge trees decoradas por barcodes*

$$\mathcal{B}_F : M_F \rightarrow \mathbf{Barcodes}, \quad e \quad \mathcal{B}_G : M_G \rightarrow \mathbf{Barcodes},$$

definimos um (ε, δ) -pareamento de \mathcal{B}_F e \mathcal{B}_G como sendo:

- Um ε -interleaving das merge trees generalizadas subjacentes, ou seja, transformações naturais $\varphi : \pi_0 \circ F \Rightarrow \pi_0 \circ G \circ T_\varepsilon$ e $\psi : \pi_0 \circ G \Rightarrow \pi_0 \circ F \circ T_\varepsilon$ satisfazendo (2.1);
- Um δ -pareamento dos barcodes de $\mathcal{B}_F(p)$ e $\mathcal{B}_G(\varphi(p))$ para todo $p \in M_F$ e um δ -pareamento entre os barcodes de $\mathcal{B}_G(q)$ e $\mathcal{B}_F(\psi(q))$ para todo $q \in M_G$.

Definição 3.3.7. Seja $p \in [1, \infty]$. A **p -distância decorada bottleneck** entre duas merge trees decoradas por barcodes \mathcal{B}_F e \mathcal{B}_G é dada por

$$d_{B,p} := \inf \{ \|(\varepsilon, \delta)\|_p \mid \exists (\varepsilon, \delta)\text{-pareamento de } \mathcal{B}_F \text{ e } \mathcal{B}_G \}.$$

Quando $p = \infty$ iremos nos referir à essa distância somente como **distância decorada bottleneck** e $d_B := d_{B,\infty}$.

No caso que tivermos um (ε, δ) -pareamento de uma merge tree decorada por barcodes, onde $\varepsilon = \delta$ chamaremos simplesmente de ε -pareamento.

Proposição 3.3.8. Sejam $\mathcal{B}_F : (M_F, \leq) \rightarrow \mathbf{Barcodes}$ e $\mathcal{B}_G : (M_G, \leq) \rightarrow \mathbf{Barcodes}$ merge trees decoradas por barcodes determinadas por suas folhas. A distância decorada bottleneck pode ser expressa por

$$d_B(\mathcal{B}_F, \mathcal{B}_G) = \inf \{ \varepsilon \geq 0 \mid \exists \varepsilon\text{-pareamento de } \mathcal{B}_F \text{ e } \mathcal{B}_G \}.$$

Demonstração. Seja

$$\hat{d}_B(\mathcal{B}_F, \mathcal{B}_G) := \inf \{ \varepsilon \geq 0 \mid \exists \varepsilon\text{-pareamento de } \mathcal{B}_F \text{ e } \mathcal{B}_G \}.$$

Queremos mostrar que $d_B = \hat{d}_B$. Perceba que d_B minimiza um conjunto maior de elementos em relação a \hat{d}_B , assim temos que⁴

$$d_B \leq \hat{d}_B.$$

Precisamos apenas mostrar que $\hat{d}_B \leq d_B$. Sejam φ e ψ funções que definem um (ε, δ) -pareamento, isto é, φ e ψ definem um ε -interleaving entre $\pi_0 \circ F$ e $\pi_0 \circ G$, δ -pareamento entre $\mathcal{B}_F(p)$ e $\mathcal{B}_G(\varphi(p))$ para todo $p \in M_F$ e δ -pareamento entre $\mathcal{B}_G(q)$ e $\mathcal{B}_F(\psi(q))$ para todo $q \in M_G$.

Se $\varepsilon \geq \delta$ então, em particular, temos que cada δ -pareamento entre $\mathcal{B}_F(p)$ e $\mathcal{B}_G(\varphi(p))$ e entre $\mathcal{B}_G(q)$ e $\mathcal{B}_F(\psi(q))$ são ε -pareamentos. Portanto existe ε -pareamento entre \mathcal{B}_F e \mathcal{B}_G .

Por outro lado, se tivermos que $\varepsilon \leq \delta$ consideramos as funções $\hat{\varphi}$ e $\hat{\psi}$ como sendo os δ -interleaving obtido pela composição de φ e ψ com suas respectivas merge trees

⁴Lembrando que: se $A \subset B$ temos $\inf A \geq \inf B$.

transladadas em $\delta - \varepsilon$. Afirmamos que existem δ -pareamentos entre todos os pares $\mathcal{B}_F(p)$ e $\mathcal{B}_G(\hat{\varphi}(p))$ e entre $\mathcal{B}_G(q)$ e $\mathcal{B}_F(\hat{\psi}(q))$. De fato, considerando a projeção canônica $\pi_F : M_F \rightarrow \mathbb{R}$ e tomamos $H = \pi_F(p)$ temos que

$$\mathcal{B}_G(\hat{\varphi}(p)) = \text{trunc}_{H+\delta-\varepsilon}(\mathcal{B}_G(\varphi(p))).$$

Do Lema 3.2.9 temos que existe δ -pareamento de $\mathcal{B}_F(p)$ e $\mathcal{B}_G(\hat{\varphi}(p))$. O mesmo argumento se repete para $\mathcal{B}_G(q)$ e $\mathcal{B}_F(\hat{\psi}(q))$ e o resultado segue. ■

A próxima proposição nos garante a existência de um (ε, δ) -pareamento na árvore toda se apenas tivermos checado o δ -pareamento em suas folhas. Isso é especialmente relevante para o aspecto computacional da teoria que estamos desenvolvendo.

Proposição 3.3.9. *Sejam $\mathcal{B}_F : (M_F, \leq) \rightarrow \mathbf{Barcodes}$ e $\mathcal{B}_G : (M_G, \leq) \rightarrow \mathbf{Barcodes}$ merge trees decoradas por folhas, φ e ψ ε -interleaving das merge trees generalizadas subjacentes M_F e M_G onde para cada folha $v \in M_F$ e cada folha $w \in M_G$ tivermos um δ -pareamento entre $\mathcal{B}_F(v)$ e $\mathcal{B}_G(\varphi(v))$ e um δ -pareamento entre $\mathcal{B}_G(w)$ e $\mathcal{B}_F(\psi(w))$. Então existe um (ε, δ) -pareamento da merge tree decorada por barcodes toda.*

Demonstração. Seja $p \in M_F$ um ponto arbitrário, escolha uma folha $v \in M_F$ com $v \leq p$ e folha $w \in M_G$ tal que $w \leq \varphi(p)$, isto é, v comparável à p e w à $\varphi(p)$. Consideramos a projeção $\pi_F : M_F \rightarrow \mathbb{R}$, como v é comparável à p , então truncando na altura de p obtemos

$$\mathcal{B}_F(p) = \text{trunc}_{\pi_F(p)}(\mathcal{B}_F(v)).$$

Analogamente, como φ e ψ são ε -interleaved temos pelos mesmo argumento que

$$\mathcal{B}_G(\varphi(p)) = \text{trunc}_{\pi_F(p)+\varepsilon}(\mathcal{B}_G(w)).$$

É obvio que um δ -pareamento entre $\mathcal{B}_F(p)$ e $\mathcal{B}_F(\varphi(p))$ induz um δ -pareamento entre os barcodes truncados. Um argumento análogo se aplica para o par $q \in M_G$ e $\psi(q) \in M_F$. ■

Antes de apresentarmos o próximo grande resultado sobre estabilidade, precisamos dar um contexto e algumas definições para que a discussão subsequente faça sentido.

Definição 3.3.10. *Um \mathbb{R} -espaço é um espaço topológico X dotado de uma função contínua $f : X \rightarrow \mathbb{R}$. Usaremos a notação X_f para representar o \mathbb{R} -espaço formado pelo espaço topológico X munido da função contínua $f : X \rightarrow \mathbb{R}$. Definimos a categoria \mathbf{H}_0 cujos os objetos são os \mathbb{R} -espaços e os morfismos são todos os homeomorfismos entre os espaços topológicos.*

Definição 3.3.11. *Sejam X_f e Y_g dois \mathbb{R} -espaços tais que X é homeomorfo a Y . A **distância natural** d_{NP} entre X_f e Y_g fica definida como*

$$d_{NP}(X_f, Y_g) := \inf_h \|f - g \circ h\|_\infty,$$

onde $h : X \rightarrow Y$ é um morfismo da categoria \mathbf{H}_0 . Caso X e Y não sejam homeomorfos, definimos $d_{NP}(X_f, Y_g) := \infty$. Destacamos que a distância natural na verdade é uma pseudo-métrica estendida.

A distância natural é interessante para comparar espaços homeomorfos, entretanto em TDA corriqueiramente fica difícil detectar espaços que sejam de fato homeomorfos. Desta forma, apresentamos a seguir uma métrica que serve para comparar espaços com o mesmo tipo de homotopia, tornando a comparação entre os espaços um pouco mais geral.

Para ficar mais compacta a escrita do próximo resultado, considere a seguinte notação

Notação: Dado um \mathbb{R} -espaço X_f , denotamos por $X_{\leq s}$ a imagem inversa $f^{-1}(-\infty, s]$.

Definição 3.3.12. *Um ε -interleaving de \mathbb{R} -espaços entre X_f e Y_g é um par de funções contínuas $\varphi : X \rightarrow Y$ e $\psi : Y \rightarrow X$ junto com homotopias*

$$H_X : X \times I \rightarrow X \quad e \quad H_Y : Y \times I \rightarrow Y,$$

onde $I = [0, 1]$, conectando 1_X e 1_Y com $\psi \circ \varphi$ e $\varphi \circ \psi$ respectivamente. Além disso, pedimos que valham as seguintes propriedades para φ, ψ, H_X e H_Y :

1. $\varphi(X_{\leq s}) \subset Y_{\leq s+\varepsilon}$ para $s \in \mathbb{R}$;
2. $\psi(Y_{\leq s}) \subset X_{\leq s+\varepsilon}$ para $s \in \mathbb{R}$;
3. $f \circ H_X(x, t) \leq f(x) + 2\varepsilon$, para todo $t \in I$ e $x \in X$;
4. $g \circ H_Y(y, t) \leq g(y) + 2\varepsilon$, para todo $t \in I$ e $y \in Y$.

A **distância de tipo homotópico persistente** entre \mathbb{R} -espaços X_f e Y_g é definida como sendo

$$\delta_I(X_f, Y_g) := \inf \{ \varepsilon \geq 0 \mid X_f \text{ e } Y_g \text{ são } \varepsilon\text{-interleaved} \}.$$

Caso não exista tal ε -interleaving para $\varepsilon \geq 0$, então definimos $\delta_I(X_f, Y_g) := \infty$.

Lema 3.3.13. *Considere X_f, Y_g dois \mathbb{R} -espaços homeomorfos, isto é, X homeomorfo a Y . Então, vale a seguinte propriedade*

$$\delta_I(X_f, Y_g) \leq d_{NP}(X_f, Y_g).$$

Caso tenhamos $X = Y$, segue que

$$\delta_I(X_f, Y_g) \leq \|f - g\|_\infty.$$

A demonstração que δ_I é de fato uma pseudo-métrica estendida pode ser encontrado em [17], juntamente com a demonstração do Lema 3.3.13. Destacamos também que a abordagem proposta em [17] é um pouco mais geral do que escolhemos fazer aqui, uma vez que os autores tratam de espaços topológicos munidos de funções contínuas com imagem em \mathbb{R}^n , $n \geq 1$. Para o que estamos dispostos em fazer nesse texto, tomar $n = 1$ já é mais do que o necessário.

Outra observação que fazemos é que se vale a primeira parte do Lema 3.3.13, demonstrar o segundo caso se torna trivial, uma vez que os morfismos de \mathbf{H}_0 são os homeomorfismos entre os objetos, temos portanto

$$\inf_h \|f - g \circ h\|_\infty = \|f - g \circ 1_X\|_\infty = \|f - g\|_\infty.$$

O próximo resultado ordena as métricas até aqui apresentadas e ordenando os teoremas de estabilidade para as merge trees decoradas.

Teorema 3.3.14 (Hierarquia dos Teoremas de Estabilidade). *Sejam $f : X \rightarrow \mathbb{R}$ e $g : Y \rightarrow \mathbb{R}$ funções contínuas cujos sublevel-sets são localmente conexos, F e G suas respectivas filtrações de sublevel-set, vistas como espaços de persistência. Denotando as merge trees decoradas categorial de grau homológico n por \tilde{F}_n e \tilde{G}_n e as suas merge trees decoradas por barcodes associadas por \mathcal{BF}_n e \mathcal{BG}_n , temos*

$$\theta_I(M_F, M_G) \leq d_B(\mathcal{BF}_n, \mathcal{BG}_n) \leq d_I(\tilde{F}_n, \tilde{G}_n) \leq \delta_I(X_f, Y_g).$$

Além disso, se $X = Y$ temos

$$\delta_I(X_f, Y_g) \leq \|f - g\|_\infty.$$

Demonstração. Primeiramente perceba que temos

$$\theta_I(M_F, M_G) \leq d_B(\mathcal{BF}_n, \mathcal{BG}_n)$$

da definição de distância decorada bottleneck, uma vez que pedir um (ε, δ) -pareamento entre \mathcal{BF}_n e \mathcal{BG}_n implica que M_F e M_G sejam ε -interleaved.

Considere $\tilde{\varphi} : \tilde{F}_n \Rightarrow \tilde{G}_n \circ T_\varepsilon$ e $\tilde{\psi} : \tilde{G}_n \Rightarrow \tilde{F}_n \circ T_\varepsilon$ ε -interleaving qualquer de uma merge tree decorada

$$\begin{array}{ccccc}
 (\mathbb{R}, \leq) & \xrightarrow{T_\varepsilon} & (\mathbb{R}, \leq) & \xrightarrow{T_\varepsilon} & (\mathbb{R}, \leq) \\
 \tilde{F}_n \downarrow & \tilde{\varphi} \Downarrow & \tilde{G}_n \downarrow & \tilde{\psi} \Downarrow & \tilde{F}_n \downarrow \\
 \mathbf{pVec} & \xrightarrow{1_{\mathbf{pVec}}} & \mathbf{pVec} & \xrightarrow{1_{\mathbf{pVec}}} & \mathbf{pVec}
 \end{array}$$

Agora aplicando o functor *dom* nas transformações naturais $\tilde{\varphi}$ e $\tilde{\psi}$ obtemos ε -interleaving φ e ψ dos conjuntos persistentes subjacentes. De fato, lembramos que

$$\begin{aligned}
 \tilde{F}_n &:= H_n \circ \tilde{F} : (\mathbb{R}, \leq) \rightarrow \mathbf{pVec} \\
 & s \mapsto I(s)
 \end{aligned}$$

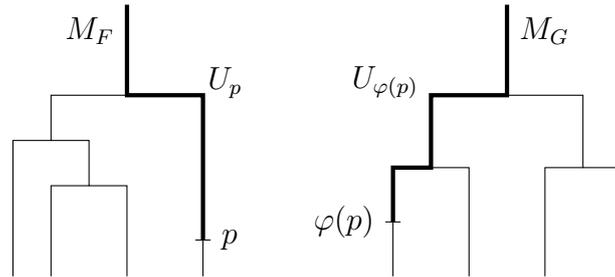
onde

$$\begin{aligned}
 I(s) &: \pi_0 \circ F(s) \rightarrow \mathbf{Vec} \\
 (\pi_0 \circ F)_i &\mapsto H_n(F(s)_i),
 \end{aligned}$$

com $(\pi_0 \circ F)_i$ sendo a i -ésima componente de $\pi_0 \circ F$. Aplicando *dom* obtemos $\pi_0 \circ F$, podemos prosseguir de maneira análoga com \tilde{G}_n . Logo pela Proposição 2.2.15, tomando $H = \text{dom}$, temos

$$\theta_I(M_F, M_G) = d_I(\pi_0 \circ F, \pi_0 \circ G) = d_I(\text{dom}(\tilde{F}_n), \text{dom}(\tilde{G}_n)) \leq d_I(\tilde{F}_n, \tilde{G}_n).$$

Note que para $p \in M_F$, restringindo \tilde{F}_n ao conjunto superior de p , temos que $\tilde{\varphi} : \tilde{F}_n \Rightarrow \tilde{G}_n \circ T_\varepsilon$ especifica um morfismo entre $\tilde{F}_n|_{U_p}$ e $(\tilde{G}_n \circ T_\varepsilon)|_{U_{\varphi(p)}}$.



Analogamente, como $\psi \circ \varphi(p)$ tem que ser uma translação em 2ε em $p \in M_F$, concluímos assim que $\tilde{F}_n|_{U_p}$ e $((\tilde{G}_n \circ T_\varepsilon) \circ \varphi)|_{U_p}$ são ε -interleaved como \mathbb{R} -módulos. Aplicando o Teorema da Estabilidade Algébrica, Teorema 2.2.17, temos um ε -pareamento entre \mathcal{BF}_n e $\mathcal{BG}_n(\varphi(p))$ para todo $p \in M_F$. De maneira análoga temos um ε -pareamento entre $\mathcal{BF}_n(\psi(q))$ e $\mathcal{BG}_n(q)$ para todo $q \in M_G$. Assim, obtemos que

$$d_B(\mathcal{BF}_n, \mathcal{BG}_n) \leq d_I(\tilde{F}_n, \tilde{G}_n).$$

Só nos resta mostrar que $d_I(\tilde{F}_n, \tilde{G}_n) \leq \delta_I(X_f, Y_g)$. O caso em que $\delta_I(X_f, Y_g) = \infty$ é trivial, então consideramos o caso em que $\varepsilon > \delta_I(X_f, Y_g)$. Segue da definição que existe um ε -interleaving de \mathbb{R} -espaços X_f e Y_g , em particular, para cada $t \in \mathbb{R}$ temos que a função φ se restringe a uma função contínua entre as filtrações de sublevel-sets

$$\Phi : X_{\leq t} \rightarrow Y_{\leq t+\varepsilon}$$

Como φ é uma função contínua definida globalmente em X , então ela especifica uma transformação natural entre os espaços de persistência $\varphi : F \Rightarrow G \circ T_\varepsilon$ com o seguinte diagrama comutativo

$$\begin{array}{ccc} F(t) & \xrightarrow{F(t \leq s)} & F(s) \\ \varphi_t \downarrow & \circlearrowleft & \downarrow \varphi_s \\ G \circ T_\varepsilon(t) & \xrightarrow{G \circ T_\varepsilon(t \leq s)} & G \circ T_\varepsilon(s) \end{array}$$

Mais ainda, como a imagem de um espaço conexo por caminhos por uma função contínua é conexo por caminhos, temos que a transformação natural φ define uma transformação natural entre os espaços persistentes que são parametrizados por suas componentes conexas por caminhos $\tilde{\varphi} : \tilde{F} \Rightarrow \tilde{G} \circ T_\varepsilon$. Prosseguindo de forma análoga, podemos mostrar que $\psi : Y \rightarrow X$ se restringe a uma função contínua $\Psi : Y_{\leq t} \rightarrow X_{\leq t+\varepsilon}$ e ψ define uma transformação natural $\tilde{\psi} : \tilde{G} \Rightarrow \tilde{F} \circ T_\varepsilon$.

Perceba que isso não significa que $\tilde{\varphi}$ e $\tilde{\psi}$ definem um interleaving entre os funtores \tilde{F} e \tilde{G} , pois isso pede que a composta $\psi(t + \varepsilon) \circ \varphi(t)$ seja igual a inclusão $i : X_{\leq t} \hookrightarrow X_{t+2\varepsilon}$ para todo $t \in \mathbb{R}$, o que não ocorre necessariamente. Por outro lado, o item (3) da Definição 3.3.12 garante que $\Psi \circ \Phi$ restrita a uma aplicação seja homotópica a $i : X_{\leq t} \hookrightarrow X_{t+2\varepsilon}$ para todo t . Deste fato e do fato que aplicações homotópicas tem o mesmo push-forward em homologia podemos concluir que as transformações naturais $\tilde{\varphi}$ e $\tilde{\psi}$ definem um ε -interleaving nas componentes de homologia persistente dos módulos \tilde{F}_n e \tilde{G}_n .

Por fim, se $X = Y$, pelo Lema 3.3.13 temos

$$\delta_I(X_f, X_g) \leq \|f - g\|_\infty.$$

■

O próximo resultado nos garante que a teoria que desenvolvemos das merge trees decoradas é de fato mais sensível que a teoria da homologia persistente usual apresentada em [14].

Teorema 3.3.15. *Para espaços de persistência localmente conexos $F, G : (\mathbb{R}, \leq) \rightarrow \mathbf{Top}_{lc}$, temos*

$$d_I(H_n \circ F, H_n \circ G) \leq d_I(\tilde{F}_n, \tilde{G}_n).$$

Demonstração. Aplicando o funtor cop em \tilde{F}_n temos ⁵

$$cop \circ \tilde{F}_n(s) = \bigoplus_{i \in \pi^{-1}(s)} H_n(F(s)_i),$$

uma vez que o coproduto na categoria \mathbf{Vec} é a soma direta. Por outro lado, pelo axioma da aditividade da homologia, temos

$$\bigoplus_{i \in \pi^{-1}(s)} H_n(F(s)_i) \cong H_n \left(\bigsqcup_{i \in \pi^{-1}(s)} F(s)_i \right).$$

Perceba que como $F(s)$ é localmente conexo por hipótese e por estarmos usando $H_n(\bigsqcup F(s)_i)$ em cada componente conexa, isso tem que ser naturalmente isomorfo à homologia de $F(s)$, isto é,

$$H_n \left(\bigsqcup_{i \in \pi^{-1}(s)} F(s)_i \right) \cong H_n \circ F(s).$$

O mesmo argumento sem muitas variações pode ser usado para o funtor G . Como módulos naturalmente isomorfos possuem a mesma distância interleaving, então novamente pela Proposição 2.2.15, tomando o funtor H como sendo o funtor cop , temos

$$d_I(H_n \circ F, H_n \circ G) = d_I(cop \circ \tilde{F}_n, cop \circ \tilde{G}_n) \leq d_I(\tilde{F}_n, \tilde{G}_n).$$

■

3.4 Decomponibilidade das Merge Trees

Nesta seção trabalharemos em prol de poder decompor as merge trees decoradas, assim como feito no Capítulo 1 no contexto de módulos de persistência. Destacamos que essa construção será mais trabalhosa do que a encontrada no Capítulo 1, pois no caso das merge trees temos um conjunto parcialmente ordenado. Desta forma, iremos nos concentrar em merge trees decoradas que se decompõem em módulos com suporte totalmente ordenado na merge tree.

Definição 3.4.1. *Uma merge tree decorada concreta $\mathcal{F} : M_F \rightarrow \mathbf{Vec}_{\mathbb{K}}$ é dita ser **decom-***

⁵Perceba que $\pi^{-1}(s)$ representa o conjunto de rótulos das componentes conexas do espaço de persistência no momento s .

ponível em intervalo real se ela pode ser expressa como soma direta

$$\mathcal{F} \cong \bigoplus_i \mathcal{F}_i,$$

onde cada $\mathcal{F}_i : M_F \rightarrow \mathbf{Vec}_{\mathbb{K}}$ é indecomponível e tem suporte totalmente ordenado.

Do fato que estamos usando a soma direta na definição anterior, fica aparente a restrição de tal definição. Boa parte dos objetos que temos em mente, possuem árvore que não é decomponível em intervalo real. Vejamos um exemplo.

Exemplo 3.4.2. *Considere o espaço apresentado na Figura 3.7.*

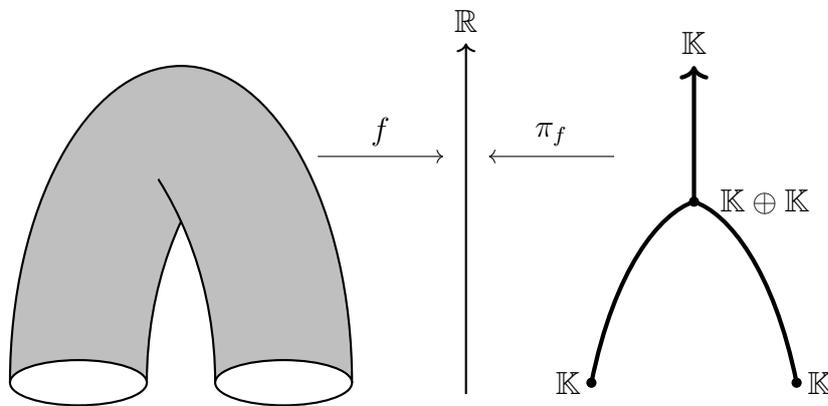


Figura 3.7: Exemplo de espaço que não é decomponível em intervalo real. A figura representa um cilindro deformado, ou ainda, um toro seccionado no meio.

Usamos a filtração de sublevel-set, podemos ver que a cada passo da construção temos duas componentes conexas, que eventualmente se conectam em uma única componente. Utilizando as merge trees concretas para descrever esse procedimento, temos

$$\mathcal{F}_i : (M_F, \leq) \rightarrow \mathbf{vec}_{\mathbb{K}}, \quad \text{com } i = 1, 2.$$

Na Figura 3.7 a árvore na direita representa justamente as componentes conexas e seus respectivos grupos de homologia. Agora este espaço não pode ser decomponível em intervalo real pois os espaços vetoriais que tratamos eventualmente tem intersecção diferente do vazio, o que impede de fazermos a soma direta dos espaços vetoriais.

Vamos estudar quando ocorre de termos a possibilidade de decompor uma merge tree decorada.

Definição 3.4.3. *Seja V um espaço vetorial qualquer dado. Uma coleção de subespaços*

vetoriais V_1, \dots, V_m de V é dita ser **independente** se para todo i ,

$$V_i \cap \left(\sum_{j \neq i} V_j \right) = 0.$$

Sobre essa hipótese temos que

$$V_1 + \dots + V_m = V_1 \oplus \dots \oplus V_m.$$

Seja (M_F, \leq) uma merge tree generalizada e $\mathcal{F} : (M_F, \leq) \rightarrow \mathbf{Vec}$ um módulo de árvore. Dizemos que \mathcal{F} é **não torcida** se toda vez que tivermos um conjunto de elementos incomparáveis entre si $p_1, \dots, p_n \in M_F$ e qualquer limitante superior $p \in M_F$ destes pontos, a seguinte coleção de subespaços vetoriais

$$\text{Im}\mathcal{F}(p_1 \leq p), \dots, \text{Im}\mathcal{F}(p_n \leq p)$$

for independente.

Definição 3.4.4. Um conjunto de persistência $S : (\mathbb{R}, \leq) \rightarrow \mathbf{Sets}$ é dito ser **construtível** se existir uma coleção $\tau = \{t_0 < t_1 < \dots < t_n\}$ tal que

i) $S(t) = \emptyset$, $t < t_0$

ii) $S(t) = \{\star\}$ para $t > t_n$

iii) $S(t \leq s)$ é uma bijeção para todo par $t \leq s \subset [t_i, t_{i+1})$.

Assumindo essas condições, o display poset associado (\mathcal{S}, \leq) é dito ser **tame**.

Podemos reescrever esta definição sem muitos esforços, e assim torná-la mais versátil.

Proposição 3.4.5. Seja (M_F, \leq) uma merge tree generalizada tame, isto é, o conjunto persistente $\pi_0 \circ F$ é construtível. Temos as seguintes consequências:

1. Qualquer cadeia maximal $C \subset M_F$ tem mínimo, ou seja, folhas.
2. Toda folha tem único menor limitante superior $p_\infty = (\star, t_n)$, que chamaremos de **raiz**.
3. Qualquer módulo de árvore de dimensão pontualmente finita $\mathcal{F} : (M_F, \leq) \rightarrow \mathbf{vec}$ é levado através da projeção natural $\pi_F : M_F \rightarrow \mathbb{R}$ em um \mathbb{R} -módulo de dimensão pontualmente finita $(\pi_F)_*\mathcal{F} : (\mathbb{R}, \leq) \rightarrow \mathbf{vec}$, que é definido pontualmente por

$$(\pi_F)_*\mathcal{F}(s) := \bigoplus_{p \in \pi^{-1}(s)} \mathcal{F}(s).$$

Teorema 3.4.6. *Seja (M_F, \leq) uma merge tree generalizada tame. Um módulo de árvore de dimensão pontualmente finita $\mathcal{F} : (M_F, \leq) \rightarrow \mathbf{vec}$ é decomponível em intervalo real se, e somente se, for não torcido.*

Demonstração. (\Rightarrow) Direto, uma vez que um módulo de intervalo real satisfaz a condição de ser não torcido e essa propriedade é herdada pela soma direta.

(\Leftarrow) A demonstração se dá por indução no número de folhas da merge tree generalizada (M_F, \leq) . Se $n = 1$, então (M_F, \leq) é totalmente ordenada, o que implica que esse módulo de árvore é não torcido, uma vez que todo elemento de M_F é comparável e assim $\text{Im}\mathcal{F}(p_i \leq p)$ é independente por definição.

Suponha que vale para $n = m$, queremos verificar tal propriedade para $n = m + 1$. Seja (M_F, \leq) uma merge tree generalizada com $m + 1$ folhas e uma raiz em p_∞ . Escrevemos M_F como sendo a união de dois subconjuntos M_1 e M_2 , onde cada um tem ao menos uma folha e $M_1 \cap M_2 = U_{p_\infty}$, que é totalmente ordenado. Veja a Figura 3.8 para um auxílio visual da construção desejada.

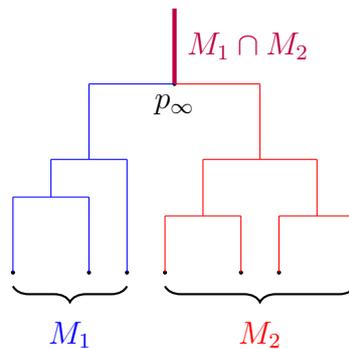


Figura 3.8: Um exemplo de árvore sendo escrita como a união de M_1 e M_2 .

Associado a qualquer módulo de árvore de dimensão pontualmente finita $\mathcal{F} : (M_F, \leq) \rightarrow \mathbf{vec}$ e uma decomposição $M_F = M_1 \cup M_2$ temos dois submódulos naturais \mathcal{F}_1 e \mathcal{F}_2 , definidos da seguinte maneira para $i \neq j$, $i, j \in \{1, 2\}$:

(1) Se $p \in M_i$ e $p < p_\infty$, então $\mathcal{F}_i(p) := \mathcal{F}(p)$.

(2) Se $p \geq p_\infty$, então

$$\mathcal{F}_i(p) := \sum_{q \in M_i | q < p_\infty} \text{Im}\mathcal{F}(q < p).$$

(3) Se $p \in M_j \setminus M_i$ então $\mathcal{F}(p) = 0$.

Como \mathcal{F}_i tem $1 \leq k \leq m$ folhas, então pela hipótese de indução, temos que \mathcal{F}_i é não torcido para $i = 1, 2$ e portanto decomponível em intervalo real, isto é,

$$\mathcal{F}_1 \cong \bigoplus_{\alpha \in I_1} \mathcal{F}_\alpha \quad \text{e} \quad \mathcal{F}_2 \cong \bigoplus_{\beta \in I_2} \mathcal{F}_\beta,$$

onde \mathcal{F}_α e \mathcal{F}_β são suportados em subconjuntos totalmente ordenados em M_1 e M_2 respectivamente.

Pela Proposição 3.4.5 $(\pi_F)_*\mathcal{F}$, $(\pi_F)_*\mathcal{F}_1$ e $(\pi_F)_*\mathcal{F}_2$ são \mathbb{R} -módulos de dimensão pontualmente finita. Mais ainda, como

$$(\pi_F)_* : \mathbf{Fun}(M_F, \mathbf{vec}) \rightarrow \mathbf{Fun}(\mathbb{R}, \mathbf{vec})$$

é um funtor aditivo por construção⁶, sabemos que

$$(\pi_F)_*\mathcal{F}_1 \cong (\pi_F)_* \bigoplus_{\alpha \in I_1} \mathcal{F}_\alpha \cong \bigoplus_{\alpha \in I_1} (\pi_F)_*\mathcal{F}_\alpha,$$

que concordam, a menos de permutações, com a decomposição de Remak de $(\pi_F)_*\mathcal{F}_1$, garantida pelo Teorema 1.2.8.⁷ Por um argumento análogo o mesmo funciona para $(\pi_F)_*\mathcal{F}_2$.

Isso implica que podemos coletar os módulos de intervalo da decomposição de Remak em três termos

$$(\pi_F)_*\mathcal{F} \cong F_1 \oplus F_2 \oplus F_3, \quad \text{onde } F_1 := (\pi_F)_*\mathcal{F}_1 \text{ e } F_2 := (\pi_F)_*\mathcal{F}_2.$$

Afirmamos que F_3 é a soma direta dos módulos de intervalos reais que nascem em $\pi_F(p_\infty)$ ou depois de $\pi_F(p_\infty)$. De fato, isso vem do fato que para $t < \pi_F(p_\infty)$ o conjunto $\pi_F^{-1}(t)$ pode ser particionado em pontos em M_1 ou (*exclusivo*) M_2 pois $M_1 \cap M_2 = U_{p_\infty}$, o que implica que $\pi_F^{-1}(t) \cap M_1 \cap M_2 = \emptyset$. Assim, para $t < \pi_F(p_\infty)$ temos

$$(\pi_F)_*\mathcal{F}(t) \cong (\pi_F)_*\mathcal{F}_1(t) \oplus (\pi_F)_*\mathcal{F}_2(t).$$

Assim, usamos esse fato para definir

$$\mathcal{F}_3 := (\pi_F)^*F_3.$$

Nos resta apenas mostrar que $\mathcal{F} = \mathcal{F}_1 \oplus \mathcal{F}_2 \oplus \mathcal{F}_3$. Essa decomposição obviamente funciona para o M_F -módulo quando restrito aos pontos $p \leq p_\infty$. Como \mathcal{F}_i , para $i = 1, 2, 3$ são submódulos de \mathcal{F} , basta mostrar que quando tivermos $p \geq p_\infty$ e $v_i \in \mathcal{F}_i(p)$ escolhidos tal que $v_1 + v_2 + v_3 = 0$, temos que $v_i = 0$.

Consideremos $v_1 + v_2 + v_3 = 0$, então escrevendo $v_3 = -(v_1 + v_2)$ temos

$$v_3 \in (\mathcal{F}_1(p) \oplus \mathcal{F}_2(p)) \cap \mathcal{F}_3(p)$$

⁶Lembrando que segue da Proposição 2.1.34 que $\mathbf{Fun}(M_F, \mathbf{vec})$ e $\mathbf{Fun}(\mathbb{R}, \mathbf{vec})$ são categorias abelianas.

⁷O Teorema 1.2.8 que está enunciado no Capítulo 1 usa explicitamente módulos de intervalos. Porém existe uma versão mais geral deste fato que garante a unicidade a menos de permutação dos módulos. Veja [20] página 115.

o que implica que $v_3 = 0$. Além disso, se v_1 e v_2 são imagens de w_1 e w_2 através das funções $\mathcal{F}_1(q_1 < p)$ e $\mathcal{F}_2(q_2 < p)$, respectivamente, então se utilizando da hipótese de indução, a condição de ser não torcida implica que v_1 e v_2 são independentes e portanto iguais a 0. ■

Usaremos esses módulos de árvores que podem ser decompostos em intervalos reais para estudar a transformada de barcode restrita a esses objetos. Como veremos a seguir, quando tivermos essa estrutura, conseguiremos garantir a injetividade da transformada de barcode. Porém, antes de enunciar tal resultado, façamos algumas considerações e convenções importantes.

Lembramos que na definição de módulo de intervalo (vide Definição 1.2.3), tínhamos $T \subset \mathbb{R}$ e $J \subset T$ um intervalo de T . Gostaríamos de colocar essa ideia de módulo de intervalos no contexto das merge trees, assim prosseguimos da seguinte maneira.

Definição 3.4.7. *Sejam (M_F, \leq) uma merge tree generalizada e $J \subset M_F$ um conjunto totalmente ordenado de (M_F, \leq) . Denotaremos por $\mathbb{I}_J : (M_F, \leq) \rightarrow \mathbf{vec}_{\mathbb{K}}$ o funtor que atribui \mathbb{K} para pontos $p \in J$ e 0 nos demais e para pares $p \leq q \in J$ temos o morfismo identidade.*

Estamos escrevendo o funtor definido acima como \mathbb{I}_J para ressaltar a correspondência que ele tem com os módulos de intervalo, no contexto do Capítulo 1.

Se a merge tree decorada $\mathcal{F} : M_F \rightarrow \mathbf{Vec}$ for decomponível em intervalo real, então podemos expressá-la da seguinte maneira

$$\mathcal{F} \cong \bigoplus_{J \in \mathcal{R}(\mathcal{F})} \mathbb{I}_J,$$

onde $\mathcal{R}(\mathcal{F}) := \{(J, m_J)\}$ é o conjunto de todos os multisets que aparecem na decomposição de Remak da merge tree \mathcal{F} .

Proposição 3.4.8. *Suponha que o módulo de árvore $\mathcal{F} : (M_F, \leq) \rightarrow \mathbf{vec}$ seja decomponível em intervalo real. Então a merge tree decorada por barcode associada*

$$\mathcal{BF} : M_F \rightarrow \mathbf{Barcodes}$$

é determinada em cada ponto por

$$\mathcal{BF}(p) := \{(U_p \cap J, m_J) \mid (J, m_J) \in \mathcal{R}(\mathcal{F})\}, \quad (3.1)$$

onde U_p é o conjunto superior de $p \in M_F$. Mais ainda, quando M_F for tame a merge tree decorada por barcodes associada é completamente descrita como a união disjunta dos

barcodes vistos a partir de cada folha

$$\mathcal{BF} = \bigsqcup_{v \in L(M_F)} \mathcal{BF}(v),$$

onde $L(M_F)$ representa o conjunto de folhas de M_F e $\mathcal{BF}(v)$ é determinado como em (3.1). Note que cada barra (I, m_I) no barcode vista a partir de v pode ser expressa como

$$(I, m_I, v) = \sum_{J \in \mathcal{R}(\mathcal{F})} U_v \cap (J, m_J), \quad \text{onde } U_v \cap J = I.$$

Informalmente, podemos escrever isso como $m_I = m_{J_1} + \cdots + m_{J_k}$.

Definição 3.4.9. Sejam $\mathcal{BF} : M_F \rightarrow \mathbf{Barcodes}$ e $\mathcal{BG} : M_G \rightarrow \mathbf{Barcodes}$ duas merge trees decoradas por barcodes associadas aos módulos de árvores $\mathcal{F} : (M_F, \leq) \rightarrow \mathbf{vec}$ e $\mathcal{G} : (M_G, \leq) \rightarrow \mathbf{vec}$ respectivamente. Dizemos que as merge trees decoradas por barcodes são **isomorfas** se existir isomorfismo entre suas merge trees generalizadas $\varphi : M_F \rightarrow M_G$ e $\psi : M_G \rightarrow M_F$ tal que

$$\mathcal{BF}(x) = \mathcal{BG}(\varphi(x)) \quad \text{e} \quad \mathcal{BG}(x') = \mathcal{BF}(\psi(x')),$$

para todo $x \in M_F$ e $x' \in M_G$.

Teorema 3.4.10 (Injetividade da Transformada de Barcode).

Sejam $\mathcal{F} : (M, \leq) \rightarrow \mathbf{vec}$ e $\mathcal{F}' : (M', \leq) \rightarrow \mathbf{vec}$ dois módulos de árvores decomponíveis em intervalos reais, com uma quantidade finita de intervalos em suas decomposições de Remak. Se suas merge trees decoradas por barcodes associadas $\mathcal{BF} : M \rightarrow \mathbf{Barcodes}$ e $\mathcal{BF}' : M' \rightarrow \mathbf{Barcodes}$ forem isomorfas, então temos $\mathcal{F} \cong \mathcal{F}'$.

Demonstração. Considere $\varphi : M \rightarrow M'$ e $\psi : M' \rightarrow M$ um isomorfismo entre as merge trees generalizadas. Para provar o teorema precisamos mostrar que $\mathcal{F} \cong \mathcal{F}' \circ \varphi$ e $\mathcal{F}' \cong \mathcal{F} \circ \psi$. Os dois argumentos seriam análogos sobre as duas merge trees generalizadas, então é suficiente considerar $M = M'$ e φ e ψ sendo ambas a identidade. Assim, queremos mostrar que $\mathcal{BF} = \mathcal{BF}'$ implica em $\mathcal{F} \cong \mathcal{F}'$ como M -módulos.

O argumento segue mostrando que cada intervalo real (J, m_J) que aparece na decomposição de Remak de \mathcal{F} são exatamente os mesmos, a menos de permutações, que aparecem na decomposição de Remak de \mathcal{F}' . Isso nos dará o isomorfismo desejado.

Vamos considerar o intervalo mais velho (I, m_I, v) em \mathcal{BF} , isto é, consideramos a projeção de I em \mathbb{R} através da projeção natural $\pi : M \rightarrow \mathbb{R}$ e temos $\inf(\pi(I)) \leq \inf(\pi(J))$ para qualquer (J, m_J, w) em \mathcal{BF} . Destacamos aqui que a existência do intervalo mais velho está garantida pela hipótese da finitude da quantidade de intervalos. Pela Pro-

posição 3.4.8 temos que (I, m_I, v) admite única expressão, a menos de permutações,

$$m_I = m_{J_1} + \cdots + m_{J_k}$$

para $J_j \in \mathcal{R}(\mathcal{F})$ com $J_j \cap U_v = I$.

Afirmamos que se (I, m_I, v) for o intervalo mais velho, então existe $(J_j, m_{J_j}) \in \mathcal{R}(\mathcal{F})$ com $(I, m_I, v) = (J_j, m_{J_j})$. De fato, se não existir único contribuidor, então isso fere a hipótese que (I, m_I, v) é o intervalo mais velho. Assim, J_j é o fator mais velho contribuindo na soma de m_I . Uma vez que a decoração por barcode associada a \mathcal{F}' é igual a \mathcal{F} , isto é, $\mathcal{BF} = \mathcal{BF}'$, assim (I, m_I, v) deve ser um intervalo mais velho em \mathcal{F}' e deve existir único $J_j \in \mathcal{R}(\mathcal{F}')$ tal que $(I, m_I, v) = (J_j, m_{J_j})$. Logo temos que

$$\mathcal{F} \cong \mathbb{I}_{J_j}^{m_{J_j}} \oplus \mathcal{G} \text{ e } \mathcal{F}' \cong \mathbb{I}_{J_j}^{m_{J_j}} \oplus \mathcal{G}',$$

com $\mathcal{BG} = \mathcal{BG}'$, pois $\mathcal{BF} = \mathcal{BF}'$ por hipótese. Repetimos esse processo para \mathcal{G} e \mathcal{G}' até que a merge tree decorada por barcode esteja vazia, esse processo tem fim justamente pela hipótese de finitude do teorema. ■

Temos a teoria necessária para traduzir a decomposição em módulos de intervalos apresentada no Capítulo 1 para esse contexto de Merge Trees Decoradas. Para isso, consideramos a seguinte definição.

Definição 3.4.11. *Considere $F : (\mathbb{R}, \leq) \rightarrow \mathbf{Top}$ uma filtração de um complexo simplicial e considere B seu barcode. Definimos o **simplexo de nascimento** σ em $I = [b, d)$, como sendo simplexo σ cujo nascimento no momento b na filtração designa a criação de um representante na classe de homologia em I .*

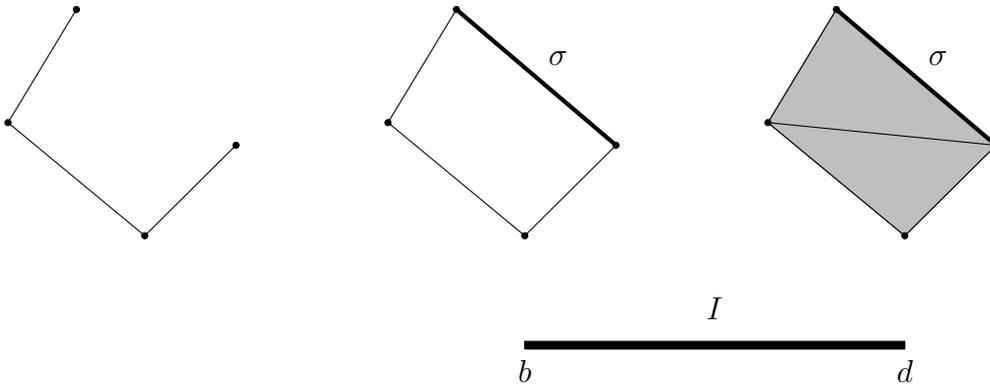


Figura 3.9: A figura acima representa um simplexo de nascimento σ . Perceba que a definição de simplexo de nascimento não pede nada sobre a morte da classe de homologia, adicionamos esta informação na figura acima apenas para melhor representação.

O ponto de nascimento de I é o único ponto $p \in M_F$ com $\pi_F(p) = b$, onde p é um ancestral de todos os vértices de σ . Seja ainda $q \in M_F$ o único ancestral de p tal que $\pi_F(q) = d$. Definimos assim o **levantamento de I** como sendo a merge tree decorada $\mathcal{I} : M_F \rightarrow \mathbf{vec}_{\mathbb{K}}$ definida por

$$\mathcal{I}(r) := \begin{cases} \mathbb{K}, & \text{se } p \leq r \leq q \\ 0, & \text{caso contrário.} \end{cases}$$

O **levantamento de B** é a merge tree decorada

$$\widehat{\mathcal{F}} := \bigoplus_{I \in B} \mathcal{I}.$$

Uma merge tree decorada é chamada de **merge tree decorada por levantamento** se for obtida como levantamento de algum barcode.

É relativamente direto determinar uma merge tree decorada por levantamento de um complexo simplicial filtrado. Entretanto precisamos garantir que essa nova merge tree é realmente isomorfa a merge tree decorada usual $\mathcal{F} : M_F \rightarrow \mathbf{vec}$. O resultado que apresentaremos a seguir será justamente para garantir quando tal isomorfismo existe, mas antes disso, apresentamos uma definição essencial.

Definição 3.4.12. *Seja $F : (\mathbb{R}, \leq) \rightarrow \mathbf{Top}$ uma filtração de um complexo simplicial com merge tree M_F e barcode de grau k denotado por B . Dizemos que F tem a **propriedade H_k -disjunção** se para qualquer par de intervalos $I = [b, d)$ e $I' = [b', d')$ em B com pontos de nascimento p e p' incomparáveis entre si, seus pontos de morte são menores que sua altura merge, isto é,*

$$\min \{d, d'\} < \text{merge}_F(p, p').$$

Proposição 3.4.13. *Se uma filtração de um complexo simplicial $F : (\mathbb{R}, \leq) \rightarrow \mathbf{Top}$ tem a propriedade H_k -disjunção, então a merge tree decorada por levantamento e a merge tree decorada concreta são isomorfas, isto é,*

$$\widehat{\mathcal{F}} \cong \mathcal{F}.$$

Demonstração. Seja $B = \{I_j \mid I_j = [b_j, d_j)\}_{j=1}^N$, com $N \in \mathbb{N}$ qualquer, o barcode de grau k de F . Suponhamos, por questão de praticidade, que todos os nascimentos no barcode sejam distintos e que os índices de tais nascimentos foram escolhidos da seguinte maneira $b_1 < b_2 < \dots < b_N$.

Para cada barra I_j escolhamos um ciclo representante c_j que gera a classe de homologia persistente relacionada a I_j . Em particular, escolhamos c_j tal que todos os simplexes estão

contidos na mesma componente conexa do simplexo de nascimento para I_j . Denotamos por $[c_j]$ a classe de homologia de c_j . Por fim, considere p_j como o ponto de nascimento de I_j .

A ideia geral é usar esses ciclos representativos para construir bases nos vários grupos de Homologia que concordam com a decomposição dos barcodes. Isso faz sentido, pois admitimos que estamos construindo grupos de homologia sobre um corpo fixado \mathbb{K} e portanto esses grupos de homologia são espaços vetoriais e sempre admitem uma base.

Considere

$$i_{b,d} : F(b) \hookrightarrow F(d)$$

a função inclusão para todo $b < d$. Denotaremos também por $i_{b,d}$ a função induzida em homologia $H_k(F(b)) \rightarrow H_k(F(d))$.

Construímos nossa base da seguinte forma:

1. Da definição de c_1 , temos que $\{v_1\}$, com $v_1 := [c_1]$, forma uma base para $H_k(F(b_1))$, já que $F(b_1)$ é o primeiro complexo simplicial na filtração cujo aparece uma classe de homologia.
2. Perceba que $\{[c_2], i_{b_1,b_2}(v_1)\}$ é um conjunto gerador para $H_k(F(b_2))$ é linearmente independente se

$$i_{b_1,b_2}(v_1) \neq 0,$$

pois caso contrário, c_2 iria representar a mesma barra de c_1 . Mais ainda, podemos escolher um escalar $\lambda_1^{(2)} \in \mathbb{K}$ tal que

$$v_2 := [c_2] + \lambda_1^{(2)} i_{b_1,b_2}(v_1)$$

tenha a seguinte propriedade

$$i_{b_2,d}(v_2) \notin \text{span} \{i_{b_1,d}(v_1)\}$$

para $d < d_2$. Em particular, $v_2 \notin \text{Ker}(i_{b_2,d})$ para $d < d_2$.

3. Suponha que já tenhamos definido v_1, \dots, v_{j-1} . Então

$$\{[c_j], i_{b_1,b_j}(v_1), \dots, i_{b_{j-1},b_j}(v_{j-1})\}$$

é um gerador para $H_k(F(b_j))$, que é linearmente independente uma vez que os zeros foram eliminados nos passos anteriores. Definimos então

$$v_j := [c_j] + \lambda_1^{(j)} i_{b_1,b_{j-1}}(v_1) + \dots + \lambda_{j-1}^{(j)} i_{b_{j-1},b_j}(v_{j-1}),$$

com escalares $\lambda_i^{(j)}$ escolhidos de tal forma que

$$i_{b_j,d}(v_j) \notin \text{span} \{i_{b_1,d}(v_1), \dots, i_{b_{j-1},d}(v_{j-1})\} \quad (3.2)$$

para todo $d < d_j$. Em particular, $v_j \notin \text{Ker}(i_{b_j,d})$ para $d < d_j$.

O índice de persistência do vetor v_j definido por esse processo representa exatamente o intervalo I_j da homologia persistente. Agora nos concentramos em mostrar o isomorfismo entre \mathcal{F} e $\widehat{\mathcal{F}}$.

Vamos verificar que \mathcal{F} é de fato decomponível em intervalo real. Para isso, basta mostrar que para cada v_j pode ser escolhido como uma combinação linear de classes com representantes que moram na mesma componente conexa dos simplexo de nascimento de I_j . Mostraremos isso por indução.

Para $j = 1$, temos que $v_1 = [c_1]$ vale trivialmente por construção. Suponha que vale para $j = k - 1$, queremos mostrar que isso ocorre para k . Ordenamos v_1, \dots, v_{k-1} em uma coleção de vetores v_{r_1}, \dots, v_{r_m} correspondendo à barras em B as quais nascem na mesma componente de c_k , isto é, os pontos de nascimento satisfazem $p_{r_1}, \dots, p_{r_m} \leq p_k$, e uma coleção de vetores v_{s_1}, \dots, v_{s_n} que não possuem essa propriedade. Então

$$v_k = [c_k] + \lambda_{r_1}^{(k)} i_{b_{r_1}, b_k}(v_{r_1}) + \dots + \lambda_{r_m}^{(k)} i_{b_{r_m}, b_k}(v_{r_m}) + \lambda_{s_1}^{(k)} i_{b_{s_1}, b_k}(v_{s_1}) + \dots + \lambda_{s_n}^{(k)} i_{b_{s_n}, b_k}(v_{s_n}).$$

Afirmamos que tomando $\lambda_{s_i}^{(k)} = 0$ para todo $i = 1, \dots, n$ temos um vetor v_k válido como definido no passo 3 da criação da nossa base, em particular, tal vetor satisfaz (3.2). De fato, se $d < d_k$ e supondo que

$$i_{b_k,d}([c_k] + \lambda_{r_1}^{(k)} i_{b_{r_1}, b_k}(v_{r_1}) + \dots + \lambda_{r_m}^{(k)} i_{b_{r_m}, b_k}(v_{r_m})) = \mu_1 i_{b_k,d}(v_{s_1}) + \dots + \mu_m i_{b_k,d}(v_{s_m}), \quad (3.3)$$

para alguns escalares μ_i . Primeiramente note que existe algum $v_{s_i} \notin \text{Ker}(i_{b_k,d})$, pois caso contrário o lado direito de (3.3) seria zero, implicando que $i_{b_k,d}(v_k) = 0$ o que é uma contradição, pois isso quer dizer que v_k morre antes de seu ponto de morte, já que $d < d_k$.

Assim, consideramos o seguinte número

$$d' := \min \{ \text{merge}_F(p_k, p_{s_i}) \mid i_{b_{s_i},d}(v_{s_i}) \neq 0 \}.$$

Por hipótese, temos que $F : (\mathbb{R}, \leq) \rightarrow \mathbf{Top}$ tem a propriedade de H_k -disjunção e como d é o mínimo de $\text{merge}_F(p_k, p_{s_i})$ então temos que $d_k < d'$ ou $d_{s_i} < d'$ para todo s_i .

Caso 1, $d_k < d'$: Temos que $d < d_k < d'$, então pela hipótese de indução existe um ciclo representante do lado esquerdo de (3.3) cujos vértices pertencem a mesma componente conexa de $i_{b_k,d}(c_k)$ e um ciclo representante do lado direito de (3.3) que pertence a uma componente conexa diferente de c_k . Mais ainda, os ciclos representantes são homólogos,

o que é um absurdo.

Caso 2, $d_{s_i} < d'$, $\forall s_i$: Existe s_i tal que $d < d_{s_i}$, uma vez que existe algum $v_{s_i} \notin \text{Ker}(i_{b_{s_i}}, d)$ e um argumento análogo ao caso anterior pode ser aplicado nos dando a contradição. ■

Por fim, podemos fazer mais um exemplo computacional para ficar claro como essa teoria se encaixa na prática.

Exemplo 3.4.14. *Iremos aplicar a teoria das merge trees decoradas ao Exemplo 3.2.2 para ver como a teoria toda se encaixa. Consideramos os seguintes datasets*

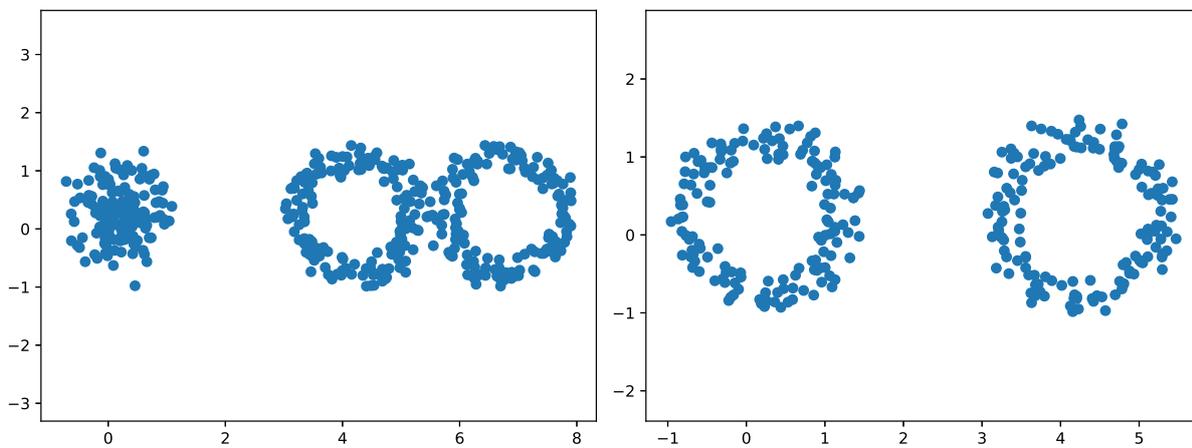


Figura 3.10: Datasets referentes ao exemplo teórico de merge tree decorada categoriais visto no Exemplo 3.2.2

Agora aplicamos a teoria usual de Persistência Homológica obtendo assim os seguintes diagramas de persistência

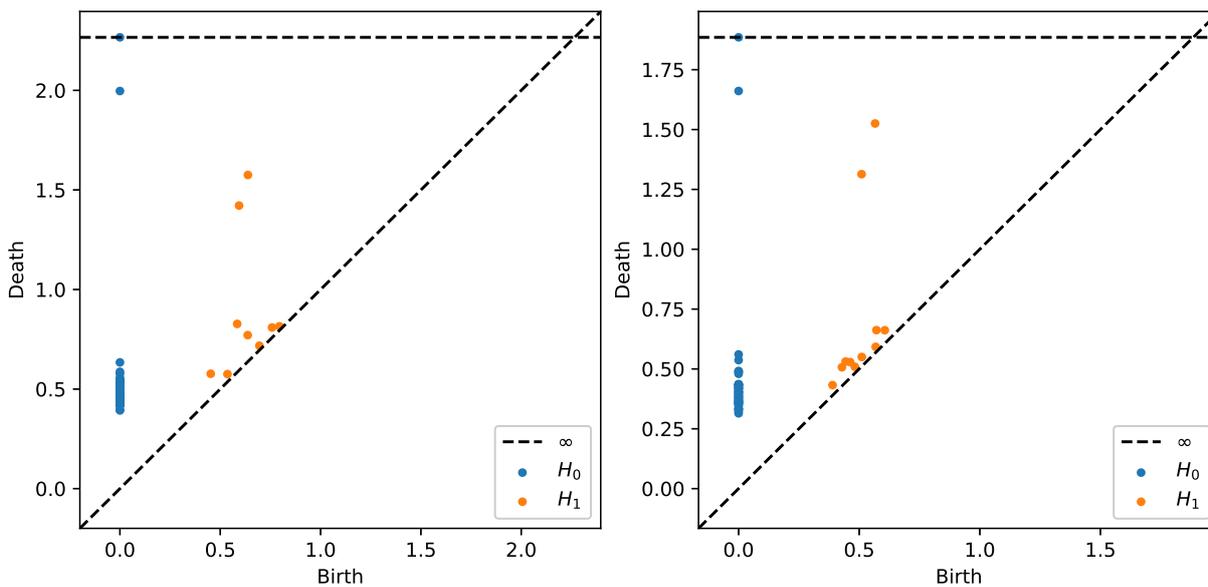


Figura 3.11: Diagrama de persistência dos respectivos bancos de dados da Figura 3.10.

Como já tínhamos argumentado no Exemplo 3.2.2, a teoria usual de Persistência Homológica não consegue distinguir as sutilezas dos dois datasets acima, uma vez que, a menos de diferenças nos ruídos, temos essencialmente os mesmos diagramas de persistência. Desta forma, nos utilizamos a teoria de merge trees decoradas para diferenciá-los.

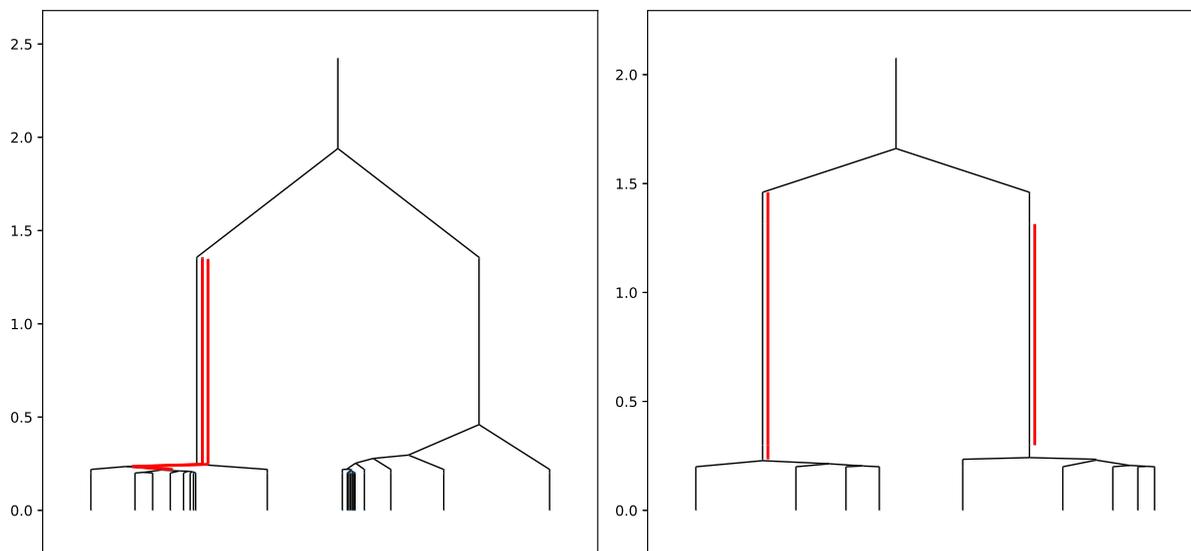


Figura 3.12: Merge trees decoradas dos respectivos datasets da Figura 3.10. A decoração posta nas Merge Trees neste caso é vinda da homologia de grau 1, isto é, representam neste caso H_1 .

As barras vermelhas na Figura 3.12 nos mostram que os datasets são de fato distintos. Usamos essa informação para rastrear a homologia H_1 de cada componente conexa, o que nos mostra que essas homologias correspondem a componentes distintas.

Existe uma discussão feita em [8] como criar um algoritmo que aproxime a computação da distância interleaving entre Merge Trees, vista que esse é um problema NP-Difícil. O algoritmo em questão se utiliza da distância de Gromov-Wasserstein para fazer essa aproximação. Entretanto, optamos por omitir essa discussão aqui, pois acreditamos que esses quesitos computacionais acabam fugindo um pouco do que tínhamos proposto inicialmente para este trabalho.

Apêndice: Códigos Usados

Aqui neste apêndice colocamos alguns dos códigos em Python utilizados ao decorrer deste trabalho. Esta foi uma recomendação da banca, para facilitar a leitura de pessoas que não tem muito contato com a parte aplicada de TDA.

Exemplo 1.1.15 e 1.1.18

```
1  #!/usr/bin/python3
2
3  import numpy as np
4  import matplotlib.pyplot as plt
5  import gudhi as gd
6  from sklearn import datasets
7  from ripser import ripser
8  from persim import plot_diagrams
9
10
11 def circulo(n, noise,centro_x,centro_y,r):
12     x=centro_x+r*np.cos(np.linspace(0,2*np.pi,n))
13     y=centro_y+r*np.sin(np.linspace(0,2*np.pi,n))
14     ruido=np.random.rand(n,2)
15
16     data=np.array([x,y]).T+noise*r*ruido
17
18     return data
19
20 array=circulo(200,0.5,1,1,2)
21 array1=circulo(200,0.4,3,1,2)
22 array2=np.append(array,array1,axis=0)
23
24 def noisy_circulo(n_samples, noise_level, center_x, center_y ,radius):
```

```
25
26     t = np.linspace(0,2*np.pi,n_samples)
27     x = center_x + radius*np.cos(t)
28     y = center_y + radius*np.sin(t)
29     noise = np.random.rand(n_samples,2)
30     data = np.array([x,y]).T + noise_level*radius*noise
31
32     return data
33
34 b=noisy_circulo(100,0.1,1,1,2)
35
36 plt.scatter(array2[:,0],array2[:,1])
37 plt.axis('scaled')
38 plt.savefig('1-pointcloud.eps', format='eps', dpi=1200)
39 plt.show()
40
41 rips_complex=gd.RipsComplex(points=array2,max_edge_length=2)
42 simplex_tree=rips_complex.create_simplex_tree(max_dimension=2)
43 diag=simplex_tree.persistence(min_persistence=0.2)
44
45 gd.plot_persistence_barcode(diag)
46 plt.savefig('1-barcode.eps', format='eps', dpi=1200)
47 plt.show()
48
49 dgms=ripsier(array2)['dgms']
50 plot_diagrams(dgms)
51 plt.savefig('1-dgms.eps', format='eps', dpi=1200)
52 plt.show()
```

Exemplo 1.1.19

```
1  #!/usr/bin/python3
2
3  import numpy as np
4  import matplotlib.pyplot as plt
5  from ripser import ripser
6  from persim import plot_diagrams
```

```

7
8 N = 15000
9 R = 4
10 r = 2
11
12 teta = np.random.rand(N)*2*np.pi
13 fi = np.random.rand(N)*2*np.pi
14 X = np.zeros((N, 4))
15 X[:, 0] = (R + r*np.cos(teta))*np.cos(fi)
16 X[:, 1] = (R + r*np.cos(teta))*np.sin(fi)
17 X[:, 2] = r*np.sin(teta)*np.cos(fi/2)
18 X[:, 3] = r*np.sin(teta)*np.sin(fi/2)
19
20 dgms2 = ripser(X, maxdim=2,coeff=2, n_perm=300)['dgms']
21 plot_diagrams(dgms2)
22 plt.savefig('2-dgms1.eps',format='eps',dpi=1200)
23 plt.show()

```

Exemplo 3.4.14

```

1 #!/usr/bin/python3
2
3 import numpy as np
4 import matplotlib.pyplot as plt
5 from ripser import ripser
6 import persim
7 import networkx as nx
8 import random
9 from scipy.optimize import linear_sum_assignment
10 from sklearn.metrics.pairwise import pairwise_distances
11
12 from DMT_tools import *
13 import ot
14 import time
15 from persim import plot_diagrams
16
17 def noisy_circulo(n, noise, centro_x, centro_y ,r):

```

```
18
19     t = np.linspace(0,2*np.pi,n)
20     x = centro_x + r*np.cos(t)
21     y = centro_y + r*np.sin(t)
22     ruido = np.random.rand(n,2)
23     data = np.array([x,y]).T + noise*r*ruido
24
25     return data
26
27 def noisy_disco(n,noise,centro_x,centro_y,r):
28
29     t = np.linspace(0,2*np.pi,n)
30     r = r*np.random.rand(n)
31     x = centro_x + np.multiply(r,np.cos(t))
32     y = centro_y + np.multiply(r,np.sin(t))
33     ruido = np.random.rand(n,2)
34     data = np.array([x,y]).T + noise*r*ruido
35
36     return data
37
38 data1=noisy_circulo(150,0.5,4,0,1)
39 data2=noisy_circulo(150,0.5,6.5,0,1)
40 data3=noisy_disco(150,0.5,0,0,1)
41 data4=np.append(data1,data2,axis=0)
42 data=np.append(data4,data3,axis=0)
43
44 data5=noisy_circulo(150,0.5,0,0,1)
45 data6=noisy_circulo(150,0.5,4,0,1)
46 Data=np.append(data5,data6,axis=0)
47
48 plt.scatter(data[:,0],data[:,1])
49 plt.axis('equal')
50 plt.savefig('ex-3-pc-1.eps',format='eps',dpi=1200)
51 plt.show()
52 plt.scatter(Data[:,0],Data[:,1])
53 plt.axis('equal')
54 plt.savefig('ex-3-pc-2.eps',format='eps',dpi=1200)
55 plt.show()
56
```

```
57 dgms = ripser(data, maxdim=1, n_perm=50)['dgms']
58 plot_diagrams(dgms)
59 plt.savefig('ex-3-diagram-1.eps',format='eps',dpi=1200)
60 plt.show()
61
62 dgms2 = ripser(Data, maxdim=1, n_perm=50)['dgms']
63 plot_diagrams(dgms2)
64 plt.savefig('ex-3-diagram-2.eps',format='eps',dpi=1200)
65 plt.show()
66
67 MT1=MergeTree(pointCloud=data)
68 MT1.fit_barcode(degree=1)
69 tree_thresh=0.2
70 barcode_thresh=0.4
71 MT1.draw_decorated(tree_thresh,barcode_thresh)
72 plt.savefig('ex-3-tree-1.eps',format='eps',dpi=1200)
73 plt.show()
74
75 MT2=MergeTree(pointCloud=Data)
76 MT2.fit_barcode(degree=1)
77 tree_thresh=0.2
78 barcode_thresh=0.4
79 MT2.draw_decorated(tree_thresh,barcode_thresh)
80 plt.savefig('ex-3-tree-2.eps',format='eps',dpi=1200)
81 plt.show()
```

Vale a pena destacar que neste exemplo estamos com dois datasets com nomes muito semelhantes: `Data` e `data`. Alternamos os dois no código para que possamos fazer as figuras correspondentes de cada dataset no mesmo trecho de código. Depois, colocamos as figuras lado a lado no \LaTeX .

Índice

- (ε, δ) -pareamento, 66
- \mathbb{R} -espaço, 68
- ε -aplicação, 62
- ε -compatíveis, 63
- ε -interleaving, 43
 - de \mathbb{R} -espaços, 69
- ε -pareamento, 22
- k -simplexo simplicial, 9
- Índice de persistência, 13
- Altura merge ($merge_F$), 51
- Barcode, 14
- Categoria, 24
 - \mathbf{H}_0 , 68
 - \mathbf{Top}_c , 40
 - \mathbf{Top}_{lc} , 40
 - \mathbf{AbGrps} , 37
 - \mathbf{CRngs} , 28
 - $\mathbf{Fun}(C, D)$, 29
 - \mathbf{Grps} , 27
 - \mathbf{Sets} , 25
 - \mathbf{Top} , 25
 - \mathbf{Vec} , 25
 - \mathbf{vec} , 25
 - abeliana, 37
 - concreta, 27
 - discreta, 25
 - dos objetos discretamente parametriza-
dos, 39
 - dual, 25
 - pequena, 25
- Categorias equivalentes, 31
- Coequalizador, 36
- Cokernel, 36
- Coleção de subespaços vetoriais independente,
75
- Complexo
 - de Čech, 11
 - de Vietoris-Rips, 11
 - simplicial, 10
- Composição vertical, 29
- Conjunto
 - superior, 53
 - persistente/ de persistência, 39
 - construtível, 75
- Convex hull, 9
- Coproducto de categorias, 30
- Custo de um pareamento, 21
- Decomponível em intervalo real, 74
- Diagrama de persistência, 16
- Display poset, 51
 - tame, 75
- Distância
 - Bottleneck (d_B), 22
 - de tipo homotópico persistente (δ_I), 69
 - interleaving, 44
 - Morozov-Beketayev-Weber (θ_I^{MBW}), 63
 - entre merge trees (θ_I), 65
 - natural (d_{NP}), 69
- Epígrafo, 50
- Epimorfismo, 27
- Equalizador, 35
- Espaço
 - persistente/ de persistência, 38

- Filtração, 12, 38
 - offset, 39
 - sublevel-set, 38
- Folhas de uma merge tree, 51
- Forgetful functor, 26
- Função tame, 23
- Functor, 26
 - π_0 , 27
 - cop*, 40
 - dom*, 40
 - pb*, 41
 - adivito, 38
 - cheio, 31
 - contravariante, 26
 - covariante, 26
 - essencialmente sobrejetor, 31
 - fiel, 31
- Grafo de Reeb, 49
- Grupo de homologia persistente, 13
- Homomorfismo entre módulos de persistência, 19
- Isomorfismo, 27
 - natural, 28
- Kernel, 36
- Levantamento
 - de um barcode B , 81
 - de I , 81
- Módulo
 - de árvore, 58
 - de intervalo, 20
 - de persistência, 19
 - decomponível, 20
- Menor ancestral comum (LCA_F), 51
- Merge Tree
 - Clássica, 50
 - conexa, 52
 - decorada
 - categorial, 57
 - concreta, 58
 - por barcode, 59
 - por folhas, 59
 - por levantamento, 81
 - generalizada, 51
- Monomorfismo, 27
- Morfismo, 24
- Multiset, 16
- Nervo, 10
- Objeto
 - inicial, 36
 - nulo, 36
 - terminal, 36
- Objetos isomorfos, 27
- Pareamento, 21
 - ótimo, 22
 - parcial, 21
- Poset, 27
- Produto de categorias, 29
- Projeção, 29
- Propriedade
 - H_k -disjunção, 81
 - Universal, 29
 - funtorial, 26
- Simplexo de nascimento, 80
- Soma direta de módulos de persistência, 20
- Teorema
 - da Continuidade da Merge Tree Decorada por Barcode, 61
 - da Estabilidade Algébrica, 47
 - da Hierarquia dos Teoremas de Estabilidade, 70
 - da Injetividade da Transformada de Barcode, 79

- de Caracterização de Equivalências de Categorias, 32
- de Crawley-Boevey, 21
- de Estabilidade, 23, 47
 - das Merge Trees, 64
- de Isometria, 47
- de Krull-Remak-Schmidt-Azumaya, 21
- dos Nervos, 10
- Topologia de intervalo, 54
- Transformação Natural, 28
- Transformada de barcode, 60
- Truncamento
 - de um barcode, 60
 - de um intervalo, 60
- Valor crítico homológico, 23

Bibliografia

- [1] BORCEUX, Francis. *Handbook of Categorical Algebra: Volume 1, Basic Category Theory*. Vol. 1. Cambridge University Press, 1994. 345 p.
- [2] BORCEUX, Francis. *Handbook of Categorical Algebra: Volume 2, Categories and Structures*. Vol. 2. Cambridge University Press, 1994. 443 p.
- [3] BUBENIK, Peter; SCOTT, Jonathan A. *Categorification of Persistent Homology*. In: *Discrete & Computational Geometry* 51.3 (2014), 600–627.
- [4] CHAZAL, Frédéric; DE SILVA, Vin; GLISSE, Marc; OUDOT, Steve. *The structure and stability of persistence modules*. arXiv preprint arXiv:1207.3674, v. 21, 2012.
- [5] COHEN-STEINER, David, EDELSBRUNNER, Herbet, HARER, John. *Stability of Persistence Diagrams* *Discrete Comput. Geom.* 28 (2006), 103–120.
- [6] CORMEN, Thomas H; LEISERSON, Charles; RIVEST, Ronald; STEIN, Clifford. *Introduction to Algorithms*. 3rd ed. Cambridge: The MIT Press, 2009. xix, 1292 p. ISBN 9780262033848.
- [7] CRAWLEY-BOEVEY, William. *Decomposition of Pointwise Finite-Dimensional Persistence Modules*. *Journal of Algebra and its Applications*, v. 14, n. 05, 1550066, 2015.
- [8] CURRY, Justin; HANG, Haibin; MIO, Washington; NEEDHAM, Tom; OKUTAN, Osman B. (2022). *Decorated Merge Trees for Persistent Topology*. *Journal of Applied and Computational Topology*, 1-58.
- [9] DANTCHEV, Stefan; IVRISSIMTZIS, Ioannis. *Efficient Construction of the Čech Complex*. *Computers & Graphics*, v. 36, n. 6, p. 708-713, 2012.
- [10] DE BERG, Mark; KREVELD, Marc; OVERMARS, Mark; SCHWARZKOPF, Otfried. *Computational Geometry*. In: *Computational Geometry*. Springer, Berlin, Heidelberg. , 2000, 367 p.
- [11] DE SILVA, Vin; CARLSSON, Gunnar E. *Topological Estimation Using Witness Complexes*. In: *PBG*. 2004. p. 157-166.

- [12] EDELSBRUNNER, Herbert; HARER, John L. *Computational Topology: An Introduction*. Providence, R.I.: American Mathematical Society, 2010. 241 p. ISBN 978-0-8218-4925-5.
- [13] EDELSBRUNNER, Herbert. *A Short Course in Computational Geometry and Topology*. Springer, 2013, 110p.
- [14] EDELSBRUNNER, Herbert; LETSCHER, David; ZOMORODIAN, Afra. *Topological Persistence and Simplification*. In: Proceedings 41st annual symposium on foundations of computer science. IEEE, 2000. p. 454-463.
- [15] EILENBERG, Samuel. MACLANE, Saunders. *General Theory of Natural Equivalences*. Transactions of the American Mathematical Society 58, no. 2 (1945): 231–94.
- [16] FRALEIGH, John. *A First Course in Abstract Algebra*. 7. ed. Boston: Addison Wesley, 2003. 520 p. ISBN 978-0-201-76390-4.
- [17] FROSINI, Patrizio; LANDI, Claudia; MÉMOLI, Facundo. *The persistent homotopy type distance*. In: Homology, Homotopy and Applications 21.2 (2019), pp. 231–259.
- [18] GREENBERG, Marvin Jay; HARPER, John L. *Algebraic Topology: A First Course*. Redwood City: Addison-Wesley, c1981. 311 p. ISBN 0-8053-3558-7.
- [19] HATCHER, Allen. *Algebraic Topology*. New York: Cambridge University Press, 2001. 544 p. ISBN 978-0-521-79540-1.
- [20] JACOBSON, Nathan. *Basic Algebra*. 2. ed. New York: W.H. Freeman, 1985. 499 p. ISBN 978-0486-47189-1.
- [21] KNUTH, Donald Ervin. *The Art of Computer Programming*. 3. ed. Reading: Addison - Wesley, 1997. 650 p. ISBN 0-201-89683-4.
- [22] LEE, John M. *Introduction to Smooth Manifolds*. New York: Springer, 2000. 494 p. (Graduate Texts in Mathematics; 218).
- [23] LEE, John M. *Introduction to Topological Manifolds*. New York: Springer, 2000. 385 p. (Graduate Texts in Mathematics; 202). ISBN 0-387-95026-5.e
- [24] LESNICK, Michael. *The theory of the interleaving distance on multidimensional persistence modules*. Foundations of Computational Mathematics, v. 15, n. 3, p. 613-650, 2015.
- [25] MACLANE, Saunders. *Categories for The Working Mathematician*. 2. ed. New York: Springer, c1971. 314 p. (Graduate texts in Mathematics; v.5). ISBN 0-387-98403-8.

- [26] MOROZOV, Dmitriy; BEKETAYEV, Kenes; WEBER, Gunther. *Interleaving Distance Between Merge Trees*. In: Discrete and Computational Geometry 49 (2013), pp. 22–45.
- [27] MUNKRES, James Raymond. *Elements of Algebraic Topology*. Redwood City: Menlo Park, 1984. 454 p. ISBN 0-201-04586-9.
- [28] MUNKRES, James Raymond. *Topology*. 2. ed. New Jersey: Prentice Hall, 2000. 537 p. ISBN 0-13-181629-2.
- [29] NEEDHAM, Tom. Decorated-Merge-Trees, 2021. Disponível em: <https://github.com/trneedham/Decorated-Merge-Trees>. Acesso em: 10 de Agosto de 2023.
- [30] SPANIER, Edwin H. *Algebraic Topology*. New York: McGraw-Hill Book, c1966. 528 p. (McGraw-Hill Series in Higher Mathematics). ISBN 0-07-099582-6.
- [31] RIEHL, Emily. *Category Theory in Context*. Courier Dover Publications, 2017.
- [32] RIPSER.PY 0.6.4 DOCUMENTATION, 2019. Disponível em: <https://ripser.scikit-tda.org/en/latest/>. Acesso em: 10 de Agosto de 2023.
- [33] ZOMORODIAN, Afra; CARLSSON, Gunnar. *Computing Persistent Homology*. In: Proceedings of the twentieth annual symposium on Computational geometry. 2004. p. 347-356.
- [34] ZOMORODIAN, Afra J. *Topology for Computing*. Cambridge university press, 2005. 243 p.