UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Geradores de homologia persistente e aplicações

Carlos Henrique Venturi Ronchi

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

ICMC USP SÃO CARLOS

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Carlos Henrique Venturi Ronchi

Geradores de homologia persistente e aplicações

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

Área de Concentração: Matemática

Orientador: Prof. Dr. Marcio Fuzeto Gameiro

USP – São Carlos
Junho de 2018

Carlos Henrique Venturi Ronchi

Persistent homology generators and applications

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

Concentration Area: Mathematics

Advisor: Prof. Dr. Marcio Fuzeto Gameiro

USP – São Carlos
June 2018

RESUMO

RONCHI, C. H. V. **Geradores de homologia persistente e aplicações**. 2018. 52 p. Dissertação (Mestrado em Ciências – Matemática) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

a.

Palavras-chave: Modelo, Monografia de qualificação, Dissertação, Tese, Latex.

ABSTRACT

RONCHI, C. H. V. **Persistent homology generators and applications**. 2018. 52 p. Dissertação (Mestrado em Ciências – Matemática) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2018.

a.

Keywords: Template, Qualification monograph, Dissertation, Thesis, Latex.

LISTA DE ILUSTRAÇÕES

LISTA DE ALGORITMOS

LISTA DE CÓDIGOS-FONTE

LISTA DE TABELAS

SUMÁRIO

CAPÍTULO

1

INTRODUÇÃO

CAPÍTULO

2

HOMOLOGIA PERSISTENTE 101

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

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

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

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

1. A entrada do algoritmo pode ser um conjunto de pontos ou alguma matriz de distância/similaridade do conjunto de dados.

2. A construção de um objeto combinatorial em cima do conjunto de dados ou da matriz de distância. Geralmente uma filtração ou um complexo simplicial.

3. A partir da filtração ou do complexo simplicial é possível extrair informações topológicas e geométricas do conjunto de dados, por exemplo o número de componentes conexas, como um algoritmo de Clustering.

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

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

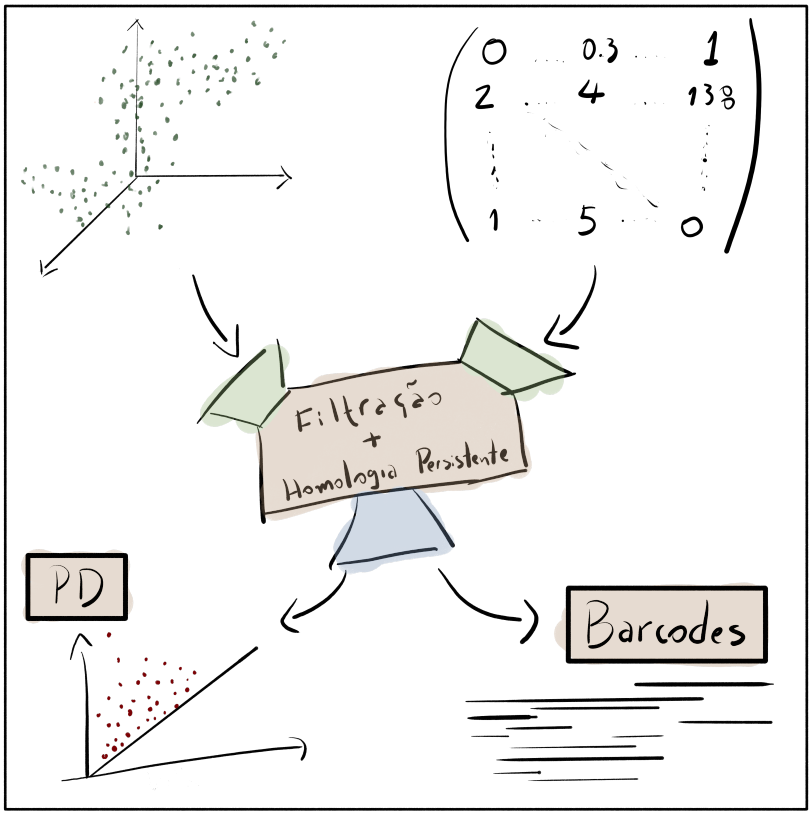

Figura 1 – Representação do pipeline para a utilização da homologia persistente com um conjunto de dados.

Fonte: Elaborada pelo autor.

2.1 Filtrações

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

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

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

$$\left\{ \sum_{i=0}^{k} \lambda_i v_i \;\middle|\; \sum_{i=0}^{k} \lambda_i = 1 \text{ e } \lambda_i \geq 0, \ \forall i \right\}. \tag{2.1}$$

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

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

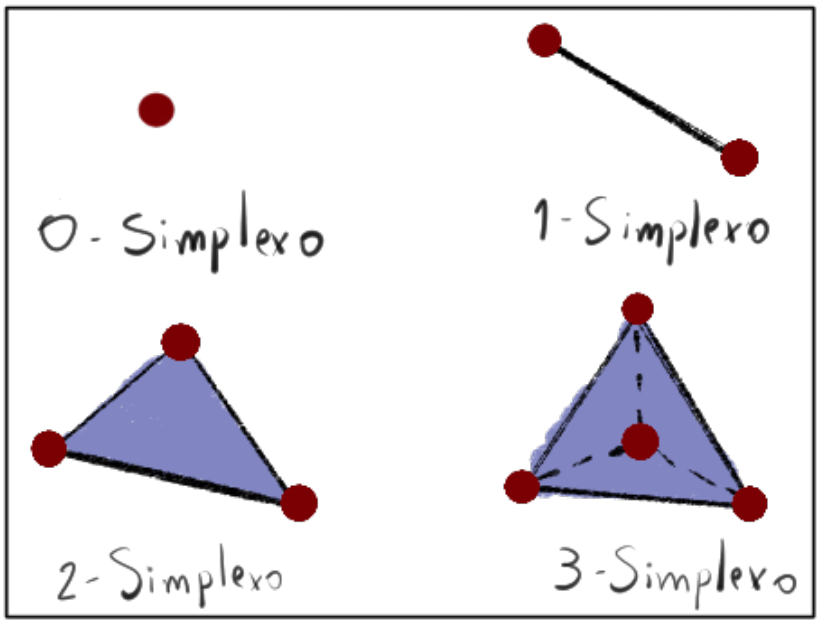

Figura 2 – Exemplos de k-simplexos para $k \in \{0, 1, 2, 3\}$.

Fonte: Elaborada pelo autor.

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

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

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

Nessa definição utilizamos o símbolo $\subset$ para indicar que uma face. Usaremos esse símbolo com essa denotação quando falarmos sobre simplexos e faces. A segunda condição

é necessária para evitar casos patológicos como mostrado na Figura 3. Dizemos que a dimensão do complexo simplicial K é a maior dimensão dentre os simplexos em K. Podemos definir agora a filtração de um complexo simplicial.

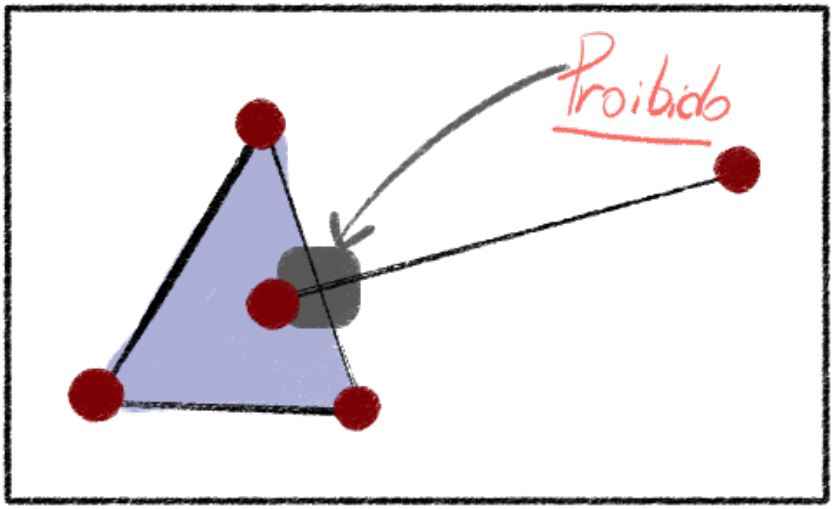

Figura 3 – Exemplo em que a interseção de dois simplexos não é um simplexo.

Fonte: Elaborada pelo autor.

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

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

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

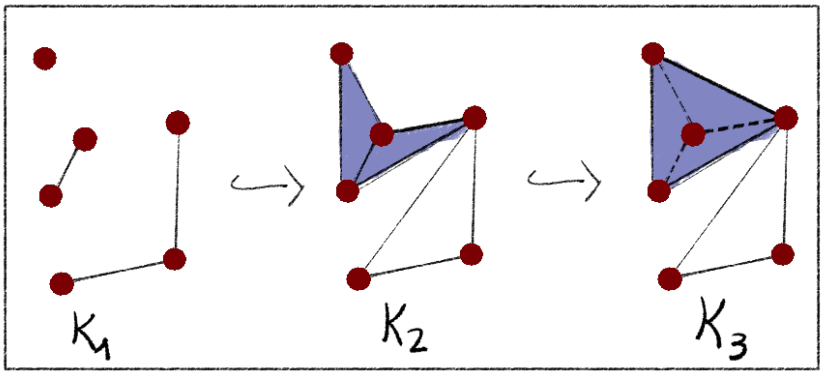

Figura 4 – Exemplo de filtração para um complexo simplicial K.

Fonte: Elaborada pelo autor.

2.1.1 Complexo de Čech

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

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

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

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

Exemplo 2.1. Seja $X = \{a, b, c\}$ e considere $F = \{\{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}\}$. Precisamos mostrar que F é um complexo simplicial abstrato. Seja $\sigma = \{a, c\}$. Note que seus subconjuntos são $\{a\}$ e $\{c\}$, além disso ambos pertencem a F. De forma análoga, mostramos que para qualquer outro simplexo, suas faces (subconjuntos) estão em F.

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

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

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

Definição 2.5. Seja X um conjunto de pontos $\{x_1, \ldots, x_n\}$ em $\mathbb{R}^d$. O complexo de Čech de X para um valor real $r > 0$ é o conjunto $C^r(X)$, onde $\sigma = \langle x_{i_1}, \ldots, x_{i_k} \rangle \in C^r(X)$ se, e somente se vale a seguinte condição

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

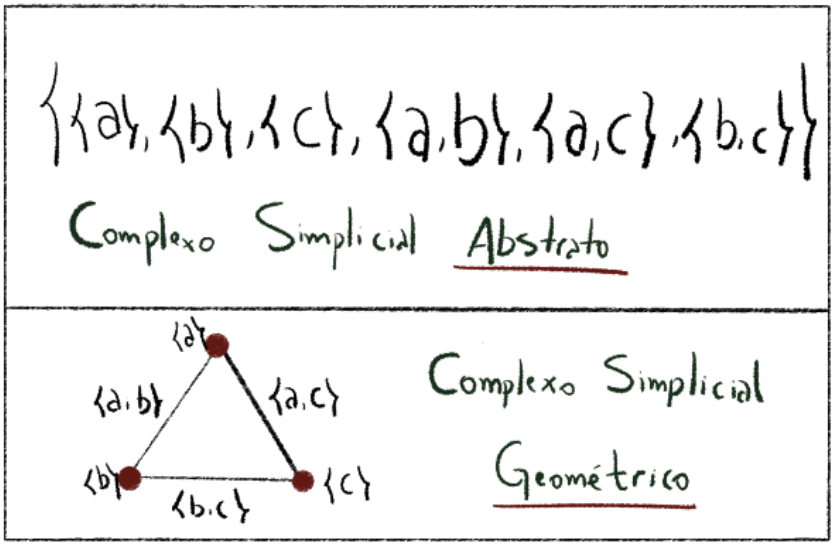

Figura 5 – Exemplo de um complexo simplicial abstrato e sua realização geométrica

Fonte: Elaborada pelo autor.

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

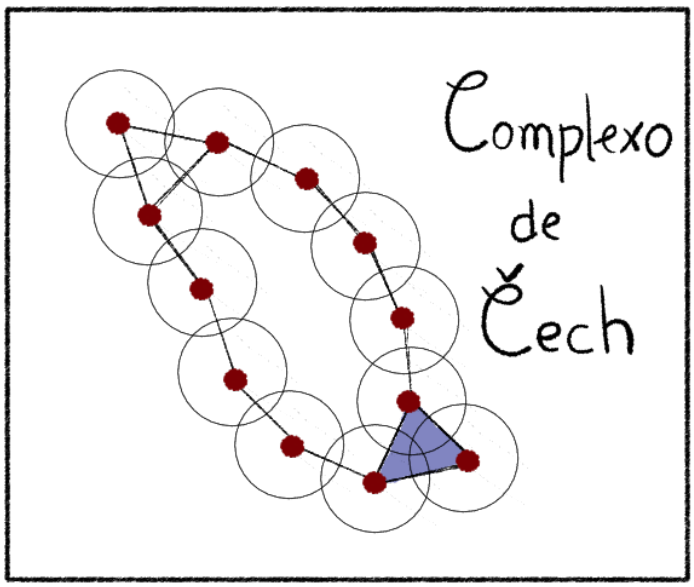

Figura 6 – Exemplo de um complexo de Čech para um raio r fixado.

Fonte: Elaborada pelo autor.

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

2.1.2 Complexo de Vietoris-Rips

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

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

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

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

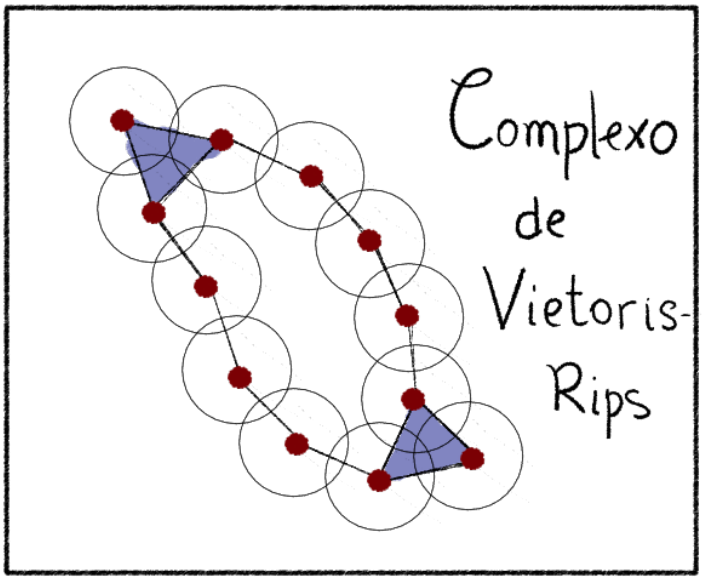

Figura 7 – Exemplo do complexo de Vietoris-Rips com os mesmos pontos utilizados para a construção na Figura 6.

Fonte: Elaborada pelo autor.

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

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

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

2.1.3 Complexo Alpha

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

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

$$V_i = \left\{ x \in \mathbb{R}^n \,\middle|\, d(x_i, x) \leq d(x_j, x),\ \forall j \in 1, \ldots, k \right\},$$

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

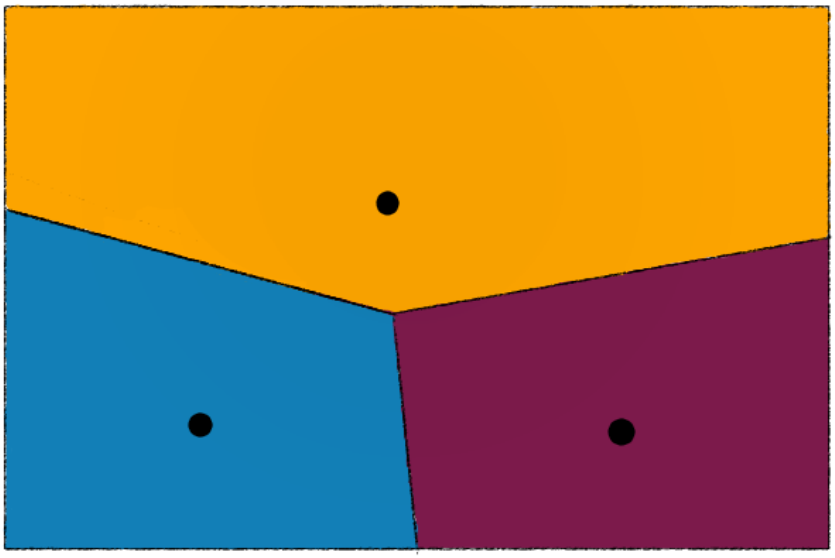

Figura 8 – Diagrama de Voronoi de três pontos no plano.

Fonte: Elaborada pelo autor.

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

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

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

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

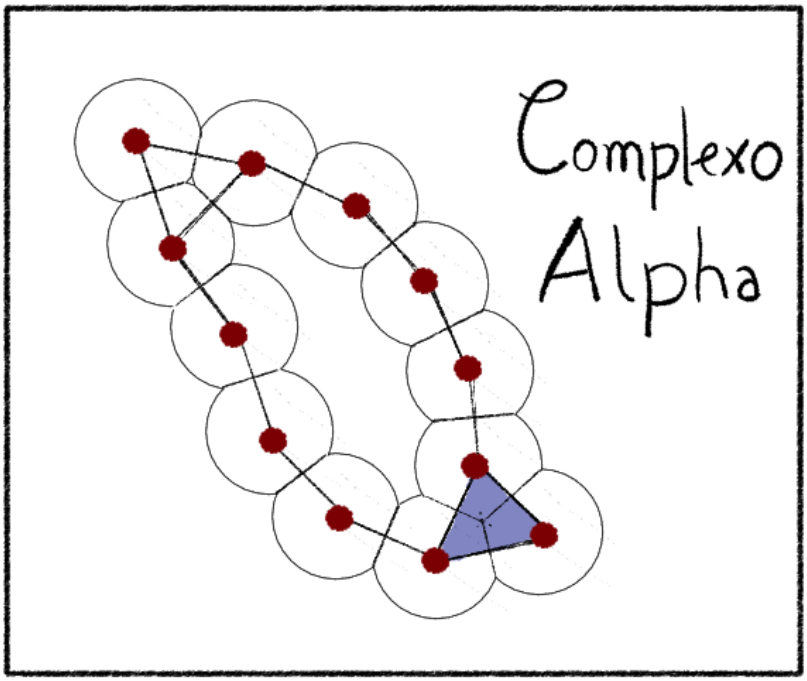

Figura 9 – Complexo Alpha para um conjunto de pontos no plano.

Fonte: Elaborada pelo autor.

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

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

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

2.2 A matriz de bordo ∂

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

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

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

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

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

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

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

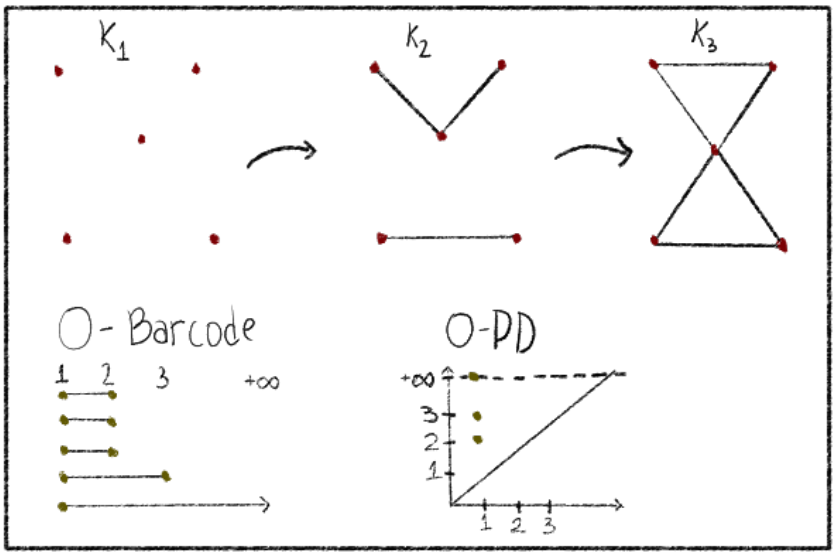

Figura 10 – Exemplo da filtração de um complexo simplicial e o barcode e diagramas de persistência associados.

Fonte: Elaborada pelo autor.

Tendo essa ferramenta, como podemos traduzi-la para o contexto dos dados, e como calcular os pares dos diagramas de persistência? Abaixo segue uma lista dos primeiros passos que devem ser feitos para a obtenção do diagrama de persistência.

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

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

Definição 2.8. Seja K um complexo simplicial, $K_1 \subset \cdots \subset K_m$ uma filtração e $\sigma_1, \ldots, \sigma_n$ uma ordenação dos simplexos de K satisfazendo as regras acima mencionadas. A *matriz de bordo* de K, denotada por ∂, é uma matriz de tamanho $n \times n$, em que cada entrada tem o seguinte valor

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

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

2.3 Redução da matriz ∂

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

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

Algoritmo 1 – Redução da matriz bordo ∂.

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

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

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

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

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

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

Agora podemos construir o diagrama de persistência para a filtração dada utilizando a matriz de redução. Denotamos por Dgm_p o p-ésimo diagrama de persistência, com $p \in \{0,\ldots,k-1\}$ onde k é a maior dimensão dentre os simplexos σ_i. Cada p representa a dimensão dos grupos de homologia descritos anteriormente. Se $p=0$, então o diagrama de persistência nos dirá quais componentes conexas apareceram na filtração e o quão persistente elas são. Para $p=1$, teremos buracos 1-dimensionais, por exemplo, um círculo vazado tem um buraco. Já um toro, tem dois buracos, um visível e outro por dentro da superfície.

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

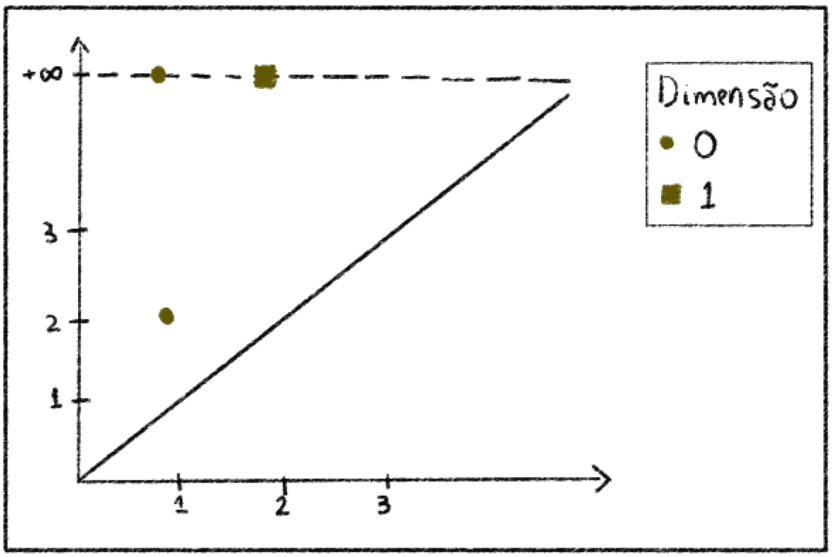

Figura 11 – 0– e 1– Diagramas de Persistência da Figura 4.

Fonte: Elaborada pelo autor.

2.4 Calculando a homologia persistente

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

Considere a filtração da Figura 12. Observe que temos 4 vértices $(\sigma_1,\ldots,\sigma_4)$, 5 arestas $(\sigma_5,\ldots,\sigma_9)$ e 1 triângulo (σ_{10}) ao total, temos ao total 10 simplexos. Diretamente da figura já podemos extrair os diagrams de persistência de dimensão 0 e 1. Note que no

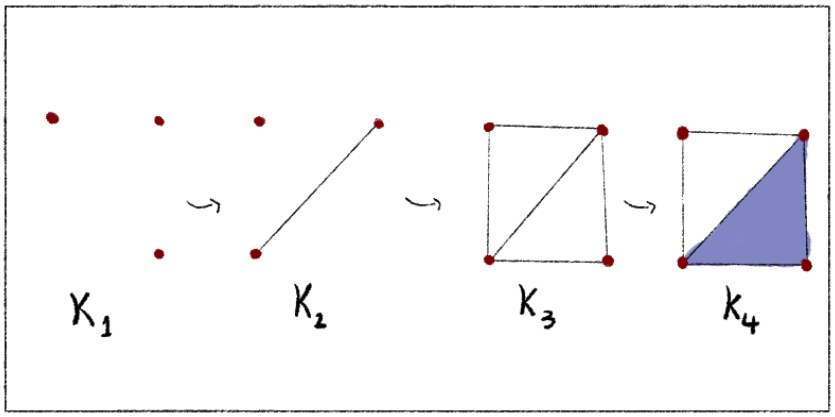

Figura 12 – Filtração de um complexo simplicial.

Fonte: Elaborada pelo autor.

primeiro passo da filtração temos 2 componentes conexas, sendo que uma delas morre no terceiro passo e a outra sobrevive até o final. Temos portanto dois intervalos e logo dois pontos no 0-diagrama de persistência: $[1,+\infty)$ e $[1,3)$.

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

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

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

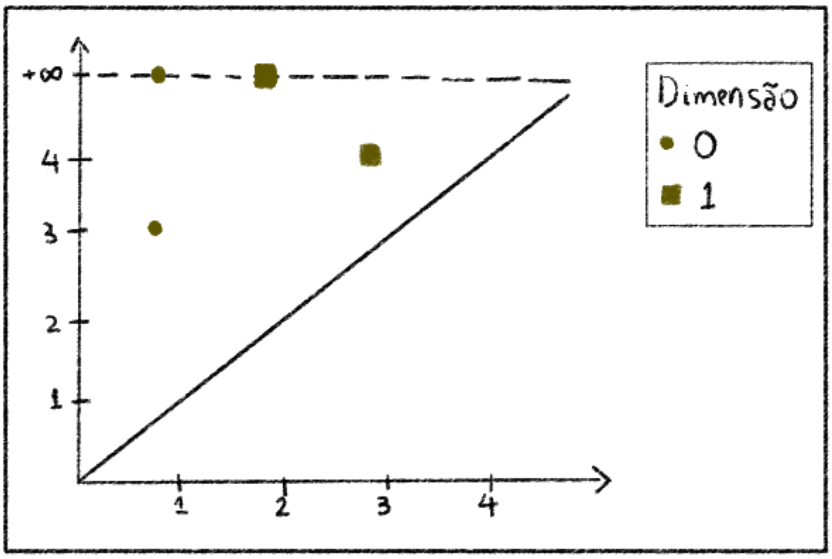

Figura 13 – 0– e 1– diagramas de persistência da filtração mostrada na Figura 12.

Fonte: Elaborada pelo autor.

que as regras para construção da matriz são satisfeitas.

$$\partial = \begin{array}{c} \\ \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \\ \sigma_7 \\ \sigma_8 \\ \sigma_9 \\ \sigma_{10} \end{array} \begin{array}{c} \begin{array}{cccccccccc} \sigma_1 & \sigma_2 & \sigma_3 & \sigma_4 & \sigma_5 & \sigma_6 & \sigma_7 & \sigma_8 & \sigma_9 & \sigma_{10} \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \tag{2.5}$$

Para reduzir vamos realizar as seguintes operações:

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

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

$$R = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{2.6}$$

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

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

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

- $[a_2, a_5) = [1,1)$, que não seria adicionado ao diagrama,
- $[a_4, a_6) = [2,2)$, que não seria adicionado ao diagrama,
- $[a_3, a_8) = [1,3)$,
- $[a_9, a_10) = [3,4)$,
- $[a_1, +\infty) = [1, +\infty)$,
- $[a_7, +\infty) = [2, +\infty)$,

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

CAPÍTULO

3

MÓDULOS DE PERSISTÊNCIA

CAPÍTULO

4

GERADORES ÓTIMOS E OUTROS CONCEITOS

4.1 Geradores ótimos

4.2 Vetorização do diagrama de persistência

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

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

4.2.1 Estabilidade da Imagem de Persistência

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

o processo para obter uma imagem de persistência.

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

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

Definição 4.1. Para um diagrama de persistência B, a correspondente superfície de persistência $\rho_B : \mathbb{R}^2 \to \mathbb{R}$ é a função dada por

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

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

Definição 4.2. Seja B um diagrama de persistência. A imagem de persistência de B é a coleção de pixeis

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

Na vetorização do diagrama alguns parâmetros precisam ser estabelecidos. Em (ADAMS *et al.*, 2017) mostra-se que as imagens são robustas sob a escolha da resolução (tamanho do grid). A outra escolha é a distribuição e dependendo a variância. Em (ADAMS *et al.*, 2017) a distribuição gaussiana é utilizada com variância dependendo do problema e sendo assim o usuário a escolhe. Por último, a escolha da função peso, que pode variar de problema pra problema. A função abaixo é um exemplo utilizado por (ADAMS *et al.*, 2017). Observe que para pontos com valores altos de persistência tem um valor maior também. Mas com problemas que pontos de baixa ou média persistência são importantes, a utilização de outros pontos se faz necessária.

$$w_b(t) = \begin{cases} 0 & \text{if } t \leq 0, \\ \frac{t}{b} & \text{if } 0 < t < b, \\ 1 & \text{if } t \geq b, \end{cases} \tag{4.1}$$

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

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

Definição 4.3. A distância p-Wasserstein definida entre dois diagramas de persistência B e B' é dada por

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

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

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

$$|h(u)-h(v)| \le |\nabla h| \, \|u-v\|_2 . \tag{4.2}$$

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

$$\|\phi_u - \phi_v\|_\infty \le |\nabla \phi| \, \|u-v\|_2 . \tag{4.3}$$

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

Lema 4.1. Sejam $u,v \in \mathbb{R}^2$, temos que $\|f(u)\phi_u - f(v)\phi_v\| \le (\|f\|_\infty |\nabla \phi| + \|\phi\|_\infty |\nabla f|) \, \|u-v\|_2$.

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

$$\begin{aligned}
|f(u)\phi_u(z) - f(v)\phi_v(z)| &= |f(u)(\phi_u(z)-\phi_v(z)) + (f(u)-f(v))\phi_v(z)| \\
&\le \|f\|_\infty |\phi_u(z)-\phi_v(z)| + \|\phi\|_\infty |f(u)-f(v)| \\
&\le \|f\|_\infty |\nabla\phi| \, \|u-v\|_2 + \|\phi\|_\infty + |\nabla f| \, \|u-v\|_2 \quad \text{por 4.3 e 4.2} \\
&= (\|f\|_\infty |\nabla\phi| + \|\phi\|_\infty + |\nabla f|) \, \|u-v\|_2 .
\end{aligned}$$

□

Teorema 4.2. A superfície de persistência é estável em relação a distância 1-Wasserstein. Dados B, B' diagramas de persistência finitos, temos que

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

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

$$\begin{aligned}
\|\rho_B - \rho_{B'}\|_\infty &= \left\|\sum_{u\in T(B)} f(u)\phi_u - \sum_{u\in T(B)} f(\gamma(u))\phi_{\gamma(u)}\right\|_\infty \\
&\le \sum_{u\in T(B)} \left\|f(u)\phi_u - f(\gamma(u))\phi_{\gamma(u)}\right\| \\
&\le (\|f\|_\infty |\nabla\phi| + \|\phi\|_\infty + |\nabla f|) \sum_{u\in T(B)} \|u-\gamma(u)\|_2 && \text{por 4.1} \\
&\le \sqrt{2}(\|f\|_\infty |\nabla\phi| + \|\phi\|_\infty + |\nabla f|) \sum_{u\in T(B)} \|u-\gamma(u)\|_\infty && \text{já que } \|\cdot\|_2 \le \sqrt{2}\|\cdot\|_\infty \text{ em } \mathbb{R}^2 \\
&\le \sqrt{10}(\|f\|_\infty |\nabla\phi| + \|\phi\|_\infty + |\nabla f|) \sum_{u\in T(B)} \|u-\gamma(u)\|_\infty && \text{já que } \|T(\cdot)\|_2 \le \sqrt{5}\|\cdot\|_\infty \\
&= \sqrt{10}(\|f\|_\infty |\nabla\phi| + \|\phi\|_\infty + |\nabla f|) W_1(B,B').
\end{aligned}$$

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

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

Teorema 4.3. A imagem de persistência é estável em relaçaõ a distância 1-Wasserstein. Se A é o valor máximo dentre todos os pixeis da imagem, então

$$\|I(\rho_B) - I(\rho_{B'})\|_\infty \le \sqrt{10}(\|f\|_\infty |\nabla\phi| + \|\phi\|_\infty + |\nabla f|) W_1(B,B'). \tag{4.4}$$

Demonstração. A demonstração segue do Teorema 4.2 e do fato que para um pixel p qualquer

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

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

4.2.2 Exemplos de Imagens de Persistência

Considere X um conjunto de pontos extraídos de um círculo com ruído, como pode ser visto na Figura 14. Na Figura 15 tem-se os diagramas de persistência do círculo de dimensão 0 e 1, assim mostrando as componentes conexas e buracos. Note que existem dois pontos longe da diagonal, um representando a componente conexa e o outro o buraco do círculo. Vamos agora análisar as imagens de persistência para cada uma das images. Escolhemos a distribuição gaussiana dada por

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

A Equação 4.1 é utilizada como função peso. Para calcular as imagens de persistência definimos três variâncias: $0.001, 0.1, 1.0$, e dois tamanhos de imagem: 10×10 e 50×50. O resultado pode ser visto na Figura 16.

Observe a diferença entre os tamanhos escolhidos para as imagens. Com um tamanho maior, a informação fica mais fina, porém a imagem fica esparsa. Além disso, com

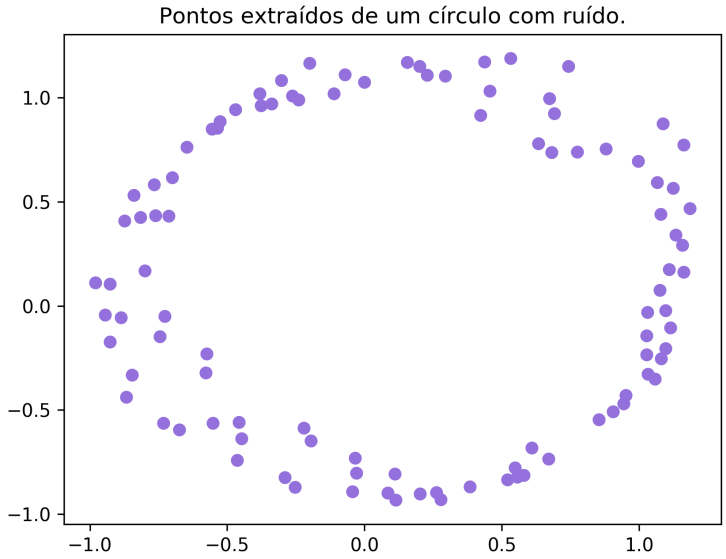

Figura 14 – Pontos extraídos de um círculo com ruídos.

Fonte: Elaborada pelo autor.

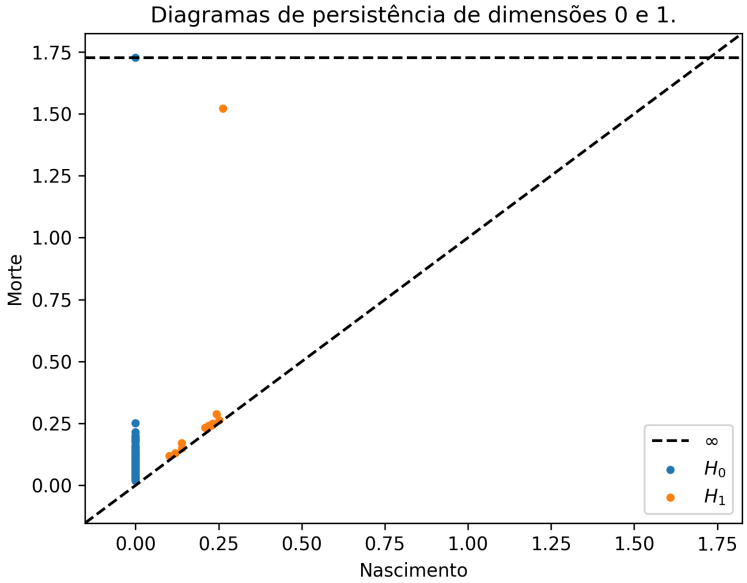

Figura 15 – Diagrams de persistência do círculo X. Em laranja o diagrama de persistência de dimensão 1, em azul o de dimensão 0. A filtração de Vietoris-Rips foi usada para calcular o complexo simplicial.

Fonte: Elaborada pelo autor.

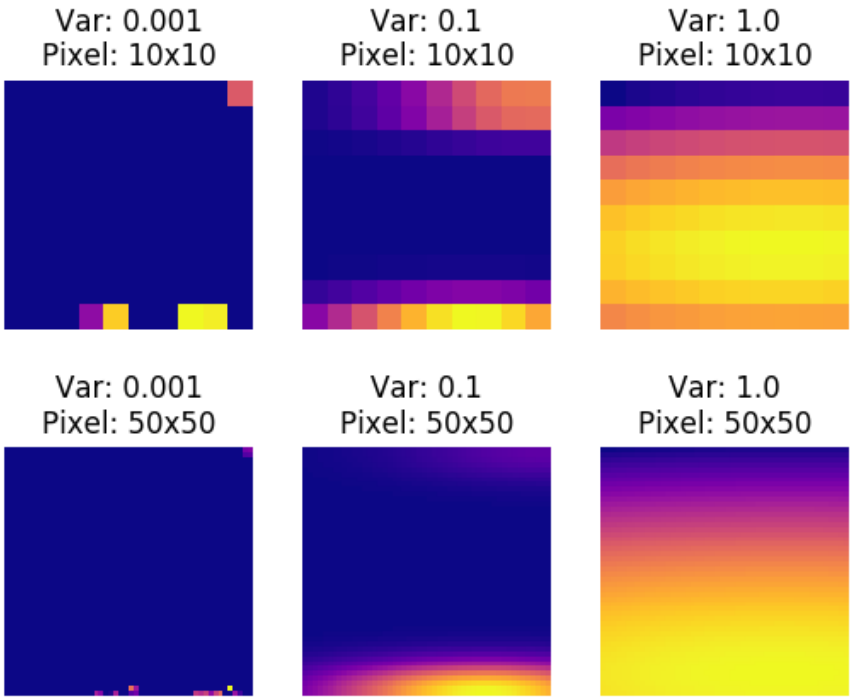

Figura 16 – 6 imagens de persistência do diagrama de dimensão 1 da Figura 15.

Fonte: Elaborada pelo autor.

uma variância mais baixa os pontos ficam mais concentrados, enquanto para valores mais altos há uma troca contínua entre as regiões dos pontos com maior frequência.

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

4.3 Mapper

CAPÍTULO

5

APLICAÇÕES

Neste capítulo serão descritas algumas aplicações utilizando geradores ótimos e imagens de persistência.

5.1 Geradores ótimos em classificadores de imagens

Utilizando imagens e rótulos associados a elas é possível criar classificadores, algoritmos que decidem os rótulos dada uma imagem. Alguns deles são Redes Neurais (MCCULLOCH; PITTS, 1943), SVM (CORTES; VAPNIK, 1995), Redes Neurais Convolucionais (abreviado por CNN, sigla em inglês) (LECUN *et al.*, 1989) e *Generative Adversarial Networks (GAN)* (GOODFELLOW *et al.*, 2014).

Nesta seção será descrito as redes neurais convolucionais e como obteve-se um classificador de imagens utilizando-as. Além disso, será descrito como outros classificadores foram gerados utilizando informações disponibilizadas pelos geradores ótimos para obter-se um classificador com melhor acurácia do que a rede neural convolucional original.

5.1.1 Redes Neurais Convolucionais (CNN)

O algoritmo de redes neurais artificiais é o precurso da CNN. Um rede neural artificial é uma composição de funções f_n que tem como contra domínio algum $\mathbb{R}^m$. O seu domínio é dado pela dimensão dos dados disponíveis, por exemplo, se temos uma imagem de tamanho $10x10$, a dimensão do domínio é 100. Logo, a rede neural pode ser descrita como uma função $Ann : \mathbb{R}^p \to \mathbb{R}^m$

$$Ann(x) = f_n(...f_2(A_2 * f_1(A_1 * x + b_1) + b_2), \tag{5.1}$$

onde A_i é uma matrix de tamanho arbitrário e $b_i \in \mathbb{R}$. Na Figura 17, temos uma imagem clássica para redes neurais.

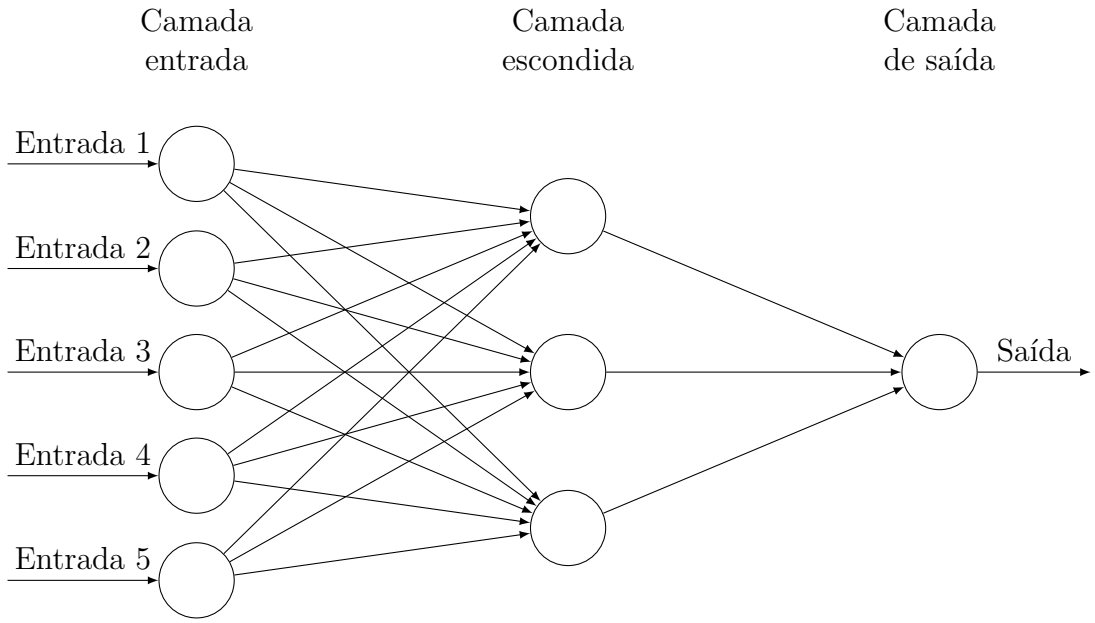

Figura 17 – Esquema de uma rede neural artificial. O número de vértices na camada escondida é determinado pelo tamanho da matriz A_i

5.2 Imagens de persistência aplicadas a proteínas

CAPÍTULO

6

CONCLUSÃO

REFERÊNCIAS

ADAMS, H.; EMERSON, T.; KIRBY, M.; NEVILLE, R.; PETERSON, C.; SHIPMAN, P.; CHEPUSHTANOVA, S.; HANSON, E.; MOTTA, F.; ZIEGELMEIER, L. Persistence images: A stable vector representation of persistent homology. **Journal of Machine Learning Research**, v. 18, n. 8, p. 1–35, 2017. Disponível em: <http://jmlr.org/papers/v18/16-337.html>. Citado nas páginas 39 e 40.

BUBENIK, P. Statistical topological data analysis using persistence landscapes. **Journal of Machine Learning Research**, v. 16, n. 3, p. 77–102, 2015. Disponível em: <http://jmlr.org/papers/v16/bubenik15a.html>. Citado na página 39.

CORTES, C.; VAPNIK, V. Support-vector networks. **Machine Learning**, Springer Nature, v. 20, n. 3, p. 273–297, set. 1995. Disponível em: <https://doi.org/10.1007/bf00994018>. Citado na página 45.

EDELSBRUNNER, H. **Computational topology : an introduction**. Providence, R.I: American Mathematical Society, 2010. ISBN 0821849255. Citado nas páginas 25, 30 e 32.

EDELSBRUNNER, H.; LETSCHER, D.; ZOMORODIAN, A. Topological persistence and simplification. In: **Proceedings 41st Annual Symposium on Foundations of Computer Science**. IEEE Comput. Soc, 2000. Disponível em: <https://doi.org/10.1109/sfcs.2000.892133>. Citado na página 32.

GOODFELLOW, I. J.; POUGET-ABADIE, J.; MIRZA, M.; XU, B.; WARDE-FARLEY, D.; OZAIR, S.; COURVILLE, A.; BENGIO, Y. Generative adversarial nets. In: **Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2**. Cambridge, MA, USA: MIT Press, 2014. (NIPS'14), p. 2672–2680. Disponível em: <http://dl.acm.org/citation.cfm?id=2969033.2969125>. Citado na página 45.

LECUN, Y.; BOSER, B.; DENKER, J. S.; HENDERSON, D.; HOWARD, R. E.; HUBBARD, W.; JACKEL, L. D. Backpropagation applied to handwritten zip code recognition. **Neural Computation**, MIT Press - Journals, v. 1, n. 4, p. 541–551, dez. 1989. Disponível em: <https://doi.org/10.1162/neco.1989.1.4.541>. Citado na página 45.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The Bulletin of Mathematical Biophysics**, Springer Nature, v. 5, n. 4, p. 115–133, dez. 1943. Disponível em: <https://doi.org/10.1007/bf02478259>. Citado na página 45.

POINCARé, H. Analysis situs. **Journal de l'École Polytechnique**, p. 1–123, 1895. Citado na página 21.

APÊNDICE

A

ALGORITMO *STANDARD* E FUNÇÕES AUXILIARES

```
# Retorna o valor low(column) descrito no texto.
# Exemplo:
# a = [1,1,0,0,1,0]
# get_low(a) retorna 5, maior índice tal que o valor de a nesse índice é 1.
function get_low(column)
    a = findlast(column .== 1)
    if a == nothing
        return 0
    else
        return a
    end
end

# Algoritmo Standard. Input: Matriz bordo P. Output: Matriz reduzida R.
function reduce_matrix(boundary)
    # número de simplexos no complexo simplicial
    nb_simplex = size(boundary, 2)
    # matriz R
    reduced = boundary
    # matrix V
    cycles = one(boundary)
    # Acessa cada coluna para reduzila
    lowest_ones = []
    for col in 1:nb_simplex
        # obtem os low(i) para i < col
        lowest_ones = [get_low(reduced[:,k]) for k in 1:(col-1)]
        # enquanto tiver algum j_0 tal que low(j_0) = low(col) e low(col) != 0, reduza a matriz
        while sum(get_low(reduced[:,col]) .== lowest_ones) != 0 && get_low(reduced[:,col]) != 0
            # opera a soma nas colunas que possuem o mesmo valor low.
            for k in 1:length(lowest_ones)
                if get_low(reduced[:,k]) == get_low(reduced[:,col])
                    reduced[:,col] = rem.(reduced[:,col] + reduced[:,k], 2)
```

```
                    cycles[:,col]  = rem.(cycles[:,col] + cycles[:,k], 2)
                end
            end
        end
    end
    # retorna a matriz reduzida R e a matriz V
    return reduced, cycles
end

# E por fim a matriz do nosso exemplo
P = [0 0 0 0 1 1 0 0 0 0;
     0 0 0 0 1 0 1 1 0 0;
     0 0 0 0 0 0 0 1 1 0;
     0 0 0 0 0 1 1 0 1 0;
     0 0 0 0 0 0 0 0 0 0;
     0 0 0 0 0 0 0 0 0 0;
     0 0 0 0 0 0 0 0 0 1;
     0 0 0 0 0 0 0 0 0 1;
     0 0 0 0 0 0 0 0 0 1;
     0 0 0 0 0 0 0 0 0 0]

# aplicando o algoritmo de redução
R, V = reduce_matrix(P)
```


ICMC
USP
SÃO CARLOS