

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

DIPLOMOVÁ PRÁCE

Algebraická topologie v analýze dat



2019

David Kočíř

Vedoucí práce:
doc. RNDr. Michal Krupka, Ph.D.

Studijní obor:
Informatika, prezenční forma

Bibliografické údaje

Autor: David Kočič
Název práce: Algebraická topologie v analýze dat
Typ práce: diplomová práce
Pracoviště: Katedra informatiky, Přírodovědecká fakulta,
Univerzita Palackého v Olomouci
Rok obhajoby: 2019
Studijní obor: Informatika, prezenční forma
Vedoucí práce: doc. RNDr. Michal Krupka, Ph.D.
Počet stran: 102
Přílohy: 1 CD/DVD
Jazyk práce: český

Bibliographic info

Author: David Kočič
Title: Algebraic topology in data analysis
Thesis type: master thesis
Department: Department of Computer Science, Faculty of Science,
Palacký University Olomouc
Year of defense: 2019
Study field: Computer Science, full-time form
Supervisor: doc. RNDr. Michal Krupka, Ph.D.
Page count: 102
Supplements: 1 CD/DVD
Thesis language: Czech

Anotace

Využití topologie je moderním trendem v analýze dat. Tato práce se konkrétně věnuje využití algebraické topologie v analýze dat: obsahuje úvod do algebraické topologie, popisuje základní metody algebraické topologie a také tato práce obsahuje několik původních analyzovaných příkladů.

Synopsis

Application of topology in data analysis is being a modern trend. This thesis deals with the use of algebraic topology in data analysis. It contains an introduction to algebraic topology and describes basic methods of algebraic topology. This thesis also contains some original analyzed examples.

Klíčová slova: topologie, algebraická topologie, simplicialní homologie, topologická analýza dat

Keywords: topology, algebraic topology, simplicial homology, topological data analysis

Děkuji vedoucímu práce za ochotu a cenné rady při konzultacích.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce

podpis autora

Obsah

1	Úvod	13
2	Obecná topologie	14
2.1	Topologické prostory	14
2.2	Homeomorfismy topologických prostorů	16
3	Algebra	19
3.1	Grupy, okruhy a pole	19
3.2	Moduly	22
3.3	Gradované okruhy a gradované moduly	25
4	Algebraická topologie	27
4.1	Homotopie	28
4.2	Simpliciální komplexy	32
4.3	Simpliciální homologie	39
4.4	Persistentní homologie	47
5	Ripsův komplex	52
6	Algoritmy simpliciální homologie	54
6.1	Algoritmus simpliciální homologie	54
6.2	Algoritmus persistentní homologie	61
7	Ukázka jednoduchého použití simpliciální homologie	70
7.1	Dataset Mall	70
7.2	Dataset US Senate Votes	73
8	Algoritmus Mapper	75
8.1	Topologická verze	75
8.2	Statistická verze	78
9	Příklady použití statistického Mapperu	82
9.1	Dataset World Happiness Report	82
9.2	Dataset World University Rankings	86
9.3	Dataset Hayes Roth	89
9.4	Porovnání statistického Mapperu se shlukováním	94
	Závěr	97
	Conclusions	98
A	Poznámky k vlastní implementaci a použitému software	99

B Obsah přiloženého CD/DVD	100
Literatura	101

Seznam obrázků

2.1	Půlkružnice bez krajních bodů P a Q je homeomorfní s přímkou.	17
4.1	Příklad homotopie funkcí. Funkce f je homotopická s funkcí g .	29
4.2	Příklad homotopie oblouků.	30
4.3	Příklady simplexů s jejich názvy.	33
4.4	Příklad geometrického simplicialního komplexu. Tento simplicialní komplex obsahuje tři 2-simplexy, nebo-li trojúhelníky a několik 1-simplexů, nebo-li hran. Všimněme si, že dva trojúhelníky nalevo společně sdílí jednu stejnou hranu a dva stejné vrcholy, zatímco trojúhelník vpravo nesdílí žádnou ze svých stěn s žádným dalším simplexem.	34
4.5	Příklad množiny simplexů, která však netvoří simplicialní komplex. Jednomu z trojúhelníků vlevo chybí jedna stěna. U trojúhelníků vpravo jsou sice v množině obsaženy všechny jejich stěny, ale jejich průnikem není společná stěna, nýbrž pouze část společné hrany.	34
4.6	Simplicialní komplex K spolu s homeomorfismem $f : K \rightarrow X$ tvoří triangulaci jednotkové kružnice. Na jednotkové kružnici jsou zvýrazněny obrazy homeomorfismu f odpovídající jednotlivým simplexům ze simplicialního komplexu K .	35
4.7	Simplicialní komplex s jednou jednodimenzionální dírou.	45
5.1	Příklad Ripsovy filtrace s parametry $\epsilon_1 < \epsilon_2 < \epsilon_3$. Na obrázku (a) je původní množina, na obrázcích (b) - (d) je znázorněna podoba Ripsova komplexu s daným parametrem ϵ_i . Na obrázku (e) jsou čárové kódy dimenze 1 k uvedené Ripsově filtraci. Pro lepší čitelnost jsou úsečky tučné a střídavě obarveny.	53
7.1	Čárové kódy datasetu Mall pro dimenzi 1. Na obrázku je zobrazeno prvních 10 homologických tříd s největší persistencí.	71
7.2	Ripsův komplex vytvořený z datasetu Mall s hodnotou parametru $\epsilon = 0.2$.	72
9.1	Výstup statistického Mapperu aplikovaného na dataset World Happiness Report. Filtrem je funkce aproximace hustoty s parametrem 0.2, otevřené pokrytí je dáno velikostí intervalu $1/7$ z celkové velikosti intervalu hodnot filtru s překryvem 50% a pro shlukování je zvolen parametr 0.45. Na obrázku (a) jsou uzly obarveny na základě hodnoty filtru: čím vyšší je průměrná hodnota filtru objektů v daném uzlu, tím je daný uzel obarven červeněji. Modře jsou pak obarveny nižší hodnoty. Na obrázku (b) je výstup pro stejné parametry a uzly jsou obarveny na základě průměrné hodnoty štěstí objektů v daném uzlu, červená barva značí vyšší hodnoty a modrá nižší. Vidíme, že graf má jeden centrální uzel, z kterého vychází další uzly spojené hranou. Těmto komponentám se říká <i>světlice</i> .	84

9.2	Výstup statistického Mapperu aplikovaného na dataset World Happiness Report: filtrem je funkce aproximace hustoty s parametrem 0.2, otevřené pokrytí je dáno velikostí intervalu $1/5$ z celkové velikosti intervalu hodnot filtru s překryvem 45% a pro shlukování je zvolen parametr 0.45. Obarvení uzlů na obrázcích (a) i (b) je stejné jako v předchozím příkladě na obrázku 9.1	85
9.3	Výstup statistického Mapperu aplikovaného na dataset World Happiness Report. Filtrem je funkce aproximace hustoty s parametrem 0.2, otevřené pokrytí je dáno velikostí intervalu $1/10$ z celkové velikosti intervalu hodnot filtru s překryvem 50% a pro shlukování je zvolen parametr 0.3. Obarvení uzlů na obrázcích (a) i (b) je stejné jako v předchozích dvou příkladech.	86
9.4	Výstup algoritmu Mapper pro dataset World University Rankings. Jako filtr je použita projekce na hodnotu atributu <code>World-rank</code> , parametr ϵ pro shlukování zvolen 0.65. Na obrázku (a) je pokrytí zadané velikostí intervalu $1/7$ z celkové velikosti intervalu hodnot filtru s velikostí překryvu 45% a na obrázku (b) je zadané velikostí intervalu $1/10$ se stejným překryvem. Uzly na obou obrázcích jsou obarveny na základě hodnoty filtru, tedy hodnoty atributu <code>World-rank</code> : čím je uzel červenější, tím je hodnota tohoto atributu vyšší.	88
9.5	Na obrázku (a) je výstup s parametry velikostí intervalu $1/3$ z celkové velikosti intervalu hodnot filtru, překryvem 50% a vzdáleností pro shlukování 3. Velký červený uzel a s ním spojené malé žlutozelené uzly obsahují pouze ty objekty, u kterých byla třída určena na základě počtu 1 a 2. Zelený uzel vpravo obsahuje pouze objekty třídy 3. Světlejší modrý uzel obsahuje objekty třídy 3 a ty objekty, které obsahují stejný počet 1 a 2 a třída u nich tedy byla určena náhodně. Poslední tmavě modrý uzel obsahuje navíc objekty třídy 1 a 2, které neobsahují buďto žádnou jedničku, nebo žádnou dvojku. Na obrázku (b) je výstup s parametry velikostí intervalu $1/3$ z celkové velikosti intervalu hodnot filtru, překryvem 50% a vzdáleností pro shlukování 3, kde pro výpočet vzdáleností při shlukování není použita hodnota atributu <code>club</code> . Velký červený uzel a s ním spojené malé žlutozelené uzly obsahují pouze ty objekty, u kterých byla třída určena na základě počtu 1 a 2, přičemž každý objekt v těchto uzlech obsahuje nejméně jednu 1 nebo jednu 2. Zelený uzel vlevo obsahuje pouze objekty třídy 3. Světlejší modrý uzel uprostřed obsahuje objekty třídy 3 a pak ty objekty třídy 1 nebo 2, které mají stejný počet 1 a 2 a tudíž u nich byla třída určena náhodně. Největší, tmavě modrý uzel pak ještě navíc obsahuje ty objekty třídy 1 nebo 2, které obsahují maximálně jednu 1 nebo jednu 2.	91

- 9.6 U této analýzy je navíc přidána podmínka, že objekty se mohou sloučit do stejného shluku jen tehdy, pokud mají stejnou třídu. Na obrázku (a) vlevo je výstup Mapperu s parametry velikostí intervalu $1/3$ z celkové velikosti intervalu hodnot filtru, překryvem 50% a vzdáleností pro shlukování 3. Velké červené uzly a žluté uzly s nimi spojené hranou obsahují pouze ty objekty třídy 1 nebo 2, podle toho, jestli daný objekt obsahuje více 1 nebo 2. Tmavší modré uzly a s nimi vždy tři spojené světle modré uzly obsahují převážně ty objekty, které obsahují stejný počet 1 a 2, a tudíž u nich byla třída určena náhodně. Zbylé tři uzly, tmavě modrý, světle modrý a zelený pak obsahují objekty třídy 3. Na obrázku (b) vpravo je výstup s parametry velikostí intervalu $1/2$ z celkové velikosti intervalu hodnot filtru, překryvem 50% a vzdáleností pro shlukování 3. V každé ze dvou trojic uzlů spojených hranou velké červené uzly obsahují ty objekty, u kterých byla třída určena na základě počtu 1 a 2. Menší tmavě modré uzly v těchto trojicích obsahují objekty, u kterých byla třída určena náhodně. Světle modré uzly uprostřed pak obsahují oba typy objektů. Modrý a zelený uzel dole obsahuje pouze objekty třídy 3. 92
- 9.7 Otevřené pokrytí je dané velikostí intervalu $1/7$ z celkové velikosti intervalu hodnot filtru s překryvem 33%. Pro shlukování byly vybrány všechny tři prvkové kombinace atributů hobby, věk, vzdělání a rodinný stav. Výstupní graf Mapperu je rozdělený na obrázcích (a) - (f) a obarvení uzlů je stejné jako na předchozích dvou obrázcích. Uzly na obrázku (a) obsahují objekty třídy 1 a 2, u kterých byla třída určena na základě počtu 1 a 2, přičemž každý objekt obsahuje nejméně jednu 1 nebo jednu 2. Na obrázku (b) uzly obsahují objekty třídy 1 a 2, u kterých byla třída určena na základě počtu 1 a 2, avšak každý objekt obsahuje buď nulový počet 1, nebo nulový počet 2. Na obrázku (d) uzly obsahují objekty třídy 1 a 2, které obsahují stejný počet 1 a 2, a tudíž u nich byla třída určena náhodně. Uzly na obrázcích (c) a (e) obsahují pouze objekty třídy 3. Uzly na obrázku (f) pak obsahují objekty třídy 3 a ty objekty třídy 1 nebo 2, které však obsahují právě jednu 1, nebo jednu 2. 93

- 9.8 Výsledek shlukování na datasetu Hayes Roth. Na obrázku (a) je výsledek pro parametr $\epsilon = 1$. Pro tento parametr je výsledek shlukování také triviální. Díky tomu, že hodnota atributu jméno je pro každý objekt unikátní, se při parametru $\epsilon = 1$ sloučí do jednoho shluku právě ty objekty, které mají shodné hodnoty na všech ostatních atributech. Na obrázku (b) je výsledek shlukování s parametrem $\epsilon = 2$. Největší shluk obsahuje všechny objekty třídy 1 a 2. Ostatní menší uzly pak obsahují různě pouze objekty třídy 3. Na obrázku (c) je výsledek shlukování pro parametr $\epsilon = 3$ a při tomto parametru jsou již všechny objekty sloučeny do jednoho shluku. 94
- 9.9 Výsledek shlukování na datasetu Hayes Roth bez použití hodnot atributu Klub. Na obrázku (a) je výsledek pro parametr $\epsilon = 1$. Do jednoho shluku se sloučí ty objekty, které kromě atributu jméno mají stejné hodnoty na ostatních atributech. Na obrázku (b) je výsledek pro parametr $\epsilon = 2$ a při této hodnotě se již všechny objekty sloučí do jediného uzlu. 95

Seznam tabulek

6.1	Grafické znázornění filtrace simplicialního komplexu. Nově přidávané simplexu jsou zvýrazněny modrou barvou, kroky přidání a označení nově přidávaného simplexu jsou napsány v řádcích pod obrázky simplicialních komplexů.	63
6.2	Stupně jednotlivých simplexů ve filtraci simplicialního komplexu z tabulky 6.1.	63
9.1	Popis použitých atributů datasetu World Happiness. Posledních 6 atributů jsou faktory. Faktory jsou sestaveny tak, že se pokouší zdůvodnit míru štěstí v dané zemi a platí, že součet hodnot všech faktorů se rovná hodnotě atributu Happiness-Score.	83
9.2	Atributy datasetu World University Rankings s krátkým popisem.	87

Seznam algoritmů

6.1	Redukce matice hraničního operátoru do sloupcově normovaného tvaru s koeficienty v \mathbb{Z}_2	60
6.2	Pseudokód algoritmu pro redukci řetězce. Algoritmus využívá jednotlivé hraniční operátory negradovaných modulů k -řetězců a při redukci počítá inverzi k prvku q v poli F	67
6.3	Pseudokód algoritmu pro výpočet \mathcal{P} -intervalů filtrovaného simplicialního komplexu. K redukci řetězců využívá proceduru <code>reduce Chain</code> znázorněnou algoritmem 6.2. Kroky algoritmu jsou podrobněji popsány v textu výše.	68
8.1	Topologická verze algoritmu <code>Mapper</code> . Význam parametrů je podrobněji popsán v textu výše.	76

1 Úvod

Cílem této práce je nastudovat základy obecné a algebraické topologie a podat souhrn vybraných aplikací algebraické topologie v analýze dat s důrazem na původní příklady.

První dvě kapitoly této práce se tedy věnují základním pojmům, které jsou nutné pro porozumění algebraické topologie. Kapitola 2 obsahuje základy obecné topologie a také slouží jako úvod do obecné topologie. Kapitola 3 shrnuje potřebné pojmy z algebry, které jsou nutné ke studiu algebraické topologie.

Kapitola 4 se věnuje základům algebraické topologie. Z celkové oblasti algebraické topologie jsem vybral ta témata, která v současné době nachází praktické uplatnění v analýze dat. Kapitola 5 pak navazuje na kapitolu 4 zavedením postupu, který rozšiřuje možnosti použití algebraické topologie v analýze dat.

V kapitole 6 jsou pak popsány základní algoritmy algebraické topologie a v kapitole 7 je ukázka jednoduchého použití těchto algoritmů při analýze dat na dvou původní příkladech.

Kapitola 8 se věnuje algoritmu *Mapper*, který se v současné době pomalu stává standardním nástrojem topologické analýzy dat. Následující kapitola 9 obsahuje původní praktické příklady využití tohoto algoritmu s krátkým srovnáním výsledků algoritmu Mapper s výsledkem velmi podobné metody analýzy dat: *shlukováním*.

V příloze A se nachází stručný popis vlastní implementace algoritmů příložených na CD a také výčet použitého software.

2 Obecná topologie

Tato kapitola slouží jako úvod do obecné topologie. Jsou zde zmíněny všechny důležité pojmy, které v této práci budou potřeba. Informace jsou čerpány ze zdroje [7].

2.1 Topologické prostory

Definice 2.1 (Topologie)

Topologie \mathcal{T} na množině X je systém podmnožin X splňující následující podmínky:

1. $\emptyset \in \mathcal{T}, X \in \mathcal{T}$.
2. Pokud $U, V \in \mathcal{T}$, pak i $U \cup V \in \mathcal{T}$.
3. Pokud $\Upsilon \subseteq \mathcal{T}$, pak i $\bigcap_{U \in \Upsilon} U \in \mathcal{T}$.

Uvažujeme tedy sjednocení libovolného počtu množin a průniky konečného počtu množin.

Definice 2.2 (Topologický prostor)

Množina X , na níž je dána topologie \mathcal{T} , se nazývá *topologický prostor*. Topologický prostor budeme značit pouze X .

Definice 2.3 (Otevřená množina a okolí bodu)

Mějme topologický prostor X s topologií \mathcal{T} . Množiny $U \in \mathcal{T}$ se nazývají *otevřené množiny na X* (zjednodušeně pouze *otevřené množiny*). Pokud U je otevřená množina a $x \in U$, kde $x \in X$, pak se množina U nazývá *okolí bodu x* .

PŘÍKLAD 2.4

Uvažujme množinu $X = \{a, b, c, d\}$. Celkem na této množině můžeme zavést 355 různých topologií. Některé z nich si ukážeme.

Nejjednodušší topologie je takzvaná *triviální topologie*. To je topologie, která obsahuje pouze množinu X a prázdnou množinu, tedy $\mathcal{T} = \{\emptyset, X\}$.

Dalším způsobem pak může být takzvaná *diskrétní topologie*. To je taková topologie, pro kterou platí $\mathcal{T} = 2^X$. V topologickém prostoru s diskrétní topologií je každá jednoprvková množina otevřená. V tomto případě jsou tedy množiny $\{a\}, \{b\}, \{c\}, \{d\}$ otevřené.

Dále pak můžeme uvažovat například topologie $\mathcal{T} = \{\emptyset, \{a, b\}, \{c, d\}, X\}$, $\mathcal{T} = \{\emptyset, \{a\}, \{d\}, \{a, d\}, \{b, c, d\}, X\}$ nebo $\mathcal{T} = \{\emptyset, \{c\}, \{c, d\}, \{a, b, c\}, X\}$.

Naopak, systém množin $\mathcal{E} = \{\emptyset, \{a\}, \{d\}, \{b, c\}, X\}$ netvoří topologii na X . Dle definice topologie by totiž systém množin \mathcal{E} měl obsahovat například množiny $\{a, d\}, \{a, b, c\}$ a $\{b, c, d\}$, které však \mathcal{E} neobsahuje.

Definice 2.5 (Topologický podprostor)

Mějme topologický prostor X s topologií \mathcal{T} , vezměme libovolnou podmnožinu $A \subseteq X$ a definujme systém množin

$$\mathcal{T}_A = \{\tau \cap A \mid \tau \in \mathcal{T}\}.$$

Systém \mathcal{T}_A je topologií na A a množina A spolu s topologií \mathcal{T}_A tedy tvoří topologický prostor. Tento topologický prostor se nazývá *topologický podprostor topologického prostoru X* .

Zjednodušeně pak také říkáme, že na $A \subseteq X$ uvažujeme *indukovanou topologii z X* . Tím se pak myslí přesně ta topologie \mathcal{T}_A , která je definovaná výše.

Často je také užitečné určit topologii pomocí báze:

Definice 2.6 (Báze topologie, Topologie generovaná bází)

Mějme množinu X . *Bází topologie* na X je právě takový systém \mathcal{B} podmnožin X , pro který platí:

1. Pro každé $x \in X$ existuje alespoň jeden bázový prvek $B \in \mathcal{B}$ takový, že $x \in B$.
2. Pokud x náleží do průniku dvou bázových prvků B_1 a B_2 , pak musí existovat bázový prvek B_3 takový, pro který platí $x \in B_3$ a zároveň $B_3 \subseteq B_1 \cup B_2$.

Topologie \mathcal{T} generovaná bází \mathcal{B} je definovaná následovně: množina $U \subseteq X$ je prvkem \mathcal{T} , jestliže pro každé $x \in U$ existuje bázový prvek $B \in \mathcal{B}$ takový, pro který platí $x \in B$ a $B \subseteq U$.

Následující lemma říká, jak pomocí báze \mathcal{B} můžeme topologii \mathcal{T} na X popsat přehledně:

Lemma 2.7 (O generování topologie pomocí báze)

Nechť X je množina a \mathcal{B} je báze topologie \mathcal{T} na X . Pak je topologie \mathcal{T} rovna systému množin obsahující všechna možná sjednocení množin z \mathcal{B} .

PŘÍKLAD 2.8

Na \mathbb{R} uvažujeme takzvanou *standardní topologii*, která je generována bází obsahující všechny otevřené intervaly ve tvaru

$$(a, b) = \{x \mid a < x < b\}.$$

Na \mathbb{R}^n pak uvažujeme topologii generovanou bází, která obsahuje všechny součiny všech otevřených intervalů ve tvaru

$$(a_1, b_1) \times (a_2, b_2) \times \cdots \times (a_n, b_n).$$

Na množině $A \subseteq \mathbb{R}$ pak uvažujeme topologii indukovanou z \mathbb{R} a na množině $A^n \subseteq \mathbb{R}^n$ topologii indukovanou z \mathbb{R}^n .

Pokud nebude uvedeno jinak, v této práci budeme na množinách \mathbb{R} , \mathbb{R}^n a na jejich podmnožinách vždy uvažovat tyto uvedené topologie.

2.2 Homeomorfismy topologických prostorů

Důležitým pojmem v topologii je takzvané *spojité zobrazení*:

Definice 2.9 (Spojité zobrazení)

Nechť X a Y jsou topologické prostory. Zobrazení $f : X \rightarrow Y$ se nazývá *spojité*, pokud pro každou otevřenou množinu $V \in Y$ platí, že množina $f(V)^{-1}$ je otevřená na X .

Definice 2.10 (Homeomorfismus)

Nechť X a Y jsou topologické prostory. Zobrazení $f : X \rightarrow Y$ se nazývá *homeomorfismus*, pokud splňuje následující podmínky:

1. Zobrazení f je spojité.
2. Zobrazení f je bijektivní a inverzní zobrazení $f^{-1} : Y \rightarrow X$ je spojité.

Z definice homeomorfismu také plyne, že množina $f(U)$ je otevřená v Y právě tehdy, když U je otevřená v X . Tudíž homeomorfismus je bijekcí nejen mezi body v X a Y , ale také mezi otevřenými množinami na X a Y .

Pokud mezi topologickými prostory X a Y existuje homeomorfismus, pak se tyto topologické prostory nazývají *homeomorfní*. Homeomorfní prostory značíme $X \cong Y$.

Pokud dva topologické prostory X a Y jsou homeomorfní, pak říkáme, že X a Y jsou *stejněho topologického typu* (někdy se také říká, že mají stejné topologické vlastnosti).

Každý topologický prostor je homeomorfní sám se sebou. Za homeomorfismus stačí vzít identitu. Pokud existuje homeomorfismus f mezi X a Y , pak f^{-1} je homeomorfismus mezi Y a X . A nakonec, pokud existuje homeomorfismus f mezi X a Y a homeomorfismus g mezi Y a Z , pak složené¹ zobrazení $(g \circ f)$ je homeomorfismus mezi X a Z . Homeomorfismus je tedy relací ekvivalence na množině topologických prostorů.

PŘÍKLAD 2.11

Interval $(-1, 1)$ je homeomorfní s \mathbb{R} . Jako homeomorfismus z \mathbb{R} na $(-1, 1)$ můžeme vzít například funkci $f(x) = \tanh(x)$. Tato funkce je spojitá a existuje k ní spojitá inverzní funkce $f(x)^{-1} = \arg \tanh(x)$.

PŘÍKLAD 2.12

Libovolné dva otevřené intervaly jsou homeomorfní. Jako homeomorfismus stačí vzít naškálování jednoho intervalu na druhý. Každý otevřený interval (a, b) je tedy homeomorfní i s intervalem $(-1, 1)$. Z předchozího příkladu pak vyplývá, že libovolný otevřený interval je homeomorfní s \mathbb{R} .

¹Složené zobrazení $g \circ f$ je definováno předpisem $(g \circ f)(x) = g(f(x))$.

PŘÍKLAD 2.13

Lze dokázat, že žádný uzavřený interval $[a, b]$ na \mathbb{R} není homeomorfní s \mathbb{R} .

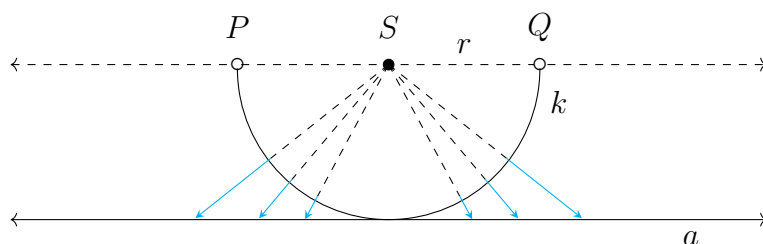
PŘÍKLAD 2.14

Uvažujme půlkružnici k se středem v bodě S o poloměru r a necht P a Q jsou její koncové body. Dále vezměme tečnu a k půlkružnici k takovou, že je rovnoběžná s přímkou vedenou koncovými body P a Q půlkružnice k .

Nyní uvažujme takovou projekci bodů z této půlkružnice, že každý bod se promítne ze středu S na přímkou a . Kromě bodů P a Q se tedy každý bod půlkružnice promítne do jednoho jediného bodu na přímce a .

Uvažujme na obou geometrických útvech topologii indukovanou z \mathbb{R}^2 . Pokud z půlkružnice odebereme její koncové body P a Q , pak je tato půlkružnice homeomorfní s přímkou a . Naopak, celá půlkružnice, tedy včetně svých krajních bodů P a Q , je homeomorfní s úsečkou. Tento homeomorfismus se často interpretuje tak, že půlkružnici lze „napřímít“.

Příklad je převzatý z [10] a je znázorněn na obrázku 2.1.



Obrázek 2.1: Půlkružnice bez krajních bodů P a Q je homeomorfní s přímkou.

PŘÍKLAD 2.15

Uvažujme zprava otevřený interval $[0, 2\pi)$ a jednotkovou kružnici. Díky tomu, že je interval zprava otevřený, existuje mezi ním a jednotkovou kružnicí bijekce. Nicméně, lze dokázat, že žádné bijektivní zobrazení z jednotkové kružnice na interval $[0, 2\pi)$ není spojitě. Tento interval a jednotková kružnice tedy nejsou homeomorfní.

Obecně platí, že žádná kružnice není homeomorfní s libovolným intervalem v \mathbb{R} , popřípadě přímkou, polopřímkou či úsečkou v rovině.

PŘÍKLAD 2.16

Necht $x = (x_1, \dots, x_n) \in \mathbb{R}^n$. Pro $p \in \mathbb{R}$, $p \geq 1$ definujeme p -normu x předpisem

$$\|x\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}.$$

Pomocí 2-normy je definována takzvaná *n-dimenzionální sféra*

$$\mathbb{S}^n = \{x \in \mathbb{R}^{n+1} \mid \|x\|_2 = 1\}.$$

Na sféře \mathbb{S}^n jako obvykle uvažujeme indukovanou topologii z \mathbb{R}^{n+1} .

Sféra \mathbb{S}^1 je pak homeomorfní s obvodem libovolného čtverce v \mathbb{R}^2 , sféra \mathbb{S}^2 je pak homeomorfní s povrchem libovolné krychle v \mathbb{R}^3 . Tento homeomorfismus pak platí i pro zobecnění krychle do vyšších dimenzí.

3 Algebra

Předtím, než se podíváme na algebraickou topologii, je nutné uvést několik pojmů a poznatků z algebry. Hlavním cílem této kapitoly je zavést moduly, gradované moduly a posléze uvést větu o struktuře modulů a větu o struktuře gradovaných modulů. Jak uvidíme později, tyto věty v algebraické topologii hrají klíčovou roli.

3.1 Grupy, okruhy a pole

Tato podkapitola obsahuje stručné připomenutí základních pojmů z algebry, které budou dále potřeba. Pojem *operace* se zde používá ve volnějším významu. Informace jsou čerpány ze zdroje [12].

Definice 3.1 (Grupa)

Množina G spolu s binární operací $+$ na G se nazývá *grupa*, jestliže operace je $+$ asociativní a dále platí:

1. Existuje prvek $e \in G$ takový, že pro všechna $x \in G$ platí:

$$e + x = x + e = x,$$

prvek e se nazývá *neutrální prvek* vzhledem k operaci $+$ na G .

2. Pro všechny prvky $a \in G$ existuje takový prvek $-a$, pro který platí:

$$(-a) + a = a + (-a) = e,$$

prvek $-a$ se nazývá *inverzní prvek* k prvku a vzhledem k operaci $+$ na G .

Grupu značíme $(G, +)$, zjednodušeně pak pouze G .

Definice 3.2 (Abelovská grupa)

Grupa $(G, +)$ se nazývá *abelovská*, pokud je operace $+$ komutativní.

Definice 3.3 (Podgrupa)

Mějme grupu $(G, +)$. Podmnožina $H \subseteq G$ spolu s operací $+$ zúženou pouze na prvky H se nazývá *podgrupa grupy* G , pokud je H uzavřená na $+$.

Definice 3.4 (Přímý součet abelovských grup)

Mějme abelovskou grupu G a systém abelovských grup $\mathcal{A} = \{A_i\}_{i \in I}$, kde $I \subseteq \mathbb{Z}$. Pokud $\bigcup_{i \in I} A_i = G$ a pro každé $i \in I$ platí $A_i \cap \left(\bigcup_{j \in I, j \neq i} A_j\right) = \{0\}$, pak se grupa G nazývá *přímým součtem grup* $\{A_i\}_{i \in I}$. Přímý součet značíme následovně:

$$G = \bigoplus_i A_i, \text{ kde } i \in I.$$

U grupy, která je definována přímým součtem se její prvky zapisují také jako n -tice, které mohou být i nekonečné. Tedy například pro grupu $G = \bigoplus_i A_i$, kde $i \in I$ a pro prvek $a \in G$ píšeme (a_1, a_2, a_3, \dots) , kde $a_i \in A_i$. Operaci na G pak můžeme definovat po složkách pomocí operací z jednotlivých grup $\{A_i\}_{i \in I}$. Necht tedy $+^i$ značí operaci grupy A_i . Pro prvky $a, b \in G$ tedy definujeme operaci $+^G$ na G takto:

$$(a_1, a_2, a_3, \dots) +^G (b_1, b_2, b_3, \dots) = (a_1 +^1 b_1, a_2 +^2 b_2, a_3 +^3 b_3, \dots).$$

Definice 3.5 (Okruh)

Neprázdná množina R spolu s operacemi $+$ a \cdot se nazývá *okruh*, jestliže jsou splněny následující podmínky:

1. $(R, +)$ je abelovská grupa.
2. Operace \cdot je asociativní a dále je distributivní vzhledem k operaci $+$.

Okruh se značí $(R, +, \cdot)$, zjednodušeně pak pouze R . Operace \cdot se ve výrazech běžně vynechává.

Definice 3.6 (Jednotkový okruh, komutativní okruh)

Pokud okruh R obsahuje neutrální prvek vzhledem k operaci \cdot , pak se daný okruh nazývá *jednotkový* (někdy také *okruh s jednotkou*). Tento neutrální prvek se nazývá *jednotka* a značíme jej 1_R . Pokud je operace \cdot komutativní, pak se daný okruh nazývá *komutativní*.

PŘÍKLAD 3.7 (POLYNOMY A OKRUH POLYNOMŮ)

Necht $(R, +, \cdot)$ je komutativní okruh s jednotkou. *Polynom* $f(x)$ s koeficienty v R je formální suma ve tvaru

$$f(x) = a_0 + a_1x^1 + \dots + a_nx^n,$$

kde $a_i \in R$ a x říkáme *proměnná*. Číslo n se říká *stupeň polynomu*. Polynom pouze s jedním nenulovým koeficientem a_d stupně d ve tvaru a_dx^d se nazývá *homogenní*.

Množina všech polynomů $f(x)$ s koeficienty v R spolu s obvyklými operacemi sčítání a násobení polynomů tvoří komutativní okruh s jednotkou, který se značí $R[x]$.

Definice 3.8 (Ideál)

Nechť $(R, +, \cdot)$ je komutativní okruh. Neprázdňá podmnožina $I \subseteq R$ se nazývá *ideál*, pokud platí následující podmínky:

1. Pokud $a, b \in I$ pak $a + b \in I$.
2. Pokud $a \in I, r \in R$ pak $ar \in I$.

Pokud ideál I obsahuje jednotkový prvek, pak platí $I = R$.

Definice 3.9 (Ideál generovaný množinou)

Nechť $(R, +, \cdot)$ je okruh s jednotkou a mějme množinu $S \subseteq R$. *Ideál generovaný množinou S* je nejmenší ideál obsahující S . Značíme jej $\langle S \rangle$.

Definice 3.10 (Obor integrity)

Komutativní okruh R s jednotkou se nazývá *obor integrity*, pokud pro všechny prvky $a, b \in R$ platí, jestliže $a \neq 0$ a $b \neq 0$, pak $ab \neq 0$.

Definice 3.11 (Pole)

Množina F spolu s operacemi $+$ a \cdot se nazývá *pole*, pokud platí:

1. Trojice $(F, +, \cdot)$ tvoří komutativní okruh s jednotkou.
2. Pro každé $a \in F$ existuje prvek a^{-1} takový, že platí $aa^{-1} = 1_F$, kde 1_F je jednotka okruhu $(F, +, \cdot)$.

Věta 3.12

Každé pole je oborem integrity. Každý konečný obor integrity je pole.

PŘÍKLAD 3.13

Trojice $(\mathbb{Z}_p, +_p, \cdot_p)$, kde p je prvočíslo a operace $+_p, \cdot_p$ značí sčítání a násobení modulo p , jsou konečné obory integrity. Podle předchozí věty tedy také tvoří pole.

Definice 3.14 (Hlavní ideál a obor hlavních ideálů)

Nechť $(R, +, \cdot)$ je okruh s jednotkou a vezměme prvek $a \in R$. *Hlavní ideál* je ideál generovaný množinou obsahující pouze prvek a . Pokud $(R, +, \cdot)$ je obor integrity, jehož každý ideál je hlavní, pak se $(R, +, \cdot)$ nazývá *obor hlavních ideálů*.

PŘÍKLAD 3.15

Množiny $\mathbb{Z}, \mathbb{Z}_p, \mathbb{R}$ a \mathbb{Q} spolu s obvyklými operacemi tvoří obory hlavních ideálů. Pokud R je pole, pak okruh polynomů $R[x]$ je oborem hlavních ideálů.

3.2 Moduly

V této kapitole jsou definovány moduly a dále je zde uvedena věta o struktuře modulů. Informace jsou čerpány ze zdroje [12].

Definice 3.16 (Modul)

Nechť R značí okruh s jednotkou. R -modul (nebo také *modul M nad R*) je neprázdná množina M spolu s dalšími dvěma operacemi. První operace, značená $+$, každé dvojici $(u, v) \in M \times M$ přiřadí prvek $u + v \in M$. Druhá operace, značená \cdot , každé dvojici $(r, v) \in M \times R$ přiřadí prvek $rv \in M$. Dále musí být splněny následující podmínky:

1. $(M, +)$ tvoří abelovskou grupu.
2. Pro všechna $r, s \in R$ a $u, v \in M$ platí:

$$r(u + v) = ru + rv,$$

$$(r + s)u = ru + su,$$

$$(rs)u = r(su),$$

$$1_R u = u.$$

Prvky R se běžně označují jako *skaláry*. Operaci $+$ budeme říkat sčítání a operaci \cdot násobení, popřípadě násobení skalárem.

Modul se nazývá *triviální*, pokud $M \cong \{0\}$. Triviální modul značíme zjednodušeně pouze 0 . Pokud modul není triviální, označuje se také jako *nenulový*.

Definice modulu připomíná vektorový prostor. Skutečně, pokud R je pole, pak R -modul tvoří vektorový prostor. Vektorové prostory jsou tedy speciálním případem modulů nad polem.

Definice 3.17 (Podmodul)

Podmodul R -modulu M je neprázdná podmnožina $S \subseteq M$, která tvoří R -modul spolu s operacemi $+$ a \cdot zúženými pouze na prvky množiny S .

Definice 3.18 (Modul generovaný množinou)

Mějme R -modul M a necht $S \subseteq M$. *Podmodul generovaný množinou S* je nejmenší modul obsahující množinu S . Tento modul značíme $\langle S \rangle$. O množině $S \subseteq M$ řekneme, že *generuje* modul M , pokud $M = \langle S \rangle$.

Definice 3.19 (Cyklický modul)

Nechť M je R -modul a $v \in M$. Podmodul generovaný množinou obsahující pouze prvek v se nazývá *cyklický modul* generovaný prvkem v .

Definice 3.20 (Konečně generovaný modul)

R -modul M se nazývá *konečně generovaný*, pokud obsahuje konečnou množinu, která jej generuje.

Definice 3.21 (Lineární nezávislost)

Prvky množiny $S \subseteq M$ se nazývají *lineárně nezávislé*, pokud pro všechny navzájem různé prvky $v_1, \dots, v_n \in S$ a libovolné skaláry $r_1, \dots, r_n \in R$ platí:

$$\text{pokud } r_1v_1 + \dots + r_nv_n = 0, \text{ pak } r_i = 0 \text{ pro všechna } i = 1, \dots, n.$$

Pokud prvky nejsou lineárně nezávislé, pak se nazývají *lineárně závislé*.

Definice 3.22 (Báze a volný modul)

Nechť M je R -modul. Množina \mathfrak{B} se nazývá *báze*, pokud její prvky jsou lineárně nezávislé a \mathfrak{B} generuje M . R -modul M se nazývá *volný modul*, pokud má bázi.

Prvky volného modulu M nad R s konečnou bází $\mathfrak{B} = \{\mathfrak{b}_1, \dots, \mathfrak{b}_m\}$ lze vyjádřit také pomocí formálních lineárních kombinací prvků této báze s koeficienty v R . Každý prvek $v \in M$ tedy lze zapsat ve tvaru:

$$v = \sum_{i=1}^m a_i \mathfrak{b}_i, \text{ kde } a_i \in R, \mathfrak{b}_i \in \mathfrak{B}.$$

Definice 3.23 (Hodnost modulu)

Nechť R je komutativní okruh s jednotkou. *Hodnost* volného R -modulu nad M je definována jako počet prvků jeho báze. Hodnost modulu se značí $\text{rank}(M)$.

Definice 3.24 (R -homomorfismus)

Nechť M a N jsou R -moduly. Zobrazení $\tau : M \rightarrow N$ se nazývá *R -homomorfismus*, pokud platí:

$$\tau(ru + sv) = r\tau(u) + s\tau(v),$$

pro všechna $r, s \in R$ a $u, v \in M$. Pokud R -homomorfismus je bijektivní, nazývá se *R -izomorfismus*. Pokud je R zřejmé z kontextu, značíme R -homomorfismus jednoduše homomorfismus.

Věta 3.25 (Jádro a obraz R -homomorfismu)

Bud' τ libovolný R -homomorfismus R -modulů M a N . Definujme jádro homomorfismu:

$$\ker(\tau) = \{v \in M \mid \tau(v) = 0\},$$

a obraz homomorfismu:

$$\text{im}(\tau) = \{\tau(v) \in N \mid v \in M\}.$$

Pak platí, že jádro $\ker(\tau)$ je podmodul M a obraz $\text{im}(\tau)$ je podmodul N .

Definice 3.26 (Faktorový modul)

Buď S podmodul R -modulu M . Následující binární relace

$$u \equiv v \Leftrightarrow u - v \in S$$

je relací ekvivalence na M . Její třídy ekvivalence jsou následující množiny

$$v + S = \{v + s \mid s \in S\}.$$

Množina všech tříd této ekvivalence tvoří R -modul a nazývá se *faktorový modul* M podle S . Značí se M/S . Operace na M/S jsou definovány následovně:

$$(u + S) + (v + S) = (u + v) + S,$$

$$r(u + S) = ru + S.$$

Nulou faktorového modulu M/S je třída $0 + S = S$.

Obecně podmoduly volných podmodulů nemusí být volné. Následující věta říká, které podmoduly volných modulů volné jsou:

Věta 3.27 (Hodnost modulů nad oborem hlavních ideálů)

Buď M volný modul nad oborem hlavních ideálů R . Pak libovolný podmodul $S \subseteq M$ je také volný a platí $\text{rank}(S) \leq \text{rank}(M)$.

Věta 3.28 (O struktuře konečně generovaných modulů nad oborem hlavních ideálů)

Libovolný konečně generovaný modul M nad oborem hlavních ideálů R je izomorfní přímému součtu cyklických podmodulů modulu M . Existuje tedy následující izomorfismus:

$$M \cong R^\beta \oplus \left(\bigoplus_{i=1}^m R/\langle r_i \rangle \right),$$

kde $r_i \in R$ a $r_i \mid r_{i+1}$ pro nějaká $\beta, m \in \mathbb{N}$. R je daný okruh a $\langle r_i \rangle$ jeho ideály. Číslo β udává počet takzvaných volných komponent. Komponenty ve tvaru $R/\langle r_i \rangle$ se nazývají torze.

3.3 Gradované okruhy a gradované moduly

V této kapitole se dostáváme ke gradovaným modulům a k větě o struktuře gradovaných modulů. Informace jsou čerpány ze zdroje [13].

Definujme si nejprve pojem gradovaný okruh:

Definice 3.29 (Gradovaný okruh)

Mějme okruh R a systém abelovských grup $\mathcal{R} = \{R_i\}_{i \in \mathbb{Z}}$, pro který platí:

$$R = \bigoplus_i R_i, \text{ kde } i \in \mathbb{Z}.$$

Okruh R spolu se systémem \mathcal{R} se nazývá *gradovaný okruh*, pokud pro všechna m, n platí:

$$\text{pokud } r \in R_n, s \in R_m, \text{ pak } rs \in R_{n+m}.$$

Systém \mathcal{R} se nazývá *gradování okruhu* R . Pokud prvek $r \in R_i$, pak říkáme, že r má stupeň i v gradování \mathcal{R} .

Definice 3.30 (Gradovaný modul)

Nechť R je gradovaný okruh s gradováním $\mathcal{R} = \{R_i\}_{i \in \mathbb{Z}}$, nechť M je modul nad R a dále mějme systém abelovských grup $\mathcal{M} = \{M_i\}_{i \in \mathbb{Z}}$, pro který platí:

$$M = \bigoplus_i M_i, \text{ kde } i \in \mathbb{Z}.$$

Modul M společně se systémem \mathcal{M} se nazývá *gradovaný modul*, pokud pro všechna m, n platí:

$$\text{pokud } r \in R_n, a \in M_m, \text{ pak } ra \in R_{m+n}.$$

Systém \mathcal{M} se nazývá *gradování modulu* M . Pokud prvek $m \in M_i$, pak říkáme, že m má stupeň i v gradování \mathcal{M} .

Definice 3.31 (Nezáporně gradované okruhy a nezáporně gradované moduly)

Gradovaný okruh se nazývá *nezáporně gradovaný*, pokud pro všechna $i < 0$ v jeho gradování platí $R_i \cong 0$. Analogicky je definováno nezáporné gradování pro moduly.

PŘÍKLAD 3.32

Nechť $R[x]$ je okruh polynomů s proměnnou x . Na $R[x]$ můžeme zavést gradování takto: pro $i < 0$ položíme $R_i \cong 0$, pro $i = 0$ položíme $R_0 = R$ a pro $i > 0$ nechť R_i obsahuje všechny homogenní polynomy stupně i .

Toto gradování se nazývá *standardní* a v této práci budeme vždy uvažovat na $R[x]$ standardní gradování.

Lemma 3.33 (Faktorový gradovaný modul)

Nechť M je gradovaný modul nad gradovaným okruhem R a nechť N je jeho podmodul. Pro faktorový modul M/N platí:

$$M/N = \bigoplus_{i \in \mathbb{Z}} M_i/N_i,$$

kde M_i a N_i jsou jednotlivé abelovské grupy v gradování příslušných modulů a M_i/N_i je faktorový modul dle definice 3.26.

Definice 3.34 (Homomorfismus gradovaných modulů)

Nechť M a N jsou gradované moduly nad gradovaným okruhem R . *Gradovaný homomorfismus* $f : M \rightarrow N$ je takový homomorfismus modulů, pro který platí $f(M_i) \subseteq N_i$ pro všechna $i \in \mathbb{Z}$.

Homomorfismus gradovaných modulů tedy zachovává stupeň prvku modulu v jeho gradování. Pokud tedy například prvek a patří v gradování modulu M do grupy M_i , pak prvek $f(a)$ v gradování modulu N patří do grupy N_i .

Věta 3.35 (O struktuře konečně generovaných nezáporně gradovaných modulů nad okruhem polynomů)

Nechť $F[x]$ je gradovaný okruh polynomů nad polem F . Nechť M je konečně generovaný nezáporně gradovaný modul nad $F[x]$. Pak existují celá čísla $i_1, \dots, i_m, j_1, \dots, j_n$ a l_1, \dots, l_n a následující izomorfismus:

$$M \cong \left(\bigoplus_{s=1}^m \langle x^{i_s} \rangle \right) \oplus \left(\bigoplus_{t=1}^n \langle x^{j_t} \rangle / \langle x^{l_t} \rangle \right),$$

kde jednotlivé $\langle x^d \rangle$ jsou ideály generované homogenním polynomem x^d .

Každý ideál $\langle x^d \rangle$ z předchozí věty můžeme uvažovat jako grupu. Každý gradovaný modul M , který splňuje podmínky uvedené v této větě, je tedy izomorfní přímému součtu grup

$$\left(\bigoplus_{s=1}^m \langle x^{i_s} \rangle \right) \oplus \left(\bigoplus_{t=1}^n \langle x^{j_t} \rangle / \langle x^{l_t} \rangle \right),$$

kde tento přímý součet spolu s operací sčítání a násobení, které jsou definované po složkách, tvoří gradovaný modul. [15]

Komponenty tohoto přímého součtu ve tvaru $\langle x^{i_s} \rangle$ obsahují polynomy stupně i_s a vyšší. Komponenty ve tvaru $\langle x^{j_t} \rangle / \langle x^{l_t} \rangle$ pak obsahují polynomy od stupně j_t do stupně $l_t - 1$ (včetně).

4 Algebraická topologie

Jedním ze základních problémů topologie je určit, zda dva dané topologické prostory jsou homeomorfní, nebo ne. Bohužel pro vyřešení tohoto problému neexistují žádné obecné postupy, především pak i z toho důvodu, že tento problém není ani rozhodnutelný.

Proto algebraická topologie přichází s jiným řešením. Na základě určitých vlastností topologických prostorů jim přiřazuje různé algebraické struktury. Tyto struktury pak slouží jako *invariant* topologického prostoru.

V našem kontextu algebraické topologie chápeme invariant jako postup, který homeomorfním prostorům přiřadí izomorfní algebraické struktury stejného typu. Například, pokud se rozhodneme přiřadit každému topologickému prostoru nějakou grupu, pak musí být splněna následující formule:

$$\text{pokud } X \cong Y, \text{ pak } G(X) \cong G(Y), \quad (4.1)$$

kde X, Y jsou topologické prostory a $G(X), G(Y)$ jsou jim přiřazené grupy.

Užitečná pak pro nás může být i kontrapozice výše uvedené formule:

$$\text{pokud } G(X) \not\cong G(Y), \text{ pak } X \not\cong Y.$$

Toto tvrzení nám říká, jak dokázat, že dva topologické prostory nejsou homeomorfní: musíme určit jejich invariant a ukázat, že se tyto invarianty nerovnej.

Algebraická topologie se dělí na dvě hlavní teorie: *homotopii* a *homologii*. Každá z teorií zaujímá zcela jiný přístup, avšak princip je v obou teoriích velmi podobný: v obou se topologickým prostorům přiřazují algebraické struktury.

V následujících podkapitolách jsou obě teorie popsány. Homotopie se však v topologické analýze dat příliš nepoužívá. Proto kapitola věnovaná homotopii je spíše informativní a poslouží nám jako výchozí bod pro algebraickou topologii. Nicméně některé pojmy z homotopie využijeme i v jiných kapitolách, zejména pak v kapitole 8.

Zbývající kapitoly se pak již věnují pouze homologii a to konkrétně jedné její verzi, zvané *simpliciální homologie*.

Simpliciální homologie a jí příbuzná témata jsou často v literatuře označovány jako *výpočetní topologie*. Tím autoři zdůrazňují fakt, že se simpliciální homologie vskutku využívá v praxi při počítačovém zpracování.

Simpliciální homologie se také různým způsobem využívá v analýze a zpracování dat. Vybrané možnosti využití simpliciální homologie si ukážeme posléze na příkladech.

4.1 Homotopie

Homotopie, podle níž je celá teorie pojmenována, je relací ekvivalence na spojitých zobrazeních. Klíčovým prvkem je pak spojitě zobrazení zvané *oblouk* a jeho speciální případ, zvaný *smyčka*. Na třídách ekvivalence smyček podle homotopie je pak definována již zmiňovaná grupa.

V této kapitole si postupně tyto pojmy vysvětlíme a na konci si ukážeme, jak vypadá taková grupa pro topologický prostor \mathbb{S}^1 .

Informace v této kapitole jsou čerpány ze zdrojů [6] a [7].

Definujme si pojem homotopie:

Definice 4.1 (Homotopie)

Spojitá zobrazení $f, g : X \rightarrow Y$ se nazývají *homotopická*, jestliže existuje spojitě zobrazení

$$F : X \times [0, 1] \rightarrow Y,$$

takové, které splňuje

$$F(x, 0) = f(x) \quad \text{a} \quad F(x, 1) = g(x).$$

Zobrazení F se nazývá *homotopie* mezi f a g .

Dva topologické prostory X a Y nazýváme *homotopicky ekvivalentní*, jestliže existují zobrazení $f : X \rightarrow Y$ a $g : Y \rightarrow X$ taková, že $g \circ f$ je homotopické s identitou na X a $f \circ g$ je homotopické s identitou na Y . Pokud je topologický prostor homotopicky ekvivalentní s bodem, nazývá se *kontraktibilní*.

PŘÍKLAD 4.2

Vezměme si zobrazení $f(x) = 2 + \sin(x)$ a zobrazení $g(x) = 4$. Definujme homotopii mezi f a g : $F(x, t) = (1 - t) \cdot f(x) + t \cdot g(x)$. Parametr t můžeme chápat jako čas. Na F se pak můžeme dívat jako na spojitou deformaci křivky vykreslenou zobrazením f na křivku vykreslenou zobrazením g : křivka vykreslená f se v čase postupně vyhladí. Příklad je znázorněn na obrázku 4.1.

Nyní si definujme pojem oblouk:

Definice 4.3 (Oblouk)

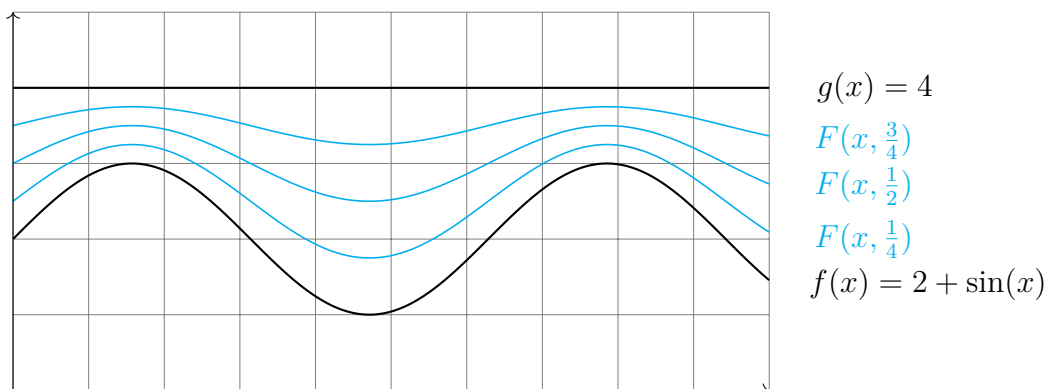
Oblouk mezi body x_0 a x_1 v topologickém prostoru X je libovolné spojitě zobrazení

$$\gamma : [0, 1] \rightarrow X,$$

které splňuje

$$\gamma(0) = x_0 \quad \text{a} \quad \gamma(1) = x_1.$$

Bod x_0 se nazývá *počáteční* a bod x_1 se nazývá *koncový*.



Obrázek 4.1: Příklad homotopie funkcí. Funkce f je homotopická s funkcí g .

Topologický prostor X se nazývá *obloukově souvislý*, pokud pro každé dva body existuje oblouk. Množina v topologickém prostoru je *obloukově souvislá*, je-li obloukově souvislá jako topologický podprostor s indukovanou topologií.

Oblouk chápeme jako spojitě zobrazení, které nás dostane z bodu x_0 do bodu x_1 . V anglické literatuře se pro oblouk používá termín „*path*“, což je trochu výstižnější. Oblouková souvislost topologického prostoru je užitečná vlastnost. Bude se nám hodit později.

Podobně jako na obecných spojitých zobrazeních, i na obloucích lze zavést homotopii.

Definice 4.4 (Oblouková homotopie)

Oblouky $f, g : [0, 1] \rightarrow X$ se nazývají *obloukově homotopické*, popřípadě *obloukově homotopicky ekvivalentní*, jestliže mají stejný počáteční bod x_0 i koncový bod x_1 a existuje spojitě zobrazení

$$F : [0, 1] \times [0, 1] \rightarrow X,$$

které splňuje

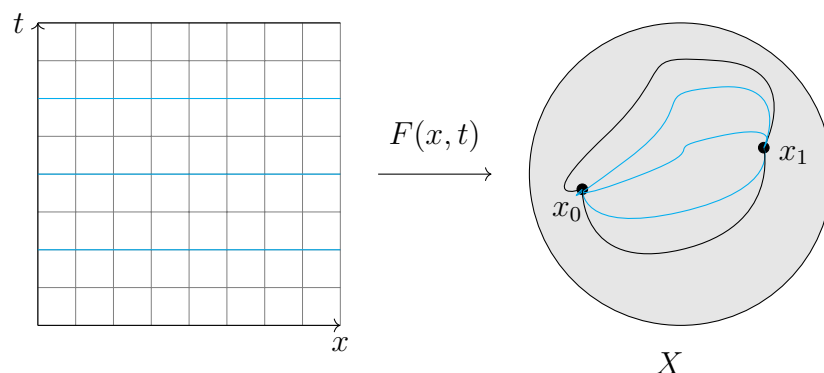
$$F(x, 0) = f(x) \text{ a } F(x, 1) = g(x),$$

$$F(0, t) = x_0 \text{ a } F(1, t) = x_1,$$

pro každé $x \in [0, 1]$ a $t \in [0, 1]$. Zobrazení F se nazývá *oblouková homotopie* mezi f a g .

První podmínka říká, že F musí být homotopií mezi f a g . Druhá podmínka říká, že pro všechna t bude $F(x, t)$ obloukem z bodu x_0 do bodu x_1 . Jinými slovy, pokud se na F díváme jako na spojitou deformaci oblouku f na oblouk g , nesmí se během deformace změnit počáteční ani koncový bod oblouku. Třídu ekvivalence podle homotopie generovanou obloukem f značíme $[f]$. Po zbytek této kapitoly, pokud budeme hovořit o homotopii mezi oblouky, budeme tím mít vždy na mysli homotopii obloukovou.

Příklad homotopicky ekvivalentních oblouků je na obrázku 4.2.



Obrázek 4.2: Příklad homotopie oblouků.

Zavedme si na množině oblouků následující operaci:

Definice 4.5 (Násobení oblouků)

Pokud f je oblouk na X z bodu x_0 do bodu x_1 a g je oblouk na X z bodu x_1 do bodu x_2 , definujeme *násobení oblouků* $f * g$ předpisem

$$h(t) = \begin{cases} f(2t) & t \in [0, \frac{1}{2}], \\ g(2t - 1) & t \in [\frac{1}{2}, 1]. \end{cases}$$

Výsledkem je tedy spojitě zobrazení h , které je obloukem z bodu x_0 do bodu x_2 . Oblouk h můžeme chápat tak, že jeho první polovina je tvořena obloukem f a druhá polovina obloukem g . Tato operace také indukuje operaci násobení na třídách homotopicky ekvivalentních oblouků. Nechť f a g jsou oblouky. Jimi tvořené třídy ekvivalence podle homotopie označme $[f]$ a $[g]$. Operace násobení na třídách homotopicky ekvivalentních oblouků je definována následovně:

$$[f] * [g] = [f * g].$$

Speciálním případem oblouku je takzvaná *smyčka*. Je to jednoduše oblouk, jehož počáteční i koncový bod je stejný. Hovoříme tak o smyčce v bodě x_0 . Pomocí smyčky je pak definována fundamentální grupa:

Definice 4.6 (Fundamentální grupa)

Mějme topologický prostor X a bod $x_0 \in X$. Množina všech tříd ekvivalence smyček v bodě x_0 podle homotopie spolu s operací $*$ tvoří grupu. Této grupě říkáme *fundamentální grupa* a značíme ji $\pi_1(X, x_0)$. Neutrálním prvkem této grupy je třída ekvivalence smyček homotopicky ekvivalentních s konstantní funkcí

$$f(t) = x_0 \text{ pro všechna } t \in [0, 1].$$

Myšlenkou homotopie je tedy každému topologickému prostoru přiřadit grupu $\pi_1(X, x_0)$ na základě všech možných existujících smyček v daném prostoru. Tato grupa je nezávislá na volbě počátečního bodu x_0 . Jinými slovy, pokud zvolíme dva různé body x_0 a y_0 , grupy $\pi_1(X, x_0)$ a $\pi_1(X, y_0)$ budou izomorfní. Takto tedy lze klasifikovat topologické prostory.

Věta 4.7 (Fundamentální grupa prostoru \mathbb{S}^1)

Fundamentální grupa $\pi_1(\mathbb{S}^1, x_0)$ je nekonečná cyklická grupa generovaná třídou homotopicky ekvivalentních smyček

$$\gamma(x) = (\cos(2\pi x), \sin(2\pi x))$$

v bodě $x_0 = (1, 0)$.

Důkaz této věty zde také uvádět nebudeme. Nicméně, pro pochopení, o co se jedná, stačí teorie uvedená v této kapitole. Libovolná smyčka na \mathbb{S}^1 může být buďto konstantní, čili homotopicky ekvivalentní s konstantním zobrazením $f(x) = x_0$, anebo musí *oběhnout* celou sféru \mathbb{S}^1 . Nicméně, všechny smyčky, které oběhnou celou sféru \mathbb{S}^1 , jsou homotopicky ekvivalentní smyčce $\gamma(x) = (\cos(2\pi x), \sin(2\pi x))$.

PŘÍKLAD 4.8

Pro libovolný prostor \mathbb{R}^n je jeho fundamentální grupa $\pi_1(\mathbb{R}^n, x_0)$ triviální. Libovolná smyčka na \mathbb{R}^n v bodě x_0 je totiž homotopicky ekvivalentní s konstantou x_0 . Obecně platí, že pro jakýkoliv prostor, který je konvexním obalem² nějaké podmnožiny \mathbb{R}^n , je jeho fundamentální grupa $\pi_1(\mathbb{R}^n, x_0)$ triviální.

Grupě $\pi_1(X, x_0)$ se někdy říká první homotopická grupa prostoru X . To naznačuje, že existují i další homotopické grupy. Konkrétně, existují grupy $\pi_n(X, x_0)$ pro všechna $n \in \mathbb{Z}_+$. Tyto grupy jsou zobecněním fundamentální grupy pro smyčky vyšších dimenzí, které jsou definované jako zobrazení $[0, 1]^n \rightarrow X$.

Tímto přístupem se však v této práci zabývat nebudeme. Definice fundamentální grupy je závislá na smyčkách, což jsou spojitá zobrazení v daném topologickém prostoru. Tato zobrazení je obtížné spočítat. Většinou se určují analyticky, a navíc, problém přiřazení určité fundamentální grupy topologickému prostoru není rozhodnutelný.

²Připomenutí pojmu *konvexní obal* je na straně 32.

4.2 Simplicialní komplexy

V předchozí kapitole jsme si ukázali, že homotopie je závislá na spojitých zobrazeních v daném topologickém prostoru. Na jejich základě je pak topologickému prostoru přiřazena patřičná grupa.

Některé topologické prostory lze reprezentovat pomocí objektů, které se nazývají *simplicialní komplexy* a tyto simplicialní komplexy jsou pak dále sestaveny z jednoduchých dílčích objektů, zvaných *simplexy*.

Nejprve si tedy popíšeme zmíněné simplicialní komplexy a v následující kapitole se pak budeme věnovat simplicialní homologii, ve které se ke klasifikaci topologických prostorů využívají právě zmíněné simplicialní komplexy.

Simplicialní komplexy jsou dvojího typu: *geometrické* a *abstraktní*. Začneme s geometrickým simplicialním komplexem a postupně přejdeme k abstraktnímu. Ukážeme si také, že oba dva typy komplexů jsou ve skutečnosti ekvivalentní. To znamená, že z geometrického simplicialního komplexu lze jistým způsobem vytvořit abstraktní simplicialní komplex a naopak.

Základní stavební jednotkou je tedy *geometrický simplex*. Než si jej definujeme, připomeňme si několik pojmů z oblasti afinní geometrie:

Afinní kombinace, afinní obal Mějme body $v_0, v_1, \dots, v_k \in \mathbb{R}^d$. Bod $x = \sum_{i=0}^k \lambda_i v_i$, kde $\lambda_i \in \mathbb{R}$, se nazývá *afinní kombinací* bodů v_0, v_1, \dots, v_k , pokud platí $\sum_{i=0}^k \lambda_i = 1$. *Afinní obal* množiny bodů je množina všech afinních kombinací bodů dané množiny.

Body v obecné poloze Řekneme, že body $v_0, v_1, \dots, v_k \in \mathbb{R}^d$ jsou v *obecné poloze*, pokud žádný z nich není afinní kombinací ostatních.

Konvexní kombinace, konvexní obal Afinní kombinace $x = \sum_{i=0}^k \lambda_i v_i$ se nazývá *konvexní kombinace*, pokud $\lambda_i \geq 0$ pro všechna $i \in \{0, \dots, k\}$. *Konvexní obal* je množina všech konvexních kombinací. Konvexní obal neprázdné množiny bodů $S = \{v_0, v_1, \dots, v_k\}$ značíme $\text{conv}(S)$.

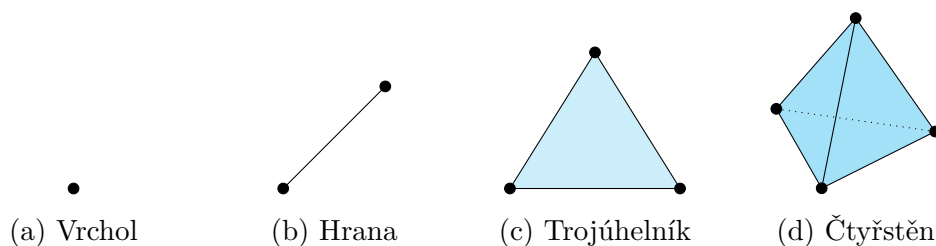
Definice 4.9 (Geometrický simplex)

Mějme neprázdnou množinu bodů $S = \{v_0, v_1, \dots, v_k\} \subseteq \mathbb{R}^d$ v obecné poloze. Konvexní obal množiny S se nazývá *geometrický k -simplex*.

Bodům v S říkáme *vrcholy simplexu*. Dimenze k -simplexu je vždy rovna číslu k a značíme ji $\dim(\sigma)$.

Definice 4.10 (Stěna)

Nechť σ je simplicialní k -simplex s množinou vrcholů $S = \{v_0, v_1, \dots, v_k\}$. Simplex $\tau \subseteq \sigma$ s množinou vrcholů $V \subseteq S$ se nazývá *stěna*. Relaci *býti stěnou* značíme $\tau \leq \sigma$.



Obrázek 4.3: Příklady simplexů s jejich názvy.

Příklady geometrických simplexů jsou na obrázku 4.3. U každého simplexu je uveden také jeho název. Tyto názvy se často využívají.

Z jednotlivých geometrických simplexů se pak sestavují geometrické simplicialní komplexy:

Definice 4.11 (Geometrický simplicialní komplex)

Geometrický simplicialní komplex K je konečná množina simplexů, pro kterou platí:

1. Pokud $\sigma \in K$ a $v \leq \sigma$, pak $v \in K$.
2. Pokud $\sigma_1, \sigma_2 \in K$, pak $(\sigma_1 \cap \sigma_2) \leq \sigma_1$ a $(\sigma_1 \cap \sigma_2) \leq \sigma_2$.

První podmínka jednoduše říká, že simplicialní komplex musí obsahovat stěnu každého simplexu jakožto samostatný simplex. Druhá podmínka pak říká, že pokud mají libovolné dva simplexu neprázdný průnik, pak tímto průnikem může být jedině další simplex. Jinými slovy, dva simplexu se sebou mohou sdílet pouze některou ze svých stěn a tato stěna se musí nacházet v simplicialním komplexu jakožto samostatný simplex.

Jako dimenzi simplicialního komplexu K bereme nejvyšší dimenzi simplexu obsaženého v K , tedy:

$$\dim(K) = \max \{ \dim(\sigma) \mid \sigma \in K \}.$$

Jako vrcholy simplicialního komplexu K uvažujeme všechny 0-simplexy v K . Množinu vrcholů značíme $\text{vert}(K)$.

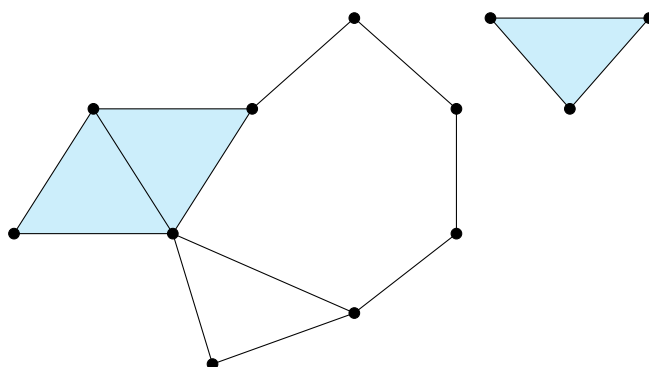
PŘÍKLAD 4.12

Na obrázku 4.4 je příklad geometrického simplicialního komplexu a na obrázku 4.5 je příklad množiny, která geometrický simplicialní komplex netvoří.

Nyní se podíváme na to, jak lze pomocí simplicialních komplexů diskrétním způsobem reprezentovat topologické prostory.

Definice 4.13 (Nosná množina geometrického simplicialního komplexu)

Nosná množina simplicialního komplexu K je množina $|K| = \bigcup_{\sigma \in K} \sigma$.



Obrázek 4.4: Příklad geometrického simplicialního komplexu. Tento simplicialní komplex obsahuje tři 2-simplexy, nebo-li trojúhelníky a několik 1-simplexů, nebo-li hran. Všimněme si, že dva trojúhelníky nalevo společně sdílí jednu stejnou hranu a dva stejné vrcholy, zatímco trojúhelník vpravo nesdílí žádnou ze svých stěn s žádným dalším simplexem.



Obrázek 4.5: Příklad množiny simplexů, která však netvoří simplicialní komplex. Jednomu z trojúhelníků vlevo chybí jedna stěna. U trojúhelníků vpravo jsou sice v množině obsaženy všechny jejich stěny, ale jejich průnikem není společná stěna, nýbrž pouze část společné hrany.

Nosná množina geometrického simplicialního komplexu K tedy tvoří topologický prostor, kde $|K|$ uvažujeme s indukovanou topologií z \mathbb{R}^d .

Definice 4.14 (Triangulace)

Triangulace topologického prostoru X je geometrický simplicialní komplex K spolu s homeomorfismem $f : |K| \rightarrow X$.

Definice tedy říká, že daný topologický prostor X musí být homeomorfní s topologickým prostorem $|K|$.

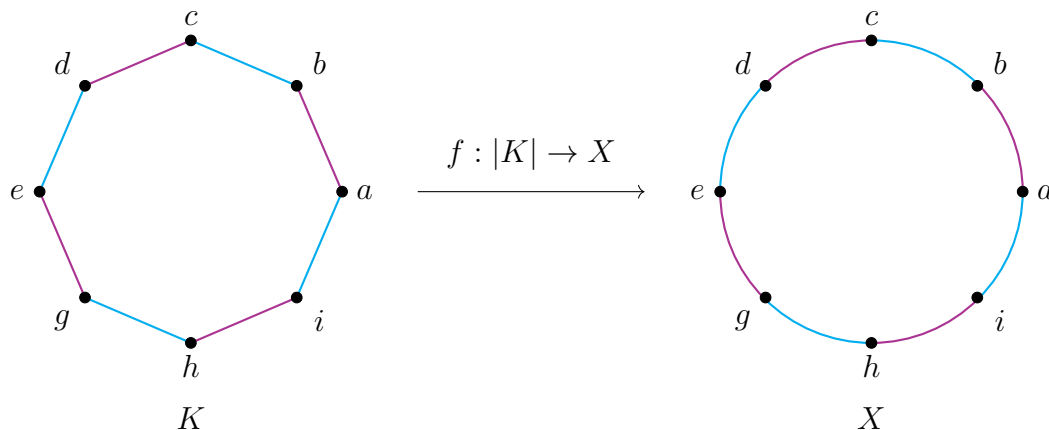
Triangulace topologických prostorů má svá omezení. Existují topologické prostory, které nelze triangulovat vůbec.

Například už jen z podmínky, že simplicialní komplex může obsahovat pouze konečný počet simplexů, plyne jedno zásadní omezení pro triangulaci topologických prostorů.

Tato omezení v této práci nebudeme řešit, protože zde budeme většinou pracovat přímo se simplicialními komplexy.

PŘÍKLAD 4.15

Na obrázku 4.6 je načrt jedné možné triangulace jednotkové kružnice.



Obrázek 4.6: Simpliciací komplex K spolu s homeomorfismem $f : |K| \rightarrow X$ tvoří triangulaci jednotkové kružnice. Na jednotkové kružnici jsou zvýrazněny obrazy homeomorfismu f odpovídající jednotlivým simplexům ze simpliciací komplexu K .

Nyní přejdeme k abstraktní verzi simpliciací komplexů:

Definice 4.16 (Abstraktní simplex)

Libovolná množina S s počtem prvků $k + 1$ se nazývá *abstraktní k -simplex*.

Podobně, jako u geometrických simplexů, jednotlivým prvkům v S říkáme vrcholy. *Stěna* simplexu je definovaná jako jeho libovolná neprázdná podmnožina.

V zásadě stejným způsobem je pak z těchto abstraktních simplexů sestaven abstraktní simpliciací komplex:

Definice 4.17 (Abstraktní simpliciací komplex)

Abstraktní simpliciací komplex je dvojice $K = (V, \mathcal{S})$, kde V je libovolná konečná množina a \mathcal{S} je systém podmnožin V splňující:

1. Pro všechny prvky $v \in V$ platí $\{v\} \in \mathcal{S}$.
2. Pokud $\tau \subseteq \sigma \in \mathcal{S}$ a $\tau \neq \emptyset$, pak $\tau \in \mathcal{S}$.

Prvkům $v \in V$ říkáme vrcholy. Množinu $\sigma \in \mathcal{S}$ nazveme k -simplex, pokud platí $|\sigma| = k + 1$.

Systém množin \mathcal{S} tedy udává, jaké abstraktní simplexu z vrcholů V jsou v simpliciací komplexu K obsaženy. Pokud je systém množin \mathcal{S} zřejmý z kontextu, označujeme daný abstraktní simpliciací komplex pouze K .

Řekneme, že dva abstraktní simpliciací komplexu $K_1 = (V_1, \mathcal{S}_1)$ a $K_2 = (V_2, \mathcal{S}_2)$ jsou *izomorfní*, pokud existuje bijektivní zobrazení $\Phi : V_1 \rightarrow V_2$ takové,

kteřé splňuje, že množina σ tvoří k -simplex v K_1 právě tehdy, když $\Phi(\sigma)$ tvoří k -simplex v K_2 .

POZNÁMKA 4.18

Prozkoumejme krátce paměťovou náročnost abstraktních simplicialních komplexů. Z definice stěny přímo plyne, že daný k -simplex obsahuje celkem $\binom{k+1}{l+1}$ různých stěn dimenze l . Celkový počet stěn k -simplexu pak odpovídá počtu všech neprázdných podmnožin tohoto simplexu:

$$\sum_{l=-1}^k \binom{k+1}{l+1} = 2^{k+1} - 1.$$

Z definice simplicialního komplexu pak plyne, že daný simplicialní komplex musí obsahovat všechny stěny všech svých simplexů jakožto samostatné simplexy.

Simplicialní komplex je tedy celkem rozměrný, a tudíž i paměťově náročný objekt. Tento problém se v praxi řeší tak, že v paměti explicitně uložíme pouze *maximální* simplexy. Tedy takové simplexy, které již nejsou stěnou žádného dalšího simplexu kromě sebe samotného. Z takto uloženého simplicialního komplexu pak lze v případě potřeby získat všechny jeho simplexy.

PŘÍKLAD 4.19

Uvažujme systém množin

$$\mathcal{S} = \{ \{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{g\}, \{h\}, \{i\}, \\ \{a, b\}, \{b, c\}, \{c, d\}, \{d, e\}, \{e, g\}, \{g, h\}, \{h, i\}, \{i, a\} \}.$$

Tento systém tvoří abstraktní simplicialní komplex spolu s množinou vrcholů

$$V = \{a, b, c, d, e, g, i\}.$$

Později v této práci využijeme i následující pojem:

Definice 4.20 (Podkomplex)

Řekneme, že abstraktní simplicialní komplex $L = (M, \mathcal{U})$ je *podkomplex* abstraktního simplicialního komplexu $K = (V, \mathcal{S})$, pokud platí $M \subseteq V$ a $\mathcal{U} \subseteq \mathcal{S}$. Fakt, že simplicialní komplex L je podkomplexem K , značíme $L \leq K$.

Následující pojem je můstkem mezi abstraktním simplicialním komplexem a geometrickým simplicialním komplexem:

Definice 4.21 (Schéma vrcholů)

Buď K geometrický simplicialní komplex s množinou vrcholů $\text{vert}(K)$. Necht \mathcal{K} je systém množin, který obsahuje právě všechny takové množiny

$$\{v_0, v_1, \dots, v_k\} \subseteq \text{vert}(K),$$

kteřé jsou vrcholy nějakého geometrického k -simplexu v K . Systém množin \mathcal{K} se nazývá *schéma vrcholů* geometrického simplicialního komplexu K .

Všimněme si, že systém množin \mathcal{K} splňuje podmínky v definici abstraktního simplicialního komplexu. Dvojice $(\text{vert}(K), \mathcal{K})$ tedy tvoří abstraktní simplicialní komplex.

Schéma vrcholů nám tedy dává návod, jak z libovolného geometrického simplicialního komplexu vytvořit jeho abstraktní verzi.

Definice 4.22 (Geometrická realizace)

Geometrickou realizací abstraktního simplicialního komplexu K nazveme takový geometrický simplicialní komplex, jehož vrcholové schéma \mathcal{K} je izomorfní s K .

Následující věta říká, že geometrickou realizaci lze provést vždy.

Věta 4.23 (O geometrické realizaci abstraktních simplicialních komplexů)

Každý abstraktní simplicialní komplex K dimenze $\dim(K) = d$ má geometrickou realizaci v prostoru \mathbb{R}^{2d+1} .

Důkaz

Mějme zobrazení $f : \text{vert}(K) \rightarrow \mathbb{R}^{2d+1}$, které je injektivní a jeho obrazy jsou body v obecné poloze. Nechtě σ_1 a σ_2 jsou abstraktní simplex v K . Připomeňme, že abstraktní simplicialní komplexy jsou jen množiny prvků. Jejich sjednocením vznikne množina velikosti:

$$\text{card}(\sigma_1 \cup \sigma_2) = \text{card}(\sigma_1) + \text{card}(\sigma_2) - \text{card}(\sigma_1 \cap \sigma_2) \leq 2d + 2.$$

Tudíž body $f(\sigma_1 \cup \sigma_2)$ jsou afinně nezávislé, jelikož z vlastností f plyne, že libovolná množina bodů velikosti menší nebo rovno $2d + 2$ je afinně nezávislá. Z toho plyne, že jakákoliv konvexní kombinace bodu $x \in f(\sigma_1 \cup \sigma_2)$ je unikátní kombinací bodů $f(\sigma_1) \cup f(\sigma_2)$.

Dostáváme, že $x \in \text{conv}f(\sigma_1)$ a $x \in \text{conv}f(\sigma_2)$ právě tehdy, když x je konvexní kombinací bodů z $f(\sigma_1) \cap f(\sigma_2)$. Z toho plyne, že průnik $\text{conv}f(\sigma_1)$ a $\text{conv}f(\sigma_2)$ je buďto prázdný, nebo tvoří geometrický simplex $\text{conv}f(\sigma_1 \cap \sigma_2)$. \square

Ke každému geometrickému simplicialnímu komplexu tedy lze vytvořit abstraktní simplicialní komplex, který je izomorfní s jeho schématem vrcholů. Stejně tak naopak, ke každému abstraktnímu simplicialnímu komplexu můžeme vytvořit jeho geometrickou realizaci. Geometrické simplicialní komplexy a abstraktní simplicialní komplexy jsou tedy v tomto smyslu ekvivalentní.

PŘÍKLAD 4.24

Geometrický simplicialní komplex na obrázku 4.6 je geometrickou realizací abstraktního simplicialního komplexu z příkladu 4.19.

V metodách topologické analýzy dat, které jsou založeny na simplicialní homologii, se většinou pracuje s abstraktními simplicialními komplexy. Výhodou abstraktní verze totiž je, že ji lze jednodušeji reprezentovat v paměti počítače. Navíc většinou ani není nutné zabývat se jejich geometrickou realizací.

V této práci také budeme většinou pracovat s abstraktními simplicialními komplexy. Pokud to však dimenze simplicialního komplexu dovolí, geometrickou realizaci využijeme k jejich znázornění. Znázornění obrázkem je totiž dle mého názoru výstižnější než pouhý výpis systému množin.

Pro potřeby simplicialní homologie je ještě nutné vrcholy simplicialního komplexu uspořádat. Na vrcholech simplicialního komplexu K tedy zvolíme libovolné uspořádání vrcholů $v_1 < v_2 < \dots < v_{n-1} < v_n$, kde n značí počet vrcholů simplicialního komplexu K . Simplex s uspořádanými vrcholy značíme následovně:

$$\sigma = [v_{i_0}, v_{i_1}, \dots, v_{i_k}] \text{ kde } v_{i_0} < v_{i_1} < \dots < v_{i_k}.$$

Zjednodušeně pak můžeme zapsat daný simplex takto:

$$\sigma = v_{i_0} v_{i_1} \dots v_{i_k}.$$

Budeme tedy předpokládat, že na simplicialním komplexu je vždy zavedeno nějaké uspořádání. Většinou je pak toto uspořádání zřejmé z kontextu. Například, pro značení vrcholů se převážně využívají písmena anglické abecedy. Je tedy přirozené uvažovat lexikografické uspořádání.

4.3 Simpliciací homologie

V této kapitole pomocí simpliciací komplexů sestavíme jisté moduly. Obsahově je pak kapitola rozdělena na dvě části. Nejprve si definujeme potřebné pojmy a poté si ukážeme, jaké topologické vlastnosti tyto moduly popisují.

Definice 4.25 (Modul k -řetězců)

Mějme simpliciací komplex K , necht' pro zvolené $k \in \mathbb{Z}_{\geq 0}$ číslo m_k značí počet k -simplexů v K a necht' R je libovolný obor hlavních ideálů. Necht' M je množina, jejíž prvky jsou všechny formální lineární kombinace ve tvaru

$$\sum_{i=1}^{m_k} a_i \sigma_i, \text{ kde } a_i \in R, \sigma_i \in K \text{ a } \dim(\sigma) = k.$$

Tyto prvky se nazývají k -řetězce. Na k -řetězcích uvažujeme standardní operace sčítání a násobení skalárem. Sčítání řetězců je definováno takto:

$$\left(\sum_{i=1}^{m_k} a_i \sigma_i \right) + \left(\sum_{i=1}^{m_k} b_i \sigma_i \right) = \sum_{i=1}^{m_k} (a_i + b_i) \sigma_i.$$

A násobení řetězců skalárem $r \in R$ takto:

$$r \cdot \left(\sum_{i=1}^{m_k} a_i \sigma_i \right) = \sum_{i=1}^{m_k} (r \cdot a_i) \sigma_i.$$

Pro $k \geq 0$, *modulem k -řetězců* rozumíme modul M nad R s bází tvořenou všemi k -simplexy v K , spolu s operacemi sčítání řetězců a násobení řetězců skalárem. Nulou modulu M je řetězec, jehož všechny koeficienty a_i , kde $i \in \{1, \dots, m_k\}$ jsou nulové. Modul k -řetězců simpliciací komplexu K s koeficienty v R značíme $C_k(K, R)$. Pokud jsou K a R zřejmé z kontextu, značí se pouze C_k .

POZNÁMKA 4.26

Platí $C_k(K, R) \cong \bigoplus_{i=1}^{m_k} R$, kde m_k značí počet k -simplexů v K . Tento fakt plyne jednoduše z toho, že modul $C_k(K, R)$ je volný a je generován bází obsahující právě m_k prvků.

POZNÁMKA 4.27

Pokud za okruh R zvolíme \mathbb{Z} , pak daný modul k -řetězců tvoří takzvanou *volnou abelovskou grupu*. Někteří autoři definují algebraickou strukturu k -řetězců pomocí volné abelovské grupy, namísto R -modulu. Nevýhoda při použití grupy je spíše kosmetická: u grup totiž nelze tak elegantně mluvit o jejich bázích. Nicméně použití volné abelovské grupy, potažmo modulu nad \mathbb{Z} , má i jistou výhodu. Tu probereme později v této kapitole.

Pro libovolný simpliciací komplex K a pro $k \in \mathbb{Z}$ a $k > \dim(K)$ jsou všechny moduly k -řetězců triviální. Platí tedy $C_k(K, R) \cong 0$. Tento fakt plyne jednoduše z toho, že simpliciací komplex K neobsahuje žádné takové k -simplexy.

Definice 4.28 (Hraniční operátor)

Nechť C_k a C_{k-1} jsou moduly k -řetězců a $(k-1)$ -řetězců simplicciálního komplexu K . Pro $k > 0$ definujme homomorfismus $\partial_k : C_k \rightarrow C_{k-1}$ určený jeho hodnotami vzhledem k prvkům bázi C_k a C_{k-1} . Nechť $\sigma = [v_0, v_1, \dots, v_k]$ značí k -simplex $\sigma \in K$. Definujme homomorfismus ∂_k předpisem

$$\partial_k(\sigma) = \sum_{i=0}^k (-1)^i [v_0, v_1, \dots, \hat{v}_i, \dots, v_k],$$

kde $[v_0, v_1, \dots, \hat{v}_i, \dots, v_k]$ značí $(k-1)$ -simplex s vynechaným vrcholem v_i .

Pro $k = 0$ a řetězec $c \in C_k$ definujme homomorfismus ∂_k předpisem

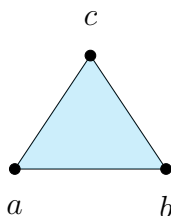
$$\partial_k(c) = 0.$$

Homomorfismus ∂_k se nazývá *hraniční operátor*.

Hraniční operátor tedy jednoduše daný simplex σ zobrazí na řetězec složený právě ze všech jeho stěn. Jelikož je hraniční operátor definován vzhledem k bázím, lze jej přirozeně rozšířit na homomorfismus modulů.

PŘÍKLAD 4.29

Ukážeme si využití předešlých definic na jednoduchém příkladu. Uvažujme následující simplicciální komplex:



Tento simplicciální komplex obsahuje jeden 2-simplex $[a, b, c]$, tři 1-simplexy $[a, b]$, $[b, c]$ a $[a, c]$ a tři 0-simplexy $[a]$, $[b]$ a $[c]$.

Sestavme z nich R -moduly k -řetězců a necht' pro obecnost je R libovolný obor hlavních ideálů. Netriviální moduly k -řetězců tohoto simplicciálního komplexu jsou tedy moduly C_2, C_1 a C_0 .

Modul 2-řetězců C_2 má bázi tvořenou pouze jediným 2-simplexem $[a, b, c]$. Hraniční operátor jej zobrazí na 1-řetězec tvořený jeho stěnami:

$$\partial_2([a, b, c]) = [b, c] - [a, c] + [a, b].$$

Všechny 2-řetězce v modulu C_2 jsou pak ve tvaru $r[a, b, c]$, kde $r \in R$. Hraniční operátor je tedy zobrazí na následující 1-řetězec:

$$\partial_2(r[a, b, c]) = r[b, c] - r[a, c] + r[a, b].$$

Báze modulu 1-řetězců C_1 je tvořena třemi 1-simplexy: $[a, b]$, $[a, c]$ a $[b, c]$. Podobně jako v předchozím případě, hraniční operátor tyto simplexy zobrazí na řetězce tvořené jejich stěnami:

$$\begin{aligned}\partial_1([a, b]) &= [b] - [a], \\ \partial_1([a, c]) &= [c] - [a], \\ \partial_1([b, c]) &= [c] - [b].\end{aligned}$$

Všechny 1-řetězce jsou pak ve tvaru $r[a, b] + s[a, c] + t[b, c]$, kde $r, s, t \in R$. Hraniční operátor je zobrazí na následující 1-řetězec:

$$\begin{aligned}\partial_1(r[a, b] + s[a, c] + t[b, c]) &= r[b] - r[a] + s[c] - s[a] + t[c] - t[b] \\ &= (r - t)[b] - (r + s)[a] + (s + t)[c].\end{aligned}$$

Modul 0-řetězců C_0 je generován bází obsahující tři simplexy: $[a]$, $[b]$ a $[c]$. Tento modul tedy obsahuje všechny řetězce ve tvaru $r[a] + s[b] + t[c]$, kde $r, s, t \in R$. Podle definice hraničního operátoru je obrazem libovolného 0-řetězce prázdný řetězec. To odpovídá intuitivní představě, jelikož 0-simplexy už žádné další stěny neobsahují.

Věta 4.30 (Základní věta hraničního operátoru)

Pro všechna $k \geq 0$ platí $\partial_k \circ \partial_{k+1} = 0$.

Důkaz

Pro $k = 0$ platí věta přímo z definice hraničního operátoru. Pro $k > 0$ máme

$$\begin{aligned}(\partial_k \circ \partial_{k+1})(\sigma) &= \partial_k \left(\sum_{i=0}^k (-1)^i [v_0, v_1, \dots, \hat{v}_i, \dots, v_k] \right) \\ &= \sum_{j < i} (-1)^i (-1)^j [v_0, v_1, \dots, \hat{v}_j, \dots, \hat{v}_i, \dots, v_k] \\ &\quad + \sum_{j > i} (-1)^i (-1)^{j-1} [v_0, v_1, \dots, \hat{v}_i, \dots, \hat{v}_j, \dots, v_k] \\ &= 0.\end{aligned}$$

□

Pomocí hraničních operátorů můžeme znázornit vztah mezi jednotlivými moduly k -řetězců v simplicialním komplexu K diagramem

$$0 \xrightarrow{0} C_k \xrightarrow{\partial_k} C_{k-1} \xrightarrow{\partial_{k-1}} \dots \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} 0,$$

kde pro všechna $k \geq 0$ platí $\partial_k \circ \partial_{k+1} = 0$. Nalevo můžeme zapsat „0“, jelikož pro $k > \dim(K)$ platí $C_k \cong 0$. Protože simplicialní komplex obsahuje pouze konečný počet simplexů, stačí nám tuto „0“ zapsat v diagramu pouze jedenkrát. Všechna ostatní zobrazení by pak totiž pouze zobrazovala nulu triviálního modulu C_k na nulu triviálního modulu C_{k-1} . Napravo můžeme diagram rozšířit o „0“ jelikož z definice hraničního operátoru platí $\partial_0(C_0) = 0$.

Věta 4.31

$\text{im}(\partial_{k+1})$ a $\ker(\partial_k)$ jsou volné podmoduly C_k .

Důkaz

Hraniční operátor ∂_k je homomorfismem modulů. $\text{Im}(\partial_{k+1})$ a $\ker(\partial_k)$ jsou tedy podmoduly C_k . Jelikož C_k je volný modul, použitím věty 3.27 dostáváme, že i $\text{im}(\partial_{k+1})$ a $\ker(\partial_k)$ jsou volné. \square

Věta 4.32

$\text{im}(\partial_{k+1})$ je volný podmodul $\ker(\partial_k)$.

Důkaz

Věta 4.30 říká, že $\partial_k \circ \partial_{k+1} = 0$. Tudíž $\text{im}(\partial_{k+1}) \subseteq \ker(\partial_k)$. Z věty 3.27 pak plyne, že i $\text{im}(\partial_{k+1})$ je volný modul. \square

Definice 4.33 (Okraje, Cykly)

Označme $B_k = \text{im}(\partial_{k+1})$ a $Z_k = \ker(\partial_k)$. Prvky B_k nazýváme *okraje*, prvky Z_k nazýváme *cykly*.

Okraje jsou tedy ty řetězce v C_k , které jsou tvořeny stěnami nějakých simplexů v C_{k+1} a dále všechny obrazy řetězců, které jsou z těchto simplexů složeny. Cykly jsou řetězce s nulovým okrajem.

PŘÍKLAD 4.34

Pokračujme v předchozím příkladu 4.29. Simplicialní komplex v něm obsahoval pouze jeden 2-simplex $[a, b, c]$. Jeho okrajem je 1-řetězec:

$$\partial_2([a, b, c]) = [b, c] - [a, c] + [a, b].$$

Podle předchozí definice je tedy modul B_1 generovaný pouze tímto jedním řetězcem. Tudíž B_1 obsahuje všechny 1-řetězce ve tvaru:

$$\partial_2(r[a, b, c]) = r[b, c] - r[a, c] + r[a, b], \text{ kde } r \in R.$$

Nyní se podíváme na Z_1 . Podle definice $Z_1 = \ker(\partial_1)$. Snadno se ale vidí, že jediné 1-řetězce, které mají nulový okraj, jsou 1-řetězce ve tvaru:

$$r[b, c] - r[a, c] + r[a, b], \text{ kde } r \in R.$$

V tomto případě tedy platí $Z_1 = B_1$. Obecně to však neplatí.

Pro $k = 0$ je situace vždy jednodušší. Z definice hraničního operátoru totiž plyne, že $Z_0 = C_0$. Modul B_0 pak obsahuje řetězce ve tvaru:

$$(r - t)[b] - (r + s)[a] + (s + t)[c], \text{ kde } r, s, t \in R.$$

Definice 4.35 (Homologie)

Vezměme $H_k = Z_k/B_k = \ker(\partial_k)/\text{im}(\partial_{k+1})$. H_k se nazývá K -tá homologie (někdy *homologická grupa*) simplicciálního komplexu K . Pokud dva cykly $z_1, z_2 \in Z_k$ spadají do stejné třídy ekvivalence, říkáme, že jsou *homologické*.

Z definice 3.26 faktorového modulu plyne, že prvky H_k jsou třídy ekvivalence ve tvaru $z + B_k$, kde $z \in Z_k$. Dva cykly $z_1, z_2 \in Z_k$ jsou tedy homologické právě tehdy, když $z_1 + B_k = z_2 + B_k$. To také znamená, že $z_1 - z_2$ je okrajem nějakého $(k+1)$ -cyklu a tedy $z_1 - z_2 \in B_k$. Nulou H_k je pak třída $0 + B_k = B_k$.

Připomeňme si větu 3.28 o struktuře konečně generovaných modulů. Pokud R je oborem hlavních ideálů, pak každý konečně generovaný modul M nad R je izomorfní s určitým přímým součtem cyklických modulů nad R . Pro tento modul pak existuje následující izomorfismus:

$$M \cong R^\beta \oplus \left(\bigoplus_{i=1}^m R/\langle r_i \rangle \right),$$

kde $r_i \in R, m, \beta \in \mathbb{N}$ a $r_i | r_{i+1}$. Číslo β udává počet takzvaných volných komponent. Komponenty ve tvaru $R/\langle r_i \rangle$, kde $r_i > 1$, se nazývají torze.

Jelikož modul H_k je faktorovým modulem dvou volných modulů, je tento modul generován konečnou množinou prvků. Platí tedy pro něj výše uvedená věta a proto můžeme homologii H_k vyjádřit následovně:

$$H_k = Z_k/B_k \cong R^\beta \oplus \left(\bigoplus_{i=1}^m R/\langle r_i \rangle \right).$$

Nyní je vhodné zdůraznit, že struktura H_k je závislá tom, jaký obor hlavních ideálů při tvorbě modulů C_k zvolíme.

Pokud například použijeme pole \mathbb{Z}_2 , nejsme schopni odhalit žádné torze. To plyne jednoduše z toho, že \mathbb{Z}_2 obsahuje pouze dva prvky, 0 a 1. Pokud tedy $r_i = 1$, pak $\mathbb{Z}_2/1\mathbb{Z}_2 \cong 0$ a pokud $r_i = 0$, pak $\mathbb{Z}_2/0\mathbb{Z}_2 \cong \mathbb{Z}_2$. Tudíž struktura H_k pak bude izomorfní se \mathbb{Z}_2^β pro nějaké $\beta \in \mathbb{N}$.

Naopak nejbohatší popis struktury H_k získáme, pokud použijeme koeficienty v \mathbb{Z} . V popisu struktury H_k totiž získáme i všechny možné torze.

Může se tedy například stát, že struktury homologií dvou simplicciálních komplexů, které by měly při použití koeficientů \mathbb{Z} stejný počet volných komponent, ale různé torze, budou mít při použití \mathbb{Z}_2 strukturu stejnou. Čili z pohledu simplicciální homologie se budou jevit jako ekvivalentní.

Nicméně v reálných datasetech se torze vyskytují zřídka kdy. Za významnou informaci se tedy většinou udává počet volných komponent ve struktuře H_k . Dostává proto i vlastní definici:

Definice 4.36 (Bettiho číslo)

Nechť H_k je k -tá homologie simplicciálního komplexu K . Definujme k -té Bettiho číslo β_k jako počet volných komponent H_k .

Protože Z_k i B_k jsou volné moduly nad oborem hlavních ideálů, platí

$$\beta_k = \text{rank}(Z_k) - \text{rank}(B_k). \quad (4.2)$$

Nyní si vysvětlíme význam čísel β_k , které lze ze struktury H_k získat. Čísla β_k udávají počet takzvaných *k-rozměrných děr* v daném simplicciálním komplexu.

Přesněji, *k-rozměrná díra* je díra tvořená nějakým *k-cyklem* $z \in Z_k, z \notin B_k$. Tento cyklus pak v H_k generuje jednu třídu ekvivalence ve tvaru $z + B_k$.

Mějme tedy *k-cyklus* $z \in Z_k, z \notin B_k$. Jelikož předpokládáme, že $z \in Z_k, z \notin B_k$, pak také víme, že při faktorizaci Z_k/B_k *k-cyklus* z bude tvořit třídu ekvivalence $z + B_k \in H_k$.

V závislosti na zvoleném okruhu koeficientů R , modul C_k pak také obsahuje všechny cykly ve tvaru $r \cdot z \in C_k, r \in R$. Z toho také plyne, že pro všechna $r \in R$ máme $r \cdot z + B_k \in H_k$.

Vidíme tedy, že pokud existuje nějaký *k-cyklus* $z \in Z_k, z \notin B_k$, pak C_k obsahuje všechny jeho násobky. Díky tomu také H_k obsahuje k těmto násobkům odpovídající třídy ekvivalence.

Je tedy zřejmé, že množina všech těchto násobků *k-cyklu* z tvoří podmodul a ten je izomorfní s R . Podobně množina jim odpovídajících tříd ekvivalencí v H_k tvoří podmodul H_k a tento podmodul je izomorfní s R .

Podmodul H_k ve tvaru $r \cdot z + B_k \in H_k, r \in R$ nazýváme *homologická třída*. Počet těchto homologických tříd v modulu H_k pak odpovídá přesně číslu β_k .

Pokud jsou dva cykly z_1 a z_2 homologické, pak oba generují tu stejnou homologickou třídu.

Počet homologických tříd dimenze 0, tedy 0-dimenzionálních děr, odpovídá počtu souvislých komponent v daném simplicciálním komplexu. Analogicky jako u grafů, souvislá komponenta je taková množina, ve které mezi libovolnými dvěma vrcholy existuje cesta. Stačí si uvědomit, že 0-cykly jsou cykly tvořené pouze vrcholy. V B_0 jsou pak ty 0-cykly, které jsou tvořeny obrazem nějakých 1-cyklů. 1-cykly jsou ale hrany.

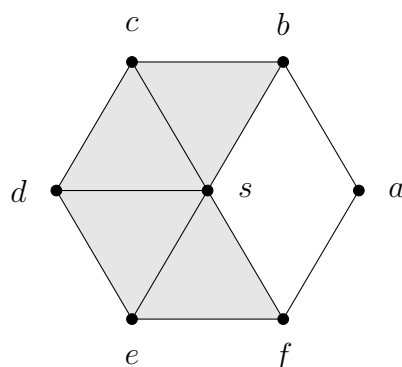
Pokud tedy H_0 obsahuje právě jednu homologickou třídu, pak to znamená, že mezi všemi vrcholy existuje nějaká cesta. Pokud H_0 obsahuje například dvě homologické třídy, pak to znamená, že v simplicciálním komplexu jsou dvě skupiny vrcholů, které nejsou mezi sebou spojeny žádnou hranou.

PŘÍKLAD 4.37

Uvažujme simplicciální komplex znázorněný na obrázku 4.7. Sestavme nyní moduly *k-řetězců* s koeficienty v \mathbb{Z}_2 . Vztahy mezi těmito moduly můžeme znázornit diagramem

$$0 \xrightarrow{0} C_2 \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} 0.$$

Určíme nyní strukturu homologie H_1 . Jelikož používáme koeficienty v \mathbb{Z}_2 , pak pro úplné určení struktury H_1 stačí určit číslo β_1 . Použitím koeficientů v \mathbb{Z}_2 také odpadá nutnost používat u řetězců znaménka. Podle rovnice 4.2 je tedy k určení čísla β_1 nutné spočítat $\text{rank}(Z_1)$ a $\text{rank}(B_1)$.

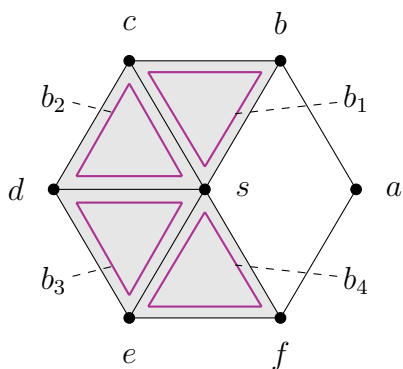


Obrázek 4.7: Simplicialní komplex s jednou jednodimenzionální dírou.

Tento simplicialní komplex obsahuje celkem čtyři 2-simplexy (trojúhelníky). Z definice B_1 pak plyne, že modul B_1 je generován bází tvořenou okraji těchto čtyř simplexů. Například okrajem 2-simplexu $[c, b, s]$ je 1-řetězec $[b, s] + [c, s] + [c, b]$. Dostáváme tedy, že $\text{rank}(B_1) = 4$. Jednotlivé okraje jsou znázorněny na obrázku a) níže.

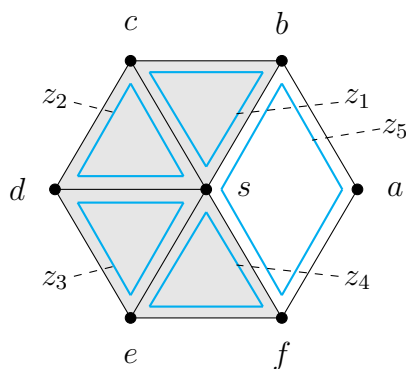
Nyní je třeba určit $\text{rank}(Z_1)$. Modul Z_1 je tvořen všemi 1-cykly, tedy 1-řetězci s nulovým okrajem. Například 1-řetězec $z_1 = [b, s] + [c, s] + [c, b]$ má nulový okraj, neboť $\partial_1(z_1) = 0$. Pro přehlednost jsou všechny bazové cykly modulu Z_1 vyznačeny na obrázku b) níže. Všimněme si, že řetězec $z_5 = [a, b] + [b, s] + [f, s] + [a, f]$ má také nulový okraj. Modul Z_1 je tedy generován bází obsahující 5 cyklů a platí tedy $\text{rank}(Z_1) = 5$.

Dostáváme tedy, že $\beta_1 = 1$. Z obrázku je vidět, že daný simplicialní komplex opravdu obsahuje jednu 1-dimenzionální díru. Tato díra je tvořena 1-cyklem z_5 . Platí tedy $H_1 \cong \mathbb{Z}_2$.



a) - okraje

$$\begin{aligned} b_1 &= [b, c] + [b, s] + [c, s] \\ b_2 &= [c, d] + [c, s] + [d, s] \\ b_3 &= [d, e] + [d, s] + [e, s] \\ b_4 &= [e, f] + [e, s] + [f, s] \end{aligned}$$



b) - cykly

$$\begin{aligned} z_1 &= [b, c] + [b, s] + [c, s] \\ z_2 &= [c, d] + [c, s] + [d, s] \\ z_3 &= [d, e] + [d, s] + [e, s] \\ z_4 &= [e, f] + [e, s] + [f, s] \\ z_5 &= [a, b] + [b, s] + [f, s] + [a, f] \end{aligned}$$

Intuitivně si situaci můžeme představit tak, že každý jeden okraj v B_1 zalepí jednu díru tvořenou cyklem ze Z_1 . Nakonec zde zbyde jedna nezalepená díra.

Teď si popíšeme situaci pomocí algebry. Modul H_1 obsahuje pouze dva prvky. První prvek je B_1 a druhý prvek je $[a, b] + [b, s] + [f, s] + [a, f] + B_1$. Z definice faktorového modulu plyne, že první prvek B_1 je nulou tohoto modulu. Druhý prvek pak generuje jednu homologickou třídu, která je započítána do čísla β_1 .

Na závěr se ještě podíváme na strukturu H_0 . Tento simplicciální komplex obsahuje pouze jednu souvislou komponentu, platí tedy $H_0 \cong \mathbb{Z}_2$.

Tímto způsobem pak probíhá sčítání libovolného počtu děr v libovolné dimenzi simplicciálního komplexu.

PŘÍKLAD 4.38

Díru dimenze 2 obsahuje například simplicciální komplex, který je triangulací sféry \mathbb{S}^2 . Každý simplicciální komplex, který je triangulací sféry \mathbb{S}^n , pak obsahuje jednu díru dimenze n .

4.4 Persistentní homologie

V minulé kapitole jsme měli jeden pevně zadaný simplicialní komplex a ten jsme pomocí homologie zkoumali. Nicméně simplicialní komplex se může vyvíjet.

V praxi se totiž simplicialní komplexy většinou vytváří inkrementálně. Začneme například s určitým simplicialním komplexem a postupně do něj přidáváme další simplex.

Pokud do nějakého simplicialního komplexu přidáme nějaký simplex, mohou se struktury homologií H_k tohoto simplicialního komplexu různě měnit. Právě tyto změny homologií během přidávání simplexů jistým způsobem zachycuje *persistentní homologie*. V této kapitole si tedy postupně definujeme potřebné pojmy a následně vytvoříme takovou algebraickou strukturu, která dokáže tyto změny zachytit.

Začneme s pojmem *filtrace*, který formálně definuje inkrementální vývoj simplicialních komplexů:

Definice 4.39 (Filtrace)

Filtrace simplicialního komplexu K je následující konečná posloupnost podkomplexů K :

$$\emptyset = K^0 \leq K^1 \leq K^2 \leq \dots \leq K^{n-1} \leq K^n = K.$$

Simplicialní komplex spolu s jeho filtrací nazýváme *filtrovaný simplicialní komplex*.

Číslům l , pro $0 \leq l \leq n$, budeme říkat *krok filtrace*. V literatuře se l někdy chápe také jako čas.

Je pouze jediná možnost, jak ze simplicialního komplexu K^l můžeme vytvořit simplicialní komplex K^{l+1} tak, aby vznikl filtrovaný simplicialní komplex K : přidat do K^l nějaké nové simplex.

Z definice simplicialního komplexu také plyne, že simplicialní komplex musí obsahovat i stěny všech simplexů jakožto samostatné simplex. To znamená, že pokud přidáváme simplex σ do simplicialního komplexu v určitém kroku, pak nejpozději v tom stejném kroku musíme do daného simplicialního komplexu přidat také všechny jeho stěny.

Mějme tedy filtrovaný simplicialní komplex K . V každém kroku l filtrace lze ze simplicialního komplexu K^l sestavit jemu příslušný modul k -řetězců C_k^l .

Pro dimenzi k tedy můžeme moduly k -řetězců C_k^l znázornit pomocí následujícího diagramu:

$$C_k^0 \xrightarrow{f^0} C_k^1 \xrightarrow{f^1} \dots \xrightarrow{f^{l-1}} C_k^l \xrightarrow{f^l} \dots \xrightarrow{f^{n-2}} C_k^{n-1} \xrightarrow{f^{n-1}} C_k^n,$$

kde zobrazení $f^l : C_k^l \rightarrow C_k^{l+1}$ je definováno jako inkluze.

Fakt, že taková inkluze f^l existuje, plyne z toho, že ve filtraci simplicialního komplexu prvky pouze přidáváme a moduly k -řetězců C_k^l jsou následně generované bází obsahující právě k -simplexy v C_k^l .

Pro každé dva moduly C_k^l a C_{k-1}^l , kde $0 \leq l \leq n$, je dán hraniční operátor ∂_k^l . Vztah mezi posloupností modulů C_k^l a C_{k-1}^l si tedy můžeme znázornit pomocí následujícího komutativního diagramu:

$$\begin{array}{cccccccccccc}
 C_k^0 & \xrightarrow{f^0} & C_k^1 & \xrightarrow{f^1} & \dots & \xrightarrow{f^{l-1}} & C_k^l & \xrightarrow{f^l} & \dots & \xrightarrow{f^{n-2}} & C_k^{n-1} & \xrightarrow{f^{n-1}} & C_k^n \\
 \partial_k^0 \downarrow & & \partial_k^1 \downarrow & & & & \partial_k^l \downarrow & & & & \partial_k^{n-1} \downarrow & & \partial_k^n \downarrow \\
 C_{k-1}^0 & \xrightarrow{f^0} & C_{k-1}^1 & \xrightarrow{f^1} & \dots & \xrightarrow{f^{l-1}} & C_{k-1}^l & \xrightarrow{f^l} & \dots & \xrightarrow{f^{n-2}} & C_{k-1}^{n-1} & \xrightarrow{f^{n-1}} & C_{k-1}^n
 \end{array}$$

Označení *komutativní* znamená, že všechna složení zobrazení ve směru šipek mezi libovolnými dvěma objekty v diagramu dávají stejný výsledek.

V každém kroku filtrace l a v každé dimenzi k tedy můžeme sestavit homologii H_k^l a určit tak Bettiho čísla β_k^l .

V minulé kapitole jsme si řekli, že to hlavní, co můžeme ze struktury homologie H_k^l zjistit, je počet děr v simplicialním komplexu K^l . Každou díru reprezentuje nějaká homologická třída. Každá homologická třída je pak generována nějakým cyklem z .

Předpokládejme, že přidáním nějakých simplexů do simplicialního komplexu K^l v kroku l vznikne k -cyklus $z \in Z_k^l$, $z \notin B_k^l$. V H_k^l pak tento cyklus generuje homologickou třídu, která je započítána do čísla β_k^l . Tato homologická třída tedy reprezentuje díru, která existuje v simplicialním komplexu K^l . Říkáme, že daná homologická třída se *narodila* v kroku l .

Dále uvažujme, že přidáním nějakých dalších simplexů do simplicialního komplexu K^m v kroku m se tento k -cyklus z stane okrajem. Nyní tedy platí $z \in B_k^m$. Zároveň také k -cyklus z již negeneruje samostatnou homologickou třídu v H_k^m . Bettiho číslo β_k^m se tedy o jedna sníží. Říkáme, že v tomto kroku daná homologická třída *zanikla*.

Persistenci této homologické třídy rozumíme číslo $m - l - 1$. Pokud daná homologická třída nikdy nezanikla, pak její persistenci pokládáme rovnu nekonečnu.

Další postup již bude velmi podobný s postupem v předchozí kapitole. Naším cílem je vytvořit jisté algebraické struktury, ze kterých pak bude možné zjistit persistence daných homologických tříd ve filtraci simplicialního komplexu.

V tomto případě to budou gradované moduly.

Mějme tedy filtrovaný simplicialní komplex K . Nejprve začneme takzvaným *stupněm simplexu*. *Stupněm simplexu* $\sigma \in K$ se rozumí číslo kroku prvního výskytu tohoto simplexu ve filtraci simplicialního komplexu K . Tento stupeň značíme $\deg(\sigma)$. Pokud je tedy simplex σ poprvé přidán do simplicialního komplexu K^l , bude jeho stupeň $\deg(\sigma) = l$.

Posloupnost modulů C_k^l použijeme k vytvoření jedné abelovské grupy pomocí přímého součtu. Každý modul C_k^l můžeme uvažovat jako abelovskou grupu, jelikož z definice modulu plyne, že množina prvků C_k^l spolu s operací sčítání abelovskou grupu tvořit musí. Nechť n je poslední krok filtrace a nechť C_k^n je odpovídající modul k -řetězců. Pro $i > n$ definujme $C_k^i = C_k^n$ a položme:

$$\mathcal{C}_k = \bigoplus_{i=0}^{\infty} C_k^i.$$

Operace na \mathcal{C}_k je definována po složkách pomocí jednotlivých operací z C_k^i .

POZNÁMKA 4.40

Prvky \mathcal{C}_k tedy lze zapsat jak pomocí n -tic, tak pomocí formálních lineárních kombinací. Většinou se používá ten zápis, který je v dané situaci výhodnější.

Nakonec ještě pro všechna $i \geq 0$ definujme inkluzi $f^i : C_k^i \rightarrow C_k^{i+1}$. Pomocí těchto pojmů nyní můžeme vytvořit gradovaný modul:

Definice 4.41 (Persistentní modul k -řetězců)

Mějme filtrovaný simplicialní komplex K , buď F libovolné pole a mějme posloupnost modulů k -řetězců C_k s koeficienty v poli F , které odpovídají simplicialním komplexům ve filtraci simplicialního komplexu K . *Persistentní modul k -řetězců*³ je definován jako gradovaný modul \mathcal{C}_k nad $F[t]$ s bází tvořenou všemi k -simplexy obsaženými v K . Operace sčítání je definována jako obvyklé sčítání k -řetězců. Operace násobení skalárem je definována jako posun prvků modulu napříč jeho gradováním pomocí dané inkluze f^i . Pro homogenní polynom $at \in F[t]$ stupně 1 a řetězec $c \in \mathcal{C}_k$, $c = (c^0, c^1, c^2, \dots)$ a kde $c^i \in C_k^i$ tedy položme

$$at \cdot (c^0, c^1, c^2, \dots) = (0, af^0(c^0), af^1(c^1), af^2(c^2), \dots)$$

a pro polynomy stupně $d > 1$ aplikujeme d posunutí definovaných pro polynom stupně 1. Persistentní modul k -řetězců značíme \mathcal{C}_k .

Tento modul je sestaven z nekonečného počtu modulů C_k^l , nicméně z filtrovaného simplicialního komplexu lze získat pouze konečný počet těchto modulů. Tento krok lze chápat pouze jako nutnou formalitu.

Gradovaná struktura modulu \mathcal{C}_k tedy do jisté míry zrcadlí průběh filtrace simplicialního komplexu. Pokud se k -simplex σ během filtrace poprvé vyskytne v komplexu K^l , bude jemu odpovídající jednoprvkový k -řetězec v modulu \mathcal{C}_k obsažen v gradování nejdříve v komponentě C_k^l .

Mějme tedy jednoprvkový k -řetězec $\sigma \in C_k^l$ reprezentující tentýž simplex $\sigma \in K$. Vynásobením polynomem t pak tento k -řetězec posune v gradování do komponenty C_k^{l+1} .

³V literatuře tento modul většinou nemá jméno explicitně uvedeno. Obecně jej lze odvodit z takzvaného *Persistentního modulu*. Viz [9], strana 100.

K -řetězec σ značí, že se simplex σ poprvé nachází ve filtraci v komplexu $K^{\deg(\sigma)}$. K -řetězec $t\sigma \in \mathcal{C}_k$ značí, že se simplex σ nachází ve filtraci v komplexu $K^{1+\deg(\sigma)}$. Obecně k -řetězec $t^n\sigma \in \mathcal{C}_k$ značí, že se simplex σ nachází ve filtraci v komplexu $K^{n+\deg(\sigma)}$.

Pro k -řetězec $c \in \mathcal{C}_k$ položme

$$\deg(c) = \max \{n + \deg(\sigma) \mid at^n \cdot \sigma \in c, \}.$$

Hraniční operátor mezi dvěma gradovanými moduly \mathcal{C}_k a \mathcal{C}_{k-1} pak definujeme s využitím hraničních operátorů mezi jednotlivými moduly \mathcal{C}_k^l a \mathcal{C}_{k-1}^l :

Definice 4.42 (Persistentní hraniční operátor)

Nechť \mathcal{C}_k a \mathcal{C}_{k-1} jsou persistentní moduly k -řetězců a $(k-1)$ -řetězců. *Persistentní hraniční operátor* $\widehat{\partial}_k : \mathcal{C}_k \rightarrow \mathcal{C}_{k-1}$ je definován předpisem

$$\widehat{\partial}_k(c^0, c^1, c^2, \dots) = (\partial_k^0(c^0), \partial_k^1(c^1), \partial_k^2(c^2), \dots),$$

kde pro $l \geq 0$ je ∂_k^l příslušný hraniční operátor $\mathcal{C}_k^l \rightarrow \mathcal{C}_{k-1}^l$.

Snadno se vidí, že platí $\widehat{\partial}_k \circ \widehat{\partial}_{k+1} = 0$. Díky tomu lze jednoduše definovat k -tou persistentní homologii:

Definice 4.43 (Persistentní homologie)

Nechť \mathcal{C}_k a \mathcal{C}_{k-1} jsou persistentní moduly k -řetězců a $(k-1)$ -řetězců a nechť $\widehat{\partial}_k : \mathcal{C}_k \rightarrow \mathcal{C}_{k-1}$ je jejich hraniční operátor. Položme $\mathcal{Z}_k = \ker(\widehat{\partial}_k)$ a $\mathcal{B}_k = \text{im}(\widehat{\partial}_{k+1})$. K -tá *persistentní homologie* je definována následovně:

$$\mathcal{H}_k = \mathcal{Z}_k / \mathcal{B}_k.$$

Jelikož \mathcal{H}_k je konečně generovaný nezáporně gradovaný modul nad okruhem polynomů $F[t]$, můžeme uplatit větu 3.35 o struktuře těchto modulů. Platí tedy:

$$\mathcal{H}_k \cong \left(\bigoplus_{s=1}^m \langle t^{i_s} \rangle \right) \oplus \left(\bigoplus_{t=1}^n \langle t^{j_t} \rangle / \langle t^{l_t} \rangle \right),$$

pro nějaká celá čísla $i_1, \dots, i_m, j_1, \dots, j_n$ a l_1, \dots, l_n .

Struktura \mathcal{H}_k jistým způsobem reprezentuje persistenci daných homologických tříd během filtrace simplicialního komplexu. Než si vysvětlíme jak, zavedeme si následující pojem:

Definice 4.44 (\mathcal{P} -interval)

\mathcal{P} -interval je uspořádaná dvojice (i, j) , kde $i \in \mathbb{Z}$, $j \in \mathbb{Z} \cup \{+\infty\}$ a $j \geq i$.

Věta 4.45

Pro každý konečně generovaný nezáporně gradovaný modul M nad $F[t]$ existuje až na pořadí právě jedna konečná posloupnost \mathcal{P} -intervalů. Pro libovolnou konečnou posloupnost \mathcal{P} -intervalů existuje až na izomorfismus právě jeden konečně generovaný nezáporně gradovaný modul M nad $F[t]$.

Důkaz

První část věty je důsledkem věty 3.35. Pro celá čísla i_1, \dots, i_m vezmeme \mathcal{P} -intervaly $(i_1, +\infty), \dots, (i_m, +\infty)$ a pro čísla j_1, \dots, j_n a l_1, \dots, l_n vezmeme \mathcal{P} -intervaly $(j_1, l_1), \dots, (j_n, l_n)$.

Pro důkaz druhé části vytvoříme na základě daných \mathcal{P} -intervalů gradovaný modul. Necht F je tedy pole a $F[t]$ okruh polynomů s koeficienty v F . Pro \mathcal{P} -interval (i, j) , kde $j \neq +\infty$ definujeme:

$$\mathcal{Q}(i, j) = \langle t^i \rangle / \langle t^j \rangle$$

a pro \mathcal{P} -interval $(i, +\infty)$ definujeme:

$$\mathcal{Q}(i, +\infty) = \langle t^i \rangle$$

Pro konečnou posloupnost \mathcal{P} -intervalů $(i_1, j_1)_1, (i_2, j_2)_2, \dots, (i_n, j_n)_n$ položme

$$\mathcal{M} = \bigoplus_{l=1}^n \mathcal{Q}(i_l, j_l)_l$$

a uvažujme \mathcal{M} modul nad $F[t]$ s operacemi sčítání a násobení skalárem, které jsou odvozené z $F[t]$ a definované po složkách. \square

V persistentní homologii tedy každé filtraci simplicialního komplexu K přiřadíme persistentní modul \mathcal{H}_k . Strukturu tohoto persistentního modulu pak lze jednoznačně reprezentovat pomocí posloupnosti \mathcal{P} -intervalů.

Jednotlivé \mathcal{P} -intervaly pak popisují vznik, zánik a persistenci jednotlivých homologických tříd během filtrace.

Konkrétně, \mathcal{P} -interval (i, j) , kde $j \neq +\infty$ značí, že v dané filtraci simplicialního komplexu existuje homologická třída, která se zrodí v kroku i a zanikne v kroku j . \mathcal{P} -interval $(i, +\infty)$ pak značí, že v dané filtraci existuje homologická třída, která se zrodí v kroku i a nikdy nezanikne.

5 Ripsův komplex

V této kapitole si ukážeme, jak lze vytvořit simplicialní komplex z libovolné množiny prvků a také jak vytvořit filtraci takového simplicialního komplexu. Ukážeme si také způsob, jakým se graficky znázorňují persistence jednotlivých homologických tříd ve filtrovaném simplicialním komplexu.

Informace v této kapitole jsou čerpány ze zdroje [2].

Začneme tedy s vytvářením simplicialního komplexu:

Definice 5.1 (Ripsův komplex)

Nechť X je konečná množina a vezměme libovolné zobrazení $d : X \times X \rightarrow \mathbb{R}$ a číslo $\epsilon \in \mathbb{R}$. *Ripsovým komplexem*⁴ rozumíme abstraktní simplicialní komplex s množinou vrcholů X , který obsahuje všechny abstraktní simplex ve tvaru:

$$\sigma = \{x_0, x_1, \dots, x_k \mid k \geq 0, x_i, x_j \in X \text{ a } d(x_i, x_j) \leq \epsilon \text{ pro } i, j \in \{0, \dots, k\}\}.$$

Ripsův komplex s parametrem ϵ nad množinou X značíme $VR(X, \epsilon)$.

Ripsův komplex $VR(X, \epsilon)$ tedy obsahuje takové simplex, jejichž vrcholy jsou dle d vzájemně vzdáleny maximálně o ϵ .

Z definice přímo plyne, že $VR(X, \epsilon_1) \leq VR(X, \epsilon_2)$ pro $\epsilon_1 \leq \epsilon_2$. Díky tomu, že množina X je konečná, existuje takové ϵ_{max} , že pro všechna $\epsilon > \epsilon_{max}$ platí $VR(X, \epsilon) = VR(X, \epsilon_{max})$.

Zobrazení d může být libovolné. Definice je tedy velmi obecná. Často se však za d volí nějaká *metrika*:

Definice 5.2 (Metrika)

Nechť X je množina a $d : X \times X \rightarrow \mathbb{R}$ je zobrazení splňující následující podmínky:

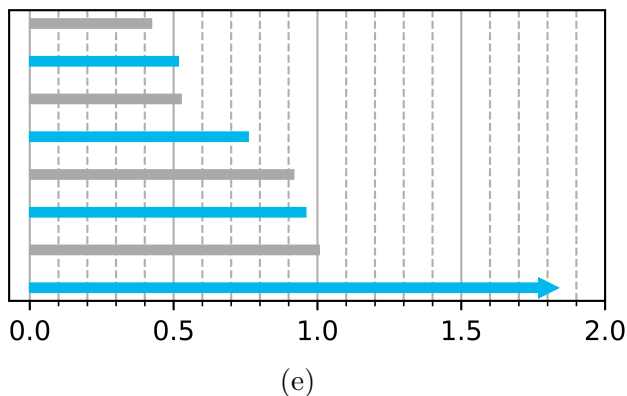
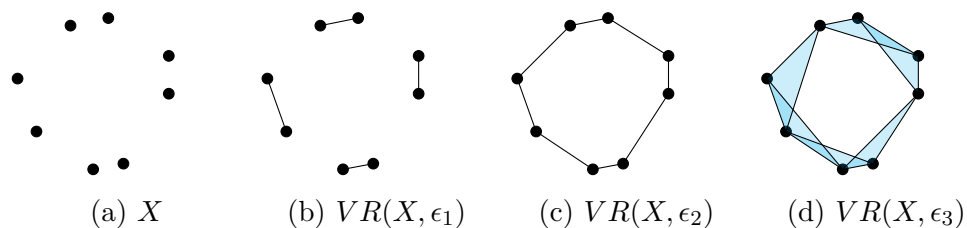
1. $d(x, y) \geq 0$,
2. pokud $d(x, y) = 0$ pak $x = y$,
3. $d(x, y) = d(y, x)$,
4. $d(x, y) + d(y, z) \geq d(x, z)$,

pro všechna $x, y, z \in X$. Zobrazení d se nazývá *metrika* na X .

Definice 5.3 (Ripsova filtrace)

Nechť X je konečná množina prvků a mějme zobrazení $d : X \times X \rightarrow \mathbb{R}$. Dále mějme posloupnost reálných čísel $\epsilon_1, \dots, \epsilon_m$, pro která platí $\epsilon_1 < \dots < \epsilon_m$. Posloupnost simplicialních komplexů $VR(X, \epsilon_1), \dots, VR(X, \epsilon_m)$ tvoří filtraci simplicialního komplexu $VR(X, \epsilon_m)$ a tuto filtraci nazýváme *Ripsova filtrace*. Posloupnosti čísel $\epsilon_1, \dots, \epsilon_m$ se také říká *parametry Ripsovy filtrace*.

⁴Někdy se v literatuře Ripsův komplex označuje také jako *Vietoris–Ripsův komplex*.



Obrázek 5.1: Příklad Ripsovy filtrace s parametry $\epsilon_1 < \epsilon_2 < \epsilon_3$. Na obrázku (a) je původní množina, na obrázcích (b) - (d) je znázorněna podoba Ripsova komplexu s daným parametrem ϵ_i . Na obrázku (e) jsou čárové kódy dimenze 1 k uvedené Ripsově filtraci. Pro lepší čitelnost jsou úsečky tučné a střídavě obarveny.

Vytvoření filtrovaného simplicálního komplexu pomocí Ripsovy filtrace je tedy přímočaré: stačí zvolit nějaké parametry $\epsilon_1, \dots, \epsilon_m$ a vytvořit posloupnost Ripsových komplexů $VR(X, \epsilon_1), \dots, VR(X, \epsilon_m)$.

Jelikož je množina X konečná, je také konečný počet všech různých vzdáleností mezi prvky této množiny. V praxi se pak jako parametry Ripsovy filtrace většinou volí všechny tyto vzdálenosti.

U Ripsovy filtrace se v \mathcal{P} -intervalech místo příslušného kroku uvádí přímo odpovídající hodnoty parametrů. Píšeme tedy $(\epsilon_{i_1}, \epsilon_{i_2})_i$ a $(\epsilon_{i_1}, +\infty)_i$.

Posloupnost \mathcal{P} -intervalů získaných z Ripsovy filtrace znázorňujeme pomocí takzvaných *čárových kódů*.

Mějme tedy posloupnost $(\epsilon_{i_1}, \epsilon_{i_2})_1, \dots, (\epsilon_{i_{n_1}}, \epsilon_{i_{n_2}})_n$ \mathcal{P} -intervalů. Čárový kód je posloupnost úseček vykreslených v grafu v rovině, kde tyto úsečky odpovídají jednotlivým \mathcal{P} -intervalům v posloupnosti. V grafu je pro každý \mathcal{P} -interval $(\epsilon_{i_1}, \epsilon_{i_2})_i$, kde $\epsilon_{i_2} \neq +\infty$, nakreslena úsečka začínající v bodě $x = \epsilon_{i_1}, y = i$ a vedoucí do bodu $x = \epsilon_{i_2}, y = i$. Pro \mathcal{P} -interval $(\epsilon_{i_1}, +\infty)_i$ je v grafu vykreslena dostatečně dlouhá vodorovná úsečka začínající v bodě $x = \epsilon_{i_1}, y = i$ a na pravém konci ukončena šipkou. Hodnoty na ose y tedy odpovídají indexu \mathcal{P} -intervalu v posloupnosti a obvykle se tato osa nevyznačuje vůbec. Osa x reprezentuje parametry Ripsovy filtrace.

Příklad Ripsovy filtrace a čárového kódu je na obrázku 5.1.

6 Algoritmy simplicialní homologie

V této kapitole jsou popsány algoritmy pro výpočet homologie a persistentní homologie. Informace v této kapitole vychází ze zdroje [9], příklady a některé definice jsou vlastní.

6.1 Algoritmus simplicialní homologie

Způsob výpočtu struktur homologií a posléze i persistentních homologií vychází z poznatků lineární algebry. Hlavní myšlenkou je vyjádřit hraniční operátory pomocí matic a následně tyto matice převést do Smithova normálního tvaru⁵. Algoritmus, který matici převede do Smithova normálního tvaru se jednoduše nazývá *redukční algoritmus*.

Tento algoritmus zde neuvеdeme celý, pouze zde popíšeme klíčové operace, které tento algoritmus provádí. Redukční algoritmus totiž musí pracovat v celočíselné aritmetice, což je v praxi obecně problém. Jednotlivé hodnoty v matici mohou během redukce narůst do extrémně velkých hodnot, a tím pádem je výpočet s takto velkými čísly paměťově i časově velmi náročný. Právě z tohoto důvodu se redukční algoritmus pro výpočet homologií nevyužívá. Nicméně, lze na něm pochopit problematiku výpočtu homologií a v tomto duchu je napsána i tato kapitola.

Na konci kapitoly je pak uveden jednoduchý algoritmus pro výpočet homologií, který využívá koeficienty v \mathbb{Z}_2 a v následující kapitole je pak popsán algoritmus, který je z redukčního algoritmu odvozen a je v praxi běžně využíván pro výpočet persistentní homologie.

Předpokládejme tedy, že máme daný simplicialní komplex K . Připomeňme si notaci z kapitoly 4.3 o simplicialní homologii:

- C_k je modul k -řetězců s koeficienty v \mathbb{Z} ,
- $B_k = \text{im}(\partial_{k+1})$ je modul řetězců, které jsou okrajem nějakého řetězce z C_{k+1} ,
- $Z_k = \ker(\partial_k)$ modul řetězců, které mají nulový okraj,
- $H_k = Z_k/B_k$ je k -tá homologie simplicialního komplexu K .

C_k je volný modul, kde k -simplexy tvoří bázi tohoto modulu. Dále pak víme, že pro všechna $k \in \mathbb{Z}_+$ jsou hraniční operátory ∂_k homomorfismy modulů.

⁵Připomenutí pojmu *Smithův normální tvar* je na straně 57.

Z lineární algebry pak víme, že hraniční operátor ∂_k lze reprezentovat pomocí matice. Definujme si tedy tuto matici:

Definice 6.1 (Matice hraničního operátoru)

Nechť je dán simplicialní komplex K , hraniční operátor $\partial_k : C_k \rightarrow C_{k-1}$ a dále necht m_k značí počet k -simplexů v K a m_{k-1} počet $(k-1)$ -simplexů v K . Uspořádejme k -simplexy do posloupnosti $\sigma_1^k, \dots, \sigma_{m_k}^k$ a stejně tak uspořádejme $(k-1)$ -simplexy do posloupnosti $\sigma_1^{k-1}, \dots, \sigma_{m_{k-1}}^{k-1}$.

Maticí hraničního operátoru ∂_k rozumíme matici \mathbf{A}_k velikosti $m_{k-1} \times m_k$ nad \mathbb{Z} , ve které jsou prvky určeny pomocí pravidla

$$a_{i,j} = \begin{cases} 1 & \sigma_i^{k-1} \in \partial_k(\sigma_j^k), \\ -1 & -\sigma_i^{k-1} \in \partial_k(\sigma_j^k), \\ 0 & \text{jinak.} \end{cases}$$

Velikost matice \mathbf{A}_k tedy odpovídá počtu simplexů v bázích C_k a C_{k-1} , což je počet k -simplexů a $(k-1)$ -simplexů v simplicialním komplexu K . Každý sloupec j pak odpovídá k -simplexu σ_j^k a každý řádek i odpovídá $(k-1)$ -simplexu σ_i^{k-1} . Ve sloupci j pak bude na řádku i hodnota 1, popřípadě -1, jen tehdy, pokud je $(k-1)$ -simplex σ_i^{k-1} , popřípadě $-\sigma_i^{k-1}$, součástí okraje k -simplexu σ_j^k .

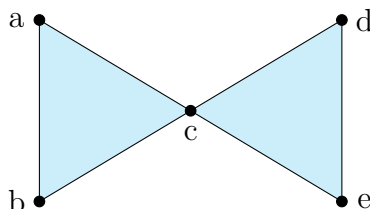
Každý sloupec je tedy $(k-1)$ -řetězec a z matice jej přečteme jednoduše tak, že se podíváme, jaké koeficienty jsou na řádcích u jednotlivých $(k-1)$ -simplexů. Libovolný k -řetězec c pak můžeme vyjádřit jako vektor nad \mathbb{Z} , ve kterém hodnota prvku a_j odpovídá koeficientu bázového simplexu σ_j^k v tomto řetězci c . Označme tento vektor u . Vynásobením vektoru u maticí \mathbf{A}_k zleva získáme vektor, který odpovídá obrazu hraničního operátoru ∂_k . Provedeme tedy

$$v = \mathbf{A}_k \cdot u$$

a ve výsledném vektoru v pak hodnota prvku a_i odpovídá koeficientu bázového simplexu σ_i^{k-1} v $(k-1)$ -řetězci $\partial_k(c)$.

PŘÍKLAD 6.2

Ukažme si konstrukci matice hraničního operátoru z předchozí definice na konkrétním příkladu. Uvažujme následující simplicialní komplex:



Tento simplicialní komplex má tedy šest vrcholů, šest 1-simplexů a dva 2-simplexy. V následující tabulce jsou uvedeny jednotlivé báze příslušných modulů.

Modul	Báze modulu
C_0	a, b, c, d, e
C_1	ab, ac, bc, cd, ce, de
C_2	abc, cde

Pro sestavení matice vezměme stejné značení i uspořádání, v jakém jsou simplexy v tabulce zapsány. Sestavené matice hraničních operátorů jsou uvedeny níže. \mathbf{A}_1 je maticí hraničního operátoru ∂_1 a \mathbf{A}_2 je maticí hraničního operátoru ∂_2 .

$$\mathbf{A}_1 = \begin{matrix} & ab & ac & bc & cd & ce & de \\ \begin{matrix} a \\ b \\ c \\ d \\ e \end{matrix} & \begin{pmatrix} -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{matrix}$$

$$\mathbf{A}_2 = \begin{matrix} & abc & cde \\ \begin{matrix} ab \\ ac \\ bc \\ cd \\ ce \\ de \end{matrix} & \begin{pmatrix} 1 & 0 \\ -1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & -1 \\ 0 & 1 \end{pmatrix} \end{matrix}$$

Nyní se podíváme na Smithův normální tvar. Jak již bylo řečeno v úvodu, celý algoritmus pro nás není podstatný. Důležité jsou pro nás operace, které tento algoritmus provádí a jakým způsobem mění jednotlivé báze.

Algoritmus tedy provádí následující operace na řádcích:

1. výměna řádku i za řádek j ,
2. vynásobení řádku i číslem -1 ,
3. nahrazení řádku i za (řádek i) + q (řádek j), kde $q \in \mathbb{Z}$ a $j \neq i$.

A dále provádí tyto operace na sloupcích:

1. výměna sloupce i za sloupec j ,
2. vynásobení sloupce i číslem -1 ,
3. nahrazení sloupce i za (sloupec i) + q (sloupec j), kde $q \in \mathbb{Z}$ a $j \neq i$.

Nyní se podívejme, co jednotlivé operace dělají s bázi modulů C_k a C_{k-1} v matici hraničního operátoru \mathbf{A}_k . Každá sloupcová operace změní určitým způsobem bázi modulu C_k a každá řádková operace změní určitým způsobem bázi C_{k-1} .

Nechť tedy posloupnost $e_1^{k-1}, \dots, e_{m_{k-1}}^{k-1}$ je bázi C_{k-1} . Na začátku algoritmu se tato posloupnost bude shodovat s posloupností $\sigma_1^{k-1}, \dots, \sigma_{m_{k-1}}^{k-1}$. Řádkové operace změní bázi následovně:

1. výměna řádku i za řádek j : prohození prvků báze e_i^{k-1} za e_j^{k-1} ,
2. vynásobení řádku i číslem -1 : vynásobení prvku báze e_i^{k-1} číslem -1 ,
3. nahrazení řádku i za (řádek i) + q (řádek j), kde $q \in \mathbb{Z}$ a $j \neq i$: nahrazení bázového prvku e_j^{k-1} za $e_j^{k-1} - qe_i^{k-1}$.

Nechť posloupnost $e_1^k, \dots, e_{m_k}^k$ je bázi modulu C_k . Na začátku algoritmu se tato posloupnost bude shodovat s posloupností $\sigma_1^k, \dots, \sigma_{m_k}^k$. Sloupcové operace změní bázi C_k následovně:

1. výměna sloupce i za sloupec j : prohození prvků báze e_i^k za e_j^k ,
2. vynásobení sloupce i číslem -1 : vynásobení prvku báze e_i^k číslem -1 ,
3. nahrazení sloupce i za (sloupec i) + q (sloupec j), kde $q \in \mathbb{Z}$ a $j \neq i$: nahrazení bázového prvku e_i^k za $e_i^k + qe_j^k$.

Pomocí uvedených operací tedy algoritmus matici redukuje do Smithova normálního tvaru:

Definice 6.3 (Smithův normální tvar)

Řekneme, že matice \mathbf{A} nad oborem hlavních ideálů je ve *Smithově normálním tvaru*, jestliže:

$$\mathbf{A} = \begin{pmatrix} q_1 & 0 & \dots & 0 \\ 0 & q_2 & & \\ & & \ddots & \\ \vdots & & & q_{l_k} & \vdots \\ & & & & \ddots & \\ 0 & & & & & 0 & 0 \\ & & \dots & & & 0 & 0 \end{pmatrix}$$

a platí $q_i | q_{i+1}$, kde $i \in \{1, \dots, l_k\}$. Čísla q_i nazýváme *invariantní faktory*.

Pokud redukuje všechny matice hraničních operátorů ∂_k do Smithova normálního tvaru, získáme tím úplný popis struktury modulů H_k .

Nechť tedy pro každé $k \geq 1$ značí $\tilde{\mathbf{A}}_k$ matici $m_{k-1} \times m_k$ ve Smithově normálním tvaru hraničního operátoru ∂_k a dále, necht $\{q_1, \dots, q_{l_k}\}$ jsou její invariantní faktory. Pak platí následující:

- (i) Torzní koeficienty ve struktuře modulu H_{k-1} jsou právě ty invariantní faktory q_i v matici \mathbf{A}_k , které jsou větší než 1.
- (ii) Množina $\{q_i e_i^{k-1} \mid 1 \leq i \leq l_k\}$ je bázi modulu B_{k-1} . Platí tedy:

$$\text{rank}(B_k) = \text{rank}(\mathbf{A}_{k+1}) = l_{k+1}$$

- (iii) Množina $\{e_i^k \mid l_k + 1 \leq i \leq m_k\}$ je bázi modulu Z_k . Platí tedy:

$$\text{rank}(Z_k) = m_k - l_k$$

Kombinací faktů (ii) a (iii) tedy můžeme dosadit do rovnice 4.2 a dostáváme úplný vzorec pro Bettiho čísla:

$$\beta_k = \text{rank}(Z_k) - \text{rank}(B_k) = m_k - l_k - l_{k+1}. \quad (6.1)$$

PŘÍKLAD 6.4

Pokračujme v předchozím příkladu. Níže jsou hraniční matice operátorů ∂_1 a ∂_2 redukované do Smithova normálního tvaru:

$$\tilde{\mathbf{A}}_1 = \begin{matrix} & \begin{matrix} ab & bc & cd & de & z_1 & z_2 \end{matrix} \\ \begin{matrix} b-a \\ c-b \\ d-c \\ e-d \\ a \end{matrix} & \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

kde $z_1 = ab - ac + bc$ a $z_2 = cd - ce + de$

$$\tilde{\mathbf{A}}_2 = \begin{matrix} & \begin{matrix} abc & cde \end{matrix} \\ \begin{matrix} b_1 \\ b_2 \\ ab \\ bc \\ cd \\ de \end{matrix} & \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \end{matrix}$$

kde $b_1 = ab - ac + bc$ a $b_2 = cd - ce + de$

Podle rovnice 6.1 tedy můžeme určit první Bettiho číslo:

$$\beta_1 = \text{rank}(Z_1) - \text{rank}(B_1) = m_1 - l_1 - l_2 = 6 - 4 - 2 = 0.$$

Pro určení ostatní Bettiho čísel si stačí uvědomit, že $\text{rank}(Z_0)$ se rovná počtu vrcholů simplicialního komplexu K a $\text{rank}(B_k) = 0$ pro všechna $k > \dim(K)$. Dostáváme tedy:

$$\beta_0 = \text{rank}(Z_0) - \text{rank}(B_0) = 5 - 4 = 1,$$

$$\beta_2 = \text{rank}(Z_2) - \text{rank}(B_2) = 0 - 0 = 0,$$

a pro $k > \dim(K)$ triviálně:

$$\beta_k = 0.$$

Jelikož všechny invariantní faktory se rovnají jedné, tento simplicialní komplex žádné torze neobsahuje.

Nyní si ukážeme algoritmus pro výpočet homologií, pokud použijeme pro výpočet koeficienty v \mathbb{Z}_2 .

To, co potřebujeme zjistit pro určení struktury modulů H_k , jsou hodnoty invariantních faktorů v redukovaných maticích hraničních operátorů. Pokud ale pracujeme v \mathbb{Z}_2 , modul H_k žádné torze neobsahuje. Stačí pak jednoduše použít jakýkoliv algoritmus na zjištění hodnosti matice.

Připomeňme si následující pojem:

Definice 6.5 (Sloupcově normovaný tvar)

Řekneme, že matice \mathbf{A} je *ve sloupcově normovaném tvaru*, jestliže:

$$\mathbf{A} = \begin{pmatrix} q_1 & 0 & \cdots & & \cdots & 0 \\ * & q_2 & 0 & \cdots & & 0 \\ * & * & \ddots & & & \\ \vdots & \vdots & & q_k & 0 & \vdots \\ * & * & \cdots & * & \ddots & \\ * & * & & \vdots & & 0 & 0 \\ * & * & \cdots & * & \cdots & 0 & 0 \end{pmatrix}$$

kde symbol $*$ značí libovolné číslo.

Čili nenulové prvky mohou ležet pouze na diagonále a nebo pod diagonálou matice. Prvnímu nenulovému prvku v každém sloupci říkáme *pivot*. Následující věta je pro výpočet homologií užitečná:

Věta 6.6 (Hodnost matice ve sloupcově normovaném tvaru)

Hodnost matice \mathbf{A} odpovídá počtu pivotů ve sloupcově normovaném tvaru matice.

Stejně tak pro nás není důležité, aby sloupce v matici byly uspořádány podle pivotů sestupně za sebou tak, jak jsou uvedeny v definici. Výsledný algoritmus pro získání hodnoty matice je uveden níže.

```
1 procedure reduceMatrix(A: m x n)
2   for j := 1 to n do
3     if pivot(j) != null:
4       for k := j + 1 to n do:
5         if pivot(k) = pivot(j):
6           column(k) += column(j)
```

Algoritmus 6.1: Redukce matice hraničního operátoru do sloupcově normovaného tvaru s koeficienty v \mathbb{Z}_2 .

Algoritmus můžeme upravit tak, aby pracoval i s jednotlivými bázemi. Zjistíme tak i konkrétní cykly, které pak generují homologické třídy v H_k .

Pokud tedy oželíme nalezení torzí, můžeme tento algoritmus použít pro výpočet homologií v libovolném simplicialním komplexu.

6.2 Algoritmus persistentní homologie

Algoritmus pro výpočet persistentní homologie vychází ze standardního redukčního algoritmu, který byl uvedený v předchozí kapitole. Rozdíl je v tom, že výpočet probíhá nad $F[t]$ -modulem. Díky některým vlastnostem $F[t]$ -modulů, které si uvedeme, ho však lze zjednodušit. Za F můžeme zvolit jakékoliv pole.

Připomeňme si pojmy, se kterými budeme pracovat:

- \mathcal{C}_k je persistentní modul k -řetězců s koeficienty v $F[t]$,
- $\mathcal{B}_k = \text{im}(\widehat{\partial}_{k+1})$ je gradovaný modul řetězců, které jsou okrajem nějakého řetězce z \mathcal{C}_{k+1} ,
- $\mathcal{Z}_k = \ker(\widehat{\partial}_k)$ je gradovaný modul řetězců s nulovým okrajem,
- $\mathcal{H}_k = \mathcal{Z}_k/\mathcal{B}_k$ je k -tá persistentní homologie simplicialního komplexu K ,
- $\deg(\sigma)$ je stupeň simplexu $\sigma \in K$.

Podobně jako v případě homologie, i nyní reprezentujeme persistentní hraniční operátor pomocí matice. Z definice 3.34 homomorfismu gradovaných modulů pak plyne, že daný homomorfismus musí zachovávat stupeň v gradování. To se jednoduše splní pomocí stupně polynomu u daného prvku v matici.

Definice 6.7 (Matice persistentního hraničního operátoru)

Nechť je dán filtrovaný simplicialní komplex K , hraniční operátor $\widehat{\partial}_k : \mathcal{C}_k \rightarrow \mathcal{C}_{k-1}$ a dále necht m_k značí počet k -simplexů v K a m_{k-1} počet $(k-1)$ -simplexů v K . Uspořádejme k -simplexy do posloupnosti $\sigma_1^k, \dots, \sigma_{m_k}^k$ a stejně tak uspořádejme $(k-1)$ -simplexy do posloupnosti $\sigma_1^{k-1}, \dots, \sigma_{m_{k-1}}^{k-1}$.

Maticí persistentního hraničního operátoru $\widehat{\partial}_k$ rozumíme matici \mathbf{A}_k nad $F[t]$ velikosti $m_{k-1} \times m_k$, ve které jsou prvky určeny pomocí pravidla

$$a_{i,j} = \begin{cases} t^{n_{i,j}} & \sigma_i^{k-1} \in \widehat{\partial}_k(\sigma_j^k), \\ -t^{n_{i,j}} & -\sigma_i^{k-1} \in \widehat{\partial}_k(\sigma_j^k), \\ 0 & \text{jinak,} \end{cases}$$

kde $n_{i,j} = \deg(\sigma_j^k) - \deg(\sigma_i^{k-1})$.

Pomocí stejné sady operací, jako v případě homologií, lze redukovat matici hraničního operátoru do Smithova normálního tvaru.

Připomeňme si, že operace, pomocí kterých redukuje matici hraničního operátoru do Smithova normálního tvaru také mění báze modulů. Z minulé kapitoly tedy víme, že z této redukované matice $\tilde{\mathbf{A}}_k$ můžeme vyčíst bázi modulu \mathcal{Z}_k . Tuto bázi tvoří množina $\{e_i^k \mid l_k + 1 \leq i \leq m_k\}$ a její prvky poznáme v redukované matici $\tilde{\mathbf{A}}_k$ tak, že všechny sloupce, které těmito bázevým prvky odpovídají, jsou nulové. Pomocí řádkových operací můžeme pak upravit matici $\tilde{\mathbf{A}}_{k+1}$ tak, aby byla vyjádřena vzhledem k bázi \mathcal{Z}_k .

Nechť je tedy $\tilde{\mathbf{A}}_{k+1}$ matice persistentního hraničního operátoru $\widehat{\partial}_{k+1}$ redukována do Smithova normálního tvaru a vyjádřena vzhledem k bázi \mathcal{Z}_k , nechť posloupnost $e_{l_k}^k, \dots, e_{m_k}^k$ je bázi \mathcal{Z}_k a vymažeme z této matice řádky, které neodpovídají bázi \mathcal{Z}_k . Z této upravené matice určíme posloupnost \mathcal{P} -intervalů následovně:

- (i) Pokud je řádek i nulový, pak přidáme do posloupnosti \mathcal{P} -interval $(\deg(e_i^k), +\infty)$.
- (ii) Pokud není řádek i nulový, pak obsahuje právě jednoho pivota t^n a do posloupnosti přidáme \mathcal{P} -interval $(\deg(e_i^k), \deg(e_i^k) + n)$.

Tato posloupnost \mathcal{P} -intervalů určuje strukturu \mathcal{H}_k . Získáme tedy informace o persistenci jednotlivých homologických tříd ve filtraci simplicialního komplexu K v dimenzi k .

Přímočarý postup je tedy takový, že získáme bázi \mathcal{Z}_k z matice \mathbf{A}_k , poté upravíme matici \mathbf{A}_{k+1} tak, aby její řádky odpovídaly této bázi a poté matici \mathbf{A}_{k+1} převedeme do Smithova normálního tvaru. Nakonec z této matice pomocí výše uvedených pravidel určíme posloupnost \mathcal{P} -intervalů.

PŘÍKLAD 6.8

Ukážeme si uvedený postup na příkladu. Uvažujme filtraci simplicialního komplexu K uvedenou v tabulce 6.1. Máme tedy filtraci simplicialního komplexu K :

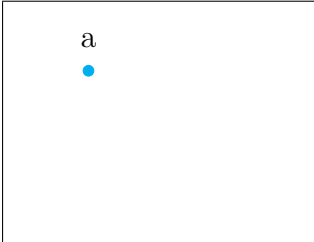
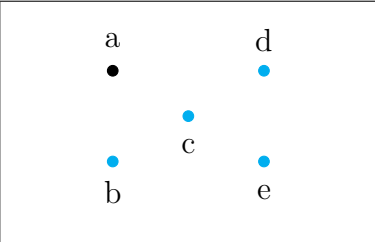
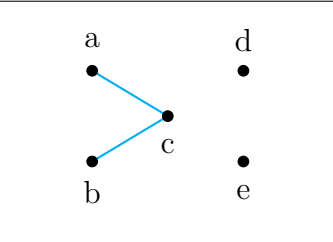
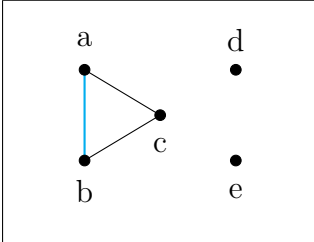
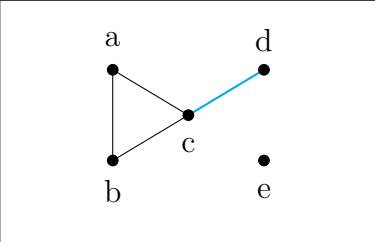
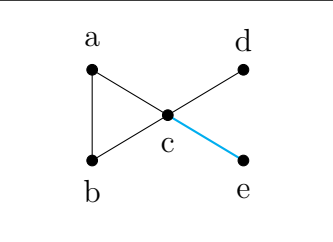
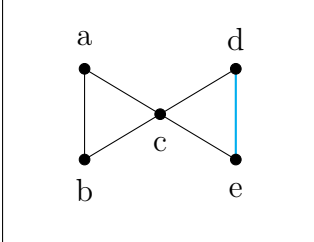
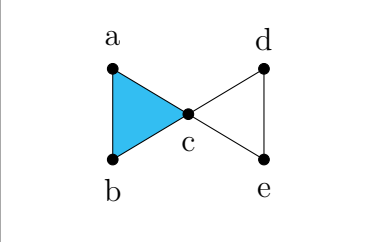
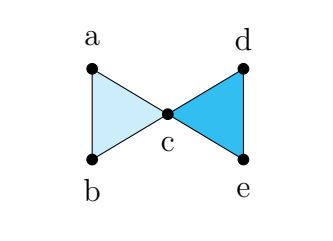
$$\emptyset = K^0 \leq K^1 \leq K^2 \leq K^3 \leq K^4 \leq K^5 \leq K^6 \leq K^7 \leq K^8 \leq K^9 = K$$

Sestavíme nyní persistentní moduly k -řetězců a pro jednoduchost použijeme koeficienty v $\mathbb{Z}_2[t]$. Jak bylo uvedeno v kapitole 4.4, jako stupně jednotlivých simplexů vezmeme kroky, ve kterých byly dané simplex přidány. Tyto stupně jsou uvedeny v tabulce 6.2.

Podle definice matice persistentního hraničního operátoru sestavíme matice \mathbf{A}_1 a \mathbf{A}_2 persistentních hraničních operátorů $\widehat{\partial}_1$ a $\widehat{\partial}_2$:

$$\mathbf{A}_1 = \begin{matrix} & \begin{matrix} ab & ac & bc & cd & ce & de \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \\ e \end{matrix} & \begin{pmatrix} t^3 & t^2 & 0 & 0 & 0 & 0 \\ t^2 & 0 & t & 0 & 0 & 0 \\ 0 & t & t & t^3 & t^4 & 0 \\ 0 & 0 & 0 & t^3 & 0 & t^5 \\ 0 & 0 & 0 & 0 & t^4 & t^5 \end{pmatrix} \end{matrix}$$

$$\mathbf{A}_2 = \begin{matrix} & \begin{matrix} abc & cde \end{matrix} \\ \begin{matrix} ac \\ bc \\ ab \\ cd \\ ce \\ de \end{matrix} & \begin{pmatrix} t^5 & 0 \\ t^5 & 0 \\ t^4 & 0 \\ 0 & t^4 \\ 0 & t^3 \\ 0 & t^2 \end{pmatrix} \end{matrix}$$

					
1	a	2	b, c, d, e	3	ac, bc
					
4	ab	5	cd	6	ce
					
7	de	8	abc	9	cde

Tabulka 6.1: Grafické znázornění filtrace simplicialního komplexu. Nově přidané simplex jsou zvýrazněny modrou barvou, kroky přidání a označení nově přidaného simplexu jsou napsány v řádcích pod obrázky simplicialních komplexů.

Simplex	a	b	c	d	e	ac	bc	ab	cd	ce	de	abc	cde
Stupeň	1	2	2	2	2	3	3	4	5	6	7	8	9

Tabulka 6.2: Stupně jednotlivých simplexů ve filtraci simplicialního komplexu z tabulky 6.1.

a redukuje tyto matice do Smithova normálního tvaru:

$$\tilde{\mathbf{A}}_1 = \begin{matrix} & ac & bc & cd & ce & z_1 & z_2 \\ \begin{matrix} c-ta \\ b-c \\ d-c \\ e-d \\ a \end{matrix} & \begin{pmatrix} t & 0 & 0 & 0 & 0 & 0 \\ 0 & t & 0 & 0 & 0 & 0 \\ 0 & 0 & t^3 & 0 & 0 & 0 \\ 0 & 0 & 0 & t^4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{matrix}$$

$$\text{kde } z_1 = ab + t \cdot ac + t \cdot bc \text{ a } z_2 = cd + t \cdot ce + t^2 \cdot de$$

$$\tilde{\mathbf{A}}_2 = \begin{matrix} & abc & cde \\ z_1 & \left(\begin{array}{cc} t^4 & 0 \\ 0 & t^2 \end{array} \right) \\ z_2 & \\ ac & \\ bc & \\ cd & \\ ce & \end{matrix}$$

Z redukované matice $\tilde{\mathbf{A}}_2$ vymažeme řádky, které neodpovídají bázi \mathcal{Z}_1 :

$$\tilde{\mathbf{A}}'_2 = \begin{matrix} & abc & cde \\ z_1 & \left(\begin{array}{cc} t^4 & 0 \\ 0 & t^2 \end{array} \right) \\ z_2 & \end{matrix}$$

Z této matice tedy získáváme dva \mathcal{P} -intervaly: (4, 8) a (7, 9), což intuitivně odpovídá průběhu filtrace.

1-cyklus z_1 vznikl v kroku filtrace 4 přidáním 1-simplexu ab . V simplicialní komplexu K^4 tedy vznikla jemu příslušná homologická třída. Přidáním 2-simplexu abc se z tohoto cyklu stal okraj, čímž daná homologická třída zanikla. Tato homologická třída tedy existovala v simplicialních komplexech K^4 , K^5 , K^6 a K^7 . Situace u cyklu 1-cyklu z_2 je obdobná.

Nyní si uvedeme dvě užitečná tvrzení, díky kterým lze zjednodušit standardní redukční algoritmus:

Věta 6.9 (\mathcal{P} -intervaly ve sloupcově normovaném tvaru)

Nechť $\sigma_1^k, \dots, \sigma_{m_k}^k$ je bázi \mathcal{C}_{k-1} , necht $\sigma_1^{k-1}, \dots, \sigma_{m_{k-1}}^{k-1}$ je bázi \mathcal{C}_{k-1} a necht \mathbf{A}_k je maticí persistentního hraničního operátoru $\hat{\partial}_k : \mathcal{C}_k \rightarrow \mathcal{C}_{k-1}$. Pokud řádky v matici \mathbf{A}_k setřídíme sestupně podle stupně jednotlivých simplexů v bázi \mathcal{C}_{k-1} , pak stačí matici převést pouze do sloupcově normovaného tvaru a získáme stejné \mathcal{P} -intervaly, jako ze Smithova normálního tvaru matice \mathbf{A}_k .

Tato věta tedy říká, že pokud zachováme požadované pořadí simplexů, pak matici \mathbf{A}_k není nutné redukovat až do Smithova normálního tvaru, ale stačí pouze sloupcově normovaný tvar.

Na začátku kapitoly jsme si uvedli, že \mathcal{P} -intervaly lze z redukované matice získat tak, že ji vyjádříme vzhledem k bázi \mathcal{Z}_k . To tedy znamená, že bychom měli matici \mathbf{A}_{k+1} pomocí řádkových operací upravit tak, aby řádky této bázi odpovídaly. Nicméně, bázi \mathcal{Z}_k získáme již z matice \mathbf{A}_k , jelikož ji pomocí sloupcových operací musíme převést do sloupcově normovaného tvaru, abychom z ní získali příslušné \mathcal{P} -intervaly.

Následující věta říká, jak lze matici \mathbf{A}_{k+1} vyjádřit vzhledem k bázi \mathcal{Z}_k , pokud již známe bázi \mathcal{Z}_k z redukce matice \mathbf{A}_k .

Věta 6.10 (Vyjádření matice persistentního hraničního operátoru vzhledem k bázi \mathcal{Z}_k)

Bud' posloupnost $\sigma_1^k, \dots, \sigma_{m_k}^k$ bázi modulu \mathcal{C}_k seřazená sestupně podle stupně jednotlivých simplexů. Necht' \mathbf{A}_k je matice persistentního hraničního operátoru $\hat{\partial}_k : \mathcal{C}_k \rightarrow \mathcal{C}_{k-1}$, necht' \mathbf{A}_{k+1} je matice persistentního hraničního operátoru $\hat{\partial}_{k+1} : \mathcal{C}_{k+1} \rightarrow \mathcal{C}_k$ a necht' sloupce matice \mathbf{A}_k a řádky matice \mathbf{A}_{k+1} odpovídají uvedné bázi modulu \mathcal{C}_k . Pokud je během redukce matice \mathbf{A}_k do sloupcově normovaného tvaru použita operace výměny sloupců, necht' je provedena výměna odpovídajících řádků v matici \mathbf{A}_{k+1} . Jakmile je matice \mathbf{A}_k redukována do sloupcově normovaného tvaru, stačí z matice \mathbf{A}_{k+1} smazat ty řádky, které neodpovídají získané bázi \mathcal{Z}_k a bázevé prvky u zbylých řádků nahradit prvky báze \mathcal{Z}_k .

S pomocí těchto dvou tvrzení si popíšeme algoritmus, který výpočet nad okruhem polynomů $F[t]$ pouze simuluje a ke spočtení \mathcal{P} -intervalů si vystačí pouze s využitím koeficientů v poli F .

Mějme tedy filtrovaný simplicialní komplex K s počtem simplexů m . Nejprve uspořádejme všechny simplexu v K do posloupnosti $\sigma_0, \dots, \sigma_{m-1}$ podle dimenze a kroku přidání. Pro takto uspořádané simplexu tedy platí, pokud $\sigma_i < \sigma_j$, pak $\dim(\sigma_i) \leq \dim(\sigma_j)$ a $\deg(\sigma_i) \leq \deg(\sigma_j)$, kde $i, j \in \{0, \dots, m-1\}$.

Dále, místo konstrukce jednotlivých matic \mathbf{A}_k pro $k \in \{1, \dots, \dim(K)\}$, sestavíme jednu velkou matici a použijeme řídkou reprezentaci matic. Víme, že sloupce v matici persistentního hraničního operátoru reprezentují k -řetězce. Pokud se v matici \mathbf{A}_k ve sloupci j na řádku i nachází nenulový prvek $a_{i,j}$, pak to znamená, že řetězec $\hat{\partial}_k(\sigma_j)$ obsahuje prvek $a_{i,j}\sigma_i$. Můžeme tedy ke každému simplexu σ_j uložit přímo daný k -řetězec a nemusíme tyto koeficienty ukládat do matice. Zavedme si tedy pole `chain` délky m . Na začátku algoritmu bude toto pole prázdné a během výpočtu do tohoto pole budeme postupně ukládat jednotlivé k -řetězce s koeficienty v poli F . Na konci algoritmu pak budou tyto řetězce odpovídat sloupcům redukovaných matic $\tilde{\mathbf{A}}_k$.

Dále je potřeba znát, které řetězce odpovídají okrajům bázevéch prvků modulu \mathcal{Z}_k . Pokud totiž nějaký $(k-1)$ -řetězec neodpovídá okraji nějakého bázevého prvku \mathcal{Z}_k , pak z věty 6.10 víme, že v matici \mathbf{A}_{k+1} stačí jednoduše smazat řádky, které neodpovídají této bázi. Pro tento účel zavedeme pole `mark` délky m , které bude obsahovat pouze hodnoty `true` a `false`. Pokud `mark[j] = true`, pak to značí, že řetězec `chain[j]` je součástí okraje nějakého bázevého prvku modulu \mathcal{Z}_k a jemu odpovídající řádek v matici \mathbf{A}_{k+1} nesmí být smazán. Vzhledem k tomu, že používáme řídkou reprezentaci matic, tak to znamená, že při uplatňování věty 6.10 daný simplex σ_j nesmí být smazán v jakémkoliv řetězci, ve kterém se během výpočtu vyskytne. Na začátku budou všechny hodnoty v poli `mark` nastaveny na `false`.

Nakonec ještě zavedeme pole `destroyer`, také délky m a pole `intervals` délky $\dim(K)$. Do pole `destroyer` budeme značit indexy simplexů, jejichž přidáním zanikla určitá homologická třída. Přesněji, pokud je v poli `destroyer` na pozici i číslo j , pak to znamená, že daná homologická třída vznikla přidáním simplexu σ_i a zanikla přidáním simplexu σ_j . Na začátku algoritmu bude toto pole prázdné. Pole `intervals` budeme využívat k uložení jednotlivých \mathcal{P} -intervalů, kde na indexu i jsou \mathcal{P} -intervaly popisující strukturu modulu \mathcal{H}_i .

Algoritmus pak prochází jednotlivé simplexly podle určeného uspořádání od nejmenšího po největší a pro každý simplex simuluje redukci matice.

Předpokládejme tedy, že se nacházíme v kroku j . Nejprve je nutné získat příslušný $(k-1)$ -řetězec odpovídající okraji k -simplexu σ_j . K tomu použijeme hraniční operátor ∂_k . Označme tedy $d = \partial_k(\sigma_j)$. Z tohoto $(k-1)$ -řetězce pak dle věty 6.10 smažeme všechny takové simplexly, které neodpovídají bázi \mathcal{Z}_{k-1} . To jsou tedy takové simplexly, pro které platí `mark[i] = false`.

Nyní je nutné řetězec d redukovat. Redukce řetězce d simuluje redukci sloupce, který v matici \mathbf{A}_k odpovídá simplexu σ_j . Z věty 6.9 víme, že abychom se vyhnuli použití řádkových operací, je nutné simplexly redukovat v sestupném pořadí podle jejich stupňů. Nejvyšší stupeň v řetězci d má vždy ten simplex, který má nejvyšší index v námi, na začátku zvoleném, uspořádání simplexů. Nechť je tedy σ_i tímto simplexem. $(k-1)$ -řetězec odpovídající tomuto simplexu se nachází v poli `chain` na pozici i . Pokud řetězec `chain[i]` není nulový, pak jej použijeme k redukování řetězce d . Nechť q je tedy koeficient $(k-1)$ -simplexu σ_i v řetězci d a nechť q^{-1} značí inverzní prvek k prvku q v poli F . K řetězci d přičteme q^{-1} násobek řetězce `chain[i]`.

Získáme tedy vždy $(k-1)$ -řetězec `chain[i]` odpovídající simplexu s nejvyšším indexem a řetězec d budeme výše uvedeným způsobem redukovat tak dlouho, dokud nebude splněna jedna z následujících podmínek: buďto je řetězec d nulový, anebo je nulový získaný řetězec `chain[i]`.

Pokud je výsledný $(k-1)$ -řetězec d nulový, pak se jedná o řetězec, který odpovídá okraji nějakého bázového prvku \mathcal{Z}_k . V tomto případě tedy přidáním k -simplexu σ_j vznikl nějaký k -cyklus z , jehož okrajem je nulový řetězec d . Cyklus z tedy generuje jednu homologickou třídu v \mathcal{H}_k a tuto skutečnost označíme nastavením `mark[j] = true`.

Pokud řetězec d není nulový, pak se jedná o část $(k-1)$ -řetězce, který odpovídá okraji nějakého k -řetězce z \mathcal{C}_k . Řetězec d netvoří celý okraj, protože dle věty 6.10 některé simplexly promazáváme. V tomto případě jsme tedy zjistili, že přidáním k -simplexu σ_j zanikla nějaká homologická třída a to v kroku $\deg(\sigma_j)$. Z věty 6.9 víme, že pivot by se v matici \mathbf{A}_k nacházel na řádce s nejvyšším stupněm. V této řídce reprezentované matici tomuto pivotu odpovídá prvek v řetězci d , jehož simplex má nejvyšší index. Nechť je tedy tímto simplexem simplex σ_i . Získáváme \mathcal{P} -interval $(\deg(\sigma_i), \deg(\sigma_j))$ a tento interval přidáme k intervalům v `intervals[k-1]`. Výsledný řetězec d uložíme do pole `chain` na pozici i a nastavíme `destroyer[i] = j`.

Na závěr je nutné ještě jednou projít všechny simplexu a získat ty intervaly, jejichž homologické třídy vznikly přidáním simplexu σ_j , ale nikdy nezanikly. Pokud tedy pro nějaké j platí $\text{mark}[j] = j$ a řetězec $\text{chain}[j]$ je prázdný, pak to znamená, že přidáním simplexu σ_j vznikl nějaký k -cyklus z , který nikdy nezanikl. Získáváme tedy \mathcal{P} -interval $(\deg(\sigma_j), +\infty)$ a tento interval přidáme k intervalům v $\text{intervals}[k]$.

Výsledný algoritmus je popsán pseudokódem v algoritmu 6.3. K redukci řetězců využívá proceduru `reduceChain`, která je popsána pseudokódem v algoritmu 6.2.

```

1 procedure reduceChain( $\sigma$ , mark, chain):
2   // initialize
3    $k := \dim(\sigma)$ 
4    $d := \partial_k(\sigma)$ 
5
6   // remove unmarked simplicies
7   for each  $\sigma_i$  in  $d$  do:
8     if not mark[i]:
9       remove  $\sigma_i$  from  $d$ 
10
11  // reduce chain
12  while  $d \neq \emptyset$  do:
13     $i := \text{maxIndex}(d)$ 
14
15    if chain[i] =  $\emptyset$ :
16      break
17    else:
18       $q := \text{coefficient of } \sigma_i \text{ in } \text{chain}[i]$ 
19       $d := d - q^{-1} * \text{chain}[i]$ 
20
21  // finish
22  return  $d$ 

```

Algoritmus 6.2: Pseudokód algoritmu pro redukci řetězce. Algoritmus využívá jednotlivé hraniční operátory negradovaných modulů k -řetězců a při redukci počítá inverzi k prvku q v poli F .

```

1 procedure computeIntervals(K):
2   // initialize
3   for k := 0 to dim(K) do:
4     intervals[k] := empty list
5
6   for j := 0 to m - 1 do:
7     chain[j] :=  $\emptyset$ 
8     mark[j] := false
9     destroyer[j] := null
10
11  // simulate matrix reduction
12  for j := 0 to m - 1 do:
13    d := reduceChain( $\sigma_j$ , mark, chain)
14
15    if d =  $\emptyset$ :
16      mark[j] := true
17    else:
18      k := dim( $\sigma_j$ )
19      i := maxIndex(d)
20      chain[i] := d
21      destroyer[i] := j
22       $\mathcal{P}$  := (deg( $\sigma_i$ ), deg( $\sigma_j$ ))
23      intervals[k-1].append( $\mathcal{P}$ )
24
25  // pair homologous classes with infinite persistence
26  for j := 0 to m - 1 do:
27    if mark[j] and chain[j] =  $\emptyset$ :
28      k := dim( $\sigma_j$ )
29       $\mathcal{P}$  := (deg( $\sigma_j$ ),  $+\infty$ )
30      intervals[k].append( $\mathcal{P}$ )
31
32  // finish
33  return mark, chain, destroyer, intervals

```

Algoritmus 6.3: Pseudokód algoritmu pro výpočet \mathcal{P} -intervalů filtrovaného simplicialního komplexu. K redukci řetězců využívá proceduru `reduceChain` znázorněnou algoritmem 6.2. Kroky algoritmu jsou podrobněji popsány v textu výše.

Na závěr prozkoumáme časovou a paměťovou složitost algoritmu. Nechť m značí počet simplexů ve vstupním filtrovaném simplicialním komplexu.

Nejprve se podívejme na algoritmus na redukci řetězce popsany v 6.2. Řetězec d získaný na řádku 4 je okrajem vstupního k -simplexu σ a může mít tedy nejvýše $k + 1$ neoznačených simplexů. Z toho tedy plyne, že operace na řádku 19 má časovou složitost $k + 1$. V nejhorším případě tato operace sníží celkový počet simplexů v řetězci d o jedna, a tím pádem cyklus na řádcích 12 až 19 bude mít $k + 1$ opakování. V simplicialním komplexu, který obsahuje m simplexů, nebude mít žádný okraj simplexu více jak m prvků. Dostáváme tedy, že časová složitost tohoto algoritmu je v nejhorším případě $\mathcal{O}(m^2)$.

Algoritmus pro výpočet intervalů popsany v 6.3 volá na řádku 13 přesně m krát proceduru na redukci řetězce. Časová složitost celého algoritmu je tedy $\mathcal{O}(m^3)$.

Paměťová složitost celého algoritmu je pak $\mathcal{O}(m^2)$. To plyne z toho, že v nejhorším případě bude potřeba uložit m nenulových řetězců, kde každý bude mít velikost $\mathcal{O}(m)$.

7 Ukázka jednoduchého použití simplicialní homologie

Tato kapitola je věnována praktické ukázce použití simplicialní a persistentní homologie v analýze dat. Obsahuje dva příklady, ve kterých se pokouším analyzovat data pomocí simplicialní a persistentní homologie a také se pokouším interpretovat výsledky. Nicméně, příklady v této kapitole jsou spíše na ukázkou, výsledky by mohly být zajímavé pro odborníka v dané oblasti, který by z nich mohl vyvodit hodnotnější závěry.

Jelikož zde budeme pracovat s daty, je dobré tento pojem formalizovat. Při formalizaci jsem se inspiroval v teorii relačních databází⁶.

Mějme množinu atributů $Y = \{y_1, \dots, y_n\}$ a necht' pro $1 \leq i \leq n$ je množina D_i obor hodnot atributu y_i . Předpokládejme navíc, že pro každou doménu D_i , $1 \leq i \leq n$ máme k dispozici hodnotu $null \in D_i$. Hodnota $null$ symbolizuje neznámou nebo nevyznačenou hodnotu. Dále mějme množinu objektů $X = \{x_1, \dots, x_m\}$, kde ke každému objektu je přiřazena n -tice

$$(d_1, \dots, d_n) \in D_1 \times \dots \times D_n.$$

Říkáme, že objekt x_i má na atributu y_i hodnotu d_i . Danou n -tici také často ztotožňujeme s daným objektem. Množině objektů X s atributy Y a výše definovanými n -ticemi budeme říkat *dataset*. Dataset v příkladech budeme jednoduše značit X .

7.1 Dataset Mall

Na tomto datasetu si ukážeme jednu ze základních možností, jak lze využít persistentní homologii. Postup bude následující:

1. Vytvoříme filtraci datasetu pomocí Ripsovy filtrace a získáme čárové kódy.
2. V čárových kódech najdeme homologické třídy s velkou persistentí.
3. Pokusíme se interpretovat význam děr, které odpovídají těmto homologickým třídám.

Tento postup je v topologické analýze již celkem osvědčený. Nevýhodou tohoto postupu je samozřejmě jeho výpočetní náročnost a pak také fakt, že se nám nemusí vždy podařit interpretovat význam takto získaných děr. Díky tomu je také persistentní homologie takto vytvořeného Ripsova komplexu často zkoumána pouze v malých dimenzích, zpravidla v 0, 1, výjimečně pak v dimenzi 2.

Pro ukázkou jsem použil dataset Mall, který jsem získal z otevřeného repozitáře Kaggle [16]. Tento dataset popisuje zákazníky fiktivního internetového obchodu. Obsahuje pět atributů: identifikátor zákazníka, pohlaví, věk,

⁶Jedná se o pojem *relační schéma*.

příjem a skóre. Atribut pohlaví může nabývat pouze dvou hodnot a to muž, nebo žena. Oborem hodnot zbývajících atributů jsou reálná čísla. Žádný z atributů neobsahuje neznámou hodnotu a dataset celkem obsahuje 200 různých zákazníků.

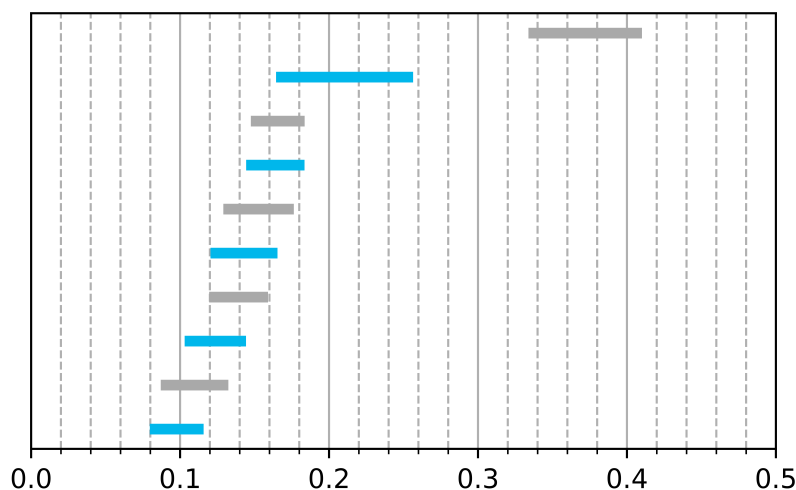
Význam atributů pohlaví, věk a příjem je zřejmý. Hodnota atributu skóre je každému zákazníkovi přiřazena na základě jeho chování v obchodě. Jednoduše řečeno značí, jak moc je pro nás daný zákazník lukrativní: jak často a jaké zboží zákazník nakupuje a jak reaguje na různé akce obchodu.

Atributy pohlaví a identifikátor zákazníka jsem pro analýzu vypustil, zbývajících hodnoty jsem naškálával na interval $[0, 1]$. Dostáváme tedy upravený dataset, který obsahuje pouze tři atributy, přičemž hodnoty těchto atributů jsou reálná čísla. Dataset pak lze zobrazit v trojrozměrném prostoru. Jednotlivá data tedy chápeme jako body tohoto prostoru a pro vytvoření Ripsovy filtrace vezeme klasickou eukleidovskou metriku. Objekty v datasetu tedy uvažujeme jako vektory hodnot a metrika je dána obvyklým předpisem

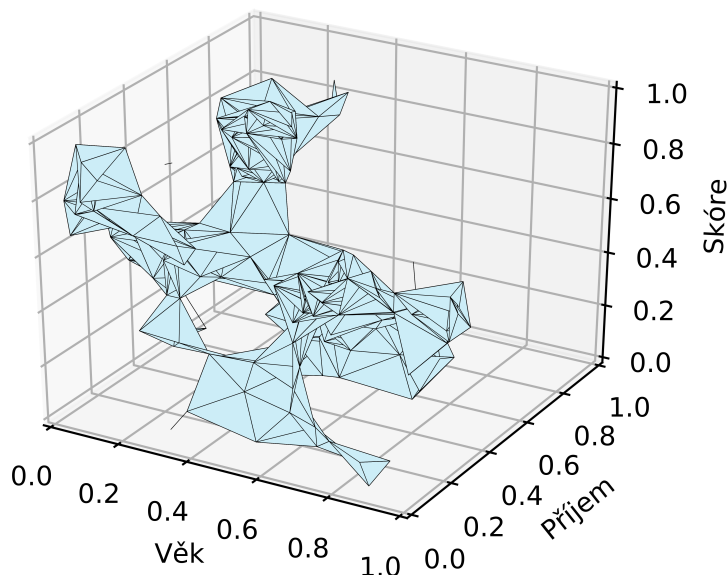
$$m(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2},$$

kde n značí počet atributů a $x, y \in X$.

Vzhledem k tomu, že dataset je trojrozměrný, má smysl zjišťovat persistentní homologii v dimenzi 0, 1 a 2. Dimenze 0 je v tomto případě nejméně zajímavá. Ripsův komplex v 0-té dimenzi totiž vytváří shluky objektů a persistentní homologie popisuje vývoj těchto shluků. V této analýze se tedy zaměříme na dimenze 1 a 2. Čárový kód dimenze 1 je na obrázku 7.1.



Obrázek 7.1: Čárové kódy dataset Mall pro dimenzi 1. Na obrázku je zobrazeno prvních 10 homologických tříd s největší persistencí.



Obrázek 7.2: Ripsův komplex vytvořený z datasetu Mall s hodnotou parametru $\epsilon = 0.2$.

Již na první pohled si lze z obrázku 7.1 všimnout, že zcela největší persiscencí má homologická třída druhá ze shora. Tato homologická třída se ve filtraci vyskytuje pro hodnoty parametru Ripsovy filtrace od 0.167 až do 0.252.

Na obrázku 7.2 je Ripsův komplex vytvořený s hodnotou parametru $\epsilon = 0.2$. Díra reprezentovaná touto nejdéle trvající homologickou třídou se nachází přímo uprostřed.

Nyní je třeba výsledek interpretovat. Na obrázku 7.2 vidíme, že zkoumaná díra se nachází přibližně uprostřed datasetu. Všichni zákazníci se nachází na okraji této díry. Znamená to tedy, že tento obchod nenavštěvují žádní průměrní zákazníci. Dle mého názoru to znamená, že pro tento obchod má smysl soustředit spíše na extrémní - zákazníky s nízkým či vysokým příjmem.

Zajímavá může být také situace ze sociologického hlediska. Můžeme uvažovat, že simplicialní komplex zachycuje vývoj populace. Vidíme, že mladí zákazníci mají skóre celkem vysoké bez ohledu na příjem. Postupně, jak zákazníci stárnou, jejich skóre se vyrovnává a navíc se rozdělují na bohatší a chudší. Dle mého názoru tento jev může být ve společnosti také běžný.

V dimenzi 2 tento dataset obsahuje pouze dvě homologické třídy s velmi krátkou persiscencí 0.015 a 0.0015. Dle mého názoru jsou persiscence těchto dvou homologických tříd tak malé, že je nemá smysl studovat.

7.2 Dataset US Senate Votes

U tohoto příkladu si ukážeme zatím nikde nepublikovaný postup, díky kterému se mi podařilo spočítat Bettiho číslo v celkem vysoké dimenzi.

Nejprve si popíšeme analyzovaný dataset. Tento dataset obsahuje záznamy hlasování senátorů kongresu spojených států u 16 vybraných zákonů z roku 1984. Celkem dataset obsahuje 17 atributů. První atribut je stranická příslušnost senátora, který nabývá buď hodnoty *demokrat*, nebo *republikán*. Zbývajících 16 atributů jsou záznamy, jak který senátor hlasoval u příslušného zákona. Mohou nabývat buď hodnoty *y*, která značí, že senátor byl pro přijetí, nebo hodnoty *n*, která značí, že senátor byl proti přijetí. Chybějící hodnota *null* pak značí, že se senátor daného hlasování nezúčastnil nebo nehlasoval. Dataset jsem získal z otevřeného repozitáře UCI [17].

Z čistě technických důvodů jsem nakonec vybral pouze 8 atributů: stranu a sedm náhodných zákonů.

Simpliciální komplex sestavíme takto: jako vrcholy vezeme všechny objekty v datasetu. Poté vezmeme všechny n -prvkové kombinace atributů a jako simplexy vezmeme takové maximální množiny objektů, které mají na dané n -tici atributů stejnou hodnotu. Dále pak simplicciální komplex doplníme o všechny potřebné stěny. Nakonec spočítáme homologii takto vytvořeného simplicciálního komplexu.

Zajímavě vyšla homologie pro $n = 5$. Při této volbě tedy vytváříme 5-ti prvkové kombinace z 8-mi atributů.

Nenulová Bettiho čísla vyšla $\beta_0 = 1$, $\beta_1 = 20$, $\beta_2 = 62$ a $\beta_3 = 1$, přičemž zajímavé je právě číslo β_3 . Je totiž vcelku vzácné, že u simplicciálního komplexu sestaveného z reálných dat lze Bettiho číslo v takto vysoké dimenzi spočítat a že vyjde nenulové.

Nyní je třeba výsledek interpretovat. Z uvedených Bettiho čísel je zřejmé, že tento simplicciální komplex nepůjde tak jednoduše vykreslit jako v předchozím příkladě. Nicméně můžeme prozkoumat cykly, které tyto homologické třídy generují.

$\beta_3 = 1$ tedy znamená, že simplicciální komplex obsahuje jednu 3-dimenzionální díru. Homologická třída reprezentující tuto díru je tedy generována 3-cyklem. Veškeré vrcholy tohoto 3-cyklu jsou tvořeny jen těmi senátory, kteří jsou v demokratické straně. Tato díra se tedy nachází uvnitř demokratické strany.

$\beta_2 = 62$ znamená, že v simplicciálním komplexu se nachází celkem 62 děr. Z toho 19 jich je uvnitř demokratické strany a 33 uvnitř republikánské. Zbývajících 10 cyklů je tvořeno senátory z obou stran. Takový cyklus vznikne pokaždé, kdykoliv existují alespoň dva senátoři, kteří na dané n -tici atributů zahlasují stejně a přitom jsou oba z jiné strany.

Pro $\beta_2 = 20$ máme 8 cyklů uvnitř demokratické strany a zbývajících 12 je tvořeno senátory z obou stran.

Lze tedy říci, že z topologického hlediska je zajímavější demokratická strana. Obecně je známo, že demokratická strana je stranou liberální a republikáni naopak stranou konzervativní. Konzervatismus bych označil za jednotnější a liberalismus za mnohem méně jednotný. Na základě toho, jakým způsobem byl z těchto dat simplicialní komplex sestaven, si myslím, že takový je i význam získaných homologií.

8 Algoritmus Mapper

Algoritmus *Mapper* byl navrhnut za účelem redukce dimenzionality dat a získávání znalostí z dat. Dimenzí dat se myslí počet atributů a redukce dimenzionality je pak proces, který vhodným způsobem sníží počet atributů. Často je cílem redukce dimenzionality redukovat počet atributů tak, aby pak šlo tato data zobrazit v nějaké vhodné grafické podobě.

Algoritmus Mapper k redukcí dimenzionality využívá topologii: redukuje vysokodimenzionální data do jednodimenzionálních simplicialních komplexů. Jak si ukážeme v následujících dvou podkapitolách, tyto simplicialní komplexy se sestaví tak, aby určitým způsobem odpovídaly *topologii dat*. Jelikož jednodimenzionální simplicialní komplex je ve skutečnosti pouze graf, lze jej pak jednoduše zobrazit a k jeho zobrazení se využívají obvyklé algoritmy k vykreslování grafů. Interpretací těchto grafů lze pak z těchto dat získat nějaké znalosti.

Algoritmus Mapper má dvě verze. První z těchto verzí se jmenuje *Topologický Mapper* a druhá verze se jmenuje *Statistický Mapper*.

Topologický Mapper poskytuje teoretické zázemí algoritmu, využívá některé pojmy z homotopie a lze jej využít ke zkoumání topologických prostorů. Statistický Mapper je z něj odvozen a je určen pro zpracování dat.

8.1 Topologická verze

V této kapitole si ukážeme, jak algoritmus Mapper můžeme aplikovat na topologické prostory. Výstupem topologické verze tohoto algoritmu je speciální abstraktní simplicialní komplex, který se nazývá *Nerv pokrytí*:

Definice 8.1 (Nerv pokrytí)

Nechť X je topologický prostor a $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ jeho libovolné pokrytí. Nerv pokrytí \mathcal{U} je abstraktní simplicialní komplex s množinou vrcholů A , ve kterém množina $\{\alpha_0, \dots, \alpha_k\}$ tvoří k -simplex právě tehdy, když platí $U_{\alpha_0} \cap \dots \cap U_{\alpha_k} \neq \emptyset$. Nerv pokrytí \mathcal{U} značíme $N(\mathcal{U})$.

Topologická verze algoritmu má pak tři vstupní parametry. Prvním parametrem je zkoumaný topologický prostor X , na který chceme Mapper aplikovat. Další dva parametry jsou spojitě zobrazení $f : X \rightarrow Z$ a konečné otevřené pokrytí $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ topologického prostoru Z . Otevřené pokrytí znamená, že prvky tohoto pokrytí jsou pouze otevřené množiny v daném topologickém prostoru. Zobrazení f se nazývá *filtr* a spolu s otevřeným pokrytím \mathcal{U} se jistým způsobem využije právě k sestavení výsledného nervu pokrytí. Celý postup algoritmu je popsán v algoritmu 8.1.

Postup algoritmu *Topologický Mapper*

Vstup:

Topologický prostor X

Filtr $f : X \rightarrow Z$

Konečné otevřené pokrytí $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ topologického prostoru Z

Výstup:

Nerv otevřeného pokrytí prostoru X

Postup:

1A: $f^{-1}(\mathcal{U}) = \{f^{-1}(U_\alpha) \mid \alpha \in A\}$

2A: $\bar{\mathcal{U}} = \emptyset$

2B: Pro každou množinu $V \in f^{-1}(\mathcal{U})$:

2C: Pro každou maximální obloukově souvislou podmnožinu $\bar{V} \subseteq V$:

2D: $\bar{\mathcal{U}} = \bar{\mathcal{U}} \cup \bar{V}$

3A: $N(\bar{\mathcal{U}}) =$ abstraktní simplicialní komplex s množinou vrcholů A

3B: Pro každou podmnožinu $\{\alpha_0, \dots, \alpha_k\} \subseteq A$:

3C: Pokud $U_{\alpha_0} \cap \dots \cap U_{\alpha_k} \neq \emptyset$:

3D: Přidej $\{\alpha_0, \dots, \alpha_k\}$ jako simplex do $N(\bar{\mathcal{U}})$

4A: Vrať $N(\bar{\mathcal{U}})$

Algoritmus 8.1: Topologická verze algoritmu Mapper. Význam parametrů je podrobněji popsán v textu výše.

V krocích 3A až 3D se vytváří nerv pokrytí, který odpovídá definici 8.1. Pro vykreslení nervu se použijí všechny 0 a 1-simplexy. To tedy znamená, že pro simplex vyšší dimenze se vykreslí všechny jeho stěny, které mají dimenzi 0 a 1.

Osvětlíme si kroky 1A a 2A až 2D. V kroku 1A vytváříme pokrytí prostoru X pomocí zobrazení f . Jelikož zobrazení f je spojitě a pokrytí \mathcal{U} otevřené, f indukuje otevřené pokrytí prostoru X . Množina $f^{-1}(\mathcal{U})$ je tedy konečným otevřeným pokrytím prostoru X .

V krocích 2A až 2D pak toto pokrytí navíc rozkládáme na obloukově souvislé množiny. Na následujícím příkladu⁷ si ukážeme, proč je důležité množiny v pokrytí $f^{-1}(\mathcal{U})$ rozkládat na obloukově souvislé množiny.

PŘÍKLAD 8.2

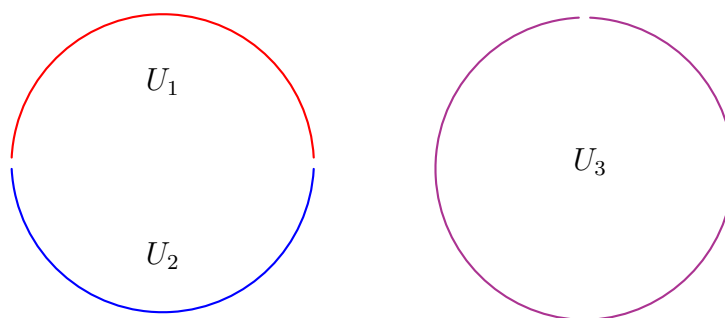
Jakožto zkoumaný topologický prostor X si vezměme jednotkovou kružnici $\mathbb{S}^1 = \{(x, y) \mid x^2 + y^2 = 1\}$. Za prostor Z si vezměme interval $[-1, 1]$ a jako filtr $f : X \rightarrow Z$ vezměme projekci bodu kružnice na jeho y souřadnici: $f(x, y) = y$. Jako pokrytí prostoru Z vezměme množinu $\mathcal{U} = \{[-1, 0), (-1, 1), (0, 1]\}$.

⁷Příklad je zpracován na základě příkladu v [1] na straně 283.

Filtrem indukované pokrytí $f^{-1}(\mathcal{U})$ prostoru X tedy obsahuje tyto tři množiny:

- $U_1 = f^{-1}([-1, 0)) = \{(x, y) \mid y > 0\}$
- $U_2 = f^{-1}((0, 1]) = \{(x, y) \mid y < 0\}$
- $U_3 = f^{-1}((-1, 1)) = \{(x, y) \mid y \neq 0\}$

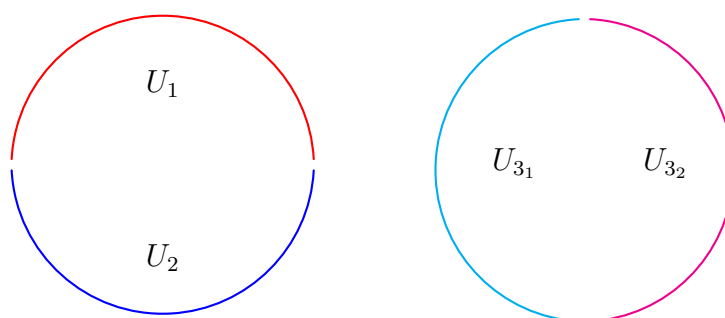
Toto pokrytí je znázorněno na následujícím obrázku:



Provedme úpravu pokrytí $f^{-1}(\mathcal{U})$ na pokrytí tvořené obloukově souvislými množinami. Jediná změna nastává u množiny U_3 , přičemž se tato množina rozdělí na dvě obloukově souvislé množiny:

- $U_{3_1} = \{(x, y) \mid y \neq 0, x < 0\}$
- $U_{3_2} = \{(x, y) \mid y \neq 0, x > 0\}$

Výsledkem úpravy je tedy pokrytí $\bar{\mathcal{U}} = \{U_1, U_2, U_{3_1}, U_{3_2}\}$, které je zobrazeno níže:



Níže jsou pak zobrazeny nervy pokrytí $N(f^{-1}(\mathcal{U}))$ a $N(\overline{\mathcal{U}})$:



Z obrázku je vidět, že nerv $N(\overline{\mathcal{U}})$ je podobný kružnici \mathbb{S}^1 , zatímco nerv $N(f^{-1}(\mathcal{U}))$ spíše připomíná úsečku. Ve skutečnosti, geometrická realizace $N(\overline{\mathcal{U}})$ je homotopicky ekvivalentní s topologickým prostorem \mathbb{S}^1 . To je obecný jev, který je shrnut ve větě 8.3.

Věta 8.3 (Nervová věta)

Bud' X topologický prostor a $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ jeho pokrytí, které je spočetné a otevřené. Dále předpokládejme, že pro všechny neprázdné podmnožiny $S \subseteq A$, je $\bigcap_{s \in S} U_s$ buďto prázdný nebo kontraktibilní. Pak topologický prostor, vytvořený geometrickou realizací $N(\mathcal{U})$ je homotopicky ekvivalentní s X .

Zde je nutné blíže upozornit na fakt, že v předpokladech věty je uvedeno, že $\bigcap_{s \in S} U_s$ musí být buďto prázdný, nebo kontraktibilní. Z toho vyplývá, že pokud chceme náš nerv zobrazit pouze jako jednodimenzionální simplicialní komplex, pak s geometrickou realizací tohoto jednodimenzionálního simplicialního komplexu budou homotopicky ekvivalentní pouze ty topologické prostory X , jejichž homotopické grupy $\pi_k(X, x_0)$ jsou pro $k > 1$ triviální. To je cena, kterou algoritmus platí za redukci dimenze.

8.2 Statistická verze

V této podkapitole je popsána statistická verze algoritmu Mapper, která slouží k získávání znalostí z dat. Vstupem do tohoto algoritmu jsou tedy data a je tedy nutné upravit pojmy použité v topologické verzi Mapperu tak, aby jej šlo prakticky implementovat a použít k analýze dat.

Pro zbytek kapitoly bude X značit zkoumaný dataset. Upravíme si pojmy filtr, otevřené pokrytí, obloukově souvislé množiny a nerv pokrytí.

Pojem filtr se změní nejméně. Můžeme totiž jednoduše použít libovolné zobrazení $f : D_1 \times \dots \times D_n \rightarrow \mathbb{R}$.

Dále je nutné zvolit otevřené pokrytí \mathbb{R} . Jelikož je množina objektů X konečná, bude také filtr f na všech objektech nabývat nějakého minima a maxima. Nechť \min značí minimum a \max značí maximum f na datasetu X . Jako otevřené pokrytí \mathbb{R} pak můžeme vzít libovolné konečné pokrytí intervalu $[\min, \max]$ otevřenými intervaly. Konkrétně tedy zvolené pokrytí bude mít tvar

$$\mathcal{U} = \{(a_1, b_1)_1, \dots, (a_n, b_n)_n\}.$$

Vhodné je také zmínit, že díky tomu, že je toto pokrytí tvořeno otevřenými intervaly, pak, pokud bude obsahovat alespoň dva intervaly, budou se tyto intervaly různě překrývat.

Z tohoto pokrytí lze snadno pomocí funkce f získat pokrytí datasetu X :

$$f^{-1}(\mathcal{U}) = \bigcup_{(a_i, b_i) \in \mathcal{U}} \{x_j \mid x_j \in X \text{ a } a_i < f(x_j) < b_i\}$$

Pojem obloukově souvislé množiny se mění nejvíce. V původním článku [4] je tento pojem nahrazen takzvaným *shlukováním*.

Konkrétně, pro danou metriku $d : X \times X \rightarrow \mathbb{R}$ a zvolený parametr ϵ definujeme relaci \sim_ϵ na X takto:

$$x_i \sim_\epsilon x_j \text{ právě tehdy, když } d(x_i, x_j) \leq \epsilon,$$

kde $x_i, x_j \in X$. Označme \simeq_ϵ nejmenší relaci ekvivalence na X takovou, která obsahuje relaci \sim_ϵ . Výsledkem shlukování je rozklad X podle relace \simeq_ϵ a třídy ekvivalence podle \simeq_ϵ se nazývají *shluky*. Čím větší je hodnota parametru ϵ , tím více objektů se sloučí do jednoho shluku.

Z $f^{-1}(\mathcal{U})$ pak vytvoříme systém množin tak, že každou množinu v $f^{-1}(\mathcal{U})$ rozdělíme na základě získaných shluků. Tuto množinu označíme \mathcal{S}_ϵ .

POZNÁMKA 8.4

Zde si dovoluji zmínit, že lze použít jakýkoliv způsob, který rozdělí objekty datasetu do nějakých množin. Není nutné se striktně držet pouze shlukování, které bylo použito v algoritmu v původním článku. Je však nutné vzít v úvahu, že zvolený způsob dělení objektů zásadním způsobem změní chování Mapperu.

Nakonec z množiny \mathcal{S}_ϵ vytvoříme objekt podobný nervu pokrytí. Ve statistické verzi se již můžeme bavit přímo o konstrukci grafu. Výstupní graf tedy vytvoříme tak, že každou množinu $v \in \mathcal{S}_\epsilon$ vezmeme jako vrchol a mezi dvěma vrcholy $u, v \in \mathcal{S}_\epsilon$ bude hrana právě tehdy, když $u \cap v \neq \emptyset$.

V této chvíli známe všechno potřebné k popsání algoritmu statistické verze Mapperu. Výstupem statistického Mapperu je tedy graf, jehož uzly jsou tvořeny množinami objektů. Mezi dvěma uzly je hrana, pokud tyto dva vrcholy sdílí nějaký objekt. Získání nějaké znalosti z dat pak zase závisí na naší schopnosti interpretovat význam tohoto grafu.

Celý postup algoritmu je popsán v algoritmu 8.2.

Postup algoritmu *Statistický Mapper*

Vstup:

Dataset X

Filtr $f : X \rightarrow \mathbb{R}$

Konečné otevřené pokrytí $\mathcal{U} = \{(a_1, b_1)_1, \dots, (a_n, b_n)_n\}$
intervalu $[\min_{x \in X} f(x), \max_{x \in X} f(x)]$

Parametr $\epsilon \in \mathbb{R}$ pro shlukování

Výstup:

Graf G

Postup:

1A: $f^{-1}(\mathcal{U}) = \cup_{(a_i, b_i)_i \in \mathcal{U}} \{x_j \mid x_j \in X \text{ a } a_i < f(x_j) < b_i\}$

2A: $\mathcal{S}_\epsilon = \emptyset$

2B: Pro každou množinu $V \in f^{-1}(\mathcal{U})$:

2C: \bar{V} = výsledek shlukování na množině V s parametrem ϵ

2D: $\mathcal{S}_\epsilon = \mathcal{S}_\epsilon \cup \bar{V}$

3A: G = graf s množinou vrcholů \mathcal{S}_ϵ a prázdnou množinou hran

3B: Pro každé dvě množiny $u, v \in \mathcal{S}_\epsilon$:

3C: Pokud $u \cap v \neq \emptyset$:

3D: Přidej hranu (u, v) do grafu G

4A: Vrať G

Algoritmus 8.2: Statistická verze algoritmu Mapper. Význam parametrů je podrobněji popsán v textu výše.

Tento algoritmus má tedy čtyři vstupní parametry: X , f , \mathcal{U} a ϵ . Volba parametru X je jasná, jedná se o dataset, ze kterého chceme získat nějaké znalosti. Volba ostatních parametrů je složitější a většinou již vyžaduje po uživateli jistý náhled do zkoumaných dat.

Filtr se zpravidla volí tak, aby co nejlépe odpovídal záměru naší analýzy. Například, pokud nás zajímá vývoj hodnot nějakého atributu, můžeme zvolit obyčejnou projekci na hodnotu tohoto atributu. Často se také volí nějaké statistické funkce, například aproximace funkce hustoty, která je definována předpisem

$$f_\epsilon(x) = C_\epsilon \sum_{y \in X} \exp\left(\frac{-d(x, y)^2}{\epsilon}\right), \quad (8.1)$$

kde pro f_ϵ platí

$$\int_X f_\epsilon(x) dx = 1.$$

Z předpisu f_ϵ je vidět, že čím více bude daný objekt $x \in X$ vzdálen od ostatních objektů v X , tím menší bude hodnota $f_\epsilon(x)$, a tím pádem jeho pravděpodob-

nost je menší. Pravděpodobnostní rozložení f_ϵ se pak blíží normálnímu rozložení pravděpodobnosti [14].

Dále pak můžeme například použít funkci

$$E_p(x) = \left(\frac{\sum_{y \in X} d(x, y)^p}{N} \right)^{\frac{1}{p}}, \quad (8.2)$$

kteřá pro zvolený parametr p vyjadřuje blízkost k imaginárnímu středu datasetu X . Objekty, které mají menší hodnotu $E_p(x)$, jsou tedy blíže k imaginárnímu středu datasetu X . Chování této funkce je velmi podobné chování předchozí funkce f_ϵ .

U zbývajících dvou parametrů většinou již nemáme možnost, jak smysluplně zvolit ten správný. Uvádím zde základní postupy, které jsem pak využil i při analýze u příkladů v následující kapitole.

Pokrytí \mathcal{U} se často vytváří na základě parametrů: zadáním velikosti intervalu v pokrytí a překryvem dvou po sobě jdoucích intervalů v pokrytí. Pro každý interval $(a_i, b_i)_i$ tedy vyžadujeme, aby platilo $b_i - a_i = l$, kde l je požadovaná velikost intervalu. Pro dva po sobě jdoucí intervaly $(a_i, b_i)_i$ a $(a_{i+1}, b_{i+1})_{i+1}$ pak vyžadujeme, aby měly zvolený překryv. Překryv intervalů se většinou udává v procentech.

Při určování hodnoty parametru ϵ , který udává velikost shluků, nám může pomoci persistentní homologie.

Pomocí 0-dimenzionální homologie můžeme zjistit, které shluky jsou klíčové. Pomocí 1-dimenzionální persistence pak můžeme objevit v grafu cykly.

Při vykreslování výstupního grafu je u velikosti uzlů vhodné zohlednit počet objektů v daných uzlech. Čím více objektů tedy daný uzel obsahuje, tím by měl být větší. Obarvení uzlů se pak volí dle situace tak, aby zohlednilo nějaké vlastnosti objektů v uzlech.

9 Příklady použití statistického Mapperu

V této kapitole se pokouším analyzovat tři datasety pomocí statistické verze algoritmu Mapper. Získané výsledky se také pokouším interpretovat. Na závěr této kapitoly je pak krátké porovnání algoritmu Mapper se shlukováním.

9.1 Dataset World Happiness Report

Tento dataset je výsledkem celosvětového průzkumu pocitu štěstí. Každý respondent byl požádán, aby na škále od 0 do 10 ohodnotil, jak šťastný žije život, přičemž hodnota 0 představuje nejhorší možný a hodnota 10 představuje nejlepší možný. V této práci jsem použil verzi z roku 2017, přičemž v tomto roce probíhal průzkum ve 155 zemích světa a v každé z těchto zemí se jej zúčastnilo okolo 3000 respondentů. Dále pak dataset obsahuje dalších 6 faktorů. Hodnoty těchto faktorů byly získány nezávisle na hlavní otázce pocitu štěstí a vědci se pomocí jejich hodnot snaží interpretovat, jakou mírou daný faktor přispívá pocitu štěstí v dané zemi. Dataset pak obsahuje průměr získaných hodnot pro každou zemi. [5]

Výčet použitých⁸ atributů je uveden v tabulce 9.1. Mým cílem je prozkoumat, jakou mírou dané faktory ovlivňují pocit štěstí v dané zemi.

Dataset jsem získal z otevřeného repozitáře Kaggle [19] a průzkum provedla společnost *Gallup* [23].

Vzhledem k uvedenému cíli analýzy jsem algoritmus spustil nad upraveným datasetem. Atributy `Country` a `Happiness-Rank` jsem vypustil a to z toho důvodu, jelikož jejich hodnoty na výsledné skóre štěstí nemají vliv. Dále jsem pro analýzu vypustil atribut `Happiness-Score`, nicméně hodnotu atributu `Happiness-Score` pak dále využívám k interpretaci získaných výsledků.

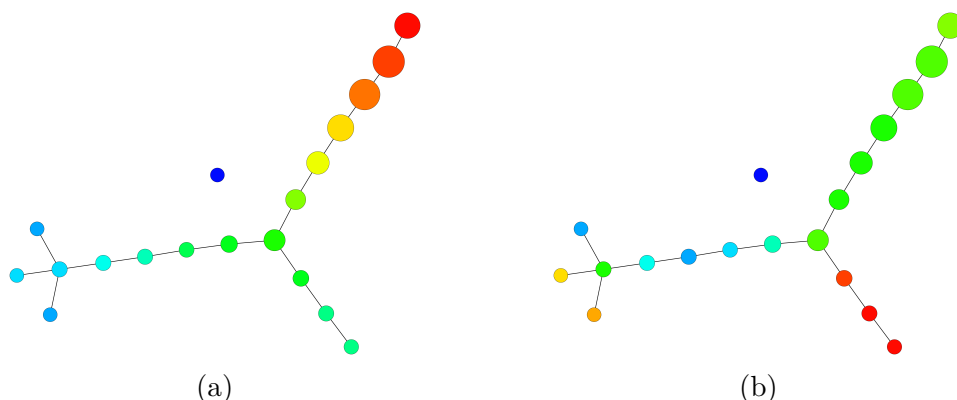
Všechny hodnoty jsem naškáloval na interval $[0, 1]$ a provedl normalizaci hodnot: každou n -tici hodnot objektu lze uvažovat jako vektor v \mathbb{R}^n . Spočítal jsem délku největšího vektoru a touto délkou jsem podělil všechny n -tice. Tyto úpravy dělám především pro pohodlnější analýzu.

⁸Původní dataset ještě navíc obsahuje několik atributů, které však neovlivňují hodnoty faktorů.

Atribut	Obor hodnot	Popis
Country	Řetězce nad anglickou abecedou	Název země
Happiness-Rank	\mathbb{N}	Umístění dané země na základě hodnoty atributu Happiness-Score
Happiness-Score	\mathbb{R}	Úroveň štěstí v dané zemi
GDP	\mathbb{R}	Hrubý domácí produkt
Family	\mathbb{R}	Faktor popisující, zda-li se respondent v případě potřeby může spolehnout na nějaké rodinné příslušníky.
Healthy-Life-Expectancy	\mathbb{R}	Pravděpodobnost, s jakou občan v momentě svého narození bude v dané zemi žít zdravě. Údaje pro výpočet tohoto faktoru jsou získány od Světové Zdravotnické Organizace.
Freedom	\mathbb{R}	Faktor popisující osobní svobodu v dané zemi
Generosity	\mathbb{R}	Faktor sestavený na základě odpovědí, zda-li v blízké minulosti respondent přispěl na nějaké dobročinné účely.
Trust-Government	\mathbb{R}	Faktor popisující důvěru občanů vládě jejich země

Tabulka 9.1: Popis použitých atributů datasetu World Happiness. Posledních 6 atributů jsou faktory. Faktory jsou sestaveny tak, že se pokouší zdůvodnit míru štěstí v dané zemi a platí, že součet hodnot všech faktorů se rovná hodnotě atributu Happiness-Score.

Výstup první analýzy je vykreslen na obrázku 9.1.



Obrázek 9.1: Výstup statistického Mapperu aplikovaného na dataset World Happiness Report. Filtrem je funkce aproximace hustoty s parametrem 0.2, otevřené pokrytí je dáno velikostí intervalu $1/7$ z celkové velikosti intervalu hodnot filtru s překryvem 50% a pro shlukování je zvolen parametr 0.45. Na obrázku (a) jsou uzly obarveny na základě hodnoty filtru: čím vyšší je průměrná hodnota filtru objektů v daném uzlu, tím je daný uzel obarven červeněji. Modře jsou pak obarveny nižší hodnoty. Na obrázku (b) je výstup pro stejné parametry a uzly jsou obarveny na základě průměrné hodnoty štěstí objektů v daném uzlu, červená barva značí vyšší hodnoty a modrá nižší. Vidíme, že graf má jeden centrální uzel, z kterého vychází další uzly spojené hranou. Těmto komponentám se říká *světlice*.

Z vlastností použitého filtru víme, že hrana je mezi těmi objekty, které mají podobnou pravděpodobnost. Hrana tedy znázorňuje plynulý přechod hodnot pravděpodobnosti.

Z obrázku 9.1 tedy vidíme, že výsledný graf obsahuje tři velké světlice a každá z těchto světlic vychází z jednoho centrálního uzlu. Na obrázku 9.1a můžeme vidět zvýrazněnou pravděpodobnost a na obrázku 9.1b zvýrazněnou hodnotu pocitu štěstí. Všimněme si, že světlice vpravo dole obsahuje nejšťastnější země světa, zatímco světlice vpravo nahoře obsahuje země s průměrnou úrovní pocitu štěstí.

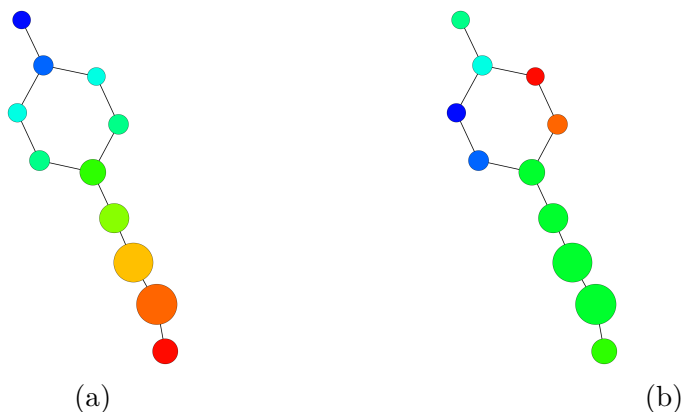
Prozkoumáním objektů v obou světlicích jsem zjistil, že největší rozdíl mezi zeměmi v těchto dvou světlicích je v hodnotě atributu *Trust-Government*. Z grafu lze tedy vyvodit, že v těch státech, ve kterých má důvěra ve vládu větší vliv na štěstí občanů, mají celkově šťastnější obyvatelstvo.

Ve třetí světlici všechny hodnoty faktorů postupně od středového uzlu klesají. Na konci této světlice se nacházejí další tři světlice. Z obrázku 9.1b vidíme, že obsahuje dva žluté šťastnější uzly, což značí jisté zlepšení. V dolním z těchto dvou žlutých uzlů vzrostly hodnoty všech faktorů celkem vyváženě. V horním žlutém uzlu naopak výrazně klesla hodnota atributu *GDP* oproti zelenému uzlu, ze kterého tato světlice vychází a přesto se zvedl celkový pocit štěstí. Naopak v modrém uzlu nahoře hodnota atributu *GDP* oproti zelenému uzlu klesla.

Lze tedy usoudit, že u zemí v těchto třech uzlech hraje důležitou roli v pocitu štěstí hrubý domácí produkt.

Nyní se podíváme na druhou komponentu tohoto grafu. Tu tvoří jediný, tmavě modrý uzel. Důvod, proč je tento uzel zcela osamocen je ten, že hodnoty faktorů se příliš liší od všech ostatních. Ze statistického hlediska je tedy pravděpodobnost výskytu těchto hodnot malá.

Pro zajímavost jsem udělal ještě jednu analýzu. Její výstup je na obrázku 9.2.

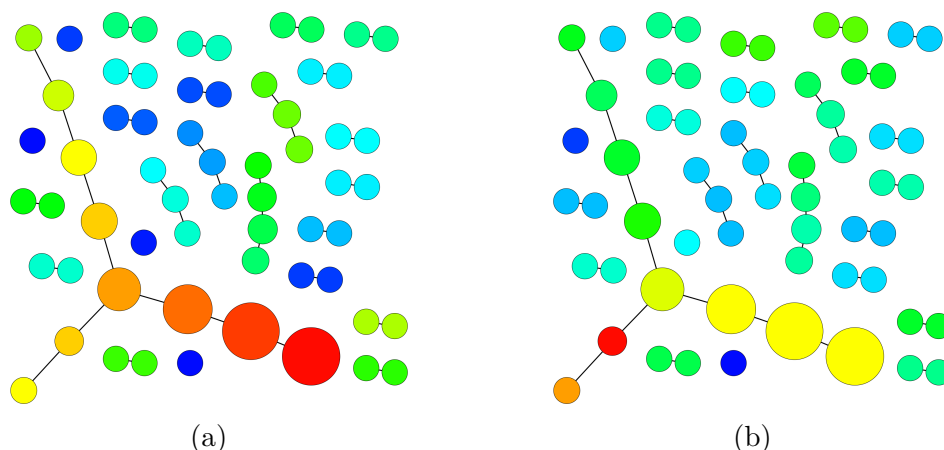


Obrázek 9.2: Výstup statistického Mapperu aplikovaného na dataset World Happiness Report: filtrem je funkce aproximace hustoty s parametrem 0.2, otevřené pokrytí je dáno velikostí intervalu $1/5$ z celkové velikosti intervalu hodnot filtru s překryvem 45% a pro shlukování je zvolen parametr 0.45. Obarvení uzlů na obrázcích (a) i (b) je stejné jako v předchozím příkladě na obrázku 9.1

Na výsledek obrázku 9.2 je vhodné zase nahlížet tím způsobem, že spojení hranou znamená plynulou změnu v hodnotách faktorů. Vidíme pak, že se v grafu nachází jedna kružnice. Z obrázku 9.2b pak vidíme, že dolní a horní část této kružnice obsahuje uzly se zcela jinou úrovní pocitu štěstí. Ze statistického pohledu tedy mají země v horní i spodní části kružnice stejnou pravděpodobnost výskytu, ale hodnoty jejich faktorů jsou odlišné a proto se v grafu vytvořila tato kružnice.

Myslím si, že tato analýza přinesla zajímavé výsledky. Ještě zajímavější by dle mého názoru mohlo být prozkoumat původní data faktorů, tedy bez předzpracování a vyhodnocení vědci z [5]. Hodnoty faktorů jsou navíc nastaveny tak, že jejich součet se rovná hodnotě atributu `Happiness-Score`. Například tedy z hodnoty faktoru `Trust-Government` nelze vyčíst, jestli jsou občané v dané zemi více spokojeni či nespokojeni s vládou, ale pouze zda toto téma více řeší. Hodnoty faktorů tedy hodnotu atributu `Happiness-Score` určují nepřímou. Pro třetí analýzu jsem tedy navíc upravil hodnoty tak, že velikost vektoru hodnot každé n -tice objektu je rovna 1.

Výsledek této analýzy je na obrázku 9.3.



Obrázek 9.3: Výstup statistického Mapperu aplikovaného na dataset World Happiness Report. Filtrem je funkce aproximace hustoty s parametrem 0.2, otevřené pokrytí je dáno velikostí intervalu $1/10$ z celkové velikosti intervalu hodnot filtru s překryvem 50% a pro shlukování je zvolen parametr 0.3. Obarvení uzlů na obrázcích (a) i (b) je stejné jako v předchozích dvou příkladech.

Z obrázku 9.3 je vidět, že tento výstupní graf již obsahuje mnoho komponent. Vývoj pravděpodobnosti tedy již není tak spojitý jako v předchozích dvou případech, nicméně největší z těchto komponent také obsahuje tři světlice. Mezi dolní pravou a dolní levou světlicí je největší rozdíl v hodnotách faktorů `Family` na prvním místě a `Trust-Government` na místě druhém. Výsledek v tomto případě tedy není tak výrazný, jako v prvním případě.

9.2 Dataset World University Rankings

Tento dataset obsahuje žebříček 1000 nejlépe hodnocených vysokých škol a univerzit z roku 2014 z celého světa. Příčka v žebříčku je dána celkovým dosaženým skóre, které se počítá z několika vybraných faktorů. Dataset tyto faktory obsahuje také. [24]

Dataset jsem získal z otevřeného repozitáře Kaggle [20] a všechny atributy tohoto datasetu i s jejich obory hodnot a s krátkým popisem jsou uvedeny v tabulce 9.2.

Dataset jsem původně úspěšně analyzoval pomocí Ripsovy filtrace a persistentní homologie podle postupu uvedeného v kapitole 7.1. V datech se mi podařilo najít jednu jednodimenzionální díru s relativně velkou persistencí a podařilo se mi i interpretovat její význam. Nicméně jsem se nakonec rozhodl využít algoritmus Mapper k vizualizaci výsledků.

Atribut	Obor hodnot	Popis
Institution	řetězce nad anglickou abecedou	Název univerzity nebo vysoké školy
Country	řetězce nad anglickou abecedou	Země, ve které se daná univerzita nebo vysoká škola nachází
Score	\mathbb{R}	Celkové dosažené skóre
World-rank	\mathbb{N}	Umístění na světovém žebříčku na základě hodnoty Score
National-rank	\mathbb{N}	Umístění na domácím žebříčku na základě hodnoty Score
Quality-of-education	\mathbb{R}	Kvalita výuky
Alumni-employment	\mathbb{R}	Zaměstnanost absolventů
Quality-of-faculty	\mathbb{R}	Kvalita zaměstnanců měřená na základě získaných ocenění zaměstnanců vědeckými cenami
Publications	\mathbb{R}	Kvalita publikací měřená na základě počtu článků s vysokým impaktem
Influence	\mathbb{R}	Vliv univerzity měřený na základě počtu článků v odborných časopisech
Citations	\mathbb{R}	Ohodnocení na základě počtu citovaných prací z dané univerzity
Broad-impact	\mathbb{R}	Celkový vliv dané instituce
Patents	\mathbb{R}	Počet patentů

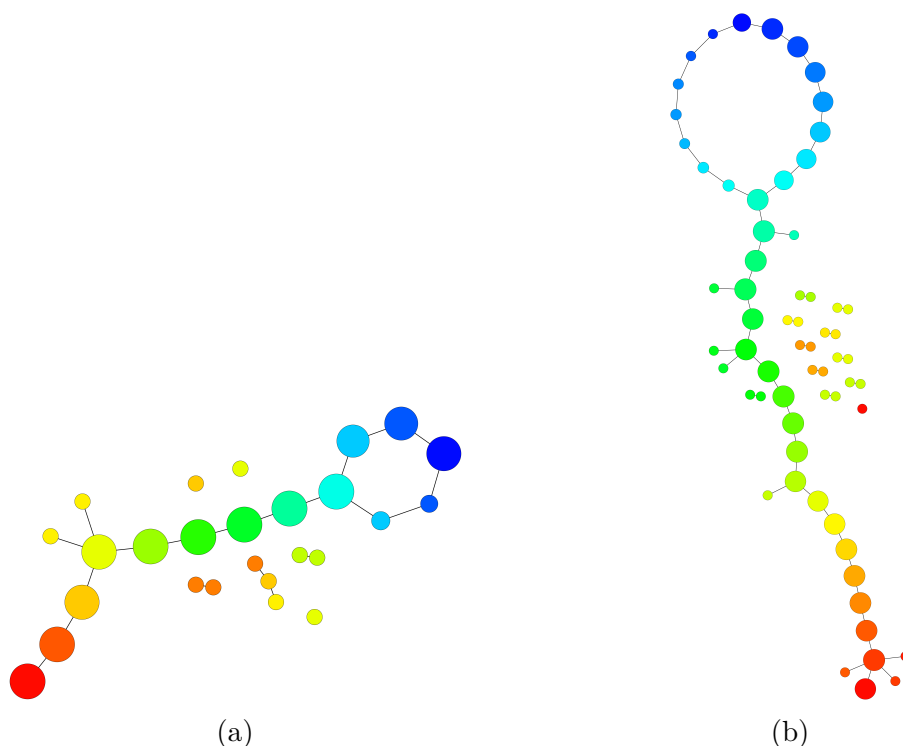
Tabulka 9.2: Atributy datasetu World University Rankings s krátkým popisem.

Pro analýzu jsem vypustil atributy `Institution` a `Country`, jelikož na umístění v žebříčku nemají vliv⁹. Dále jsem všechny hodnoty v každé n -tici podělil délkou největšího vektoru. Výstup je na obrázku 9.4.

Na obrázku 9.4 si lze všimnout cyklu tvořeného modrými uzly. Oba dva výstupy 9.4a i 9.4b zachycují tutéž informaci.

Podívejme se tedy na obrázek 9.4a: vysoké školy ve dvou větších uzlech nahoře jsou nízko umístěny ve světovém i na svém domácím žebříčku. Hodnota obou atributů `World-rank` i `National-rank` je tedy nízká. Vysoké školy ve dvou menších uzlech dole jsou umístěny ve světovém žebříčku nízko, ale ve svém domácím žebříčku vysoko. Hodnota atributu `World-rank` je tedy nízká, ale hodnota atributu `National-rank` je vysoká. Vysoké školy, které se nachází v posledním modrém uzlu pak mají hodnoty atributů `World-rank` i `National-rank` vyrovnané.

Situace na obrázku 9.4b je pak analogická.



Obrázek 9.4: Výstup algoritmu Mapper pro dataset World University Rankings. Jako filtr je použita projekce na hodnotu atributu `World-rank`, parametr ϵ pro shlukování zvolen 0.65. Na obrázku (a) je pokrytí zadané velikostí intervalu $1/7$ z celkové velikosti intervalu hodnot filtru s velikostí překryvu 45% a na obrázku (b) je zadané velikostí intervalu $1/10$ se stejným překryvem. Uzly na obou obrázcích jsou obarveny na základě hodnoty filtru, tedy hodnoty atributu `World-rank`: čím je uzel červenější, tím je hodnota tohoto atributu vyšší.

⁹Zanedbávám zde fakt, že mezi zemí univerzity a její pozicí ve světovém žebříčku může být nějaký vztah.

9.3 Dataset Hayes Roth

Dataset Hayes-Roth byl uměle vytvořen za účelem otestování různých metod klasifikace objektů. Klasifikace je postup, který na základě určitých pravidel zařadí daný objekt do určité kategorie. Této kategorii se říká *třída*. Cílem této kapitoly je ukázat, že pomocí algoritmu Mapper lze získat informace, které mohou být při klasifikaci užitečné.

V původním výzkumu se testování zúčastnili středoškolští studenti v USA. Výzkum se velmi podobal modernímu přístupu algoritmů strojového učení ke klasifikaci. Nejprve bylo každému studentovi předloženo několik objektů i s třídou, do které je daný objekt klasifikován. Během této fáze bylo úkolem studenta na základě vysvětlené metody co nejlépe pochopit, proč je daný objekt klasifikován do té či oné třídy. Poté bylo studentovi předloženo několik dalších objektů a úkolem studenta bylo určit, do které třídy patří [3].

Popišme si tento dataset podrobněji. Každý objekt datasetu popisuje fiktivní osobu. U každé osoby je zaznamenáno celkem 6 atributů: jméno, hobby, věk, vzdělání, rodinný stav a klub. Atribut jméno je pro každou osobu unikátní. Atribut hobby nabývá tři různých hodnot, atributy věk, vzdělání a rodinný stav nabývají čtyř různých hodnot. Ve verzi datasetu, který jsem získal, jsou hodnoty značeny pouze číslem.

Třídou každého objektu, do které je klasifikován, je hodnota atributu klub. Daná osoba může být buď v klubu 1, 2, anebo v žádném, což je značeno hodnotou 3. Cílem je tedy každou osobu zařadit do správného klubu, nebo konstatovat, že osoba není členem žádného klubu. Nejprve tedy byly studentům předloženy objekty i s třídou a poté jim byly předloženy objekty, u kterých jim byla třída zatajena.

V úvodu bylo řečeno, že tento dataset je uměle vytvořený. Postup vytvoření datasetu byl následující. Při tvorbě datasetu byly zvoleny dva prototypy. První obsahuje na attributech věk, vzdělání a rodinný stav pouze hodnotu 1, druhý obsahuje na těchto třech attributech pouze hodnotu 2. Poté byly z těchto prototypů vygenerovány ostatní objekty, přičemž ani jeden prototypový objekt se ve výsledném datasetu nenachází. Náhodně byly z těchto tří výše zmíněných atributů vybrány dva a jejich hodnota byla změněna na náhodné číslo. Poté, pokud na attributech věk, vzdělání a rodinný stav takto upravený objekt obsahuje hodnotu 4, je klasifikován do třídy 3 (nepatří tedy do žádného klubu). Pokud na attributech věk, vzdělání a rodinný stav obsahuje více jedniček než dvojek, je mu přiřazena třída 1. Pokud obsahuje více dvojek než jedniček, pak je mu přiřazena třída 2. Pokud obsahuje stejný počet jedniček i dvojek, pak je náhodně určena buď třída 1, nebo 2 se stejnou pravděpodobností. Hodnota atributu hobby je vygenerována náhodně a na příslušnost objektu do dané třídy nemá žádný vliv.

Student, kterému se podařilo odhalit tato pravidla, měl pak při klasifikaci 100% úspěšnost.

Nyní se přesuneme k vlastní analýze. Pro pohodlnější analýzu nahradíme hodnoty atributu *hobby* takto: hodnotu 1 nahradíme hodnotou A , hodnotu 2 hodnotou B a hodnotu 3 hodnotou C . Tato úprava nijak nemění význam generování, ale usnadní nám komentáře během interpretace získaných výstupů.

Vzhledem k tomu, že obory hodnot všech atributů v tomto datasetu jsou diskrétní, je vhodné použít takzvanou *Hammingovu vzdálenost*. Tato vzdálenost je pro dva objekty $x, y \in X$ dána počtem atributů, na kterých se hodnoty těchto objektů liší. Hammingovu vzdálenost budeme značit d_H a použijeme jak při shlukování, tak i ve filtru.

Jako filtr vezmeme funkci E_p , která je definovaná předpisem 8.2 a upravíme ji tak, aby vzdálenost byla počítána pouze k objektům ze stejné třídy. Tím docílíme toho, že hodnota získaná touto upravenou funkcí bude reprezentovat blízkost daného objektu k pomyslnému středu všech objektů jeho třídy.

Nechť tedy c_x značí třídu objektu $x \in X$. Upravená funkce je definována předpisem

$$E'_p(x) = \left(\frac{\sum_{y \in X, c_y = c_x} d_H(x, y)^p}{|\{y \in X \mid c_y = c_x\}|} \right)^{\frac{1}{p}}.$$

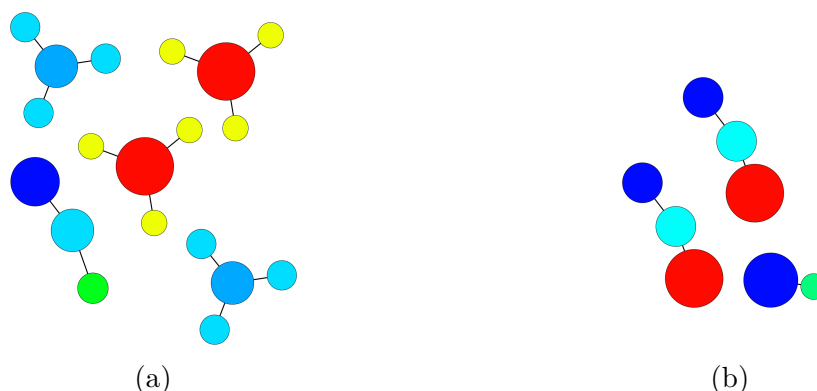
Hodnota $E'_p(x)$ je pak tím menší, čím více je objekt x blíže k imaginárnímu středu objektů stejné třídy. Navíc ze způsobu, jakým byl dataset Hayes-Roth vygenerován, vyplývá, že E'_p také měří vzdálenost od jednotlivých prototypů, přestože se tyto prototypy v datasetu nenacházejí. Ve všech příkladech jsem zvolil $p = 1$.

Obarvení uzlů ve všech výstupech Mapperu v této kapitole odpovídá průměru hodnot funkce E'_p získaných z objektů v daném uzlu: čím je uzel červenější, tím je průměrná hodnota E'_p nižší a tím pádem jsou dané objekty blíže ke středu.

Výstup takto upraveného statistického Mapperu je na obrázku 9.5. Na obrázku 9.6 je výstup Mapperu, kde je navíc při shlukování přidána podmínka, že dva objekty mohou být sloučeny do stejného shluku jen tehdy, pokud mají stejnou třídu.



Obrázek 9.5: Na obrázku (a) je výstup s parametry velikostí intervalu $1/3$ z celkové velikosti intervalu hodnot filtru, překryvem 50% a vzdáleností pro shlukování 3. Velký červený uzel a s ním spojené malé žlutozelené uzly obsahují pouze ty objekty, u kterých byla třída určena na základě počtu 1 a 2. Zelený uzel vpravo obsahuje pouze objekty třídy 3. Světlejší modrý uzel obsahuje objekty třídy 3 a ty objekty, které obsahují stejný počet 1 a 2 a třída u nich tedy byla určena náhodně. Poslední tmavě modrý uzel obsahuje navíc objekty třídy 1 a 2, které neobsahují buďto žádnou jedničku, nebo žádnou dvojku. Na obrázku (b) je výstup s parametry velikostí intervalu $1/3$ z celkové velikosti intervalu hodnot filtru, překryvem 50% a vzdáleností pro shlukování 3, kde pro výpočet vzdáleností při shlukování není použita hodnota atributu `k1ub`. Velký červený uzel a s ním spojené malé žlutozelené uzly obsahují pouze ty objekty, u kterých byla třída určena na základě počtu 1 a 2, přičemž každý objekt v těchto uzlech obsahuje nejméně jednu 1 nebo jednu 2. Zelený uzel vlevo obsahuje pouze objekty třídy 3. Světlejší modrý uzel uprostřed obsahuje objekty třídy 3 a pak ty objekty třídy 1 nebo 2, které mají stejný počet 1 a 2 a tudíž u nich byla třída určena náhodně. Největší, tmavě modrý uzel pak ještě navíc obsahuje ty objekty třídy 1 nebo 2, které obsahují maximálně jednu 1 nebo jednu 2.



Obrázek 9.6: U této analýzy je navíc přidána podmínka, že objekty se mohou sloučit do stejného shluku jen tehdy, pokud mají stejnou třídu. Na obrázku (a) vlevo je výstup Mapperu s parametry velikostí intervalu $1/3$ z celkové velikosti intervalu hodnot filtru, překryvem 50% a vzdáleností pro shlukování 3. Velké červené uzly a žluté uzly s nimi spojené hranou obsahují pouze ty objekty třídy 1 nebo 2, podle toho, jestli daný objekt obsahuje více 1 nebo 2. Tmavší modré uzly a s nimi vždy tři spojené světle modré uzly obsahují převážně ty objekty, které obsahují stejný počet 1 a 2, a tudíž u nich byla třída určena náhodně. Zbylé tři uzly, tmavě modrý, světle modrý a zelený pak obsahují objekty třídy 3. Na obrázku (b) vpravo je výstup s parametry velikostí intervalu $1/2$ z celkové velikosti intervalu hodnot filtru, překryvem 50% a vzdáleností pro shlukování 3. V každé ze dvou trojic uzlů spojených hranou velké červené uzly obsahují ty objekty, u kterých byla třída určena na základě počtu 1 a 2. Menší tmavě modré uzly v těchto trojicích obsahují objekty, u kterých byla třída určena náhodně. Světle modré uzly uprostřed pak obsahují oba typy objektů. Modrý a zelený uzel dole obsahuje pouze objekty třídy 3.

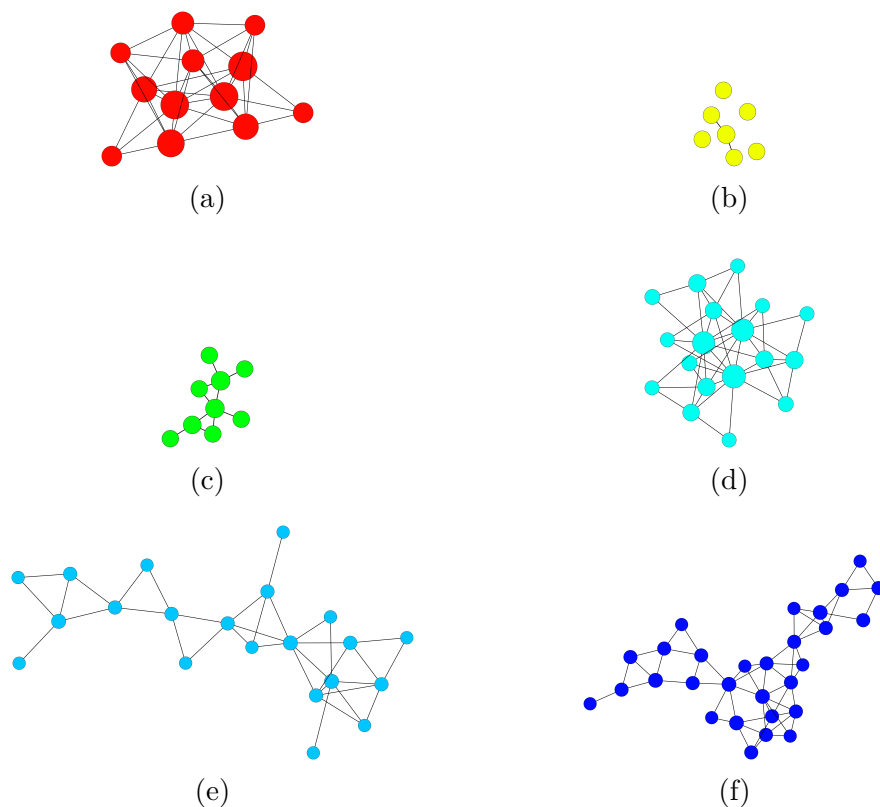
Z výstupních grafů je vidět, že zohledňují způsob, jakým je dataset vygenerován, a také jakým způsobem je objektům přidělena třída.

Také by se zde nabízela možnost použít na získané grafy ještě nějaké další metody určené k získávání znalostí z dat, které by odhalení těchto souvislostí dále usnadnily.

Výsledek je dobrý především díky použitému filtru E'_p . Tento filtr jsem po krátké úvaze zvolil přesně na míru danému problému. Nicméně si myslím, že jeho použití by bylo opodstatněné i v případě, kdy by objekty nebyly generovány na základě existence nějakých prototypů.

Vzhledem k tomu, že hodnoty atributů jsou diskrétní, můžeme uvažovat také následující shlukování: jako shluk dat vezmeme největší množinu objektů takovou, ve které se hodnoty atributů objektů rovnají na dané n -tici atributů. Pro více zvolených n -tic atributů pak vytvoříme množinu všech shluků tak, že sjednotíme shluky získané pro každou n -tici ze zvolených n -tic. Při tomto shlukování se jeden objekt může nacházet ve více shlucích, a tím pádem již shluky netvoří třídy ekvivalence.

Výstup statistického Mapperu s tímto shlukováním je na obrázku 9.7.



Obrázek 9.7: Otevřené pokrytí je dané velikostí intervalu $1/7$ z celkové velikosti intervalu hodnot filtru s překryvem 33%. Pro shlukování byly vybrány všechny tři prvkové kombinace atributů hobby, věk, vzdělání a rodinný stav. Výstupní graf Mapperu je rozdělený na obrázcích (a) - (f) a obarvení uzlů je stejné jako na předchozích dvou obrázcích. Uzly na obrázku (a) obsahují objekty třídy 1 a 2, u kterých byla třída určena na základě počtu 1 a 2, přičemž každý objekt obsahuje nejméně jednu 1 nebo jednu 2. Na obrázku (b) uzly obsahují objekty třídy 1 a 2, u kterých byla třída určena na základě počtu 1 a 2, avšak každý objekt obsahuje buď nulový počet 1, nebo nulový počet 2. Na obrázku (d) uzly obsahují objekty třídy 1 a 2, které obsahují stejný počet 1 a 2, a tudíž u nich byla třída určena náhodně. Uzly na obrázcích (c) a (e) obsahují pouze objekty třídy 3. Uzly na obrázku (f) pak obsahují objekty třídy 3 a ty objekty třídy 1 nebo 2, které však obsahují právě jednu 1, nebo jednu 2.

Z obrázku 9.7 je tedy vidět, že výstupní graf ještě lépe odpovídá rozdělení objektů do tříd a také mnohem více odpovídá pravidlům použitým při generování datasetu. Tento výsledek je dobrý právě díky kombinaci filtru E'_p s použitým shlukováním. Shlukování objektů na základě stejných hodnot na n -ticích atributů lépe vystihuje způsob, kterým je tento dataset vygenerován.

9.4 Porovnání statistického Mapperu se shlukováním

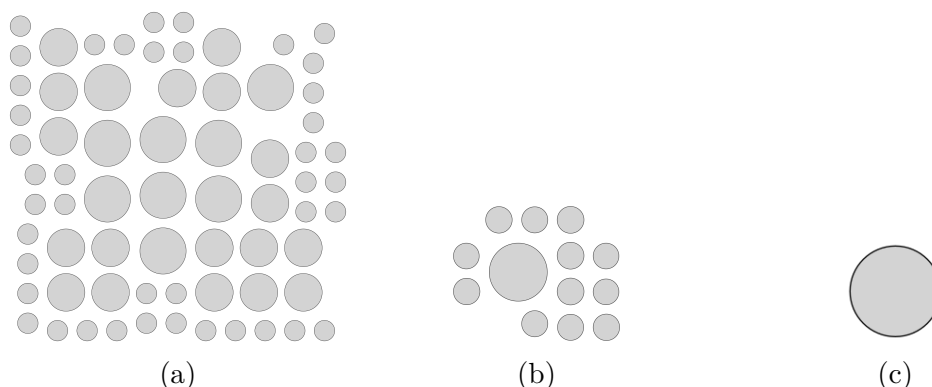
Jelikož klíčovou částí statistického Mapperu je shlukování, nabízí se udělat srovnání přímo se shlukováním samotným. Výsledkem samostatného shlukování jsou pak pouze shluky objektů.

Hned na první pohled velkou výhodou Mapperu je, že mezi shluky dat vytváří hrany. Tyto hrany pak mohou přispět k interpretaci samotných shluků. Dále pak pomocí samotného shlukování se mi některých výsledků ani nepodařilo docílit.

Jako takový příklad ukážu analýzu datasetu Hayes-Roth z předchozí kapitoly. Výsledkem aplikace Mapperu na tento dataset byl graf, ze kterého šlo vyčíst informaci o existenci prototypů a celkově byly také objekty ve výsledných uzlech v grafu dobře rozděleny dle svých tříd.

Analyzujeme tedy tento dataset pouze pomocí shlukování a pro vytváření shluků použijme Hammingovu vzdálenost. Na tomto datasetu získáme pomocí shlukování různé výsledky pouze pro parametry 0, 1, 2 a 3. Pro parametry 3 a vyšší se totiž všechny objekty sloučí do jednoho shluku. Pro parametr $\epsilon = 0$ je výsledek shlukování triviální: každý objekt bude tvořit samostatný shluk.

Výsledek shlukování pro hodnoty 1, 2 a 3 je na obrázku 9.8.



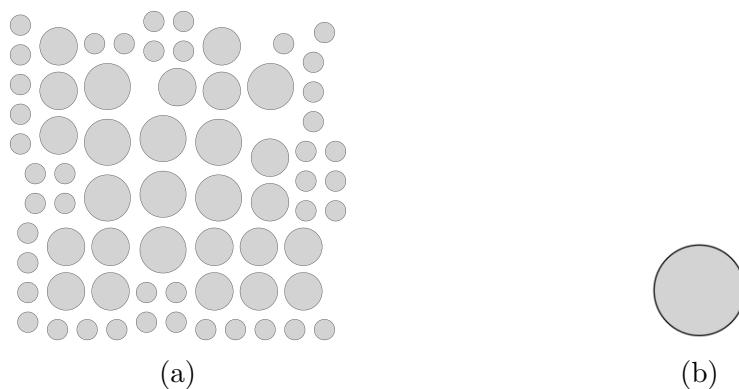
Obrázek 9.8: Výsledek shlukování na datasetu Hayes Roth. Na obrázku (a) je výsledek pro parametr $\epsilon = 1$. Pro tento parametr je výsledek shlukování také triviální. Díky tomu, že hodnota atributu `jmeno` je pro každý objekt unikátní, se při parametru $\epsilon = 1$ sloučí do jednoho shluku právě ty objekty, které mají shodné hodnoty na všech ostatních attributech. Na obrázku (b) je výsledek shlukování s parametrem $\epsilon = 2$. Největší shluk obsahuje všechny objekty třídy 1 a 2. Ostatní menší uzly pak obsahují různě pouze objekty třídy 3. Na obrázku (c) je výsledek shlukování pro parametr $\epsilon = 3$ a při tomto parametru jsou již všechny objekty sloučeny do jednoho shluku.

Z obrázku 9.8 je tedy patrné, že smysl má zkoumat pouze shluky získané s parametry $\epsilon = 2$ a $\epsilon = 3$.

Při použití parametru $\epsilon = 2$ se objekty třídy 1 a 2 sloučily do jednoho shluku. To znamená, že každé dva tyto objekty se liší maximálně ve dvou hodnotách atributů. Tato informace by mohla být užitečná, avšak celkový výsledek je horší než například obrázku 9.5a. Na parametru $\epsilon = 3$ je zajímavé, že právě pro hodnotu 3 se celý dataset shlukne do jednoho jediného shluku. Dataset má totiž celkem 6 atributů a hodnota parametru 3 tedy může naznačovat nějaké souvislosti mezi hodnotami atributů. Nicméně dále je zkoumání tohoto shluku stejné jako zkoumání celého datasetu a z této informace již nelze nic víc získat.

Pro úplnost si ukážeme i výsledek shlukování bez použití atributu `klub`. Při tomto shlukování má smysl prozkoumat pouze parametry 0, 1 a 2, přičemž výsledek shlukování pro parametr $\epsilon = 0$ je triviální a pro parametry $\epsilon = 2$ a vyšší se již celý dataset sloučí do jednoho shluku.

Výsledek pro parametry 1 a 2 je tedy na obrázku 9.9.



Obrázek 9.9: Výsledek shlukování na datasetu Hayes Roth bez použití hodnot atributu `klub`. Na obrázku (a) je výsledek pro parametr $\epsilon = 1$. Do jednoho shluku se sloučí ty objekty, které kromě atributu `jméno` mají stejné hodnoty na ostatních attributech. Na obrázku (b) je výsledek pro parametr $\epsilon = 2$ a při této hodnotě se již všechny objekty sloučí do jediného uzlu.

Z obrázku 9.9 je tedy vidět, že bez použití atributu `klub` shlukování nepřináší žádnou užitečnou informaci. Algoritmus Mapper však přinesl zajímavý výsledek i při shlukování bez tohoto atributu. Pro srovnání se můžeme podívat například na výsledek na obrázku 9.5b, ve kterém byla použita informace o třídách ve filtru.

Srovnání provádím pouze na jednom datasetu a nabízí se tedy otázka, zda-li někdy algoritmus Mapper může vrátit i horší výsledek, než shlukování. Odpověď je, že nemůže: parametry algoritmu Mapper totiž lze vždy nastavit tak, aby algoritmus vrátil přesně takové výsledky, jako shlukování. Stačí za otevřené pokrytí zvolit jediný interval, který pokryje celý rozsah hodnot zvoleného filtru.

Na závěr je vhodné podívat se kriticky i na algoritmus Mapper. Nevýhodou algoritmu je jistě velké množství způsobů, kterými lze zvolit jeho vstupní parametry. V tom vychází Mapper samozřejmě hůře než shlukování, jelikož pro shlukování stačí volit pouze jediný parametr. Navíc, jak můžeme srovnat na obrázcích [9.5b](#) a [9.7](#), výsledek Mapperu velice závisí i na použitém shlukování.

Dle mého názoru jsou však výhody algoritmu větší než tyto nevýhody.

Závěr

Cílem této práce bylo nastudovat obecnou a algebraickou topologii a dále podat souhrn vybraných metod a výsledků v oblasti topologické analýzy dat s důrazem na původní příklady. Tato práce obsahuje celkem 5 původních příkladů a v tomto ohledu tedy práce zadání plní.

Příklady použití algoritmu Mapper, které byly prezentovány v kapitole 9 považuji za užitečné. Dále pak analýza datasetu Hayes-Roth, která byla uvedena v příkladu v kapitole 9.3 spolu s následným srovnáním v kapitole 9.4 ukázala výhody použití principů algebraické topologie v analýze dat.

Příklady použití obyčejné simplicialní homologie, které jsou uvedeny v kapitole 7 příliš hodnotné výsledky nepřináší, avšak domnívám se, že pro experty v dané oblasti by tyto výsledky mohly být přínosné.

Conclusions

The aim of this thesis was to study general and algebraic topology and to present a summary of selected methods and results in the field of topological data analysis with emphasis on original examples. Five examples have been shown in this thesis and so the work fulfills the aim.

I consider examples of using Mapper algorithm presented in chapter 9 as useful. Moreover, the analysis performed in the example in the chapter 9.3 followed with the comparison in the chapter 9.4 shows the advantages of using the principles of algebraic topology in data analysis.

The examples of using simplicial homology, that are listed in the chapter 7, do not produce very valuable results. However, I think that experts in given fields could gain some useful knowledge from these results.

A Poznámky k vlastní implementaci a použití software

V této kapitole velmi stručně popisují vlastní implementaci a použitý software.

Pro vlastní implementaci algoritmů jsem použil jazyk Python. Implementoval jsem algoritmus persistentní homologie, který je popsán v kapitole 6.2, a statistický Mapper, který je popsán v kapitole 8.2.

Implementovaný algoritmus persistentní homologie se nachází v souboru `persistence_intervals.py`. Třída `FilteredSimplicialComplex` reprezentuje filtrovaný simplicialní komplex. Filtrovaný simplicialní komplex se nastaví pomocí metody `set_filtration`. Tato funkce má jediný vstupní parametr, kterým je seznam dvojic ve tvaru `(symbol, číslo)`, kde `symbol` je libovolné unikátní označení simplexu a `číslo` značí krok přidání tohoto simplexu. Metoda `compute_intervals` pak implementuje algoritmus 6.3 pro výpočet \mathcal{P} -intervalů. Tento algoritmus lze využít také ke spočítání obyčejné simplicialní homologie: stačí jako krok přidání všech simplexů zadat libovolné stejné číslo.

Pro výpočet Ripsovy filtrace a výpočet \mathcal{P} -intervalů z Ripsovy filtrace jsem použil program *ripser* [21]. Tento program se na CD nenachází, uživatelská příručka a pokyny pro instalaci jsou uvedeny taktéž v [21].

Implementovaný algoritmus Mapper, respektive jeho statistická verze, se nachází v souboru `mapper.py`. Algoritmus reprezentuje třída `Mapper`.

Zdrojový kód obsahuje v této práci analyzované příklady. Tyto příklady jsou implementovány jako samostatné funkce bez vstupních parametrů a začínají vždy prefixem `example_`. Spuštěním těchto funkcí lze tedy reprodukovat v textu uvedené příklady. Tvar výstupních grafů se může mírně lišit od grafů uvedených v této práci v závislosti na použitých verzích veškerých knihoven.

Pro vykreslení grafů jsem použil knihovnu *graphviz* [22]. Uživatelská příručka a pokyny pro instalaci knihovny jsou uvedeny také v [22]. Tuto knihovnu lze nainstalovat také jako balík pomocí správce balíků *PIP*, který slouží ke správě balíků pro jazyk Python. Tato knihovna je vyžadována pro spuštění mé implementace algoritmu Mapper.

B Obsah přiloženého CD/DVD

Přiložené CD má následující adresářovou strukturu:

doc/

Text práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce, včetně všech příloh, a všechny soubory potřebné pro bezproblémové vygenerování PDF dokumentu textu.

src/

Zdrojové kódy implementovaných algoritmů.

src/datasets/

Tato složka obsahuje použité datasey.

readme.txt

Instrukce pro spuštění implementovaných algoritmů, včetně všech požadavků.

Literatura

- [1] CARLSSON Gunnar, *Topology and data*. Bull. Amer. Math. Soc. 46 (2009), 255-308. Leden 2009. [cit. 2018-06-06]. S 0273-0979(09)01249-X.
- [2] GHRIST Robert, *Barcodes: The persistent topology of data*. Bull. Amer. Math. Soc. 45 (2008), 61-75. 26. říjen 2008. [cit. 2018-06-06]. DOI: <https://doi.org/10.1090/S0273-0979-07-01191-3>.
- [3] HAYES-ROTH B., HAYES-ROTH F., *Concept learning and the recognition and classification of exemplars*. Journal of Verbal Learning and Verbal Behavior, 16, 321-338. 1977. [cit. 2018-06-06].
- [4] SINGH Gurjeet, M'EMOLI Facundo, CARLSSON Gunnar, *Topological methods for the analysis of high dimensional data sets and 3D object recognition*. In Proc. Eurographics Symposium on Point-Based Graphics. 2007. [cit. 2019-12-04].
- [5] HELLIWELL J., LAYARD R., SACHS J., *World Happiness Report 2017*. Sustainable Development Solutions Network, New York. Březen 2017. [cit. 2019-06-06]. ISBN 978-0-9968513-5-0. Dostupné z <https://worldhappiness.report/ed/2017/>.
- [6] HATCHER Allen, *Algebraic Topology*. [ONLINE]. 2001. [cit. 2019-12-04]. Dostupné z: <https://pi.math.cornell.edu/~hatcher>.
- [7] MUNKERS James R., *Topology*. Prentice Hall, Upper Saddle River, druhé vydání. 2000. [cit. 2019-12-04]. ISBN 0-13-181629-2.
- [8] MUNKERS James R., *Elements of Algebraic Topology*. Addison-Wesley, Redwood City, California. 1984. [cit. 2019-12-04]. ISBN 0-201-04586-9.
- [9] ZOMORODIAN Afra J., *Topology for Computing*, Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press. Březen 2005. [cit. 2019-12-04]. ISBN 978-0-521-83666-1.
- [10] BOLTJANSKI V., JEFREMOVIČ, V. A, *O topologii*. Pokroky matematiky, fyziky a astronomie, Vol. 4 (1959), No. 3, 265–288. 1959.
- [11] MCINNES Leland, HEALY John, MELVILLE James, *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. 2. verze. 7. prosinec 2018. [cit. 2018-07-07]. arXiv:1802.03426v2.
- [12] ROMAN Steven, *Advanced Linear Algebra*, Graduate Texts in Mathematics. Springer, třetí vydání. 2008. [cit. 2019-12-04]. ISBN-13 978-0-387-72828-5.
- [13] MURFET Daniel, *Graded Rings and Modules*. [ONLINE]. 16 květen 2006. [cit. 2018-05-06]. Dostupné z <http://therisingsea.org>.

- [14] SILVERMAN B. W., *Density estimation for statistics and data analysis*. Monographs on Statistics and Applied Probability. Chapman & Hall, Londýn. 1986. [cit. 2019-07-04].
Dostupné z <https://ned.ipac.caltech.edu/level5/March02/Silverman/paper.pdf>.
- [15] BLAND Paul E., *Rings and Their Modules*. 28 leden 2011. [cit. 2019-07-04]. ISBN 978-3-11-025022-0.
- [16] Dataset *Mall_Customers*.
Dostupné z <https://www.Kaggle.com/shwetabh123/Mall-customers>.
- [17] SCHLIMMER Jeff, *Congressional Voting Records Data Set*. UCI Machine Learning Repository. 1987. [cit. 2018-05-06]. Dostupné z <https://archive.ics.uci.edu/ml/datasets/congressional+voting+records>.
- [18] DAVID W. Aha. UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science. Dostupné z <http://archive.ics.uci.edu/ml>.
- [19] SUSTAINABLE DEVELOPMENT SOLUTIONS NETWORK. *World Happiness Report*. 7. září. 2018. [cit. 2019-06-06].
Dostupné z <https://www.Kaggle.com/unsdsn/world-happiness>.
- [20] Dataset *World University Rankings*. [cit. 2019-01-02]. Dostupné z <https://www.Kaggle.com/mylesoneill/world-university-rankings/data>.
- [21] BAUER Ulrich, *Ripser: a lean C++ code for the computation of Vietoris–Rips persistence barcodes*. Verze 1.0.1, září 2016.
Dostupné z <https://github.com/Ripser/ripser>.
- [22] BANK Sebastian, *Graphviz*. Dostupné z <https://graphviz.readthedocs.io>.
- [23] *Gallup, Inc.* Dostupé z <https://www.gallup.com>.
- [24] *Center for World University Rankings*. Dostupné z <https://cwur.org>.