

Asymptotisk normalitet for tessellation baseret Betti tal

Asymtropic normality for tessellation-based Betti numbers

Bjarke Hautop Kristensen

Bachelorprojekt i statistik

Studienummer: 202005674

Vejleder: Christian Pascal Hirsch

13. juni 2023

1 Abstract

Topological data analysis is a powerful tool for analyzing complex datasets and extracting meaningful topological properties. In this thesis, we explore the use of persistent Betti numbers, which capture a multiscale perspective of topological changes. We present approaches for computing persistent Betti numbers and establish two central limit theorems for persistent Betti numbers. The first theorem applies to general filtrations that satisfy certain conditions, with the Čech filtration being an example that satisfies these conditions. The second theorem is for the Voronoi tessellation-adapted filtration. These theorems provide a fundamental theoretical framework for statistical analysis and hypothesis testing in computational topology. Additionally, we conduct simulations to investigate the behavior of persistent Betti numbers under varying parameters, and propose a hypothesis test to determine if the number of points follows a fixed distribution or stems from a Poisson distribution. Our findings demonstrate the effectiveness of Voronoi tessellation in analyzing the topological structure of datasets.

Indhold

1	Abstract	1
2	Indledning	3
3	Notation og redskaber	4
3.1	Rumlig statistik og topologisk dataanalyse	4
3.2	Tessellations	12
4	Centrale grænseværdisætning for persistent Betti tal	14
4.1	Poissonprocess	15
4.2	Voronoi tessellation	15
5	Beviser	15
5.1	Poissonprocess	15
5.2	Voronoi tessellation	20
6	Simulationer	24
6.1	Persistent Betti tal	24
6.2	Total persistence	26
7	Litteratur	28
8	Appendix	29

2 Indledning

I de seneste år er topologisk dataanalyse opstået som et kraftfuldt værktøj til analyse af komplekse datasæt og ekstraktion af meningsfulde topologiske egenskaber [2]. En sådan egenskab er Betti tal, der giver værdifulde indsigter i de topologiske ændringer på tværs af forskellige niveauer i et datasæt. Betti tal beskriver intuitivt antallet af q -dimensionelle huller i et simplicial kompleks. Simplicial komplekser er geometriske objekter, der opbygges ved at forene punkter, kanter, trekanter, tetraedere og polytoper af højere dimensioner. De udgør et kraftfuldt værktøj til repræsentation af komplekse topologiske rum på en måde, der er tilgængelig for dataanalyse. Ved at fange de fundamentale strukturelle elementer af disse rum tilbyder simpliciale komplekser en værdifuld ramme til udforskning og fortolkning af data inden for forskellige områder, for eksempel i materialevidenskab [5]. Vi vil fokusere på persistent Betti tal, hvor persistent Betti tal også lader os forstå tidsperspektivet for Betti-tal. Persistent Betti-tal giver information om, hvor længe de topologiske egenskaber er til stede i datasættet og i hvilke niveauer.

Vi antager i hele afhandlingen, at vores datapunkter kommer fra en homogen Poissonprocess, som er en grundlæggende stokastisk process, der er meget udbredt. Poissonprocessen modellerer forekomsten af begivenheder i kontinuerlig tid eller rum, og har nogle vigtige egenskaber som uafhængighed, hukommelsesløs og en konstant rate. I vores sammenhæng fungerer den homogene Poissonprocess som en naturlig model for fordelingen af punkter i et datasæt. Disse er nogle idealer, som sandsynligvis ikke vil gælde i den virkelige verden, men dette kan bruges som en model under en nulhypotese.

Først etablerer vi i Afsnit 4.1 en central grænseværdisætning for persistent Betti tal med en filtrering, der opfylder visse betingelser. Beviset laves i Afsnit 5.1 og det består af at vise, at det persistent Betti tal er svagt stabiliserende samt, at det opfylder Poisson begrænset moment kriterium [7].

Derefter præsenterer vi i Afsnit 4.2 en central grænseværdisætning for persistent Betti tal med Voronoi tessellation-tilpassede filtrering. Voronoi tessellation er en geometrisk opdelingsteknik, som laver en celle for hvert punkt med det tilhørende område, der er tættere på dette punkt end på noget andet punkt. Beviset laves i Afsnit 5.2 og beviset minder meget om det

andet bevis, men nogle af argumenterne skal tilpasses det nye setup.

I afsnit 6 laver vi simulationer af den centrale grænseværdisætning med Voronoi-tesselering og undersøger, hvad der sker, når vi ændrer parametrene. Vi opstiller også to hypotesetests. Den ene har nulhypotesen om et fast antal punkter og den alternative hypotese om, at antallet af punkter følger en Poissonfordeling. Denne test har en simuleret styrke på 0,25. Den anden hypotesetest har nulhypotesen om, at antallet af punkter følger en Poissonfordeling, og den alternative hypotese om et fast antal punkter. Denne test har en simuleret styrke på 0,01

3 Notation og redskaber

I dette afsnit kommer vi med relevante definitioner, der er nødvendige for at gå i dybden med topologisk dataanalyse og tessellationer. I Afsnit 3.1 definerer vi blandt andet, hvad en Poissonprocess er, hvad et simplicial complex er, hvad Betti tal er samt hvad en filtering er. Ud fra disse definitioner kan vi nu formulere Sætning 4.1, som er en af hovedsætningerne, vi vil vise.

I Afsnit 3.2 definerer vi hvad Voronoi tessellation er, hvilket giver os mulighed for at formulere Sætning 4.2, som er den anden hovedsætning, vi vil vise.

3.1 Rumlig statistik og topologisk dataanalyse

Poissonprocess i \mathbb{R}^N

Vi kommer i de resultater, der bliver vist i dette bachelorprojekt, til at bruge antagelsen, at vores punkter kommer fra en Poissonprocess. Det er en relevant situation at betragte, da antagelsen om, at vores punkter kommer fra en Poissonprocess i \mathbb{R}^N betyder, at punkterne er uafhængige af hinanden, og at sandsynligheden for at finde et punkt i et givet rumligt område er proportional med størrelsen af dette område, som er antagelserne, der ofte giver god mening at antage.

Først skal vi bruge nogle indledende begreber, hvor definitioner tager udgangspunkt i [3] og [4].

Definition 3.1 (Lokalt endeligt mål). Et Borel mål er et *lokalt endeligt mål*, hvis der for ethvert punkt $p \in \mathbb{R}^N$ findes en åben omegn N_p om p sådan at μ -målet af N_p er endeligt.

Definition 3.2 (Punktprocess). Lad N_{lf} betegne rummet af lokale endelige mængder af punkter $\varphi \subseteq \mathbb{R}^N$ og udstyr den med den mindste σ -algebra, \mathcal{B} , så afbildningerne

$$\varphi \mapsto \varphi(A) := \#(\varphi \cap A)$$

er målelige for alle Borel mængder $A \subseteq \mathbb{R}^N$. Den tilhørende stokastiske variabel $X = \{X_i\}_{i \geq 1}$ med værdier i N_{lf} hedder en *punktprocess*.

Til at definere, hvad en Poissonprocess er, får vi også brug for følgende definition

Definition 3.3 (Intensitetsmål). For enhver punktprocess \mathcal{P} på \mathbb{R}^N definerer vi *intensitetsmålet* $\Lambda : \mathcal{B} \rightarrow [0, \infty]$ som

$$\Lambda(B) = \mathbb{E}[\mathcal{P}(B)], \quad B \in \mathcal{B}.$$

Definition 3.4 (Ikke atomisk mål). Et *ikke atomisk mål* μ er et mål, hvor der for ethvert måleligt sæt A med $\mu(A) > 0$ eksisterer en delmængde B af A sådan, at

$$\mu(A) > \mu(B) > 0.$$

Vi kan nu definere, hvad en Poissonprocess er

Definition 3.5 (Poissonprocess). Lad Λ være et lokalt endeligt ikke-atomisk Borel-mål på \mathbb{R}^N . En *Poissonprocess* \mathcal{P} med intensitetsmål Λ er en punktprocess, som opfylder følgende

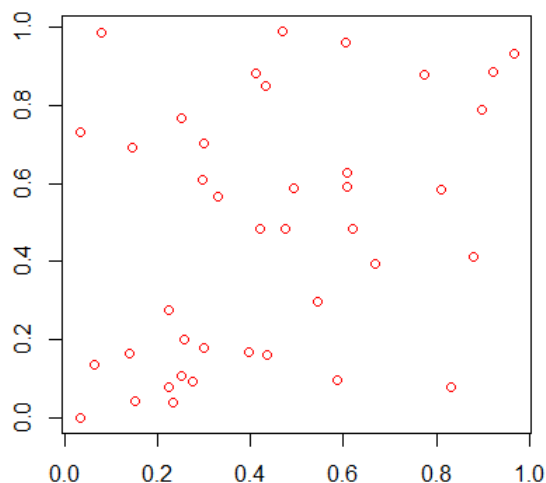
1. $\mathcal{P}(A) \sim \text{Poi}(\Lambda(A))$ for enhver mængde $A \subseteq \mathbb{R}^N$,
2. De stokastiske variable $\mathcal{P}(A_1), \dots, \mathcal{P}(A_n(x))$ er uafhængige for alle parvis disjunkte $A_1, \dots, A_n(x) \subseteq \mathbb{R}^N$.

Ofte har vi brug for kun at betragte en Poissonprocess på en mængde og ikke på hele \mathbb{R}^N , så derfor er følgende definition naturlig

Definition 3.6 (Restringerede Poissonprocess). For en Poissonprocess \mathcal{P} definerer vi for enhver mængde $A \subseteq \mathbb{R}^N$ den *restringerede Poissonprocess* \mathcal{P}_A som Poissonprocessen defineret på A .

En speciel klasse af Poissonprocesser er homogene Poissonprocesser, hvor det forventede antal af punkter i en mængde kun afhænger af volumen af mængden.

Definition 3.7 (Homogen Poissonprocess). En poissonprocess, hvor intensitetsmålet $\Lambda = \nu\lambda$, hvor ν er Lebesgue målet og λ er en konstant, kaldes en *homogen Poissonprocess*.



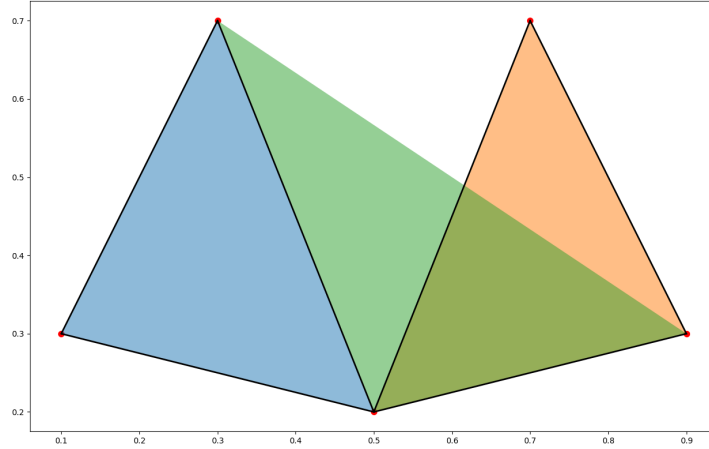
Figur 1: *Simulering af 40 punkter fra en homogen Poissonprocess i $[0, 1] \times [0, 1]$.*

Simplicial complex

Simplicial complexes er et geometrisk objekt, som er opbygget af forening af punkter, kanter, trekanter, tretrader og højere dimensionelle polytopper. Formelt bruger vi følgende definition

Definition 3.8 (Simplicial complex). Et *Simplicial complex* er en samling af K ikke tomme delmængde af en mængde K_0 , så der gælder, at $\{v\} \in K$ for alle $v \in K_0$, og hvis $\tau \subset \sigma$ og $\sigma \in K$, så er $\tau \in K$. Elementerne af K_0 kalder vi *hjørnerne* af K , og elementerne af

K kalder vi *simplicies*. Endvidere siger vi, at et simplex har *dimension* q eller er et q -*simplex*, hvis den har kardinalitet $q + 1$. Vi bruger K_q til at betegne samlingen af q -simplicies.



Figur 2: Et eksempel på et simplicial complex.

Et vigtigt eksempel på et simplicial complex er, at givet en endelig mængde af punkter K_0 og $\epsilon > 0$ kan vi konstruere *Čech complex* $\check{C}_\epsilon(K_0)$ ved at tage elementerne af K_0 som hjørnerne i $\check{C}_\epsilon(K_0)$ og for ethvert $\sigma \subset K_0$ lader vi $\sigma \in \check{C}_\epsilon(K_0)$, hvis ϵ -kuglen centreret i punkterne i σ har en ikke tom snitmængde, altså at

$$\sigma = \{x_0, \dots, x_q\} \in \check{C}_\epsilon(K_0) \iff \bigcap_{i=0}^q \overline{B}_t(x_i) \neq \emptyset,$$

hvor $\overline{B}_t(x_i)$ betegner kuglen med centrum x_i og radius t .

Homologi

Ideen bag homologi er, at forstå den topologiske struktur af ens datasæt på en måde, som er invariant under rotation, skalering med mere. Specielt vil vi fokusere på Betti tal og senere persistent Betti tal, som intuitivt siger noget om antallet af q -dimensionelle huller i et simplicial complex K .

Vi betragter et q -simplex $\sigma = \{i_0, \dots, i_q\}$ som en sorteret mængde. For $q > 0$ definerer vi en ækvivalensrelation $i_{j_0}, \dots, i_{l_0} \sim i_{l_q}, \dots, i_{j_q}$ på to sorteringer af σ med lige permutationer. Lad $\langle \sigma \rangle = \langle i_{j_0}, \dots, i_{j_q} \rangle$ være ækvivalensklassen hvor $i_{j_0} < \dots < i_{j_q}$.

Vi betragter legemet \mathbb{F}_2 med basis givet af q -simplicies'ene af K , da vi ikke har brug for orienteringen af vores simplex. Vi kan så konstruere et \mathbb{F}_2 -vektorrum $C_q(K)$ som

$$C_q(K) = \text{Span}_{\mathbb{F}_2} \{ \langle \sigma \rangle \mid \sigma \in K_q \},$$

for $K_q \neq \emptyset$ og $C_q(K) = 0$ for $K_q = \emptyset$. Her betegner $\text{Span}_{\mathbb{F}_2}(A)$ for en mængde A et vektorrum over \mathbb{F}_2 , så elementerne af A former en basis af vektorrummet. Vi definerer boundary map $\partial_q : C_q(K) \rightarrow C_{q-1}(K)$ som

$$\partial_q(\langle i_0, \dots, i_q \rangle) = \sum_{l=0}^q \langle i_0, \dots, \tilde{i}_l, \dots, i_q \rangle, \quad (1)$$

hvor \tilde{i}_l betyder, at i_l er fjernet. Vi sætter $Z_q(K) = \text{kernel } \partial_q$ og $B_q(K) = \text{image } \partial_{q+1}$. Vi kan betragte boundary map som en matrix

$$M_q = (M_{\sigma, \tau})_{\sigma \in K_{q-1}, \tau \in K_q},$$

hvor koefficienterne er givet ved (1).

For at vores definition af Betti tal, som kommer under, giver mening, har vi brug for følgende sætning.

Sætning 3.9. *For ethvert $q \geq 0$ har vi, at*

$$B_q(K) \subseteq Z_q(K)$$

Bevis. Hvis vi viser, at $\partial_q \circ \partial_{q+1}(\sigma) = 0$ for ethvert $\sigma = \langle i_0, \dots, i_{q+1} \rangle \in K_{q+1}$ så er vi færdige. Så hvis $q = 0$ har vi, at $\partial_q \circ \partial_{q+1}(\sigma) = 0$ per definition. Så antag, at $q > 0$. Så har vi, at

$$\begin{aligned} \partial_q \partial_{q+1}(\sigma) &= \partial_q \sum_{l=0}^{q+1} \langle i_0, \dots, \tilde{i}_l, \dots, i_{q+1} \rangle \\ &= \sum_{l=0}^{q+1} \partial_q \langle i_0, \dots, \tilde{i}_l, \dots, i_{q+1} \rangle \\ &= \sum_{l=0}^{q+1} \left[\sum_{j=0}^{l-1} \langle i_0, \dots, \tilde{i}_j, \dots, \tilde{i}_l, \dots, i_{q+1} \rangle + \sum_{j=l+1}^{q+1} \langle i_0, \dots, \tilde{i}_l, \dots, \tilde{i}_j, \dots, i_{q+1} \rangle \right] \\ &= 0. \end{aligned}$$

□

Vi kan nu definere Betti tal.

Definition 3.10 (Betti tal). For $q \in \mathbb{N}_0$ definerer vi den q 'ende homologi af et simplicial complex K som kvotient vektorrummet

$$H_q(K) := \text{kernel}(\partial_q) / \text{image}(\partial_{q+1}).$$

Dimensionen

$$\beta_q(K) := \dim H_q(K) = \dim \text{kernel}(\partial_q) - \dim \text{image}(\partial_{q+1})$$

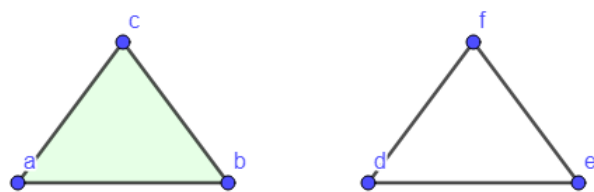
kalder vi det q 'ende *Betti tal* af K .

Vi har, at $\beta_0(K)$ geometrisk kan forstås som antallet af sammenhængende komponenter og $\beta_1(K)$ kan forstås som 1-dimensionelle huller.

Eksempel 3.11 (Beregning af Betti tal). Vi vil udregne Betti tal for følgende to simplicial complexes

$$K = \{\{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$$

$$K' = \{\{d\}, \{e\}, \{f\}, \{d, e\}, \{d, f\}, \{e, f\}\}$$



Figur 3: *Geometrisk realisation af henholdsvis K og K' . Trekanten til venstre er farvet, da vi har $\{a, b, c\}$ med i vores simplicial complex, mens trekanten til højre ikke er farvet, da $\{d, e, f\}$ ikke er med i vores simplicial complex.*

Så jeg starter med at udregne Betti tal for K . Vi har følgende følge af vektorrum og lineære afbildninger:

$$0 \longrightarrow \mathbb{F}_2 \xrightarrow{\partial_2} \mathbb{F}_2^3 \xrightarrow{\partial_1} \mathbb{F}_2^3 \xrightarrow{\partial_0} 0.$$

Hvis vi sorterer baserne af vektorrummet leksikografisk så har vi, at matrix-repræsentationen af ∂_2 og ∂_1 er givet ved

$$M_2 = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}^T$$

$$M_1 = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix},$$

samt at kernen af ∂_0 er $\{\{a\}, \{b\}, \{c\}\}$. Så vi bemærker, at rækkerummet af ∂_1 er

$$\left\{ \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T, \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T \right\}.$$

Dermed har vi, at

$$\beta_0(K) = \dim \ker(\partial_0) - \dim \text{image}(\partial_1) = 3 - 2 = 1.$$

Vi har, at dimensionen af billedet af M_2 er 1, samt at kernen af M_1 er nulvektoren, så vi har, at

$$\beta_1(K) = \dim \ker(\partial_1) - \dim \text{image}(\partial_2) = 1 - 1 = 0.$$

Og at alle højere Betti tal er 0.

Vi vil nu udregne Betti tal for K' . Vi har her følgende følge af vektorrum og lineære afbildninger:

$$0 \longrightarrow \mathbb{F}_2^3 \xrightarrow{\partial_1} \mathbb{F}_2^3 \xrightarrow{\partial_0} 0.$$

Eneste forskel ved udregning af Betti tal for K' er, at $\dim \text{image}(\partial_2) = 0$ her, altså har vi, at

$$\beta_0(K) = \dim \ker(\partial_0) - \dim \text{image}(\partial_1) = 3 - 2 = 1$$

$$\beta_1(K) = \dim \ker(\partial_1) - \dim \text{image}(\partial_2) = 1 - 0 = 1,$$

og alle højere Betti tal er 0.

Filtering

En filtering er en voksende følge af simplicial komplekser, som giver os et værktøj til at undersøge strukturen af vores data.

Lad $\mathcal{F}(\mathbb{R}^N)$ betegne alle endelige (ikke tomme) delmængder af \mathbb{R}^N . For en funktion f på $\mathcal{F}(\mathbb{R}^N)$ eksisterer der en permutations invariant funktion \tilde{f}_k på $(\mathbb{R}^N)^k$ for ethvert $k \geq 1$ sådan, at $f(\{x_1, \dots, x_k\}) = \tilde{f}_k(x_1, \dots, x_k)$. Vi siger, at hvis f er målelig, så er \tilde{f}_k målelig på $(\mathbb{R}^N)^k$ for ethvert $k \geq 1$.

Lad $\kappa : \mathcal{F}(\mathbb{R}^N) \rightarrow [0, \infty]$ være en målelig funktion, som opfylder følgende:

(K1) $0 \leq \kappa(\sigma) \leq \kappa(\tau)$, hvis σ er en delmængde af τ ,

(K2) κ er translations-invariant, altså at $\kappa(\sigma + x) = \kappa(\sigma)$ for alle $x \in \mathbb{R}^N$, hvor $\sigma + x := \{y + x : y \in \sigma\}$,

(K3) Der er en voksende funktion $\rho : [0, \infty] \rightarrow [0, \infty]$ med $\rho(t) < \infty$ for $t < \infty$ sådan at

$$\|x - y\| \leq \rho(\kappa(\{x, y\})),$$

hvor $\|\cdot\|$ er den euklidiske norm på \mathbb{R}^N .

Et vigtigt eksempel af κ er

$$\kappa_C(\{x_0, \dots, x_q\}) = \inf_{w \in \mathbb{R}^N} \max_{0 \leq i \leq q} \|x_i - w\|$$

som definerer Čech filteret $\mathbb{C}(\varphi) = \{C(\varphi, t)\}_{t \geq 0}$. Det inducerer Čech complexet, som vi definerede tidligere.

Givet sådan en funktion κ konstruerer vi en filtering $\mathbb{K}(\Xi) = \{K(\Xi, t) : 0 \leq t < \infty\}$ af simplicial complexes fra en endelig punkt konfiguration $\Xi \in \mathcal{F}(\mathbb{R}^N)$ med

$$K(\Xi, t) = \{\sigma \subset \Xi : \kappa(\sigma) \leq t\},$$

altså at $\kappa(\sigma)$ er fødselstiden af simplex σ i filteringen $\mathbb{K}(\Xi)$.

Definition 3.12 (Persistent Betti tal). For en filtering \mathbb{K} er det (r, s) 'te *persistent Betti tal* defineret som

$$\beta_q^{r,s}(\mathbb{K}) = \dim \frac{Z_q(K_r)}{Z_q(K_r) \cap B_q(K_s)}, \quad (r \leq s),$$

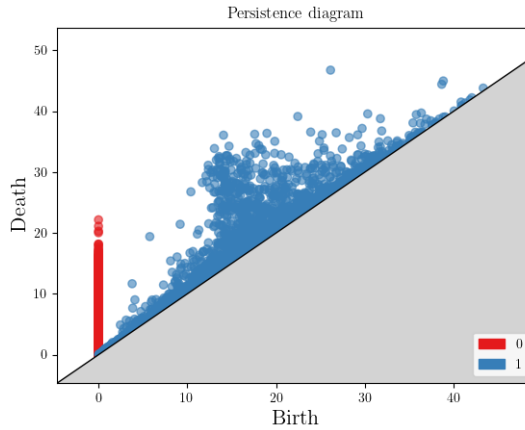
hvor $Z_q(K_r)$ og $B_q(K_r)$ er henholdsvis den q 'te cycle gruppe og q 'te boundary gruppe.

Persistent Betti tal kan fortælle os om antallet af q -dimensionelle huller i \mathbb{K} , som eksisterer mellem r og s .

For filteringen $\mathbb{K}(\Xi)$ betegner vi det q -te persistent diagram med

$$D_q(\Xi) = \{(b_i, d_i) \in \Delta : i = 1, \dots, n_q\},$$

hvor $\Delta = \{(x, y) \in \overline{\mathbb{R}}^2 : 0 \leq x < y \leq \infty\}$ er bestemt af den unikke nedbrydning af persistent homologi. Det vil sige, at der findes punkter $\{(b_i, d_i) \in \Delta\}$ med $\beta_q^{r,s}(\mathbb{K}) = \#\{i : b_i \leq r, d_i > s\}$. Altså betegner b_i fødslen og d_i hvornår det forsvinder igen. Vi sætter $d_i = \infty$ hvis den i 'te homology klasse overlever for evigt. Det betyder, at det (r, s) -persistent betti tal $\beta_q^{r,s}(\mathbb{K})$ tæller antallet af fødsels-døds par i et persistence diagram $D_q(K)$ med fødsel før r og døden efter s .



Figur 4: *Et eksempel på et persistence diagram.*

3.2 Tessellations

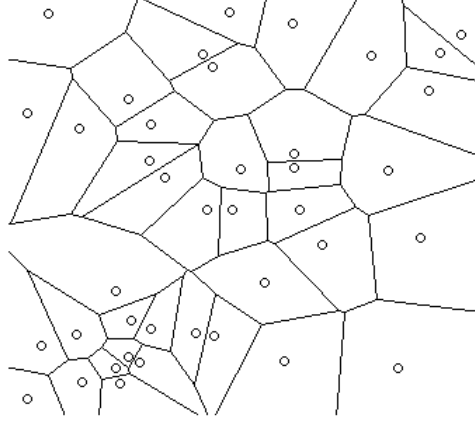
Tessellations opdeler et rum med punkter i forskellige celler baseret på nogle kriterier. En meget anvendt tessellation er Voronoi tessellation, som laver en celle for hvert punkt med det tilhørende område, der er tættere på dette punkt end på noget andet punkt. Dette skaber en partition af rummet i celler, som kaldes Voronoi celler. Et eksempel på en Voronoi tessellation kan ses i Figur 5. Vi bruger følgende definition

Definition 3.13 (Voronoi tessellations). Lad φ være en lokal endelig delmængde af \mathbb{R}^N .

Voronoi cellen $C(x, \varphi)$ for et $x \in \varphi$ er defineret som

$$C(x, \varphi) = \{y \in \mathbb{R}^N : \|y - x\| \leq \|y - x'\| \text{ for alle } x' \text{ i } \varphi.\}$$

Voronoi tessellationen af φ er mængden $V(\varphi) = \{C(x, \varphi) : x \in \varphi\}$.



Figur 5: Punkterne fra Figur 1 med Voronoi tessellation indtegnet.

For at beregne persistent Betti-tal vil vi bruge en tessellations-tilpasset filtrering, hvor hele simplicies tilføjes på filtreringstidspunkter, der er givet af deres omskrevne cirkels radiuser. Med andre ord tilhører et q -simplex $f \in \Xi_n^q$, defineret af punkterne P_0, \dots, P_m , filtreringen på niveau $t > 0$, hvis og kun hvis dens omskrevne cirkels radius, $r_C(f)$, opfylder $r_C(f) \leq t$, hvor $r_C(f)$ defineres som

$$r_C(f) := \min_{y \in \mathbb{R}^N} \max_{i \leq m} |y - P_i|.$$

Vi vil betegne dette som den tessellations-tilpassede filtrering, der er forbundet med fladerne i tessellationen. Lad \mathbb{K}_{Vor} betegne den Voronoi tessellation-tilpassede filtrering. Bemærk, at \mathbb{K}_{Vor} ikke opfylder (K3).

Dette minder meget om Čech-filtreringen, hvor forskellen er, hvornår simplicies bliver tilføjet til simplicial komplekset. I Čech-filtreringen tilføjes simplicies ud fra parvis afstand mellem punkter.

4 Centrale grænseværdisætning for persistent Betti tal

Vi vil i dette afsnit formulere centrale grænseværdisætninger for en Poissonprocess og i tilfældet med Voronoi tessellations. Man kan ved at bruge eksistensen af følgende centrale grænseværdisætninger, teste om ens punkter kommer fra en homogen Poissonprocess med enhedsintensitet.

Vi angiver her de nødvendige egenskaber for funktioneller til at opnå en version af den centrale grænseværdisætning. Lad $\{W_n\}$ være Borel delmængder af \mathbb{R}^N , som opfylder følgende betingelser:

$$(A1) \quad |W_n| = n \text{ for alle } n \in \mathbb{N}.$$

$$(A2) \quad \bigcup_{n \geq 1} \bigcap_{m \geq n} W_m = \mathbb{R}^N$$

$$(A3) \quad \lim_{n \rightarrow \infty} \|(\partial W_n)^{(r)}\|/n = 0 \text{ for alle } r > 0, \text{ hvor } A^{(r)} \text{ er mængden af de punkter, som er højest } r \text{ væk fra } A.$$

$$(A4) \quad \text{Der eksisterer en konstant } \gamma > 0, \text{ så } \sup_{x,y \in W_n} |x - y| \leq \gamma n^\gamma.$$

Givet sådan en følge, sæt $\mathcal{W} = \mathcal{W}(\{W_n\})$ være samlingen af alle delmængder A i \mathbb{R}^N på formen $A = W_n + x$ for et W_n i følgen og et punkt $x \in \mathbb{R}^N$.

Lad H være en functional, som tager reelle værdier i $\mathcal{F}(\mathbb{R}^N)$. Vi siger, at H er *translationsinvariant*, hvis $H(\mathcal{X} + y) = H(\mathcal{X})$ for alle $\mathcal{X} \in \mathcal{F}(\mathbb{R}^N)$ og $y \in \mathbb{R}^N$. Lad D_0 betegne den forøgelse i H , som fås ved at tilføje $\{0\}$

$$D_0 H(\mathcal{X}) = H(\mathcal{X} \cup \{0\}) - H(\mathcal{X}), \quad \mathcal{X} \in \mathcal{F}(\mathbb{R}^N).$$

Vi siger funktionallen H er *svagt stabiliseret* på \mathcal{W} , hvis der eksisterer en stokastisk variabel $D(\infty)$, så $D_0 H(\mathcal{P}_{A_n(x)}) \xrightarrow{n.s.} D(\infty)$ for $n \rightarrow \infty$ for enhver følge $\{A_n(x) \in \mathcal{W}\}_{n \geq 1}$ som går mod \mathbb{R}^N . Poisson begrænset moment kriterium på \mathcal{W} er givet ved

$$\sup_{0 \in A \in \mathcal{W}} \mathbb{E}[(D_0 H(\mathcal{P}_A))^4] < \infty.$$

Vi kan nu formulere og bevise nogle centrale grænseværdisætninger.

4.1 Poissonprocess

Vi vil vise følgende centrale grænseværdisætning for en Poissonprocess

Sætning 4.1. *Lad \mathcal{P} være en homogen Poisson punkt process med enhedsintensitet. Antag, at følgen $\{W_n\}$ overholder betingelserne (A1)-(A4). Så for ethvert $0 \leq r \leq s < \infty$ gælder der*

$$\frac{\beta_q^{r,s}(\mathbb{K}(\mathcal{P}_{W_n})) - \mathbb{E}[\beta_q^{r,s}(\mathbb{K}(\mathcal{P}_{W_n}))]}{n^{1/2}} \xrightarrow{d} N(0, \sigma_{r,s}^2) \quad \text{for } n \rightarrow \infty.$$

Det kan vises, at $\sigma_{r,s}^2 > 0$ og Proposition 9.1 i [1] indeholder nogle ideer til, hvordan man kan vise dette.

4.2 Voronoi tessellation

Vi vil vise følgende centrale grænseværdisætning for en Poissonprocess med en Voronoi tessellation. Vi vil i forbindelse med Voronoi tessellation altid betragte vinduet $W_n = [-\frac{n}{2}, \frac{n}{2}]^N$. Vi kan ikke anvende Sætning 4.1, da Voronoi tessellation-tilpassede filtering ikke opfylder (K3).

Sætning 4.2. *Lad \mathcal{P} være en homogen Poisson punkt process med enhedsintensitet. Så for ethvert $0 \leq r \leq s < \infty$ gælder der*

$$\frac{\beta_q^{r,s}(\mathbb{K}_{\text{Vor}}(\mathcal{P}_{W_n})) - \mathbb{E}[\beta_q^{r,s}(\mathbb{K}_{\text{Vor}}(\mathcal{P}_{W_n}))]}{n^{1/2}} \xrightarrow{d} N(0, \sigma_{r,s}^2) \quad \text{for } n \rightarrow \infty.$$

5 Beviser

5.1 Poissonprocess

Beviserne i dette afsnit er primært baseret på [3].

Lemma 5.1. *Lad H være en funktional som antager reelle værdier defineret på $\mathcal{F}(\mathbb{R}^N)$. Antag at H er translations-invariant og svagt stabiliserende på \mathcal{W} og opfylder Poisson begrænset moment betingelse. Så eksisterer der en konstant $\sigma^2 \in [0, \infty)$ så $n^{-1} \mathbb{V}[H(\mathcal{P}_{W_n})] \rightarrow \sigma^2$ og*

$$\frac{H(\mathcal{P}_{W_n}) - \mathbb{E}[H(\mathcal{P}_{W_n})]}{n^{1/2}} \xrightarrow{d} N(0, \sigma^2) \quad \text{for } n \rightarrow \infty.$$

Bevis. Se lemma 3.1 i [7].

Lemma 5.2. Lad $\mathbb{K} = \{K_t\}_{t \geq 0}$ og $\tilde{\mathbb{K}} = \{\tilde{K}_t\}_{t \geq 0}$ være filteringer med $K_t \subset \tilde{K}_t$ for $t \geq 0$. Så gælder der, at

$$|\beta_q^{r,s}(\tilde{\mathbb{K}}) - \beta_q^{r,s}(\mathbb{K})| \leq \sum_{j=q,q+1} (|\tilde{K}_{s,j} \setminus K_{s,j}| + |\{\sigma \in K_{s,j} \setminus K_{r,j} : \tilde{t}_\sigma \leq r\}|),$$

hvor $\tilde{K}_{t,j}$ (eller $K_{t,j}$) er mængden af j -simplicies i \tilde{K}_t (eller K_t) og \tilde{t}_σ (eller t_σ) er fødselstiden af σ i filtrationen $\tilde{\mathbb{K}}$ (eller \mathbb{K}).

Bevis. Se Lemma 2.11 i [3].

Korollar 5.3. Funktionalen $\beta_q^{r,s}(\mathbb{K})$ er svagt stabiliserende.

Bevis for sætning 4.1. For et fast $r \leq s$ betragter vi det persistente Betti tal $\beta_q^{r,s}(\mathbb{K}(\cdot))$ som et funktional på $\mathcal{F}(\mathbb{R}^N)$ og tjekker de 3 betingelser givet i Lemma 5.1. Vi har, at det er translationsinvariant, da κ er translationsinvariant fra (K2). Korollar 5.3 giver os, at det er svagt stabiliserende, så det eneste vi mangler er, at vise begrænset Poisson begrænset moment kriterium på \mathcal{W} . Lad $F_q(\Phi, r)$ være antallet af q -simplicies i $K(\Phi, r)$. Vi har så følgende

$$\begin{aligned} |D_0 \beta_q^{r,s}(\mathbb{K}(\mathcal{P}_A))| &= |\beta_q^{r,s}(\mathbb{K}(\mathcal{P}_A \cup \{0\})) - \beta_q^{r,s}(\mathbb{K}(\mathcal{P}_A))| \\ &\leq \sum_{j=q,q+1} |K_j(\mathcal{P}_A \cup \{0\}, s) \setminus K_j(\mathcal{P}_A, s)| \\ &\leq \sum_{j=q,q+1} F_j(\mathcal{P}_{\bar{B}_{\rho(s)}(0)}, s) \\ &\leq \sum_{j=q+1,q+2} \mathcal{P}(\bar{B}_{\rho(s)}(0))^j \end{aligned}$$

Hvor vi i første ulighed brugte Lemma 5.2, i anden ulighed brugte vi betingelsen (K3), som giver os, at afstanden er begrænset af kuglen med radius $\rho(s)$. Sidste ulighed har vi fra, at vi kan begrænse antallet af q -simplicies med antallet af punkter opløftet til $q + 1$. Vi kan nu bemærke, at

$$\begin{aligned} \mathbb{E} [|D_0 \beta_q^{r,s}(\mathbb{K}(\mathcal{P}_A))|^4] &\leq \mathbb{E} \left[\left(\mathcal{P}(\bar{B}_{\rho(s)}(0))^{q+1} + \mathcal{P}(\bar{B}_{\rho(s)}(0))^{|q+2} \right)^4 \right] \\ &\leq \mathbb{E} \left[\left(2 \cdot \max \left\{ |\mathcal{P}_{\bar{B}_{\rho(s)}(0)}|^{q+1}, |\mathcal{P}_{\bar{B}_{\rho(s)}(0)}|^{q+2} \right\} \right)^4 \right] \\ &\leq \mathbb{E} \left[2^4 \cdot \max \left\{ |\mathcal{P}_{\bar{B}_{\rho(s)}(0)}|^{4q+4}, |\mathcal{P}_{\bar{B}_{\rho(s)}(0)}|^{4q+8} \right\} \right] \\ &\leq 2^4 \mathbb{E} \left[|\mathcal{P}_{\bar{B}_{\rho(s)}(0)}|^{4q+4} \right] + 2^4 \mathbb{E} \left[|\mathcal{P}_{\bar{B}_{\rho(s)}(0)}|^{4q+8} \right]. \end{aligned}$$

Da $\mathcal{P}(\bar{B}_{\rho(s)}(0))$ er Poisson-fordelt har vi, at alle momenter er begrænset og dermed har vi, at overstående udtryk er endeligt. Da det endelige udtryk ikke afhænger af A har vi altså, at

$$\sup_{0 \in A \in \mathcal{W}} \mathbb{E}[D_0 \beta_q^{r,s}(\mathbb{K}(\mathcal{P}_A))^4] < \infty,$$

som ønsket. □

Lemma 5.4. *Lad $P \subseteq \mathbb{R}^N$ være en lokal endelig mængde. For ethvert fast $r \leq s$ eksisterer der D_∞ og $R > 0$ så*

$$D_0 \beta_q^{r,s}(\mathbb{K}(P_{\bar{B}_a(0)})) = D_\infty$$

for alle $a \geq R$.

Til at bevise Lemma 5.4 kommer vi til at skulle bruge følgende to dimensions-lemmaer.

Lemma 5.5. *Lad A, B, U, V være underrum af et vektorrum, hvor der gælder at, $A \subset U$ og $B \subset V$. Så har vi*

$$\dim \frac{U \cap V}{A \cap B} \leq \dim \frac{U}{A} + \dim \frac{V}{B}.$$

Bevis. Da vi har formelen $\dim(U \cap V) + \dim(U \cup V) = \dim U + \dim V$ samt at $\dim(U/A) = \dim U - \dim A$ så har vi, at

$$\dim \frac{U \cap V}{A \cap B} = \dim \frac{U}{A} + \dim \frac{V}{B} + (\dim U \cup V - \dim A \cup B) \leq \dim \frac{U}{A} + \dim \frac{V}{B},$$

som ønsket. □

Lemma 5.6. *Lad $D = \begin{bmatrix} A & B \end{bmatrix}$ være en matrice med delmatricer A og B . Lad l være antallet af søjler i B . Så har vi*

$$\dim \ker D \leq \dim \ker A + l$$

Bevis. Lad $B = \begin{bmatrix} b_1 & \dots & b_l \end{bmatrix}$, hvor b_i er den i 'te søjle vektor af B og sæt $D^{(i)} = \begin{bmatrix} A & b_1 & \dots & b_i \end{bmatrix}$. Så har vi for ethvert i , at

$$\dim \ker D^{(i)} \leq \dim \ker D^{(i-1)} + 1.$$

Ved at anvende dette iterativt får vi det ønskede. □

Vi vil bruge, at vi næsten sikkert har, at en homogen Poissonprocess \mathcal{P} har uendelige punkter i \mathbb{R}^N , som ikke har akkumulerings-punkter.

Bevis for Lemma 5.4 Sæt $P' = P \cup \{0\}$. Lad $K_{r,a} = K(P_{\bar{B}_a(0)}, r)$ være det simplicial complex defineret på $P_{\bar{B}_a(0)}$ med parameter r og tilsvarende sæt $K'_{r,a} = K(P'_{\bar{B}_a(0)}, r)$. Ved at bruge Definition 3.12 kan vi skrive $D_0\beta_q^{r,s}(\mathbb{K}(P_{\bar{B}_a(0)}))$ som

$$\begin{aligned} D_0\beta_q^{r,s}(\mathbb{K}(P_{\bar{B}_a(0)})) &= \dim \frac{Z_q(K'_{r,a})}{Z_q(K'_{r,a} \cap B_q(K'_{s,a}))} - \dim \frac{Z_q(K_{r,a})}{Z_q(K_{r,a} \cap B_q(K_{s,a}))} \\ &= (\dim Z_q(K'_{r,a}) - \dim Z_q(K_{r,a})) \\ &\quad - \dim(Z_q(K'_{r,a}) \cap B_q(K'_{s,a})) + \dim(Z_q(K_{r,a}) \cap B_q(K_{s,a})) \end{aligned}$$

Så vi kan betragte stabiliteten mht. a for $\dim Z_q(K_{r,a})$ og $\dim Z_q(K_{r,a}) \cap B_q(K_{s,a})$. Så lad os starte med $\dim Z_q(K_{r,a})$. Da \dim kun tager ikke negative værdier er det nok at vise, at den er begrænset og ikke-aftagende for at grænsen eksisterer. Vi har, at $K_{r,a} \subseteq K'_{r,a}$ og derfor har vi, at $Z_q(K_{r,a}) \subseteq Z_q(K'_{r,a})$. Vi skriver nu $K'_{r,a}$ som en disjunkt forening $K'_{r,a} = K_{r,a} \sqcup K_{r,a}^0$, hvor $K_{r,a}^0$ er mængden af simplices, der har punktet 0 og lad $K_{r,a,q}^0$ være mængden af simplices med orden q , så $K_{r,a,q}^0 = \{\sigma \in (K'_{r,a})_q : 0 \in \sigma\}$.

Lad $\partial_{q,a}$ og $\partial'_{q,a}$ være det q 'te boundary afbildning på henholdsvis $K_{r,a}$ og $K'_{r,a}$. Så får vi følgende blok matrice

$$\partial'_{q,a} = \begin{bmatrix} M_{1,\rho} & \mathbf{0} \\ M_{2,\rho} & \partial_{q,a} \end{bmatrix},$$

hvor de første kolonner og søjler er ordnet efter simplices i $K_{r,a,q}^0$ og $K_{r,a,q-1}$ og i anden kolonne og søjle svarer til simplices i $K_{r,a}$.

For ethvert simplex $\sigma \in K(P, r)$ som indeholder punktet 0 har vi, at det er inkluderet i $\bar{B}_{\rho(r)}(0)$. Derfor er mængden $K_{r,a,q}^0$ uafhængig af a for $a \geq \rho(r)$, som vi betegner med $K_{r,* ,q}^0$. Nu kan vi så bruge Lemma 5.6 på matrix formen givet ovenover, hvor vi sætter $D = \partial'_{q,a}$,

$$A = \begin{bmatrix} \mathbf{0} \\ \partial_{q,a} \end{bmatrix} \text{ og } B = \begin{bmatrix} M_{1,\rho} \\ M_{2,\rho} \end{bmatrix}. \text{ Så har vi}$$

$$\dim Z_q(K'_{r,a}) - \dim Z_q(K_{r,a}) \leq |K_{r,a,q}^0| = |K_{r,* ,q}^0|,$$

for $a \geq \rho(r)$ og det er dermed begrænset for ethvert fast r .

Vi vil nu vise, at den er ikke aftagende. Så vi definerer en homomorfi givet ved

$$f : \frac{Z_q(K'_{r,a_1})}{Z_q(K_{r,a_1})} \ni [c] \mapsto [c] \in \frac{Z_q(K'_{r,a_2})}{Z_q(K_{r,a_2})}$$

for $a_1 \leq a_2$. Den her afbildning er veldefineret, da $Z_q(K_{r,a_1}) \subseteq Z_q(K_{r,a_2})$ og tilsvarende at $Z_q(K'_{r,a_1}) \subseteq Z_q(K'_{r,a_2})$. Vi har at kernen af f er trivial, da hvis $f[c] = 0$ så er $c \in Z_q(K_{r,a_2})$ og da $a_1 \leq a_2$ har vi også $c \in Z_q(K_{r,a_1})$. Da kernen er trivial så er f injektiv. Dermed har vi uligheden

$$\dim Z_q(K'_{r,a_1})/Z_q(K_{r,a_1}) \leq \dim Z_q(K'_{r,a_2})/Z_q(K_{r,a_2}),$$

hvilket viser at den er ikke aftagende. Altså har vi nu vist stabiliteten af $\dim Z_q(K_{r,a})$.

Vi vil nu vise stabiliten af $\dim(Z_q(K_{r,a}) \cap B_q(K_{s,a}))$. Vi vil bruge samme bevisteknik som ovenover. Fra Lemma 5.5 har vi, at

$$\dim \frac{Z_q(K'_{r,a}) \cap B_q(K'_{s,a})}{Z_q(K_{r,a}) \cap B_q(K_{s,a})} \leq \dim \frac{Z_q(K'_{r,a})}{Z_q(K_{r,a})} + \dim \frac{B_q(K'_{s,a})}{B_q(K_{s,a})}.$$

Ligesom før har vi igen, at $|K_{s,a,q+1}^0| = |K_{s,*,q+1}^0|$ for alle a store nok. Dermed har vi, at

$$\dim Z_q(K'_{r,a}) \cap B_q(K'_{s,a}) - \dim Z_q(K_{r,a}) \cap B_q(K_{s,a}) \leq |K_{r,*,q}^0| + |K_{r,*,q+1}^0|,$$

og dermed har vi vist det er begrænset. Tilsvarende som før kan vi vise, at kernen er trivial og dermed injektiv i følgende homomorfi

$$f : \frac{Z_q(K'_{r,a_1}) \cap B_q(K'_{s,a_1})}{Z_q(K_{r,a_1}) \cap B_q(K_{s,a_1})} \ni [c] \mapsto [c] \in \frac{Z_q(K'_{r,a_2}) \cap B_q(K'_{s,a_2})}{Z_q(K_{r,a_2}) \cap B_q(K_{s,a_2})},$$

for $a_1 \leq a_2$. Dette medfører, at det er ikke aftagende som ønsket. Dette slutter beviset for lemmaet. \square

Bevis for Korollar 5.3 Lad $R > 0$ være valgt således at det opfylder Lemma 5.4 og lad $\{A_n \in \mathcal{W}\}_{n \geq 1}$ være en følge som går mod \mathbb{R}^N . Så eksistere der et $n_0 \in \mathbb{N}$ så $B_R(0) \subset A_n$ for alle $n \geq n_0$.

For $n \geq n_0$ definere vi $L_{r,n} = K(\mathcal{P}_{A_n}, r)$ være det simplicial complex defineret på \mathcal{P}_{A_n} med parameter r . Da A_n er begrænset eksistere der er $a > R$ så

$$B_R(0) \subset A_n \subset B_a(0).$$

Med samme fremgangsmåde som i Lemma 5.4, hvor vi viste injektivitet, kan vi vise, at

$$\frac{Z_q(K'_{r,R})}{Z_q(K_{r,R})} \subseteq \frac{Z_q(L'_{r,n})}{Z_q(L_{r,n})} \subseteq \frac{Z_q(K'_{r,a})}{Z_q(K_{r,a})},$$

hvor $K_{r,a} = K(\mathcal{P}_{\bar{B}_a(0)}, r)$ som tidligere. Fra vores valg af R har vi, at dimensionen af $Z_q(K'_{r,R})/Z_q(K_{r,R})$ og $Z_q(K'_{r,a})/Z_q(K_{r,a})$ er lig hinanden for $n \geq n_0$. Dermed har vi så, at

$$\dim Z_q(K'_{r,R}) - \dim Z_q(K_{r,R}) = \dim Z_q(L'_{r,n}) - \dim Z_q(L_{r,n}).$$

Vi kan på tilsvarende måde vise at $\dim Z_q(L'_{r,n}) \cap B_q(L'_{r,n}) - Z_q(L_{r,n}) \cap B_q(L_{r,n})$ er invariant for $n \geq n_0$. Vi har altså vist at det bliver konstant og dermed har vi specielt, at $\beta_q^{r,s}(\mathbb{K})$ er svagt stabiliserende, som ønsket. \square

5.2 Voronoi tessellation

Der er kun et par forskelle i beviset for Sætning 4.2 sammenlignet med Sætning 4.1. Forskellene er

- (1) Nogle af argumenterne i Lemma 5.4 som skal bruges til at vise at $B_q^{r,s}(\mathbb{K}_{\text{Vor}})$ er svagt stabiliserende, som analog til beviset for Korollar 5.3.
- (2) Vise at $B_q^{r,s}(\mathbb{K}_{\text{Vor}}(\cdot))$ opfylder Poisson begrænset moment kriterium på \mathcal{W} , som analog til beviset for Sætning 4.1.

Vi vil starte med nogle indledene bemærkninger. Sæt $C_{A,0}$ til at være Voronoi cellen for 0 i $\mathcal{P}_A \cup \{0\}$. Når cellen $C_{A,0}$ tilføjes til cellerne for \mathcal{P}_A vil de celler, som er tættest på, ændres. Så sæt $R_{A,0}$ til at være det mindste positive heltal, så alle endelige Voronoi celler, som ændres ved tilføjelsen af $C_{A,0}$, er indeholdt i $B_{R_{A,0}}(0)$. Ideen bag beviset er, at vi kan sikre os at $R_{A,0}$ ikke bliver for stor ved at lave en kasse rundt om $\{0\}$ og inddele kassen i et gitter, og vise at det kun er punkter inde i dette gitter, der bliver påvirket af tilføjelsen af $\{0\}$. Endvidere sætter vi $\Xi_{A,0}^{(0)}$ til at være Voronoi hjørner for $\mathcal{P}_A \cup \{0\}$ og $\Xi_A^{(0)}$ til at være Voronoi hjørner for \mathcal{P}_A .

Bevis for (1). I beviset for Lemma 5.4 skal nogle af argumenterne ændres. Til at begrænse $\dim Z_q(K'_{r,a}) - \dim Z_q(K_{r,a})$ kan vi gøre følgende. Lad $K_{r,a,q}^U$ betegne mængden af simplices

udenfor stabilitetsradiussen $R_{A,0}$ og $K_{r,a,q}^I$ betegne mængden af simplices indenfor stabilitetsradiussen $R_{A,0}$. Vi bemærker, at $K_{r,a,q}^U$ er uafhængig af a for $a > R_{A,0}$, som vi betegner med $K_{r,*,q}^U$. Ved at bruge Lemma 5.6 to gange har vi, at

$$\begin{aligned} \dim Z_q(K'_{r,a}) - \dim Z_q(K_{r,a}) &= \dim Z_q(K'_{r,a}) - \dim Z_q(K_{r,a}) - |K_{r,a,q}^U| + |K_{r,a,q}^U| \\ &\leq |\dim Z_q(K_{r,a}) - |K_{r,a,q}^U|| + |\dim Z_q(K'_{r,a}) - |K_{r,a,q}^U|| \\ &\leq |K_{r,a,q}^U| + |K_{r,a,q}^I| \\ &= |K_{r,*,q}^U| + |K_{r,a,q}^I|, \end{aligned}$$

og da $K_{r,a,q}^I$ er endelig, da P er en lokal endelig mængde, er det dermed begrænset for ethvert fast r .

Bevis for (2). Vi vil vise dette i tilfældet, hvor A er en kasse med centrum x , altså at A er på formen

$$A_n(x) = [-n/2, n/2]^N + x$$

Så ligesom i Sætning 4.1 skal vi vise, at

$$\sup_{0 \in A \in \mathcal{W}} \mathbb{E} \left[|K_j(\mathcal{P}_A \cup \{0\}, s) \setminus K_j(\mathcal{P}_A, s)|^4 \right]$$

for $j = q, q+1$ er begrænset.

Definer

$$E_{r,n,x} := \left\{ A_{\sqrt{r}}(\sqrt{r}z) \cap \mathcal{P}_A \neq \emptyset \text{ for alle } \sqrt{r}z \in \sqrt{r}\mathbb{Z}^N \cap A_n(x) \right\}$$

til at betegne det udfald, at for ethvert punkt $\sqrt{r}z \in \sqrt{r}\mathbb{Z}^N \cap A_n(x)$, så indeholder \sqrt{r} -kassen $A_{\sqrt{r}}(\sqrt{r}z)$ mindst et punkt fra \mathcal{P}_A . Til at vise moment betingelsen får vi brug for følgende lemma.

Lemma 5.7. ($E_{r,n,x}$ sker med stor sandsynlighed). Der eksisterer et $c_1 > 0$, så for alle $r > 1$ gælder der, at

$$\sup_{n \geq 0} \sup_{x \in \mathbb{R}^N} \mathbb{P}(E_{r,n,x}^c) \leq \exp(-c_1 r^{N/2})$$

Bevis. Vi kan skrive $E_{r,n,x}$ som

$$E_{r,n,x} = \bigcap_{\sqrt{z} \in \sqrt{r}\mathbb{Z}^N \cap A_n(x)} \{A_{\sqrt{r}}(\sqrt{r}z) \cap \mathcal{P} \neq \emptyset\}.$$

Vi kan nu bemærke, at $|\sqrt{r}\mathbb{Z}^N \cap A_n(x)| \leq (\sqrt{r} + 1)^N$, så dermed har vi, at

$$\mathbb{P}(E_{r,n,x}^c) \leq |\sqrt{r}\mathbb{Z}^N \cap A_n(x)| \mathbb{P}(A_{\sqrt{r}}(\sqrt{r}z) \cap \mathcal{P} = \emptyset) \leq C(r+1)^N \exp(-r^{N/2}),$$

hvor vi begrænsede $\mathbb{P}(A_{\sqrt{r}}(\sqrt{r}z) \cap \mathcal{P} = \emptyset)$ ved at bruge \mathcal{P} er en Poissonprocess. Vi bemærker nu, at højresiden er $\exp(-r^{N/2}(1+o(1)))$ og dermed har vi, at der findes en konstant $c_1 > 0$, så

$$\sup_{n \geq 0} \sup_{x \in Q_n} \mathbb{P}(E_{r,n,x}^c) \leq \exp(-c_1 r^{N/2}),$$

som ønsket. □

Lemma 5.8. *Der gælder for ethvert $k \geq N^2 + 9$, at*

$$E_{k/(4N),n,x} \subseteq \{R_{A,0} \leq k\},$$

Bevis. Vi vil starte med at vise, at vi for ethvert $k \geq N^2 + 9$ har, at

- (a) Under hændelsen $E_{k/(4N),n,x}$ er enhver celle med centrum i $A_{k/(4N)-(2N+2)\sqrt{k/(4N)}}(x)$ indeholdt i $A_{k/(4N)-2\sqrt{k/(4N)}}(x)$,
- (b) Under hændelsen $E_{k/(4N),n,x}(x)$ er snitmængden mellem enhver celle med centrum udenfor $A_{k/(4N)+(N-1)\sqrt{k/(4N)}}(x)$ og med $A_{k/(4N)-2\sqrt{k/(4N)}}(x)$ tom.

Bevis for (a). Lad C være en celle med centrum i $P \in A_{k/(4N)-(2N+2)\sqrt{k/(4N)}}(x)$ og betragt et vilkårligt punkt P' udenfor $A_{k/(4N)-2\sqrt{k/(4N)}}(x)$. Lad P'' være det tætteste punkt på randen af $A_{k/(4N)-2\sqrt{k/(4N)}}(x)$. Så har vi, at $|P' - P| \geq |P' - P''| + \frac{(2N+2)\sqrt{k/(4N)} - 2\sqrt{k/(4N)}}{2} = |P' - P''| + N\sqrt{k/(4N)}$. Under hændelsen $E_{k/(4N),n,x}$ er afstanden fra P' til et af centrene højst $|P' - P''| + \sqrt{Nk/(4N)} < |P' - P|$ og dermed er $P' \notin C$.

Bevis for (b). Lad C være en celle med centrum $P \in \mathbb{R}^N \setminus A_{k/(4N)+(N-1)\sqrt{k/(4N)}}(x)$ og lad P' være et vilkårligt punkt inde i $A_{k/(4N)-2\sqrt{k/(4N)}}(x)$ og lad P'' være det tætteste punkt på randen af $A_{k/(4N)-2\sqrt{k/(4N)}}$. Så har vi, at $|P' - P| \geq |P' - P''| + N\sqrt{k/(4N)}$. Men under hændelsen $E_{k/(4N),n,x}$ er distancen fra P' til et af centrene højst $|P' - P''| + \sqrt{Nk/(4N)} < |P' - P|$ og dermed er P ikke indeholdt i $A_{k/(4N)-2\sqrt{k/(4N)}}$.

Nu giver dette, at under $E_{k/(4N),n,x}$, så er en celle med centrum i $A_{k/(4N)-(2N+2)\sqrt{k/(4N)}}(x)$ indeholdt i $A_{k/(4N)-2\sqrt{k/(4N)}}(x)$, og dermed kan ingen celle med centrum udenfor $A_{k/(4N)+(N-1)\sqrt{k/(4N)}}(x)$ ændre sig ved tilføjes af $\{0\}$ til \mathcal{P}_A .

Vi kan begrænse $k/(4N) + (N-1)\sqrt{k/(4N)}$ -kassen med en kugle med radius $\sqrt{N}(k/(4N) + (N-1)\sqrt{k/(4N)})/2 = k/(4\sqrt{N}) + \sqrt{k}N/4 - \sqrt{k}/4$, altså har vi, at

$$A_{k/(4N) + (N-1)\sqrt{k/(4N)}}(x) \subseteq B_{k/(4\sqrt{N}) + \sqrt{k}N/4 - \sqrt{k}/4}(x)$$

Vi bemærker nu, at for $k \geq N^2 + 9$

$$\begin{aligned} \frac{k}{4\sqrt{N}} + \frac{\sqrt{k}N}{4} - \frac{\sqrt{k}}{4} &\leq \frac{k}{4} + \frac{\sqrt{k}N}{4} \\ &\leq \frac{k}{4} + \frac{3\sqrt{k} \cdot \sqrt{k}}{4} \\ &= k \end{aligned}$$

og dermed for ethvert $k \geq N^2 + 9$ har vi, at

$$E_{k/(4N), n, x} \subseteq \{R_{A,0} \leq \frac{k}{8\sqrt{N}} + \frac{N-1}{2} \cdot \sqrt{\frac{k}{4}}\} \subseteq \{R_{A,0} \leq k\},$$

som ønsket. □

Vi er nu klar til at bevise Sætning 4.2.

Bevis for Sætning 4.2. Ved at bruge overstående har vi, at

$$\begin{aligned} \mathbb{E}\left[|K_q(\mathcal{P}_A \cup \{0\}, s) \setminus K_q(\mathcal{P}_A, s)|^4\right] &\leq \mathbb{E}\left[|\Xi_{A,0}^{(0)}(B_{R_0,A}(0))|^{4q+4}\right] \\ &\leq \sum_{k \geq 0} \mathbb{E}\left[|\Xi_{A,0}^{(0)}(B_{k+1}(0))|^{4q+4} \cdot \mathbf{1}_{\{R_{A,0} \in (k, k+1]\}}\right] \\ &\leq \sum_{k \geq 0} \sqrt{\mathbb{E}\left[|\Xi_{A,0}^{(0)}(B_{k+1}(0))|^{8q+8}\right] \cdot \mathbb{P}(R_{A,0} > k)}, \end{aligned}$$

hvor vi i første ulighed brugte, at vi kan begrænse antallet af q -simplicies med antallet af punkter opløftet til $q+1$ og hvor vi i sidste ulighed brugte Cauchy-Schwarz. Vi vil nu begrænse de to udtryk inde i summen hver for sig. Vi har fra Lemma 7 i [5], at der eksisterer en konstant $c_2 > 0$, så

$$\sqrt{\mathbb{E}\left[|\Xi_{A,0}^{(0)}(B_{k+1}(0))|^{8q+8}\right]} \leq (k+1)^{c_2}.$$

Ved at bruge Lemma 5.8 så har vi for $k \geq N^2 + 9$, at

$$\mathbb{P}(R_{A,0} \leq k) \geq \mathbb{P}(E_{k/(4N), n, x})$$

og dermed ved at bruge Lemma 5.7 har vi, at der eksisterer et $c_1 > 0$ så

$$\mathbb{P}(R_{A,0} > k) \leq \mathbb{P}(E_{k/(4N),n,x}^c) \leq \exp(-c_1 \cdot (k/(4N))^{N/2}).$$

Nu kan vi konkludere, at der eksisterer konstanter $c_1, c_2 > 0$, så

$$\begin{aligned} \sup_{0 \in A \in \mathcal{W}} \mathbb{E} \left[\left| K_q(\mathcal{P}_{A_n(x)} \cup \{0\}, s) \setminus K_q(\mathcal{P}_{A_n(x)}, s) \right|^4 \right] &\leq \sum_{k=0}^{N^2+8} (k+1)^{c_2} \cdot \sqrt{\mathbb{P}(R_{A,0} > k)} \\ &+ \sum_{k=N^2+9}^{\infty} (k+1)^{c_2} \cdot \sqrt{\exp(-c_1 \cdot (k/(4N))^{N/2})}, \end{aligned}$$

hvor den sidste sum konvergerer ifølge kvotientkriteriet, som ønsket. \square

6 Simulationer

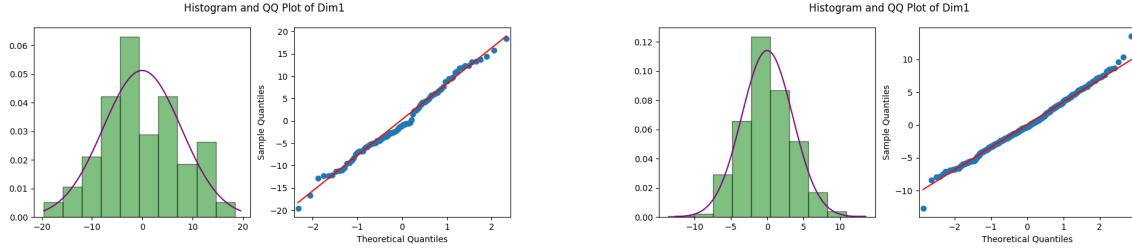
Vi vil i dette afsnit simulere Sætning 4.2 for forskellige vinduer $W \subseteq \mathbb{R}^3$, og for forskellige antal simulationer, m . Det gør vi, fordi vi gerne vil vise, at asymptotisk normalitet gælder allerede for mellemstore vinduer. Simulationerne laves ved at simulere punkter fra en homogen Poissonprocess i vinduet og finder Voronoi tessellations for disse punkter. Dernæst udtrækkes hjørnerne, kanterne og fladerne fra de Voronoi tessellations, som var udregnet og det tilhørende persistencediagram bliver udregnet. Det ser ud til, at simuleringstiden afhænger lineært af antallet af simuleringer og afhænger kvadratisk af antallet af punkter.

Vi undersøger for normalitet ved at lave et QQplot og et histogram. Histogrammet inkluderer en normalfordeling med middelværdi 0 og empirisk varians.

6.1 Persistent Betti tal

Et typisk persistence diagram ser ud som i Figur 4, så ud fra dette vælger vi at betragte det første $(15, 25)$ -persistent Betti tal, $\beta_1^{15,25}$. Jeg undersøger for normalitet ved at lave et QQplot og et histogram for de standardiserede total persistence. Histogrammet inkluderer en normalfordeling med middelværdi 0 og empirisk varians.

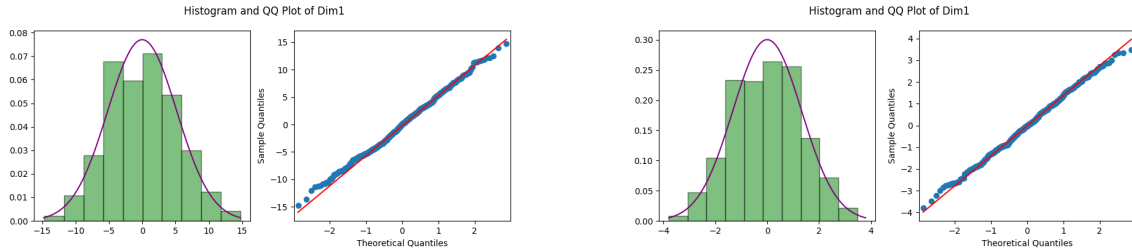
I de følgende plots er vinduet $W = [-200, 200]^3$, antal punkter er $n = 500$ og antallet af simulationer, m , er angivet.



Figur 6: Venstre plot er for $m = 100$ og højre plot for $m = 500$.

Vi ser i Figur 6, at det fint ligner en normalfordeling selv for et lille m .

Hvis vi prøver at ændre antallet af punkter til at komme fra en Poissonfordeling med intensitet $500/400^3$, og dermed en middelværdi på 500, i stedet for et fast antal af punkter vil vi forvente en større spredning. Vi ser i Figur 7, at den empirisk spredning er cirka halvanden gang så stor her sammenlignet med før, hvor vi havde et fast antal af punkter. Ved at gøre vinduet W større, vil vi forvente en en større spredning, da punkterne bliver mere spredt. Specielt ser vi, at den empiriske middelværdi og spredning for $W = [-200, 200]^3$ var henholdsvis $\mu = 1180$ og $s = 3, 5$, mens for $W = [-400, 400]^3$ var det $\mu = 295$ og $s = 1, 3$.



Figur 7: Venstre plot er for $m = 500$, middelværdi 500 samt $W = [-200, 200]^3$. Højre plot er for $m = 500$, $n = 500$ og $W = [-400, 400]^3$.

Hypotese test

Vi kan prøve at opstille følgende hypotese

$$H_0 = \text{Fast antal punkter}$$

$$H_A = \text{Poissonfordeling.}$$

Til at teste denne hypotese og finde styrken af testen, vil vi genere data fra vinduet $W = [-200, 200]^3$, $M = 500$ simulationer og henholdsvis $n = 200$ eller intensitet $200/400^3$, altså

middelværdi 200. Vi vil gentage dette 100 gange og dermed udregne et 95%-konfidensinterval for middelværdien baseret på asymptotisk normalitet. Konfidensintervallet blev $[209, 1; 214, 1]$.

Vi vil forkaste nulhypotesen, hvis den empiriske middelværdien er udenfor konfidensintervallet. Vi kan så udregne styrken af denne test ved at generere data fra den alternative hypotese og se, hvor ofte den empiriske middelværdien er udenfor vores konfidensinterval. Dette gjorde vi 100 gange, og den empiriske middelværdi var udenfor konfidensintervallet 25 ud af 100 gange. Altså har vi at den simulerede styrke af testen er 0,25.

Tilsvarende kan vi opstille den modsatte hypotese, altså

$$H_0 = \text{Poissonfordeling}$$

$$H_A = \text{Fast antal punkter.}$$

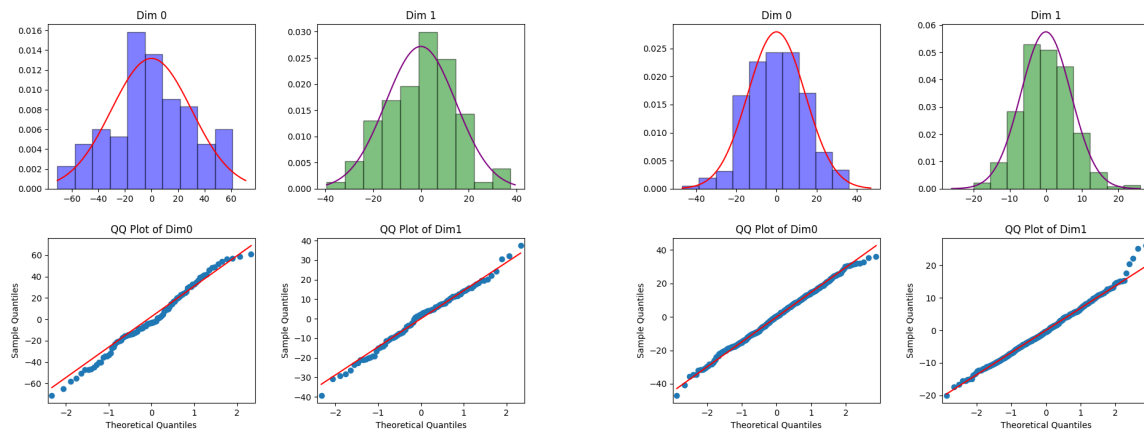
og udregne 95%-konfidensinterval for middelværdien. Konfidensintervallet blev $[208, 9; 216, 2]$. Igen generede vi data under den alternative hypotese 100 gange og en empiriske middelværdi var i konfidensintervallet 1 ud af 100 gange. Altså har vi den simulerede styrke af testen er 0,01.

Så vi kan konkluderer, at vores test er nogenlunde til at se afvigelser fra fast antal punkter, mens mindre god til til at se afvigelser fra antal punkter følger en Poissonfordeling.

6.2 Total persistence

I [5] vises det at der findes en functional central grænseværdisætning for det beslægtet M -begrænset persistent Betti tal. Man kan forstille sig, at der også findes functional central grænseværdisætning for persistent Betti tal. Vi vil derfor prøve at undersøge summen af alle levetider, som vi vil kalde total persistence.

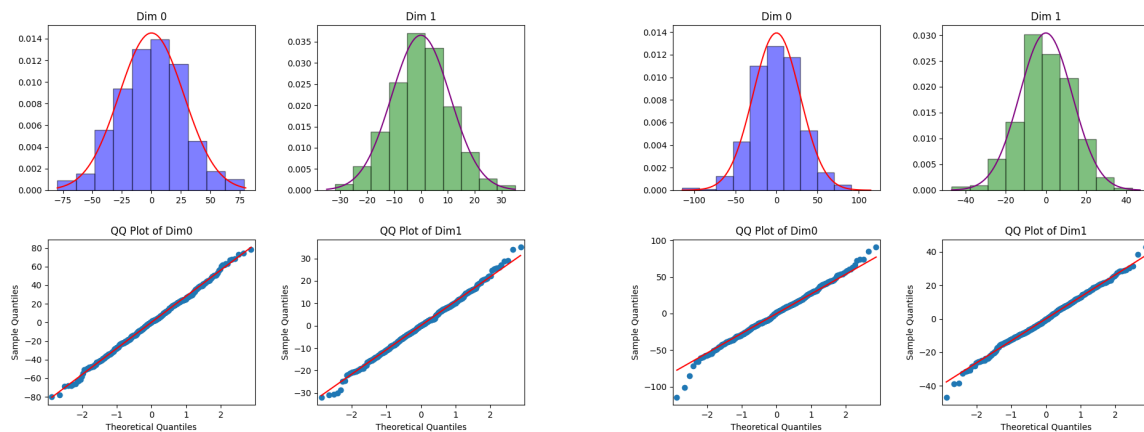
I de følgende plots er vinduet $W = [-200, 200]^3$, antal punkter er $n = 500$ og antallet af simulationer, m , er angivet.



Figur 8: Venstre plot er for $m = 100$ og højre plot for $m = 500$.

Vi ser også i Figur 8, at for total persistence, at det fint ligner en normalfordeling selv for et lille m .

Ligesom før prøver vi at ændre n til at følge en poissonfordeling med middelværdi 500 og prøver at ændre vinduet til $W = [-400, 400]^3$.



Figur 9: Venstre plot er for $m = 500$, middelværdi 500 samt $W = [-200, 200]^3$. Højre plot er for $m = 500$, $n = 500$ og $W = [-400, 400]^3$.

Vi ser i 9 samme tendens som før, altså at spredningen cirka bliver halvanden gang større når vi lader antal punkter komme fra en poissonfordeling. Når vinduet blev større ser vi igen samme tendens, at middelværdien og spredningen bliver større.

7 Litteratur

- [1] Christophe A. N. Biscio, Nicolas Chenavier, Christian Hirsch, and Anne Marie Svane. Testing goodness of fit for point processes via topological data analysis. *Electronic Journal of Statistics*, 14(1):1024 – 1074, 2020.
- [2] Alessandra Cipriani, Christian Hirsch, and Martina Vittorietti. Topology-based goodness-of-fit tests for sliced spatial data. *Computational Statistics & Data Analysis*, 179:107655, 2023.
- [3] Yasuaki Hiraoka, Tomoyuki Shirai, and Khanh Duy Trinh. Limit theorems for persistence diagrams. *The Annals of Applied Probability*, 28(5):2740–2780, 2018.
- [4] Christian Hirsch. Caput Statistics: Topological Data Analysis.
- [5] Christian Hirsch, Johannes Krebs, and Claudia Redenbach. Persistent homology based goodness-of-fit tests for spatial tessellations. *arXiv preprint*, 2022.
- [6] Nina Otter, Mason A Porter, Ulrike Tillmann, Peter Grindrod, and Heather A Harrington. A roadmap for the computation of persistent homology. *EPJ Data Science*, 6(1), aug 2017.
- [7] Mathew D. Penrose and Joseph E. Yukich. Central limit theorems for some graphs in computational geometry. *The Annals of Applied Probability*, 11(4):1005 – 1041, 2001.
- [8] Claudia Redenbach and André Liebscher. *Random Tessellations and their Application to the Modelling of Cellular Materials*, pages 73–93. Springer International Publishing, Cham, 2015.

8 Appendix

Koden brugt til at lave simuleringsdata brugt i Afsnit 6.

Beregning af tessellation

```
import numpy as np
from scipy.stats import gamma
from scipy.spatial.distance import pdist, euclidean
from scipy.spatial import Delaunay, ConvexHull
from tess import Container
import multiprocessing as mp
import seaborn as sns
import pyvoro
import time

start_time = time.time()

n = 500

import numpy as np
minv = np.array([-200, -200, -200])
maxv = np.array([200, 200, 200])
def containment(cell, minv = minv, maxv = maxv):
    return np.all(minv < np.min(np.vstack(cell.vertices()),
                                axis = 0)) and
           np.all(maxv > np.max(np.vstack(cell.vertices()), axis = 0))

npool = 10
```

```

nsims = 10
pois = [np.random.rand(npoi, 3) * (maxv - minv) + minv
for _ in range(nsims * npool)]
print( 'sim_done' )

import pickle
import gc

def face_verts(pt):
    """Face verticces
    # Arguments
        pt: cell centers
    # Result
        Face verticces
    """
    return [np.array(c.vertices())[f]
            for c in Container(pt,
                            limits = [minv, maxv],
                                radii = None,
                                periodic = True)
            for f in c.face_vertices()
            if containment(c)]

path = './sim_data/tess/'

wsiz = 1
for pts, lab in zip([pois],
                    ['pois']):
    print(lab)
    def dump_vor(i):

```

```

        pickle.dump(pyvoro.compute_voronoi(pts[i],
        list(np.array(list(zip(*[minv, maxv])))),
        100, radii = [], periodic = [True] * 3),
        open('{}{}{}.p'.format(path, lab, i), 'wb'))
    [dump_vor(i) for i in range(len(pts))]

end_time = time.time()

execution_time = round(end_time - start_time)

execution_time_in_minutes = round(execution_time / 60)

print("Execution_time:", execution_time_in_minutes, "minutes")

```

Ekstraktion af hjørner, kanter og flader

```

from scipy.cluster.hierarchy import single, fcluster
import pyvoro
from scipy.spatial import distance
import pandas as pd
import numpy as np
import time

start_time = time.time()

minv = np.array([-200, -200, -200])
maxv = np.array([200, 200, 200])
def containment(cell, minv = minv, maxv = maxv):
    return np.all(minv < np.min(np.vstack(cell['vertices']),
    axis = 0)) and

```



```

np.all(maxv> np.max(np.vstack( cell[ 'vertices' ] ) ,
axis = 0))

def cluster_vertices( tes , eps = 1e-5):
    """ Clustering of vertices
    # Arguments
        tes: tessellation
        eps: threshold for clustering
    # Result
        clustered vertices together with list describing the
        association of the unclustered vertices to the clusters
    """

    #collect vertices from cells — take into account periodic bc
    tes = [c for c in tes if containment(c)]
    verts_unc = [np.array(vert)
                  for cell in tes
                  for vert in cell[ 'vertices' ]
                  if containment(cell)]

    cl_map = fcluster( single( distance.pdist(
verts_unc , 'euclidean' ) ) ,
eps ,
criterion = 'distance' ) - 1

    verts_clust = [[p] + list(v)
                   for p, v in zip( cl_map ,
                                   verts_unc )]

    return pd.DataFrame( verts_clust ).groupby(0).

```

```

mean().values, cl_map

#compute offsets for each cell to global vertex id
compute_offset = lambda tes: [0] +
list(np.cumsum([len(cell['vertices'])
                  for cell in tes]))

def clustering(tes):
    """List of cavities, faces and edges given by ids of
    clustered vertices
    # Arguments
        tes: tessellation
    # Result
        List of cavities, faces and edges given by ids of
        clustered vertices
    """
    tes = [c for c in tes if containment(c)]
    offset = compute_offset(tes)
    _, cl_map = cluster_vertices(tes)

    #unclustered cavities and their vertices
    cavs_unc = [cl_map[range(l, h)]
                 for l, h in zip(offset,
                                np.roll(offset, -1))][: -1]

    #unclustered faces and their vertices
    faces_unc = [np.roll(cl_map[np.array(face['vertices']) + o],
                          -np.argmax(cl_map[np.array(face['vertices']) + o]))
                 for cell, o in zip(tes,

```

```

                                offset)
        for face in cell['faces']
        if containment(cell)]

#order faces
faces_unc = [list( np.roll( facc[:, -1],
-np.argmax(facc[:, -1]))
                if facc[1] > facc[-1]
                else facc)
            for facc in faces_unc]

faces_unc = [list( np.roll( facc[:, -1],
-np.argmax(facc[:, -1]))
                if facc[1] > facc[-1]
                else facc)
            for facc in faces_unc if
            len(np.unique(facc)) > 2]

#unclustered edges
edges_unc = [(cl_map[np.array(vert) + o]).tolist()
              for cell, o in zip(tes,
                                offset)
              for face in cell['faces']
              for vert in list(zip(face['vertices'],
                                   np.roll(face['vertices'], 1)))
              if containment(cell)]
return [[list(x) for x in set([tuple(x) for x in

```

```

[sorted(cfe, key = kf)
for cfe in cfes]]])
for cfes, kf in zip([cavs_unc, faces_unc, edges_unc],
[lambda x: 0, lambda x: 0, None]])

from multiprocessing import Pool
from scipy.spatial import KDTree, distance

import warnings
import pickle
import pandas as pd
import numpy as np

path = './sim_data/tess/'
path_cu = './sim_data/clust/'

labs = ['pois']

nsample = 500

clustered_vor = []
verts_vor = []

for lab in labs:
    print(lab)
    with warnings.catch_warnings():
        warnings.simplefilter("ignore")

```

```

def clv(i):
    if i%1==0:
        print(i)
    tess = pickle.load(open('{}_{}_{}.p'.format(path, lab, i),
        'rb'))
    pickle.dump(clustering(tess), open('{}clu{}_{}.p'.format(
        path_cu, lab, i), 'wb'))
    print('clustering_done')
    pickle.dump(cluster_vertices(tess)[0],
    open('{}clv{}_{}.p'.format(path_cu, lab, i), 'wb'))
    del tess
    [clv(i) for i in range(nsamples)]

end_time = time.time()

execution_time = round(end_time - start_time)

execution_time_in_minutes = round(execution_time / 60)

print("Execution_time:", execution_time_in_minutes, "minutes")

```

Beregning af persistence diagram

```

import warnings
from scipy.spatial.qhull import QhullError
import gudhi
from sklearn.decomposition import PCA
from multiprocessing import Pool
import pickle

```

```

from scipy.cluster.hierarchy import single, fcluster
from scipy.spatial import distance
import pandas as pd
import numpy as np
import time

start_time = time.time()

def cluster_vertices(tes,
                    eps = 1e-5):
    """Clustering of vertices
    # Arguments
        tes: tessellation
        eps: threshold for clustering
    # Result
        clustered vertices together with list describing the
        association of the unclustered vertices to the clusters
    """

    #collect vertices from cells — take into account periodic bc
    verts_unc = [np.array(vert)%1
                  for cell in tes
                  for vert in cell['vertices']]

    cl_map = fcluster(single(distance.pdist(
        verts_unc, 'euclidean')),
                    eps,
                    criterion = 'distance') - 1

    verts_clust = [[p] + list(v)

```

```

        for p, v in zip(cl_map, verts_unc)]

    return pd.DataFrame(verts_clust).groupby(0).mean().values,
           cl_map

#compute offsets for each cell to global vertex id
compute_offset = lambda tes: [0] +
list(np.cumsum([len(cell['vertices'])
                for cell in tes]))

def clustering(tes):
    """List of cavities, faces and edges given by ids of
    clustered vertices
    # Arguments
        tes: tessellation
    # Result
        List of cavities, faces and edges given by ids of
        clustered vertices
    """
    offset = compute_offset(tes)
    _, cl_map = cluster_vertices(tes)

    #unclustered cavities and their vertices
    cavs_unc = [cl_map[range(1, h)]
                for l, h in zip(offset,
                                np.roll(offset, -1))][: -1]

    #unclustered faces and their vertices
    faces_unc = [np.roll(cl_map[np.array(face['vertices']) + o],

```

```

-np.argmax(cl_map[np.array(face['vertices']) + o]))
        for cell, o in zip(tes,
                            offset)
        for face in cell['faces']]

#order faces
faces_unc = [list( np.roll( face[:, -1],
-np.argmax(face[:, -1]))
                if face[1] > face[-1]
                else face)
            for face in faces_unc]

faces_unc = [list( np.roll( face[:, -1],
-np.argmax(face[:, -1]))
                if face[1] > face[-1]
                else face)
            for face in faces_unc if len(np.unique(face)) > 2]

#unclustered edges
edges_unc = [(cl_map[np.array(vert) + o]).tolist()
              for cell, o in zip(tes,
                                offset)
              for face in cell['faces']
              for vert in list(zip(face['vertices'],
                                np.roll(face['vertices'], 1))))]

return [np.unique([sorted(cfe,
                        key = kf)
                    for cfe in cfes],

```



```

axis = a) for a, cfs, kf
in zip([None, None, 0],
[cavs_unc, faces_unc, edges_unc],
[lambda x: 0, lambda x: 0, None]])

import cechmate as cm
from itertools import combinations

def filt(cavs,
        faces,
        edges,
        cav_verts,
        fac_verts,
        edg_verts):
    """Compute filtration times for cavities
    # Arguments
        cavs: cavities
        faces: faces
        edges: edges
        cav_verts: coords of cavities
        fac_verts: coords of cavities
        edg_verts: coords of cavities
    # Result
        Filtration times for cavities
    """

    #pca to account for lower dimensions
    ch = cm.Alpha(maxdim = 2,
                  verbose = False)

    pca = PCA(2)

```

```

#filtration times for cavities and faces
cf_times = [[[[sorted([cav[z] for z in x]), y]
               for x,y in ch.build(cv)
               if len(x) == 4]
             for cav, cv in zip(cavs, cav_verts)],
            [[[[sorted([face[z] for z in x]), y]
               for x, y in ch.build(pca.fit_transform(fv))
               if len(x) == 3]
             for face, fv in zip(faces, fac_verts)]]

#replace by maximal filtration time per cell/face and
add lower dimensions.
cf_times_max = [[ max(y
                      for [x, y] in cf_time)
                 if len(cf_time) > 0 else 999
                 for cf_time in cf_timess]
                 for cf_timess in cf_times]

#add lower dimensional structures to ensure it is
a simplicial complex
return [[[list(z), cf_time_max]
           for cf_time, cf_time_max in zip(cf_timess,
           cf_timess_max)
           for x, _ in cf_time
           for z in [x] + [list(v)
                         for u in [list(combinations(x, i))
                                   for i in r]
                         for v in u]]]

```

```

    for cf_timess, cf_timess_max, r in zip(cf_times,
        cf_times_max,
        [range(2, 4), range(2, 3)]) \
    + [[[edge, distance.euclidean(ev[0], ev[1])/2]
        for edge, ev in zip(edges, edg_verts)]] \
    + [[[[x], 0]
        for x in range(np.max(edges) + 1)]]

```

```

def update(*tes):
    """Update filtration times in lower dimensional structures
    # Arguments
        tes: tessellation
    # Result
        3 arrays containing the filtration times in cavities,
        faces and edges, respectively
    """
    fc, ff, fe, fv = filt(*tes)

    #cavities
    dim1c, dim2c, dim3c = [[a for a in fc
        if len(a[0]) == i + 2]
        for i in range(3)]

    #merge faces with cavities
    dim1f, dim2f = [[a for a in ff
        if len(a[0]) == i + 2]
        for i in range(2)]
    dim2fc = dim2c + dim2f
    dim2fc = [[[int(z) for z in b[:-1]], b[-1]]

```

```

        for b in pd.DataFrame([a[0] + [a[1]]
                                for a in dim2fc]).groupby(
                                list(range(3))).min().reset_index().values]

    #merge edges with rest
    dim1efc = fe + dim1f + dim1c
    dim1efc = [[int(z) for z in b[:-1]], b[-1]]
                for b in pd.DataFrame([a[0] + [a[1]]
                                        for a in dim1efc]).groupby(
                                        list(range(2))).min().reset_index().values]

    return fv + dim1efc + dim2fc + dim3c

dgms = []

path_cu = './sim_data/clust/'

labs = ['pois']

nsample = 500

for lab in labs[0:1]:
    with warnings.catch_warnings():
        warnings.simplefilter("ignore")
        print(lab)

    #Create diagram
    def dgms_fun(i):
        if i%1==0:
            print(i)

```

```

cv = pickle.load( open( '{}clu{}{}.p'.format(
path_cu, lab, i), 'rb'))

vw = np.array(pickle.load( open( '{}clv{}{}.p'.format(
path_cu, lab, i), 'rb'))))

vv = ([[x for x in vw[f]]
        for f in cvv]
        for cvv in cv])

upd = update([], *(cv[1:]), [], *(vv[1:]))
st = gudhi.SimplexTree()
[st.insert(*splx) for splx in upd]
pd.DataFrame([([x[0]] +list(x[1])
                for x in st.persistence(
homology_coeff_field=2)
if x[1][1]!= np.inf])).to_csv(
'{}dgm{}{}.csv'.format(
path_cu, lab, i),
header=None, index=False)

[dgms_fun(i) for i in range(nsampl)]

end_time = time.time()

execution_time = round(end_time - start_time)

execution_time_in_minutes = round(execution_time / 60)

print("Execution_time:", execution_time_in_minutes, "minutes")

```