

パーシステントホモロジーの optimal volume と stable volume について

大林一平*

November 26, 2021

Abstract

This paper reviews an optimal volume[1] and a stable volume[2], which are helpful for data analysis using persistent homology.

The purpose of this paper is to explain an optimal volume and a stable volume to PH users who are not familiar with mathematical theory of homology, persistent homology, and computational geometry.

1 はじめに

位相的データ解析 [3, 4] は数学のトポロジーを活用したデータ解析手法の総称で、その中でもパーシステントホモロジー (PH)[5, 6] という概念がよく使われている。PH は分子遺伝学 [7] や材料科学 [8, 9] など幅広い領域で利用が進んでいる。

PH の出力はパーシステント図 (Persistence diagram, PD) と呼ばれる二次元散布図もしくは二次元ヒストグラムである。PD 上の各点はデータのループ構造, 空洞構造に対応しており, その構造を抽出できると便利である。このような解析を我々は逆解析と呼んでいる。PH の逆解析にはいくつかの手法が知られているが, 本稿では筆者の大林が提案した optimal volume[1] という手法を紹介する。

optimal volume は有用な手法であるが, (1) ノイズに弱い (2) 最小の構成要素の抽出に時々失敗する, という弱点がある。筆者は最近この問題の解決策として stable volume という手法 [2] を提案した。

本稿ではこの2つのアイデアについて解説する。読者としてはこれらツールとして利用する利用者を想定し, そのデータ解析における意義と利用方法, 直感的な説明など重視し, 数学の専門家以外にもわかりやすいように解説する。数学的な定式化や数学的性質, アルゴリズム等については説明しないので, 調べたい方はそれぞれの論文を参照して欲しい。

*岡山大学サイバーフィジカル情報応用研究コア (Cypher)

e-mail: i.obayashi@okayama-u.ac.jp

本研究の一部は JSPS 科研費 JP19H00834, JP19KK0068, JP20H05884, および JST さきがけ JPMJPR1923, JST CREST JPMJCR15D3 の助成を受けたものである。

2 PH について

この節では図 1 によってデータから PD を計算する方法について説明しよう。

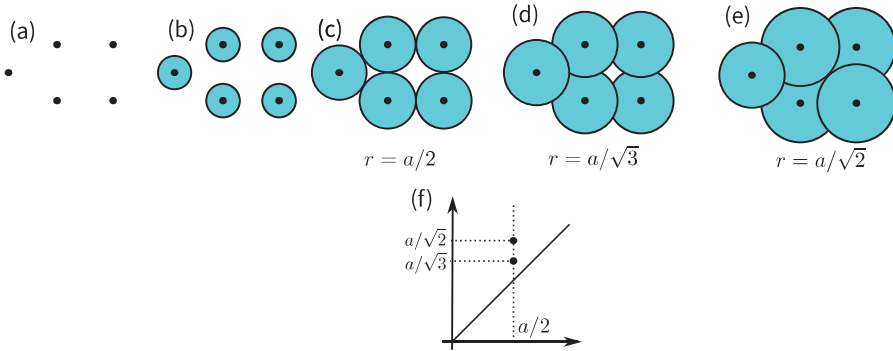


図 1: 5 点のポイントクラウドから PD を計算する方法 (a) 入力ポイントクラウド (b) 半径 r が $a/2$ 以下の円を置いた場合 (c) $r = a/2$ (d) $r = a/\sqrt{3}$ (e) $r = a/\sqrt{2}$ (f) PD

図 1(a) のような 5 点を入力データとする (このような点の集まりのデータはポイントクラウドと呼ばれる。この図は正三角形と正方形から構成されているように見える (a を一辺の長さとする)。この図自体はトポロジカルな情報を持っていないが、トポロジー的構造を持っているようにも見える。そこで半径 r の円を各点の上に置くことでトポロジカルな構造を構築する (図 1(b-e))。

ここで問題なのが半径をどう決めるかである。半径を大きくすると、孔の数や連結成分の数が (b) から (e) の間で変わっていくのがわかるだろう。PH の本質的アイデアは半径を一つ定めるのではなく、図 1(b) から (e) までのように図形が増大するプロセスを考えることである (このような図形増大プロセスをフィルトレーションと呼ぶ)。このプロセスにおいて、図 (c) で孔が 2 つ発生し、(d) でそのうちのひとつが消え、(e) でもうひとつが消える。PH の数学理論によって孔の発生と消滅の 1 対 1 対応を作ることができる。この発生の半径 birth time、消滅の半径を death time と呼ぶ。さらに birth time と death time のペアを birth-death pair と呼ぶ。そして birth-death pair の集まりをパーシステント図 (PD) と呼ぶのである。PD は二次元の散布図やヒストグラムなどで可視化される。

図 1 の例では (c) で発生した孔の一方が (d) で消え、もう一方が (e) で消えると対応付けることができるので、PD は $\{(a/2, a/\sqrt{3}), (a/2, a/\sqrt{2})\}$ であり、図 (f) のように可視化される¹。

ここまでは 2 次元のポイントクラウドについて述べてきたが、3 次元ポイントクラウドの場合にも球を置いて半径を大きくしていくことで同様のことができる。3 次元の場合、トポロジーの理論からは二種類の孔を定義できる (トンネル的な通り抜けができる「孔」、風船的な「空洞」) ので、それに対応して 2 通りの PD が定義される。それぞれ 1 次元 PD, 2 次元 PD などと呼ばれる。

¹計算幾何学や位相的データ解析の慣習で半径の二乗を使う場合もしばしばある

3 Optimal volume

図 1(a) のようなポイントクラウドから図 1(f) のような PD が得られたとき、PD の各 birth-death pair に対応する構造を抽出することができると便利である。この図の例では $(a/2, a/\sqrt{3})$ が正三角形、 $(a/2, a/\sqrt{2})$ が正方形に対応しているのでそれを取り出せるとよい。こういった解析を我々は逆解析と読んでいる。

しかしこの問題は実は簡単な問題ではない。これを図 2 で説明しよう。この図は孔が 1 つあるが、その孔に対応する構造は z_1, z_2 の二種類ある。孔の定式化はホモロジーと呼ばれる数学が用いられる。ホモロジーにおいては孔というのはそれを取り囲むループで定式化される。図 2 の例題の場合孔を取り囲むループというのは z_1, z_2 の二種類が考えられ、三角形が張られている z_3 の部分をキャンセルする (数学的には商という代数的な操作をする) ことで z_1 と z_2 が同一視される。こうして孔が 1 つというのが定式化されるのであるが、そのためこの孔は z_1, z_2 のどちらでも妥当に表現できるのである。では z_1, z_2 のどちらがより良いか、という点直感的には z_1 のほうが良いであろう。というのは z_1 のほうが z_2 よりもタイトに孔を取り囲んでいるからである。

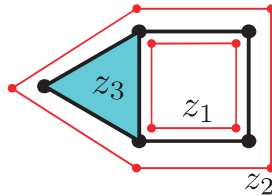


図 2: 孔が一つあるような図形

するとこの問題はある種の最適化問題として定式化できる。ループの「タイトさ」の指標を決めておいて候補の中から最も良いものを探すのである。PH に関しては指標の違いや候補の取り方の違いに応じていくつかの定式化が提案されている [10, 1, 11]。Optimal volume[1] はその定式化の一つで、候補を孔の発生消滅と整合するように取り決め、ループで囲まれる領域の面積を指標としてこれを最小化している²。二次元 PD (つまり空洞) の場合には同様に空洞の体積を最小化する³。Optimal volume は HomCloud[12] に実装され実際のデータ解析に既に利用可能である。

ここで読者は次の点が気になるかもしれない。図 1 では円の集まりが作るトポロジカルな構造を考えているが、一方図 2 では頂点、辺、三角形といったパーツの集まりが作るトポロジカルな構造を考えている点である。実はアルファ複体というものをを用いると円の集まりを図 2 のような構造に還元できることが知られている。アルファ複体については本稿の A で解説しているので参考にするとよい。

内部的にアルファ複体を使うため、optimal volume の出力は図 2 のような点を直線で繋いだループや点を頂点とする多面体などとなる。円や球とは直接は対応しない出力がなされるという点は注意するとよい。

²他の手法ではループの長さを最小化することが多い

³同様に他の手法では表面積を最小化することが多い

4 Stable volume

Optimal volume のような手法を用いれば PD の各点に対応するループや空洞が抽出できるので、これで十分に思えるが実はそうでもない場合がある。実はこういった手法には次に示すような弱点がある：

- ノイズに対して不安定な場合がある
- 期待されるような最小の構成要素を取り出すことに失敗する場合がある

これらの問題について図 1 にノイズを加えた状況を考えて説明し、この問題を解決する一つの方法である stable volume[2] について説明しよう。

図 1(f) の 2 つの birth-death pair は図 1(a) の正三角形と正方形に対応している。すると optimal volume や類似の手法を使ってこれらの形が得られることが期待される。これは入力データに小さなノイズが加わったときも同等であると期待される。しかし実際にノイズを加えると期待通りには動かない。

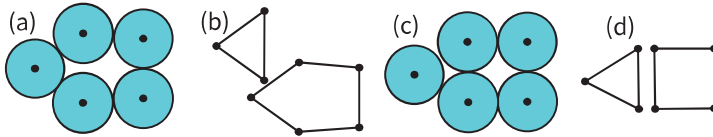


図 3: 小さなノイズで optimal volume が大きく変わる例

ノイズが加わった結果図 3(a) のような点配置になると、円を大きくしていく際にまず 5 個の円が一続きにつながる。そのため optimal volume の出力は 3(b) のような五角形と三角形が得られる。一方図 3(c) のようにノイズが加わると optimal volume の出力は図 3(d) のような四角形と三角形となる。PH のアルゴリズムはどの円同士が先に繋がったを見て PD を計算するので、このようなことが生じるのである。このように小さなノイズの影響で出力が大きく変わってしまう。

ここで 3(b) と 3(d) を比べると、(d) のほうが期待される結果であろう。というのはこのポイントクラウドの「最小の構成要素」は三角形と四角形であり、この 2 つがくっついた五角形ではないからである。

このように、optimal volume は小さなノイズによって出力が大きく変化したり、ノイズの具合によって最小の構成要素をうまく捉えられなかったりする。このことは optimal volume による解析結果の信頼性に問題をもたらす。この現象は optimal volume に限らず、PH に関する類似の手法でも同様の問題がある。

PD に関しては PH の安定性定理 [13, 14, 15, 16] と呼ばれる数学的定理があり、このような問題が生じないことが知られている。この定理が数学的に証明していることは、入力のポイントクラウドの位置の変位に対して出力の PD はその変位の大きさ程度しか変化しないことである。PH の安定性定理は PD によるデータ解析の信頼性を保証してくれる定理なのであり、一方 optimal volume のような逆解析手法は数学的には安定性定理がないと言える。

この問題に対する一つの解決策として stable volume が提案された。stable volume は optimal volume をもとにしていて、最適化問題の定式化ををちょっと変更すると得られる (詳しくは論文 [2] を参照)。

Stable volume の定式化のためにはノイズ幅パラメータというパラメータ ϵ が導入されている. この ϵ はある意味で仮想的なノイズの大きさであり, 以下の事実が論文 [2] で証明されている.

Theorem 4.1. P を \mathbb{R}^D 上のポイントクラウドとし, $A(P)$ を P のアルファ複体によるフィルトレーションとする. するとこのフィルトレーションから k 次元 PD が得られるが, これを $D_k(A(P))$ と書くことにする. $(b, d) \in D_k(A(P))$ と birth-death pair を一つ固定しておくこととする. $\epsilon > 0$ を一つ決めておき, (b, d) に対するノイズ幅パラメータ ϵ の stable volume を $S_\epsilon(b, d)$ と表わすことにする.

$R_\epsilon(P)$ を $A(P)$ に最大 $\epsilon/2$ のノイズを加えたフィルトレーション全体からなる集合とする. すると PH の安定性定理より各 $F \in R_\epsilon(P)$ に対する k 次元 PD $D_k(F)$ は (b, d) に十分近い点 (b_F, d_F) を持つ. この (b_F, d_F) に対する optimal volume の内側の領域を $O(F)$ と表わすことにしよう. ここで内側というのは, $k = 1$ ならばループの内側の領域, $k = 2$ ならば空洞内部の領域を表わす.

すると, $k = D - 1$ で, ϵ が十分小さく, P や (b, d) が適当な良い条件を満たすならば,

$$(1) \quad S_\epsilon(b, d) = \bigcap_{F \in R_\epsilon} O(F)$$

を満たす.

上の定理は内容の厳密さは無視している. 例えば「適当な良い条件」や「フィルトレーションにノイズを加える方法」などはちゃんと説明していない. より正確なことが知りたいならば元の論文を参照して欲しい.

直感的に説明すると, この定理は次のことを意味する.

1. 入力のパポイントクラウドは D 次元ユークリッド空間上にあるとする
2. ここから得られる $D - 1$ 次元 PD の birth-death pair (b, d) を一つ固定する
3. この birth-death pair に対する optimal volume が計算できる (図 4 の実線)
4. ノイズ幅パラメータ ϵ も適当に小さい正の値で固定しておく
5. 入力に最大 $\epsilon/2$ の大きさのノイズを加えると k 次元 PD も変化するが, その変動は PH の安定性定理から小さいことが保証されており, (b, d) に非常に birth-death pair がある
6. その非常に近い birth-death pair をうまく取って optimal volume を計算すると, ノイズの影響で元の optimal volume とずれる (図 4 の破線)
7. ノイズの最大値は固定して様々なノイズの入れかたをすると対応する optimal volume は変動する (図 4 の破線)
8. Optimal volume の共通部分を取る (図 4 の灰色の領域)

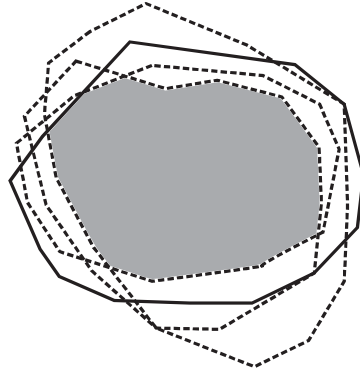


図 4: Optimal volume と stable volume の模式図

9. この灰色の領域が stable volume で得られるものである

この定理は \mathbb{R}^D 上のポイントクラウド $D-1$ 次元 PD でしか成立しないが、実際に様々な例に適用するとそれ以外の次元の PD でも stable volume はうまく機能することが多い。この理由はおそらくは元データが何らかの低次元構造を部分的に持ち、その低次元構造が stable volume の計算に効いているからではないかと予想される。

4.1 Stable sub-volume

前節で Stable volume が機能する理由として、元データが持つ低次元構造が効いてきているからではないか、という予想を述べた。この予想を利用して *stable sub-volume* という概念も [2] で定義されている。

Stable sub-volume とは optimal volume の領域の上に制限したときの stable volume として定義されている。つまり optimal volume を低次元構造とみなして計算するのである。このため、

- Stable sub-volume は stable volume より保守的な結果を返す
- Stable sub-volume はノイズ幅パラメータを色々変えて試したいときには stable volume より高速計算できる

という特徴がある。2つ目のほうは明らかな利点だが、1つ目のほうは利点とも欠点とも言いがたい。実例による stable volume と stable sub-volume の差異は [2] の論文で詳細に調べられているのでそちらを参考にして欲しい。

5 例

ここでは optimal volume と stable volume についていくつかの例を紹介する。

まずは optimal volume をアモルファスシリカに適用した例を紹介する。シリカとは二酸化ケイ素 (SiO_2) のことで、原子レベルで様々な空間構造を取る

ことが知られている。例えば水晶が良く知られたシリカの結晶である。シリカは結晶構造以外にも化学結合によるランダムなネットワークが発達した構造が良く知られており、このようなランダム構造を持つ個体のことをアモルファスと呼ばれる。我々が「ガラス」と呼んでいるものは基本的にアモルファスの一種である。

図5(a)はその原子配置で、(b)がそこから計算した1次元PDである。原子配置のデータはISAACS[17]の例 (<http://isaacs.sourceforge.net/ex.html>) からダウンロードしたもので、X線放射光の計測データから逆モンテカルロ法で構築した原子配置である。読者は1次元PDに特徴的に見える縦筋が気になるであろう。そこでこの縦筋から1点 birth-death pair を取りだして(座標は(0.684, 3.999)), そこから計算した optimal volume が図5(c)である。赤と青の点はそれぞれOとSiの原子を表している。抽出されたリング(緑)はOとSiが交互に表われていることがわかる。これはSiとOの間の化学結合によって形成されるリング構造である。この縦筋から birth-death pair を取り出して同様に計算するとOとSiが交互に表われるというのは共通しているがその形は様々である。birth time が揃っているのは化学結合の距離が一定であることに由来し、それが縦筋のように見えるのはリングの形や大きさが多様であることに対応していることがわかる。また図(c)の半透明の赤色の領域はリングに囲まれた領域で、optimal volume はこの領域を最小化している。このようにして optimal volume を利用することでPDが抽出した情報を深く理解することができるようになる。[2]では同じデータに対して optimal volume と stable volume を比較しているのでそちらも見るとよいだろう。

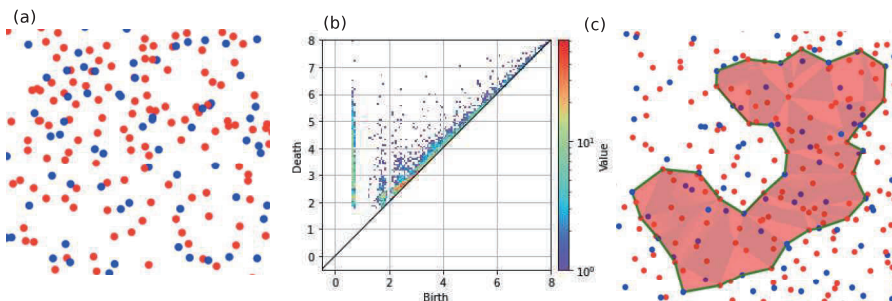


図 5: (a) アモルファスガラスの原子配置。赤が酸素原子で青がケイ素原子 (b) 原子配置から計算した 1 次元 PD (c) Optimal volume

図6は[2]から引用した optimal volume と stable volume の図である。(a)が入力データで格子点に小さな摂動を加え、その後ランダムに点を取り除いたものである。(b)はこのポイントクラウドの1次元PDである。(c)は(0.496, 4.371)という birth-death pair に対応する optimal volume である。(d)は同じ birth-death pair に対する stable volume(ノイズ幅パラメータは0.10)である。

この図を見ると、(c)(d)はどちらもポイントクラウドの隙間の領域を取り出しているが、(c)は不自然に大きい領域を取り出してしまっているように見える。一方(d)はより自然な抽出に成功しているように見える。このように

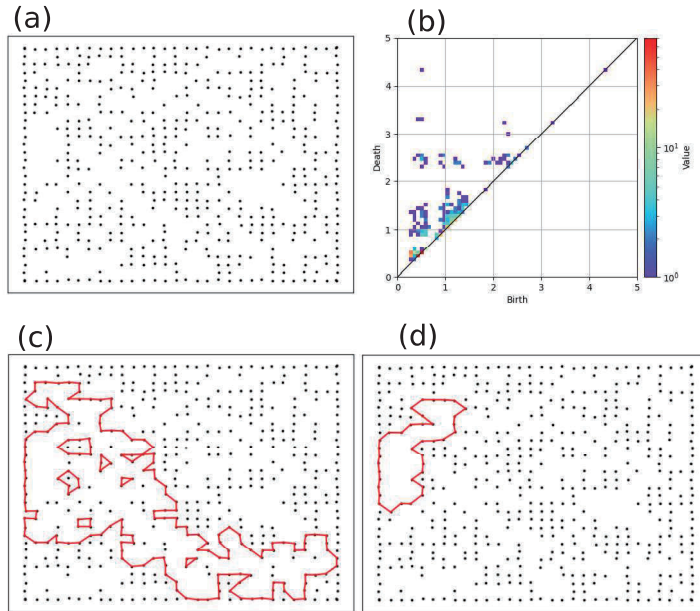


図 6: Optimal volume と stable volume. [2] の Figure 7 より引用

stable volume のほうが「自然に見える」をうまく取り出せる可能性がある。

ただ、stable volume をうまく使うにはノイズ幅パラメータをうまく決める必要がある。以下の図 7 は横軸にノイズ幅パラメータを、縦軸に対応する stable volume のサイズをプロットしたものである。ここでサイズというのはリング内部の三角形の個数で測った面積である。

この図を見るとノイズ幅パラメータを大きくすると stable volume は小さくなっていくことがわかる。これはこのパラメータの意味を考えれば当然である。この図で注目すべきは横軸が 0.06 から 0.27 の間に大きなプラトーがあることである。パラメータがこの間にある限り stable volume は変化しないということである。この事実の意味することはこのあたりでは stable volume はノイズ幅パラメータの変化に対しても頑健であるということである。これはノイズ幅パラメータをこの範囲にとれば信頼できる結果を得られることを示唆している。

[2] にはこの例の他にも様々な例を載せているのでそちらも参考にして欲しい。

6 他の手法との比較

4節で述べた逆解析の弱点(ノイズに弱い等)を解決する別の手法として Bendich らによって提案された統計的な手法 [18] がある。これは実際に入力データにノイズを加えて逆解析の手法を適用し、その空間的な平均を取る方法である。stable volume ではノイズは仮想的なものであったが、この手法で

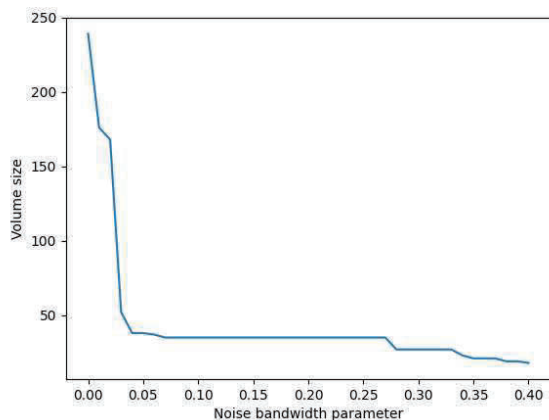


図 7: Stable volume のサイズ. [2] の Figure 11 より引用

は実際にノイズを加えて計算するのである。この統計的な手法は賢いアイデアで実装も簡単であるが、平均が十分良い値を持つ程度に繰り返し計算する必要があるという欠点がある。[18] では 1000 から 10000 回繰り返し計算している。これほどの繰り返し数は不要ではあるが、おそらく 10 から 100 回は繰り返しが必要だろう。そういう意味で stable volume は軽量性と取り扱いのしやすさに利点がある。

7 おわりに

本稿では PH によるデータ解析に有用な optimal volume, stable volume という手法について紹介した。これらの手法は既に HomCloud[12] に実装されており、実際のデータ解析に利用可能である。是非利用して欲しい。利用した結果を出版する際には [1, 2] を参考文献に載せると良い。

A アルファ複体

この節ではアルファ複体というものについて解説する。図 1 では円の集まりが作るトポロジカルな構造を考えたが、図 2 では頂点、辺、三角形といったパーツの集まりが作るトポロジカルな構造を考えている。図 2 のような構造は単体複体と呼ばれるものである。ポイントクラウドを解析するには円の集まりを考えるほうが都合が良く、一方コンピュータ上に実現するには単体複体のほうが都合が良い。実はアルファ複体を用いると円の集まりと同じトポロジカルな情報を持つ単体複体が構成できるのである。そのため PH のソフトウェアではポイントクラウドの解析のためになアルファ複体が良く使われている。

まずは単体複体について説明しよう。単体複体というのは単体を有限個うまく貼り合わせた図形として定義される。そして単体というのは頂点 (0 次元

単体), 線分 (1次元単体), 三角形 (2次元単体), 三角錐 (3次元単体), さらに次元の高い単体..., などである。

より形式的に定義すると次の通りになる。 D 次元ユークリッド空間上の d 次元単体 ($0 \leq d \leq D$ の範囲で考える) というのは一般の位置にある $d+1$ 個の頂点 $\{\mathbf{v}_0, \dots, \mathbf{v}_d\}$ の凸包として定義される。ここで一般の位置にあるというのは $\mathbf{v}_1 - \mathbf{v}_0, \dots, \mathbf{v}_d - \mathbf{v}_0$ という d 個のベクトルが一次独立であることを意味する。また, 凸包というのはその $d+1$ 個の頂点の「内側」, より正確には

$$(2) \quad \mathbf{v}_0 + t_1(\mathbf{v}_1 - \mathbf{v}_0) + \dots + t_d(\mathbf{v}_d - \mathbf{v}_0)$$

というベクトルであらわされる領域全体 (t_1, \dots, t_d は $0 \leq t_1, \dots, t_d, 0 \leq t_1 + \dots + t_d \leq 1$ という範囲を動く) を表している。

次に単体の面について定義する。 $\{\mathbf{v}_0, \dots, \mathbf{v}_d\}$ という頂点で定義される単体の面とは, この集合の部分集合を頂点とする単体のことである。一般の位置の条件は部分集合を取っても成り立つためこれが単体の定義を満たすことは明らかであろう。また面は元の単体の一部であることも簡単にわかる。三角錐の面としては三角形, 辺, 頂点であり, 三角形の面は辺と頂点, 線分の面は頂点なる。

単体という概念を用いて単体複体の定義について説明しよう。単体複体とは

1. 単体の有限個の集合 X であり
2. $\sigma \in X$ ならば σ の面はすべて X に含まれ
3. $\sigma, \tau \in X$ が共通部分を持つならその共通部分は X に含まれる

という条件を満たすものである。条件3が「うまく貼り合わせる」というのを定式化したものである。単体は頂点の組み合わせだけで表現できるので, 図形をコンピュータで表現するには非常に都合がよい。

ここまでの準備のもとアルファ複体について定義しよう。アルファ複体はポイントクラウドに対して, そのボロノイ分割を用いて定義される。図8の例を用いて説明する。図8(a)の4点が入力データとしよう (これを母点と呼ぶ)。このときボロノイ分割とは図8(a)の線で表されているような分割で, ユークリッド空間の各点をどの母点に近いかで領域分けしたものである。二次元平面の場合には母点の垂直二等分線の一部で領域分けされることとなる。三次元空間の場合もボロノイ分割が定義されるが, その場合には母点の垂直二等分面の一部と同様に領域分けされる。数学的にきちんと定義すると, n 個の点 $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ に対してそのボロノイ領域 $V(\mathbf{v}_i)$ は

$$(3) \quad V(\mathbf{v}_i) = \{\mathbf{x} \in \mathbb{R}^D \mid \|\mathbf{x} - \mathbf{v}_i\| \leq \|\mathbf{x} - \mathbf{v}_j\| \text{ がすべての } j \text{ で成立する}\}$$

として定義され, n 個のボロノイ領域の集合 $\{V(\mathbf{v}_1), \dots, V(\mathbf{v}_n)\}$ をボロノイ分割と呼ぶ。

図8(b)はボロノイ分割に円をオーバーラップした図である。ここで2つの領域が交わっているときに対応する2つの頂点の間に線分を張り, 3つの領域が交わっているときに対応する3つの頂点に三角形を置く, というルールで単体複体を構築する。この例では図8(b)の P は3つの領域の共通点になっ

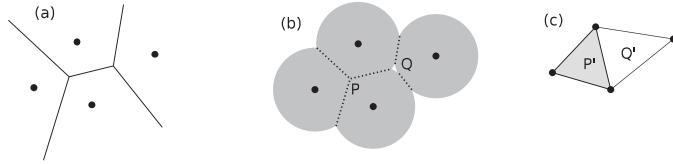


図 8: 4 点のポイントクラウドのアルファ複体 (a) ポイントクラウドとそのボロノイ分割 (b) ボロノイ分割と半径 r の円をオーバーラップさせたもの (c) アルファ複体

ているので図 8(c) で三角形 P' を置く. 一方図 8(b) の Q は 3 つの領域の共通点になっていないので図 8(c) では Q' は置かれぬ. 2 つの領域の交わりは図 8(b) では点線で表現されており, 図 8(c) で対応するところに線分があるのがわかるだろう. 3 次元上のポイントクラウドの場合はさらに「4 つの領域の交わり 三角錐」と対応させることで単体複体が構成できる. 数学的にきちんと定義するため, 領域 $R(\mathbf{v}_i, r)$ を

$$(4) \quad R(\mathbf{v}_i, r) = V(\mathbf{v}_i) \cap \bar{D}_r(\mathbf{v}_i)$$

で定義しよう. ただし $\bar{D}_r(\mathbf{v}_i)$ は中心 \mathbf{v}_i , 半径 r の閉円盤である. このとき, 半径パラメータ r のアルファ複体 $\text{Alp}(r)$ は

$$(5) \quad \text{Alp}(r) = \{\{\mathbf{v}_{i_0}, \dots, \mathbf{v}_{i_d}\} \mid R(\mathbf{v}_{i_0}) \cap \dots \cap R(\mathbf{v}_{i_d}) \neq \emptyset\}$$

という頂点の組み合わせで定義できる. 例えば図 8(b) は点 P で 3 つの領域が共通点を持つので, 図 8(c) に三角形 P' を置いている. 一方図 8(b) では Q の所で 3 領域が共通点を持っていないので図 8(c) には三角形 Q' が置かれていない. 2 つの領域が共通点を持つ所は図 8(b) で点線で表現されている部分で, 図 8(c) の対応する所に線分が置かれているのがわかるだろう.

アルファ複体の重要な性質として, 円/球の和集合とアルファ複体と同じトポロジカルな構造を持つことである. 図 8(b) と図 8(c) は確かに Q と Q' の所に 1 つ孔を持っており, 孔の数という意味では同じ情報を持っている. PH で考慮する情報はすべてアルファ複体が保持していることは数学的に証明されており (脈体定理と呼ばれる定理を用いる), 我々は安心してアルファ複体を利用してポイントクラウドの解析ができる. また半径パラメータ r を大きくすると単体複体のフィルトレーションができ, このフィルトレーションによって PH によるデータ解析が可能となる.

最後にアルファ複体に関して実用的な注意をしよう. アルファ複体は CGAL[19] のような計算幾何学のソフトウェアを用いれば数値的に計算可能である. しかし, 次のような条件が成り立ってしまうと数値的な不安定性が生じ, 結果が信頼できなくなる:

- 3 点が同一直線上にある
- 4 点が同一平面上にある (3 次元空間の場合)
- 4 点が同一円周上にある

- 5 点が同一球面上にある (3 次元空間の場合)

PH を用いるときにもこういった問題は生じる. 特に結晶構造を持つ場合に問題になる. そういった場合にはデータに微小ノイズを加えてこの条件を壊すと良い.

References

- [1] Ippei Obayashi. Volume-optimal cycle: Tightest representative cycle of a generator in persistent homology. *SIAM Journal on Applied Algebra and Geometry*, 2(4):508–534, 2018.
- [2] Ippei Obayashi. Stable volumes for persistent homology. *arXiv:2109.11711*.
- [3] Herbert Edelsbrunner and John Harer. *Computational topology: an introduction*. American Mathematical Soc., 2010.
- [4] Gunnar Carlsson. Topology and data. *Bull. Amer. Math. Soc.*, 46:255–308, January 2009.
- [5] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 28(4):511–533, Nov 2002.
- [6] Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, Feb 2005.
- [7] Joseph Minhow Chan, Gunnar Carlsson, and Raul Rabadan. Topology of viral evolution. *Proceedings of the National Academy of Sciences*, 110(46):18566–18571, 2013.
- [8] Yasuaki Hiraoka, Takenobu Nakamura, Akihiko Hirata, Emerson G. Escolar, Kaname Matsue, and Yasumasa Nishiura. Hierarchical structures of amorphous solids characterized by persistent homology. *Proceedings of the National Academy of Sciences*, 113(26):7035–7040, 2016.
- [9] Akihiko Hirata, Tomohide Wada, Ippei Obayashi, and Yasuaki Hiraoka. Structural changes during glass formation extracted by computational homology with machine learning. *Communications Materials*, 1(1):98, Dec 2020.
- [10] Emerson G. Escolar and Yasuaki Hiraoka. *Optimal Cycles for Persistent Homology Via Linear Programming*, pages 79–96. Springer Japan, Tokyo, 2016.
- [11] Tamal K. Dey, Tao Hou, and Sayan Mandal. Persistent 1-cycles: Definition, computation, and its application. In Rebeca Marfil, Mariletty Calderón, Fernando Díaz del Río, Pedro Real, and Antonio Bandera, editors, *Computational Topology in Image Context*, pages 123–136, Cham, 2019. Springer International Publishing.

- [12] Ippei Obayashi and HomCloud developer team. HomCloud. <https://homcloud.dev>.
- [13] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, Jan 2007.
- [14] Frédéric Chazal, David Cohen-Steiner, Marc Glisse, Leonidas J. Guibas, and Steve Y. Oudot. Proximity of persistence modules and their diagrams. In *Proceedings of the Twenty-Fifth Annual Symposium on Computational Geometry*, SCG '09, page 237–246, New York, NY, USA, 2009. Association for Computing Machinery.
- [15] Ulrich Bauer and Michael Lesnick. Induced matchings of barcodes and the algebraic stability of persistence. In *Proceedings of the Thirtieth Annual Symposium on Computational Geometry*, SOCG'14, page 355–364, New York, NY, USA, 2014. Association for Computing Machinery.
- [16] Michael Lesnick. The theory of the interleaving distance on multidimensional persistence modules. *Foundations of Computational Mathematics*, 15(3):613–650, Jun 2015.
- [17] Sébastien Le Rouxa and Valeri Petkova. Isaacs – interactive structure analysis of amorphous and crystalline systems. *Journal of applied crystallography*, 43(1):181–185, 2010.
- [18] P. Bendich, P. Bubenik, and A. Wagner. Stabilizing the unstable output of persistent homology computations. *J Appl. and Comput. Topology*, 4:309–338, 2020.
- [19] CGAL, Computational Geometry Algorithms Library. <https://www.cgal.org>.