TDA 맛보기

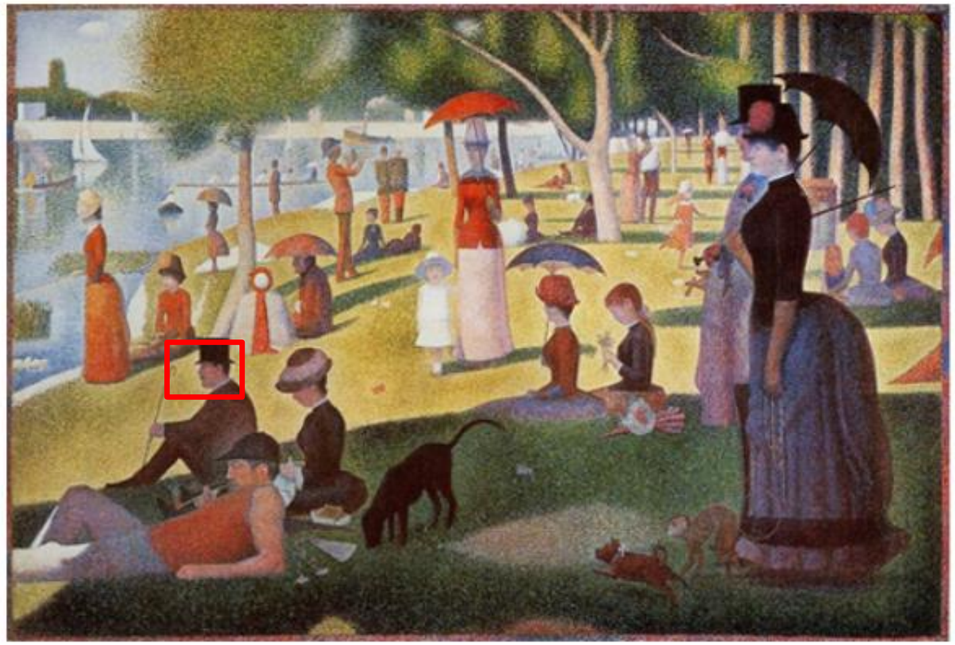
명서준

목차

목차

01 Motivation

- **Why use TDA?**



멀리서 보면 연속적인 곡선과 면으로 이루어진 그림 같지만
자세히 보면 무수히 많은 점으로 이루어져 있음

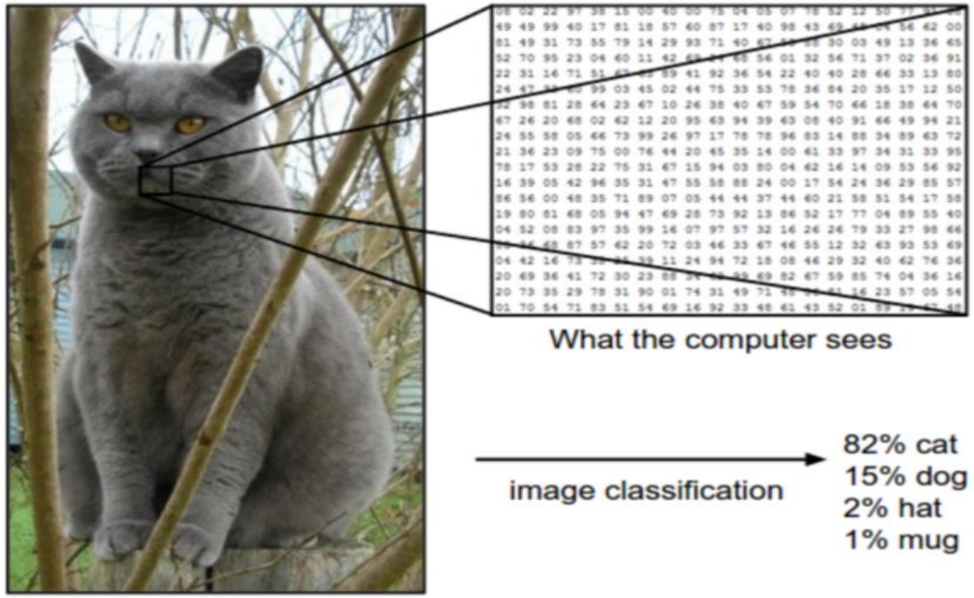
이렇게 사람의 눈과 뇌는 무수히 많은 점을 보고 이산적으로 느끼기 보다는 연속적으로 느낌

그러나 컴퓨터는 모든 것을 이산적으로 받아들임

- **Why use TDA?**

  어떻게 하면 컴퓨터가 이산적인 데이터로부터 연속적인 성질을 인식하게 할까?

  -> 데이터의 기하학적인 특징을 추출해내 보자! (graph, topology 등등..)



  이미지 데이터는 왜 지금까지 ai가 계산을 잘했을까?

  -> 이미지는 실세계에 있던 연속적인 대상을 이산적으로 잘 샘플링했다고 볼 수 있음!

- **Why use TDA?**

  결국, 사람에게는 이산적인 정보로부터 연속적인 대상을 생각하고
  그 성질을 인식하는 사고능력이 있음

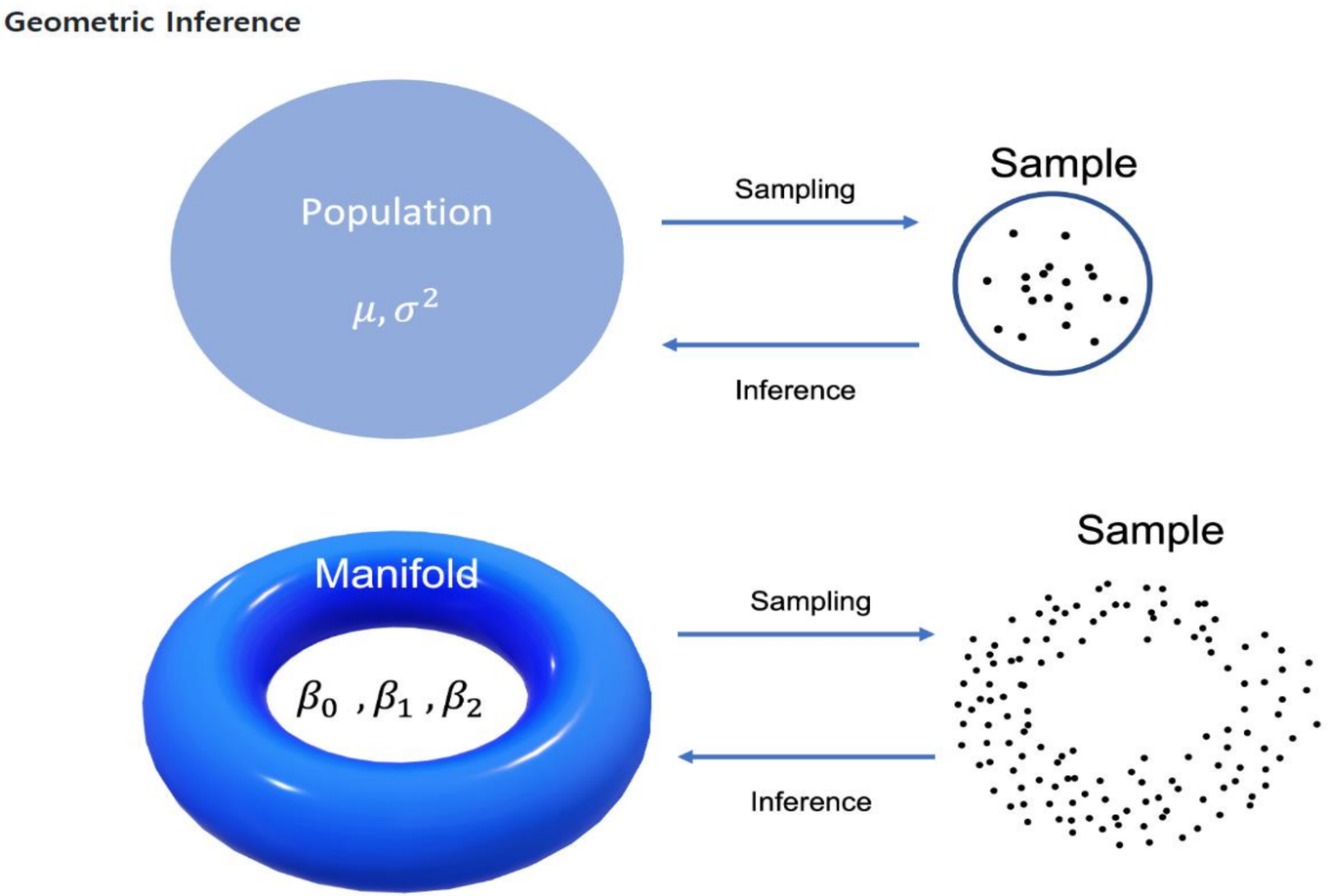
  이러한 사고과정을 이산적인 데이터가 사실은 어떤 미끄러운 기하학적 대상(Manifold)으로부터
  표본 추출된 것이라고 생각하고

  기하학적 대상이 가지고 있던 성질을 이산적인 데이터로부터 추론하는 것이라 할 수 있음

  -> 이러한 방법론을 기하학적 추론이라고 함

  우리는 이 기하학적 추론을 Persistent Homology를 이용해서 할 것임

- **Why use TDA?**


Geometric Inference
Population
μ, σ^2
Sampling
Inference
Sample
Manifold
β_0 , β_1 , β_2
Sampling
Inference
Sample

02 Simplicial Complex

- 양해바람^^

이 ppt는 포스텍 대학원생분이 작성한 블로그 글을 제 말로 풀어쓴 것에 불과..^^

위상수학은 연결성을 다루고자 한다는 철학만 마음속으로 받아들이고 가자..!

Q. TDA로 연구하는 것이 어렵나요.

A. "아니요. 대학 이공계 학과의 기초 과목인 '선형대수학'과 TDA에 쓰이는 특정 데이터 분석 도구만 공부하면 돼요. 실제로 우리 학교에서 TDA로 교양 수업을 진행하고 있는데, 3달 정도 공부하고 학생들이 팀을 이뤄서 프로젝트성 보고서를 쓰고 있어요. 그중 남녀 사이 연애 스타일이 회피형, 공격형, 통제형 중에 어떤 것인지 파악해서 시각화하는 재미있는 연구도 있어요."

그 외 많은 내용이 생략되어 있습니다..ㅎ

- **Def) Graph**

  Let V : 점들의 집합, **E :** 점들의 순서쌍 집합(⊆ $V \times V$)
  Then Graph G : (V,E).
  (이때 **V**의 원소 : **Vertex**, **E**의 원소 : **edge**)

- **Def) Simplicial Complex**

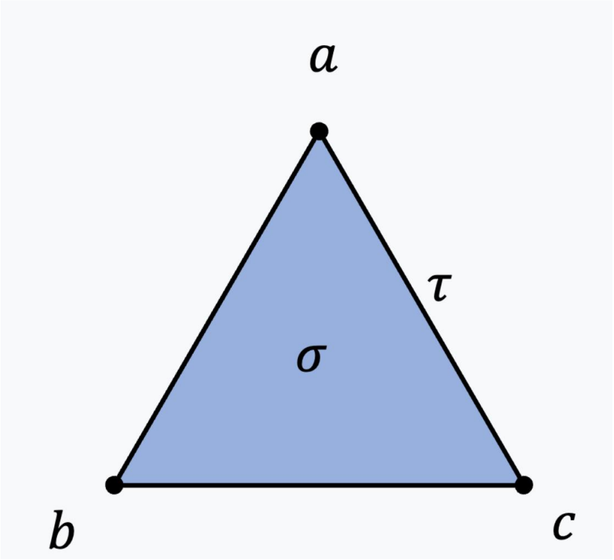
  Let some set K given.
  And let S be a collection of subset of K s.t.
  1) $\forall v \in K,\ \{v\} \in$ **S**
  2) $\tau \subseteq \rho \in$ **S** $\Rightarrow \tau \in$ **S**
  Then we say (K,S) is Simplicial Complex.

- **Def) Simplicial Complex**

  Let some set K given.
  And let S be a collection of subset of K s.t.
  1) $\forall v \in K,\ \{v\} \in$ **S**
  2) $\tau \subseteq \rho \in$ **S** $\Rightarrow \tau \in$ **S**
  Then we say (K,S) is Simplicial Complex.
  (문맥상 의미가 정확하다면, K를 Simplicial Complex라 부르기도 하고, S를 Complex 라고 부름.)

- **Ex) K =** $\{a, b, c\}$

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

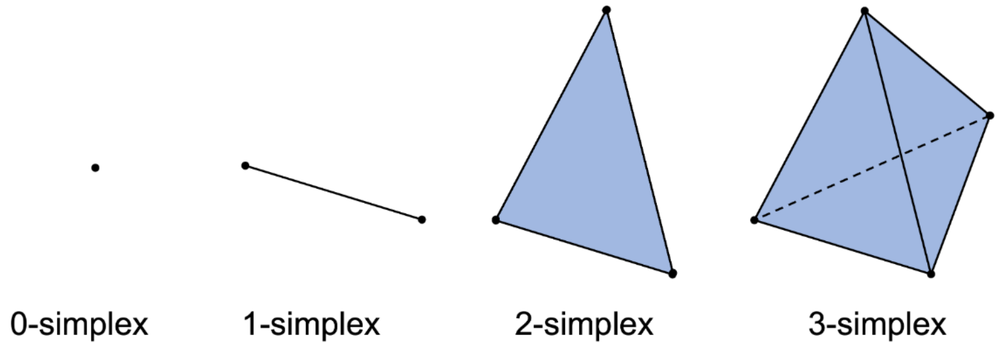


=> (K,S) : 삼각형의 연결성을 표현하는 **Simplicial Complex**

- **Simplicial Complex**는 그래프의 고차원 일반화라고 볼 수 있음
- 1번 조건 : **K**의 각 원소는 **Simplicial Complex**의 기본단위가 되는 점들의 집합이 **Simplex**가 된다.
- 2번 조건 : **Simplex** ρ 를 구성하는 파츠 τ가 있다면 τ 도 **Simplex**이다.

- **Def) Simplex의 차원**

  하나의 **Simplex**를 구성하는 데 필요한 꼭짓점의 수 -1



- **K-Simplex**를 유클리드 공간에 시각화 하는 규칙

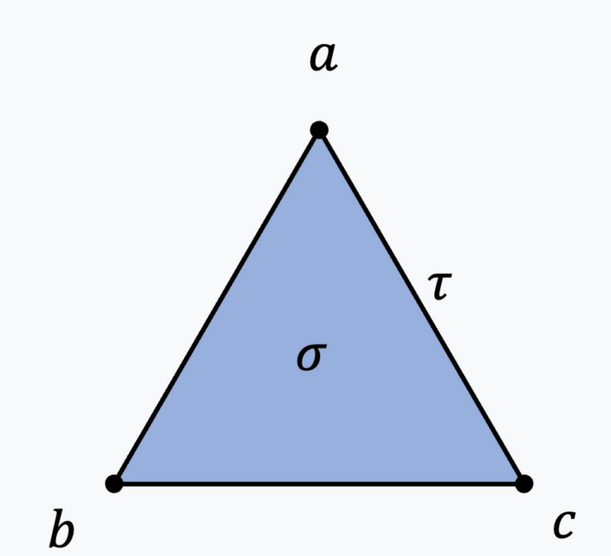
  1) **Affinely independent**
     K+1개의 점을 적절히 잘 흩어지게끔 (한 선분위에 3개 이상의 직선 X)
  2) **Convex hull**
     Simplicial Complex를 구성하는 **Simplex**의 공통면은 접하게끔

  => 좌표, 거리, 길이에 의존하지 않고 시각화 가능

  => 데이터를 무작정 유클리드 공간에 올리고 좌표계와 거리에 의존시키는 가정으로 부터 오는 오류를 줄임

- **Def) Sub Complex**

Let K is simplicial complex.
If L(⊆K) is simplicial complex, we say L is subcomplex.



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

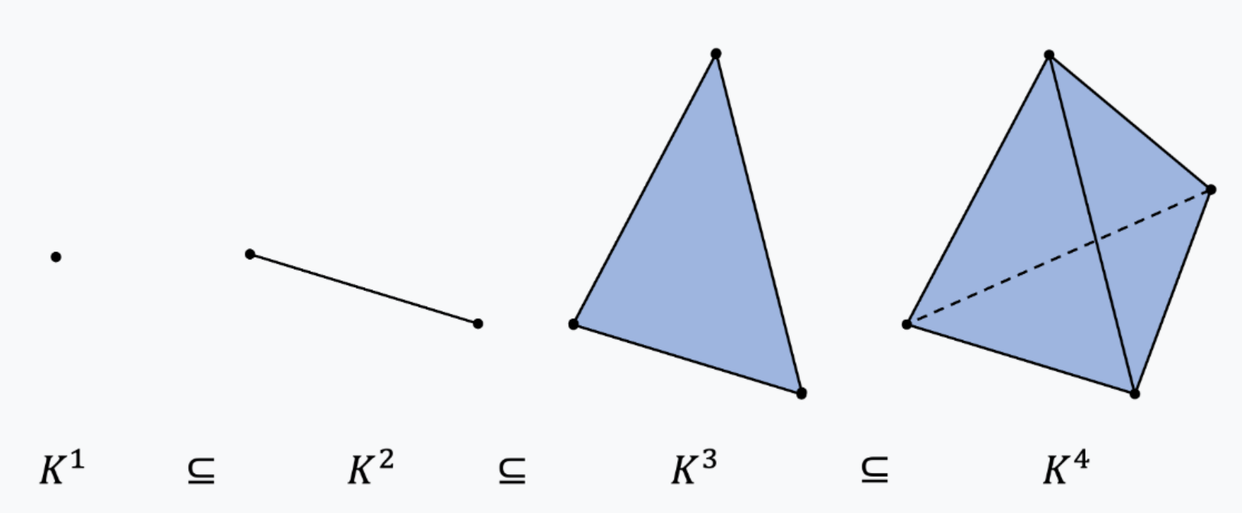
L = $\{a, c\}$
τ **=** $\{\ \{a\}, \{c\}, \{a, c\}\ \}$

주어진 **Simplicial Complex**의 **subcomplex**의 **subcomplex**의 **subcomplex**..를 생각해보자

언젠간 공집합이 나타날 것이고, 이들끼리는 한방향으로 포함관계가 있을 것임.

- **Def) Filtration**

Let K is simplicial complex.
Then we say $\emptyset = K^0 \subseteq K^1 \subseteq K^2 \subseteq \cdots \subseteq K^m = K$ is filtration.
(nested subsequence of subcomplex of K)



- **Why use Filtration?**

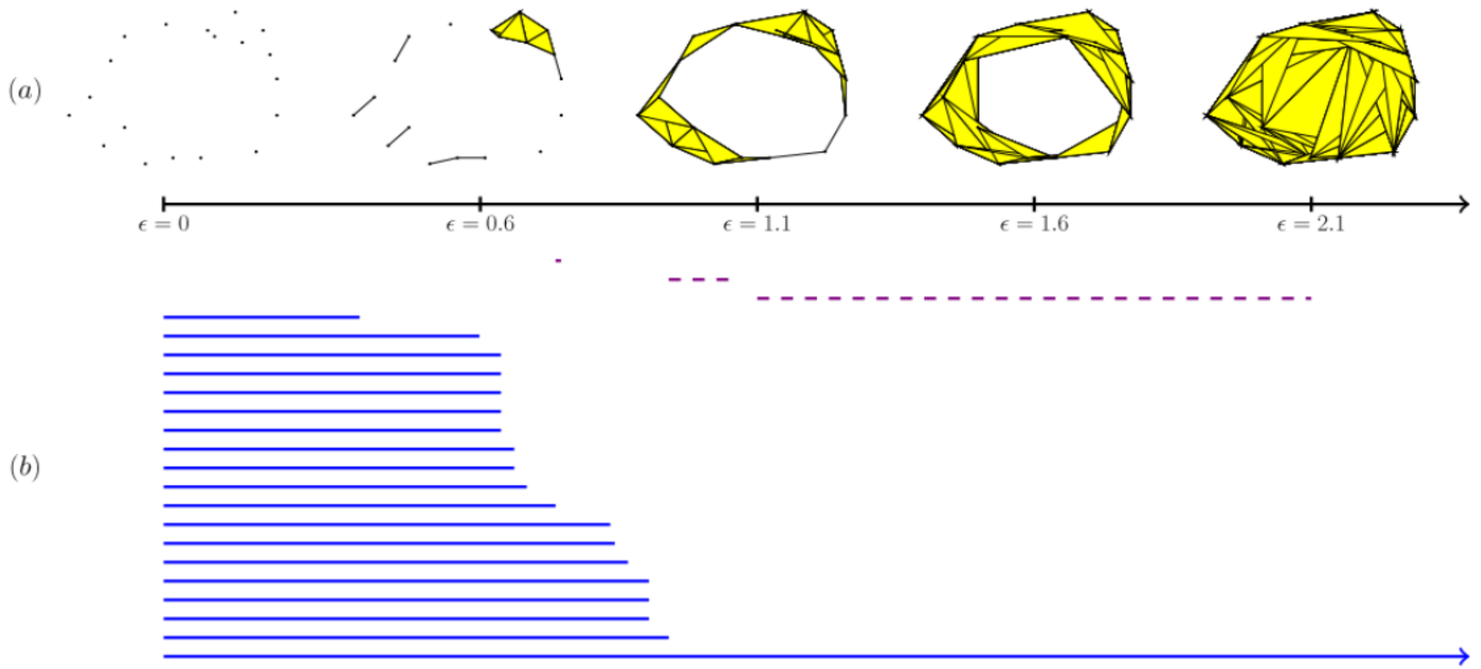
만약 거대한 데이터 집합을 **simplicial complex**로 구현한다면 그것은 굉장히 복잡한 형태일 것이고, 한번에 관찰하기 쉽지 않을 것임.

수학에서는 이런 복잡한 구조를 관찰하고자 할 때, 이들의 부분집합들을 분석해나가는 방법을 이용

Filtration은 **simplicial complex**의 부분집합들을 점점 커지도록 나열하여 현재의 **simplicial complex**가 어떠한 과정을 통해 만들어 졌는지를 표현함

TDA에서는 이러한 **simplicial complex**의 진화과정 속에서 연결성분이나 구멍과 같은 기하학적 정보가 얼마나 지속(**Persistent**)되는지를 측정하고 이를 데이터의 중요한 특징으로 활용

- **Why use Filtration?**



TDA는 **filtration**을 이용해 데이터로부터 **simplicial complex**의 진화과정을 만들고 그 과정 중에 연결성분(파란막대)나 구멍(점선)의 지속성을 측정한다.

- **How construct Filtration?**

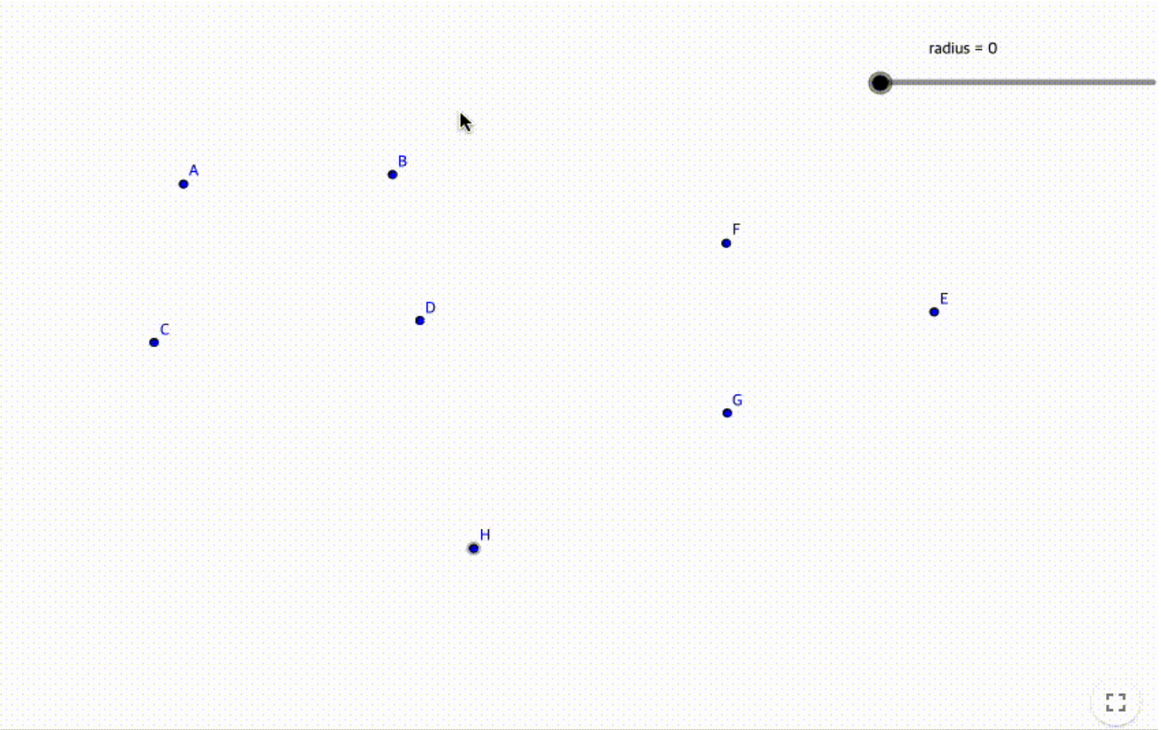
포인트 클라우드 데이터 집합 K가 주어졌다고 하자.
$\epsilon \in [0, \infty)$에 대하여 다음 알고리즘을 생각하자.

1. 거리 파라미터 ϵ을 선택한다.
2. 두 점 사이의 거리가 2ϵ 이하가 되면 두 점을 변으로 연결한다.
   (즉, 각 점을 중심으로 반지름이 ϵ인 원을 그려 두 원이 만나면 변으로 연결한다.)
3. 세 점에 대하여, 각각 두 점 사이의 거리가 2ϵ 이하가 되면 세 변을 삼각형으로 연결한다.
   (즉, 세 점이 변으로 서로 연결되어 삼각형을 그리면 그 삼각형을 면으로 채운다.)

그러면 각 ϵ에 대하여, K를 꼭짓점으로 하는 **Simplicial Complex** K_ϵ이 만들어진다.
(이렇게 만들어진 **simplicial complex**를 **Rips Complex**라 부른다.)

이때 ϵ들의 수열 $0 = \epsilon_0 \leq \epsilon_1 \leq \epsilon_2 \leq \cdots \leq \epsilon_n$을 택하면
Filtration $\emptyset = K_{\epsilon 0} \subseteq K_{\epsilon 1} \subseteq K_{\epsilon 2} \subseteq \cdots \subseteq K_{\epsilon n} = K$을 얻는다.
(이렇게 만들어진 **filtration**을 **Rips filtration**이라 부른다.**)**

- **How construct Filtration?**



이렇게 **TDA**에서 제일 먼저 해야할 것은 데이터의 **filtration**을 구축하는 것.

이를 위해서 **simplicial complex**라는 기하학적 대상을 계산이 가능한 대수적 대상으로 대응 시키는 일이 필요. => **Homology**의 등장.

01 Motivation

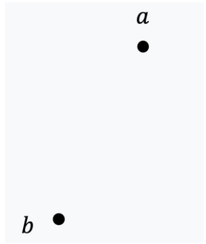
02 Simplicial Complex

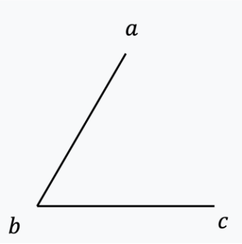
03 Homology

04 Persistent Homology

05 Python Tutorial

- **Def) Chain Module**





Let simplicial complex K be given.

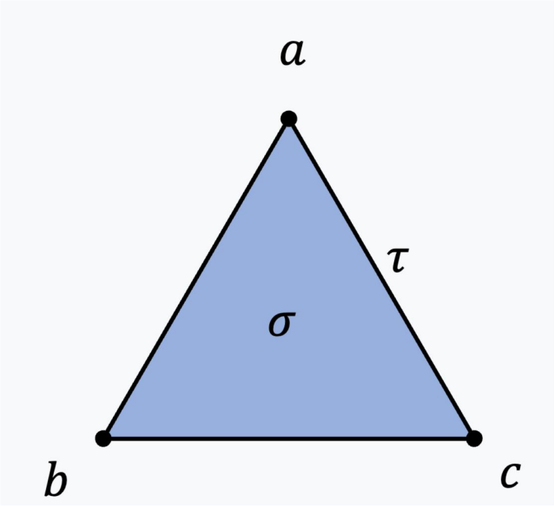
Let + : 병렬배치

Let Σ_r : r-simplex들의 집합

Let $C_r(K) = \{a_1x_1 + a_2x_2 + \cdots + a_nx_n \ : \ a_i \in F_2, \ x_i \in \Sigma_r \ \ for\ all\ i\}$

(Σ_r 를 basis로 하고, $F_2 = \{0, 1\}$을 계수(filed)로 하는 Vector Space!!)

We say $C_r(K)$ is r-th chain module of K and the sequence of $C_r(K)$ is chain complex.



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

$\Sigma_0(K) = \{ \{a\}, \{b\}, \{c\} \}$
$\Sigma_1(K) = \{ \{a, b\}, \{b, c\}, \{a, c\} \}$
$\Sigma_2(K) = \{ \{a, b, c\} \}$

- **Def) Chain Module**

  우리는 지금 기하학적 구조에 선형대수라는 도구를 이용하려고 준비중..!!

  Simplicial Complex의 꼭짓점, 변, 면을 따로 떼어서 **Vector Space** $C_0(K)$, $C_1(K)$, $C_2(K)$를 만듦

- **Notation)**

  K-simplex $\sigma(\in \Sigma_k)$가 **k+1**개의 꼭짓점 $v_0, v_1, \cdots, v_k$에 의해 결정된다고 하자.

  이 경우 $\sigma = [v_0, v_1, \cdots, v_k]$로 표기한다.

  $\sigma = [v_0, v_1, \cdots, v_k]$에서 꼭짓점 v_i를 제거해 만든 **(k-1) simplex**는 $[v_0, v_1, \cdots, \hat{v}_i, \cdots, v_k]$라 하자.

- **Thm) Linear Extension Theorem**

  Let $\{v_0, v_1, \cdots, v_n\}$ **be the basis of Vector Space V and** $\{w_0, w_1, \cdots, w_n\}$ **be the element of W.**
  Then $\exists! \, T : V \rightarrow W \; s.t. \; T(v_i) = w_i$ **for all i**

  ≫ 위 정리에 따라서 **Chain Module**의 기저가 되는 **K-Simplex**를 어디로 대응시켜 줄지만 정하면 선형변환 정의 가능

  ≫ 수학자들은 높은 차원의 **simplex**가 낮은 차원의 **simplex**로 둘러싸여 있다는 점에 주목했음

- **Def) Boundary Operator**

  Let simplicial complex K be given and C_k, C_{k-1} be the consecutive chain module of K

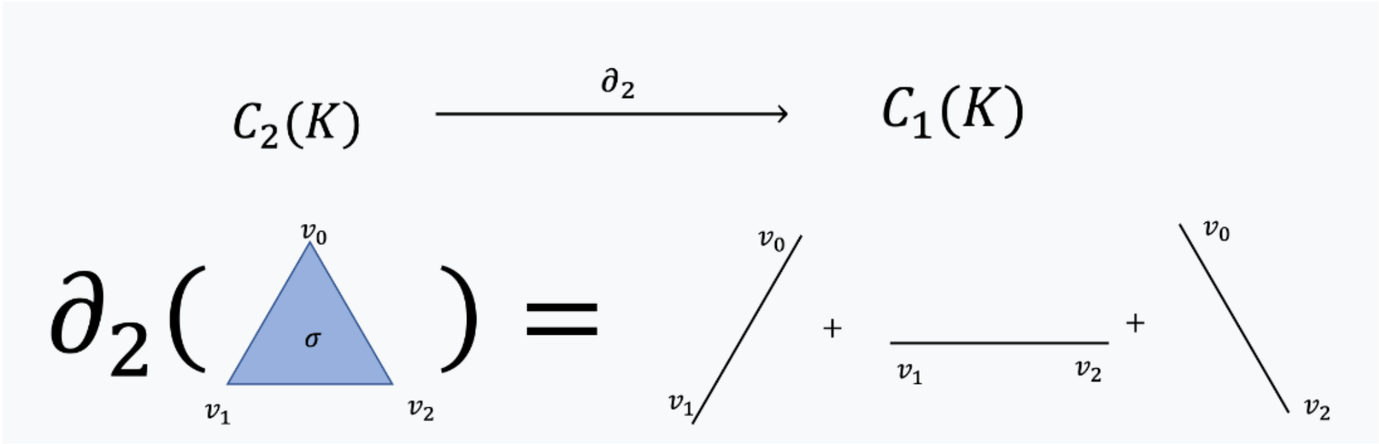
  Let define Boundary Operator $\partial_k : C_k \to C_{k-1}$ as

$$\partial_k(\sigma) = \sum_{i=0}^{k} [v_0, v_1, \cdots, \hat{v}_i, \cdots, v_k] \ \textit{for all}\ \sigma = [v_0, v_1, \cdots, v_k] \in \Sigma_k$$

- **Ex)**

  Let k-simplex $\sigma = [v_0, v_1, v_2] \in C_2(K)$ be given.

  Then $\partial_k(\sigma) = [v_1, v_2] + [v_0, v_2] + [v_0, v_1] \in C_1(K)$



- 정리!!

  - **TDA**는 데이터의 연결성을 표현하기 위해 **Simplicial Complex**라는 개념을 사용한다.

  - 복잡한 연결성을 분석하기 위해 작은 **Simplicial Complex**에서 부터 점점 진화시켜나가는 **Filtration**이라는 기법을 사용한다.

  - **Simplicial Complex**를 분석하기 위해 각 차원별 **Complex**로 분리해 **Chain Module**이라는 벡터공간을 정의했다.

  - **Chain Module**들을 **Boundary Operator**로 연결시켰다.

- **Coment**

  - 우리는 선형대수를 이용해 우리의 직관 속에 있던 **Simplex**를 넣으면 그 테두리를 뱉어주는 함수를 쉽게 정의 할 수 있었다.

  - 선형대수가 가진 또 하나의 강력한 힘은 선형변환을 행렬로 표현할 수 있다는 점이다.
    (인간의 언어로 표현된 선형변환 이란 규칙을 행렬로 바꿔 컴퓨터가 계산하기 쉽게 했다.)

  - 이것이 선형대수를 공부해야 하는 이유이고 **TDA**가 데이터분석에 응용될 수 있는 실용적인 이유이다.

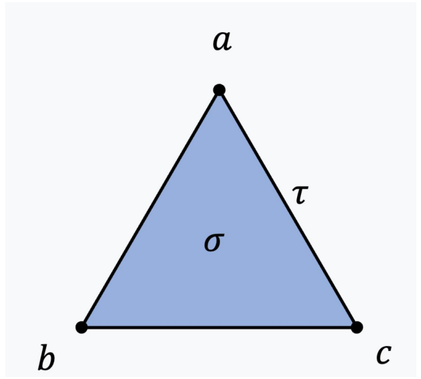
- **Def) Cycle group and Boundary group**

  Let ∂_k be Boundary Operator.

  Then we say $Z_k = ker(\partial_k)$ is k-th cycle group
  $B_k = \text{img}(\partial_{k+1})$ is k-th boundary group.

  And we say element of Z_k is k-cycle
  element of B_k is k-boundary.

- **Ex)**



$$Z_1 = \{\sigma \in C_1 \mid \partial_1(\sigma) = 0\}$$
$$= \{\, t_1[a,b] + t_2[b,c] + t_3[c,a] \mid t_1, t_2, t_3 \in F_2, \partial_1(\sigma) = 0\}$$

$$\partial_1(\sigma) = t_1([a] + [b]) + t_2([b] + [c]) + t_3([c] + [a])$$
$$= (t_1 + t_3)[a] + (t_1 + t_2)[b] + (t_2 + t_3)[c]$$
$$= 0$$

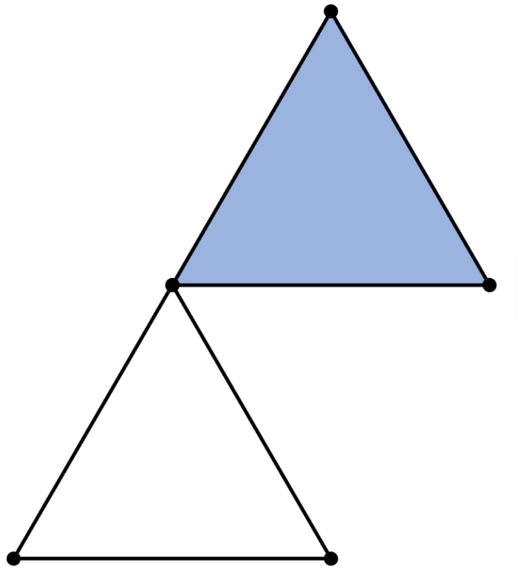
$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} t_1 \\ t_1 \\ t_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$\rightarrow t_1 = t_2 = t_3 \rightarrow Z_1$의 기저 : $[a,b] + [b,c] + [c,a] \rightarrow$ **cycle group** 이라 부름

정의에 따라 B_1은 C_2의 **boundary**가 나옴으로 **boundary group**이라 부름

Note that $B_k = \text{img}(\partial_{k+1}) \subseteq ker(\partial_k) = Z_k$

- **Let's Think ..**



다음과 같은 **Simplicial Complex**가 주어졌다고 가정하자.

이 그림에서 구멍의 개수는**?** **1**개

이를 어떻게 수학적으로 정의할 수 있을까**?**

위쪽 삼각형은 세 변이 하나의 면으로 연결되어 있지만 아래쪽 삼각형은 그렇지 않다.

즉, 이 **simplicial complex**에서는 위쪽 삼각형만이 **2-simplex**를 가지고
아래쪽 삼각형은 이의 테두리에 해당하는 **connected 1-simplex**만을 가진다.

그럼 이 **simplicial complex**의 **1-cycle**과 **1-boundary**는 각각 무엇이 될까**?**

그럼 **1-cycle**의 개수에서 **1-boundary**의 개수를 빼면 구멍의 개수가 나오겠네**!?**

이렇게 얻어진 구멍들로 만들어진 공간을 **Homology**라고 한다.

- **Def) Homology**

Let simplicial complex K be given and Z_k is k-th cycle group of K
and B_k is k-th boundary group of K
Then k-th homology of K is defined as

$$H_k(K) \coloneqq Z_k \,/\, B_k$$

- **Def) Betti number**

K-th Betti number of K is dimension of $H_k(K)$

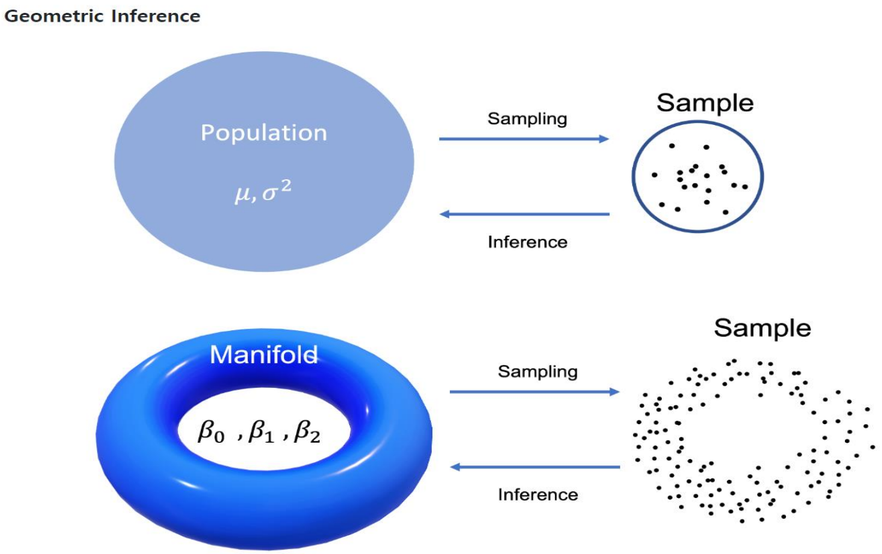
$$\beta(K) \coloneqq \dim(H_k(K)) = \dim(Z_k) - \dim(B_k)$$

(simplicial complex K가 가진 **k**차원 구멍의 개수**)**

우리는 이제 남들이 **homology**가 뭐죠**?** 라고 물으면
Simplicial complex의 구멍을 계산해주는거요^^ 라고 답할 수 있게 되었다...

- **Motivation)**

Recall p.17~19



통계적 추론 :
sample data가 어떠한 분포를 따르는 모집단으로 부터 왔다는 가정 후 모집단의 분포가 가지는 특성을 합리적으로 추론

기하학적 추론 :
sample data가 어떠한 **mani fold**(고차원 기하학적 대상)으로 부터 왔다는 가정 후 **mani fold**가 가지는 특성을 합리적으로 추론

그런데.. 저 데이터들을 무작정 다 연결하는게 옳을까..?

→ 실제 데이터엔 노이즈가 섞여 있어 한 공간 위에 모두 놓여있을 수 **X**

→ 그럼 얼마나 가까울 때 연결해야 해?

04 Persistent Homology

- **Motivation)**

TDA가 내놓은 답

1. 데이터들을 조금씩 연결시켜 가면서 **Simplicial Complex**를 성장시켜 나가자.

2. 그 과정에서 새롭게 태어나거나**(born)** 사라지는**(die) Homology**의 생성원을 관찰하자.

3. 이때 오래 지속되는**(persistent) Homology**의 생성원들로부터 **manifold**의 기하학적 정보를 추론하자.

여기서

1. Simplicial Complex의 성장 과정은 **Filtration**의 구축으로 : $K_0 \subseteq K_1 \subseteq K_2 \subseteq \cdots \subseteq K_n$

2. 그 후 각각의 K_i에 대해 $H_k(K_i)$ 계산

그런데...

3. 각각의 K_i에 대해 계산한 **Homology**는 서로 독립적인거 아니야**?**

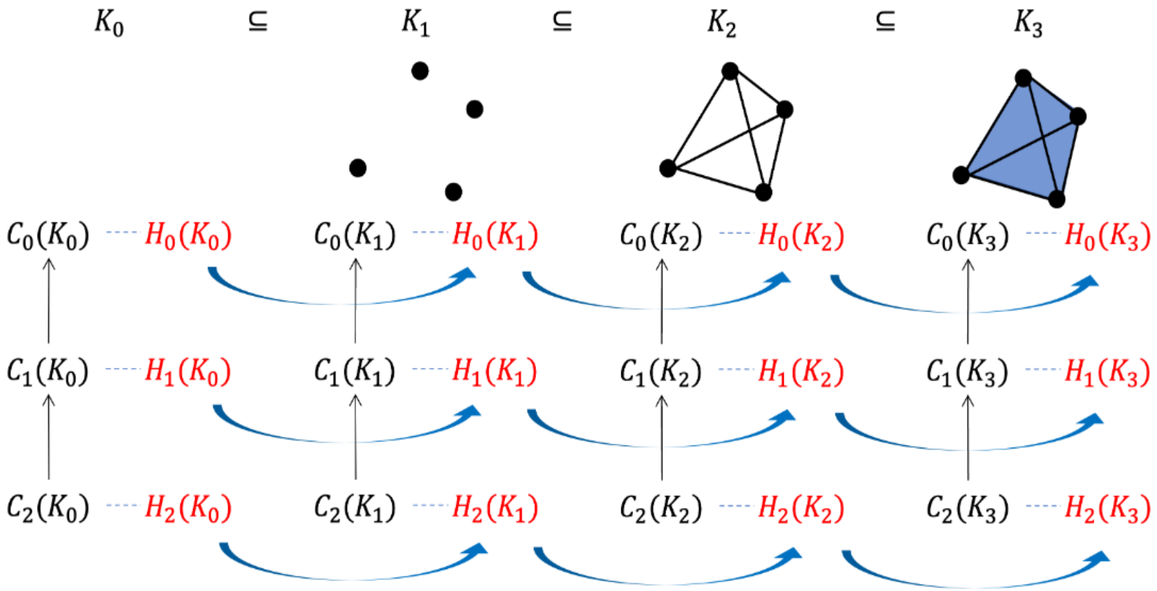
→ 이것을 해결해주는 것이 **Persistent Homology**

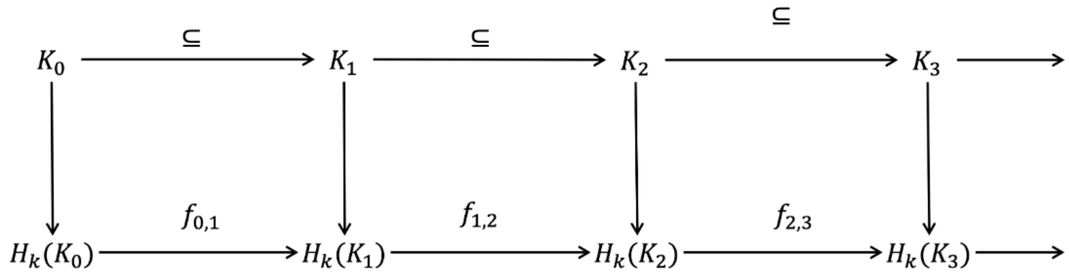
(persistent homology는 서로 다른 sub complex들로 얻어진 각각의 독립적인 homology들을 연결시켜줌 **)**

(또한 모든 scale에 대해 정보를 획득하기 때문에 얼마나 가까이 있는 점들을 연결해야 하는지도 걱정할 필요 X **)**

- **Motivation)**

  우리한테는 총 두 종류의 수열이 주어져있음 : **Filtration, Chain Complex**





By Category Theorem...