

Statistics on the Space of Persistent Diagrams with Applications



By

Waqar Hussain Shah

CIIT/FA19-RMT-067/ISB

MS Thesis

In

Mathematics

COMSATS University Islamabad, Islamabad Campus

Pakistan

Spring, 2021



COMSATS University Islamabad

Statistics on the Space of Persistent Diagrams with Applications

A Thesis Presented to
COMSATS University Islamabad
In partial fulfillment
of the requirement for the degree of

MS Mathematics

By
Waqar Hussain Shah
CIIT/FA19-RMT-067/ISB
Spring, 2021

Statistics on the Space of Persistent Diagrams with Applications

A Post Graduate Thesis submitted to the Department of Mathematics as partial fulfillment of the requirement for the award of Degree of MS in Mathematics.

Name	Registration No.
Waqar Hussain Shah	CIIT/FA19-RMT-067/ISB

Supervisor

Dr. Sohail Iqbal
Department of Mathematics,
COMSATS University Islamabad,
Islamabad Campus.
July, 2021

Final Approval

This thesis titled

Statistics on the Space of Persistent Diagrams with Applications

By

Waqar Hussain Shah

CIIT/FA19-RMT-067/ISB

Has been approved

For the COMSATS University Islamabad

External Examiner: _____

Dr. Muhammad Aslam,
Associate Professor,
Department of Mathematics,
Quaid-e-Azam University Islamabad.

Supervisor: _____

Dr. Sohail Iqbal
Assistant Professor,
Department of Mathematics,
COMSATS University Islamabad,
Islamabad Campus.

HOD: _____

Dr. Abdullah Shah
Associate Professor,
Department of Mathematics,
COMSATS University Islamabad,
Islamabad Campus.

Declaration

I, **Waqar Hussain Shah** registration no. **CIIT/FA19-RMT-067/ISB**, hereby declare that I have produced the work presented in this thesis, during the scheduled period of study. I also declare that I have not taken any material from any source except referred to wherever due that amount of plagiarism is within acceptable range. If a violation of HEC rules on research has occurred in this thesis, I shall be liable to punishable action under the plagiarism rules of the HEC.

Date: _____

Signature of student:

Waqar Hussain Shah
CIIT/FA19-RMT-067/ISB

Certificate

It is certified that **Waqar Hussain Shah**, registration no. **CIIT/FA19-RMT-067/ISB** has carried out all the work related to this thesis under my supervision at the Department of Mathematics, COMSATS University Islamabad and the work fulfills the requirement for award of MS degree.

Date: _____

Supervisor:

Dr. Sohail Iqbal
Department of Mathematics,
COMSATS University Islamabad,
Islamabad Campus.

Head of Department:

Dr. Abdullah Shah
Department of Mathematics,
COMSATS University Islamabad,
Islamabad Campus.

Dedicated to my great parents,
especially to my supervisor, sibling,
and friends

Acknowledgement

In the name of ALLAH, the Most Gracious and the Most Merciful, all praise is for ALLAH; we praise Him, seek His help, and ask for His forgiveness. I am thankful to ALLAH, who gave me the courage, the guidance, and helped me throughout in completing the research. Also, I cannot forget the ideal man of the world and most respectable personality for whom ALLAH created the whole universe, Prophet Muhammad (Peace Be Upon Him).

Foremost, I would like to express my heartfelt gratitude to the most cooperative and good communicator my supervisor **Dr. Sohail Iqbal**, Assistant Professor, COMSATS, Islamabad who suggested the problem. The doors towards the supervisor were always opened whenever I ran into a trouble spot. His guidance helped me in all the time of research and writing of this thesis. He has always helped me out and tolerated my untimely disturbances. He consistently allowed this thesis to be my work but steered me in the right direction whenever he thought I need it. I express my sincere respect to all instructors in the department of Mathematics for their kind cooperation.

I am sincerely thankful to HOD, **Dr. Abdullah Shah**, Department of Mathematics, COMSATS for providing a favorable environment in which the students can conduct their projects and research work. I must express my very profound gratitude to my dear parents, siblings, friends (**Muhammad Adil Habib & Laeeq Ahmed**), and roommates for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

Waqar Hussain Shah

ABSTRACT

Statistics on the Space of Persistent Diagrams with Applications

Data is shape and shape is data. We use Topological Data Analysis (TDA) to extract information from the data, but also TDA assumes that data have a shape. TDA uses principles from algebraic topology to comprehensively measure shape in data sets. Using a function that relates the similarity of data points to each other. We can monitor the evolution of topological features—connected components, loops, and voids. One of the efficient techniques of TDA is Persistent Homology. The principle aim of persistent homology is to study the key features of the dataset. The homological functor that is used in the development of persistent homology is simplicial homology. The homology features, that remain stable across scales are the ones that provide effective analysis of the shape of the point cloud.

Eventually, we are applying the TDA in bio-medical field. In this work, We are quantifying the topological properties based on French-American-British Classification, a shaped standard for leukemia cells. Persistent Homology estimates the homological key features of the cells and smooth Euler characteristic transform describe the boundary of the cell images.

Contents

1	Inroduction	1
1.1	Preliminaries	3
1.2	Simplicial Homology	5
1.2.1	Homological Algebra	12
1.3	Inner Framework of Simplicial Complex	14
1.4	Building Simplicial Simplexes	16
1.5	Techniques in Topological Data Analysis	20
1.5.1	Persistent Homology	21
1.5.2	Persistent Diagram	26
2	Metrics on the Space of Persistent Diagrams	30
2.1	Distances	31
2.2	Persistent Landscape	32
2.2.1	Persistent landscape properties	35
2.3	Persistence Image	36
2.3.1	Stability of Persistence Images and Surfaces	38
2.4	Riemannian Framework	38
2.5	Persistent Homology Transform	41
2.6	Smooth Euler Characteristic Transform	44
2.6.1	Euler characteristic transform	46
3	Computational Techniques for Persistent Homology	48
3.1	JavaPlex	51
3.2	Ripser	52
3.3	Dinoysus	54
3.4	Perseus	54
3.4.1	Persistent Homology of Vietoris-Rips Complexes	55

3.5	PHAT-Persistent Homology Algorithm Toolbox	55
3.6	GUDHI	56
3.7	DIPHA	56
3.8	Persim	57
3.9	Comparison Analysis	58
4	Morphological and Geometrical Features	60
4.1	Features Extraction by Image processing	61
4.1.1	Image Representation	63
4.1.2	Image Preprocessing	63
4.2	Image Optimization	65
4.3	Image Analysis	66
4.4	Image Compression	67
4.5	Morphological features	67
4.6	Texture features	69
5	Classification of Acute Lymphoblastic Leukemia Cells	72
5.1	Basic Hematology	73
5.2	FAB (French-American-British) Classification	74
5.3	Dataset of C-NMC-2019	76
5.4	Image Segmentation	79
5.5	Results of smooth Euler characteristics trnsform	83
5.6	Machine Learning and Topological Data Analysis	84
5.6.1	Supervised Machine Learning	85
5.6.2	Unsupervised Machine Learning	85
5.7	Classification Evaluation	87
5.7.1	Evaluation Metrics	88
5.8	Results for Cancer Classifier	89
5.9	Results for Leukemia Cells	90
5.10	Conclusion and Future Work	90
A	Homologies Computations	95
A.1	Simplicial	96
A.2	Distances Computation	96

A.2.1	Hausdorff Distance	96
A.2.2	Bottleneck distance	96
A.2.3	Sliced Wasserstein distance	97
A.2.4	Wasserstein distance	97
A.2.5	Heat Kernel Distance	97
A.3	Persistent Landscape	97
A.4	Persistence images	98
A.5	Riemannian framework	98
A.6	Persistent Homology Transform	98
A.7	Smooth Euler Characteritics Transform	98
B	TDA-Packages	100
B.1	Javaplex	101
B.2	Ripser	101
B.3	Dinoysus	101

List of Figures

1.1	Counterclockwise trip and clockwise trip	6
1.2	Simplicial complex	7
1.3	A complex	8
1.4	Simplicial homology	9
1.5	The Five lemma	13
1.6	Exact sequence of chain complexes	14
1.7	Star of a simplex	15
1.8	Closure of a simplex	15
1.9	Link of a simplex	15
1.10	Point cloud and Simplicial complex	16
1.11	Nerve of a covering	16
1.12	\hat{C} ech complex	17
1.13	Vietoris-Rips complex	17
1.14	Cubical Complex	20
1.15	Points in space	22
1.16	Covering of space with ϵ balls	22
1.17	Chain map	24
1.18	Filtrations and barcodes for ϵ increasing horizontally	26
1.19	Point cloud datasets	27
1.20	Persistent diagrams for 1.19	27
1.21	Life span of features	28
2.1	Persistent landscape	34
2.2	Persistence Image	38
2.3	Multivariate Normal Distribution	39
2.4	0th-dimensional PD to the letter W filtrations	43
2.5	Euler Characteristic table	45

2.6	(EC) injectivity for fixed direction.	47
3.1	computational steps for persistent homology	50
3.2	Barcodes using JavaPlex	51
3.3	Lifetime of generators	52
3.4	Framework for Ripser and Ripser++	53
4.1	An infected blood cell in 2D	62
4.2	An infected blood cell in 3D	62
4.3	Erosion of an Image	64
4.4	Dilation of an Image	64
4.5	scale=0.	66
4.6	Correlation graph	71
5.1	Prototypical Human Cell	73
5.2	Subtypes of Acute Lymphoblastic Leukemia	74
5.3	Analytical description of hematological disorders of leukocytes	75
5.4	C-NMC-2019-Dataset samples, (a) to (c)are malignant, and (d) to (d) are healthy cells	76
5.5	FAB Classification	78
5.6	Segmented image	81
5.7	Segmented Nucleus and Cytoplasm of a Cell	81
5.8	Point cloud from segmented images	82
5.9	Normal cell and their Euler curves	83
5.10	Normal cell and their Euler curves	83
5.11	Leukemia cell and their Euler curves	84
5.12	Leukemia cell and their Euler curves	84
5.13	Clearly labeled data as circles and crosses	85
5.14	The clustering are being formed from unlabeled data	85
5.15	SVM separating the data of blue and red dots.	86
5.16	Data points, covered with dashed circles shows the support vectors that are the data points touching the boundaries of marginal planes.	87

List of Tables

3.1	Elapsed time in seconds for each package	58
3.2	Topological Data Analysis Tools	59
4.1	Morphological features of Normal and ALL cells	69
4.2	Texture features of Normal and ALL cells	71
5.1	Description of C-NMC-2019 Dataset	77
5.2	Results for Cancer	89
5.3	Results for Leukemia cells	90

Chapter 1

Introduction

Presently in every field of life, we are investigating with a data set that is huge in amount. This massive data carries a lot of information. In this age, the main objective is to analyze the data in a brainy way and extract the appropriate information from it. In our concern, we will analyze data either mathematically or statistically. We will extract the fruitful information and make our further decision on it. In the current era, there are many approaches available in the field of mathematics and statistics, but we are interested in evaluating the data by their shape and geometry. The technique that we are using is a subfield of Algebraic Topology that is Topological Data Analysis (TDA). TDA offers to explain the geometry and the topological features of the quantitative data. TDA is a technical way to scrutinize noisy data and extracting worthy information from it. TDA develops a topological method for the analysis of noisy data and gives us statistical tools.

The technique that we are using is Persistent Homology (PH) from TDA. PH is a flagship tool of TDA. PH is a homology theory that is used to study the qualitative characteristic of the dataset. It computes a ‘persistent diagram’ and there will be no change in topological features. Simplicial Homology provides a mathematical language for holes in the topological space. In Algebraic Topology combination of points, line segments, triangles, and also higher-dimensional counterparts generate a Simplicial Complex.

There are many kinds of simplicial complexes such as Čech, Vietoris-Rips, weak witness, strong witness, and Delaunay complexes. Homology of the simplicial complex gives the algebraic measure based on cycles that are not boundaries. In Simplicial Homology, we mainly deal with the simplicial complex. The main knowledge is to triangulate the space which is under consideration, which will help us in calculating the homology of such spaces under some conditions. The idea of simplicial homology is coming out from the Algebraic Topology in which the topological spaces are homeomorphic to each other if they have the same number of holes. The main features of these complexes are they are also algorithmically computable.

In the computational era, there are many techniques and algorithms are available, we will use effective algorithms for more accurate results. Also, in computational topology especially in TDA, there are many ways to study logical data. The technique of persistent landscape, Riemannian frameworks, and smooth Euler characteristic transform (SECT). In a Persistent landscape, persistent diagrams are mapped to function space or even a Hilbert space. There are computational techniques to compute Persistent Landscape and that allows to apply tools from machine learning and statistics. With the second technique, a framework is built based on Riemannian geometry in which persistent diagrams (PDs) are expressed as a 2D probability density function by applying the kernel density function. SECT will use a collection of smooth curves to summarize statistics, and then we will be quantifying these curves. SECT permits the execution of present models based on statistics in functional data analysis (FDA); basically, it agrees to use the information of tumor shape in the frameworks of the regression process as a covariate.

In Bio-medical field TDA have many applications. Persistent homology evaluates the homological key features of the cell images. Further more we are applying these topological data analysis technique to classify the C-NMC-2019 leukemia dataset. Acute lymphoblastic leukemia belongs to the category of blood cancer.

1.1 Preliminaries

Definition 1.1.1 (Category). A category \mathcal{C} consists of three ingredients

1. A collection of objects in \mathcal{C} , denoted by set $\text{obj}\mathcal{C}$,
2. Morphisms between objects $\text{Mor}(\theta, \zeta)$ in \mathcal{C} , for every ordered pair $\theta, \zeta \in \text{obj}\mathcal{C}$,
3. Composition of morphisms $\text{Mor}(\theta, \zeta) \times \text{Mor}(\zeta, \eta) \rightarrow \text{Mor}(\theta, \eta)$ in \mathcal{C} , for every $\theta, \zeta, \eta \in \text{obj}\mathcal{C}$.

Such that these three ingredients satisfy the following properties,

1. The family of $\text{Mor}(\theta, \zeta)$'s is pairwise disjoint,
2. Composition is always associative,

3. For each $\theta \in \text{obj}\mathcal{C}$, there exists an identity $1_\theta \in \text{Mor}(\theta, \theta)$ satisfying

$$1_\theta \circ h = h \quad h \circ 1_\theta = h.$$

for every morphism h .

Example 1.1.1. Consider the category of all topological spaces \mathcal{C}

1. $\text{Obj}\mathcal{C}$ = topological spaces,
2. $\text{Hom}(C, D) = \{\text{all continuous functions } C \rightarrow D\}$, and
3. Composition of functions is the usual composition.

Definition 1.1.2 (Equivalence in category). An equivalence in a category \mathcal{C} is a morphism $g : A \rightarrow B$ for which \exists a morphism $h : B \rightarrow A$ as,

$$g \circ h = 1_B$$

and

$$h \circ g = 1_A.$$

Definition 1.1.3 (Functors). If \mathcal{B} and \mathcal{D} are categories, a functor $T : \mathcal{B} \rightarrow \mathcal{D}$ is a function, that is,

1. $b \in \text{Obj}\mathcal{B}$ implies $Tb \in \text{Obj}\mathcal{D}$,
2. If $f : B \rightarrow B'$ is a morphism in \mathcal{B} . Then $TF : TB \rightarrow TB'$ is a morphism in \mathcal{D} ,
3. If g, h are morphisms in \mathcal{B} for which $g \circ h$ is defined, then

$$T(g \circ h) = (Tg) \circ (Th).$$

Definition 1.1.4 (Affine Subset). Let A be a subset of Euclidean space and for every pair of distinct points $z, z' \in L$, the line resolute by z, z' is contained in L .

Example 1.1.2. An affine set in \mathbb{R}^2 is either ϕ , or a singleton, or line, or \mathbb{R}^2 .

Definition 1.1.5 (Affine combination). Affine combination is define as, let $\{u_0, u_1, \dots, u_m\} \in \mathbb{R}^n$ is a point x with

$$x = z_0 u_0 + z_1 u_1 + \dots + z_m u_m. \tag{1.1}$$

where $\sum_{i=0}^m z_i = 1$. A convex combination is an affine combination for which $z_i \geq 0$.

1.2 Simplicial Homology

Simplicial complex, homology gives the algebraic measure on the behalf of cycles which are not boundaries. The main knowledge is to triangulate the space which is under consideration, which will help us in calculating the homology of such spaces. The idea of simplicial homology [27] is coming from the algebraic topology in which the topological spaces are homeomorphic to each other if they have the same number of holes.

The simplicial homology of any simplicial complex gives us the n-dimensional holes in it. Simplicial homology β_0 gives us the connected components present in the simplicial complex. β_1 denotes the 1-dimensional holes inside the simplicial complex. β_2 denotes the 2-dimensional holes inside the simplicial. Simplicial homology studies the n-dimensional holes in complexes and these number of holes are denoted as Betti numbers.

Simplicial is a package from Python to create, manipulate, and explore simplicial complexes. It seeks to provide programmers and mathematicians with a useful collection of features while remaining scalable to tackle big complexes. Simplicial is not ideal for handling massive image data sets or extremely high-dimensional spaces that require extensive approaches to programming. In A.1 we describe this library in detail. This library has many features are as following:

- Allows complexes to be embedded into arbitrary dimensional spaces to carry out geometric and topological computations.
- Computes derived structures such as flag complexes, Vietoris-Rips complexes.
- Perform homology calculations, calculate Euler characteristics, integrate Euler characteristics and other relevant procedures.

Definition 1.2.1 (Simplex). An affine independent subset let say, $\{u_0, u_1, \dots, u_m\}$ of \mathbb{R}^n . Convex set spanning by this set, represented by $[u_0, u_1, \dots, u_m]$, is called affine m-simplex, and with vertices u_0, u_1, \dots, u_m .

Definition 1.2.2 (Braycenter). Let $\{u_0, u_1, \dots, u_m\}$ be an affine independent, then braycenter of $[u_0, u_1, \dots, u_m]$ is $\left(\frac{1}{m+1}\right)(u_0, u_1, \dots, u_m)$.

Example 1.2.1. The 2-simplex $[u_0, u_1, u_2]$ is triangle with interior and vertices are u_0, u_1, u_2 .

Definition 1.2.3 (Faces of simplex). Let $[u_0, u_1, \dots, u_m]$ be m-simplex. The faces opposed to u_i

$$[u_0, \dots, \hat{u}_i, \dots, u_m] = \left\{ \sum z_j u_j : z_j \geq 0, \sum z_j = 1, \text{ and } z_i = 0 \right\}. \quad (1.2)$$

Boundary of $[u_0, u_1, \dots, u_m]$ is union of all faces.

Definition 1.2.4 (Linear ordering of a simplex). An orientation of $\Delta^n = [e_0, e_1, \dots, e_n]$ is linear ordering of all its vertices.

Example 1.2.2. Consider a 2-simplex $[e_0, e_1, e_2]$. There are two types of orientation one is counterclockwise and the second is clockwise are as following;

Case 1:

$$e_0 < e_1 < e_2$$

This orientation gives counterclockwise trip of the triangle.

Case 2:

$$e_0 < e_2 < e_1$$

This orientation gives clockwise trip of the triangle.

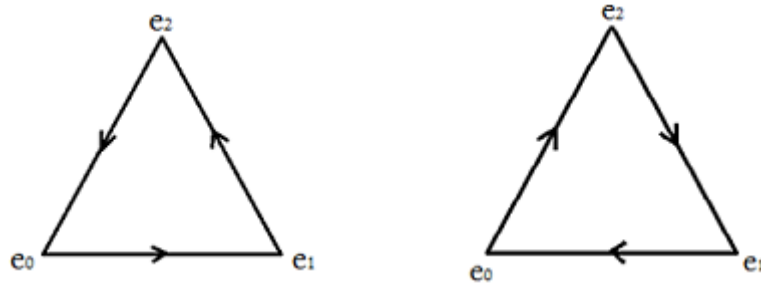


Figure 1.1: Counterclockwise trip and clockwise trip

Definition 1.2.5 (Orientation of faces). Let Δ^m be m-simplex its orientation of faces is an induced orientation of faces defining by orienting the i th in the sense

$$(-1)^i [e_0, \dots, \hat{e}_i, \dots, e_n] \quad (1.3)$$

Here, the negative sign stated as follows $[e_0, \dots, \hat{e}_i, \dots, e_n]$ has the orientation opposite to the $[e_0, \dots, \hat{e}_i, \dots, e_n]$.

Example 1.2.3. Consider a 2-simplex $\Delta^2 = [e_0, e_1, e_2]$. An orientation of this simplex is

$$e_0 < e_1 < e_2$$

Now the 0th face of Δ^2 is $(-1)^0[\hat{e}_0, e_1, e_2] = [e_1, e_2]$. Similarly the first face of Δ^2 is given by $(-1)^1[e_0, \hat{e}_1, e_2] = -[e_0, e_2] = [e_2, e_0]$ which mean the face is oriented from e_2 to e_1 .

The boundary of Δ^2 is $[e_1, e_2] \cup [e_2, e_0] \cup [e_0, e_1]$.

Definition 1.2.6 (Vertex set). If $s = [v_0, v_1, \dots, v_n]$ is a n-simplex its vertex set is define as $\text{Vert}(s) = \{v_0, v_1, \dots, v_n\}$.

Example 1.2.4. Consider a 2-simplex $\Delta^1 = [e_0, e_1]$ then vertex set is given by $\text{Vert}(s) = \{v_0, v_1\}$

Definition 1.2.7 (Simplicial Complex). Consider K is a simplicial complex in space of Euclidean and its a collection of finite simplexes such that;

1. If $x \in K$ then each face of s belongs to K .
2. If $x, y \in K$, then $x \cap y$ is either empty or a common face of x and y .

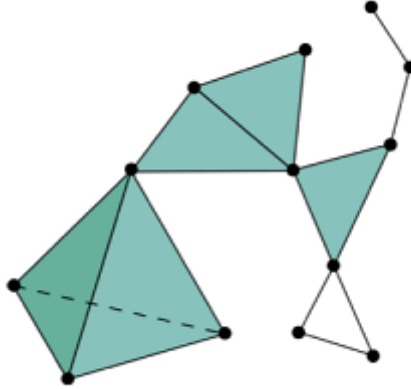


Figure 1.2: Simplicial complex

Definition 1.2.8 (Underlying space). If K is a simplicial complex, then $|K|$ is called underlying space which is subspace of ambient Euclidean space,

$$|K| = \cup_{s \in K} S$$

which is the union of all simplexes in K .

Definition 1.2.9 (Dimension). If K is a simplicial complex, then its dimension is denoted by $\dim K$ and define as,

$$\dim K = \sup \{ \dim s \mid s \in K \}.$$

Definition 1.2.10. Let $q \geq 0$ and K be an oriented simplicial complex, and consider $C_q K$ be the Abelian group with the following presentation.

Generators: all $(q + 1)$ -tuples (u_0, u_1, \dots, u_q) with $u_i \in \text{Vert}(K)$ such that $\{u_0, u_1, \dots, u_q\}$ spans a simplex in K .

Relations:

- $(u_0, u_1, \dots, u_q) = 0$ if some vertex is repeated.
- $(u_0, u_1, \dots, u_q) = \text{sgn } \pi(u_{\pi(0)}, u_{\pi(1)}, \dots, u_{\pi(q)})$, here π is a permutation of $\{0, 1, 2, \dots, q\}$.

Definition 1.2.11 (Boundary operator). Define boundary operator $\partial_q : C_q K \rightarrow C_{q-1} K$ by setting

$$\partial_q(\langle u_0, u_1, \dots, u_q \rangle) = \sum_{i=0}^q (-1)^i \langle u_0, u_1, \dots, \hat{u}_i, \dots, u_q \rangle. \quad (1.4)$$

Where \hat{u}_i means delete u_i and extending by linearity.

Example 1.2.5. Consider a 2-simplex $K_1 = [e_0, e_1, e_2]$. Now we calculate the $\partial_2(K_1)$

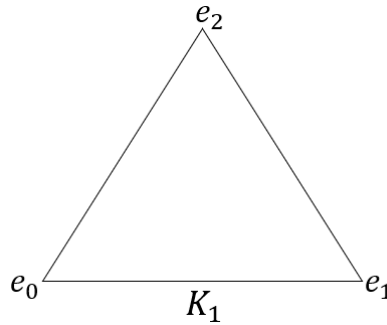


Figure 1.3: A complex

$$\partial(K_1) = (-1)^0 \langle \hat{e}_0, e_1, e_2 \rangle + (-1)^1 \langle e_0, \hat{e}_1, e_2 \rangle + (-1)^2 \langle e_0, e_1, \hat{e}_2 \rangle \quad (1.5)$$

$$\partial(K_1) = \langle e_1, e_2 \rangle - \langle e_0, e_2 \rangle + \langle e_0, e_1 \rangle \quad (1.6)$$

Lemma 1. $\partial_q : C_q(K) \longrightarrow C_{q-1}(K)$ and $\partial_{q-1} : C_{q-1}(K) \longrightarrow C_{q-2}(K)$. Then the composition of two consecutive boundary operator is $\partial_q \circ \partial_{q-1} = 0$.

Definition 1.2.12. If K is an oriented simplicial complex, then

- $Z_q K = \text{Ker} \partial_q$ represents the **qth-simplicial cycles**.
- $B_q K = \text{Im} \partial_{q+1}$ represents the **qth-simplicial boundaries**.
- $H_q K = Z_q K / B_q K$ represents the **Homology group**.

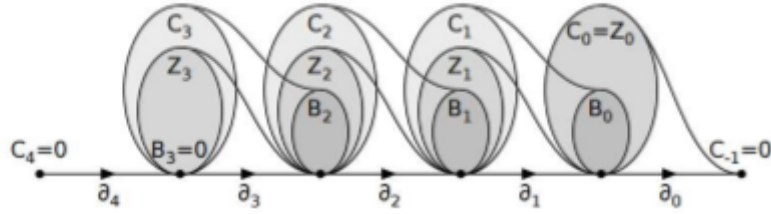


Figure 1.4: Simplicial homology

Theorem 1.2.1. A group G that is Abelian (finitely generated) has the same isomorphism as the direct product of cyclic groups in the form;

$$\mathbb{Z}_{(u_1)^{s_1}} \times \mathbb{Z}_{(u_2)^{s_2}} \times \cdots \times \mathbb{Z}_{(u_n)^{s_n}} \times \mathbb{Z} \times \mathbb{Z} \times \cdots \mathbb{Z},$$

where the u_i are primes, not necessarily distinct, and also in the form

$$\mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \times \cdots \times \mathbb{Z}_{n_s} \times \mathbb{Z} \times \mathbb{Z} \times \cdots \mathbb{Z},$$

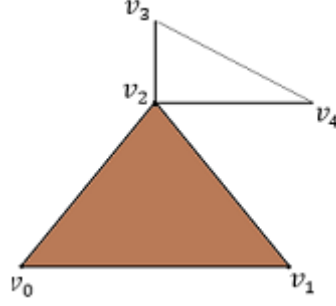
where n_i divides n_{i+1} .

It gives information about groups in form of Betti numbers and homology is being determine of different topological spaces.

Definition 1.2.13. The q th homology group with ring of coefficients R is

$$H_q(M; R) = Z(M; R) / B(M; R).$$

Example 1.2.6. Consider a simplicial complex,



here we can see that $\dim K = 2$ and

$$C_q(K) = 0 \quad q > 2.$$

So we have the chain complex,

$$0 \xrightarrow{\partial_3} C_2(K) \xrightarrow{\partial_2} C_1(K) \xrightarrow{\partial_1} C_0(K) \xrightarrow{\partial_0} 0.$$

All the 2 simplices are the generators of $C_2(K)$. Let it be

$$\rho = \langle v_0, v_1, v_2 \rangle,$$

The classification theorem(1.2.1) here plays a role and we have the following isomorphism of finitely generated abelian group $C_2(K)$.

$$C_2(K) = \{n\rho : n \in \mathbb{Z}\} \cong \mathbb{Z}.$$

Similarly 1-simplices are the generators of $C_1(K)$, We denote them by

$$\sigma_1 = \langle v_0, v_1 \rangle, \sigma_2 = \langle v_0, v_2 \rangle, \sigma_3 = \langle v_1, v_2 \rangle, \sigma_4 = \langle v_2, v_3 \rangle, \sigma_5 = \langle v_2, v_4 \rangle, \sigma_6 = \langle v_3, v_4 \rangle$$

hence we have

$$C_1(K) = \{m_1\sigma_1 + \dots + m_6\sigma_6 : m_i \in \mathbb{Z}\} \cong \mathbb{Z}^6.$$

Similarly $C_0(K)$ has all the 0 simplices as generators of the simplicial complex K . Hence we have

$$C_0(K) = \{m_1v_0 + \dots + m_5v_5 : m_i \in \mathbb{Z}\} \cong \mathbb{Z}^5.$$

Now to calculate the homology we will calculate all the $Z_q(K)$ and $B_q(k)$.

B₂(K), Since $C_3(K) = 0 \Rightarrow B_2(K) = 0$.

Z₂(K), We will first take the image under ∂_2 of the generator of $C_2(K)$.

$$\partial_2(\rho) = \sigma_3 - \sigma_2 + \sigma_1.$$

Let $c_2 \in C_2(K)$ so

$$\partial_2(c_2) = \partial_2(n\rho) = n\partial_2(\rho) = n(\sigma_3 - \sigma_2 + \sigma_1),$$

so we have $\partial_2(c_2) = 0$ iff $n = 0$ hence $Z_2(K) = 0$ so

$$H_2(K) = 0.$$

Z₁(K), we will calculate $\ker \partial_1$.

Let $c_1 \in C_1(K)$ be a general element

$$\partial_1(c_1) = \partial_1(m_1\sigma_1 + \dots + m_6\sigma_6),$$

by using linearity of boundary operator, solving and rearranging we get

$$\partial_1(c_1) = (-m_1 - m_2)\langle v_0 \rangle + (m_1 - m_3)\langle v_1 \rangle + (m_2 + m_3 - m_4 - m_5)\langle v_2 \rangle + (m_4 - m_6)\langle v_3 \rangle + (m_5 + m_6)\langle v_4 \rangle.$$

Let the coefficients be zero and using them in terms of a single coefficient. So the general element of $Z_1(K)$ has the form

$$c_1 = \sum_{i=1}^6 m_i \sigma_i = m_1(\sigma_3 - \sigma_2 + \sigma_1) + m_4(\sigma_4 - \sigma_5 + \sigma_6),$$

let $z_1 = \sigma_3 - \sigma_2 + \sigma_1, z_2 = \sigma_4 - \sigma_5 + \sigma_6$ so we have

$$Z_1(K) = \{m_1 z_1 + m_4 z_2\} \cong \mathbb{Z}^2.$$

B₁(K), We know that $B_1(K) \subset Z_1(K)$ because composition of boundary operators are always zero.

We also see that

$$z = \sigma_3 - \sigma_2 + \sigma_1 = \partial_2(\rho).$$

Hence

$$H_1(K) = Z_1(K)/B_1(K) = \mathbb{Z}^2/\mathbb{Z} = \mathbb{Z}.$$

B₀(K), We claim that : an element $\sum_{i=0}^4 r_i \langle v_i \rangle$ belongs to $B_0(K)$ if and only if $\sum_{i=0}^5 r_i = 0$ which can be shown easily,

Hence

$$\sum_{i=0}^4 r_i \langle v_i \rangle \in B_0(K).$$

We also know that $Z_0(K) = C_0(K)$. Now consider the following homomorphism

$$f : C_0(K) \rightarrow \mathbb{Z}$$

defined by

$$f\left(\sum_{i=0}^4 r_i \langle v_i \rangle\right) = \sum_{i=0}^4 r_i.$$

Here $\ker f = B_0(K)$, so by using first isomorphism theorem we have

$$H_0(K) = Z_0(K)/B_0(K),$$

$$H_0(K) = C_0(K)/B_0(K),$$

$$H_0(K) \cong \mathbb{Z}.$$

1.2.1 Homological Algebra

Homological functors are studied in homological algebra which is a branch of Mathematics. The emerging category theory was closely related to the development of homological algebra. One ubiquitous and quite useful concept in mathematics is chain complexes.

Proposition 1. Consider a simplicial complex K and \exists a vertex x of K with these features.

If vertices v_0, v_1, \dots, v_q span a simplex of K then so do x, v_0, v_1, \dots, v_q

then $H_0(K) \cong \mathbb{Z}$, $H_q(K)$ is zero group $\forall q > 0$.

Definition 1.2.14 (Exact sequence). Consider R be a ring with identity, and I, J, K be R -modules. Let

$$f : I \rightarrow J, g : J \rightarrow K$$

be R -module homomorphisms. Then the sequence

$$\dots I \xrightarrow{f} J \xrightarrow{g} K \dots$$

is exact at J iff,

$$\text{Im } (f : I \rightarrow J) = \text{Ker } (g : J \rightarrow K)$$

If a sequence of modules and homomorphisms is exact at each module in the sequence, then it is said to be exact.

Definition 1.2.15. An exact sequence is of the form of

$$0 \rightarrow F \rightarrow G \rightarrow H \rightarrow 0$$

called the short exact sequence.

Example 1.2.7. Assume P and Q are two modules on a ring R . Consider the following module sequence:

$$0 \rightarrow P \xrightarrow{f} P \oplus Q \xrightarrow{g} Q \rightarrow 0$$

where

$$f(p) = (p, 0)$$

and

$$g(p, q) = q.$$

The sequence above is exact at each node and is hence an exact short sequence, by the first isomorphism theorem of modules we have:

$$(P \oplus Q)/P \cong Q.$$

Lemma 2 (Five lemma). Let R be a unity ring, then R -module homomorphism and R -module with the following commutative diagram are exact sequences. Where A, B, C, D, E, R are R -modules and $\alpha, \beta, \gamma, \delta$ are R -module homomorphism. If f, h are isomorphism then g is also an isomorphism.

$$\begin{array}{ccccccccc} 0 & \longrightarrow & A & \xrightarrow{\alpha} & B & \xrightarrow{\beta} & C & \longrightarrow & 0 \\ & & \downarrow f & & \downarrow g & & \downarrow h & & \\ 0 & \longrightarrow & D & \xrightarrow{\gamma} & E & \xrightarrow{\delta} & R & \longrightarrow & 0 \end{array}$$

Figure 1.5: The Five lemma

The five lemma are usually used to calculate the homology or cohomology of a particular object for long, exact sequences.

Definition 1.2.16 (Exact sequences). A short exact sequence of chain complexes (A_*, ∂_i) , (B_*, ∂_i) , (C_*, ∂_i) , and chain maps $p_n : A_* \rightarrow B_*$ and $q_n : B_* \rightarrow C_*$ such that the sequence,

$$0 \rightarrow A_n \xrightarrow{p_i} B_n \xrightarrow{q_i} C_n \rightarrow 0$$

is exact for each $n \in \mathbb{Z}$. The sequence is an exact short sequence iff commutative property holds in the following diagrams.

$$\begin{array}{ccccccc}
& \vdots & & \vdots & & \vdots & \\
& \downarrow \partial_{i+2} & & \downarrow \partial_{i+2} & & \downarrow \partial_{i+2} & \\
0 & \longrightarrow & A_{i+1} & \xrightarrow{p_{i+1}} & B_{i+1} & \xrightarrow{q_{i+1}} & C_{i+1} \longrightarrow 0 \\
& & \downarrow \partial_{i+1} & & \downarrow \partial_{i+1} & & \downarrow \partial_{i+1} \\
0 & \longrightarrow & A_i & \xrightarrow{p_i} & B_i & \xrightarrow{q_i} & C_i \longrightarrow 0 \\
& & \downarrow \partial_i & & \downarrow \partial_i & & \downarrow \partial_i \\
0 & \longrightarrow & A_{i-1} & \xrightarrow{p_{i-1}} & B_{i-1} & \xrightarrow{q_{i-1}} & C_{i-1} \longrightarrow 0 \\
& & \downarrow \partial_{i-1} & & \downarrow \partial_{i-1} & & \downarrow \partial_{i-1} \\
& & \vdots & & \vdots & & \vdots
\end{array}$$

Figure 1.6: Exact sequence of chain complexes

Source: [34]

Note that the columns of this diagram are chain complexes and rows are exact sequences.

1.3 Inner Framework of Simplicial Complex

Simplicial homology performs an essential role in Persistent homology which is a subfield of TDA. In simplicial homology, the internal structure of a simplicial complex consists of three ingredients:

1. Star
2. Closure
3. Link

Definition 1.3.1 (Star). If K be a simplicial complex and P be a collection of simplices in K , then a star is the union of each complex in P , and it is denoted as $St(P)$

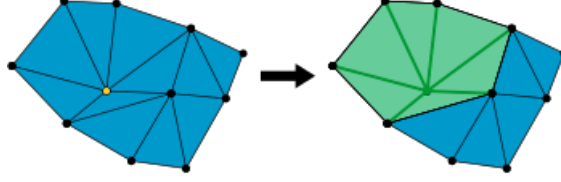


Figure 1.7: Star of a simplex

Source: [24]

It depends upon the single vertex of a simplicial complex and also varies from vertex to vertex. A star is a local neighborhood of a single vertex, and generally, it is not simplicial complex. To make it simplicial complex we take the closure of a star.

Definition 1.3.2 (Clouser). If K be a simplicial complex and P be a collection of simplices in K , then closure is the smallest simplicial complex of P that contains each complex in S . It is denoted by $Cl(St(P))$.

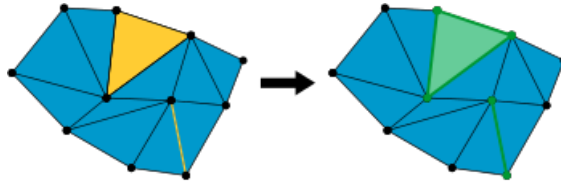


Figure 1.8: Closure of a simplex

Source: [24]

Definition 1.3.3 (Link). If K be a simplicial complex and P be a collection of simplices in K , then the link of a simplicial complex is the closed star of P minus the stars of all faces of P . It is denoted by the $Lk(P) = Cl(St(P)) - St(P)$.

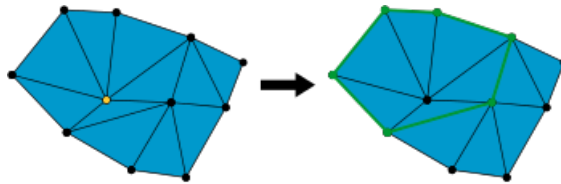


Figure 1.9: Link of a simplex

Source: [24]

1.4 Building Simplicial Simplexes

A simplicial complex structure is formed by the combination of points, intervals, triangles, and higher dimensional polyhedrons. The simplicial complexes cover topological space in a very precise way. Furthermore, we will discuss the structure of simplicial complexes.

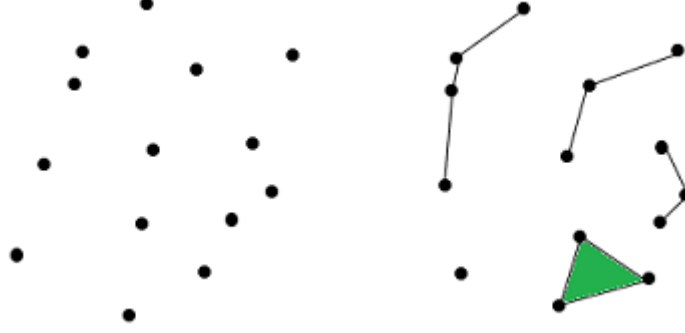


Figure 1.10: Point cloud and Simplicial complex

Definition 1.4.1 (Nerve of a space). If X is a topological space with covering $\mathcal{P} = \{U_\alpha\}_{\alpha \in I}$, then nerve of \mathcal{P} will be the simplicial complex with the vertex set I , where a k -simplex is spanning by $\{\alpha_1, \alpha_2, \dots, \alpha_k\}$ if and only if $\{U_{\alpha_0} \cap \dots \cap U_{\alpha_k}\} \neq \emptyset$.

Examining the data's shape is an essential task. In the nerve theorem, we will see that nerve is the best estimation for the topological spaces, depending on the optimal covering. The illustration depicts the nerve of space covering.

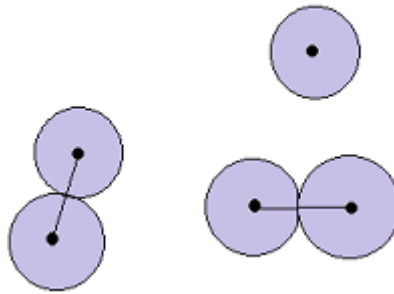


Figure 1.11: Nerve of a covering

Theorem 1.4.1 (Nerve theorem). Let X be a topological space and \mathcal{P} be the covering and consider that covering consist of open sets; then the nerve of \mathcal{P} is homotopy equivalent to

$$\bigcap_{A \in \mathcal{P}}$$

is either contractible or empty. This is a well-known covering approach, which is provided by the family of ϵ closed balls $\mathfrak{B}_\epsilon(X) = \{B_\epsilon(x)\}_{x \in X}$, for some $\epsilon > 0$. More generally, $Y \subset X$ for which $X = \bigcup_{y \in Y} B_\epsilon(y)$ for any subset.

Definition 1.4.2 (\hat{C} ech complex). The construction of nerve from the covering $\{B_\epsilon(y)\}_{y \in Y}$ is denoted by $\hat{C}(Y, \epsilon)$, known as the \hat{C} ech complex devoted to Y and ϵ .

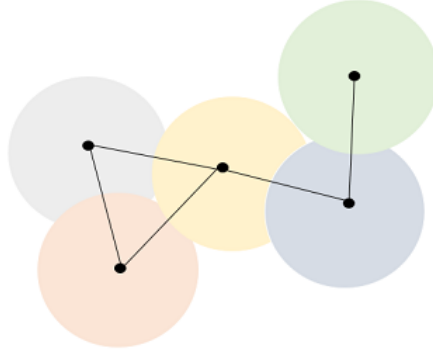


Figure 1.12: \hat{C} ech complex

\hat{C} ech complex is roughly constructed from the nerve and needs to store en-
large the amount of intersection. This construction of simplex is computationally
expensive and to overcome this problem we construct another structure of the
complex.

Definition 1.4.3 (Vietoris-Rips complex). If (X, d) be a metric space. Then
the vietoris-Rips complex of the space X with some parameter ϵ is represented
by $VR(X, \epsilon)$ will be the simplicial complex whose vertex set is X , and where
 $\{x_0, x_1 \dots x_k\}$ span a k -simplex iff $d(x_u, x_v) \leq \epsilon \forall 0 \leq u, v \leq k$.

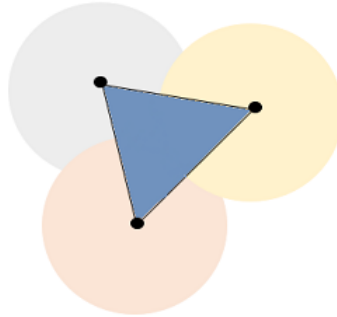


Figure 1.13: Vietoris-Rips complex

Vietoris-Rips complex depends upon the radii(ϵ) the covering ball of a space. The distance between the centre of the balls is technically VR -complex. VR -complex cover the one-dimensional holes very effectively. For a subset of Euclidean space Y , it approximates $\hat{C}ech$ complex in terms of parameters $\epsilon > 0$, such that

$$\hat{C}(Y, \epsilon) \subset VR(Y, \epsilon).$$

VR - complex is computationally less expensive than $\hat{C}ech$ complex and it is often used in studying the topology of the whole space.

Definition 1.4.4 (Voronoi Decomposition). Let Y be any metric space, and let $\mathcal{L} \subset Y$ called the set of landmark points, also let $\alpha \in \mathcal{L}$ we define the Voronoi cell related to α , V_α by

$$V_\alpha = \{y \in Y | d(y, \alpha) \leq d(y, \alpha')\} \quad (1.7)$$

The collection of V_α sets is known as Voronoi Decomposition. The Voronoi decomposition is splitting the metric space Y into sections, actually forms a covering of Y with respect \mathcal{L} . Now we can define another construction known as Delaunay Complex. Voronoi Decomposition is also called the Voronoi diagram or Voronoi partition.

Definition 1.4.5 (Delaunay Complex). Delaunay triangulation is isomorphic to the nerve of a Voronoi diagram for a finite collection of points P in \mathbb{R}^d but the nerve is not necessarily embedded into \mathbb{R}^d .

Delaunay complex gives us a small symmetrical control of the space in the case of finite metric spaces. It mostly covers the information of points and line segments, because of the existence of points having equal distance with any pair of landmark points. Meanwhile Delaunay complex tackles, higher dimensional spaces quite easily. The number of simplexes and vertex sets is restricted to choose landmark points. Its complexity depends on the dimension of the space, making it a computationally expensive method for the high dimensional spaces.

Definition 1.4.6 (α -complex). Let (Y, d) be a metric space, with the set of landmark points \mathcal{L} . Consider a Voronoi diagram $\{V_{l_1}, V_{l_2}, \dots, V_{l_n}\}$, and for any $\epsilon > 0$

there exists a collection of closed balls $\mathfrak{B}_\epsilon(\mathcal{L}) = \{B_\epsilon(l)\}_{l \in \mathcal{L}}$, then the nerve of a covering

$$\bigcup_{l \in \mathcal{L}} V_l \cap B_\epsilon(l)$$

is known as α -complex.

α -complex is very similar to Čech complex with smaller in size and a subcomplex of Delaunay triangulation. Hence α -complex satisfies the following inclusion,

$$\alpha - \text{complex} \subseteq \check{\text{Cech complex}} \subseteq \text{Delaunay triangulation}.$$

Definition 1.4.7 (Clique complex). Let (X, d) be a metric space and consider a graph $G(X, E')$, where X is the vertex set and E' is edge set of the graph G . For a parameter $\epsilon > 0$, we define $E' = \{[x, y] | x, y \in X \wedge d(x, y) \geq \epsilon\}$. Consider a subgraph $S \subset G$ such that for any $x \neq y$ we have,

$$x, y \in S \implies x, y \in E'$$

known as clique. A clique S with vertices spans a k -simplex, and a collection of all the cliques in G are known as clique complex.

Definition 1.4.8 (Strong witness complex). Let (X, d) be a metric space, with a finite set of \mathcal{L} landmarks points and threshold $\epsilon > 0$. Consider for any $x \in X$, we let s_x denoting the minimum distance from the x to the set \mathcal{L} . Then the strong witness complex devoted to this data in the complex $W^s(X, \mathcal{L}, \epsilon)$ whose vertex set is \mathcal{L} , and where a set $\{l_0, l_1 \dots l_k\}$ span a k -simplex iff there is a point $x \in X$ (the witness) so that $d(x, l_i) \leq s_x + \epsilon$ for all i .

Definition 1.4.9 (weak witness complex). Let (X, d) be a metric space, with a finite set of \mathcal{L} landmarks points together with the parameter $\epsilon \geq 0$. Consider a subset $\mathfrak{L} = \{l_0, l_1 \dots l_k\}$ of \mathcal{L} such that it spans a k -simplex if there exists $x \in X$ a weak witness, such that $d(x, l) + \epsilon \geq d(x, l_i) \forall i$ and all $l \notin \mathfrak{L}$, admitted by each $l_i \in \mathfrak{L}$.

Definition 1.4.10 (Lazy witness complex). Let (X, d) be a metric space, with set of \mathcal{L} landmarks points together with a parameter $m \in \mathbb{N}$. If $m = 0$ then $s_x = 0 \forall x \in X$. If $m \geq 0$, then we consider s_x to be the minimum distance from x

to m -th closest landmark point, then the set $\{l_0, l_1 \dots l_k\}$ spans a k -simplex if \exists a witness $x \in X$ such that $d(l_i, x) \leq \epsilon + s_x$. Then $LW_m(X, \mathcal{L}, \epsilon)$ denotes the lazy witness complex with a vertex set \mathcal{L} .

Definition 1.4.11 (CW complex). A CW complex is a topological space built out of smaller spaces, iteratively by a process called attaching cells. A \mathcal{K} -cell is a \mathcal{K} -dimensional disc.

$$D^{\mathcal{K}} = \{x \in \mathbb{R}^{\mathcal{K}} : |x| \leq 1\}.$$

When a \mathcal{K} cell attaches to another space X means, forming the union of X and $D^{\mathcal{K}}$. The dimension of CW complex X is the supremum of n such that X has n cells.

Definition 1.4.12 (Cubical Complex). A cubical complex K in \mathbb{R}^n is a set of p -cubes where $0 \leq p \leq n$ such that each face of a cube in K and the intersection of any two cubes of K is either empty or common face. The dimension m of K is the greatest integer p for which K includes a p -elementary cube (in which case K is referred to as a m simplex).

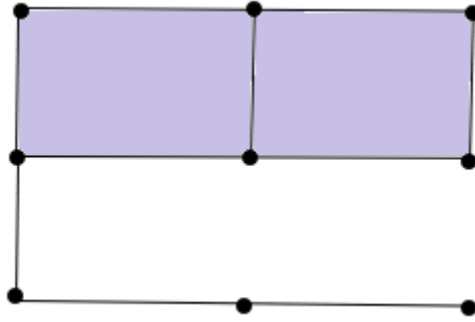
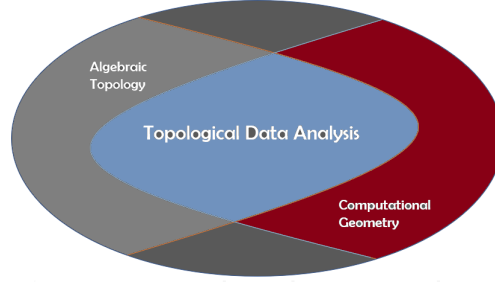


Figure 1.14: Cubical Complex

1.5 Techniques in Topological Data Analysis

Topological data analysis (TDA) is an intersection of algebraic topology and computational geometry. TDA can quantify the shape and structures in data by computing topological points that summarize key factors as connected components, loops, and voids. There are numerous applications of TDA in real-world problems, among which image compression, cancer research, and shape or pattern

recognition. To build tools to examine qualitative data aspects, TDA shall use ideas as well as geometry and topology results.



To reach this purpose, you must accurately define qualitative key features, use technologies to calculate these characteristics in practice, and as well as guarantee the robustness of those features. Persistent Homology (PH) is one approach in TDA of addressing all three problems.

1.5.1 Persistent Homology

Persistent homology [10] is a flagship tool of TDA. Persistent homology is a homology theory that is used to study the qualitative characteristic of the dataset. It computes a ‘Persistent diagram’ and there will be no change in topological features. The principle aim of PH is to examine the structure and contingency of abstract mathematical objects, such as sets and spaces. In homology, we are interested in the holes of geometrical objects, where the homology groups offer a mathematical language to describes these holes.

Betti numbers (β_n) represents the ranks of these homology groups. Betti numbers are used in algebraic topology to discriminate the topological spaces founded on the connectivity of n-dimensional simplicial complex. The few definitions are as following:

- β_0 represent the number of connected components.
- β_1 represent the one-dimensional holes.
- β_2 represent the two-dimensional voids.

Similarly the n-th betti number represent the rank of n-th homology group denoted by H_n .

Example 1.5.1. As in the following we can see both are path-connected spaces and also qualitatively distinct in that there are only one loop in the letter P to the left and two loops in the letter B to the right.



The homology group of $H_n(K)$ denotes the n -th homology group of a k -simplicial complex, such that when a point cloud data is discretized into simplicial complexes then $H_0(K)$ represents the collection of vertices of a simplicial complex, $H_1(K)$ represents the edges of a simplicial complex, and $H_2(K)$ represents the faces of a simplicial complex. There are many methods to associate a simplicial complex with data points, for example, Čech complex, Clique complex, Vietoris-Rips complex, strong witness and weak witness complex, etc. These simplicial complexes having their advantages, disadvantages, and also they have their computational limits, and the choice of the simplicial complex depends upon the real-life problem. In various constructions for simplicial complexes, the threshold value is an important tool for extracting the homological data from the space.

The philosophy of varying threshold values combined with filtration of homology groups is persistent homology.

To get motivation to consider a metric space X with open covering, whose points are randomly spread illustrated in figure 1.15.



Figure 1.15: Points in space

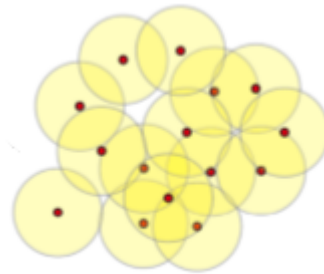


Figure 1.16: Covering of space with ϵ balls

Then varying the threshold (radius of open balls) we reached a point in figure 1.16 where the open balls cover the noise in the space. Types of data sets to study using PH include finite metric spaces, real-valued functional level sets, networks, and digital images. Formally, finite metric spaces are point cloud data set in TDA. In topological terms, there is no relevant information in finite metric spaces. Therefore, the thickening of a point cloud on many resolution levels is considered and the development of the resulting shape throughout each resolution scale is analyzed. The qualitative properties are determined by topological invariants and can be composed to summarize the shape of the data by representing a change of these invariants in the various resolution scales.

Persistent homology is the main workhorse of TDA based on algebraic topology and it computes a data structure of stable topological features to summarize the space. The Vietoris-Rips filtration (VR) is the most popular strategy used to generate persistent diagrams. VR is prohibitively slower, however lately a library called Ripser [32] was developed to include all known VR filtration with simple implementation and computationally faster than others packages. There are also many other algorithmic techniques such as Javaplex, GUDHI, Dionysus, and PHAT but [32] much more efficient. We can also see the comparison of these packages in [25].

Definition 1.5.1. Consider \mathcal{C} be any category, \mathcal{P} a partially ordered set. \mathcal{P} is a category with objects of \mathcal{P} , and with distinct morphisms from a to b whenever $a \leq b$. Then \mathcal{P} -persistence object in \mathcal{C} refers to a functor $\phi : \mathcal{P} \rightarrow \mathcal{C}$.

Note that the \mathcal{P} -persistence objects in \mathcal{C} form a category, with morphisms of natural transformation from a family $\{c_a, \phi_{ab}\}$ to a family $\{c_b, \psi_{ab}\}$ in \mathcal{C} . Although, if h is a partial order preserving map between two partially ordered sets \mathcal{P}, \mathcal{Q} also regarded as a functor. If φ is a \mathcal{Q} -persistence object in \mathcal{C} then the composition of functor $g* = \varphi \circ h$ is known as an evident functor.

Definition 1.5.2 (Persistent Module). A persistent module M is a family of R -modules M_i together with homomorphisms $\psi_i : M_i \rightarrow M_{i+1}$.

Definition 1.5.3 (Chain complex). A C_* chain complex is a doubly infinite se-

quence of modules $\{C_i : i \in \mathbb{Z}\}$ over some ring of unity, with homomorphisms $\partial_i : C_i \rightarrow C_{i-1}$ for each $i \in \mathbb{Z}$, such that $\partial_i \circ \partial_{i+1} = 0 \forall i$.

We have a chain of module and homomorphisms diagrammatically,

$$\dots \xrightarrow{\partial_{q+1}} C_q \xrightarrow{\partial_q} C_{q-1} \xrightarrow{\partial_{q-1}} \dots$$

such that any two successive maps are zero. Observe that the condition $\partial_i \circ \partial_{i+1} = 0$ implies that

$$\text{Im} \partial_{i+1} \subset \ker \partial_i.$$

Definition 1.5.4 (Chain map). Let C, D be two chain complexes; then a chain map $f : C \rightarrow D$ is a collection of morphisms $\{f_n : C_n \rightarrow D_n\} \forall n \in \mathbb{Z}$ such that all the diagrams are commutative,

$$\begin{array}{ccc} C_{n+1} & \xrightarrow{g_n} & C_n \\ f_{n+1} \downarrow & & \downarrow f_n \\ D_{n+1} & \xrightarrow{g'_n} & D_n \end{array}$$

Figure 1.17: Chain map

hence such that all the equations,

$$f_n \circ g_n = g'_n \circ f_{n+1}.$$

Proposition 2 (Chain homology). For $f : C \rightarrow D$ a chain map; so that $\forall n \in \mathbb{Z}$ it respects boundaries and cycles it restrict to a morphism,

$$B_n(f) : B_n(C) \rightarrow B_n(D)$$

and

$$Z_n(f) : Z_n(C) \rightarrow Z_n(D)$$

In particular it also respects chain homology;

$$H_n(f) : H_n(C) \rightarrow H_n(D).$$

Theorem 1.5.1 (Correspondence). The χ correspondence defines category equivalence between the category of finite type persistence modules over \mathbb{R} and the category of finite non-negatively graded modules over $\mathbb{R}[t]$.

The classification of these $F[t]$ -modules isomorphic classes are parametrized through the intervals, where the first sum corresponds to the semi infinite interval and the second sum equaling to finite intervals. Let these intervals be defined as $V(x, y)$ where V is,

$$V(x, y)_t = \begin{cases} 0 & t \leq x \\ F & x \leq t \leq y \end{cases}$$

where $x_i \in \mathbb{N}$ and $y_i \in \mathbb{N} \cup \infty$.

Further defining a filtration that every \mathbb{N} -persistence F vector spaces M_y are consecutively isomorphic and finite-dimensional, known as tame. This tame extract these intervals by the decomposition below

$$\{M_y\}_y \cong \bigoplus_{i=0}^N V(x_i, y_i).$$

The period of surviving of these intervals shows the lifetime of a homology group or the homological info by varying ϵ of any simplicial complex construction, enabling us to recover the possible features by overcoming the noise of the data. The intervals are known as barcodes of PH.

The period of surviving of these intervals shows the life span (birth & death) of the homology group or the homological info by varying ϵ of any simplicial complex construction, enabling us to recover the possible features by overcoming the noise of the data.

The discussion is summarized as follows:

- Find an order preserving map $h : \mathbb{N} \rightarrow \mathbb{R}$.
- Construct \mathbb{N} -persistence chain complexes with coefficients in F .
- Compute the barcodes of \mathbb{N} -persistence F vector spaces.

In definition (1.2.13), if we choose a field F as our ring of coefficients, the homology groups become torsion-free vector spaces such that $H_q(K_\epsilon, F) \cong F^r$ where r is the rank of vector space. Hence, it clears the last step of our summary in which \mathbb{N} -persistence F vector spaces $(H_q(C_n(K_\epsilon), F))$ are being computed.

These barcodes or intervals of life span contain the topological key features like connected components or holes in our dataset. The small intervals denote noise and the one which persists for a long gives us the feature information. PH gives us a strategy to view data in terms of barcodes and extract features.

Example 1.5.2. In figure 1.18, a small point cloud is taken and the bars below show us the life period of a homology group for the threshold value. The long bar of H_1 group represents the big circle in the fourth step, giving us the feature.

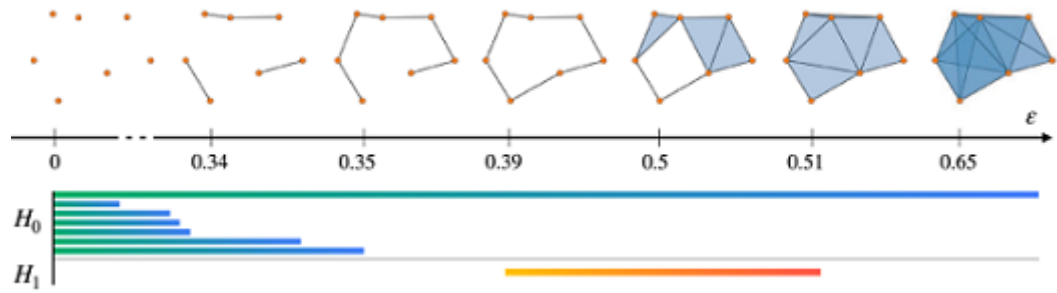


Figure 1.18: Filtrations and barcodes for ϵ increasing horizontally

Source: [16]

1.5.2 Persistent Diagram

A persistent diagram (PD) is a multiset that is a union of a finite multiset of points in \mathbb{R}^2 with the multiset of points on the diagonal $\Delta = \{(m, n) \in \mathbb{R}^2 | m = n\}$, where an infinite multiplicity of each point on that diagonal.

- Example 1.5.3.**
1. 0-dimensional topological feature is a cluster or connected components.
 2. 1-dimensional topological feature is a hole.
 3. 2-dimensional topological feature is a void up to so on the higher dimensional analogous.

Example 1.5.4. The well known libraries for datasets are teaspoon and scikit-tadatasets also for generating the datasets and algorithmically calculating the PDs we used [32]. Other dependencies are Cython, Numpy, Pandas, Scikit-learn, and Matplotlib.

In figure 1.19 a very well known library of Python, **teaspoon** is used for generating the datasets in which number of points are ($N = 300$), noise = 0.05, and seed = 0. In figure 1.20 there are the Persistent diagrams for the above point cloud datasets.

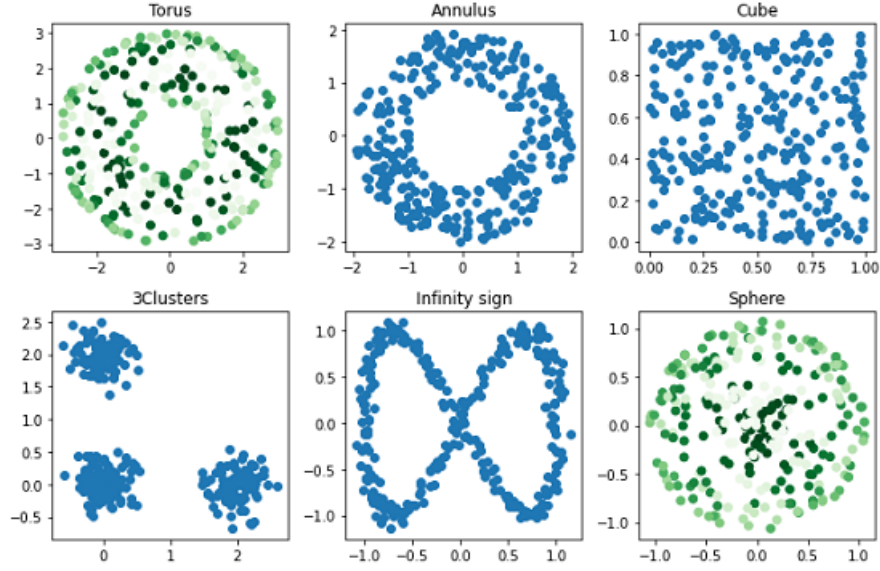


Figure 1.19: Point cloud datasets

PDs are the a finite multisets of points in space of \mathbb{R}^2 . Here, below H_0 , H_1 represents the number of connected components, and holes in datasets.

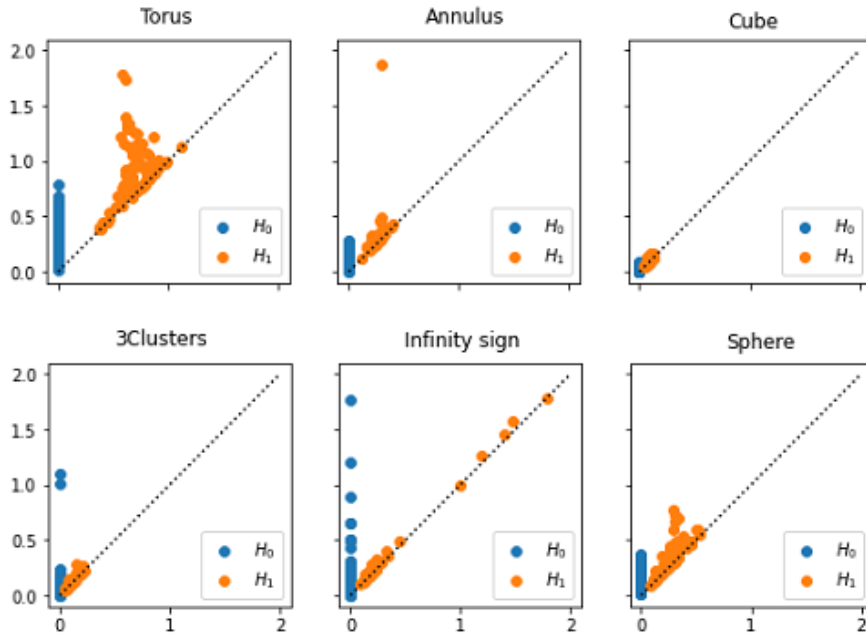


Figure 1.20: Persistent diagrams for 1.19

The level at which a generator for the homology groups is created is called its birth time and the level at which a generator merges with another homology group is called its death time. Algorithmically there are different packages for the creation of persistent diagrams such as Ripser and Dionysus.

A PD from a data set can be constructed in different ways. The construction of the corresponding PDs is straightforward when the data is images or functions. If the data are a collection of points, the construction of a PD enables the construction of the function producing the underlying filtration. Below you can see an example of a PD in which points near the diagonal represent short-lived functions and those points that are far away from the diagonal represent long-lasting features. In 1.21 the life span of features is given:

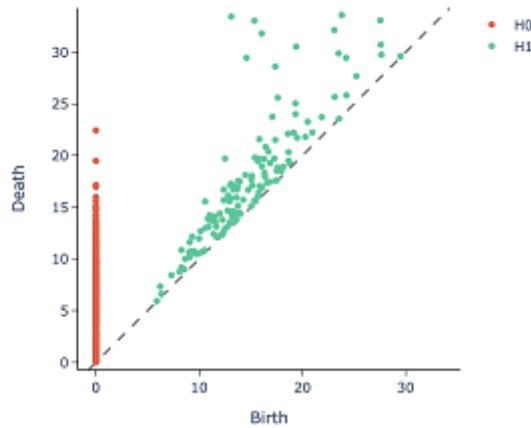


Figure 1.21: Life span of features

Source:[5]

Definition 1.5.5 (Filtered Simplicial Complex). Let K be a finite simplicial complex, let $\phi = K^0 \subset K^1 \subset K^2 \subset \dots \subset K^n = K$ be a finite sequence of subcomplexes. For generality, we let $K^i = K^n$ for all $i \geq n$. A filtered simplicial complex is referred to as a simplicial complex K with such a sequence of subcomplexes.

In other words, if $a_1 < a_2 < \dots < a_n$ are the function value of the simplexes in K and $a_0 = -\infty$, we call these sequences of complexes the filtration of f and construction by adding the chunks of simplices at time. We want to sum up how topology of filtration changes over time. We have an inclusion map just like in

[18] of simplicial complexes $i' : K_a \rightarrow K_b$ for every $i < j$. This leads to maps for inclusion

$$i' : B_k(K_a) \rightarrow B_k(K_b) \text{ and } i' : Z_k(K_a) \rightarrow Z_k(K_b).$$

This induce homomorphisms (which are generally not inclusions)

$$i'_k{}^{a \rightarrow b} : H_k(K_a) \rightarrow H_k(K_b).$$

for each dimension k . Thus, filtration corresponds to a sequence of homology groups linked by homomorphisms,

$$0 = H_k(K_0) \rightarrow H_k(K_1) \rightarrow \cdots \rightarrow H_k(K_n) = H_k(K).$$

again one for each dimension k .

Definition 1.5.6 (Persistent Homology Groups). The K -th persistent homology groups are the images of the homomorphisms induced by inclusion, $H_k^{i,j} = \text{Im } f_k^{i,j}$ for $0 \leq i \leq j \leq n$. The corresponding k -th persistent Betti numbers are the rank of these groups, $\beta_k^{i,j} = \text{rank } H_k^{i,j}$.

Similarly, we can define reduced persistent Betti numbers and reduced persistent homology groups. The PH groups are the K_i homology classes that are still alive at K_j or, more suitably,

$$H_k^{i,j} = Z_k^{K_i} / (B_k(K_j) \cap Z_k(K_i)).$$

here $H_k^{i,j}$ is the group of homology classes in $H_k(K_i)$ which persist to $H_k(K_j)$ or in other words image of $i'_k{}^{a \rightarrow b}$. The persistent homology transform gives a method of performing shape statistics. Given an object $N \subset \mathbb{R}^d$ and $t \in S^{d-1}$, let $X_k(N, t)$ be the K th dimensional persistent diagram corresponding to the height function t ,

$$f(x) = x \cdot t \quad \forall \quad x \in N.$$

Two objects N and N' are close if for every t , the corresponding diagrams $X(N, t)$ and $X(N', t)$ are close.

Chapter 2

Metrics on the Space of Persistent Diagrams

Persistent diagrams are useful for capturing the topological features of point cloud data. Although the metrics used in TDA are similar to those used in optimum transportation, the formalism based on optimal transport for studying PDs and similar topological key descriptors still needs to be established. The persistence diagrams are a space for discrete measures, such measures naturally occur in TDA when continuous representations of persistence diagrams are considering. In this chapter, we are mainly dealing with the space of persistent diagrams. Different metrics are discussed later on such as bottleneck, Hausdorff, and Wasserstein distances and other techniques are persistent landscape, persistence images, Riemannian framework, persistent homology transform and smooth Euler characteristics transform.

2.1 Distances

Homological functor allows us to see the difference in different point cloud datasets. Betti number is one thing that makes difference in two different point cloud data. While things are not simple in the case of PH. We studied the behavior of homology in terms of bar diagrams or barcodes, and these should be carefully treated to find the difference between the topological features of two-point clouds. Mathematician defines different metrics between these bar diagrams to find the discrepancy of topological features.

There are many choices of metrics on the persistent diagram. In PDs, the coordinates of the points have a special meaning and hence deserve to deal with them individually. Let Y, Z be two PDs, with α is a bijection between the diagonal copies and points in Y to the diagonal copies and points in Z .

Definition 2.1.1 (Hausdorff Distance). Let M, N be the multisets of points for $U = (u_1, u_2)$ and $V = (v_1, v_2)$ in \mathbb{R}^2 and $\|U - V\|_\infty$ be the maximum of $\|u_1 - v_1\|$ and $\|u_2 - v_2\|$, similarly for function ρ and ω , let $\|\rho - \omega\|_\infty = \sup_m |\rho(m) - \omega(m)|$. Formula of Hausdorff distance is,

$$d_H(M, N) = \max\{\sup_m \inf_n \|m - n\|_\infty, \sup_n \inf_m \|n - m\|_\infty\}. \quad (2.1)$$

For algorithmic computations code for Hausdorff distance are detailed in A.2.1.

Definition 2.1.2 (Bottleneck Distance). Let M, N be the multisets of points for $U = (u_1, u_2)$ and $V = (v_1, v_2)$ in \mathbb{R}^2 , $\|U - V\|_\infty$ be the maximum of $\|u_1 - v_1\|$ and $\|u_2 - v_2\|$, similarly for function ρ and ω , let $\|\rho - \omega\|_\infty = \sup_m |\rho(m) - \omega(m)|$. The Bottleneck distance defined as,

$$d_B(M, N) = \inf_{\zeta} \sup_m \|m - \zeta(m)\|_\infty. \quad (2.2)$$

The distance of two PDs, U and V with the maximum distance of two points between U and V is computed by the Bottleneck. The bottleneck distance is stable and expensive in computation as it requires one-to-one correspondence on a PD between each point. For algorithmic computations code for bottleneck distance is detailed in A.2.2.

Definition 2.1.3 (qth-Wasserstien Distance). Wasserstein distance has come into existence because Bottleneck distance is not good with the bijections details so, for two PDs M, N it is defined as,

$$W_q = \left(\inf_{\eta: M \rightarrow N} \sum_{m \in M} \|m \rightarrow \zeta(m)\|_\infty^q \right)^{1/q}. \quad (2.3)$$

With a bijection $\zeta : M \rightarrow N$ where $\|m \rightarrow n\|_\infty$ denotes infinity norm. The same correspondence issue remains in the qth-Wasserstein distance. It makes this method computationally expensive.

For algorithmic computations code for Wasserstein distance are detailed in A.2.4. Codes for the other metrics on PDs are discussed in A.2.3, A.2.5.

2.2 Persistent Landscape

A PD is a plotting of birth rate against death rate for key features of a specific type of homology groups. A persistence barcode is also a representation of a feature's lifespan, and the feature's birth to death rate is represented in the length of each bar.

An alternative function has also been proposed in known as the landscape. The points of a PD are rotated from birth-death pairs (b, d) to $(x, y) = ((d+b)/2, (d-b)/2)$. For each point, a linear function is developed and the k th landscape function

is the largest value of the tents at any point in the t horizontal axis. The landscape function can differentiate the two apparent clusters in the diagram.

Definition 2.2.1 (Persistent Module). Consider k be a field and M be a persistent module of a K -vector spaces $\{M(p)|p \in \mathbb{R}\}$ together with K linear map $\{v_p^q : M(p) \rightarrow M(q)|p \leq q\}$ such that,

1. $\forall p, v_p^p : M(p) \rightarrow M(p)|p \leq q$ is the identity map.
2. if $p \leq q \leq r$ then $v_p^r = v_r^q \circ v_p^q$.

In TDA, a persistence module is derived from the homology of a filtered simplicial complex. A barcode can represent all of the information included in a persistence module.

Example 2.2.1. Consider Y is a topological space and $g : Y \rightarrow \mathbb{R}$ is a function. $\forall p \in \mathbb{R}$ a sublevel set is define as the subset $G_p := \{y \in Y | g(y) \leq p\} \subset Y$. Note that $p \leq q \implies G_p \subset G_q$, and we have an inclusion map $i_p^q : G_p \hookrightarrow G_q \forall p \leq q$. This inclusion map originate a linear map,

$$H_n(i_p^q) : H_n(G_p; K) \rightarrow H_n(G_q; K).$$

Singular homology groups of degree $n \geq 0$ with coefficients in K , as a result we get a persistence module $HG : \mathbb{R} \rightarrow Vect_K$ given by $HG(p) = H_n(G_p; K)$ and $HG(p \leq q) = H_n(i_p^q)$.

Definition 2.2.2 (Persistent Landscape). Consider M be a persistent module the persistent landscape is a function $\lambda^* : \mathbb{N} * \mathbb{R} \rightarrow \mathbb{R}$ given by,

$$\lambda^*(K, t) = \sup\{h \geq 0 | \text{rank} M(t - h \leq t + h) \geq K\}.$$

An other way to define a persistent landscape for a PD is $D = \{(p_i, q_i)\}$ and $i \in I$, and for $p < q$, is

$$f_{(p,q)}^{(t)} = \max(0, \min(p + t, q - t)).$$

Then

$$\lambda^*(k, t) = k\text{max}\{f_{(p_i, q_i)}(t)\} \forall i \in I,$$

where k -max is the largest k th element.

As a sequence of functions the persistence landscape may be viewed as $\lambda_1^*, \lambda_2^*, \dots : \mathbb{R} \rightarrow \mathbb{R}$, with λ_k^* being the k th persistent landscape function. λ_k^* is a piecewise linear function having a slope of either -1, 0 or 1. Critical points are the points, where the slope is to change. The persistence landscape's set of critical points λ is the sum of the collection of critical points for λ_k^* . The sequences of critical points of the persistence landscape functions encode a persistence landscape that is understood by detecting the critical points.

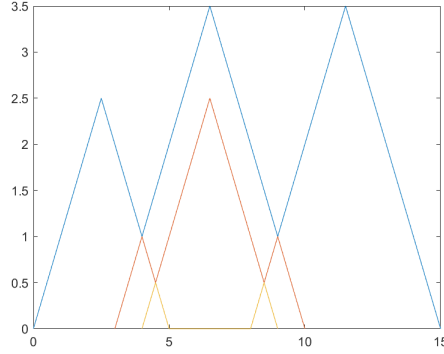


Figure 2.1: Persistent landscape

A common functional summary of a PD is the landscape function of persistence. The PDs are rotated at 45 degrees clockwise to create a landscape function and then from each point right triangles (isosceles) in the rotated persistence diagram of a specific homological dimension are drawn. Individual functions are drawn from the set of right triangles, where the point-wise maximum of all triangles is the first landscape function.

Definition 2.2.3 (Average Persistent Landscape). The average persistence landscape of the persistence landscapes $\lambda_1^*, \lambda_2^*, \dots, \lambda_N^*$ is

$$\bar{\lambda}^*(K, t) = \frac{1}{N} \sum_{j=1}^N \lambda_{(K,t)}^{*(j)}.$$

Distances between persistent landscape and average persistent landscapes can be given by applying the L^∞ norms,

$$\|\lambda^* - \lambda^{*'}\|_\infty = \sup_{(k,t)} |\lambda_k^*(t) - \lambda_k^{*'}(t)|.$$

or the L^p norm for $1 \leq p < \infty$.

$$\|\lambda^* - \lambda^{*'}\|_p = \left[\sum_{k=1}^\infty \int |\lambda_k^*(t) - \lambda_k^{*'}(t)|^p dt \right]^{\frac{1}{p}}.$$

In [8] it is shown that the persistent landscape is stable concerning the distance L^p for $1 \leq p \leq \infty$. That is, assuming a reasonable hypothesis, sufficiently minor changes in functions by the supremum norm lead to modest perturbations in the persistent landscape of the PH of sublevel sets under the L_p norm of that function.

2.2.1 Persistent landscape properties

Some persistence landscape properties are as following;

1. Invertibility. The persistence module is contained entirely within a barcode, which is a collection of intervals. We obtain the PD when there is a mapping from each interval to its endpoints. In both directions there exist maps between these topological summaries and Persistent landscape functions. The mapping is invertible from PDs to persistence landscapes.
2. Stability. For stability theorems are as follows;

Theorem 2.2.1. Consider D_1 and D_2 be two PDs and also λ_1^* and λ_2^* are their persistent landscape function. Also $\forall t$ and K , $|\lambda_{1(k)}^* - \lambda_{2(k)}^*| \leq d_B(D_1, D_2)$, where d_B denotes the bottleneck distance.

Theorem 2.2.2. Consider M_1, M_2 are two persistent modules, and let λ_1^* and λ_2^* be their persistent landscape function. Also $\forall t$ and K , $|\lambda_{1(k)}^* - \lambda_{2(k)}^*| \leq d_i(M_1, M_2)$, where d_i denotes their interleaving distance.

3. The persistence landscapes and parameters.

One benefit is there is no role of parameters in the definition the persistence landscape, so there is no risk of overfitting and no need for tuning.

4. Nonlinearity and Computability of persistence landscapes.

The nonlinearity of the persistence landscape is also an important advantage for machine learning and statistics. Where S is a summary of the linear vector of space persistent diagrams if the $D1$ and $D2$ are two persistent diagrams, then $S(D1 \cup D2) = S(D1) + S(D2)$. The landscape of persistence is not linear. The persistence landscape is computed with fast algorithms and software. The persistent landscape toolbox for persistent landscape is

available in A.3. In practice, it appears that in the calculation of the PD the corresponding persistence landscape is always slower.

5. The persistence landscape kernel. From the PDs to $L^2(\mathbb{N} * \mathbb{R})$ and provided by, the persistence landscape is an associated kernel, and its called the Persistence landscape kernel.

$$K(D_{(1)}, D_{(2)}) = \langle \lambda^{*(1)}, \lambda^{*(2)} \rangle = \sum_k \int \lambda_k^{*(1)} \lambda_k^{*(2)} = \sum_{k=1}^{\infty} \int_{-\infty}^{\infty} \lambda_k^{*(1)}(t) \lambda_k^{*(2)}(t) dt.$$

2.3 Persistence Image

The advent of computational topology has brought considerable interest in the analysis of data, which has become an area of research in its own right and that is TDA. Two standard ways to represent PH information are PD and barcodes. These tools indicate at which scale (parameters) topological features first appear are ‘born’ and no longer remain ‘die’. There is a barrier that how we can use machine learning tasks based on PDs in a parallel way. In this era, there is still not a concrete answer to when and how to use machine learning and computational topology at the same time. In [12] they propose a representation of a PD to this fundamental problem called Persistence image.

Definition 2.3.1 (Persistence image). A Persistence image (PI) a finite dimensional vector representation of a PD.

Definition 2.3.2 (persistence surface). A map of PDs, D to an integrable function $\omega_D : \mathbb{R}^2 \rightarrow \mathbb{R}$, is known as a persistence surface.

First, we map PD to an integral function called persistence surface. The stability surface ω is determined as the sum of weighted Gaussian functions centered at every point in the PD. Then, a discretization is made by the stability surface sub-domain which outputs in a mesh. As a result, PI is acquired by integrating the stability surface over every mesh square, which gives us a pixel value matrix. While calculating PIs is the wide range of weighting functions to choose from, to weigh the Gaussian functions. Usually, high stability points or lifetime are seen as more important than low stability points. In these instances, the weighting function for the persistence value of each point in the PD may be non-diminishing.

This “vectorization” of a PD brings to bear a host of new tools for comparing persistence diagrams, including all of the various metrics for measuring the distance between finite-dimensional vectors and a broad range of ML techniques.

Specifically, let D be a PD in birth-death coordinates. Consider $F : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ be a linear transformation $F(r, s) = (r, s - a)$, and let $F(D)$ be the transformed multi-set in birth persistence coordinates, where each point $(r, s) \in D$ corresponds to a point $(r, s - r) \in F(D)$. Let $\phi_u : \mathbb{R}^2 \rightarrow \mathbb{R}$ is a probability distribution which is also differentiable with mean $u = (u_r, u_s) \in \mathbb{R}^2$ we pick this distribution is to be symmetric normalized Gaussian $\phi_u = g_u$ with mean u and variance σ^2 defined as

$$g_u(r, s) = \frac{1}{2\pi\sigma^2} e^{-[(r-u_r)^2 + (s-u_s)^2]/2\sigma^2}.$$

The non-negative $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ weighting function is zero along the x-axis and continuous, and the function is differentiable in piecewise form. With these components, we transform PD to a scalar function on the plane.

Definition 2.3.3. Let D be a PD, the corresponding stability surface $\omega_D : \mathbb{R}^2 \rightarrow \mathbb{R}$ is the function

$$\omega_D(z) = \sum_{u \in F(D)} f(u) \phi_u(z).$$

The F-weighting function is defined for stable conversion from PD to a stable surface (persistence).

As an output, the surface $\omega_D(z)$ is diminished to a vector of finite dimensions by defining a relevant subdomain, and integrating $\omega_D(z)$ over every domain in the discretization. In the plane with n boxes (pixels), a mesh is particularly fixed, with every ω_D integral assigned to that region.

Definition 2.3.4. The persistent image of a Persistent diagram D is the collection of pixels $I(\omega_D)p = \int \int_p \omega_D d_s d_r$.

PI in [1] provides a suitable way to join PDs with distinct homological dimensions into one object. Suppose that in one experiment, PDs were calculated for H_0, H_1, \dots, H_k . The H_0, H_1, \dots, H_k are PI vectors that may be sequenced into

a single vector that represents the same dimensions and used as input for ML algorithms.

The three choices for constructing a PI are precision, distribution (and related parameters), and weighting. PIs are flexible, the strength is that they are not essential.

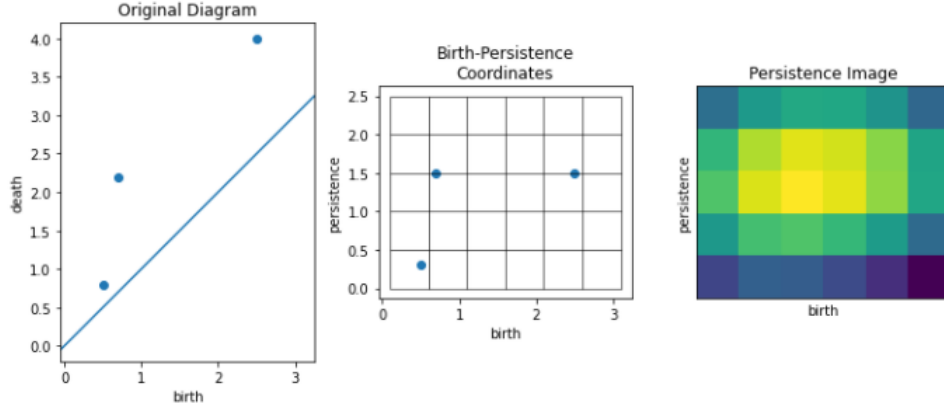


Figure 2.2: Persistence Image

The PI resolution corresponds to the superimposed grid on the PD. This approach requires the selection of a probability distribution related to every point in PD. The weighting function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ is to be continuous, piecewise, and zero along the x-axis (diagonal similarity in birth persistent axis). The algorithm for the computation of persistence images is available at A.4, and 2.2 is also generated by the same algorithm.

2.3.1 Stability of Persistence Images and Surfaces

As noise or measuring errors inevitably occur, data analyzing methods for minor input perturbations should be reliable. One of the reasons why PD is so popular in TDA is that it is consistent (Lipschitz) for bottleneck measures and stable for Wasserstein metrics given some lenient assumptions on the underlying data.

2.4 Riemannian Framework

Riemannian geometry is an entirely new framework in which PDs are approximated as 2D (probability) density functions which are denoted in the square root framework on a nice space (Hilbert). Algorithmically Wasserstein and bottleneck

distances are computationally expensive rather than the Riemannian framework. It was introduced in [2] a powerful tool that allows us to do the statistical analysis of barcodes as well.

Definition 2.4.1 (Joint Probability Density Function). Let Y_1, Y_2 be jointly continuous random variables, if there exists a positive function

$$g_{Y_1 Y_2} : \mathbb{R}^2 \rightarrow \mathbb{R}$$

where any set $L \in \mathbb{R}^2$ such that $(Y_1, Y_2) \in L$ we have,

$$\mathcal{P}(L) = \int \int_L g_{Y_1 Y_2}(y_1, y_2) dy_1 dy_2. \quad (2.4)$$

the function $g_{Y_1 Y_2}(y_1, y_2)$ here represents a joint probability density function of y_1 and y_2 .

For the case when $L = \mathbb{R}^2$, our integral becomes

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g_{Y_1 Y_2}(y_1, y_2) dy_1 dy_2 = 1.$$

When the random variable are continuous;

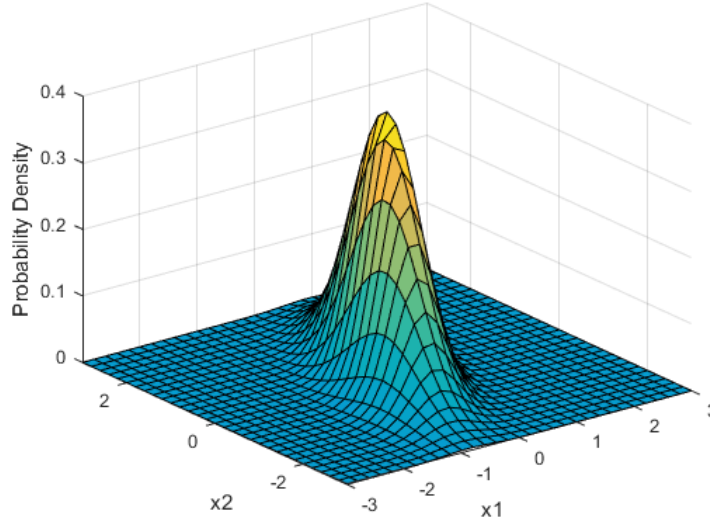


Figure 2.3: Multivariate Normal Distribution

Matlab code for the 2.3 is given in A.5.

Example 2.4.1. Let's suppose that particle movement is restricted to region L , bounded by a x -axis coordinate, line $x = 1$, and line $y = x$, respectively. Suppose

(Y_1, Y_2) is the location of the particle at a certain moment. The joint probability density function is given by

$$g_{Y_1 Y_2}(y_1, y_2) = 8y_1 y_2 \quad (y_1, y_2) \in L.$$

Definition 2.4.2. Let's assume Y_1 and Y_2 with a joint probability density function $g_{Y_1 Y_2}(y_1, y_2)$, then

$$E(Y_1) = \int_{-\infty}^{\infty} y_1 g_{Y_1}(y_1) dy_1 = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} y_1 g_{Y_1 Y_2}(y_1, y_2) dy_2 dy_1.$$

Definition 2.4.3 (Riemannian Manifold). A tangent bundle of a smooth manifold \mathcal{M} gives a vector space $V_n(\mathcal{M})$ (tangent space of \mathcal{M} at each point $n \in \mathcal{M}$) such that we have an inner product for any n

$$g_n : V_n(\mathcal{M}) \times V_n(\mathcal{M}) \rightarrow \mathbb{R}.$$

with a norm $|\cdot| : V_n(\mathcal{M}) \rightarrow \mathbb{R}$ defined as

$$|x| = \sqrt{g_n(x, x)} \quad \text{s.t for any } x, \quad |x|^2 > 0 \quad \text{whenever } x \neq 0.$$

Hence the pair (\mathcal{M}, g) is known as Riemannian manifold.

PDs are approximated as 2D or joint probability density functions in the Riemannian framework by applying kernel density estimation, with a Gaussian kernel of variance σ^2 and mean equals zero,

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

The expected values

$$E(Y_1) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} y_1 g_{Y_1 Y_2}(y_1, y_2) dy_1 dy_2, \quad E(Y_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} y_2 g_{Y_1 Y_2}(y_1, y_2) dy_1 dy_2.$$

stores the life span values of barcodes. The space of these 2D probability density functions is

$$L = \{l : [0, 1] \times [0, 1] \rightarrow \mathbb{R} \mid \forall y_1, y_2 \quad |l(y_1, y_2)| \geq 0 \wedge \int_0^1 \int_0^1 l(y_1, y_2) dy_1 dy_2 = 1\}. \quad (2.5)$$

Where L is Riemannian manifold. The square root form, that is a closed form metric, is used to get more simple geodesics. Hence (2.5) takes the form,

$$H = \{h : [0, 1] \times [0, 1] \rightarrow \mathbb{R} \mid \forall y_1, y_2 \quad |h(y_1, y_2)| \geq 0 \wedge \int_0^1 \int_0^1 h^2(y_1, y_2) dy_1 dy_2 = 1\}. \quad (2.6)$$

The $h = \sqrt{l}$ form is converted to a Hilbert unit sphere, which provides the standard metric inner product. The geodesic distance are computed between these h points is given by

$$d_H(h_1, h_2) = \cos^{-1}(\langle h_1, h_2 \rangle). \quad (2.7)$$

where

$$\langle h_1, h_2 \rangle = \int_0^1 h_1(t)h_2(t)dt.$$

Let $v \in T_h(H)$ such that $\|v\|_h = \sqrt{\langle v, v \rangle}$ and the exponential map is defined as

$$\exp_h(v) = \cos(\|v\|_h)h + \sin(\|v\|_h)\frac{v}{\|v\|_h}.$$

The barcodes are then mapped to Hilbert sphere and the geodesic to compute distances on the sphere is given as,

$$\pi(s) = \frac{(1-s)h_1 + sh_2}{s^2 + (1-s)^2 + 2s(1-s)\langle h_1, h_2 \rangle}.$$

Under the Principal Component Analysis (PCA) to our Hilbert sphere, the space of our PDs in the square root form. Now considering the mean of the PDs and mapping it to the closest point on the sphere. Hence every h_i turns into a low dimensional vector after removing the principal components of the tangent space of tangent vectors h_i from the mean. So, the distance between two PDs is computed using (1.4). The correspondence problem is also solved in this approach as geodesics are calculated between these PDFs on Hilbert Sphere. The closed-form expressions of geodesics also extract information about statistical tools, like clustering, the sample means, PCA. Hence, this method is a very good approach to study the behavior of PDs.

2.5 Persistent Homology Transform

Persistent Homology Transform (PHT) is a statistical tool to achieve statistical shape interpretation on objects and shapes in \mathbb{R}^3 and \mathbb{R}^2 . Most commonly a collection of landmark points are picked and then local information is described at each of these landmark points, combining this information gives a summary of the shape. Obtaining a representation of the shape that is applied in statistical

models is almost the central problem in modeling shapes and surfaces.

In TDA, the dominant tool is PH, which is the multi-scale approach of these homology groups to data, and the results are not lost. The maximum dimension of these homology groups' key factors is restricted by the size of data. A data representation based on the sub-level sets of a filtration function is created using simplices. During the construction, we detect changes in homology and quantify these changes in terms of birth and death rates. The lifespan of these key factors is shown as a bar code or PD. The PD is a collection of points in \mathbb{R}^2 , capture important topological and geometric information about the underlying space.

Definition 2.5.1 (Filtration in Persistent homology transform). Let Q be a subset of \mathbb{R}^d that can be represented as a finite simplicial complex, and we can construct a filtration $Q(t)$ of Q parameterized by a height h for each unit vector $t \in S^{d-1}$,

$$Q(t)_h = \{x \in Q : x \cdot t \leq h\}$$

is the subcomplex of Q that contains all the simplices in the direction t below height h .

Definition 2.5.2. The PHT of $Q \subset \mathbb{R}^d$ is the function,

$$\text{PHT}(Q) : S^{d-1} \rightarrow D^d.$$

$$w \mapsto (Y_0(Q, w), Y_1(Q, w), \dots, Y_{d-1}(Q, w)). \quad (2.8)$$

in the above equation Y_0, Y_1 , and Y_{d-1} denote the dimensions of PDs.

PHT in [33] performs statistical shape analysis on shapes and surfaces. PHT is injective when the domain is Q_d for $d=2,3$. PD track features such that quantifying either where they are born or die from the perspectives of the direction. PHT is theoretically invertible due to its injectivity.

The PHT can define a metric distance between shapes or surfaces.

$$\text{dist}_{Q_d}(Q_1, Q_2) := \sum_{k=0}^d \int_{S^{d-1}} \text{dist}(Y_k(Q_1, w), Y_k(Q_2, w)) dw.$$

$Q2$ or $Q3$ is a class of independent interset simplicial complexes that are homeomorphic to a sphere. The zeroth-dimensional PDs are adequate for this class to differentiate between the simplicial complex. Computationally the zeroth-dimensional PDs are very fast.

Proposition 3. Given a simplicial complex $Q \subset \mathbb{R}^3$ or \mathbb{R}^2 which is homeomorphic to S^2 or S^1 , we can construct $Y_k(Q, w)$ from $Y_0(Q, w)$ and $Y_0(Q, -w)$ for $k = 1, 2$.

Definition 2.5.3. The 0th-dimensional PHT of $Q \subset \mathbb{R}^d$ is a function

$$\begin{aligned} \text{PH}_0\text{T}(Q) : S^{d-1} &\rightarrow D. \\ t &\mapsto (Y_0(Q, w)) \end{aligned}$$

Example 2.5.1. Here is an example of simplicial complex looks like letter W in 2 dimensional space. An its 0th-dimensional PHT is shown in figure below. The PD is constructed in eight distinct directions using the height function. The PHT of this specific letter W in 2D included here is roughly discretized.

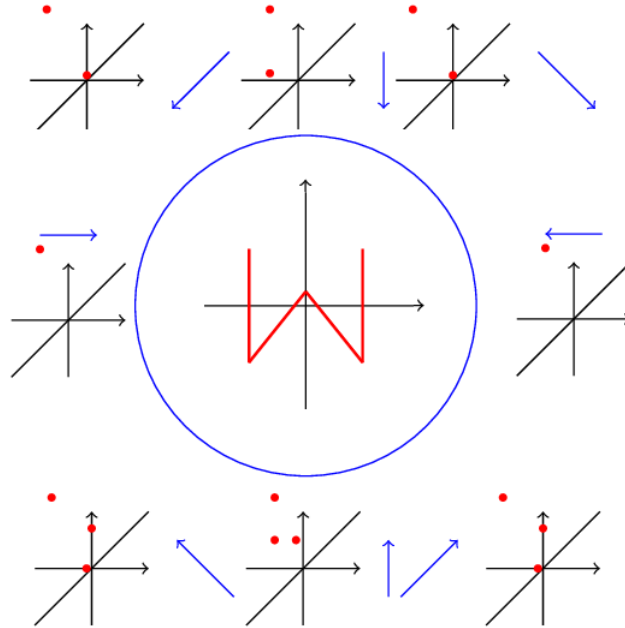


Figure 2.4: 0th-dimensional PD to the letter W filtrations

In collecting PDs of all homologies for every possible shape direction, the PHT collects shape information. In more formal terms, PHT results for d-dimensional shape in d-many persistence diagrams resulting from filtration of the hight function over infinitely many numbers of directions. The PDs are complex spaces, but

theoretically clearly defined in [21]. It is in particular a metric space, which means that distances between PDs can be defined. Due to the shape comparison, the difference between PHT summary statistics is significant. Source code for PHT is available at A.6.

It is particularly suitable for collecting shape information in every possible direction on the surface of the sphere. An important element of PHT, which we keep in consideration of all directions on the surface when developing the Smooth Euler Characteristic Transform. By expanding PHT into an extensive collection of partly linear and continuous functions in Hilbert's \mathbb{L}^2 space.

2.6 Smooth Euler Characteristic Transform

Smooth Euler Characteristic Transform (SECT) utilizes the same core mathematical concepts as the PHT, it creates continuous, partly linear functions, rather than PDs. In its application, the SECT uses the Euler characteristic, a topological invariant which takes place in several fields of Mathematics. In homology, the EC is an alternate sum of the ranks of homology groups i.e. the Betti-number, (β_k and H_k) and reduces mathematical interpretation of holes from an algebraic group structure to integer in the topological space.

Definition 2.6.1 (Euler Characteristic). The topological space S , $H_k(X)$ denote the k -th homology group of S , and β_k denote the homology group's rank. The alternating sum of S is the Euler Characteristic(EC) $\chi(S)$.

$$\chi(S) = \beta_0 + \beta_1 + \beta_2 \cdots = \sum_{k=0}^{\infty} (-1)^k \beta_k.$$

Same as the EC, for a discrete shape or three-dimensional surface, may be described by number of K simplices as a simplicial complex K ;

$$\chi(S) = V - E + F.$$

The number of vertices, edges, and faces are denoted by V , E , and F .

The generalized formula for the EC is the alternating sum of Betti numbers,

$$\chi = b_0 - b_1 + b_2 - b_3 + \dots$$

Here in 2.5 there are some examples of Euler values of different objects;




Name	Image	Vertices V	Edges E	Faces F	Euler Characteristic V-E+F
Tetrahedron		4	6	4	2
Cube		8	12	6	2
Octahedron		6	12	8	2

Figure 2.5: Euler Characteristic table

Just as filtration to persistent homology can increase homology, filtration calculated through EC. The output is an EC curve that follows the continuity of EC as a filtration function.

Example 2.6.1. Fix a direction v on the surface of a unit circle or sphere S^{d-1} (where $v \in S^{d-1}$) with the dimension $d = \{2, 3\}$.

The Euler characteristic curve (ECC) that indicates the EC value of the structure at each point of the filtration can be calculated easily, given the PH of a

filtered structure. While ECCs contain less information than PDs, for example, they are usually much easier to numerically compute using techniques not very different than Euler's nearly three centuries ago. From a practical perspective, ECCs often seem to have the majority of the useful information required for data analysis, and it has turned out, from a more theoretical point of view, that is for random structures are much more readily studied than PH or even Betti numbers at some filtration.

2.6.1 Euler characteristic transform

Euler characteristic transform (ECT) across the S^{d-1} , sphere and calculation of the associated EC curves χ_v^K from K_v for each direction of $v \in S^{d-1}$ is the finite simplicial complex representations, is defined as follows;

$$\begin{aligned} \text{ECT}(K) : S^{d-1} &\rightarrow \mathbb{Z}^{\mathbb{R}}. \\ v &\rightarrow \chi_v^K. \end{aligned}$$

The ECT of one shape gathers the EC curves of shape on the sphere surface in every direction. The EC curve and ECT function are partially constant and integer value function.

Definition 2.6.2 (smooth Euler characteristic curve). The smooth Euler characteristic curve (SEC), for a fixed direction $v \in S^{d-1}$ is describe as, $\forall n \in \mathbb{R}$ then,

$$\begin{aligned} \text{SEC}(K) : \mathbb{R} &\rightarrow \mathbb{L}^2. \\ F_v^K(n) &= \int_{-\infty}^n Z_v^K(m) dm. \end{aligned}$$

The SEC is a piecewise linear continuous function with compact support $[a_v, b_v]$ by definition. As a result, it belongs to the Hilbert space \mathbb{L}^2 of square-integrable functions on \mathbb{R} .

Definition 2.6.3 (SECT). The smooth Euler characteristic transform (SECT) of a shape $M \subset \mathbb{R}^d$ for a simplicial complex K , with $d = \{2, 3\}$, is the map

$$\begin{aligned} \text{SECT}(K) : S^{d-1} &\rightarrow \mathbb{L}^2[a_v, b_v]. \\ v &\rightarrow F_v^K(b_v) \end{aligned}$$

for all $v \in S^{d-1}$. Each curve F_v^K is also lies in the space of Hilbert \mathbb{L}^2 .

The distances between two simplicial complexes (discrete shape representations) K_1 and K_2 are calculated using the following metric,

$$dist_{M_{d-1}}^{SECT}(K_1, K_2) = \left(\int_{S^{d-1}} \|F_v^{K_1} - F_v^{K_2}\|^2 dv \right)^{1/2}.$$

The SECT overview has an advantage over the PHT in that it is a collection of curves with a Hilbert spatial structure. It enables quantitative comparisons to be made functional and non-parametric by the vast range of the statistical approach.

Corollary 1. *For two and three dimensional shapes the smooth Euler characteristic transform is injective, i.e. when the domain is M_{d-1} for $d = \{2, 3\}$.*

The SECT's injective property suggests it summarizes original shape data in concise terms. Mathematically the SECT maps M_{d-1} and Hilbert space \mathbb{L}^2 between all shapes of a finite simplicial complex. There is therefore a (once) corresponding shape between these two spaces with some finitely complex representation in M_{d-1} for a given SECT statistics in \mathbb{L}^2 . Note, however, that the corollary should have enough directions $v \in S^{d-1}$, because in any fixed direction the EC curve (whom the SECT construction depends upon) is not injective. The Euler Characteristic (EC) injectivity counterexample for a fixed direction is shown in 2.6.

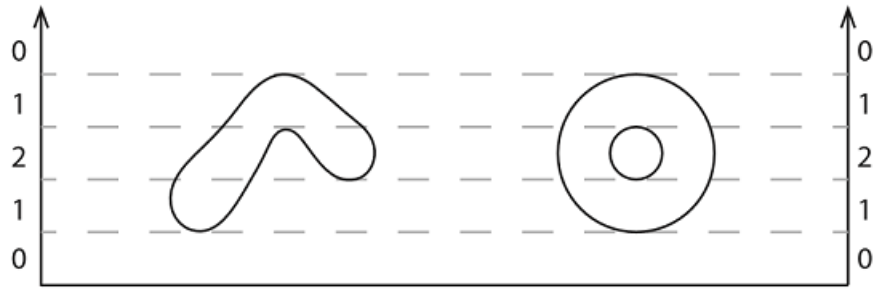


Figure 2.6: (EC) injectivity for fixed direction.

Source:[13]

Code repository for the SECT is detailed discussed in A.7. Lorin Crawford in [13] proposed an algorithm as SECT image calculation software is available publicly in the R and MATLAB code on Github.

Chapter 3

Computational Techniques for Persistent Homology

In the previous chapter we have discussed, different metric spaces that can be used in TDA. From the TDA point of view, point cloud datasets are finite metric spaces. Raw data is neither visible nor includes an exceptional key feature that motivates a topologist's interest. Algebraic topology gives us the tools for evaluating topological features from data on different scales such as persistent homology, multi-persistent homology. In this chapter, we are discussing the different computational techniques of persistent homologies that deal with point cloud datasets.

This gives the data life and the behavior of the data that can be seen. Two data sets can be distinguished based on these features. As we know the data is everywhere so the implementation of TDA is very productive. TDA is a dimensional reduction approach that maps data from its domain's high-dimensional space to a smaller, easier-to-understand, and visualize space. In TDA, the latent variables are homologies with topological features.

There is a user-friendly tool in TDA for visualization that allows topologists, as well as biologists and clinicians, to harness the potential of TDA without having any programming experience. TDA offers a wide range of high-dimensional data applications since it can handle huge datasets with tens of thousands of data points.

Several wide applications in PH are available. In [20], the weighted homology is reviewed. Where the weight value reflects the physical, chemical, and biological characteristics of simplexes. Moreover, bio-molecular analysis of DNA structures is continued by the homology of such simplexes.

In this direction of tumor detection, the TDA also makes a bond with machine learning and artificial intelligence. Neural networks based on PH were used as a function vector for detection of cancer, as in [26]. The approach to handling biomolecular data is not limited to PH, but multidimensional persistence is used in [35] assessment of bio-molecular data.

TDA is very popular, and every application cannot be discussed. The latest work is discussed from several fields which prompt us to view different kinds of data (pictures, point clouds, networks, etc.) through TDA tools. Concentrating on PH means large data sets with the computing tools for PH when all these applications are involved.

Persistent homology calculations change quickly. Algorithms and packages produce less computational and more efficient compared to others in this field. Multiple packages calculate PH with different features and limitations. For calculating PH steps are discussed in 3.1:

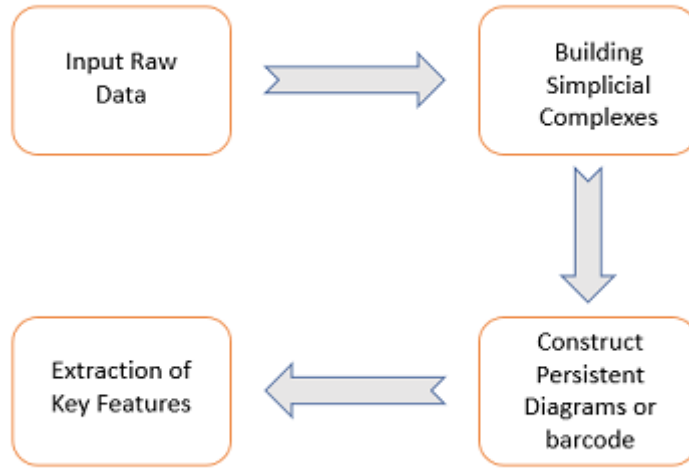


Figure 3.1: computational steps for persistent homology

For the calculation of persistent homology, the prominent libraries are:

1. JavaPlex
2. Ripser
3. Dionysus
4. Perseus
5. PHAT
6. GUDHI
7. DIPHA
8. Persim

3.1 JavaPlex

The main reason for Javaplex’s existence is to provide a new uniform library to researchers in the field of TDA. The fourth version of the Plex family is the javaplex package. The computational topology research group’s members have developed these programs during the last decade. Each consecutive version incorporated the results of new developments in the field of computational topology and TDA which are relatively rapidly developing.

Javaplex is a Java software package that is used to execute the persistent homology of filtered simplicial complexes (or chain complexes in general) with a particular emphasis on TDA applications. Andrew Tausz is the main author of this software. Javaplex is a rewriting of Harlan Sexton’s and Mikael Vejdemo–JPlex Johansson’s package. A flexible platform supporting new directions in topological data analysis and computational persistent homology has been the principal motivation for Javaplex development.

The PH of the point cloud data calculated in Javaplex, necessary to consider the Vietoris Rips complex, and Javaplex code is available in MATLAB and discussed in B.1, and figure 3.2 shows javaplex visualization of the above point cloud data in the form of barcodes.

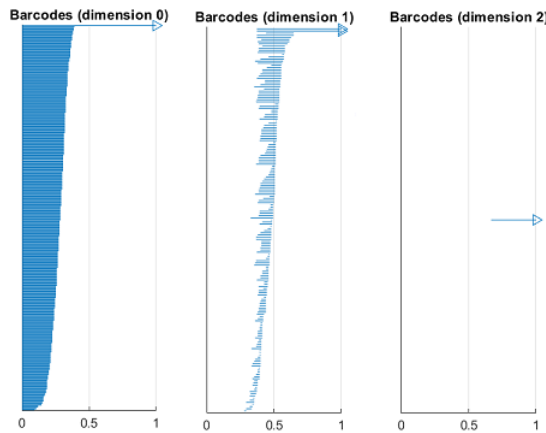


Figure 3.2: Barcodes using JavaPlex

3.2 Ripser

Ripser is a Python lean PH software. On the basis of blazing quick C++ Ripser as the core computer engine, an intuitive interface is provided for Ripser in [3];

1. Calculating sparse and dense data sets persistence cohomology.
2. Visualizing PDs.
3. To compute lowerstar filtrations on images.
4. To compute representative cochains.

Ripser is the development of the original project C++ Ripser. Currently, Ripser outputs are faster in terms of 40 time computationally and 15 times in memory efficiency than other codes (Dionysus, DIPHA, GUDHI, Perseus, PHAT). PHAT does not include Vietoris–Rips filtration generation code.

Ripser code for core generation C++ is available in Python and is given an example in B.2. For this dataset, we draw the generators' lifetime in Python, with Numpy, sklearn, matplotlib, and persim, and it's shown in 3.3.

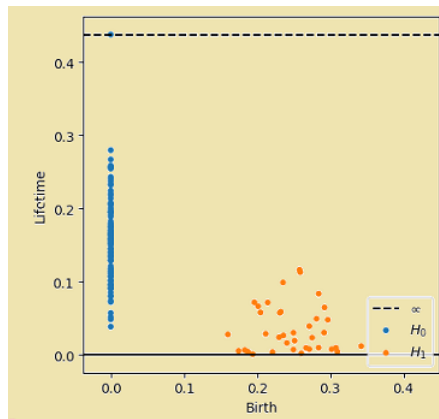


Figure 3.3: Lifetime of generators

Ripser supports the following input data:

- Comma-separated values full distance matrix.
- Point cloud data.
- Binary lower triangular distance matrix.

- DIPHA distance matrix data.
- Comma-separated values lower triangular distance matrix.
- Comma-separated values upper triangular distance matrix.
- Sparse distance matrix in sparse triplet format.

By mathematical and algorithmic opportunities in [36] Ripser++ utilizes massive parallelism in the calculation of Vietoris-Ripps. Up to 30x speed can be achieved over the overall runtime of Ripser, up to 2.0x memory per CPU, and up to 1.58x less memory used in GPU than Ripser’s CPU.

There are two phases of computation after dimension 0 in the original Ripser: the filtration construction with clearing and the reduction in the matrix. With Ripser++, the filtration structure is massively paralleled to the clearing stage, extracting hidden parallelism from the matrix reduction “apparent pairs” of the whole GPU, leaving the CPU calculation of the sub-matrix reduction of the remaining non-apparent columns. Up to 99.9% of the columns in a cleared co-boundary matrix are visible in [36] empirical findings.

It has been found in [36] that Ripser intensively operates on many datasets and has a lack of efficiency on CPU. Two main performance problems exist. In every dimension, to process every single simplex, Ripser’s matrix reduction uses an enumeration and column addition style. The calculation is very dependent on the columns, however.

For Ripser to process such columns one by one, the sequential frame of Figure 3.4 contains rich parallelisms deriving from a huge percentage of pairs **apparent**. Secondly, the application of clearing lemma and predefined thresholds is separate from simplifying in the filtration structure with the clearing stage.

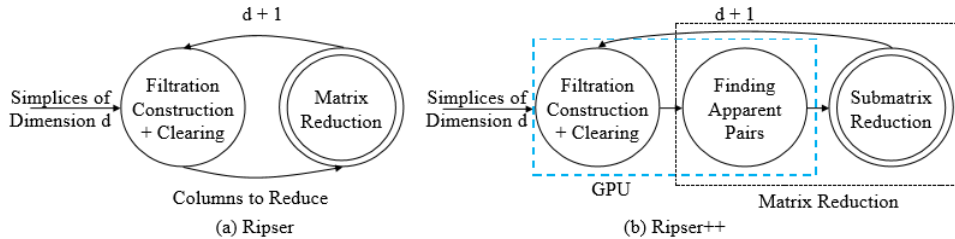


Figure 3.4: Framework for Ripser and Ripser++

Source: [36]

3.3 Dinoysus

Dinoysus is a PH-focused computational topology package. The Dinoysus algorithm, written in C++, is binding with python. This is a rewritten version of the previous one. Some dependencies are as follows for dinoysus;

1. Boost.
2. Scipy.
3. Matplotlib.

As a Python packaging, the easiest way to install Dionysus is with PyPI; and available at B.3. In Dinoysus a sequence of homology groups linked by linear map is obtained to calculate the PH of a simplicial complex, using the homology functor to filtration.

$$H_*(K_1) \rightarrow H_*(K_2) \rightarrow \cdots \rightarrow H_*(K_n).$$

To determine the decomposition of this above sequence the command is discussed in B.3, i.e. persistence barcode, which returns the interior representation of the reduced boundary matrix. The following algorithms are implemented by Dinoysus;

- Persistent homology computation.
- Persistent cohomology computation.
- Zigzag persistent homology.
- Alpha, Rips, and Cech complex construction.

3.4 Perseus

After certain homological Morse theoretical reductions are first made, Perseus calculates the PH of various types of filtered simplicial complexes. In all circumstances, the user should save the input filtration as a text file with correct formatting (see formatting guidelines below) and read the PH intervals output, which is also provided as text files.

Perseus effectiveness is not dependent on the peculiarities of a specific complex shape or dimension because it is based on discrete Morse theory. This being said, in the field of data analysis, certain types of complexes are much frequent. It's best to work with cubical data structures when dealing with films and images.

The suitable representation, on the other hand, consists of top-cell information on a simplicial complex with a manifold triangulation as the data source. Complexes based on Vietoris-Rips are commonly used to process point cloud data. Here's a quick reference guide to the most popular complex types:

- **cubtop**: For a dense, top-level cubical grid information.
- **scubtop**: For a sparse, high-level, cubical grid information.
- **sintop**: Top-level triangulation data for the simplicial complex in a uniform format.
- **nmfsintop**: A general, simplicial complex from high-level simplex information that is not uniform.
- **rips**: Vietoris-Rips complex has a non-uniform birth point information.
- **brips**: For a uniform information birthplace Vietoris-Rips complex.
- **distmat**: For a distance matrix Vietoris-Rips complex.

3.4.1 Persistent Homology of Vietoris-Rips Complexes

Vietoris is entirely defined by the 1-skeleton underneath. This skeleton is represented as a symmetric distance matrix, with entries originating from points in a point cloud or a number of different sources. The PH of the three forms of data created by the Vietoris complexes: uniform birth-points, non-uniform birth-points, and distance matrices can all be calculated using Perseus.

3.5 PHAT-Persistent Homology Algorithm Toolbox

PHAT is an Open-Source C++ library aimed at the development of software for TDA to compute PH by reduction of a matrix. They provide a wide variety of re-

duction strategies and data types in [4] for storing and manipulating the boundary matrix. The various combinations are compared by extensive experimental evaluations and optimization techniques can be identified which are good for practical situations.

In addition, PHAT provides a generic framework for reducing boundary matrices. To store column matrices during reduction processes, PHAT provides several data structures. The effect of selecting a column representation is overlooked by other libraries that maintain homology and the GUDHI library's simplex tree is an exception.

3.6 GUDHI

Gudhi library provides an open-source library for the analysis of computed PH and TDA. It provides cutting-edge algorithms to design different types of simplicial complexes, data structures, and algorithms to calculate geometrical approximations and persistent homologies.

The GUDHI library offers the following modules:

- Complexes:
 1. Simplicial: Witness, Rips, Cech, and Alpha complexes.
 2. Cubical.
 3. Cover: Nerve and Graph induced complexes.
- Topological descriptors tools:
 1. Bottleneck distance.
 2. Statistical tools.
 3. Persistent diagrams and barcodes.

3.7 DIPHA

DIPHA is a C++ software package to calculate the persistent homology. DIPHA (Distributed persistent homology algorithm). In addition to allowing parallel execution on a single system, DIPHA may be applied on a cluster of many machines

using MPI. The founder of this project is Jan Reininghaus. DIPHA currently supports three types of input are as following;

1. **d-dimensional gray-scale image data.** Internally, the data is interpreted as a weighted cubical cell complex. and is filtrated with a lower star for the following persistence calculation.
2. **distance matrix data.** The data is analyzed as a Vietoris–Rips complex with as many points as the matrix’s columns. Then the input dates define the distance between any two points. For extended data analysis, DIPHA provides sparse distance matrices, which only encode finite distances to save space.
3. **regular cell complexes with a weighted (co-)boundary matrix form.** This domain type (fall-back) permits the calculation of the PH of e.g. Alpha complexes, complex estimation of Vietoris–Rips or Witness complexes.

The DIPHA output has a persistent diagram. The size of the homological feature it represents and the corresponding value of birth and death shall define each item in the diagram. Input and output are done with binary files with the format below specified. MATLAB features in DIPHA to create and visualize input files.

3.8 Persim

Persim is a Python package for a variety of tools for the analysis of persistent diagrams. It currently includes implementations of the most popular persistence diagram working methods;

- Persistence Images.
- persistent Landscape.
- Sliced Wasserstein Kernel.
- Diagram Plotting.
- Bottleneck and Modified Gromov–Hausdorff distances.

There are also other packages for computing the homological features such as CHomP, Rivet, Hera, and SimPers are in the space of C++. Another platform for computing the computational topology is Eirene in Julia.

3.9 Comparison Analysis

We are interested in our data for buildings such as *Čech* and Vietoris-rips because they regard the data set as their vertex. Because of the high number of connections, the *Čech* setting made it more computationally expensive than Vietoris-rips and the simplest thing in a greater scope than underlying data space. In contrast, Vietoris-rips builds the complex with only data points on the edge. In general, for a further evaluation of the persistent homology of our data sets, we consider Vietoris-rips construction.

In [25] every package was discussed thoroughly, and Ripser’s results show that it is the most successful library in building Vietoris-rips. A single example taken from the [25] table, for time efficiency of ripser, in the construction of Vietoris-rips, over other libraries, in 3.1 table. The weighted undirected network with each node representing a neuron and its edges representing a snap or gap crossover, called **eleg**, is given.

On this data set, for the maximum dimension of 3, the PH of the Vietoris-rips complex is calculated. The threshold (ϵ) was the maximum distance between two points from the data set. The size of the simplicial complex is 3.2×10^8 . The following table shows the time spent for each package on the computer, where “—” indicates that the package is not computed. The following table shows the time spent on each package for the computation.

Package	time in seconds
Dionysus	—
Ripser	2
GUDHI	381
DIPHA	926
JavaPlex	13,607
Perseus	—

Table 3.1: Elapsed time in seconds for each package

In the case of time elapsed in seconds, the ripser here clearly dominates during the calculation. After Ripser the DIPHA and GUDHI are preferred for the case of Vietoris-rips. About the size, Ripser is no less than the other packages. Therefore, dealing with the maximum size of a simplicial complex, Vietoris rips is currently the quickest to compute persistent homology.

In the conclusion of the benchmarking in [25], they suggested that Ripser Package calculates most effectively the distance matrix data established with the Vietoris-rips construction. So in our PH calculation, we considered Ripser.

In 3.2 we are showing the different modalities of the packages in Matlab, likewise, their installation, dealing with complexes, and visualization. Most of these packages are also work in the space of Python and C++.

Ripser	GUDHI	DIPHA	Dinoyesus	Persus	Javaplex	Software
---	---	---	---	<	<	Installation
<	<	<	<	<	<	Complex
<	<	<	<	<	<	Boundary matrix
<	<	<	<	<	<	Barcodes
<	---	<	---	<	<	Visualization
Large	Large	Large	Medium	Small	Small	Data set size
easy	hard	hard	medium	easy	easy	Ease of Use

Table 3.2: Topological Data Analysis Tools

Chapter 4

Morphological and Geometrical Features

The recovered features describe the texture or contour of the segmented pattern in general, assisting in reducing the picture's dimensionality to produce more useful and less redundant results than the original image. The basic goal is to extract illustrative information from an image in the same way that domain experts can. The second most challenging task is the proper selection of the features in the field of automatic classification of Leukemia cells. Different articles were to be studied and also published to construct an effective set of features.

In this work, we implemented different morphological so-called topological, and statistical or texture features. Convolution Neural Network (CNN) is another approach to extract features, which extract the features vectors.

4.1 Features Extraction by Image processing

Image processing is a collection of tools to perform certain operations on an image to obtain some useful information from it. It is like a gateway that transmits the image from it to give its specific characteristics or features an output after performing several operations. Image processing is currently one of the fast-growing technologies.

There are mainly two types of image processing:

1. Analogue image.
2. Digital image.

Definition 4.1.1 (Analogue Image). The manipulation of hard copies, like prints and photographing, is called analogue image processing.

Definition 4.1.2 (Digital Image). Digital image processing techniques are useful for manipulating computers in digital images. The three general phases of pre-processing, optimization and display, information extraction, which all types of data need to undergo while using digital image processing.

Image processing in biomedical imaging is very applicable, as the only tool to see if or not the cell is infected may be in many disease detections, for example in cancer classification.

Here below is a pictorial depiction of a leukemia cell,

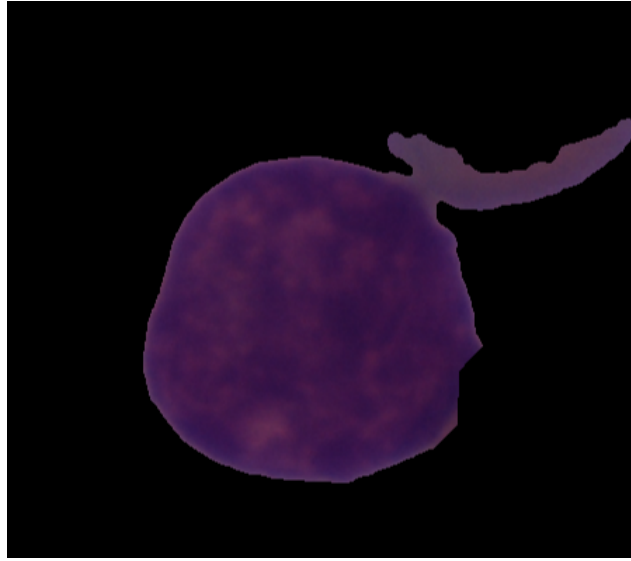


Figure 4.1: An infected blood cell in 2D

Here below there is a three dimensional depiction of 4.1.

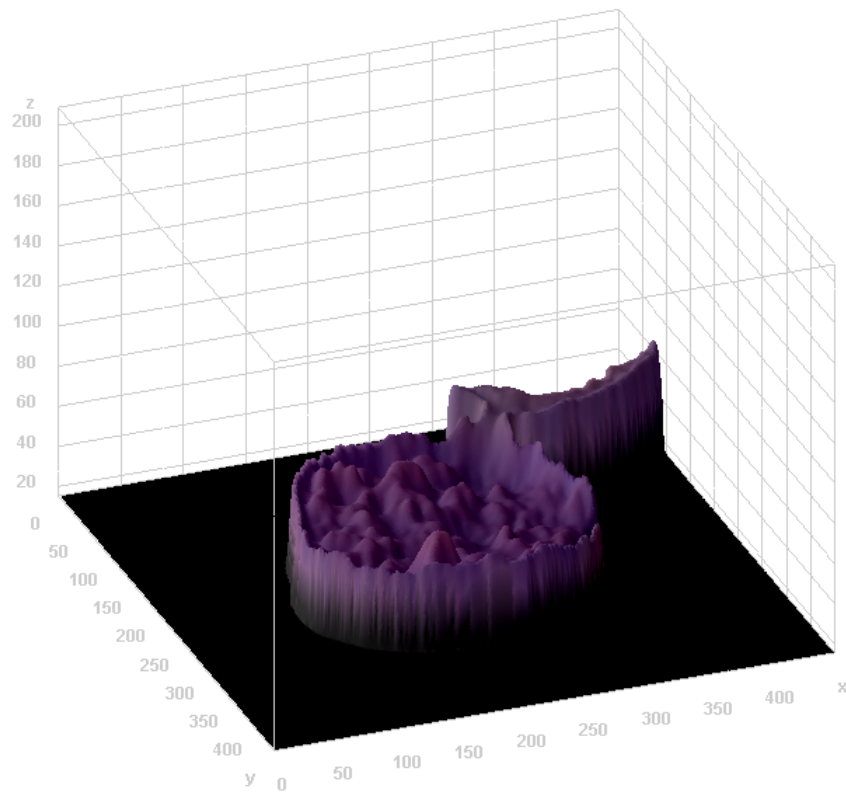


Figure 4.2: An infected blood cell in 3D

Image processing techniques can generally be summarized in the following points for digital images,

- Image Representation.
- Image Preprocessing.
- Image Optimization.
- Image Analysis.
- Image Compression.

4.1.1 Image Representation

The digital image is the function $g : \mathbb{R}^2 \rightarrow \mathbb{R}$, such that $(a, b) \in \mathbb{R}$ is the location of the function is \mathbb{R} and $g(a, b) \in \mathbb{R}$ that point is the intensity. The intensities are also referred to as the pixel. A digital image is the finite array of these pixel values.

4.1.2 Image Preprocessing

This step allows digital images to be improved, to specify the interesting attributes. The attributes are highlighted by using multiple tools, snipping, resize, denoising, segmentation, and morphological activities. This enables us to improve the distribution of images in front or background or to throw away inappropriate areas. Let us discuss commonly used image segmentation and morphology techniques.

Morphology is a wide range of imaging processes that process shape-based images. Morphological activities are applied to an image input to create an equal size output image. The pixel value in the output image is dependent on the comparison, during morphological processes, of the pixels in the input image with their neighbors.

Morphological Dilation and Erosion

The most important morphological procedures are dilation and erosion. Dilation adds pixels to picture borders, whereas erosion removes pixels on object boundaries. The addition or removal of pixels from an illustration depends on the size

and shape of the structure element used to treat the image. In morphological dilation and erosion procedures, the status of each pixel is determined by applying a rule in the input image to the pixel and the neighbors.

The morphological dilation of the output pixel is the highest value for each pixel in the neighborhood. In a binary image, a pixel is set to 1 if one of the surrounding pixels is set to 1. The output pixel value is the least value for all adjacent pixels during morphological erosion. A pixel in a binary image is defined as 0 when one of the nearest pixels is 0.

Definition 4.1.3 (Erosion). Erosion of binary image $A \subseteq \mathbb{R}^n$ with structuring element B is define as,

$$A \ominus B = \bigcap_{b \in B} \{a - b : a \in A \quad \wedge \quad b \in B\}.$$



Figure 4.3: Erosion of an Image

Definition 4.1.4 (Dilation). Dilation of binary image $A \subseteq \mathbb{R}^n$ with structuring element B is defined as,

$$A \oplus B = \bigcup_{b \in B} \{a + s : a \in A \quad \wedge \quad b \in B\}.$$

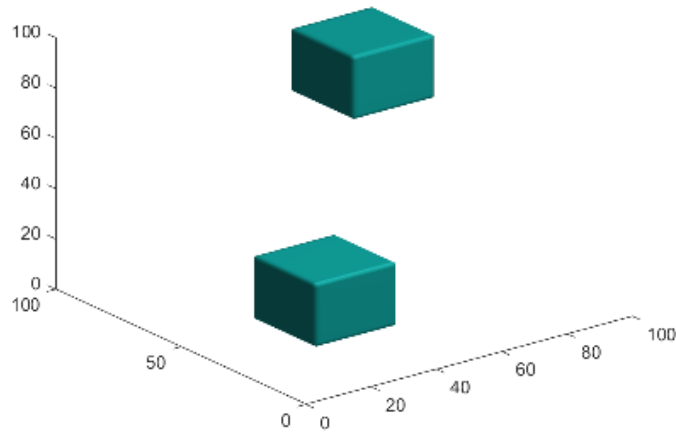


Figure 4.4: Dilation of an Image

Now let us talk about some commonly used image preprocessing **segmentation** technology.

- K-means clustering.
- Superpixels.

Segmentation methods split the image into several groups of regions desired and unwanted. K means clustering is an image segmentation with k centroids in k clusters. It's an unsupervised algorithm for machine learning. This algorithm's objective is to minimize the following function,

$$\sum_{n=1}^k \sum_{m=1}^p ||x_m - c_n||^2.$$

Where x_m are the p elements of each k cluster with c_n centroid. In image preprocessing, clustering means that the useful regions are separated from the background. For image segmentation, it is a useful measure.

Superpixels, on the other hand, are also used to segment an image. It consists of several pixels that share some common features, like the intensity of the pixels, intensity of the media, etc. These superpixels represent the multiple regions in a single image. For superpixel generation, the SLIC algorithm is used. In [37], they are discussed more properly. For each preprocessing tool, the various algorithms are available on MATLAB you can enjoy it easily.

4.2 Image Optimization

Image optimization are classified as methods of spatial and frequency domain. An image is enhanced in two ways by spatial methods.

- Gray scale transformation.
- Histogram equalization.

Gray scale transformation: Transformation of the grey scale is a simple transformation to work with each input image pixel in the 0,1,2,...,255 set. The process is conducted in two ways,

1. Average method.
2. Luminosity method.

The colored image is turned into a grey image using each pixel's average values in the **Average method**.

$$(Red + Green + Blue)/3.$$

This transforms the greyscale of the input image, but the luminosity has been lost. The **Luminosity method**, however, maintains the image luminosity during transformation. The formula is

$$0.3Red + 0.59Green + 0.11Blue.$$

Red color effect reduction and green color improvement during blue proportion adjustment. This eliminates image saturation and hue without decreasing luminosity.

Histogram equalization: Equalization of histograms is a common technique for improving the image appearance. The histogram is used for the visual properties assessment of the image. If the picture is mostly dark then the histogram is skewed in the lower part of the greyscale, and if it has a brighter part it is skewed in the higher part histogram.

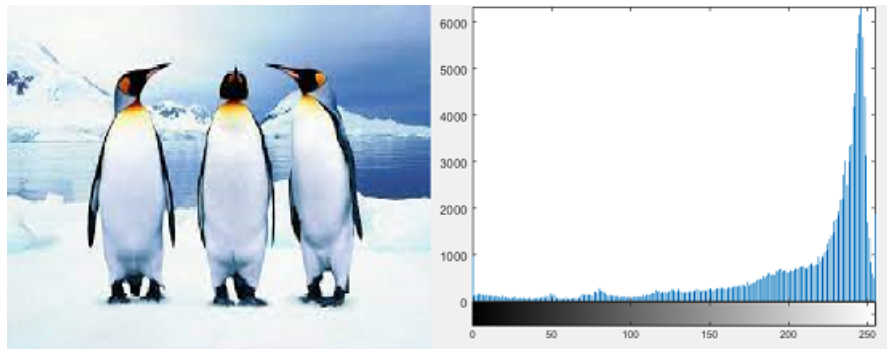


Figure 4.5: An RGB image transformed into histogram visualization

4.3 Image Analysis

In this area of image processing with scene analysis, pattern recognition, edge detection, extraction of local features (centroids, solidities, excentricity), and much more the principal features are examined. The numerical image data is generated.

4.4 Image Compression

Image compression is the process of coding information using fewer bits (or other data-carrying units) than would be used by the encoded representation using special encoding schemes. Compression is helpful because it helps decrease costly resource consumption, such as disc space or bandwidth (computing). The compression of images is an application of digital image data compression. Image compression reduces bytes in a graphics file to a minimum and does not decrease the image quality to an unacceptable level.

One of the techniques is lossy compression, using inaccurate approximations and partial discarding of data. Lossless compression is better than loss compression to get a high content image. This method is better when small image information is available, such as biomedical images, technical drawings, etc.

4.5 Morphological features

According to a Hematologist, Nucleus shape is a good critical component for immature cell identification. In [6] different geometrical and texture features are given in detail. Apart from basic measurements like the nucleus and cytoplasm area and nucleus perimeter, the following shape indicators were taken into account:

Nucleus-Cytoplasm Ratio

The ratio of the area of cell nucleus and cytoplasm. For the recognition of cell maturity, this key feature is important. There is an inverse relationship such that nucleus size decreases while increasing the maturity of leukocytes.

Compactness of Nucleus

The degree to which WBCs are compacted is determined by their maturity and type, as well as the nucleus shape. Multi lobed nuclei with lobes linked by bands or thin strands are common in mature cells. In certain circumstances, the nucleus resembles a horseshoe-shaped contour or a kidney bean.

Leukemia cell nuclei, on the other hand, are ovoid or spherical and have higher overall compactness than mature cell nuclei. The compactness formula is as follows:

$$\text{Compactness} = \frac{\text{perimeter}^2}{\text{area}}.$$

Nucleus form factor

The size of the object has no bearing on shape abnormalities. A round nucleus, in general, has the largest area to perimeter ratio, which is equal to 1 for a perfect circle. As a result, the ratio departing from roundness has a smaller value in normal cells, whereas it converges to 1 in leukemic cells. The form factor is calculated using the following formula:

$$\text{Form factor} = \frac{4*\pi*\text{area}}{\text{perimeter}^2}.$$

Eccentricity of Nucleus

Nucleus eccentricity indicates a departure from a round shape. This important component is calculated as the ratio of the ROI's (region of interest) smallest bounding rectangle's length and breadth. In contrast to the shape factor, this measurement includes the nucleus elliptic or circular lobes.

Elongation of Nucleus

Abnormal bulging is indicated by nucleus elongation. This key feature is calculated as the ratio of minimum to the maximum distance from the center of the object to the boundary.

Solidity of Nucleus

The concavity of the shape where the shape is convex or concave is defined by nucleus solidity and the formula to compute is

$$\text{Nucleus solidity} = \frac{\text{area}}{\text{convex hull area}}.$$

Here below there are some morphological features,

Morphological features					
UID_1_2_1_all.bmp	Area: 49531	Eccentricity: 0.7027	Solidity: 0.8072	Perimeter: 1.1526e+03	MaxFeretAngle: 150.2374
UID_1_2_2_all.bmp	Area: 37785	Eccentricity: 0.4880	Solidity: 0.9920	Perimeter: 694.2370	MaxFeretAngle: -114.9590
UID_1_3_1_all.bmp	Area: 48169	Eccentricity: 0.6547	Solidity: 0.9805	Perimeter: 811.0270	MaxFeretAngle: -123.2489
UID_H6_1_1_hem.bmp	Area: 37429	Eccentricity: 0.5058	Solidity: 0.9796	Perimeter: 702.3320	MaxFeretAngle: -176.1378
UID_H6_1_2_hem.bmp	Area: 46776	Eccentricity: 0.4231	Solidity: 0.9857	Perimeter: 786.1350	MaxFeretAngle: -100.6729
UID_H6_2_1_hem.bmp	Area: 41905	Eccentricity: 0.4652	Solidity: 0.9808	Perimeter: 750.6880	MaxFeretAngle: 118.8798

Table 4.1: Morphological features of Normal and ALL cells

4.6 Texture features

Other essential key features based on variations in the nuclear chromatin pattern reflecting DNA creation and on cytoplasmic variations are used for the recognition of leukemia blast cells. To detect the important information of the nucleus and the entire cell just like the structural arrangement there were two types of statistical quantities.

In greyscale image the first order statistical measure are built on the theory of histogram e.g the nucleus and cytoplasm mean color, and from gray level co-occurrence matrix (GLCM) the second order statistical measure is derive, which transmits information about the 3D relationships of the image pixels. The 2nd-order statistical key factors are defined by the equations below, where $T(x, y)$ is the element of the normalized (GLCM) at the coordinates x and y , N_g represents the number of different gray levels and μ_i , μ_j and σ_x , σ_y denotes the standard deviations (SD) and means of the normalize (GLCM).

Nucleus Energy

Texture base local information of gray scale image and this property is also known as uniformity of energy and its formula is,

$$\text{Nucleus Energy} = \sum_{x,y}^{N_g-1} (T_{x,y})^2.$$

Nucleus Correlation

In GLCM the nucleus correlation denotes the linear dependency of gray tone values. Formula for the calculation of nucleus correlation is defined as,

$$\text{Nucleus Correlation} = \sum_{x,y} \frac{(x-\mu_x)(y-\mu_y)T(x,y)}{\sigma_x\sigma_y}.$$

Cell Contrast

In GLCM the number of local variations is called the cell contrast and formula is given as,

$$\sum_{x,y} |x - y|^2 T(x, y).$$

Homogeneity

In GLCM the closeness of the distribution of element is represented by homogeneity and given by,

$$\sum_{x,y} \frac{T(x,y)}{1+|x-y|}.$$

Cell Entropy

Entropy is a statistical measurement of randomness that is used to characterize the texture of the domain image,

$$- \sum (T \cdot \log_2(T)).$$

where T contain the normalized histogram count.

The above texture features are given below in a table and to drive the statistics from GLCM and correlation plotting of different images is given as above. In refgra First three graphs in first row are of the Leukemia cells and graphs in second row are of the normal cells.

Texture Features					
UID_1_2_1_all.bmp	Contrast: 0.0141	Correlation: 0.9710	Energy: 0.6183	Entropy: 2.2794	Homogeneity: 0.9948
UID_1_2_2_all.bmp	Contrast: 0.0129	Correlation: 0.9606	Energy: 0.6957	Entropy: 1.7971	Homogeneity: 0.9962
UID_1_3_1_all.bmp	Contrast: 0.0197	Correlation: 0.9710	Energy: 0.6219	Entropy: 2.2979	Homogeneity: 0.9936
UID_H6_1_1_hem.bmp	Contrast: 0.0123	Correlation: 0.9753	Energy: 0.6866	Entropy: 1.8173	Homogeneity: 0.9957
UID_H6_1_2_hem.bmp	Contrast: 0.0167	Correlation: 0.9663	Energy: 0.6364	Entropy: 2.2382	Homogeneity: 0.9950
UID_H6_2_1_hem.bmp	Contrast: 0.0152	Correlation: 0.9772	Energy: 0.6491	Entropy: 2.0153	Homogeneity: 0.9926

Table 4.2: Texture features of Normal and ALL cells

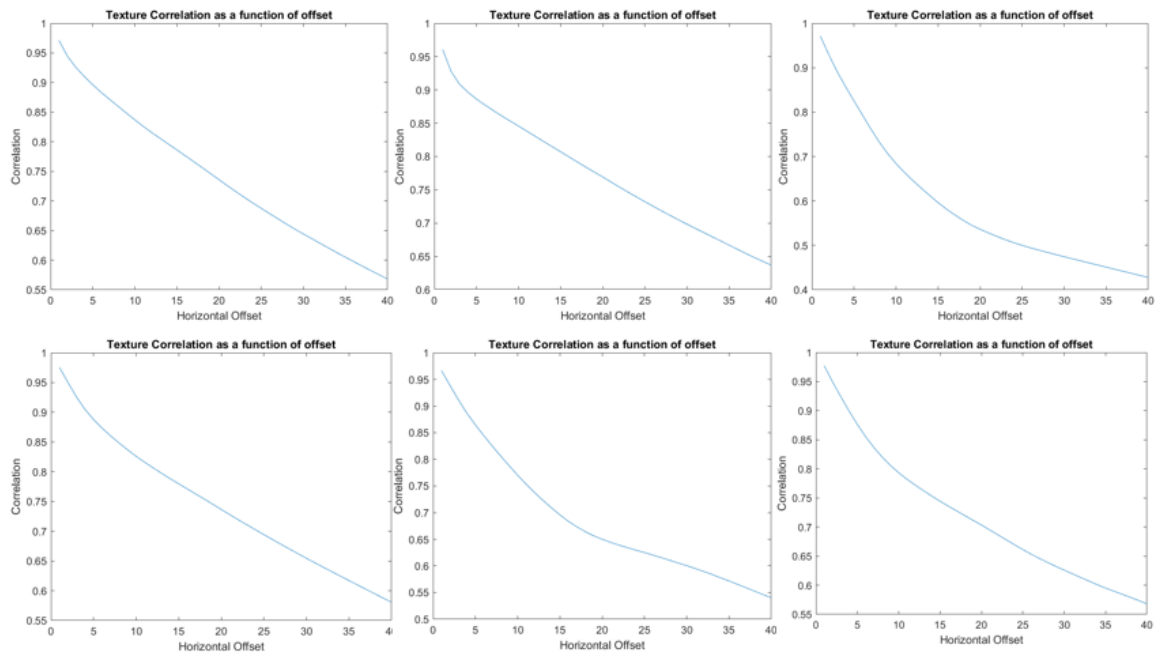


Figure 4.6: Correlation graph

Chapter 5

Classification of Acute Lymphoblastic Leukemia Cells

5.1 Basic Hematology

Blood is a body fluid that flows in the blood vessels of all animals. RBCs (red blood cells), WBCs (white blood cells), and Platelets are the three primary components of blood. In hematology, which is the study of blood, changes that occur in the function and shape of leukocytes are also called leukocyte abnormalities. Leukemia, myeloma, and lymphoma are the three primary forms of blood cancer. Leukemia is caused by the excessive increase of WBCs and classified into two forms Acute (most aggressive and fast-growing) and Chronic (slow-growing). A pictorial depiction of WBCs is in 5.1 given below,

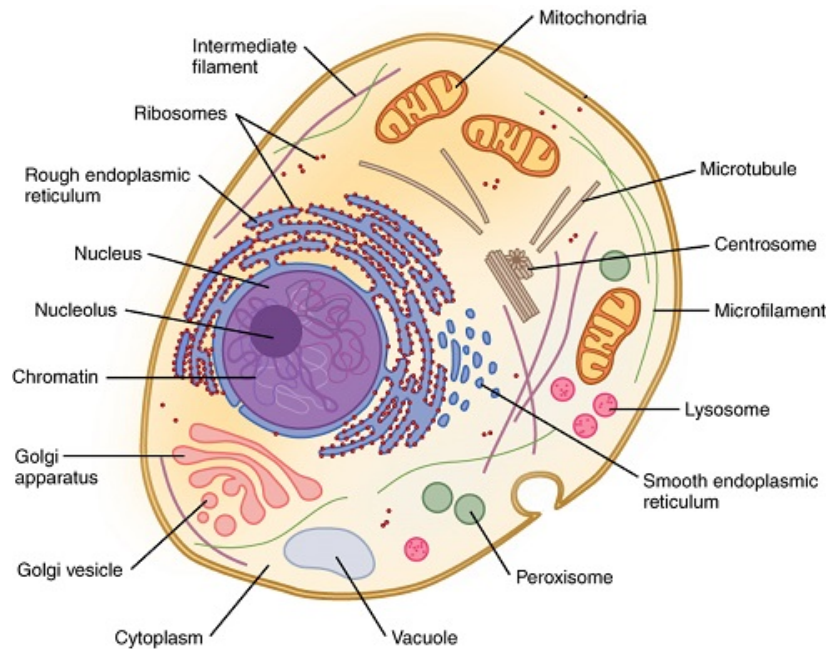


Figure 5.1: Prototypical Human Cell

Source: [31]

Leukemia symptoms are as following:

- Fever
- Weight loss
- Fatigue
- Pain in joints

Acute Lymphocytic Leukemia (ALL) which is also called Acute Lymphoblastic Leukemia is a type of pervasive childhood blood cancer that occurs with rapid and continuous production of WBCs and after all, it disturbs the immune system. B-cell ALL and T-cell ALL are the other two major kinds. B- cells, which are immature WBCs present in the blood and bone marrow, are the most common type of ALL. The highest charge of ALL being in kids among three and seven years old with 75% of diagnoses taking place earlier than the age of 6.

5.2 FAB (French-American-British) Classification

A number of classifications of hematological diseases are defined in French American British (FAB) classification systems. It was released for the first time in 1976. ALL is divided into three subtypes under the FAB categorization system. These types are defined according to different criteria such as the appearance of individual cytologic features and the degree of heterogeneity among the leukemic cells.

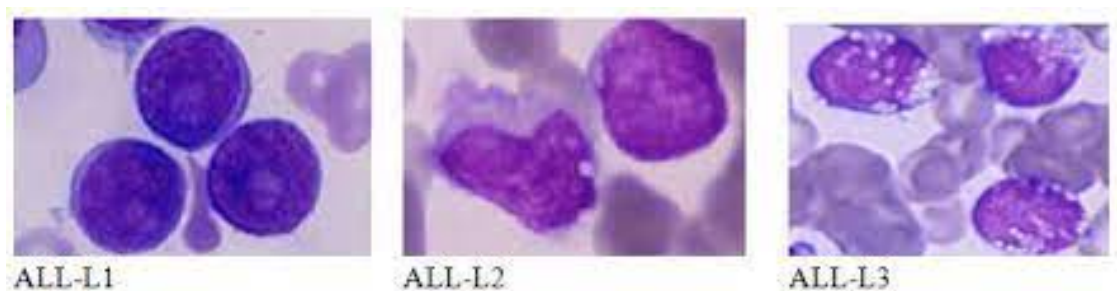


Figure 5.2: Subtypes of Acute Lymphoblastic Leukemia

Source:[19]

The FAB Classification recognize three subtypes of ALL.

1. L1
2. L2
3. L3

In L1 there are small predominate homogenous cells with scanty cytoplasm. The size of the nucleus is homogenous with a regular shape. The nucleus to cytoplasm

ratio is high and also there are invisible nucleoli. In L2 there are large heterogeneous cells with variable to often moderately abundant cytoplasm. The size of the nucleus is variable of heterogeneous with irregular clefting. The nucleus to cytoplasm ratio is low and also there are prominent nucleoli. In L3 the size of the nucleus is largely homogenous with moderately abundant cytoplasm. The size of the nucleus is homogenous with a regular shape. The nucleus to cytoplasm ratio is low and also there are prominent nucleoli.

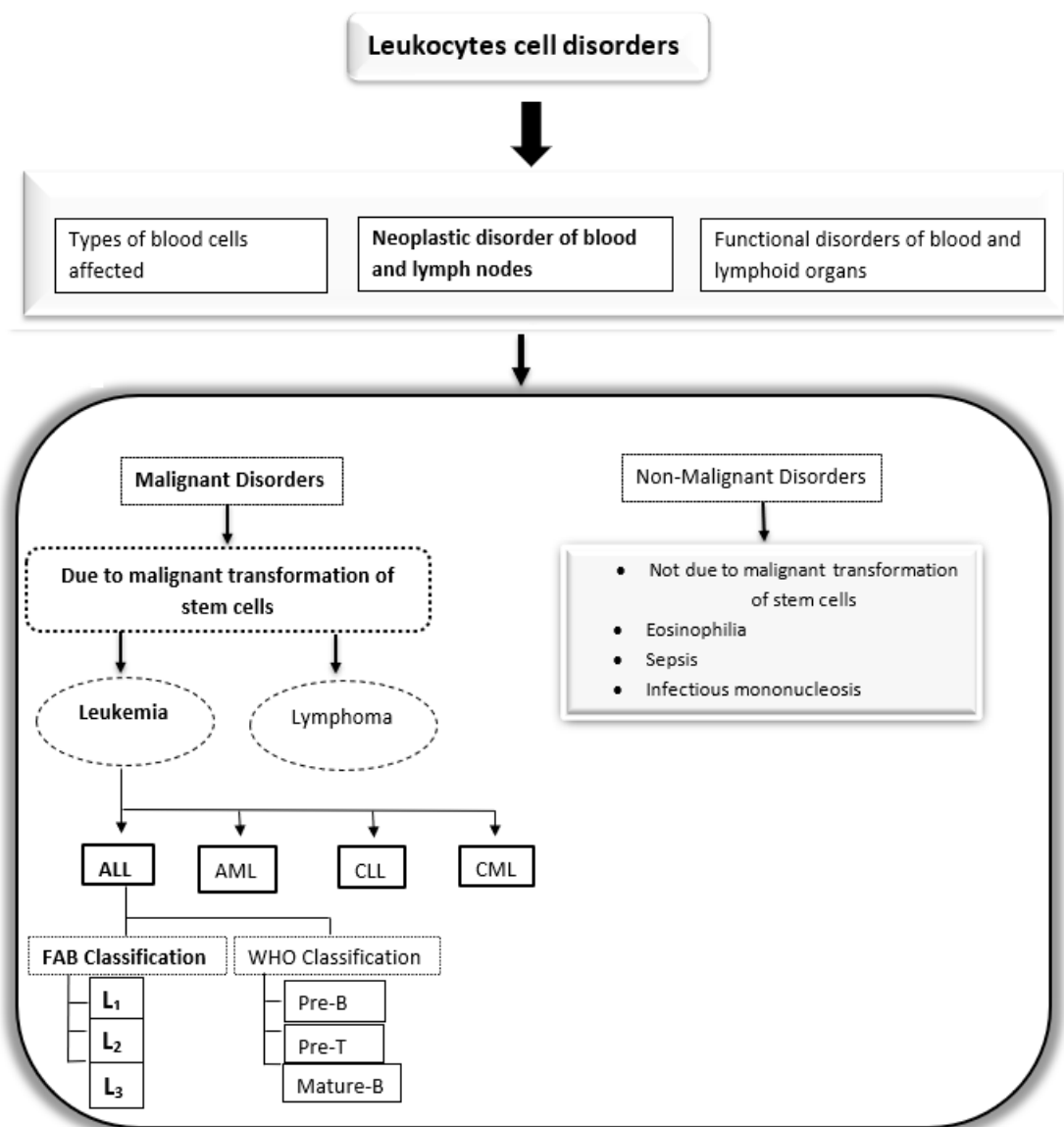


Figure 5.3: Analytical description of hematological disorders of leukocytes

Acute lymphoblastic leukemia (ALL) cannot be detected by imaging tests such as CT (scan), MRI, X-rays, or ultrasound. The appearance and count of blood cells

are used in the initial diagnostic tests for ALL. A complete blood count (CBC), for example, determines a prognosis solely based on cell count, but a peripheral blood smear examination evaluates the emergence of cells within the blood. Bone marrow-based evaluations, such as bone marrow biopsy and aspiration are critical in ALL prognosis and evaluate bone marrow to determine leukemia signs and symptoms. Cytochemistry, immunohistochemistry, and flow cytometry, which are based on the reaction of staining chemicals with the proteins of blood cells, are more accurate techniques for determining the identification of leukemia.

5.3 Dataset of C-NMC-2019

The Cancer Imaging Archive (TCIA) has made the C-NMC-2019 dataset. This C-NMC-2019 dataset was also used in the medical imaging challenge Classification of Normal vs Malignant Cells in B-ALL White Blood Cancer Microscopic Image: ISBI 2019, which was held at the IEEE international symposium on biomedical imaging (ISBI), 2019.

Acute lymphoblastic leukemia (ALL) accounts for around a quarter of all childhood malignancies. In general, distinguishing premature leukemic blasts from normal cells under the microscope is difficult since the appearances of two cells are morphologically identical.

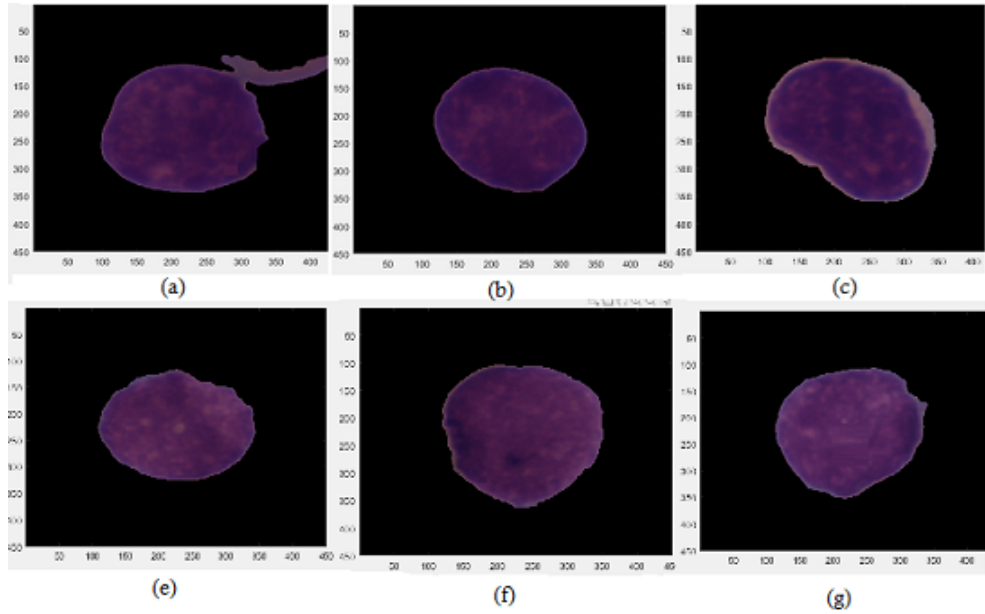


Figure 5.4: C-NMC-2019-Dataset samples, (a) to (c) are malignant, and (d) to (g) are healthy cells

The data is structured at the subject level, with a total of 118 subjects divided into the training and testing sets. In the training set, the total subjects are 73 of which ALL(Cancer) are 47 and Normal are 26. In this training set total cell images are 10,661 of which ALL (Cancer) cells are 7272, and Normal cells are 3389. In training data, there are sevenfold for cross-validation, where each fold contains the equal ratio of images of both (healthy and cancer) classes. The folds are also constructed at the subject level, with all the cells of a subject in the same fold. In the Preliminary set total subjects are 28 in which ALL(Cancer) is 13 and Normal is 15. In this Preliminary set total cell images are 1867 in which ALL (Cancer) cells are 1219, and Normal cells are 648. In the final set total subjects are 17 in which ALL(Cancer) is 9 and Normal is 8. In this set total cell images are 2586. This dataset is also briefly describe in [15].

Dataset	ALL Subjects	Normal Subjects	ALL Cells	Normal Cells	Total Cells
Training set	47	26	7272	3389	10,661
Preliminary set	13	15	1219	648	1867
Final set	9	7	-----	-----	2586

Table 5.1: Description of C-NMC-2019 Dataset

Leukemia, a category of malignancies of the blood often develop in the Bone marrow and increases the lifespan of immature white blood cells (WBCs). The excessive growth of these types of (WBCs) cells, which are referred to as blasts or leukemic cells, leads to the covering of healthy leukocytes and eliminates normal hematopoiesis which causes difficulty fighting infection, Oxygen transportation, and bleeding control system. Clinically, leukemia has been divided into chronic and acute types on the rapid development of the disease. The acute form of leukemia is rapidly developing and quickly develops the leukemic cells. The myelogenous and lymphoid types of leukemia are further separated based on which type of cell the malignant infection is developing.

Acute leukemia is a leukocyte disease and its successors. Acute lymphoblast leukemia (ALL) is the second most leading form of adult leukemia and pediatric

cancer. Genetic changes and chromosome mutation of the lymphocyte progenitor cells are causing heterogeneous malignancy in an early stage of cell differentiation. In 2015, ALL harmed around 876,000 people worldwide, resulting in over 111,000 fatalities [14]. These lymphoblasts that don't become mature B and T lymphocytes replace normal bone marrow cells and move to important organs such as the spleen, lymphatic nodes, liver and central nervous system.

ALL diagnoses require a broad set of data from many modalities including morphology, cell phenotypic, cytochemistry, cytogenetics, and molecular genetics. Despite advances in medicine, morphology is still the most used method of hematological diagnosis. Several computational approaches have been investigated to reduce human intervention and solve the constraints mentioned above. Traditional image processing and machine learning approaches, such as segmentation, feature extraction, and classification methods, are used in the bulk of these systems. The segmentation and extraction phases are the most important and difficult job [23].

The major reason for this is the vast range of blood smear pictures collected under various conditions, as well as the possible morphological variations between blast cells. Although some of these methods are faster and less expensive than manual inspection, their effect and accuracy are still insufficient [29].

In this study, we are attempting to solve the most difficult aspects of the detection procedure, with FAB classification as our primary goal. Different features are considered such as cell size, nucleoli, chromatin, nuclear shape, and degree of basophilia in the cytoplasm, and the presence of vacuoles in the cytoplasm.

Morphological Classification of ALL	
FAB Types	Features
L_1	Small uniform cells with regular nuclei and scant cytoplasm. There is a condensed chromatin and indistinct nucleoli which is not visible.
L_2	Large heterogeneous cells with irregular nuclei and mild to moderate cytoplasm. There is a clefting of nucleus and large and prominent nucleoli.
L_3	Large cells with regular nuclei with moderate to abundant vacuolated cytoplasm. There is an oval-to-round nucleus and prominent nucleoli.

Figure 5.5: FAB Classification

5.4 Image Segmentation

The extraction of morphological and textural characteristics from specific cell areas, similar to the visual interpretation of a domain expert, improves the classifier's performance. The majority of previously suggested techniques involve sequential image preprocessing, cell segmentation, extraction, and cell classification. The main goal of pre-processing phase is to increase the image quality in preparation for future processing.

Many authors have improved blood smears by converting them into another color domain that highlights the key features of the objects and thus improves region detection effectiveness. In [17] Hariprasath, converts RGB to CMYK, and in [22] Moradianin converts RGB to HSV this reduced the relationship between the color channels in comparison with RGB and allowed separate treatment of the three H, S, and V channels. The first challenging task is the extraction of the nucleus and cytoplasm from the WBCs. Different techniques are available like morphological operations, histogram equalization, threshold methods, stain normalization, and color deconvolution.

In the field of robotics, bioinformatics, and general artificial intelligence, color normalization was used for object identification on color images when it is important to remove all color value values while preserving color values. The color values in an image are distributed based on the illumination, which may vary depending on lighting conditions and other factors.

Hematoxylin and eosin stain often abbreviated as H&E stain or HE stain) is one of the main tissue stains used in histology. In medical diagnosis, this stain is most commonly used. The hematoxylin and eosin were combined by two histological stains. The cell nuclei of the hematoxylin color are purely blue, and the eosin stain the cytoplasm pink and extracellular matrix, and other structures taking on different hues, shades, and combinations of these colors.

The camera acquires the intensities I_R , I_G , and I_B for each pixel in the RGB channel. The picture intensity data cannot be used directly to isolate and quantify each channel since the relative intensity of each channel is dependent on the non-linear stain concentration. The logarithmic ratio of incoming radiation to transmitted radiation across a channel described as the optical density (OD) for each channel.

$$\text{Optical Density (OD)} = -\log\left(\frac{I}{I_0}\right).$$

where I_0 represents the original light intensity and I is the light intensity passing through the sample. The OD of each channel is proportional to the absorption of the absorbing material. Each space will be characterized by a specific optical density into the three RGB channels, which may be represented by a 3 by 1 OD vector defining the OD converted RGB color space. For example, measurements of a hematoxylin-stained sample yielded OD values of 0.18, 0.20, and 0.08 for the R, G, and B channels, respectively.

The vector length is proportional to the amount of stain, while vector relative values describe the actual OD for the detecting channels. The color system can be expressed as a formed matrix in [28] for three channels.

$$\begin{bmatrix} N_{11} & N_{12} & N_{13} \\ N_{21} & N_{22} & N_{23} \\ N_{31} & N_{32} & N_{33} \end{bmatrix}$$

For every channel Red, Green, and Blue the rows in the matrix represent the specific stain and columns represent the optical density. An ortho-normal transformation is to be done for the separation of the stain of the RGB channel to get the individual stain of each contribution. For the combination of Hematoxylin, eosin, and DAB an Optical density matrix is;

R	G	B	
0.18	0.20	0.08	Hematoxylin
0.01	0.13	0.01	Eosin
0.10	0.21	0.29	DAB

Source [28]

Deconvolution is the inverse operation of convolution. In signal processing and image processing, both operations are used. Convolution, for example, can be used to apply a filter and the original signal can be retrieved by deconvolution. Deconvolution is a computer-intensive image processing technique that is increasingly used to improve the microscope image contrast and resolution.

By applying the above method of color deconvolution to our dataset the first segmented image is given below.

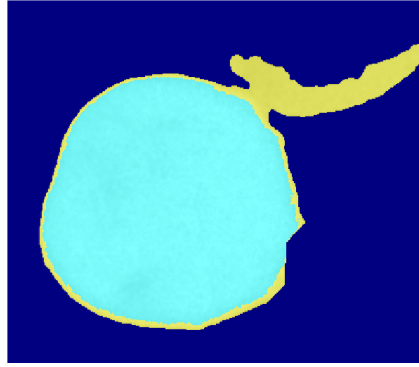


Figure 5.6: Segmented image

After that further pre-processing techniques are differentiating the nucleus and cytoplasm of a cell image. In figure 5.7 (a) is the original cell image, (b) is the segmented nucleus, and (c) is the segmented cytoplasm.

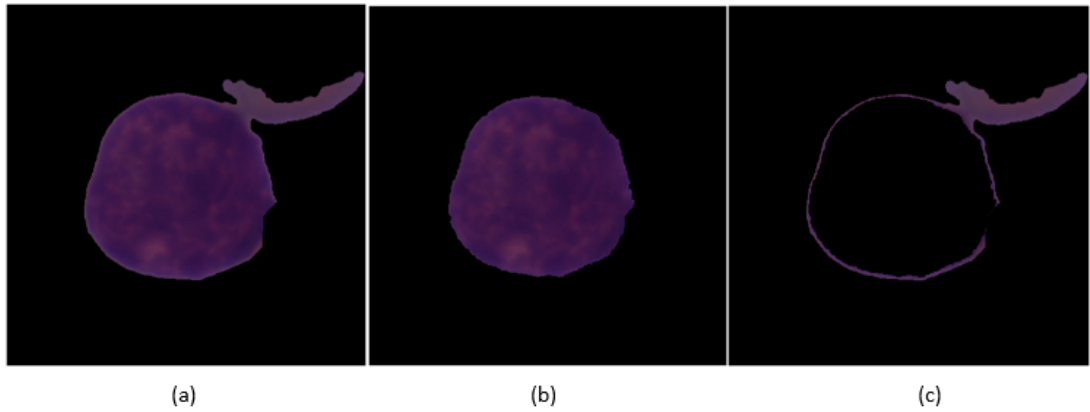


Figure 5.7: Segmented Nucleus and Cytoplasm of a Cell

When we separate the nucleus and cytoplasm of cell images we have to conclude the point cloud data from these images. As in figure 5.7 there is a segmented nucleus and cytoplasm so for these image here below in figure 5.8 a pictorial depiction of point cloud. These point cloud data will be used for further model.

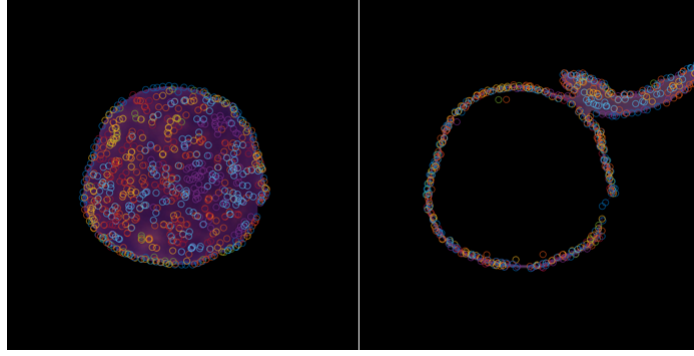
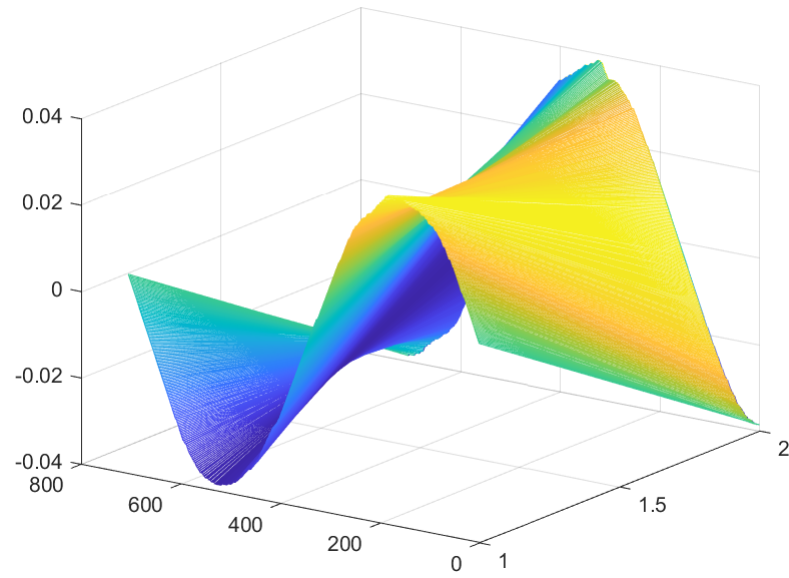


Figure 5.8: Point cloud from segmented images

In smooth Euler characteristics transform we have also trace the boundary of images. Here below is the pictorial representation of figure 4.2.



5.5 Results of smooth Euler characteristics transform

As we have already discussed the SECT in our previous chapter so in our model the purpose of SECT is to determine the clefing of nucleus. In FAB classification its given that the nucleus of L_1 and L_3 is uniform while in L_2 there is a clefing in nucleus, so SECT determines the euler curves (clefing) of the images.

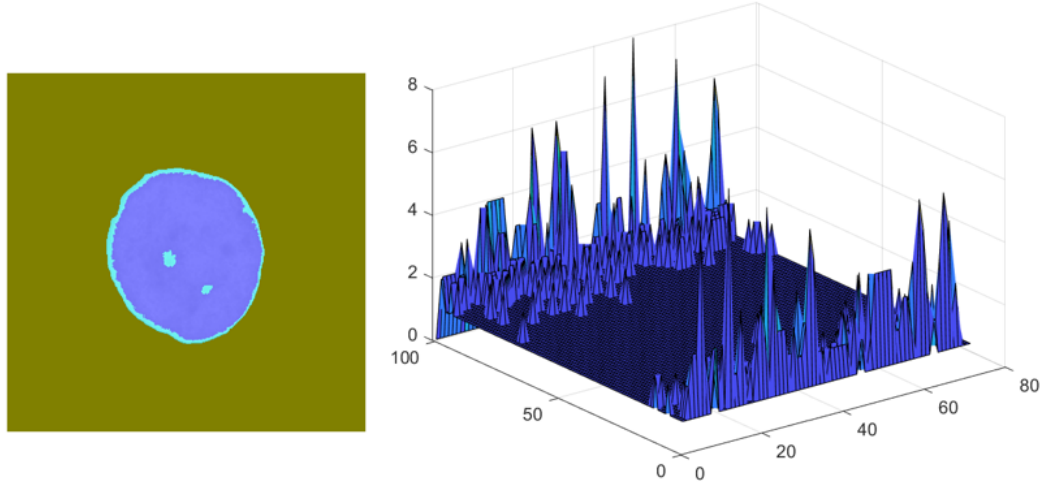


Figure 5.9: Normal cell and their Euler curves

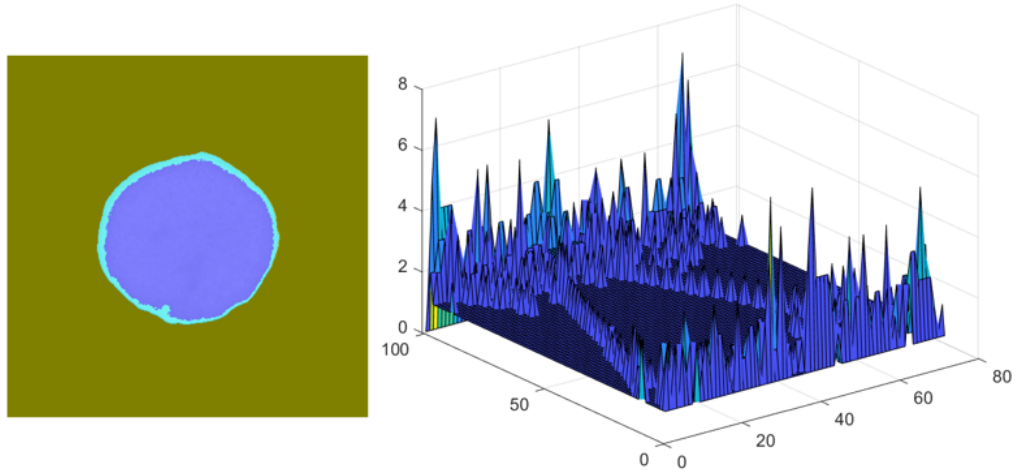


Figure 5.10: Normal cell and their Euler curves

Here below are euler curves of leukemia cell images.

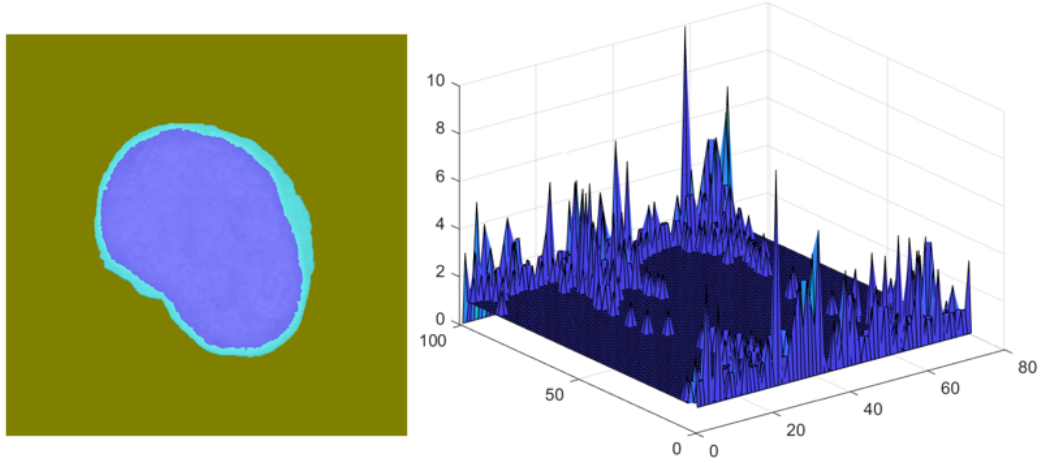


Figure 5.11: Leukemia cell and their Euler curves

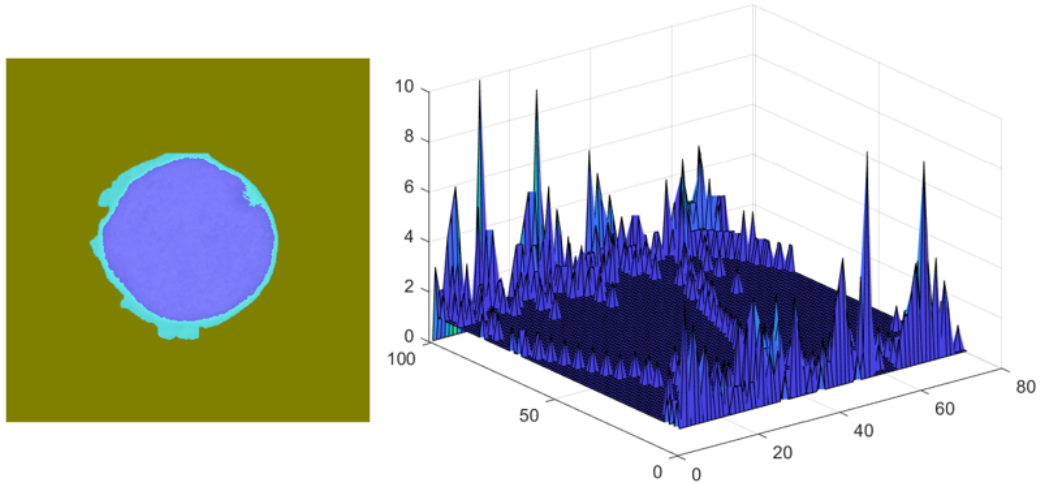


Figure 5.12: Leukemia cell and their Euler curves

5.6 Machine Learning and Topological Data Analysis

TDA and Machine learning (ML) amalgamation can retrieve effective result in Bio-medical field. In this work, we used TDA tools and its applications on classification of images using ML. ML is the scientific study of algorithms and statistical models used to train computers to do certain tasks without the need for external instructions, mostly through patterns and inference. Machine learning may be divided into two categories.

1. Supervised Machine Learning.
2. Unsupervised Machine Learning.

5.6.1 Supervised Machine Learning

Data in the training sample containing information on the available inputs and their labelled outcomes for some certain behaviour. This approach is said to supervised ML. There are numerous methods in the area of supervised machine learning, such as k-nearest neighbours, support vector machine (SVM), and so on. Here below is an example of supervised ML.

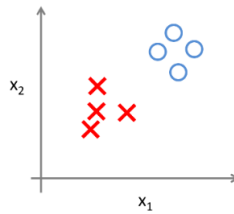


Figure 5.13: Clearly labeled data as circles and crosses

5.6.2 Unsupervised Machine Learning

Unsupervised ML does not categorized incoming data with specific labels; instead, the machine generates response based on similarities between the input data.

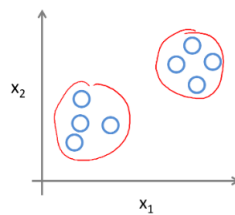


Figure 5.14: The clustering are being formed from unlabeled data

Unsupervised machine learning algorithms include clustering and anomaly detection. In our subsequent discussion, we will apply supervised machine learning algorithms, namely support vector machines. Consider the fundamental understanding of this framework.

Support Vector Machine

SVM is a supervised machine learning method that learns by dividing data into categories or labels. Using those classifications, the model then predicts the data. In a support vector machine, data is separated using hyperplanes. For example, if the data is on a 2D plane, the hyperplane that separates data sets for prediction is a line.

support Vector Machine

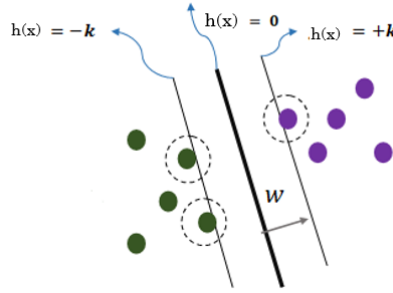


Figure 5.15: SVM separating the data of blue and red dots.

Let us now define the SVM mechanism. If $z \in \mathbb{R}^m$ is a data point, we define a hyperplane separated by hyperplane.

$$h(z) = w^t z + c.$$

Where w is a coefficient vector and c is a shift from origin. Where $w^t z = w \cdot z = w_1 z_1 + w_2 z_2 + \dots + w_n z_n$. If $p = (u_0, v_0)$ is a point and is $au + bv + c = 0$ is a plane, then distance between point and plane is calculated by

$$d(p) = \frac{|au_0 + bv_0 + c|}{\sqrt{a^2 + b^2}} \quad (5.1)$$

In generalized form for n-dimensional point $u = (u_1, u_2, \dots, u_n)$ and hyperplane we can write (5.1) as

$$d_h(x) = \frac{|w^t u + c|}{\sqrt{w_1^2 + w_2^2 + \dots + w_n^2}} = \frac{w^t u + c}{\|w\|_2}.$$

Let v_n be the labels of classes, given $h(u) = 0$ is the hyperplane in figure 5.15 with $h(u) = k$ and $h(u) = -k$ marginal shifts. Consider, $v_n = +1$ and $v_n = -1$ labeling. Hence, SVM should maximizes margin k where,

- $-w^t u + c \geq k$ for $u_n = 1$ for red circles.
- $-w^t u + c \leq k$ for $u_n = -1$ for blue circles.

Let us reformulate our minimum distance of hyperplane with point to be equals

1. So the marginal condition will be

$$v_n(w^t u + c) \geq 1 \quad \forall \quad n$$

support Vector Machine

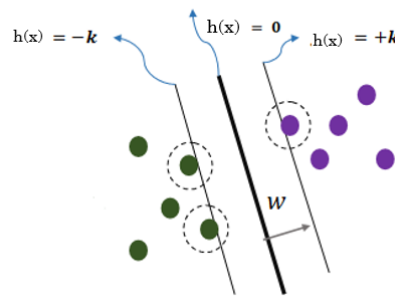


Figure 5.16: Data points, covered with dashed circles shows the support vectors that are the data points touching the boundaries of marginal planes.

The support vectors assist us in determining the margins. They can be expressed as follows:

$$v_n(w^t u + c) = 1 \quad \forall \quad n.$$

When the data is spread properly, SVM classifies the data using the hyperplane. Any computational software like MATLAB or python is used to calculate the SVM results.

5.7 Classification Evaluation

For the data set, SVM acts as a model builder. Hyperplanes are used to divide data points according to their necessary class in binary classification. Support vectors maximise separation while optimizing margins. This is where the SVM training method comes in. A new data point is identified and tagged to its class for the testing stage. Now that the classification is complete, how do we read or classify it? There are various assessment measures that can help us determine

how robust our SVM model is. Binary classification has 4 outputs, a true/false positive(T/F) +ive and a true/false negative(T/F) -ive.

- T +ive: when the prediction and actual value is positive.
- T -ive: when the prediction and actual value is negative.
- F +ive: when the prediction is positive but actual value is negative.
- F -ive: when the prediction is negative but actual value is positive.

For the classification case, both the false predictions are actually wrong. In fact, the true pre-dictions are the labels that we require.

5.7.1 Evaluation Metrics

Let's define some of the widely used evaluation metrics.

Definition 5.7.1 (Accuracy). It is the ratio between the accurate predictions and all the others.

$$\text{Accuracy} = \frac{(T + ive) + (T - ive)}{(T + ive) + (T - ive) + (F + ive) + (F - ive)}.$$

Definition 5.7.2 (Precision). It is the ratio between true positive predictions and all other positive predictions.

$$\text{Precision} = \frac{(T + ive)}{(T + ive) + (F + ive)}.$$

Definition 5.7.3 (Recall). It is a ratio of true positive predictions to all actual positive values.

$$\text{Recall} = \frac{(T + ive)}{(T + ive) + (F - ive)}.$$

Definition 5.7.4 (Specificity). It is a ratio to true negative predictions to all actual negative values.

$$\text{Specificity} = \frac{(T - ive)}{(F + ive) + (T - ive)}.$$

Definition 5.7.5 (F1 score). F1 scores combines the precision and recall. It is defined as the harmonic mean of the precision and recall.

$$\text{F1 score} = \frac{2 \times (\text{Recall} \times \text{Precision})}{(\text{Recall} + \text{Precision})}.$$

The best metric for accuracy is if there were a comparable cost (F -ive) and a similar cost (F +ive), but F1 exceeds precise costing. Precision provides information on (T+ive) existence and makes sure you don't overlook positive information. The specificity simply clears the wrong negative predictions. Precision, Recall, and Specificity target a certain section, such as positive or negative portions, whereas accuracy and F1 score gives the over all result. The algorithmic steps for making a model are given below,

Algorithm
Step 1: INPUT: Leukemia cell Images Step 2: Image processing Step 3: Extracting of features Step 4: Construction of point cloud Step 5: Compute persistent homology by Vietoris-rips Step 6: Construction of space of Persistent diagrams Step 7: Feature collection using Riemannian framework and PGA Step 8: OUTPUT: Construction of classifier using SVM classification

5.8 Results for Cancer Classifier

We have used only 200 microscopic images to train our model there are also some computational limits due to which our $F1$ score is not productive. In the future, we are trying to train our model for the whole dataset for more effective and accurate results. We gave the F1 scores of our classifier as given in the following table,

Results	
Method	F1 Score
Our Model for Cancer	78%

Table 5.2: Results for Cancer

5.9 Results for Leukemia Cells

The following table shows the comparative results with previous methods, and our method results are as following.

Results	
Method	F1 Score
Our Model for Leukemia cells	78%
Pan	91.0%
Xia	84.8%
Ding	85.5%
Gehlot	90.4%

Table 5.3: Results for Leukemia cells

5.10 Conclusion and Future Work

In this work, we have studied the TDA techniques and their current implementations in the different fields of research. In particular, we have seen the TDA tools, Persistent Homology as a discriminatory technique among the images of Normal and ALL (Acute Lymphoblastic Leukemia) cells. Moreover, we have used Machine Learning tools for classification. This effort will lead to significant progress in the field of medical imaging. We have applied the image analysis techniques on microscopic images for their classification as pathologists can do. In the future, we are plan to build a model to detect the effective topological descriptor for the best and accurate FAB classifications that depend upon geometrical features. Our aim is this work will be able to be carried out in different medical imaging issues and better results can be achieved by utilizing the different metrics on the space of Persistent diagrams.

Bibliography

- [1] Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18, 2017.
- [2] Rushil Anirudh, Vinay Venkataraman, Karthikeyan Natesan Ramamurthy, and Pavan Turaga. A riemannian framework for statistical analysis of topological persistence diagrams. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 68–76, 2016.
- [3] Ulrich Bauer. Ripser: efficient computation of vietoris-rips persistence barcodes. *arXiv preprint arXiv:1908.02518*, 2019.
- [4] Ulrich Bauer, Michael Kerber, Jan Reininghaus, and Hubert Wagner. Phat—persistent homology algorithms toolbox. *Journal of symbolic computation*, 78:76–90, 2017.
- [5] Paul Bendich, James S Marron, Ezra Miller, Alex Pieloch, and Sean Skwerer. Persistent homology analysis of brain artery trees. *The annals of applied statistics*, 10(1):198, 2016.
- [6] Alexandra Bodzas, Pavel Kodytek, and Jan Zidek. Automated detection of acute lymphoblastic leukemia from microscopic images based on human visual perception. *Frontiers in Bioengineering and Biotechnology*, 8:1005, 2020.
- [7] Peter Bubenik. Statistical topological data analysis using persistence landscapes. *J. Mach. Learn. Res.*, 16(1):77–102, 2015.
- [8] Peter Bubenik. The persistence landscape and some of its properties. In *Topological Data Analysis*, pages 97–117. Springer, 2020.
- [9] Peter Bubenik and Paweł Dłotko. A persistence landscapes toolbox for topological statistics. *Journal of Symbolic Computation*, 78:91–114, 2017.

- [10] Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.
- [11] Mathieu Carriere, Marco Cuturi, and Steve Oudot. Sliced wasserstein kernel for persistence diagrams. In *International Conference on Machine Learning*, pages 664–673. PMLR, 2017.
- [12] Sofya Chepushtanova, Tegan Emerson, Eric Hanson, Michael Kirby, Francis Motta, Rachel Neville, Chris Peterson, Patrick Shipman, and Lori Ziegelmeier. Persistence images: an alternative persistent homology representation. *arXiv preprint arXiv:1507.06217*, 7, 2015.
- [13] Lorin Crawford, Anthea Monod, Andrew X Chen, Sayan Mukherjee, and Raúl Rabadán. Functional data analysis using a topological summary statistic: the smooth euler characteristic transform. *arXiv preprint arXiv:1611.06818*, 2016.
- [14] V Feigin et al. Global, regional, and national incidence, prevalence, and years lived with disability for 310 acute and chronic diseases and injuries, 1990-2015: a systematic analysis for the global burden of disease study 2015. *The Lancet*, 388(10053):1545–1602, 2016.
- [15] Shiv Gehlot, Anubha Gupta, and Ritu Gupta. Sdct-auxnet θ : Dct augmented stain deconvolutional cnn with auxiliary classifier for cancer diagnosis. *Medical image analysis*, 61:101661, 2020.
- [16] Marco Guerra, Alessandro De Gregorio, Ulderico Fugacci, Giovanni Petri, and Francesco Vaccarino. Homological scaffold via minimal homology bases. *Scientific Reports*, 11(1):1–17, 2021.
- [17] S Hariprasath, T Dharani, and M Santh. Detection of acute lymphocytic leukemia using statistical features. In *4th International Conference on Current Research in Engineering Science and Technology*, 2019.
- [18] Sohail Iqbal, H Fareed Ahmed, Talha Qaiser, Muhammad Imran Qureshi, and Nasir Rajpoot. Classification of covid-19 via homology of ct-scan. *arXiv preprint arXiv:2102.10593*, 2021.

- [19] PS Kumar and S Vasuki. Automated diagnosis of acute lymphocytic leukemia and acute myeloid leukemia using multi-sv. *Journal of Biomedical Imaging and Bioengineering*, 1(1):1–5, 2017.
- [20] Zhenyu Meng, D Vijay Anand, Yunpeng Lu, Jie Wu, and Kelin Xia. Weighted persistent homology for biomolecular data analysis. *Scientific reports*, 10(1):1–15, 2020.
- [21] Yuriy Mileyko, Sayan Mukherjee, and John Harer. Probability measures on the space of persistence diagrams. *Inverse Problems*, 27(12):124007, 2011.
- [22] Morteza MoradiAmin, Nasser Samadzadehaghdam, Saeed Kermani, and Ardeshtir Talebi. Enhanced recognition of acute lymphoblastic leukemia cells in microscopic images based on feature reduction using principle component analysis. *Frontiers in Biomedical Technologies*, 2(3):128–136, 2015.
- [23] Siew Chin Neoh, Worawut Srisukkham, Li Zhang, Stephen Todryk, Brigit Greystoke, Chee Peng Lim, Mohammed Alamgir Hossain, and Nauman Aslam. An intelligent decision support system for leukaemia diagnosis using microscopic blood images. *Scientific reports*, 5(1):1–14, 2015.
- [24] B. D. O’Gwynn. A topological approach to shape analysis and alignment. 2011.
- [25] Nina Otter, Mason A Porter, Ulrike Tillmann, Peter Grindrod, and Heather A Harrington. A roadmap for the computation of persistent homology. *EPJ Data Science*, 6:1–38, 2017.
- [26] Talha Qaiser, Korsuk Sirinukunwattana, Kazuaki Nakane, Yee-Wah Tsang, David Epstein, and Nasir M Rajpoot. Persistent homology for fast tumor segmentation in whole slide histology images. *Procedia Computer Science*, 90:119–124, 2016.
- [27] Joseph J Rotman. *An introduction to algebraic topology*, volume 119. Springer Science & Business Media, 2013.

- [28] Arnout C Ruifrok, Dennis A Johnston, et al. Quantification of histochemical staining by color deconvolution. *Analytical and quantitative cytology and histology*, 23(4):291–299, 2001.
- [29] Sarmad Shafique and Samabia Tehsin. Acute lymphoblastic leukemia detection and classification of its subtypes using pretrained deep convolutional neural networks. *Technology in cancer research & treatment*, 17:1533033818802789, 2018.
- [30] Abdel Aziz Taha and Allan Hanbury. An efficient algorithm for calculating the exact hausdorff distance. *IEEE transactions on pattern analysis and machine intelligence*, 37(11):2153–2163, 2015.
- [31] Gerard J Tortora and Bryan H Derrickson. *Principles of anatomy and physiology*. John Wiley & Sons, 2018.
- [32] Christopher Tralie, Nathaniel Saul, and Rann Bar-On. Ripser. py: A lean persistent homology library for python. *Journal of Open Source Software*, 3(29):925, 2018.
- [33] Katharine Turner, Sayan Mukherjee, and Doug M Boyer. Persistent homology transform for modeling shapes and surfaces. *Information and Inference: A Journal of the IMA*, 3(4):310–344, 2014.
- [34] David R Wilkins. Algebraic topology, 1988.
- [35] Kelin Xia and Guo-Wei Wei. Multidimensional persistence in biomolecular data. *Journal of computational chemistry*, 36(20):1502–1520, 2015.
- [36] Simon Zhang, Mengbai Xiao, and Hao Wang. Gpu-accelerated computation of vietoris-rips persistence barcodes. *arXiv preprint arXiv:2003.07989*, 2020.
- [37] Bao Zhou. Image segmentation using slic superpixels and affinity propagation clustering. *Int. J. of Science and Research*, 4(4):1525–1529, 2015.

Appendix A

Homologies Computations

A.1 Simplicial

Persistent homology provides an interval collection for each homology degree. We can use the simplicial module in Python to determine both simplicial and PHs. To install this library, you can use the install pip method in our console by typing “pip install simplicial”. Basic commands that used in Simplicial are,

```
from simplicial import *
C = SimplicialComplex()
SimplicialComplex.numberOfSimplices()
SimplicialComplex.bettiNumbers(ks=None).
```

A.2 Distances Computation

A.2.1 Hausdorff Distance

To calculate the Hausdorff distance algorithmically codes are available in Matlab and also for python. An efficient algorithm for calculating the Exact Hausdorff distance in [30] by using the simple command in console;

```
pip install hausdorff.
```

A.2.2 Bottleneck distance

To calculate the bottleneck distance of Persistent diagrams there are different packages here we explain its computational commands in Matlab and Python;

1. A Matlab wrapper for the persistent landscape toolkit that was developed by Pawel Dlotko, in which they also compute the bottleneck distance by following,

```
bnd = landscapeBottleneckDistance(dgm_1, dgm_2).
```

2. To compute the bottleneck distance in Python we are using the library ”Persim”, and also we can visualize it for that we are using the following thing;

```
distance_bottleneck, matching = persim.bottleneck(dgm_1,
dgm_2,matching=True).
```

For Visualization of Bottleneck distance,

```
persim.bottleneck_matching(dgm_1, dgm_2, matching,  
labels=['Clean $H_1$', 'Noisy $H_1$']).
```

A.2.3 Sliced Wasserstein distance

Sliced Wasserstein Kernels for persistence diagrams were introduced in [11]. The general idea is to compute an approximation of the Wasserstein distance by computing the distance in 1-dimension repeatedly and use the results as a measure. Here is the command for computing the Sliced Wasserstein distance in Persim, Python library.

```
persim.sliced_wasserstein(dgm_1, dgm_2).
```

A.2.4 Wasserstein distance

In the environment of Matlab for the computation of Wasserstein distance this is the following command;

```
d = landscapeDistance(dgm_1, dgm_2, n).
```

A.2.5 Heat Kernel Distance

This is another very approach towards the PDs distance here is the command;

```
persim.heat(dgm_1, dgm_2).
```

A.3 Persistent Landscape

Peter Bubenik in [7] proposed a stable and invertible space called persistent landscape. The mathematical background of the persistent landscape is vector-space. It easily combines the tools of machine learning and statistics. In [7] shows that due to its stability it can be used to provide a lower bound for the Bottleneck and Wasserstein distances. They also purposed a toolbox called persistent landscape toolbox [9].

As well as in Python environment libraries are available for this technique. Persim a python module is openly available and the command for persistent landscape;

```
persim.landscapes.plot_landscape_simple(persim.PersLandscapeExact  
(dgms_torus, hom_deg=1),title="---", ax=axis[0]).
```


A.4 Persistence images

Persistent homology information is represented by two standard ways, PD and barcodes. In [12] they propose a representation of a PD to this fundamental problem called Persistence image. PI is obtained by integrating the stability surface over each grid square, which gives us a matrix of pixel values. Codes for the Persistence images are available in both environments Matlab and Python;

For Matlab the command is;

```
[ PIs ] = make_PIs(interval_data, res, sig, weight_func,params,type).
```

For Python the command is;

```
pimgr = PersistenceImager(pixel_size=0.2, birth_range=(0,1)).
```

A.5 Riemannian framework

Riemannian framework is an entirely new framework in which PDs are approximated as 2D (probability) density function. Matlab code for 2.3 is given below,

```
Sigma, x1 and x2 are variables,  
F = mvnpdf([X1(:) X2(:)],mu,Sigma).
```

A.6 Persistent Homology Transform

In [33] Katharine Turner introduce a code for calculating PHT. There are many different packages in many different languages for computing persistence diagrams from filtrations of simplicial complexes. She uses Dmitry Morozov's code which is useable in python. The source of the code is available at,

```
http://sma.epfl.ch/~kturner/#code.
```

A.7 Smooth Euler Characteristics Transform

SECT summarizes the shape information in the form of smooth curves. This technique also uses the implementation of Functional data analysis. The mathematical background of SECT is persistent homology transform. In [13] they show that SECT permits us to map the shapes into a space that is represented by the

collection of curves and a well define inner product structure. For the segmentation of medical images [13] they used a C++ environment toolbox that is MITKATS and is located as

`https://github.com/RabadanLab/MITKats`.

For the implementation of SECT, the code is available in both environments of R and Matlab openly available at this repository,

`https://github.com/lorinanthony/SECT`.

Appendix B

TDA-Packages

B.1 Javaplex

Javaplex code in the environment of Matlab;

```
"pc=pointcloud;" % point cloud as an input

"max_dimension = 3;"      % max dimension
"max_filtration_value = 1;" % maximum threshold
"num_divisions = 150;"    % number of divisions

create a Vietoris-Rips stream
stream = api.Plex4.createV-RipsStream(pc,max_dim,max_fil_val,num_div);

num_simplices = stream.getSize()

get persistence algorithm over  $\mathbb{Z}/2\mathbb{Z}$ 
persistence = api.Plex4.getModularSimplicialAlgorithm(max_dim, 2);

compute the intervals
intervals = persistence.computeIntervals(stream).
```

B.2 Ripser

Ripser implementation in Python;

```
n_samples=100
dgms = ripser(data)['dgms']
plot_diagrams(dgms, lifetime=True).
```

B.3 Dinoysus

```
pip install --verbose dinoysus
homology_persistence(...).
```