Fall 2015

Clustering Web Concepts Using Algebraic Topology

Harleen Kaur Ahuja
San Jose State University

Clustering Web Concepts Using Algebraic Topology

A Writing Project
Presented to
The Faculty of the Department of Computer Science
San José State University

In Partial Fulfillment
Of the Requirements for the Degree
Master of Science

By
Harleen Kaur Ahuja
December 2015

The Designated Committee Approves the Writing Project Titled

CLUSTERING WEB CONCEPTS USING ALGEBRAIC TOPOLOGY

By

Harleen Kaur Ahuja

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE
SAN JOSÉ STATE UNIVERSITY
December 2015

Dr. Tsau Young Lin	Department of Computer Science
Dr. Robert Chun	Department of Computer Science
Mr. Anupam Ahuja	Juniper Networks

ABSTRACT

In this world of Internet, there is a rapid amount of growth in data both in terms of size and dimension. It consists of web pages that represents human thoughts. These thoughts involves concepts and associations which we can capture. Using mathematics, we can perform meaningful clustering of these pages. This project aims at providing a new problem solving paradigm known as algebraic topology in data science.

Professor Vasant Dhar, Editor-In-Chief of Big Data (Professor at NYU) define data science as a generalizable extraction of knowledge from data. The core concept of semantic based search engine project developed by my team is to extract a high frequency finite sequence of keywords by association mining. Each frequent finite keywords sequences represent a human concept in a document set. The collective view of such a collection concepts represent a piece of human knowledge. So this MS project is a data science project.

By regarding each keyword as an abstract vertex, a finite sequence of keywords becomes a simplex, and the collection becomes a simplicial complexes. Based on this geometric view, new type of clustering can be performed here. If two concepts are connected by n-simplex, we say that these two simplex are connected. Those connected components will be captured by Homology Theory of Simplicial Complexes. The input data for this project are ten thousand files about data mining which are downloaded from IEEE explore library.

The search engine nowadays deals with large amount of high dimensional data. Applying mathematical concepts and measuring the connectivity for ten thousand files will be a real challenge. Since, using algebraic topology is a complete new approach. Therefore, extensive testing has to be performed to verify the results for homology groups obtained.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisor Dr. TY Lin for his aspiring guidance and giving a direction to my project. His comments and suggestions has helped me to come up with new ideas and implementing them throughout my project work. Without my advisor’s support, it would not be possible to complete this project successfully.

I am sincerely grateful to my committee members Dr. Robert Chun and Mr. Anupam Ahuja for accepting my project and providing extensive support and encouragement throughout my project work. My special thanks to my family and my friends who always stand beside me and motivate me in every step of my life.

TABLE OF CONTENT

1. Introduction 1

2. Related Work 2

3. Important Concepts and Algorithms 4

3.1 Granular Computing 4

3.2 Apriori Basics 5

3.3 Keywords and TFIDF 5

3.4 Isomorphic Associations 6

4. Algebraic Topology 7

4.1 Homeomorphism in Topology 9

4.2 The Topology of Polyhedra 12

4.3 Simplicial Complex 13

4.4 Orientation 14

4.5 Chain Complexes and Boundary Operator 16

4.6 Simplicial Homology 17

4.7 Computing Homology Groups 18

5. Homology of Tetrahedron; Manual Calculation 20

6. Project Flow: Scope of Project 24

7. Implementation: Algorithm and Approach 25

7.1 Rank Nullity Theorem 27

7.2 Gaussian Elimination 27

7.3 Smith Normal Form 27

7.4 Pseudo Code 28

8. Evaluation of Results 29

8.1 Homology Group of Tetrahedron 29

8.2 Homology Group of Mobius strip 31

9. Web Clustering 36

10. Future Scope and Conclusion 38

11. References 39

LIST OF FIGURES

1. Department store data vector representation 5

2. Tuple representing Tetrahedron 5

3. Closed-3 Simplex, Solid Tetrahedron 6

4. Tuples representing three tetrahedrons 6

5. Geometrical Simplicial Complexes of relational tuples 7

6. Topological space explaining homeomorphism 7

7. Spaces homeomorphic to Circle 8

8. One Dimensional Simplex 10

9. Holes in topological space as an important topological invariant 11

10. Example of Simplicial Complex 12

11. Two Dimensional Simplex 10

12. Three Dimensional Simplex 11

13. Simplicial Complexes 12

14. Frequent Token sets obtained from sematic search engine 13

15. Maximal Simplicial Complexes 13

16. Orientation of one dimensional spaces 14

17. Orientation of two dimensional space 15

18. Boundary Operator 17

19. Chain Complex 18

20. Topological Space: Tetrahedron 20

21. Chain Complex and Boundary Operator 21

22. Project Architecture 24

23. Tree Structure for storing chain complexes 25

24. Results for homology of Tetrahedron 30

25. Results for homology of Mobius strip 31

26. Program Screenshot 32

27. Web clustering on a simple simplicial complex 34

1. INTRODUCTION

Internet today has completely changed our lives. There is a continuous growth both in the amount of data as well as the dimension of data that is becoming a part of web each second. Considering this, it is very essential for a search engine to implement new ideas instead of simply fetching the pages relevant to the keywords given by a user. To get an instance of massive usage of web by humans, From Google Search Statistics, every second 51,534 queries are made, and per day the number is about 6,000,000,000. Google search engine uses Page rank algorithm to compute relevance of a webpage. But it does not take concepts of user and often returns inconsistent and disorganized search results. So in order to make our search more organized, we firstly need to understand the user concepts rather than concentrating on individual keyword.

As per Dr. TY Lin paper on "Knowledge Based Search Engine", Web pages represent human thoughts and these human thoughts contains concepts and associations which can be captured. From our semantic search engine, developed by our team under the guidance of Dr. TY Lin, we obtain the high frequency occurring keywords. These keywords are representing the concepts in a given document set. Similarly, all these concepts finally develops simplicial complex of concepts which are taken as knowledge base of this new semantic based search engine.

This project provides a very new approach to document clustering which is known as algebraic topology. If we have a set of documents, we can capture associations among documents which develop a simplicial complex. On these concepts, we can do clustering of the given documents into different categories. An efficient search engine is one which can discriminate from a millions of available web pages that whether a particular document is relevant to the search in a very short time. So, the main goal here would be to improve this current state of a search engine and implement this approach of association in clustering of documents.

For an instant, if we have two tokens – computer and science, then the association of these two tokens means far more than two different words. This can be represented as a line segment (x, y) such that it represent a one dimensional geometry in which the two zero dimensional points – computer and science does not have much contribution. Thus our clustering will help search engine further to categorize the results. In this case, our categories can be Computer science by

Major, Computer science by universities, Computer science careers and so on. Our project includes concepts of (1) Granular Computing (2) Algebraic Topology (3) Homology (3) Simplicial Complexes (4) Web Clustering.

3. IMPORTANT CONCEPTS AND ALGORTIHMS

This project report provides a novel approach to web clustering. Here we have to capture the "connected components" (will be defined) within the documents. In order to capture these connections, we will be using Homology theory of simplicial complexes. This theorem uses the basics of Linear Algebra. Before finding these connections, we need the concepts crawled from the web documents. Professor TY Lin and team has developed a semantic search engine that crawls the web and provides a knowledge base. This search engine is based on the concept which is mentioned in [21] by Dr. TY Lin. In this paper, the search engine provides the search results by concentrating on the user's concept (simplex of keywords) instead of individual keywords. So, the user can easily obtained the main concepts present within the documents.

This knowledge base acts as an input to this project and using homology theory of Simplicial complexes, we will find the connected components present in this data. Before moving on to theory of simplicial complexes and homology, we will be explaining briefly about the concepts that have been used to develop our knowledge base.

3.2 **Apriori Basics**

The Apriori algorithm is a very powerful data mining algorithm which is then used for creating association rules.

The key concept of Apriori algorithm includes-

a) The items which are repeated greater than a threshold value are considered as frequent items.
b) The subset of frequent itemset will naturally be a frequent item. Once the frequent items are obtained, they are then used to generate the association rules.

In association, two measurements are used which are support and confidence. The support of keyword is the frequency of documents in which that keyword is existing.
The use of Apriori algorithm is that it helps in excluding the concepts which are not that important and relevant. For an instance, Let say we have {SJSU Computer Science} as a frequent keyword set then each of the individual subsets of this set should also be a frequent set. If say {Computer} is not a frequent item then any set that contains this Computer item cannot be included.

3.3 **Keywords and TFIDF**

TFIDF is calculated that will be considered as the weight of each document. Now, here a token will be considered in the keyword set if its TFIDF value is large.

TFIDF is calculated by – tf * idf where
tf: The term tf indicates the term frequency which calculates the frequency of a token in any document. This frequency can be calculated by dividing the number of times a token appears in a document with number of words in that document.

idf: The term idf indicates inverse document frequency which is measured by taking natural log of total number of documents to the number of documents in which a particular token is appearing.

This calculation can be explained by taking a simple example. Let suppose, we have 2 million documents.
The token that we are considering = computer
Total number of documents in which computer token is appearing = 10000
Number of times the word computer is appearing = 100

Term Frequency of computer = (100) / (10000) = 0.01

Inverse Document frequency = ln ((2000000) / (10000)) = 5.29

Therefore, TFIDF for computer token = tf * idf
= 0.01 * 5.29
= 0.0529

3.4 Isomorphic Associations

The most important task in data mining is to obtain frequent patterns in large databases. In large datasets, these frequent patterns can provide a reflection of hidden relationships and connectivity among them. Association rule is a term which is used in data mining. It has two measurements factors: support and confidence. Support determines how often a rule is applicable to a given dataset, while confidence determines how frequently items in Y appear in transactions that contain X.

Formally these are defined as below:

$$\text{Support, } s(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{N}$$

$$\text{Confidence, } c(X \rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$$

Here, we have performed data mining on Relational tables. A relational database usually comprises of rows and columns. Here we will consider every tuple as a text. Without considering the different attributes, if there are tokens that occurring frequently in different tuples regardless of their location and position, they will be counted towards the frequency of occurrence of that token. These isomorphic relational tables have isomorphic patterns. It is very important in data mining to extract the correct features and patterns from these relational tables. Here, using isomorphic theorems, all hidden features of any table can be obtained.

In general, if there exists a one to one onto map between two attributes, then such a map is termed as isomorphism. In order to explain this isomorphic associations, for an instance, consider a relational table from a department store as shown below:

Diaper	Beer	Milk
1	1	0
1	0	0
1	1	0
1	1	0

Figure-1- Department store data vector representation

Here, column names are the different item names. This table represent that if a customer has purchased an item, it will be marked with 1 otherwise 0 in the column of that particular item. Now, instead of these names, we can represent these in the form of unit vectors.

First item as (1, 1, 1, 1)
Second item as (1, 0, 1, 1)

We are actually transforming a relational table to geometric simplicial complexes (Refer section 4). Here, consider only two items diaper and beer. In our set of documents, if diaper and beer are frequently occurring individually, then we say we are having two concepts and these concepts will be represented by two points. Further, if instead we have a frequent occurrence of the combination of these two items (Diaper and beer together), then we have one single concept represented by one edge.

Transforming a relational data into geometrical simplicial complex can be demonstrated by following examples-

A	B	C	D	…
1	1	1	1	0
…	…	…	…	…

Figure-2- Tuple representing Tetrahedron

The information in the first tuple can be summarized by a (Closed-3 Simplex) tetrahedron as shown below-

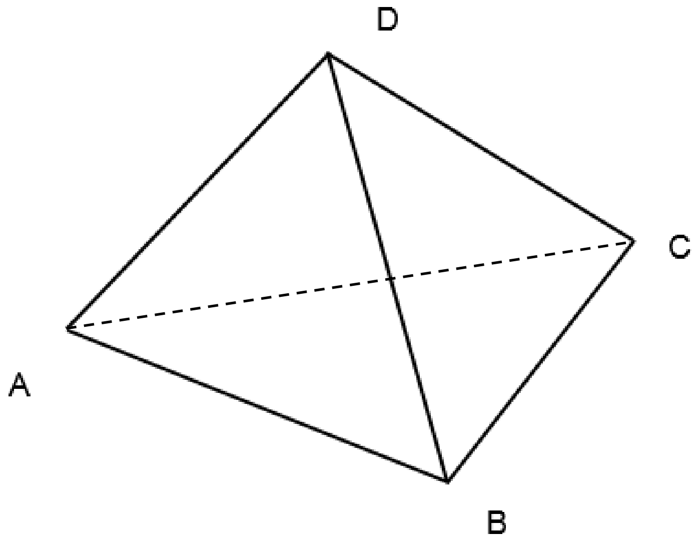


Figure- 3- Closed-3 Simplex, Solid Tetrahedron

The algorithms that will be applied for data mining will depend upon these figures not on the relational tables. Here, we will have somewhat complex relational table given below:

A	B	C	D	E	F	G
1	1	1	1	0	0	0
0	1	1	0	1	0	0
0	1	0	1	1	0	0
0	0	1	1	1	0	0
1	0	1	0	0	1	1

Figure-4- Tuples representing two solid tetrahedrons and one hollow tetrahedron

This can be converted into geometrical simplicial complexes. This will produce three tetrahedrons. The first and last tuple of this relational table are transformed into two solid tetrahedrons and the other indicates the hollow figure. The geometric figure obtained is as:

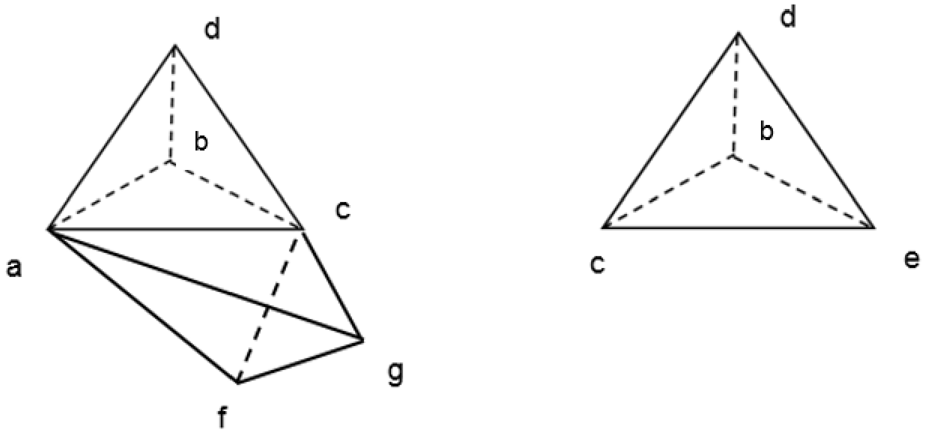


Figure-5- Geometrical Simplicial Complexes of relational tuples

In the above figure, the tetrahedrons {a b c d} and {c b d e} are glued together with the face {b c d}. The tetrahedron {a b c d} and {a f g c} are glued together by an edge {a c}. The first tuple is for solid tetrahedron {a b c d}. The second tuple summarize for a triangle face {b c e} and so on for other next two tuples. The last tuple in the relational table again summarize a solid tetrahedron {a c f g}. So, here each tuple is transformed into concepts which are shown as geometric simplicial complexes.

Summarizing the concepts that are defined above-

1. A relational table is a bag relation where repeated tuples are allowed.
2. An item is an attribute value.
3. A q-sub tuple is an association pattern, if its occurrences are greater than or equal to a given threshold.

4. ALGEBRAIC TOPOLOGY

Topological Data Analysis is a subarea of computational topology that develops topology algorithms to analyze high dimensional scientific data. It actually provides a shape to our data. It is a field of mathematics that deals with the study of shape. The term topology measures the properties or shapes that don’t change even if we change the shape or change the coordinate system in which the object is viewed. It classifies the shape according to its connectivity such as number of loops, connections and presence of a boundary. The properties and attributes in topology are independent of continuous deformation.

Here, in order to explain the continuous deformation term, let us say we have a line in Euclidian space which is growing continuously in both the directions as shown below-

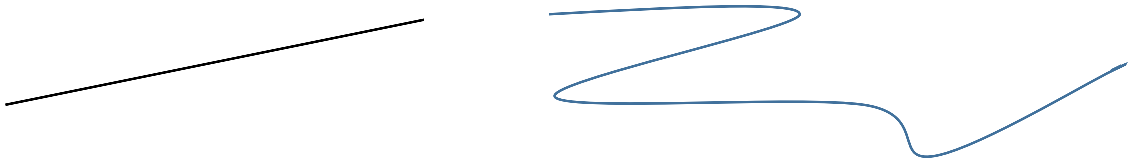


Figure-6- Topological space explaining homeomorphism

Now again let say we can mold this line, stretch it, extend it to look like a second figure. If we say geometrically, these two figures are completely different. But topologically, these are same. In the language of algebraic topology we say that these two are homeomorphic objects.

Topological Space

A topological space by definition is a set U, in which certain subsets are distinguished called as open subsets. These open sets has to follow a set of axioms and conditions. These conditions as mentioned in [4] can be stated as-

"Any union of open sets is open"

"Any intersection of open sets is open"

"The whole space and the empty set are open"

Here consider the set U = {a, b, c, d} to be open, which implies a set which has no boundary or limits. To explain the algebraic topology, take the following set of examples:

{ ϕ , {a, b}, {c, d}, U}

The above set **can** be considered as a topology since it is satisfying the above two properties of union and intersection.

{ ϕ , {a, b, c}, {c, d}, U}

But here, this **cannot** be a topology since the intersection property is not valid here. As, the intersection of {a, b, c}, {c, d} is {c} which is not one of the subset of this.

After adding the subset {c} as below this will now become a topology set:

{ ϕ , {a, b, c}, {c, d}, {c}, U}

A closed subset of the topological space U is the complement of an open set. For the closed sets, following axioms must be satisfied-

"Union of finite number of closed sets is closed"

"Intersection of closed sets is closed"

"The whole space and the empty set are closed"

If U_0 is a subset of the topological space U, the induced topology in U_0 is that in which the open sets are the intersections with U_0 of the open sets of U. Subsets always follow the induced topology.

4.1 **Homeomorphism in Topology**

Two Topological spaces X and Y are homeomorphic, X ~ Y, ⇔ that there is a continuous map δ: X → Y, which has a continuous inverse.

It implies that this map δ is a bijection (1:1) known as onto relation.

Let X, Y be topological spaces.

A function f : X → Y is said to be a **homeomorphism** if and only if

(i) f is a bijection

(ii) f is continuous over X

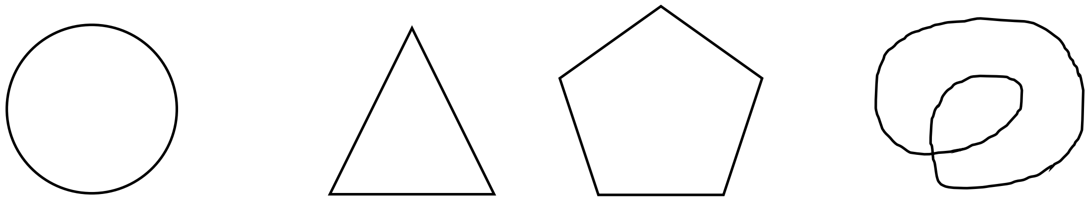
(iii) f^{-1} is continuous over Y

If a homeomorphism exists between two spaces, the spaces are said to be **homeomorphic.** These are mappings that preserve all the topological properties of a given space.

The term homeomorphic will be explained more clearly using few other examples.

Here, we are not interested in measuring the distance and making exact measurements, since it is not a geometry. We are in fact focusing on those features of mathematic objects which are invariant under such kind of continuous deformation.

Considering one another 1-dimensional object which is circle, let us call as S1. Circle can be represented in one of the following forms as shown below-



a) Euclidean space b) Polygonal c) Loop of String/Rope

Figure-7-Spaces homeomorphic to Circle

All these topological spaces are said to be homeomorphic objects. Since by continuous deformation they can be transformed to one another. Therefore we can say, two topological spaces X and Y are homeomorphic, when there is a continuous map X→Y which has a continuous inverse also Y→X.

4.2 The Topology of Polyhedra

Combinatorial topology is concerned with those topological spaces which admit dissections into suitably regular pieces. The Topology of Polyhedra is a field in mathematics in which various topological spaces are given polyhedral structures like triangles, polygon, tetrahedron and further higher dimensions of triangles. Simplexes are generalizations of triangles. Triangles are naturally very simple basic objects used to construct spaces. And simplexes are higher dimensional generalizations of triangles.

We have a standard way to represent these simplexes.

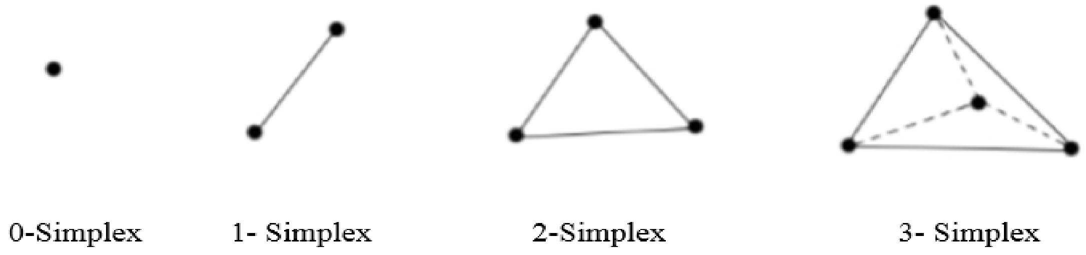


Figure-9-Example of Simplicial Complex

0-Simplex

A 0-simplex represent a **point** which lies on a X1 = 1

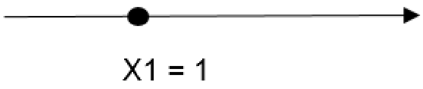


Figure-10- Zero Dimensional Simplex

1-Simplex

It will be an open segment which implies the end points are not included. 1-simplex represents a line segment which lies on a line given as X1 + X2 = 1

Figure-11- One Dimensional Simplex (open Simplex: End points not included)

2-Simplex
Here, 2-simplex represents a triangle which lies on a plane given by X1 + X2 + X3 = 1

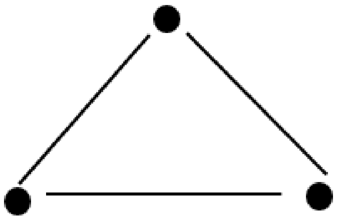


Figure-12-Two Dimensional Simplex (Open Simplex: No Boundary Included)

3-Simplex
It will be an open tetrahedron as shown below where boundary is not included.

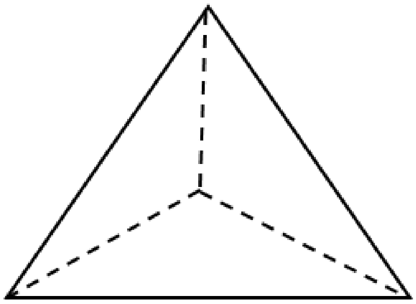


Figure-13- Three Dimensional Simplex (Open: No Boundary Included)

Thus generalizing this concept to higher dimension forms a simplex.

4.3 Simplicial Complexes

The idea of a simplicial complex is that we need to build up a space using simplexes which are not of same dimensions. These simplexes can either be disjoints (does not intersect) or if they intersect they must meet in a common face.

A formal definition as described in [19]

"*A Simplicial Complex S is a set of simplexes that satisfies below conditions:*

- *Any face of simplex in S is also in S*
- *Intersection of any two simplexes x1 and x2 is either null set or a face that exists in both x1 as well in x2*".

Simplexes may be fitted together to form interesting configurations. These configuratons are actually geometric simplicial complexes.

4.3.1 Abstract Simplicial complexes

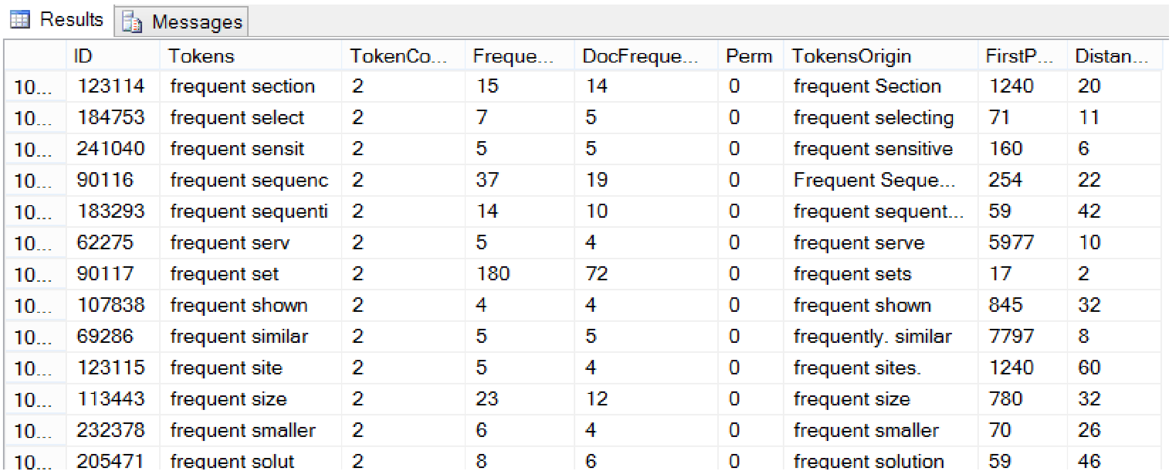
Topology is concerned with abstract polyhedra rather than concrete realizations as polyhedra of geometric simpicial complexes. So in other words, abstract simplicial complexes consider set of vertices, list of subsets or simplices instead of complete Eucledian space in which the points lies. By definition as in [13],

A finite abstract simplicial complexes is a set of objects called vertices, a^1,….a^x, and a set K of subsets of vertices, The simplexes of K satisfies the condition that any subset of a simplex of K is also a simplex of K.

In general, a simplex having n+1 vertices will be called as n-simplex. What we are doing here is that we are associating a combinatorial structure that is a simplicial complex with a polyhedron. Then we will extract the invariants of these simplicial complexes. The invariants here implies the connected components which will help us in clustering the human concepts.

In our project, we have to extract the human's concept from the documents. We have to represent each token with a vertex. If there is any link between two tokens, it will be represented by an edge joining those two vertices. Similarly, if three tokens produce one concept, they will be represented by a triangle. We have to capture all such concepts using our semantic search engine. It will finally produce a set of these keywords as n-simplex. Here n will be equal to the number of distinct keyword in our keyword set.

Below is the snapshot from SQL server query ran against database after running the semantic search engine project which provides the keywords as simplicial complexes.



	ID	Tokens	TokenCo...	Freque...	DocFreque...	Perm	TokensOrigin	FirstP...	Distan...
10...	123114	frequent section	2	15	14	0	frequent Section	1240	20
10...	184753	frequent select	2	7	5	0	frequent selecting	71	11
10...	241040	frequent sensit	2	5	5	0	frequent sensitive	160	6
10...	90116	frequent sequenc	2	37	19	0	Frequent Seque...	254	22
10...	183293	frequent sequenti	2	14	10	0	frequent sequent...	59	42
10...	62275	frequent serv	2	5	4	0	frequent serve	5977	10
10...	90117	frequent set	2	180	72	0	frequent sets	17	2
10...	107838	frequent shown	2	4	4	0	frequent shown	845	32
10...	69286	frequent similar	2	5	5	0	frequently. similar	7797	8
10...	123115	frequent site	2	5	4	0	frequent sites.	1240	60
10...	113443	frequent size	2	23	12	0	frequent size	780	32
10...	232378	frequent smaller	2	6	4	0	frequent smaller	70	26
10...	205471	frequent solut	2	8	6	0	frequent solution	59	46

Figure-14- Frequent Token sets obtained from sematic search engine

In our project, we will mainly concentrate on the maximal simplicial complexes. For more understanding, let us take a simplicial complex and find out the concepts from it.

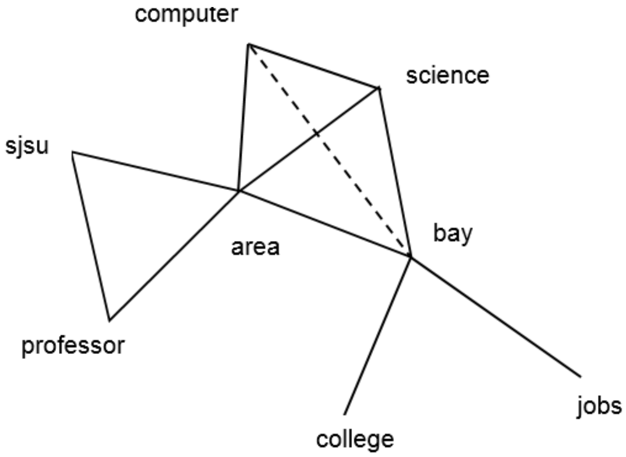


Figure-15-Maximal Simplicial Complexes

From above figure, we will get the following maximal simplexes:

One 3-simplex

computer science bay area

One 2-simplex

sjsu computer professor

Two 1-simplex

bay college

bay jobs

The next step is to compute the homology group of these simplicial complexes. These groups will then perform the clustering of the web pages/documents. The homology concepts and how we can compute the homology group using linear algebra is explained in detail rest of this paper.

4.4 Orientation

The first step to compute homology is to orient your simplicial complexes. As per the formal definition of orientation of simplicial complex dictated in [19].

“An orientation of a k-simplex is given by ordering of the vertices, written as ($v_{0,}$ v_1…..v_k), with a rule that two orderings define the same orientation if and only if they differ by an even mutation.”

Taking 1 dimensional simplex S:

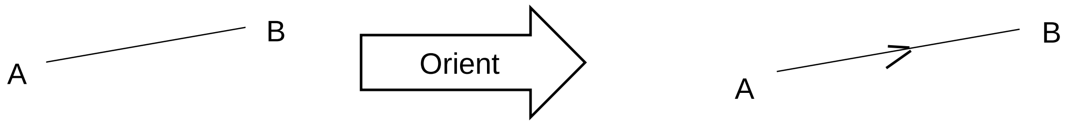


Figure-16-Orientation of one dimensional spaces

After orientation, we denote the simplicial complex by ordered vertices, as here it will be named as
S = (A B)

Using this orientation, we can define the boundary of this simplex denoted by
∂ (S) = B – A
Now, considering it further in a 2 dimensional simplex S:

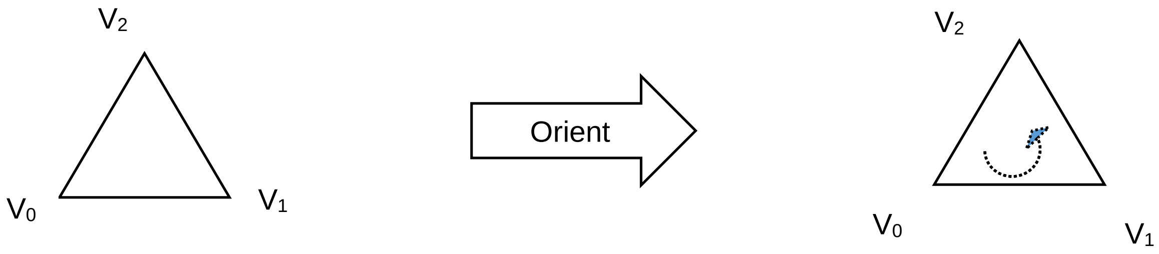


Figure-17-Orientation of two dimensional space

Here, we can orient this simplex in two different ways-
(v0 v1 v2) ~ (v1 v2 v0) ~ (v2 v0 v1)
(v0 v2 v1) ~ (v2 v1 v0) ~ (v1 v0 v2)

In this case, considering the first one as the orientation of this simplex which is (v0 v1 v2), we can again define its boundary.

To define ∂(S) for higher dimensions, we use a hatted approach according to which when we have an ordered set [v0 v1 v2 v3] then we will select that subset which is not having that hat (here hat denoted by underline) which in this case will be [v0 v2 v3].

Following this approach, we can write the boundary of the above triangle as-
∂(S) = (v0 v1) – (v0 v2) + (v1 v2)

As per [19], the generic formula for boundary operator can be written as –

$$\partial_k(\sigma) = \sum_{i=0}^{k} (-1)^i (v_0, \ldots, \widehat{v_i}, \ldots, v_k),$$

Where ^ means omit this vertex and take others.

There is a fundamental formula in algebraic topology called as double boundary identity –
∂^2 = 0

We can also prove this formula on the above simplex S as-

$\partial(\partial(v0\ v1\ v2)$

$= \partial(v1\ v2) – \partial\ (v0\ v2) + \partial\ (v0\ v1)$

$= (v2 – v1) – (v2 – v0) + (v1 – v0)$

$= v2 – v1 – v2 + v0 + v1 – v0$

$= 0$

Which proves the above theorem.

4.5 Chain Complexes and Boundary Operator

A chain complex is defined as a set of vector spaces that strictly follows the double boundary identity theorem $\partial^2 = 0$. We have sequence of groups called as chains. These chain groups actually represent the dimension of the object we are considering.

C_0: 0 dimension chains: points

C_1: 1 dimension chains: lines and edges

C_2: 2 dimension chains: discs

C_3: 3 dimension chains: solid figures.

Also, between these different chain groups there exists a mapping from one group to another. It is defined as $\partial_n = C_n \rightarrow C_{n-1}$ called as boundary operator.

Geometrically we can explain the boundary map as a mapping going from higher dimension to lower ones. As we move from the faces to edges and edges to points we are actually mapping from one dimension to another. This can be diagrammatically shown as –

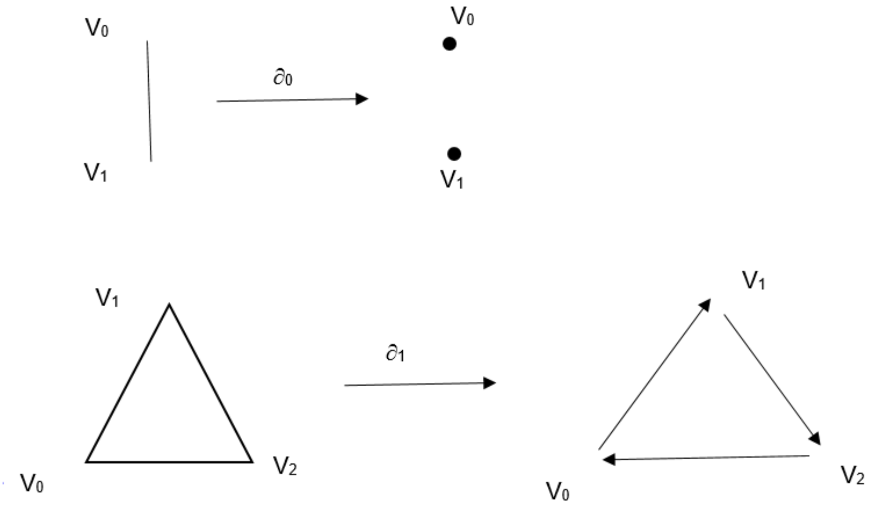


Figure-18-Boundary Operator

These chain complexes explains a relationship that exists between the boundaries of different dimensions.

4.6 Simplicial Homology

The homology provides the fundamental details of characteristics of a topological space. The information it provides is about the number of holes present in the data and thus helps in indicating the similarities among the data. In linear algebra, computing homology groups has number of implementations and algorithms as mentioned in [2], [5], [13], and [29]. All these algorithms deals with how in a topological space, we can capture the invariants properties by obtaining the homology groups.

These algorithms mostly has to initially create the triangulations, polyhedral structure and then find homology groups. We will also create simplicial complexes in the first stage. Regarding brief introduction to homology theory, it is a branch of algebraic topology that works on finding similarities and differences between various topological spaces. This is done by measuring the invariants properties of these topological spaces.

Though there are a number of approaches of computing the homology groups, but the most simplex and basic method is simplicial homology. Simplicial homology deals with representing your data in the form of triangulations and then obtain these homology groups. The main idea here is that if we are having different triangulations for the same space, they will definitely be having same homology groups. This is the reason Simplicial homology is one of the important base for doing a well-defined clustering of documents.

4.7 Computing Homology Groups

For computing the homology group, we again need to concentrate on the chain complexes that we discussed earlier in this paper. The chain complexes follows a double boundary identity. This theorem implies that if we start anywhere in the diagram (given below) and start with any group, take boundary operator and again take the boundary, it will definitely give a 0 in the result. This is the basic theorem that defines the definition of homology.

Let us suppose for any given space X, we have the chain complex as given below-

$$\begin{tikzcd}
C_3 \arrow[r, "\partial_3"] & C_2 \arrow[r, "\partial_2"] & C_1 \arrow[r, "\partial_1"] & C_0 \arrow[r, "\partial_0"] & 0 \\
\text{3-dim chains} & \text{2-dim chains} & \text{1-dim chains} & \text{0-dim chains} &
\end{tikzcd}$$

Figure-19-Chain Complex

Now, at every stage of this chain complex we can get the homology group. We can ask two questions from the above chain complex.

1) Kernel

   Considering the 1 dimension chain group, the kernel of dimension-1 is a sub group of C_1 chain group and these are those subsets that are send to dimension-0 under the mapping of ∂_1 .These subgroups are actually contained in the C_1 chain group.

   Ker (∂_1) = Z_1 → this denotes the group of cycles.

2) Image

Now, considering the other side of C_1 chain group, we can see the image of $\partial_{2.}$ These 2-dimension images will also be the sub groups of C1 and they are called as group of boundary.

Img (∂_2) = B → this denotes the group of boundaries.

Finally the double boundary theorem tells us that when we apply boundary operator at any group chain to get a boundary and then again apply the same operator, we will get a 0.
This tells us B (boundary) are a part of Z (cycles).

Img (∂_2) contained in Ker (∂_1)

If something is a boundary, it is necessary a cycle.
So, it is the relationship between these two groups that generates the homology.

Homology is the quotient group of cycles to the boundaries, that is-

$H_n = \frac{Zn}{Bn}$ $Z_n = \ker \partial_n$

$B_n = \text{img } \partial_{n+1}$

The result for homology group can end up with different objects. For an example, they could be finite or infinite commutative group. We might get 0, Z, Z+Z or some higher dimensions of Z. Z here denotes the Integer values.

In most of the research papers as defined in [29], algorithms are generally concentrating in finding the rank of this homology group. This expression can be denoted as-

$\beta_{n\,=}$ rank ($H_n(S)$)

These are called as Betti numbers. These betti numbers can of any dimensions. In simple terms, these numbers measures the count of n- dimensional holes in our space S.

5. HOMOLOGY OF TETRAHEDRON: MANUAL CALCULATION

In order to completely understand the computation of homology groups, I have worked to firstly manually compute these homology groups of a very simple topological space which is a tetrahedron.

Then I have compare the results of these manual calculation with the results that I have obtained by running my program for computing homology groups.

Firstly, consider a polyhedral as shown below

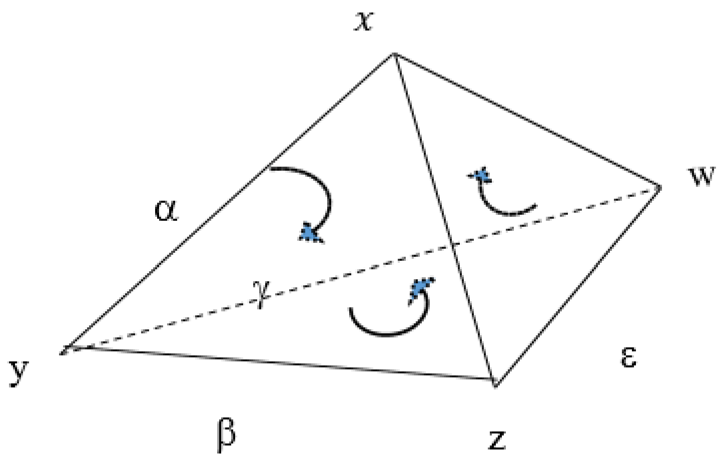


Figure-20-Topological Space: Tetrahedron

From this figure:

Δ_0 = 0 dimensional < (x), (y), (z), (w) > COUNT = 4

Δ_1 = 1 dimensional < (x, y), (y, z), (z, w), (w, x), (w, y), (x, z)> COUNT = 6

Δ_2 = 2 dimensional < (x, y, z), (x, y, w), (x, z, w), (y, z, w) > COUNT = 4

$$\begin{tikzcd}
C_3 \arrow[r, "\partial_3"] \arrow[d] & C_2 \arrow[r, "\partial_2"] \arrow[d] & C_1 \arrow[r, "\partial_1"] \arrow[d] & C_0 \arrow[r, "\partial_0"] \arrow[d] & 0 \\
\begin{array}{l}\text{Since we have}\\ \text{Hollow space, therefore}\\ \text{For 3-dim, B = 0}\end{array} & \begin{array}{l}(x\ y\ z)\ (x\ z\ w)\\ (y\ w\ z)\ (x\ w\ y)\end{array} & \begin{array}{l}(x\ y)\ (y\ z)\ (z\ x)\\ (x\ w)\ (w\ z)\ (w\ y)\end{array} & x,\ y,\ z,\ w &
\end{tikzcd}$$

Figure-21-Chain Complex and Boundary Operator

From Above definition, we can calculate the homology group by taking the quotient of cycles to boundary as-

$$H_n = \frac{Zn}{Bn}$$

Where,

$Z_n = \ker \partial_n$ cycles

$B_n = \text{img } \partial_n + 1$ boundaries

a. Homology for 0- Dimension, H_0

The cycles and boundary for this chain group will be-

$Z_n = C_0 = < x, y, z, w >$

$B_0 = < y – x, z – y, x – z, w – x, z – w, y – w>$

Put $B_0 = 0$

⇨ x = y = z = w all vertices are identical

Thus, for 0-dimension, the homology group will be Z.

$$\boxed{\frac{Z0}{B0} = Z^1}$$

b. Homology for 1- Dimension, H_1

The cycles and boundary for this chain group will be-

$z_1 = \ker \partial_1$ (1 dimension chains whose boundary = 0)

Now, in this case, every element in C_1 can be written as

$\alpha\,(x\ y) + \beta(y\ z) + \gamma(z\ x) + \delta\,(x\ w) + \varepsilon(w\ z) + \eta(w\ y)$

Assuming, boundary = 0

- ⇨ $x(-\alpha + \gamma - \delta) + y(\alpha - \beta + \eta) + z(\beta - \gamma + \varepsilon) + w(\delta - \varepsilon - \eta) = 0$

Here, we have to perform some Linear algebra and using matrix row reduction, solve the following equations:

$$\begin{array}{cccccc} \alpha & \beta & \gamma & \delta & \varepsilon & \eta \\ \end{array}$$
$$\begin{bmatrix} -1 & 0 & 1 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & -1 & -1 \end{bmatrix}$$

All solutions will be multiple of:

$$\left.\begin{array}{l} \alpha + \beta + \gamma \\ -\alpha - \beta + \delta + \varepsilon \\ -\alpha + \delta + \eta \end{array}\right] \text{———} \quad 1$$

All these solutions will be geometrically cycles only.

Therefore, the image in this case will be

Ker = Z^1

Now, boundaries:

$B_1 = \operatorname{img} \partial_2$

$B_1 = \langle\ (\alpha + \beta + \gamma), (-\gamma + \delta + \varepsilon),\ (-\beta + \varepsilon - \eta), (\delta + \eta - \alpha)\ \rangle$

Looking closely, each term of boundary can be written as combination of (1) equations Again, taking quotient of image and kernel, we get the homology group for this dimension also.

$$\boxed{H_1 = \frac{Z1}{B1} = 0}$$

c. Homology for 3-Dimension, H2

The cycles and boundary for this chain group will be-

Cycles:

$z_2 = <(x\ y\ z) + (x\ z\ w) + (y\ w\ z) + (x\ w\ y)>$

- ⇨ All combinations of which will give $\partial = 0$
- ⇨ Existing cycles

$z_2 = < A+B+C+D> \approx Z^1$

Boundary:

We have hollow figure, so we don't have any 3-dimensional face or figure.

Thus the boundary in such case will be equal to 0.

Thus taking quotient of cycles over boundary, we get homology group for dimension – 3.

$$\boxed{H_2 = \frac{Z2}{B2} = Z^1}$$

This concludes that the homology group for a simple hollow tetrahedron will be:

$$H_0 = Z_1$$
$$H_1 = 0$$
$$H_2 = Z_1$$
$$\text{And, } H_n(x) = 0, \quad n \geq 3$$

This is denoted for all other higher dimensions, homology will be zero in this case.

6. PROJECT FLOW - SCOPE OF THE PROJECT

This section provides a high level architecture of our project. It mention the different components and steps done to accomplish the required results.

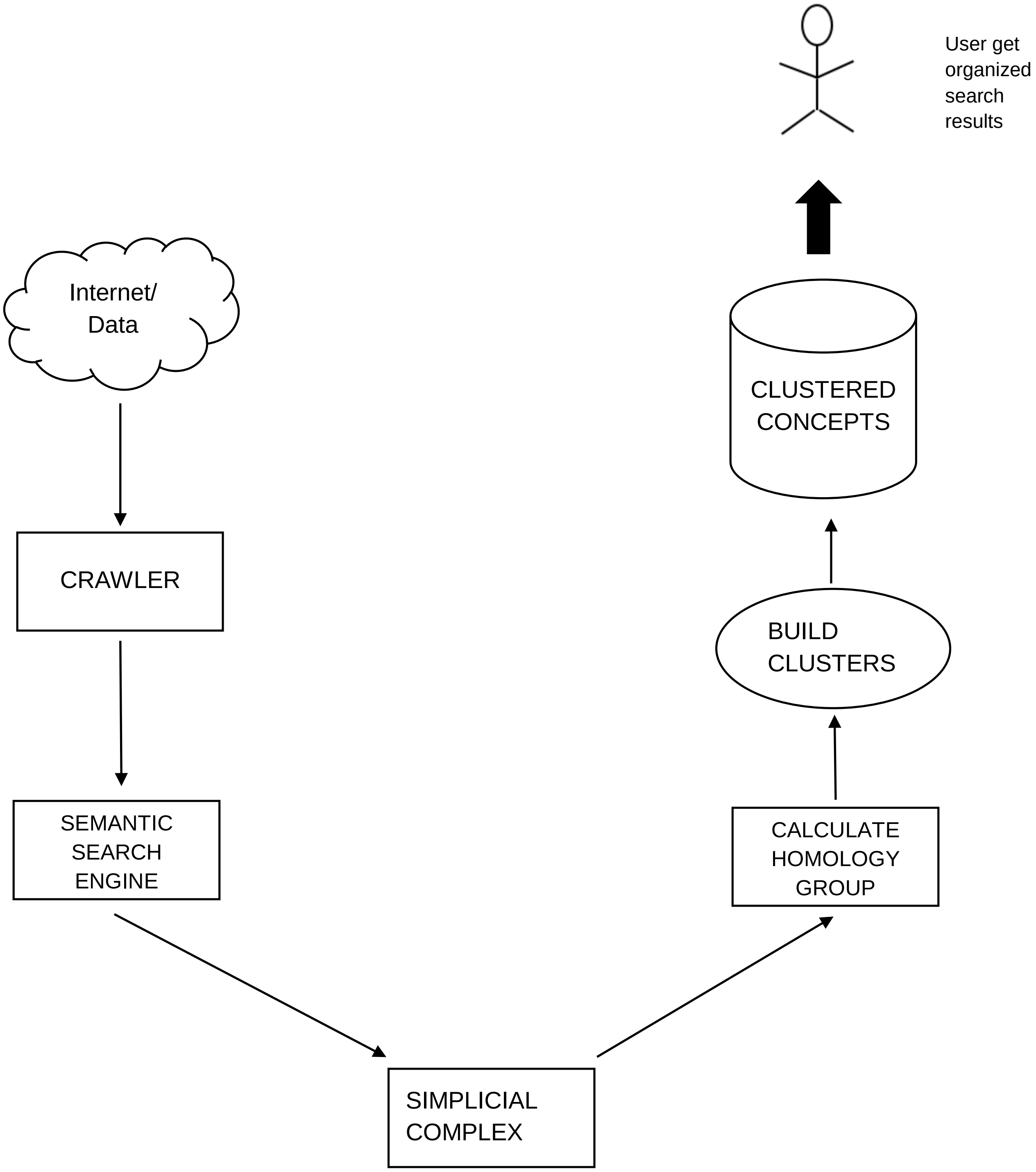


Figure-22-Project Architecture

7. IMPLEMENTATION: ALGORITHM AND APPROACH

The implementation part includes a number of mathematic and linear algebra theorems. Here, more concentration is on finding the homology for each dimension. So, we will be talking about the betti numbers and how we can find them along with the homology group.

We will be given maximal simplexes as input. The first stage is extracting all the simplexes from the given set of maximal simplexes. Then, we need to represent these in some algebraic fashion. Here, tree is used as a data structure to store the extracted simplexes where every simplex is presented as one node of a tree.

In this tree, the nodes are arranged in an ascending manner. By this we means that each level of tree represents the next level of simplex. For example, let's say we have a maximal simplex [0, 1, 2]. This will arranged in the following way-

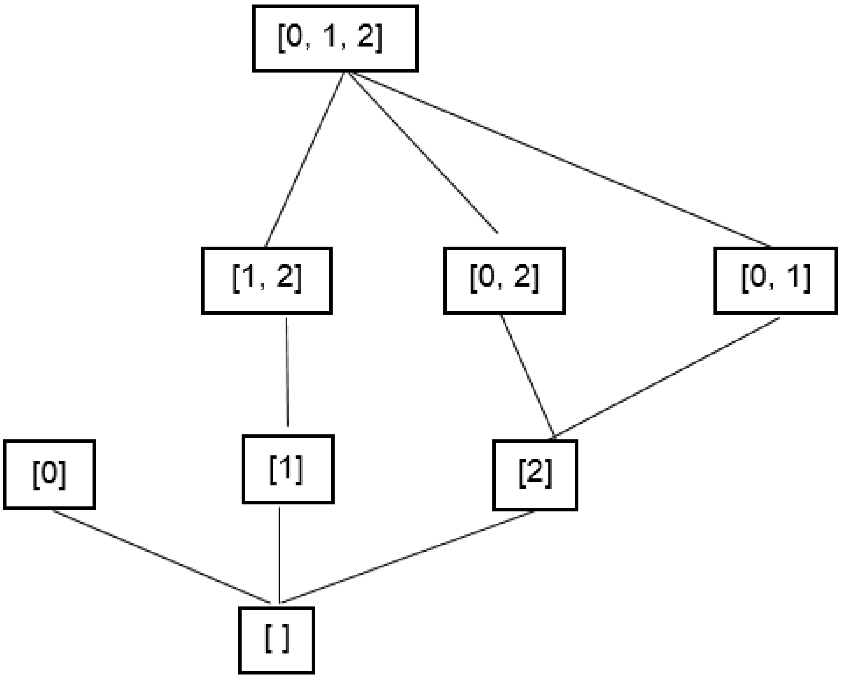


Figure-23-Tree Structure for storing chain complexes

This tree structure in actual provide us the different chain complexes of each dimension at every level of the tree. These chains are the vectors which can be written in the form of matrices.

For every dimension C_n a matrix will be created that denote the boundary termed as boundary matrix.

For $\partial_1 = C_0 \rightarrow C_1$

The chain groups of 0 dimension, C_0 will form the rows of matrix and chain groups of C_1 will become the columns of this boundary matrix.

Thus the boundary matrix will be shown as:

$$\partial_1 = \begin{array}{c|cccc|} & [xy] & [yz] & [zx] & \ldots \\ x & & & & \\ y & & & & \\ z & & & & \\ w & & & & \end{array}$$

Once we have the boundary matrices form of these chain vectors, we can then find the two main components required to compute homology which are image and kernel.

For obtaining these components, we firstly need to convert the boundary matrix into reduced form. This can be done by converting your boundary matrix to smith normal form. Using rank nullity theorem we will calculate the rank of matrix. In the reduced boundary matrix, Knock off all the zeroes and the rest non-zeroes entries are stored.

For Chain group $C_{1,}$

Kernel = Number of non-zero entries in the reduced form of boundary matrix ∂_1

Image = Number of non-zero entries in the reduced form of boundary matrix ∂_2

Finally, taking quotient of these two, we will get the homology at each dimension.

7.1 Rank Nullity theorem

In linear algebra, the rank nullity theorem states that in a matrix we have rank and nullity, and if we add both these it will give the total number of columns in that matrix.

Rank (A) + Null (A) = Total number of columns of matrix A

Thus using this theorem, we can easily find the rank of a given boundary matrix to obtain the kernel and image.

7.2 Gaussian Elimination

In mathematics, Gaussian elimination is one of the fundamental procedure to solve the equations which are of the form: Mx = N

These matrix equations can be solved by firstly putting in the augmented format. Perform row operations until we obtain the echelon form of matrix. In this the upper triangulation is obtained in a matrix. Finally then then solving simple equations we can solve these complex matrix equations.

7.3 Smith Normal Form

For a given simplicial complex, it is the most well defined algorithm used to compute the homology group. Smith normal form has the most important application of finding homology groups. In SNF form, a matrix have to be converted in a form where only diagonal entries are non-zero. It is not essential that only square matrix can be converted into SNF form. Any matrix of m*n size can be converted to this form. In order to convert any matrix to SNF, we have to perform continuous row and column operations.

Let suppose, we have a matrix names P, we have to select two identity matrix R and C of the corresponding dimension. We have to perform continuous rows and column operations on P until we get a diagonal matrix.

The important point here is, whenever a row operation is done on P, similar operation has to be performed on R. Whenever a column operation is performed on P, similar operation needs to be performed on C.

This needs to be repeated until a diagonal matrix is obtained.

7.4 Pseudo Code

```
Computing Homology (max_simplex, i, dem)
{
        Extract all simplexes from max_simplex
        Index = 0
        For every simplex
        {
                Simple[i] = index
                Index = index +1
        }
        Generate a tree structure from these simplex which will form a structure of chain
        complexes
        Level 0 contains 0-simplex
        Level 1 contains 1-simplex
        Level 2 contains 2-simle and so on.

Build boundary matrix from these chain groups
Reduce the boundary matrix to SNF form
Find image and kernel for each chain group
Find the quotient of kernel and image to obtain the homology group

}
```

8. EVALUATION OF RESULTS

For testing the results obtained from or program, we firstly run our program by finding homology group of basic tetrahedron space. We calculate manually the homology groups of this tetrahedron and then compare the homology groups obtained with the results that we received by running our program.

As per our expectations, we got the same results that indicate our algorithm is running correctly. Furthermore, we try to compute the homology for few other basic topological space.

8.1 Homology Group of tetrahedron

This is the most basic topological space to compute the homology group. The input for this was [['X','Y','Z','W']] which make up a tetrahedron.

While manually calculating the homology group, the results obtained were-

$H_0 = Z_1$
$H_1 = 0$
$H_2 = Z_1$

The computed results after running our program are as below:

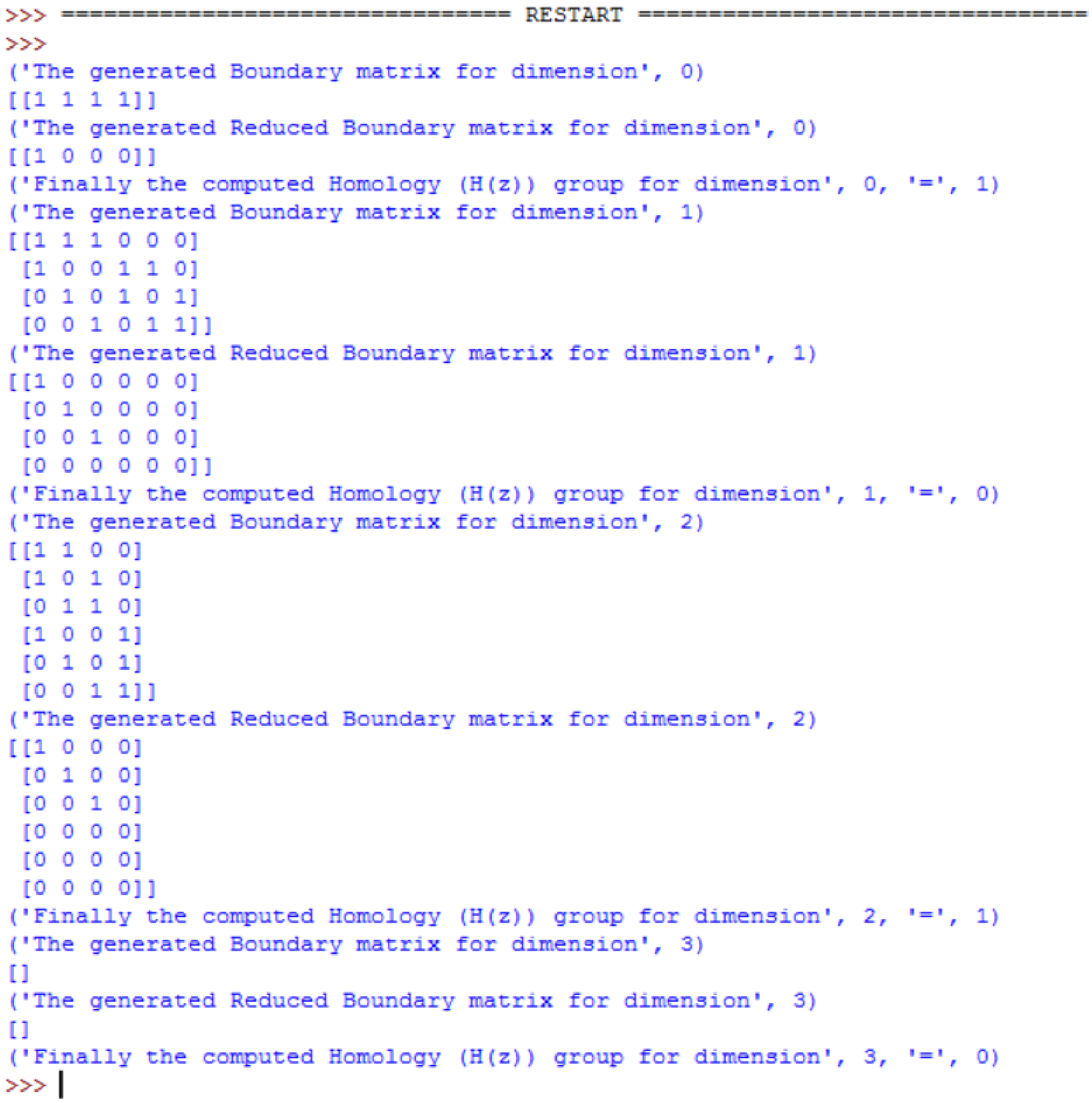
Input:
[['X','Y','Z','W']]

Output:

$H_0 = Z_1$
$H_1 = 0$
$H_2 = Z_1$

The two results are exactly similar indicating the correctness of our program. Also, in our program instead of giving Z, it is giving integer values like 0, 1, 2 and so on.

The alphabet Z that we used while manual calculating actually indicate the integer values. In our program, the integer value 2 means Z to the power 2.



```
>>> ================================ RESTART ================================
>>>
('The generated Boundary matrix for dimension', 0)
[[1 1 1 1]]
('The generated Reduced Boundary matrix for dimension', 0)
[[1 0 0 0]]
('Finally the computed Homology (H(z)) group for dimension', 0, '=', 1)
('The generated Boundary matrix for dimension', 1)
[[1 1 1 0 0 0]
 [1 0 0 1 1 0]
 [0 1 0 1 0 1]
 [0 0 1 0 1 1]]
('The generated Reduced Boundary matrix for dimension', 1)
[[1 0 0 0 0 0]
 [0 1 0 0 0 0]
 [0 0 1 0 0 0]
 [0 0 0 0 0 0]]
('Finally the computed Homology (H(z)) group for dimension', 1, '=', 0)
('The generated Boundary matrix for dimension', 2)
[[1 1 0 0]
 [1 0 1 0]
 [0 1 1 0]
 [1 0 0 1]
 [0 1 0 1]
 [0 0 1 1]]
('The generated Reduced Boundary matrix for dimension', 2)
[[1 0 0 0]
 [0 1 0 0]
 [0 0 1 0]
 [0 0 0 0]
 [0 0 0 0]
 [0 0 0 0]]
('Finally the computed Homology (H(z)) group for dimension', 2, '=', 1)
('The generated Boundary matrix for dimension', 3)
[]
('The generated Reduced Boundary matrix for dimension', 3)
[]
('Finally the computed Homology (H(z)) group for dimension', 3, '=', 0)
>>> |
```

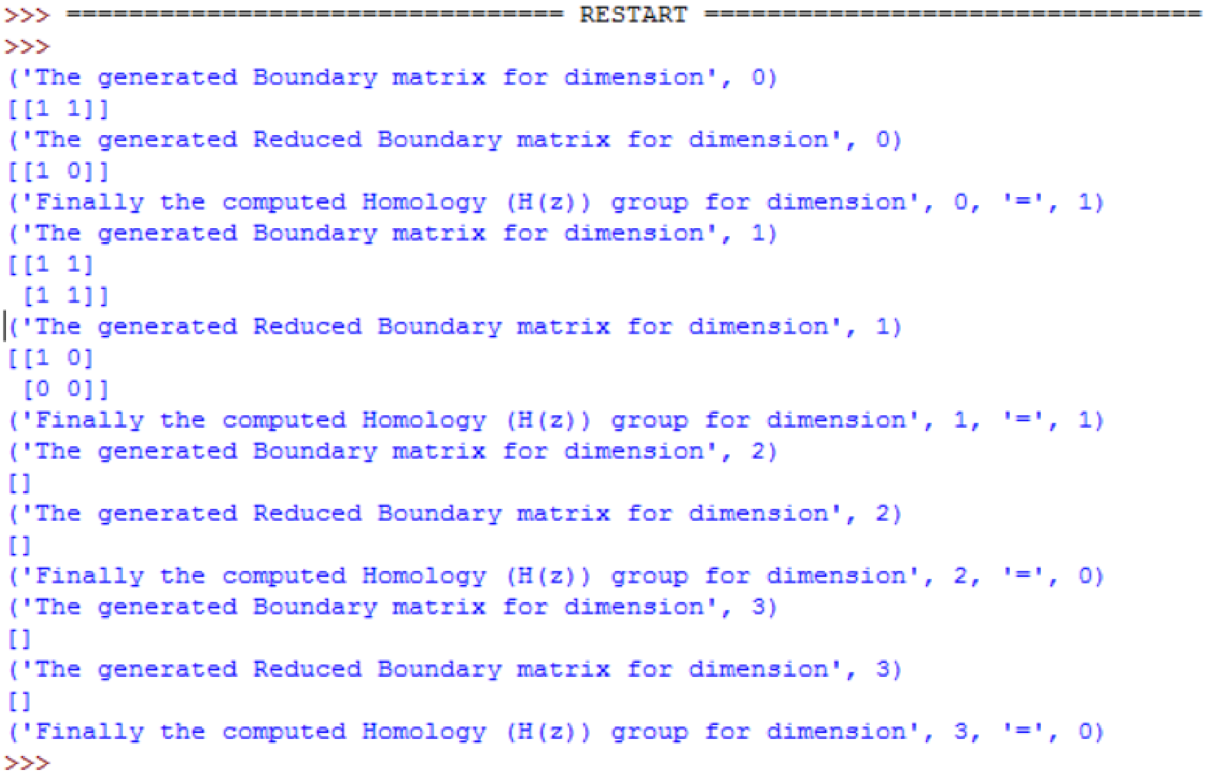
Figure-24-Results for homology of Tetrahedron

8.2 Homology Group of a Mobius strip

Mobius strip is one of the interesting non-orientable surface in mathematics. It is a space which has only one side and one boundary only.

Topologically, this can be represented as
[['A','B'], ['B','A']].

This is because, it is generated by taking the ends of one, mold it and glued it together with the other opposite end. When [['A','B'], ['B','A']] is given as input to our program, we receive homology group given as below:



```
>>> ================================ RESTART ================================
>>>
('The generated Boundary matrix for dimension', 0)
[[1 1]]
('The generated Reduced Boundary matrix for dimension', 0)
[[1 0]]
('Finally the computed Homology (H(z)) group for dimension', 0, '=', 1)
('The generated Boundary matrix for dimension', 1)
[[1 1]
 [1 1]]
('The generated Reduced Boundary matrix for dimension', 1)
[[1 0]
 [0 0]]
('Finally the computed Homology (H(z)) group for dimension', 1, '=', 1)
('The generated Boundary matrix for dimension', 2)
[]
('The generated Reduced Boundary matrix for dimension', 2)
[]
('Finally the computed Homology (H(z)) group for dimension', 2, '=', 0)
('The generated Boundary matrix for dimension', 3)
[]
('The generated Reduced Boundary matrix for dimension', 3)
[]
('Finally the computed Homology (H(z)) group for dimension', 3, '=', 0)
>>>
```

Figure-25-Result for homology of Mobius strip

9. WEB CLUSTERING

There are millions of web pages existing on the internet. In order to organize these millions of web pages into different categories as per their relevance, a very efficient information retrieval methods are required. Dr. TY Lin in his paper [22] has discussed about how we can perform organized clustering using combinatorial topology. Also, he has mentioned that every document is represented by concepts. In our project, we had already gathered concepts from the knowledge base and have represented these concepts in the form of simplicial complexes. Now, two or more documents can be clustered together if they are having a connection in their concepts means they have similar concepts that are existing in both the documents.

A new concept called layered clustering has been discussed in [24]. According to this paper, a set of frequently co-occurring keywords can represent a document and the concepts that it contains. Using these extracted associations, we can perform Layered clustering.

When we create a simplicial structure of our concepts, they are always created in a skeleton fashion. The most basic property of a maximal simplicial complex is that if in a complex we have a set of simplexes and all the subsets of these simplexes will also be a part of that complex. Thus, layered wise these simplicial complexes are arranged in a tree fashion. This can be shown as follows-

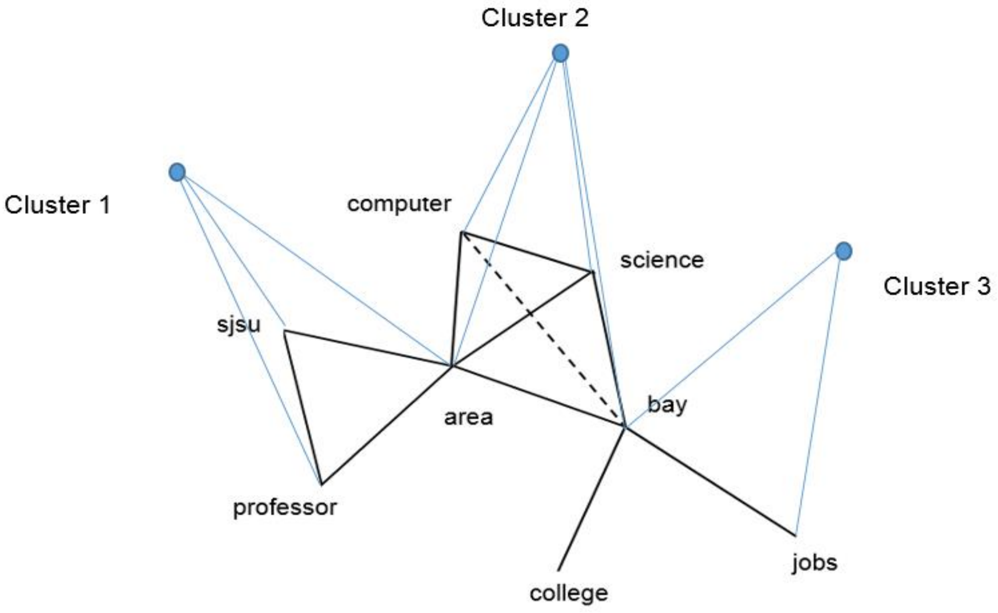


Figure-26-Web Clustering on a simple simplicial complex

Finding homology of the simplicial complexes of concepts has further filtered and organized these results and clustering. Homology provides the connections between the spaces. Also, if we have different simplicial complexes having same invariants properties, they will be having same homology group values.

Thus, instead of using measured distance, rank etc., homology will use the topological invariant properties to find the relevance of documents and thus the clustering done using this approach will be much more efficient and defined as compared to the already existing clustering techniques.

10. FUTURE SCOPE AND CONCLUSION

This paper presents a very unique approach toward clustering where the concepts are used to perform the clustering. We are using homology theory of simplicial complexes to capture the connected components. A number of algorithms have come to existence as mentioned in [1], [18] and [25]. But they cannot provide an efficient method when high dimensions come to existence. Here, it is not one token or keyword that represents a documents. Instead a set of association defines the concepts in a document.

Here concepts are represented by a simplicial complexes in high dimensions. It is finally then arranged in a Skelton fashion. We have evaluated from above results that this is an efficient way to completely identify the connections among web documents.

The homology groups that we have found has many useful applications. Below are listed some of the areas where these homology groups can be used-

a) Biological Applications
b) Information retrieval – A new approach of a search engine

Furthermore, there are a number of ways in which we can extend this project. One of the extension on which we can work is to implement this algorithm using Hadoop as on back end. Though our results shows that we are getting correct and perfect results for all of the topological spaces. Also, for testing the data that our project can handle, we have run our project with 10,000 documents. But still, web pages present on the internet are in billions and with every second they are increasing in numbers. To handle such kind of unstructured, high dimension big data, Hadoop can be used to further increase the performance of our algorithm. Hadoop uses a Hadoop Distributed File System. It provides features like fault tolerance and cheap hardware. Nowadays, it is the most reliable platform in the applications that make use of big data. It also takes care of load balancing and scheduling. The homology for such a huge dataset can easily be calculated using a parallel approach with increased performance.

11. REFERENCES

[1] Ajitha Annadurai, MTech in Information Technology, Madras Institute of Technology "Architecture of Personalized Web Search Engine Using Suffix Tree Clustering".

[2] Darrell Allgaier (Grove City College), David Perkinson (Reed College), Sarah Ann Stewart (North Central College), John Thurber (Eastern Oregon University), "Homology of Simplicial Complexes", 2004.

[3] Eric Weisstein, "Gaussian Elimination", Wolfram MathWorld, http://bit.ly/1T2gXDs.

[4] Eric Weisstein, "Topological Space", Wolfram Research, http://bit.ly/1R6cb9E.

[5] Gunnar Carlsson, Rick Jardine, Topological Data Analysis and Machine Learning Theory, 2012.

[6] Gunnar Carlsson, Topology and Data, AMERICAN MATHEMATICAL SOCIETY, S 0273-0979(09)01249-X, 2009.

[7] George Karypis, Eui-Hong (Sam) Han, Vipin Kumar, "Chameleon: Hierarchical Clustering Using Dynamic Modeling", University of Minnesota.

[8] Guodong Hu, Wanli Zuo, Fengling He,Ying Wang, College of Computer Science and Technology, Jilin University, "Semantic-based Hierarchicalize the Result of Suffix Tree Clustering".

[9] Gunaar Carlsson, "Topology and Data", American Mathematical Society, Volume 46, Published January 29, 2009.

[10] Gunnar Carlsson (Stanford University), Rick Jardine (University of Western Ontario), Dmitry Feichtner-Kozlov (University of Bremen), Dmitriy Morozov (Lawrence Berkeley National Laboratory) "Topological Data Analysis and Machine Learning Theory", Oct 2012.

[11] Greg Friedman Texas Christian University, "An elementary illustrated introduction to simplicial sets", Mathematics Subject Classification: 18G30, 55U10 Keywords: Simplicial sets, simplicial homotopy, 2011.

[12] Moises Goldszmid, Mehran Sahami, "A Probabilistic Approach to Full-Text Document Clustering" Computer Science Department Stanford University.

[13] P. J. Hilton, S. Wylie, Homology Theory: An Introduction to Algebraic Topology.

[14] Peter Saveliev, Intelligent Perception, http://bit.ly/1I9y5qt, 2010.

[15] Rank nullity Theorem, Wikipedia, http://bit.ly/1T2gK3e.

[16] Ramakrishnan Srikant, Rakesh Agrawal, “Fast Algorithms for Mining Association Rules”, IBM Almaden Research Center, 20th VLDB Conference, 1994.

[17] Simplicial Homology, Wikipedia, https://en.wikipedia.org/wiki/Simplicial_homology.

[18]Tsau Young (T. Y.) Lin, Albert Sutojo and Jean-David Hsu; Concept Analysis
And Web Clustering using Combinatorial Topology, 2006.

[19] T.Y. Lin, Wesley W. Chu, Foundations and Advances in Data Mining.

[20]Tsau Young (T. Y.) Lin, I-Jen Chiang, Clustering Web Pages by Granulating the Latent Semantic Space, 2005.

[21] Tsau Young (T. Y.) Lin, Jean-David Hsu, “Knowledge Based Search Engine: Granular Computing on Web”, Department of Computer Science, San Jose State University.

[22] Tsau Young Lin, I-Jen Chiang, “A simplicial complex, a hypergraph, structure in the latent semantic space of document clustering”.

[23] Tapas Kanungo, Senior Member, IEEE, David M. Mount, Member, IEEE, Nathan S. Netanyahu, Member, IEEE, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu, Senior Member, IEEE, “An Efficient k-Means Clustering Algorithm: Analysis and Implementation”.

[24] Tsau Young (’T. Y.’) Lin Department of Computer Science San Jose State University, Jen Chiang Graduate Institute of Medical Informatics Taipei Medical University “Granulate and Conquer: Clustering Web Pages Semantically using Combinatorial Topology”.

[25] Triangleinquality, “Abstract Simplicial Complex in Python”, http://bit.ly/1MBA062.

[26] Vanessa Robins, Research School of Physics and Engineering, “Computing Homology’, http://bit.ly/1NrCRlt.

[27] Xiaojin Zhu, "Persistent Homology: An Introduction and a New Text Representation for Natural Lan- guage Processing," The 23rd International Joint Conference on Artificial Intelligence (IJCAI), 2013.