USING TOPOLOGICAL DATA ANALYSIS FOR TEXT CLASSIFICATION

by

Pratik Doshi

A thesis submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Master of Science in
Computer Science

Charlotte

2018

Approved by:

Dr. Wlodek Zadrozny

Dr. Srinivas Akella

Dr. Zackery Wartell

ABSTRACT

PRATIK DOSHI. Using Topological Data Analysis for Text classification. (Under the direction of DR. WLODEK ZADROZNY)

I show that by applying discourse features derived through topological data analysis (TDA), namely homological persistence, we can improve classification results on the task of movie genre detection, including identification of overlapping movie genres. On the IMDB dataset we improve prior art results, namely we increase the Jaccard score by 4.7% over a recent results by [1]. I also significantly improve the F-score (by over 15%) and slightly improve the hit rate (by 0.5%, ibid.). The limitations of my work, mostly due to the smaller data set, are also discussed in the end. I see my contribution as threefold: (a) for general audience of computational linguists, I want to increase their awareness about topology as a possible source of semantic features; (b) for researchers using machine learning for NLP tasks, I want to propose the use of topological features when the number of training examples is small; and (c) for those already aware of the existence of computational topology, I see this work as contributing to the discussion about the value of topology for NLP, in view of mixed results reported by others.

ACKNOWLEDGEMENTS

It is with immense gratitude that I acknowledge the help and support Dr. Wlodek Zadrozny throughout my thesis. Dr. Zadrozny's guidance and encouragement has fueled the my curiosity and motivated me to push my limits throughout the process. I consider it an honor to have worked with Dr. Zadrozny.

I would like thank my Thesis Committee Members Dr. Srinivas Akella and Dr. Zackery Wartell for being a positive support by always asking me the right questions, encouraging me to give my best.

My special thanks to the College of Computing and Informatics for providing me the necessary infrastructure and all the resources required for my thesis.

At last, but most importantly, I would like my to thank my parents for inspiring me at every step in my life. Thesis is full of challenges, especially when you are introducing a new topic to the world, but being focused and dedicated is something that got me through. I would like to thank my elder sister, Trusha Doshi for imbibing this valuable lesson in me and being my role model when it comes to perfection and perseverance.

PREFACE

In this paper I describe an experiment paper on using Topological Data Analysis (TDA) as a tool for feature extraction for text classification. I show that at least when the text can be interpreted as describing a progression of events (as in movies), topological features, namely, homological persistence, when added to representation of text, can significantly improve classification accuracy. This work can be viewed as a contribution to discussion about the value of TDA for text processing, and is one of only handful such attempts in the NLP community. [2] and [3] are the only papers presenting applications of TDA for text processing, I compare their results in detail in later sections.

In this work, I apply Topological Data Analysis to the problem of text classification, and specifically to the classification of movie genres. I show that adding topological features derived from text structure improves classification: I significantly improve the Jaccard score by 4.7% compared to the baseline and previously published results [1]; my work also slightly improve the hit rate.

Very roughly, and in the simplest case shown in this paper, Topological Data Analysis (TDA), when applied to text classification, can be thought of as adding graph- and hypergraph-based features derived from the document structure, such as repetitions of words in different sentences. These features can be used in a stand alone fashion, or in addition to the usual collection of features based on words, multiword expressions or word embeddings [2]. The topological features are derived on the basis of sophisticated mathematical and computational properties.

I see my contribution as threefold: (a) for general audience of computational linguists, I want to increase their awareness about topology as a possible source of

semantic features; (b) for researchers using machine learning for NLP tasks, I want to propose the use of topological features when the number of training examples is small; and (c) for those already aware of the existence of computational topology, I see this work as contributing to the discussion about the value of topology for NLP.

This work uses the methods and tools introduced by [2] and applies them to a non-trivial discourse processing task; namely, the classification of movie genres on IMDB data. As explained by [2] TDA might be useful when processing text with a series of events and so we start with classifying movie plots.

I present significant improvements in accuracy when topological features (homological persistence) are used in classification. I perform this classification on the basis of persistence homology diagrams (barcodes) generated using Javaplex for Matlab. Section 3 details my experiment and examples. Section 4 discusses the advantages and limitations of these experiments; the advantages being in the small number of examples needed for training, and the limitation in the small number of test cases I considered.

My work can be contrasted with recent results of [3] showing no improvements from persistence on the sentiment analysis tasks on two corpora of which one was IMDB.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

The problem of classification has been widely studied in the database, data mining, and information retrieval communities [4]. With the exponential growth of textual data, the problem for efficiently classifying textual data gains significant attention. Alot of machine learning algorithms have been implemented over the years to improve the accuracy of the classification problem. However, the approach of using Topological Data Analysis (TDA) has been seldomly used for textual data.

In this chapter I provide a brief introduction to modules of Data Science that form the foundation of my thesis. I explain each of these in depth in their respective chapters.

1.1 Text Classification

The problem of classification has been widely studied in the database, data mining, and information retrieval communities. The problem of classification is defined in [4] as follows: we have a set of training records D = X1, . . . , XN, such that each record is labeled with a class value drawn from a set of k different discrete values indexed by 1 . . . k. The training data is used in order to construct a classification model, which relates the features in the underlying record to one of the class labels. For a given test instance for which the class is unknown, the training model is used to predict a class label for this instance. In the hard version of the classification problem, a particular label is explicitly assigned to the instance, whereas in the soft version of the classification problem, a probability value is assigned to the test instance [4]. Other variations of the classification problem allow ranking of different class choices for a test instance, or allow the assignment of multiple labels to a test instance.[5]

1.2 Multi-Label Text Classification

Three different paradigm exists under text classification and they are single label (binary) , multiclass and multi label[6]. Under single label a new text document belongs to exactly one of two given classes, in multiclass case a new text document belongs to just one class of a set of m classes and under multi label text classification scheme each document may belong to several classes simultaneously [7]. In real practice many approaches exists for the binary case and multi class case even though in many applications text documents are inherently multi label in nature. Eg. In the process of classification of online news article, the news stories can belong to classes like sports, politics, country, etc. Multilabel text classification problem refers to the scenario in which a text document can be assigned to more than one classes simultaneously during the process of classification. It has attracted significant attention from researchers and plays a crucial role in many applications such as web page classification, classification of news articles, information retrieval etc.[6]

1.3 Topological Data Analysis

Topological Data Analysis (TDA) can broadly be described as a collection of data analysis methods that find structure in data. This includes: clustering, manifold estimation, non-linear dimension reduction, mode estimation, ridge estimation and persistent homology [8]. As the name suggests, these methods make use of topological ideas. Often, the term TDA is used narrowly to describe a particular method called persistent homology (discussed in 4.1). Considering a broader perspective, use the term TDA to refers to a large class of data analysis method that uses notions of shape and connectivity. Zomorodian [9] explains that topology allows the larger group of homeomorphisms that deform an object by stretching or shrinking. Topology classifies a shape according to its connectivity, such as its number of pieces, loops, or

presence of boundary. as been geometric methods for visualizing documents and information flow, e.g. based on differential geometry [9] . Zhu [2] also explains that as a branch of topological data analysis,persistent homology has the advantage of capturing novel invariant structural features of documents. Intuitively, persistent homology can identify clusters (0-th order holes), holes (1st order, as in our loopy curve), voids (2nd order holes, the inside of a balloon), and so on in a point cloud. Considering the importance of clustering today, the value of these higher order structures is tantalizing. Indeed, in the last few years persistent homology has found applications in data analysis, including neuroscience [10], bioinformatics [11], sensor networks [12, 13], medical imaging [14],shape analysis [15],and computer vision [16].

CHAPTER 2: TEXT CLASSIFICATION

The problem of classification has been widely studied in the database, data mining, and information retrieval communities[4]. A typical classification problem can be stated as follows: given a set of labelled examples belonging to two or more classes (training data), we classify a new test sample to a class with the highest similarity. Document retrieval, routing and filtering systems, can often be viewed as a two-class classification problem where a document is labelled as relevant or non-relevant.[17]

Text classification presents many challenges and difficulties. First, it is difficult to capture high-level semantics and abstract concepts of natural languages just from a few key words. For instance, there are many ways to represent similar concepts (e.g. agent, softbot, robot, or bot) and the same word can represent different meanings (e.g. bank can be either be related to a finance problem or a river)[17]. Furthermore, semantic analysis, which is a major step in designing a natural language information retrieval system, is not well understood, although there are some techniques that have been successfully applied to limited domains [18]. Second, high dimensionality (thousands of features) and variable length, content and quality are the characteristics of a huge number of documents on the Web. These place both efficiency and accuracy demands on classification systems.

A number of methods have been discussed in the literature for document classification. These include the naive Bayes Classifier [17], decision trees [19], nearest neighbour classifier [20], linear discriminant analysis (LDA) [21], logistic regression [22] and neural networks [22]. Lewis and Ringutte [19] compared their ProBayes method and

a decision tree classifier on two data sets (Reuters newswire benchmark and MUC-3) with different numbers of features. They showed that the maximum effectiveness was reached for both algorithms when the term (feature) selection was based on collection frequency and mutual information. Schutze et al. [22] have empirically analysed how feature selection affects the three statistical classification techniques (LDA, logistic regression and neural networks) for the routing (two class) problems. They used optimal term selection (X^2 measure) and latent semantic indexing (LSI) to reduce the number of features. Their experimental results showed that features based on LSI are more effective for techniques such as LDA and logistic regression, whereas neural network based classification performs well with both the feature selection methods. Method to be used for a particular classification problem, at times, depends heavily on the dataset.

The classification problem assumes categorical values for the labels, though it is also possible to use continuous values as labels. The latter is referred to as the regression modelling problem. The problem of text classification is closely related to that of classification of records with set-valued features [23]; however, this model assumes that only information about the presence or absence of words is used in a document. In reality, the frequency of words also plays a helpful role in the classification process, and the typical domain-size of text data (the entire lexicon size) is much greater than a typical set-valued classification problem. A broad survey of a wide variety of classification methods can be found in [24, 25], and a survey which is specific to the text domain may be found in [26]. A relative evaluation of different kinds of text classification methods can be found in [1, 27].

Aggarwal et al. [4] generalize various text classification applications into siz major domains, which are as follows:

- News filtering and Organization: Most of the news services today are electronic in nature in which a large volume of news articles are created very single day by the organizations. In such cases, it is difficult to organize the news articles manually. Therefore, automated methods can be very useful for news categorization in a variety of web portals [28]. This application is also referred to as text filtering.

- Pattern (Rule)-based Classifiers: In rule-based classifiers we determine the word patterns which are most likely to be related to the different classes. We construct a set of rules, in which the left hand side corresponds to a word pattern, and the right-hand side corresponds to a class label. These rules are used for the purposes of classification.

- SVM Classifiers: SVM Classifiers attempt to partition the data space with the use of linear or non-linear delineations between the different classes. The key in such classifiers is to determine the optimal boundaries between the different classes and use them for the purposes of classification.

- Neural Network Classifiers: Neural networks are used in a wide variety of domains for the purposes of classification. In the context of text data, the main difference for neural network classifiers is to adapt these classifiers with the use of word features. We note that neural network classifiers are related to SVM classifiers; indeed, they both are in the category of discriminative classifiers, which are in contrast with the generative classifiers [29].

- Bayesian (Generative) Classifiers: In Bayesian classifiers (also called generative classifiers), the attempt is to build a probabilistic classifier based on modeling the underlying word features in different classes. The idea is then to classify text based on the posterior probability of the documents belonging to the different classes on the basis of the word presence in the documents.

- Other Classifiers: Almost all classifiers can be adapted to the case of text data. Some of the other classifiers include nearest neighbor classifiers, and genetic algorithm-based classifiers. We will discuss some of these different classifiers in some detail and their use for the case of text data.

CHAPTER 3: MULTI-LABEL TEXT CLASSIFICATION

Three different paradigm exists under text classification and they are single label (binary) , multiclass and multi label [6]. Under single label a new text document belongs to exactly one of two given classes, in multiclass case a new text document belongs to just one class of a set of m classes and under multi label text classification scheme each document may belong to several classes simultaneously [7]. In real practice many approaches are exists and proposed for binary case and multi class case even though in many applications text documents are inherently multi label in nature. Eg. In the process of classification of online news article, the news stories can belong to classes like sports, politics, country, etc. [6]

Multilabel text classification problem refers to the scenario in which a text document can be assigned to more than one classes simultaneously during the process of classification. Mark Berger [30] explains that broadly, multi-label classification is the task of assigning a set of labels from a fixed vocabulary to an instance of data. For multi-label text classification, this often involves labelling a piece of text with a set of tags. Since each document has an indeterminate number of labels, the task is significantly harder than multi-class classification. Where as multi-class classification has $\mid C \mid$ potential choices, a multi-label classification problem has $2 \mid L \mid$ possible outputs.

Dharmadhikari et al. [6] outlines between different supervised, unsupervised and semi-supervised learning methods for Multi label classifiers. In supervised methods only labelled text data is needed for training. Unsupervised methods relies heavily on only unlabelled text documents; whereas semi supervised methods can effectively use

unlabelled data in addition to the labelled data [31, 32]. The traditional approach towards multi-label learning either decomposes the classification task into multiple independent binary classification tasks or identifies rank to find relevant set of classes. But these methods do not exploit relationship among class labels. Few popular existing methods are binary relevance method, label power set method, pruned sets method, C4.5, Adaboost.MH & Adaboost.MR, ML-kNN, Classifier chains method etc. [33]. However, these are all based on supervised learning methods.

While designing a multi label text classifier the major objective is not only to identify the set of classes belonging to given new text documents but also to identify most relevant out of them to improve accuracy of overall classification process. Graph based approaches are known for their effective exploration of document representation and semi supervised methods explores both labelled and unlabelled data for classification that's why accuracy of multi label text classifier can be improved by using graph based representation of input documents in conjunction with label propagation approach of semi supervised learning [34, 35].

Multi Label text classification has attracted significant attention from lot of researchers for playing crucial role in many applications such as web page classification, classification of news articles, information retrieval etc. Generally supervised methods from machine learning are mainly used for realization of multi label text classification. But as it needs labelled data for classification all the time, semi supervised methods are used now a day in multi label text classifier. Many approaches are preferred to implement multi label text classifier [6]. [30] is proposing label propagation approach for multi label text classifier, it uses existing label information for identifying labels of unlabelled documents. They represent input text document corpus in the form of graph to exploit the ambiguity among different text documents. The ambiguity is

represented in the form of similarity measures as a weighted edge between text documents. [30] use word embeddings with CNN and RNN to improve the multi label classification over normal CNN and RNN.

Nowadays, we notice that multi-label classification methods are increasingly required by modern applications, such as protein function classification [36], music categorization [37] and semantic scene classification [38]. In semantic scene classification, a photograph can belong to more than one conceptual class, such as sunsets and beaches at the same time. Similarly, in music categorization a song may belong to more than one genre. For example, several hit songs of the popular rock band Scorpions can be characterized as both rock and ballad. One of the most common one being, movie categorization as usually movies belong to more than one genres.

CHAPTER 4: TOPOLOGICAL DATA ANALYSIS

Wasserman [8] mentions Topological Data Analysis (TDA) can broadly be described as a collection of data analysis methods that find structure in data. This includes: clustering, manifold estimation, non-linear dimension reduction, mode estimation, ridge estimation and persistent homology. Topological Data Analysis (TDA) refers to statistical methods that find structure in data. Often, the term TDA is used narrowly to describe a particular method called persistent homology.

4.1 Persistent Homology

Persistent homology is a multi-scale approach to quantifying topological features in data [39, 40, 41]. This is the branch of TDA that gets the most attention and some researchers view TDA and persistent homology as synonymous. Persistent homology finds "holes" by identifying equivalent cycles [2].

Singh et al. [10] explains the concept of TDA with an example. Consider a world where objects are made of elastic rubber. Two objects are considered equivalent if they can be deformed into each other without tearing the material. If such a transformation between X and Y exists, we say they are topologically equivalent and write X $\sim$ Y; otherwise, we write X Y. This notion of equivalence is illustrated in 4.1, where the reader is invited to mentally visualize the possible transformations between the various objects to verify the stated equivalence relationships. From this exercise, it is evident that a possible reason for two objects not to be equivalent is that they differ in the number of holes. Thus, simply counting holes can provide a signature

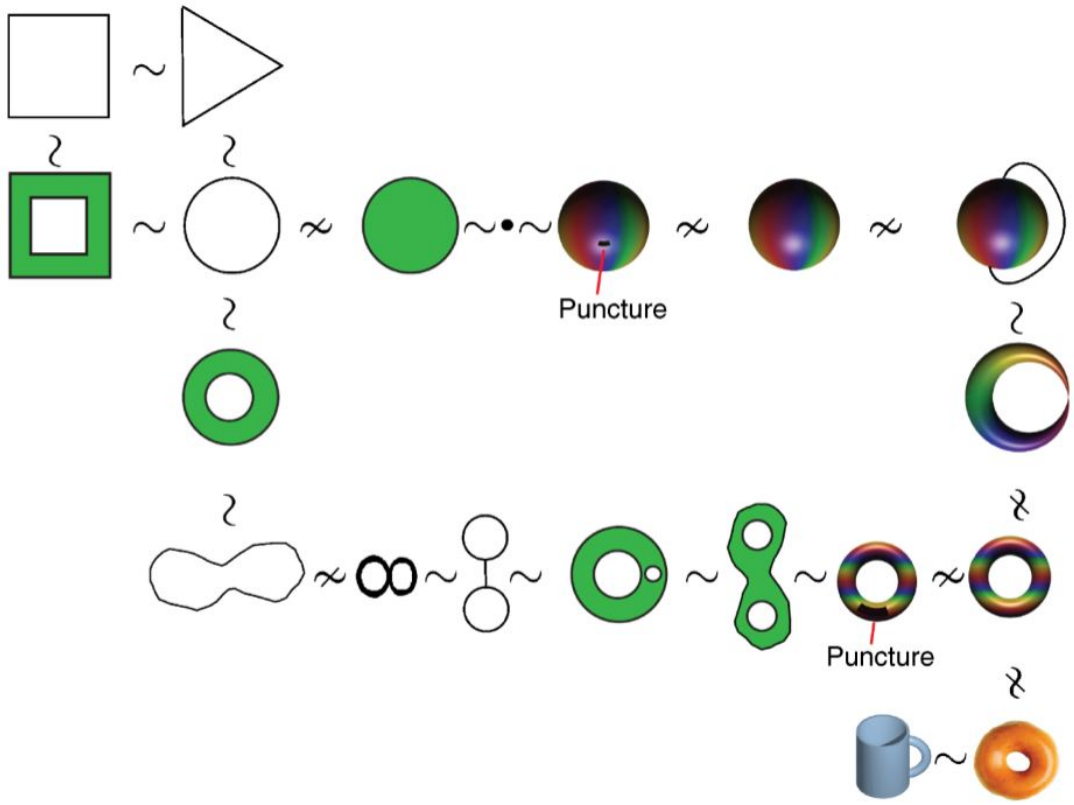


Figure 4.1: Topological equivalence in rubber-world
The figure illustrates the notion of equivalence by showing several objects (topological spaces) connected by the symbols when they are equivalent or by when they are not. The reader should think that all the objects shown are made of an elastic material and visualize the equivalence of two spaces by imagining a deformation between to objects. Source: Singh et al. [10]

for the object at hand. Holes can exist in different dimensions. A one-dimensional hole is exposed when a one-dimensional loop (a closed curve) on the object cannot be deformed into a single point without tearing the loop. If two such loops can be deformed into one another they define the same hole, which should be counted only once. Analogous definitions can be invoked in higher dimensions. For example, a two-dimensional hole is revealed when a closed two-dimensional oriented surface on the object cannot be deformed into a single point (4.2).

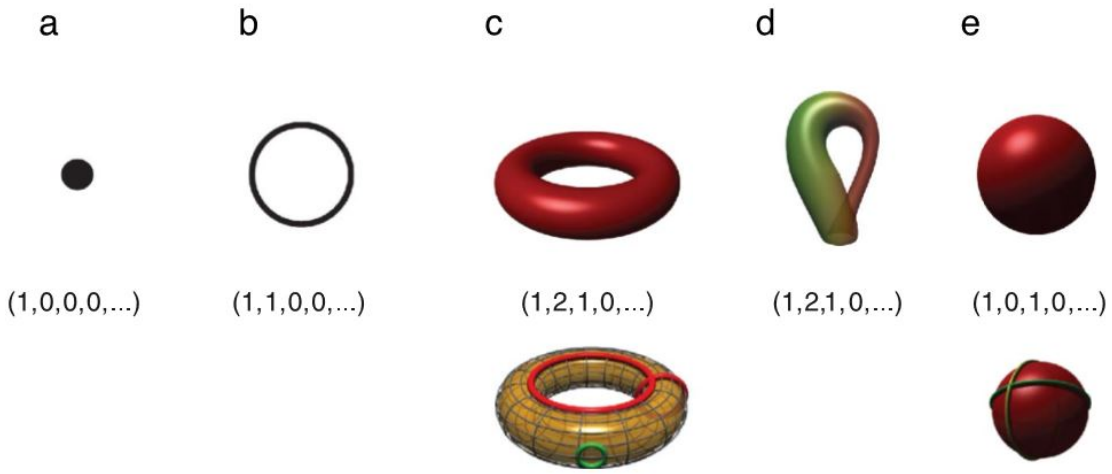


Figure 4.2: Betti numbers provide a signature of the underlying topology. Illustrated in the figure are five simple objects (topological spaces) together with their Betti number signatures: (a) a point, (b) a circle, (c) a hollow torus, (d) a Klein bottle, and (e) a hollow sphere. For the case of the torus (c), the figure shows three loops on its surface. The red loops are "essential" in that they cannot be shrunk to a point, nor can they be deformed one into the other without tearing the loop. The green loop, on the other hand, can be deformed to a point without any obstruction. For the torus, therefore, we have b1 = 2. For the case of the sphere, the loops shown (and actually all loops on the sphere) can be contracted to points, which is reflected by the fact that b1 = 0. Both the sphere and the torus have b2 = 1, this is due to the fact both surfaces enclose a part of space (a void). Source: Singh et al. [10]

4.2 Betti Numbers

The notion of counting holes of different dimensions is formalized by the definition of Betti numbers. The Betti numbers of an object X can be arranged in a sequence, b(X)=(b0, b1, b2, I), where b0 represents the number of connected components, b1 represents the number of one dimensional holes, b2 the number of two-dimensional holes, and so forth. An important property of Betti sequences is that if two objects are topologically equivalent (they can be deformed into each other) they share the same Betti sequence. A single point has a Betti sequence equal to (1, 0, 0, ...) since it has only one component and no holes of any dimension (Figure 2a). A circle has a Betti sequence (1, 1, 0, 0,...), as there is a single connected component and there is a single loop that cannot be deformed into a single point (Figure 2b). A torus has

a Betti sequence (1, 2, 1, 0, ...), since it has a single connected component, two different loops that cannot be deformed into a point (shown in red in the bottom panel of Figure 2c), and there is a two-dimensional surface that cannot be deformed into a point (shown in orange in Figure 2c). The Klein bottle has the same sequence as the torus (1, 2, 1, 0, ...). This shows that while two objects that are equivalent must have the same Betti sequences, two objects that are not equivalent do not necessarily have different sequences. Finally, a sphere has a sequence (1, 0, 1, 0, ...), as any one-dimensional loop on its surface can be deformed into a point. The Betti sequence therefore provides a signature (albeit not unique) of the underlying topology of the object.

In general, β_p is the number of independent p^{th} holes. For example, a tetrahedron has $\beta_0 = 1$ since the shape is connected, $\beta_1 = \beta_2 = 0$ since there is no holes or voids. A hollow tetrahedron has $\beta_0 = 1$, $\beta_1 = 0$, $\beta_2 = 1$ because of the void. Further removing the four triangle faces but keeping the six edges, the skeleton has $\beta_0 = 1$, $\beta_1 = 3$ (there are 4 triangular holes but one is the sum of the other three), $\beta_2 = 0$ (no more void). Finally removing the edges but keeping the four vertices,$\beta_0 = 4$ (4 connected components each a single vertex) and $\beta_1 = \beta_2 = 0$.

4.3 Rips Complex and Barcode

To estimate the Betti numbers for a given object we generate the Rips complexes, based on a given distance function d(x, y) between any two points. To build the Rips complex, we begin by selecting a fixed parameter ϵ and we proceed to connect all points for which d(x, y)$< \epsilon$ with edges, all triplets for which all pairwise distances are smaller than ϵ with triangles, all quadruplets for which all pairwise distances are smaller than ϵ with tetrahedra, and so on. The Betti numbers are then computed based on the Rips complexes at different values of ϵ. The parameter ϵ effectively controls the "spatial scale" of analysis [10].

The basic idea of the method is to track the different "holes" across different spatial scales of analysis. We visualize the results of the analysis by plotting "birth" and "death" intervals of individual holes of different dimensions as the spatial scale ϵ goes from zero to infinity. For each Betti number, we keep a separate graph. Connected components are drawn as horizontal lines in the b0 graph, one-dimensional holes correspond to horizontal lines in the b1 graph, two-dimensional holes in the b2 graph, and so on. For each hole, the horizontal line has its endpoints at the values of ϵ at which the structure was first created and then destroyed. The set of all these lines together is called a barcode.

The x-axis is ϵ. Each horizontal bar represents the birth-death of a separate homology class. Longer bars correspond to more robust topological structure in the data [2]. An example of the computation of the Rips complex at various levels of ϵ and the corresponding barcode is shown in 4.3 by [10] . The data are randomly sampled points from a torus. In each panel, the left three graphs show the barcode obtained from this calculation. The graph on the top corresponds to b0, meaning that each horizontal line represents a different connected component; the middle graph corresponds to b1, where each horizontal line corresponds to a one-dimensional loop; the bottom graph corresponds to b2, where horizontal lines represent two dimensional holes. The illustrations to the right of the barcode show the state of the Rips complex for the selected value of ϵ, which is indicated by the red vertical bars in the barcodes. For a small value of ϵ, only one edge exists and the resulting structure has many different connected components and no holes of any dimension (Figure 3a, top panels). The Betti sequence for this value of ϵ can be recovered by counting how many horizontal lines (corresponding to different holes) the red vertical line crosses in each of the graphs. The resulting sequence is (b0, b1, b2)= (50, 0, 0). At a higher value of ϵ, we

see more edges being added (thereby reducing the number of connected components) but still no holes of any dimensions (Figure 3b). The corresponding sequence is (b0, b1, b2) = (38, 0, 0). At the next higher value of ϵ, we finally obtain a single connected component (Figure 3c). The vertical bar in the graph corresponding to b1 crosses two horizontal lines, meaning that there are two one-dimensional holes. However, the red line does not intersect any horizontal lines in the graph for b2 meaning that there are no two dimensional holes at this scale (this is because some of the triangles still need to be filled-in). The Betti sequence for this value of ϵ is then (b0, b1, b2) = (1, 2, 0). Finally, at a slightly higher value of ϵ, the correct signature of the torus emerges (b0, b1, b2) = (1, 2, 1) (Figure 3d). This Betti sequence then persists for a long interval of ϵ.

4.4 Topological Data Analysis for Textual Data

Zhu [2] presents one of the first applications of persistent homology for natural language processing. His Similarity Filtration with Time Skeleton (SIFTS) algorithm identifies holes that can be interpreted as semantic "tie-backs" in a text document, providing a new document structure representation. A brief overview of the approach is explained by [2]: imagine dividing a document into smaller units such as paragraphs. A paragraph can be represented by a point in some space, for example, as the bag-of-words vector in R^d where d is the vocabulary size. All paragraphs in the document form a point cloud in this space. Now let us "connect the dots" by linking the point for the first paragraph to the second, the second to the third, and so on. What does the curve look like? Certain structures of the curve capture information relevant to Natural Language Processing (NLP). For instance, a good essay may have a conclusion paragraph that "ties back" to the introduction paragraph. Thus the starting point and the ending point of the curve may be close in the space. If we further connect all points within some small ϵ diameter, the curve may become a loop with a hole in

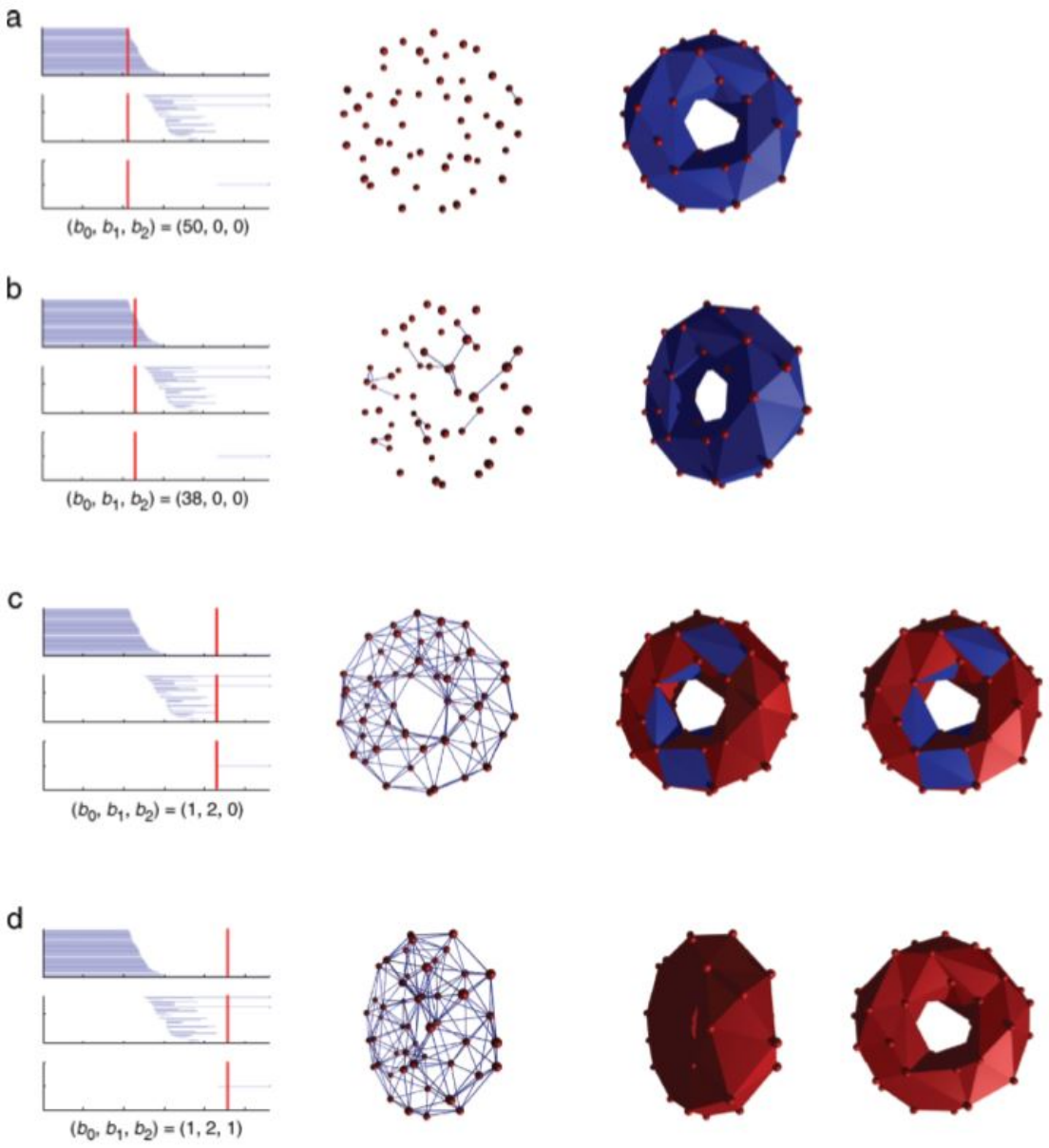


Figure 4.3: Rips Complex and corresponding Bacodes
The figure illustrates the construction of the Rips complex and the generation of barcodes (only the first three Betti numbers are displayed) for 50 points randomly sampled from the surface of a torus. Panels a to show the barcode "sliced" at different values of ϵ (the horizontal axis) with the corresponding Rips complexes shown to the right. The corresponding Betti numbers for each level of ϵ can be obtained by counting the number of horizontal lines crossed by the vertical red line in each graph. Source: Singh et al. [10]

the middle. In contrast, an essay without any tying back may not contain holes, no matter how large ϵ is. This approach of [2] forms the basis of my thesis as I go ahead with using Zhu's code to a IMDB dataset, trying to solve the problem of multi label text classification using TDA.

Before we go ahead, here's a basic summary of TDA that can be useful to keep up with the rest of the work:

- TDA is a relatively recent domain of data analysis and applied mathematics. It focuses on the study of the data sets that have geometric structure (possibly in high dimensions). The premise of TDA is that this geometry if properly recovered should help with typical tasks of data and predictive analytics.

- TDA has been widely used in many data analysis applications, for example in financial and health care domains, as we well in research areas ranging from archaeology to genetics. [1].

- One of its key tools is *persistent homology* which tries to find geometric patterns such as clusters and holes in different dimensions and different resolutions, simultaneously.

- The result of this analysis is represented as *barcodes* or equivalently *persistence diagrams*.

- For machine learning tasks such as classification, these can be viewed as additional numerical features.

- The specific technique used in this paper is called Similarity Filtration with Time Skeleton (SIFTS). This "algorithm identifies holes that can be interpreted as semantic "tie-backs" in a text document, providing a new document structural representation." [2]

[1] `https://www.ayasdi.com/blog/` has examples of many such applications and relevant references

- Background information about SIFTS, and *the software itself is available* at Prof.J.Zhu's website.[2] The background info also includes "a gentle tutorial on homology, and an application in machine learning."

- A lot of the above material is derived from [10]. I highly recommend going through their paper to gain more in-depth knowledge of TDA.

[2] `http://pages.cs.wisc.edu/~jerryzhu/publications.html`

CHAPTER 5: EXPERIMENT

In this section I explain my experiment and the data preparation process. I evaluate whether topological features are effective in text classification by predicting movie genres based on plot descriptions (as in the example below). I perform multiple preprocessing steps before generating barcodes, as explained in subsections below. I also explain the initial experiments I performed to get a better understanding of the procedure, and this I hope might help the reader to follow up the paper.

Table 5.1 outlines my entire experiment procedure. The dataset used for the experiment is the IMDB dataset. I explain Data Preparation in detail in the section 5.2. The steps mentioned in the table are simplified to get a clear understanding of the experiment procedure. I perform a 4-step text preprocessing and use Lancaster Stemming to avoid writing style bias (i.e. make different forms of the same word like 'jump' and 'jumping' equivalent). Once I create the vector matrices, I use Javaplex with Matlab to generate the persistence diagrams i.e. barcodes (shown in the next section)[1] . Section 5.3 explains the Data Distribution and Results.

5.1 Experiment

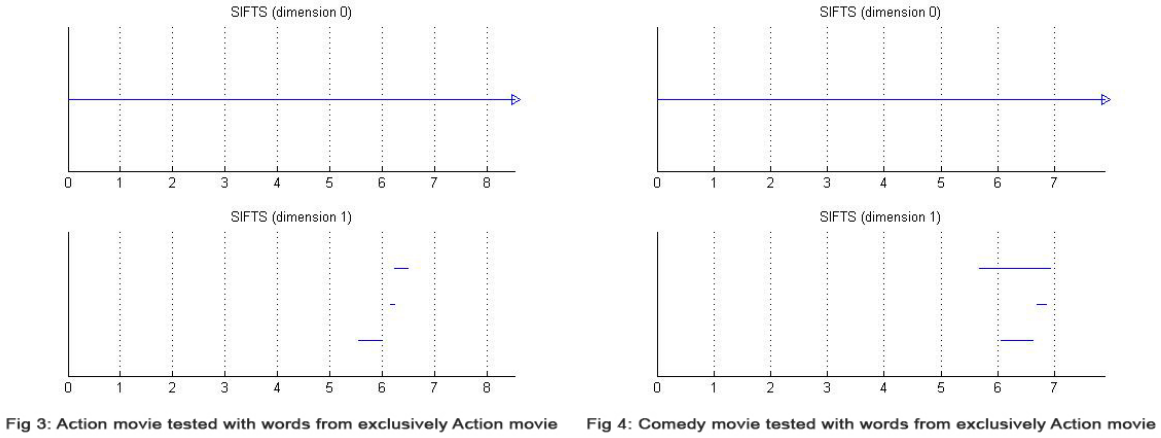
In this section I explain my initial experiment and lay the foundation for the final experiment.

Central idea of [2] experiment was that persistence diagrams can outline the strength

[1] `TauszA,Vejdemo-JohanssonM,AdamsH.Javaplex:Aresearchsoftwarepackageforpersistent(co)homology.Softwareavailableathttp://code.google.com/javaplex.`

Table 5.1: Full description of the experiment

Task	Identifying genres of movie plots
Data Used	Movie plot summary; Movie Genre from IMDB Dataset.
Data Preparation	1. Stop words removal using NLTK 2. Removing numeric tokens using Python RegEx 3. Removing Punctuation marks 4. Using Lancaster Stemming to reduce tokens to their base form
Testing Data Conditions	1. Plot should contain >100 tokens 2. Plot should contain >3 sentences 3. Plot should belong to >1 genres
Steps	1. Perform Data Preparation for all the movie plots in dataset. 2. Create four lists of top words; one for each genre; using TF-IDF. a. Separate movies belonging to only one genre b. Using TF-IDF identify top words. c. Filter each list to remove common nouns like film; depicts; etc. 3. Select the plot satisfying 'Testing Data Conditions'. 4. Generate Term Frequency Matrix; one for each list of top words. a. Columns of the matrix represent the tokens from the list of top words b. Rows represent each sentence in the plot c. In a $X^{action}_{M\times N}$ matrix; M represents number of tokens in top words list of Action genre. N represents the number of sentences in movie plot X. $X_{I\times J}$ gives the frequency of token I in sentence J 5. Using Javaplex for Matlab; generate persistent homology diagrams (barcodes) using each Term Frequency Matric as input; one at a time. 6. Identify genres of a movie plot by comparing the number of 1-dimensional holes/loop in each barcode for the respective plot.
Results	Hit Rate: 83.3% Jaccard Index: 54.8% 1. Hit Rate: Proportion of dataset where model predicted at least one genre correctly 2. Jaccard Index: Defined as the number of correctly predicted labels divided by the union of predicted and true labels $\frac{\mid T \cap P \mid}{\mid T \cup P \mid}$ 3. F-score: Defined as the harmonic mean of Precision and Recall $$\frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$
Data Distribution	1. For Identifying Top Words for each genre: Action Genre: 3286 movies | Comedy Genre: 3000 movies Horror Genre: 2176 movies | Romance Genre: 3500 movies 2. Testing Dataset: 250 movie plots.



(a) Barcode for Action movie plot (b) Barcode for Comedy movie plot

Figure 5.1: Comparing the barcodes in initial experiment
The existence of 1-dimensional holes/loops in a document is shown by the horizontal bars in the barcodes. The count of the horizontal bar would give the number of 1-dimensional holes in the document. Length of the bar determines the persistence of holes. More the number of bars and length of the bars gives the strength of similarity between rows of the vector for a particular genre. Thus, with the presence of 1-dimensional holes/loops we say the movie belongs to the respective genre.(a) and (b) both showcase same number of 1-dimensional loops/holes. However, we can see a clear distinction in the persistence of these holes. In the (a), the holes are not as persistent as in (b), if not more. The sentences of both the movie plots were converted to vector forms using the same set of words belonging to movies exclusively from Action genre. Hence, when the Comedy movie represents stronger bond between the vectors, it indicates the ambiguity in the approach followed for generating vectors.

of the main idea of a text. Following this, in an initial experiment, I compare the barcodes for two movie plots (one Action genre and one Comedy genre), from the IMDB's website, in my experiment to identify the movie plot. I use the [42] dataset to identify the top words from the Action genre using the TF-IDF measure. Using these top words, I generate Term Frequency matrix for both the movie plots, as explained in section 5.2. Understanding that an Action movie plot would have more words common with the top words list and hence the Action movie vector would present stronger looping between the sentences.

The barcodes in Figure5.1 indicated three major issue in the first experiments: 1.

Not considering overlapping genres 2. Writing Style of the text 3. No use of semantics. Movies usually belong to more than one genres and not considering this while classifying them could result in sub-par results. I used the top words for vectors from [42] dataset while tested on an IMDB's movie plot. To account for change of writing styles and implicitly for semantics, I simply use the Lancaster stemming in my final experiment. This is clearly a very simple solution, and others, e.g. based on word2vec-like solutions will perhaps be more appropriate for future work to produce better semantic representations.

5.2 Data preparation and generation of topological signatures

As I performed some basic preprocessing on the text and used NLTK's stopwords list for removing the stopwords. I also remove the punctuation marks and numerical characters from the sentences. Next, I separate each sentence from the text into a new line, thereby letting me handle each sentence individually. To make the experiment neutral to writing styles, I use Lancaster Stemming and reduce each token to its base form. After the data preparation, next is creating the Term Frequency Matrix for each genre in consideration per movie plot. Hence, each column in the matrix represented the top words from the respective genre and each row was the vector representation of the respective line from the text. I create this matrix for every movie plot that satisfied 'Testing Data Conditions'.

5.3 Experiment and Results

Data Selection: From the dataset I filtered out the details regarding TV episodes and short/documentary as that would not be inline with the problem statement. With the movies, I collect the plots, genres and a movie id, for uniquely identifying each movie in the database. As my major comparison is between the movies from genre

Action, Romance, Comedy and Horror, I segregate the movie belonging exclusively to only these four genres i.e. no overlap. Thus I got 3286 movies belonging to only action genre, 3500 movies belonging to only comedy genre, 2176 movies belonging to only horror genre and 18000 movies belonging to comedy genre of which I randomly select 3000 movies belonging to comedy genre, to avoid the problem of over fitting.

Data Preparation: I preprocess all the movies belonging one of the four genres, individually. Thereby removing the stop words, punctuation marks, numeric tokens, stemming to reduce tokens to their same base forms. Next, I use TF-IDF to get the top 1000 words from each genre. With the four lists of 1000 top tokens of each genre, I perform subtraction of list to remove the common nouns from each list, as nouns like 'second', 'depict', etc added noise resulting in incorrect final outputs. After manually removing the common nouns left after subtraction, I'm left with top words belonging to each genres without any noise, each list is approximately 200 tokens long. I use these lists of top tokens for creating sentence vectors for the test documents.

Selection of Testing Data: While picking movies for testing, I found that one movie could have more than one plot summaries, provided by different authors. Hence, database had multiple entries for a movie with same name, however with a unique movie id, for each different plot summary. Hence, for each movie plot, I first assign the longest movie plot provided for each movie. For testing, I randomly pick 250 movies from the database satisfying the basic criteria, i.e. belonging to more than one genres, having more than 100 tokens in the plot and having at least 4 sentences. I found 66000 movies satisfying the above criteria and I randomly picked 250 movies, keeping no preference for genre combinations or length of the plots.

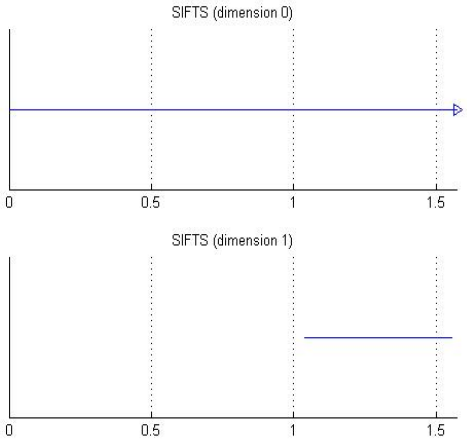
Testing: I want to find the 1-dimension holes across the sentences. Using the vec-

tors generated with the top words from each genre as explained above, I run them through Javaplex for generating the homology complexes. I compare the four 1-dimension homology complexes generated for each movie plot to determine the genre for the respective movie. Using the barcode representation of the 1-dimension homology complexes, the program is able to correctly identify the genres of 208 movies with overlapping genres, giving a hit rate of 0.8333%. The output was considered correct if I were able to correctly predict at least one genre if a movie belonged to n genres. As hit rate is considered a weak metric, I calculate Jaccard index and F-score using the formula mention in Table 5.1. I obtained a Jaccard index of 54.8% and an F-score of 71.88%.

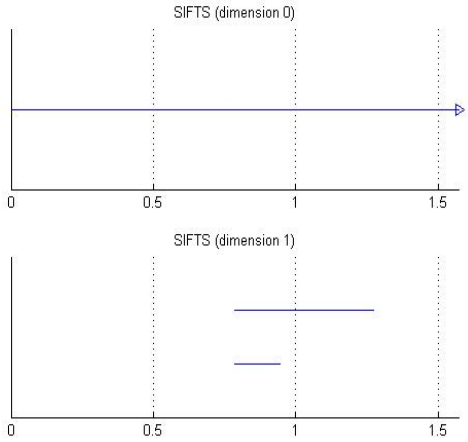
In Figure 5.2 I present an example of the output. Here, I consider a movie plot form the IMDB dataset which is listed as Comedy and Romance genre. I, generate the Term Frequency Matrix and run TDA to generate the barcodes presented in Figure 5.2. Plot of the movie tested is highlighted below.

Example Movie Plot: *Al Bennett and Alice Cook are blissfully happy and are building their house in an isolated area; using Al's savings of $7;000. Troubles begin when Al's uncle and boss; Uncle George; unaware of Al's engagement; inspects the house and announces he will live with him. Furthermore; he wants Al to marry Minnie Spring; a wonderful girl George just met; so the three of them can live happily ever after. Troubles mount when Alice's parents come to see the partially built house. They bring with them eleven other family members; some of whom expect to live with the couple when they marry. Badgered on all sides; Al finally yells "There's too many Cooks." Alice breaks their engagement; and George fires him. As if these problems weren't enough; the Carpenter's Union calls a strike; but a determined Al decides to finish the house himself. Months later; lonely and depressed; Al puts his finished*

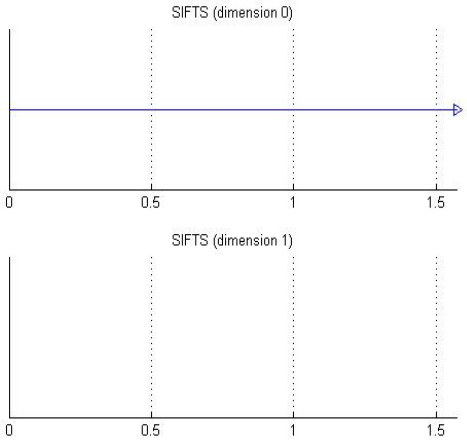
house up for sale; but things are looking up for Al.



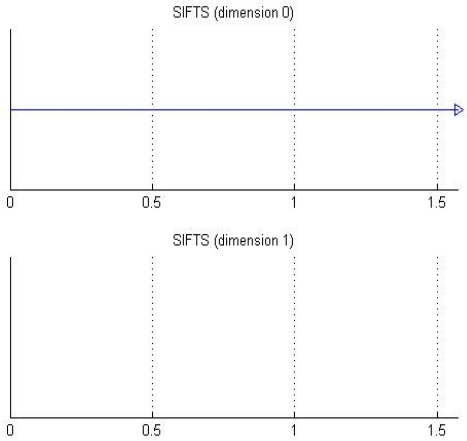
(a) Barcode for matrix using Comedy words



(b) Barcode for matrix using Romance words



(c) Barcode for matrix using Action words



(d) Barcode for matrix using Horror words

Figure 5.2: Comparing four barcodes

Comparing the four barcodes, we can see that (a) and (b) have persistent 1-dimensional holes whereas (c) and (d) have none. For (a) and (b), top words from Comedy genre and Romance genre, respectively, were used to represent the columns for the Term Frequency Matrix. For (c) and (d), top words from Action genre and Romance genre, respectively, were used to represent the columns in the Term Frequency Matrix. This clearly shows the inclination of the movie plot in question, towards Comedy and Romance genre, which are the original genres for the movie in IMDB dataset.

CHAPTER 6: RESULTS AND CONCLUSIONS

TDA, although a very active area of research, as mentioned above, is very rarely applied to textual data. The main reference in this space is [2], which can also serve as an excellent introduction to the field. Since the paper is accompanied by software, it was relatively painless for me to run the experiments described above. The works cited here might be (as of Jan 2018) complete or close to complete list of papers using persistent homology for text processing (and not just mentioning it as a possibility). I briefly discuss them now.

There is substantial amount of work on classification of movie plots, as the problem appears in movie recommendations. A recent article by [1] is closest to my work, as it is focused on the same task of predicting movie genres based on plots,and contains state of the art results. The author reports results of experiments with more than 250,000 movies using neural networks for classification. The produced model achieves the Jaccard-score of 50%, the F-score of 56% and the hit rate of 80.5%.

While my results look better, this is not exactly an apples-to-apples comparison: while Hoang performs the experiment on a very large data set, my test is limited to a sample of 250 movies. On the other hand, this limitation is also the strength of the method. Namely, I only need to train on approximately 1% of the data to obtain a very good performance.

Secondly, I perform my experiment on 4 most popular genres i.e. Action, Comedy, Horror and Romance, while [1] also considers not-so popular genres like Adventure,

Sport, Mystery, Family. I believe, the F-score for TDA approach would be higher when compared to [1] result for the same four genres, since I have better precision and recall. However I estimate the difference in the F-score of 15.88% might come down to 7% since [1] model performed better for popular genres compared to other genres. Therefore I do not want to emphasize the F-score improvements.

I also observe that in most cases, model predicted 'Action' genre as well when the plot belonged to 'Romance' genre and vice-versa. The major reason for this is a lot of common words between the genres in the list of top words. Although the top 10 words in both the genres are completely different. I was only able to incorporate the presence of words not the priority of words in each genre. Incorporating the priority of each word could have a positive impact on the results.

However, I want to bring the reader's attention to the fact that my 250 movies were explicitly chosen to belong to overlapping genres. The reason for that choice was that (a) my initial experiment on a few dozen movies showed 100% accuracy of TDA on movies with no overlapping genres; (b) I decided in late 2017 to run the experiment on overlapping categories, which arguably can be viewed as a more difficult problem; (c) I only became aware of the results in [1] in late Jan. of 2018. Unfortunately, I couldn't access the code for [1] experiment to reproduce the results and do a more suitable comparison.

Although 250 serves as a good sample set to compare the results, next step would be to test TDA on a larger dataset that would allow us remove the limitations of this comparison. One of the issues here is that for a larger set we need to change the set of TDA tools to a more efficient program than used in the reported experiment;

There are only a few other examples of application of TDA to text. Michel et

al. [3] perform sentiment classification on both sentence level, on the Cornell Sentence Polarity (the CSP-corpus of [43]), and on IMDB movie reviews (following [44]). They conclude "using persistence diagrams for text representation does not seem to positively contribute to document clustering and sentiment classification tasks". Although they leave open the possibility of topological features contributing to other NLP tasks such a parsing. On the other hand [45] show that topological features can improve extraction of multiword expressions and in document summarization.

Finally, I want to mention [46] who argue for the applicability of TDA to visualization of texts; this is an important issue, but somewhat orthogonal to the tasks discussed in this paper.

In the reported experiment, I was able to show the ability of Topological Data Analysis (TDA) to perform text classification. TDA not only matches the performance of widely used algorithms like Multinomial Naive Bayes, Logistic Regression for binary text classification, but also can also outperform more advanced techniques like neural networks when in multi-label text classification.

On the task of classification of movies according to four most common genres, I obtain Jaccard score of 54.8%, F-score of 71.88%, and hit rate of 83.3%. This is a significant improvement in 2 of the 3 measures over recently reported results of [1]: (+4.7%,+15.88%, and +0.5%)-respectively. As noted above, one limitation of my work is that the testing has only been performed on a (random) sample of 250 movies, moreover I limited myself to four categories. However, I also choose – on purpose – the data consisting of the movies which belong to multiple categories, and thus made the task harder. (And this harder task was my initial objective, as I found out about [1] after most of the experiments were done).

Clearly, the main message of this exercise is that TDA can be seriously used as a tool for discourse classification, notwithstanding mixed results of other experiments. Topology can be a source of useful features. And while this work focused on low dimensional persistence, higher dimensional topological features can be a source of additional insights by representing more complex repeating patterns, even though they take more time to compute.

REFERENCES

[1] Q. Hoang, "Predicting movie genres based on plot summaries," *arXiv preprint arXiv:1801.04813*, 2018.

[2] X. Zhu, "Persistent homology: An introduction and a new text representation for natural language processing.," in *IJCAI*, pp. 1953–1959, 2013.

[3] P. Michel, A. Ravichander, and S. Rijhwani, "Does the geometry of word embeddings help document classification? a case study on persistent homology based representations," *arXiv preprint arXiv:1705.10900*, 2017.

[4] C. C. Aggarwal and C. Zhai, *A Survey of Text Classification Algorithms*, pp. 163–222. Boston, MA: Springer US, 2012.

[5] "European bioinformatics institute, http://www.ebi.ac.uk/."

[6] S. C. Dharmadhikari, M. Ingle, and P. Kulkarni, "Towards multi label text classification through label propagation," *Editorial Preface*, vol. 3, no. 6, 2012.

[7] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining*, vol. 3, no. 3, 2006.

[8] L. Wasserman, "Topological data analysis," *Annual Review of Statistics and Its Application*, no. 0, 2016.

[9] A. Zomorodian, "Topological data analysis," *Advances in applied and computational topology*, vol. 70, pp. 1–39, 2012.

[10] G. Singh, F. Memoli, T. Ishkhanov, G. Sapiro, G. Carlsson, and D. L. Ringach, "Topological analysis of population activity in visual cortex," *Journal of vision*, vol. 8, no. 8, pp. 11–11, 2008.

[11] P. M. Kasson, A. Zomorodian, S. Park, N. Singhal, L. J. Guibas, and V. S. Pande, "Persistent voids: a new structural metric for membrane fusion," *Bioinformatics*, vol. 23, no. 14, pp. 1753–1759, 2007.

[12] V. De Silva, R. Ghrist, *et al.*, "Coverage in sensor networks via persistent homology," *Algebraic & Geometric Topology*, vol. 7, no. 1, pp. 339–358, 2007.

[13] V. De Silva and R. Ghrist, "Homological sensor networks," *Notices of the American mathematical society*, vol. 54, no. 1, 2007.

[14] M. K. Chung, P. Bubenik, and P. T. Kim, "Persistence diagrams of cortical surface data," in *International Conference on Information Processing in Medical Imaging*, pp. 386–397, Springer, 2009.

[15] J. Gamble and G. Heo, "Exploring uses of persistent homology for statistical analysis of landmark-based shape data," *Journal of Multivariate Analysis*, vol. 101, no. 9, pp. 2184–2199, 2010.

[16] D. Freedman and C. Chen, "Algebraic topology for computer vision," *Computer Vision*, pp. 239–268, 2009.

[17] Y. H. Li and A. K. Jain, "Classification of text documents," *The Computer Journal*, vol. 41, no. 8, pp. 537–546, 1998.

[18] D. Oard, "A survey of information retrieval and filtering methods," *Univ. of Maryland Institute for Advanced Computer Studies Report, College Park*, 1995.

[19] D. D. Lewis and M. Ringuette, "A comparison of two learning algorithms for text categorization," in *Third annual symposium on document analysis and information retrieval*, vol. 33, pp. 81–93, 1994.

[20] S. Weiss, S. Kasif, and E. Brill, "Text classification in usenet newsgroups: A progress report," in *Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access*, pp. 125–127, 1996.

[21] D. Hull, J. Pedersen, and H. Schutze, "Document routing as statistical classification," in *AAAI Spring Symposium on Machine Learning in Information Access*, vol. 12, pp. 49–54, 1996.

[22] H. Schütze, D. A. Hull, and J. O. Pedersen, "A comparison of classifiers and document representations for the routing problem," in *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 229–237, ACM, 1995.

[23] D. T.-H. Chang, Y.-Z. Weng, J.-H. Lin, M.-J. Hwang, and Y.-J. Oyang, "Protemot: prediction of protein binding sites with automatically extracted geometrical templates," *Nucleic acids research*, vol. 34, no. suppl_2, pp. W303–W309, 2006.

[24] D. Demner-Fushman, B. Few, S. E. Hauser, and G. Thoma, "Automatically identifying health outcome information in medline records," *Journal of the American Medical Informatics Association*, vol. 13, no. 1, pp. 52–60, 2006.

[25] B. Gu, "Recognizing nested named entities in genia corpus," in *Proceedings of the Workshop on Linking Natural Language Processing and Biology: Towards Deeper Biological Literature Analysis*, pp. 112–113, Association for Computational Linguistics, 2006.

[26] D. A. Lindberg, B. L. Humphreys, and A. T. McCray, "The unified medical language system," *Methods of information in medicine*, vol. 32, no. 04, pp. 281–291, 1993.

[27] "The national center for text mining, http://nactem.ac.uk/."

[28] K. Humphreys, G. Demetriou, and R. Gaizauskas, "Two applications of information extraction to biological science journal articles: Enzyme interactions and protein structures," in *Biocomputing 2000*, pp. 505–516, World Scientific, 1999.

[29] L. C. Lee, F. Horn, and F. E. Cohen, "Automatic extraction of protein point mutations using a graph bigram association," *PLoS computational biology*, vol. 3, no. 2, p. e16, 2007.

[30] M. J. Berger, "Large scale multi-label text classification with semantic word vectors," 2015.

[31] X. Zhu, "Semi-supervised learning literature survey," 2005.

[32] O. Chapelle, B. Schfolkopf, and A. Zien, "Semi-supervised learning2006, 03-08."

[33] S. C. Dharmadhikari, M. Ingle, and P. Kulkarni, "A comparative analysis of supervised multi-label text classification methods," 2011.

[34] T. Lansdall-Welfare, I. Flaounas, and N. Cristianini, "Scalable corpus annotation by graph construction and label propagation," 2012.

[35] P. P. Talukdar and F. Pereira, "Experiments in graph-based semi-supervised learning methods for class-instance acquisition," in *Proceedings of the 48th annual meeting of the association for computational linguistics*, pp. 1473–1481, Association for Computational Linguistics, 2010.

[36] M.-L. Zhang and Z.-H. Zhou, "A k-nearest neighbor based algorithm for multi-label classification," in *Granular Computing, 2005 IEEE International Conference on*, vol. 2, pp. 718–721, IEEE, 2005.

[37] T. Li and M. Ogihara, "Detecting emotion in music," 2003.

[38] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern recognition*, vol. 37, no. 9, pp. 1757–1771, 2004.

[39] H. Edelsbrunner and J. Harer, *Computational topology: an introduction*. American Mathematical Soc., 2010.

[40] H. Edelsbrunner and J. Harer, "Persistent homology-a survey," *Contemporary mathematics*, vol. 453, pp. 257–282, 2008.

[41] H. Edelsbrunner, D. Letscher, and A. Zomorodian, "Topological persistence and simplification," in *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pp. 454–463, IEEE, 2000.

[42] D. Bamman, B. O'Connor, and N. A. Smith, "Learning latent personas of film characters," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, p. 352, 2014.

[43] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Proceedings of the 43rd annual meeting on association for computational linguistics*, pp. 115–124, Association for Computational Linguistics, 2005.

[44] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pp. 142–150, Association for Computational Linguistics, 2011.

[45] H. Guan, W. Tang, H. Krim, J. Keiser, A. Rindos, and R. Sazdanovic, "A topological collapse for document summarization," in *Signal Processing Advances in Wireless Communications (SPAWC), 2016 IEEE 17th International Workshop on*, pp. 1–5, IEEE, 2016.

[46] I. R. Sami and K. Farrahi, "A simplified topological representation of text for local and global context," in *Proceedings of the 2017 ACM on Multimedia Conference*, pp. 1451–1456, ACM, 2017.