THE HIDDEN SHAPE OF DATA: TOPOLOGICAL DATA ANALYSIS FOR ANXIETY DETECTION IN TEXT

by

Morgan Byers

HONORS THESIS

Submitted to Texas State University
in partial fulfillment
of the requirements for
graduation in the Honors College
May 2021

Thesis Supervisor:

Vangelis Metsis

FAIR USE AND AUTHOR'S PERMISSION STATEMENT

Fair Use

Duplication Permission

DEDICATION

To my dad, who introduced me to programming. What began as a means to keep me entertained while you worked turned into my career. Thank you.

ACKNOWLEDGEMENTS

I would like to thank all who have helped me through this process, from the professors who believed in me to the friends who kept me grounded. I could never have completed this thesis without the unrelenting support of those around me.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Figure **Page**

ABSTRACT

A significant portion of the veteran population experiences post-traumatic stress disorder, or PTSD, as a result of their service. This is often accompanied by social anxiety disorder. In particular, student veterans are especially vulnerable as they struggle to integrate into a college lifestyle. In an effort to improve the support systems available for student veterans, we employ approaches from topological data analysis, an emerging area of research, to anxiety detection in text. Our models provide a tool to support psychologists and social workers in treating social anxiety. The results detailed in this paper could also have broader impacts in fields such as pedagogy and public health.

I. INTRODUCTION

Post-traumatic stress disorder, or PTSD, affects a significant proportion of the veteran population and is often accompanied by social anxiety disorder. About 8 - 10% of returning combat veterans have PTSD, and of those, another 7 - 13% also have social anxiety disorder [1]. In particular, student veterans can have difficulty transitioning to life on a college campus [2]. Student veterans often face difficulty adjusting to the lack of structure in college classes and can struggle to relate with their younger, less experienced peers [1]. This radical lifestyle shift from service to campus in conjunction with limited access to resources, may negatively impact a student veteran's quality of life [2].

In order to assist psychologists and social workers in diagnosing and treating social anxiety disorder, we explore several applications of machine learning to anxiety detection. We began with a set of ten interviews between a psychologist and student veterans. Those interviews were transcribed, and the veterans' responses were divided into 1,187 sentences. Then, each sentence was rated by three independent coders, each with a background in psychology or social work, for anxiety on a scale of zero to three. Each of their ratings were averaged and rounded to the nearest integer. This coding ultimately resulted in a set of four target classes taking on values zero, one, two or three, where a class of zero indicates no anxiety and class three represents high anxiety.

Terminology

In machine learning, the general goal is to take some input data and develop a set of rules from which one can derive an output. Input data consists of a collection of instances, which are typically represented as vectors $\vec{x}$. Each entry in the vector is

referred to as an attribute or feature. The question of what constitutes an attribute is one with many answers depending on how the problem is formulated, and is discussed further in Chapter I. The output value derived from our set of rules typically consists of one value, $\hat{y}$, called a label or target. This value can be numerical or categorical, depending on the problem at hand. In the case of our problem, each sentence is an instance, and the rating on a scale of zero to three is a label.

The set of rules that allow us to predict the label of an instance is referred to as a model. Models can perform different tasks, namely regression or classification. In a regression problem the model tries to predict a continuous value that is typically a floating-point number, e.g. 3.14. In a classification problem the model is concerned with providing a discrete value as output. For example, determining the salary of an NBA player given their season statistics is a regression problem, whereas predicting the breed of dog based on an image of said dog is a classification problem. The problem we address in this thesis is one of classification.

It is useful to be able to discern between the true label given to an instance and the label predicted by our model. We denote the true label given to an instance with y. As mentioned above, labels can be numerical or categorical. Our data is both ordinal and categorical, meaning that each label (zero, one, two or three) corresponds to a particular category, called a class (no, low, medium or high anxiety, respectively), and that these categories imply some sense of order. It is important to acknowledge when data are ordinal because it allows us to better assess the viability of our model. For instance, a model predicting that a sentence has no anxiety when the sentence is actually high in anxiety is a much graver error than a model predicting that the same sentence has medium anxiety (rather than its true label of high anxiety).

Overview of Topological Data Analysis

Topological data analysis (TDA) is an emergent area of interest in the data science community [3]. The underlying idea of TDA is that our data form shapes with specific mathematical properties. We can use these mathematical properties to discern the different classes to which our data belong. The tool that we use to perform TDA is known as persistent homology. Like in a game of connect-the-dots, persistent homology allows us to impose a continuous image on a discrete set of data. In particular, persistent homology is concerned with the number of holes that appear in our data at each dimension when we try to create a continuous shape.

Because topological data analysis is robust to outliers, it is very attractive for use in situations where data are particularly noisy [4]. We employ topological data analysis as a method of feature extraction before applying different machine learning models.

Overview of Methods

As mentioned above, we begin by exploring the ways that we can use TDA as a method for feature extraction. In order to use TDA for textual data, we embed each sentence into a real-valued vector space. Then, after applying persistent homology, we employ a number of machine learning algorithms to classify the level of anxiety in each sentence.

Outline

We begin with a discussion of related work in Chapter II, before proceeding to explain our methodology in Chapter III. The results of our work are discussed in Chapter IV. Finally, a conclusion that includes discussions of limitations and areas for future work is included in Chapter V.

II. RELATED WORK

Detecting emotion from text has been a task of interest for a long time, and much research has been dedicated to training classifiers capable of detecting a range of affect states. With the advent of popular social network sites such as Twitter, many researchers have turned to the internet to collect data [5, 6, 7] and have achieved promising results. However, recognizing emotion from text alone is still quite challenging, so text data is sometimes accompanied by other supplemental data like audio, resulting in even greater levels of classification accuracy [8, 9].

Our project differs from most other attempts to classify emotion because we analyze the nuances in a single emotion rather than a range of different feelings. Although some work exists in detecting anxiety through text [6], the work is limited in scope, analyzing only tweets containing the words 'work' and 'feeling.' In addition, our approach pairs anxiety recognition with topological data analysis, an approach that is, to the best of our knowledge, completely novel.

As is discussed in Chapter III, the word vector representation, or embedding, used can have a significant impact on model accuracy [10]. Popular word embedding methods include GloVe [11], which we use in this paper, and word2vec [10].

Topological data analysis, as mentioned above, is a relatively new field of interest in the world of data science and machine learning. Many researchers have explored the possible applications of TDA to time series data [12, 13, 14, 3] and have achieved promising results. There has also been recent interest in the applications of TDA to specifically text-based data [15, 16]. A standing research question is concerned with the ways in which we can use the information yielded by persistent homology as input suitable for machine learning tasks. Popular kernel-based learners such as support vector machine (SVM) classifiers require more than just a persistence diagram in order to make predictions. For example, they

need either a stable kernel like those provided in [17, 12] or a vector-based representation of persistence diagrams such as the ones detailed in [18, 19].

When working with traditional machine learning options, we first chose to explore support vector classifiers both because of their performance on text classification [5] and their ability to perform well on small data sets [9]. We also chose to implement a logistic regression classifier due to the promising results obtained by others using this algorithm in conjunction with persistence images [19]. Finally, to provide a baseline for comparing the efficacy of persistence images to that of solely persistence diagrams, we implement extreme gradient boosting.

III. METHODOLOGY

Data Collection

Our data was collected with the help of colleagues from the School of Social Work at Texas State University and the Graduate College of Social Work at the University of Houston. Ten student veterans with PTSD and social anxiety disorder agreed to be interviewed by a professional psychologist. Their interviews were transcribed and broken into 1,187 responses of varying lengths. Then, three independent coders with experience in mental health research and treatment of PTSD and anxiety listened to recordings of each interview and rated the transcribed responses for anxiety following a scale of zero (no anxiety present in the sentence) to three (high anxiety present in the sentence). Two sentences were missed by our judges, bringing our final sample size down to 1,185 responses. The data collection methods used for this project are further described in [1].

Type	**ICC**	**F Statistic**	**p-value**	**95% CI**
ICC(3,3)	0.678	3.103	0	[0.64, 0.71]

Table 1: ICC(3,3) statistic for our data set.

We measured the amount of agreement between our three coders by using the intraclass correlation coefficient (ICC), which is a measure of inter-rater reliability. In other words, the ICC quantifies how much our different judges agree with each other. We used $ICC(3, 3)$ because each judge rated each instance, and we considered the mean of their ratings to be our target [20]. Our results, found in Table 1, suggest a reasonable amount of agreement between our three coders.

After each judge rated every target, their ratings were averaged and rounded to

the nearest integer, which then became our final target class. The distribution of average ratings is right-skewed, with only 18 instances of high stress responses out of 1,185 total rated instances. The distribution of our target classes can be seen in Figure 1.

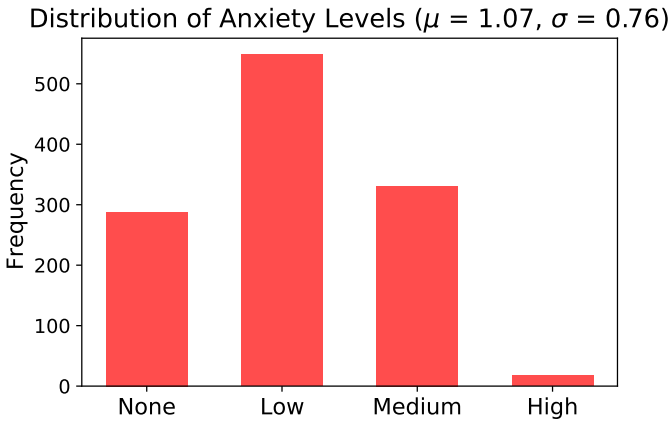


Figure 1: The distribution of average anxiety ratings across all responses.

Word Embeddings

In order to perform machine learning on our data, we must first find some scheme with which we can transform our data into real-valued vectors. This process is known as an *embedding*, whereby a single word is projected into $\mathbb{R}^d$ for $d \in \mathbb{N}$. Thus, if we consider a sentence $x = \{w_0, w_1, w_2, ..., w_n\}$ consisting of a set of n words, each word w_i gets transformed into a real-valued vector, $\vec{w_i} \in \mathbb{R}^d$ for $i = 0, 1, ..., n$, so that x becomes a matrix of vectors of the form $\vec{w_i}$.

A suitable word embedding will do more than assign random numbers as entries in each word vector. Ideally, an embedding retains certain semantic qualities that exist within a word. A good word embedding will typically assign words that are closer in meaning to vectors that are closer together in space. For instance, one should expect the vector representation of the word 'dog' to exist closer to the vector representation of the word 'cat' than to that of the word 'truck.' In fact,

certain embeddings even allow for arithmetic on word vectors to yield a meaningful answer. For example, the analogy 'queen is to woman as princess is to girl' will be represented in the vector space as queen - woman = princess - girl.

We employ GloVe, which stands for Global Vectors, in order to embed each sentence into a vector space. GloVe is a popular word embedding algorithm which has been shown to yield state-of-the-art results on a myriad of natural language processing tasks [11].

Topological Data Analysis

After embedding each sentence into a vector space $\mathbb{R}^d$, we are left with a discrete collection of vectors on which we can employ persistent homology. We do this by constructing *simplicial complexes* from our data. A simplicial complex is a collection of *simplices*. We can envision a 0-simplex as being a singular point, a 1-simplex as a line between two points, and a 3-simplex as a triangle [15]. Simplices are defined in all dimensions, and a k-simplex consists of a collection of points in (k+1) dimensions. A visualization of select simplices and a simplicial complex from [15] is given in Figure 2.

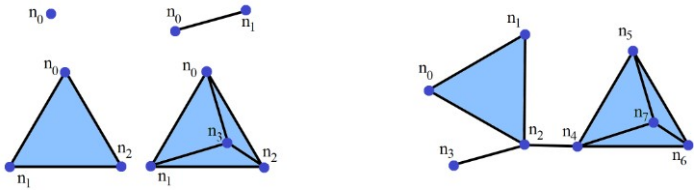
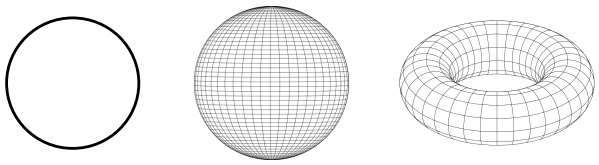


Figure 2: A 0-simplex, 1-simplex, 2-simplex, and 3-simplex (left). A simplicial complex (right). This figure first appeared in [15].

As mentioned above, we begin with each sentence consisting of a discrete set of vectors residing in $\mathbb{R}^d$. This is sometimes referred to as a *data cloud*. With this data cloud, we can construct what is known as a *Vietoris-Rips complex*, often abbreviated as a Rips complex. A Rips complex is a simplicial complex consisting of



$\beta_0 = 1$ $\beta_0 = 1$ $\beta_0 = 1$
$\beta_1 = 1$ $\beta_1 = 0$ $\beta_1 = 2$
$\beta_2 = 0$ $\beta_2 = 1$ $\beta_2 = 1$

Figure 3: An example of some shapes and their Betti numbers.

k-simplices whose components $\{x_\alpha\}_0^k$ are pairwise within some distance ϵ [21].

Of interest to us is the number of holes that appear in our data as we construct this simplicial complex. Formally, the i^{th} *Betti number*, β_i is the rank of the i^{th} homology group of a topological space. Informally, the i^{th} Betti number is the number of i-dimensional holes in our data. We are primarily interested in the number of components, β_0, and loops, β_1, present in our data cloud. Some examples of familiar shapes and their Betti numbers are given in Figure 3.

However, the Betti numbers of a single Rips complex constructed in isolation tells us little about our data. We must delve further, and ask which of these holes persist in the Rips complexes that are constructed at different values of ϵ [21]. We do this through the construction of a *persistence diagram*.

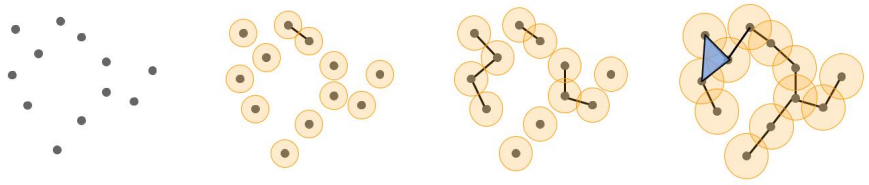


Figure 4: An example of Rips filtration.

A persistence diagram is a graph that gives us information about the components and loops that appear in our data as ϵ increases in value. Imagine that we are given a set of discrete points such as an incomplete connect-the-dots. Rather than playing the game traditionally, wherein we would connect the dots according to

a predefined order, we take a different approach. We begin by drawing an arbitrarily small circle around each point in our connect-the-dot game. Then we slowly start to draw larger and larger circles around each point until two or more of those circles intersect. When two dots' circles intersect, we connect them. As we begin to connect more dots, loops will start to appear (and eventually disappear) in our data and individual components will start to disappear. For each component or loop, we plot a point (x, y) in our persistence diagram so that x represents the radius of the circles when the component or loop first appeared and y represents the radius of the circles when the component or loop disappeared. This is our persistence diagram.

After constructing a persistence diagram, we will likely find that many points in the diagram are clustered around the line $y = x$. Those points that are further away from the line $y = x$ are considered persistent, and of interest to us [14]. An example of one of our persistence diagrams is shown in Figure 5.

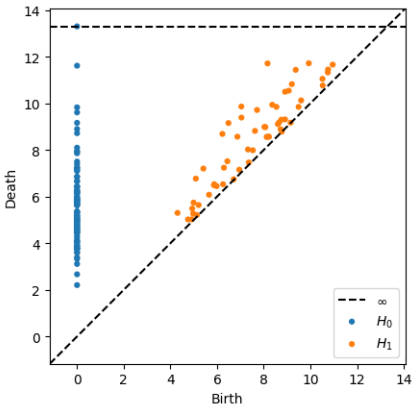


Figure 5: An example of a persistence diagram (right). Note that 'Birth' refers to the ϵ value at which the loop appeared and 'Death' refers to the ϵ value at which the loop disappeared.

Although persistence diagrams are powerful, they are not conducive as input for many popular machine learning algorithms. This is where what is known as a *persistence image* becomes useful. As explained in [18], persistence images offer a stable, vector-based representation of persistence diagrams that work as input for

many popular machine learning algorithms.

As described in [18], to create a persistence image, we begin by mapping a persistence diagram PD to an integrable function $\rho_{PD} : \mathbb{R}^2 \to \mathbb{R}$ that's defined as a weighted sum of probability density functions (one centered at each point in PD). Then a grid is defined by taking a discretization of a subdomain of ρ_{PD}. Finally, a persistence image is yielded by taking an integral of the function on each grid box (Fig. 6).

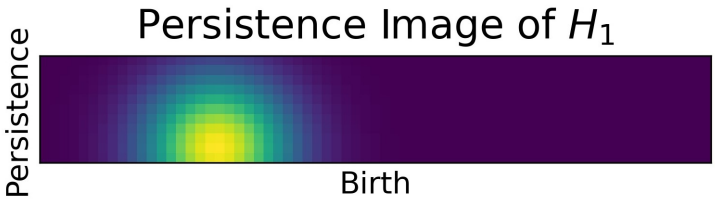


Figure 6: The persistence image of an instance of class one.

Materials

We used Sci-kit Learn, TensorFlow, and Keras to implement the machine and deep learning models. To extract the persistence diagrams from our data we used Ripser [22]. Finally, to convert the persistence diagrams to persistence images we used Persim [23].

Methods

We began preparing the data by stripping each response of punctuation and converting all letters to the same case e.g. lowercase. Because the data consist of transcribed interviews we did not need to remove emojis or links, which can be the case when analyzing text data from social media. Then, we tokenized each response and embedded them into a vector space using GloVe (Fig. 7).

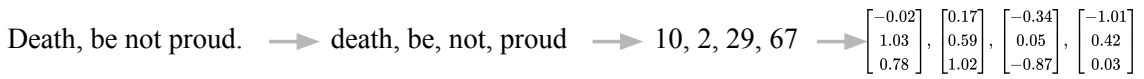


Figure 7: An example of text processing, from a complete sentence through the embedding.

After embedding our data in a feature space, we used Ripser [22] to extract the persistence diagram from each sentence. Then we used Persim [23] to convert each diagram into a persistence image. Persistence images begin as matrices, so in order to convert them into suitable input for our machine learning algorithms we flattened each matrix into a vector using NumPy [24].

As mentioned previously, we used this set of input vectors to train two different models: a support vector classifier and logistic regression model. Additionally, we used the raw persistence diagrams as input to an extreme gradient boosting (XGB) classifier. Since XGB classifiers do not require persistence images in order to train, this algorithm serves as a baseline of comparison in order to evaluate whether extracting these persistence images does in fact allow for better prediction accuracy. Finally, to act as a litmus test, we also trained a long short-term memory (LSTM) network with an attention layer on the original embedded word vectors themselves. Since LSTMs have been shown to perform well on text classification tasks [25], they provide a reliable basis of comparison between our approach and pre-existing classification methods.

In order to more accurately assess the performance of our models, we used five-fold cross validation. In five-fold cross validation, one fifth of the data are set aside, while the other four fifths are used to train the model. When the model is done training it is evaluated for accuracy on the previously-withheld fifth of data. This process is then repeated, except now the fifth of the data reserved for testing consists entirely of instances that were previously in the training set. This process is again repeated until every instance has been in the test set once and the training set

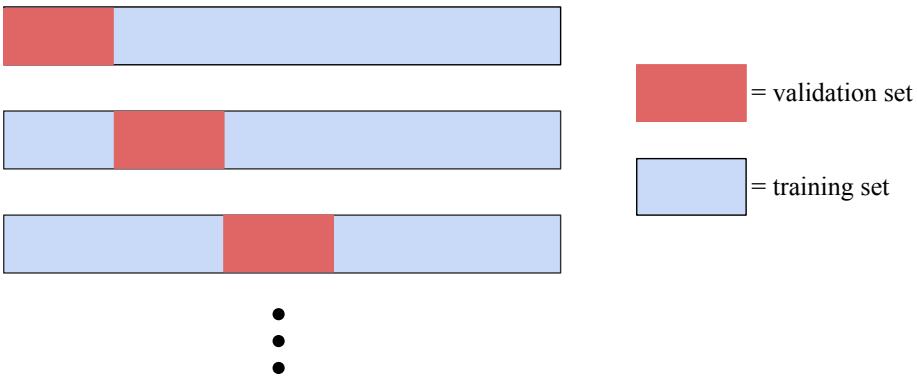


Figure 8: A visualization of five-fold cross validation.

four times. This process of validation allows us to ensure that our model performs well on all of our data, i.e. it didn't just get an "easy" set of test data. We used the accuracy given along each fold to construct an average classification accuracy for all of our models (see Chapter IV).

IV. RESULTS

In order to properly evaluate our models we developed two baselines. The first baseline is a blind guess, which would result in about a 25% accuracy score since there are four classes. The second baseline accuracy is 46.2%, which is yielded by predicting only the most common class.

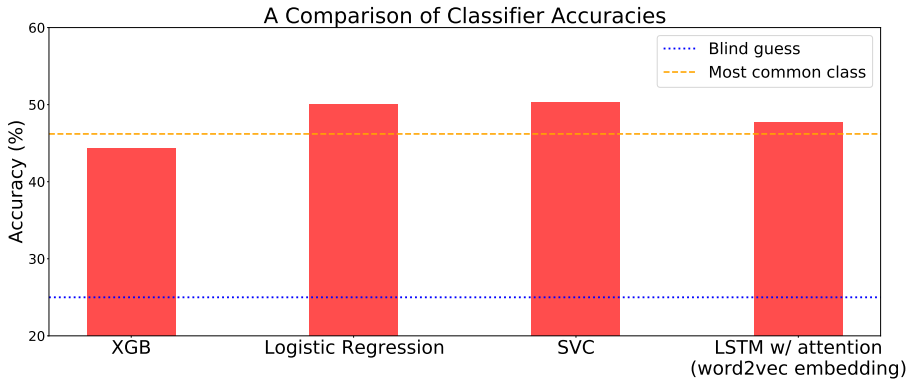


Figure 9: The five-fold cross-validated accuracy scores of our models.

Our LSTM with attention achieved a cross-validated accuracy of 47.70%, just barely above our second baseline. Deep learning models typically require a large volume of training data, which we do not have. The limitations highlighted by the performance of this model elucidate our motivation for exploring topological data analysis as a means of feature extraction in the first place.

Similarly, our XGB classifier achieved an accuracy of 44.25%. This is the only classifier that performed below our second baseline accuracy of 46.2% (Fig. 9). This model's poor performance justifies our need for the ability to train a series of more powerful classifiers like support vector machines and logistic regression.

Our logistic regression model did perform above our baseline, and in fact it achieved an accuracy of 50.08%, outperforming even our deep learning model. Our support vector classifier achieved a similar accuracy of 50.25%.

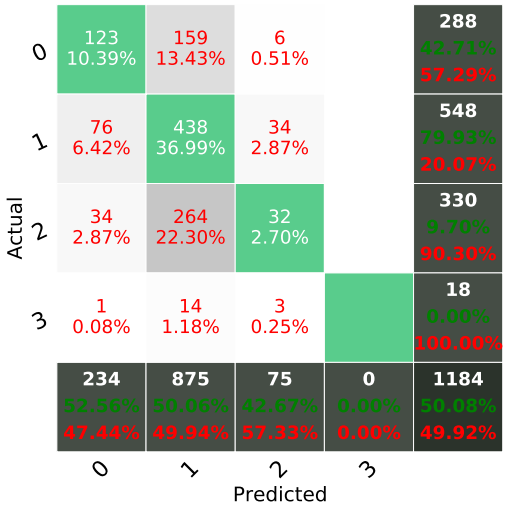


Figure 10: The confusion matrix produced via five-fold cross-validation of our logistic regression classifier.

Model	**Input Data**	**Accuracy(%)**
Extreme Gradient Boosting	Persistence Diagrams	44.25
Support Vector Machine	Persistence Images	50.25
Logistic Regression	Persistence Images	50.08
LSTM with attention	Embedded Sentences	47.70

Table 2: An overview of our classifiers, the input data they were trained on, and their accuracy.

Discussion

Both our support vector machine and logistic regression models outperformed the LSTM with attention as well as our baselines. The confusion matrices from both of these models highlight that although they outperform our second baseline, our class distribution is so skewed that even our best models struggle to learn from this data (Figs. 10 & 11).

Of particular note is that none of our classifiers were able to correctly identify instances of our third class. However, as was previously explained in II, our data is ordinal, meaning it is better to mislabel instances of class three as class two than it

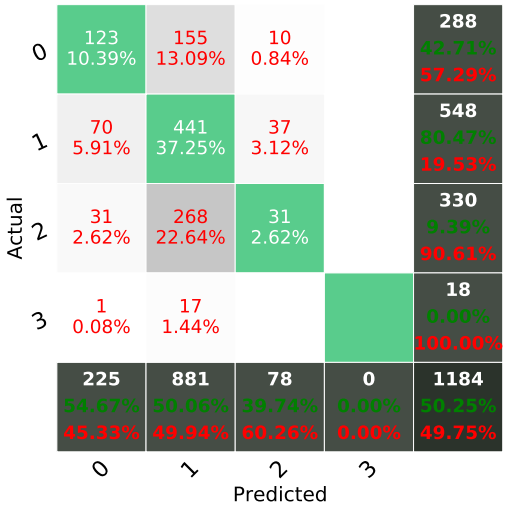


Figure 11: The confusion matrix produced via five-fold cross-validation of our support vector classifier.

is to mislabel them as class one. In this regard, our logistic regression classifier came the closest to correctly recognizing sentences of class three, as three out of a total of eighteen instances were labeled as class two by the model.

V. CONCLUSION

The research detailed in this project portrays topological data analysis as a viable method of feature engineering for text classification tasks. Notably, our work highlights how persistent homology paired with traditional machine learning algorithms can outperform neural networks. Our work produced tools that can help to support the decision making of psychology and social work professionals who are treating social anxiety.

Beyond the immediate scope of our problem, our research is applicable to many other areas as well. For instance, automatic emotion recognition can aid in identifying subject areas in which students don't feel confident [25] or help analyze societal responses to stressful situations en masse [6].

Limitations

Our psychologists listened to the recorded interviews while rating each response for anxiety, which resulted in a few instances of sentences having the same text content with different anxiety levels. For example, the lone word 'yeah' appears as a response in our data set 68 times, and the anxiety level of the response varies between instances.

Additionally, deep learning typically requires a large amount of training data, so the performance of our deep learning models is also likely limited by the small sample size of our data. This is compounded by the deeply skewed distribution of our classes (Fig. 1), which is difficult for most machine learning models to overcome.

Areas for Future Work

Many areas of interest remain available to be pursued further. Of particular interest is the possibility of using persistence images as input to a neural network. Extracting the persistent homology of sentences before using them to train a neural network is a difficult question, and one that has as of yet been unanswered. However, persistence images show promise in exploring the applications of visual attention to topological data analysis.

A second direction for future work is the exploration of how TDA can be used as a supplement for additional features. For instance, [16] discusses how augmenting a traditional feature set with topological features can improve classification accuracy by up to 5%. Given the accuracy of our current models, this approach provides a promising method for increasing classifier performance.

REFERENCES

[1] M. H. Trahan, A. R. Ausbrooks, K. S. Smith, V. Metsis, A. Berek, L. H. Trahan, and K. Selber, "Experiences of student veterans with social anxiety and avoidance: A qualitative study," *Social Work in Mental Health*, vol. 17, no. 2, pp. 197 – 221, 2019.

[2] M. Trahan, R. Morley, E. Nason, N. Rodrigues, L. Huerta, and V. Metsis, "Virtual reality exposure simulation for student veteran social anxiety and ptsd: A case study," *Clinical Social Work Journal*, pp. 1 – 11, 2019.

[3] S. Gholizadeh and W. Zadrozny, "A short survey of topological data analysis in time series and systems analysis," *arXiv*, 10 2018.

[4] H. Edelsbrunner and J. Harer, "Persistent homology-a survey," 1991.

[5] I. Fatima, B. U. D. Abbasi, S. Khan, M. Al-Saeed, H. F. Ahmad, and R. Mumtaz, "Prediction of postpartum depression using machine learning techniques from social media text.," *EXPERT SYSTEMS*, vol. 36, no. 4, 2019.

[6] D. Gruda and S. Hasan, "Feeling anxious? perceiving anxiety in tweets using machine learning," *Computers in Human Behavior*, vol. 98, pp. 245 – 255, 2019.

[7] Z. Rajabi, A. Shehu, and O. Uzuner, "A multi-channel bilstm-cnn model for multilabel emotion classification of informal text," *2020 IEEE 14th International Conference on Semantic Computing (ICSC)*, pp. 303 – 306, 2020.

[8] J.-S. Kim, "Multimedia emotion prediction using movie script and spectrogram," *Multimedia Tools and Applications: An International Journal*, 2020.

[9] Z. jing Chuang and C. hsien Wu, "Multi-modal emotion recognition from speech and text," in *International Journal of Computational Linguistics and Chinese Language Processing*, pp. 1–18, 2004.

[10] 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*, (Portland, Oregon, USA), pp. 142–150, Association for Computational Linguistics, June 2011.

[11] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

[12] L. M. Seversky, S. Davis, and M. Berger, "On time-series topological data analysis: New data and opportunities."

[13] S. M. Kennedy, J. D. Roth, and J. W. Scrofani, *A Novel Method for Topological Embedding of Time-Series Data*. 2018.

[14] V. D. Silva, P. Skraba, and M. Vejdemo-Johansson, "Topological analysis of recurrent systems," 2012.

[15] S. Gholizadeh, A. Seyeditabari, and W. Zadrozny, "Topological signature of 19th century novelists: Persistent homology in text mining," *Big Data and Cognitive Computing*, vol. 2, pp. 1–10, 12 2018.

[16] S. Gholizadeh, K. Savle, A. Seyeditabari, and W. Zadrozny, "Topological data analysis in text classification: Extracting features with additive information," 3 2020.

[17] J. Reininghaus, S. Huber, U. Bauer, and R. Kwitt, "A stable multi-scale kernel for topological machine learning."

[18] H. Adams, T. Emerson, M. Kirby, R. Neville, C. Peterson, P. Shipman, E. Hanson, F. Motta, and L. Ziegelmeier, "Persistence images: A stable vector representation of persistent homology," 2017.

[19] I. Obayashi and Y. Hiraoka, "Persistence diagrams with linear machine learning models," *arXiv*, 6 2017.

[20] S. PE and F. JL., "Intraclass correlations: Uses in assessing rater reliability," *Psychological Bulletin*, vol. 86, no. 2, pp. 420–428, 1979.

[21] R. Ghrist, "Barcodes: The persistent topology of data," 2007.

[22] C. Tralie, N. Saul, and R. Bar-On, "Ripser.py: A lean persistent homology library for python," *The Journal of Open Source Software*, vol. 3, p. 925, Sep 2018.

[23] N. Saul, "Persim." `https://github.com/scikit-tda/persim`, 2019.

[24] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, pp. 357–362, Sept. 2020.

[25] X. Feng, Y. Wei, X. Pan, L. Qiu, and Y. Ma, "Academic emotion classification and recognition method for large-scale online learning environment-based on a-cnn and lstm-att deep learning pipeline method.," *International journal of environmental research and public health*, vol. 17, no. 6, 2020.