Understanding Bias in Datasets using Topological Data Analysis

Ramya Srinivasan [1*] and **Ajay Chander**[1]
[1]Fujitsu Laboratories of America
{ramya, achander}@us.fujitsu.com

Abstract

From the data that goes into the AI pipeline to the choice of models and target application, AI safety demands examination at various levels of abstraction. Additionally, the notion of safety has to be assessed across different types of humans-in-the-loop involved in the AI pipeline—from AI scientists and software engineers to various types of consumers. Much of the research on AI safety has focused on catering to the needs of AI scientists (such as in design of systems robust to adversarial attacks and ethically grounded algorithms) and consumers (such as in engendering trust and facilitating model interpretation). Choosing the right AI model, tuning various parameters, and processing datasets are some of the many issues that engineers face. A wrong choice in any of these steps can aggravate safety issues in an inconspicuous manner and can harm the interests of the consumer. There is thus a need to provide software engineers with a much more accessible tool whereby they can be better aware of their decisions and the consequences those decisions bear to the consumer. In this paper, we propose a persistence homology based visualization that can aid software engineers in understanding bias in datasets. Unlike other machine learning methods, this topological data analysis method imposes less burden in the sense that the human-in-the-loop does not need to select the right metric or tune parameters, and can determine the bias based on the data before choosing any model. Experiments on the German credit dataset demonstrates the effectiveness of the proposed method in identifying the bias in the dataset due to age.

1 Introduction

Amidst the massive upsurge of AI-based applications, there has been growing concern amongst regulatory bodies, policy makers, and consumers about AI being a black-box technology. Beyond just AI scientists, today, a variety of stakeholders are involved in AI based decision pipelines. These stakeholders could include business executives, software engineers, and consumers. Thus, successful adoption and safety of AI systems relies on how much different types of stakeholders can trust the AI based decision and understand its functionality.

Most existing works study AI safety from the perspective of either an AI scientist or the consumer. Some such efforts include building systems robust to adversarial attacks, designing ethically grounded and fair algorithms, and creating explainable AI models to facilitate model interpretation and to engender trust in the consumer. However, there are significant knowledge gaps between an AI scientist who designs a model, a software engineer who implements and integrates various models, and a consumer who uses a model for their custom applications. These gaps could affect safety of AI systems in inconspicuous ways.

Consider the role of software engineers in the AI pipeline. People in these roles are responsible for building software. These engineers could choose some off-the-shelf AI modules and integrate them with various APIs and other software components. They do have to understand various parameters involved, tune them, and perhaps re-configure various architectural blocks to suit the requirements of an application. At the outset, the job of such developers may seem to be mostly engineering oriented without having to worry about the consequences of how the model's decision affects the consumer. However, this is not the case.

For one, there are several models available, which is the right model for a particular application? Next, there are millions of parameters involved, which ones need to be tuned and how to set their values? Even before choosing the right model, the data has to be processed and set into a format that is amenable for the model to process. The data itself could be biased or limited. How to ensure accurate data preprocessing? Such questions have to be carefully examined and appropriately addressed. Failure to do so can aggravate safety issues in an inconspicuous manner and can cause serious consequences to the consumer.

As an instance, consider prediction of loan defaulting using AI. A software engineer unaware of bias in training data may inadvertently use it to train models basing his judgement merely on validation and test accuracy. Suppose the training data was biased- say there were too many young people who defaulted- then the model is likely to predict the same

*Corresponding Author

on test data. This can have serious consequences on a young applicant who actually may not default. Thus, there are several safety critical issues that need to be addressed from the perspective of a software engineer. Engineering trust-worthy AI software architectures necessitates accessible and explainable methods that allow software engineers to seamlessly preprocess the data, select the right model, and integrate various AI modules into the use cases of their interest. Given the widespread adoption of AI and the scarcity of people skilled in AI, there is an even greater need for building such accessible tools in order to ensure AI safety.

With the number of biased systems expected to increase within the next five years, understanding safety in the context of bias and ensuring fair decisions has been a major area of interest across several AI based systems used in banking, insurance, hiring, to name a few. In this paper, we propose a method based on topological data analysis (TDA) to enable software engineers to visualize the bias in datasets prior to even applying any bias mitigation algorithm. Specifically, we leverage a technique called persistence homology which can be viewed as a complement to standard feature representation techniques used in AI. Unlike other feature representation techniques which require guidance from a machine learning expert regarding the choice of algorithm, model architecture and parameters, this technique does not require the human-in-the-loop to select any metric or tune parameters, and can work with sparse datasets as well. Experiments on the German credit dataset demonstrates the effectiveness of the proposed method in uncovering the bias due to age in the prediction of loan defaulting. The specific contributions of the paper can be summarized as follows:

1.1 Contributions

- We study AI safety from the perspective of software engineers who form a critical link in the AI pipeline. In particular, we describe an accessible method by which software engineers can understand bias in datasets.
- We elucidate a novel application of topological data analysis to quantify bias due to different attributes in a dataset. Presence of bias is visualized by means of persistence barcodes and is also validated through permutation tests (Section 3).
- We demonstrate the effectiveness of the proposed method in detecting bias due to age on the German credit dataset (Section 4).
- We provide an accessible guideline to facilitate software engineers to easily use the method.

The rest of the paper is organized as follows. A review of related work is provided in Section 2. Section 3 provides the details of the method. Results are discussed in Section 4. Section 5 lists some common questions and answers in order to enhance comprehension about the accessibility of the approach. Conclusions are provided in Section 6.

2 Related work

We review related works concerning TDA and its applications in machine learning. We also review works concerning AI safety, and bias and accessibility of AI systems.

2.1 TDA

TDA is an interdisciplinary field spanning topology and data analysis, and is used as a tool to uncover patterns in data. TDA is based on the philosophy that data has shape, and that shape has meaning. Persistence homology (PH) is a technique from TDA that can identify clusters, holes, and voids within a set of points. Persistence homology can be viewed as a complement to standard feature representation techniques used in artificial intelligence, and offers the advantage of being applicable to sparse datasets as well. Unlike other feature representation techniques which require guidance from a machine learning expert regarding the choice of algorithm, model architecture and parameters, this technique does not require any parameter tuning.

TDA can be used as an independent tool to uncover patterns in data or it can also be used in conjunction with machine learning (ML) techniques as a feature extractor. TDA has been used in several computer vision applications such as for shape analysis [Zhou *et al.*, 2017; Wang *et al.*, 2011], for texture analysis [Zeppelzauer *et al.*, 2018], for medical imaging [Pachauri *et al.*, 2011], for pose estimation [Nguyen *et al.*, 2018], and for structure recognition [Li *et al.*, 2014]. It has also been used in NLP applications for detecting structural similarity in texts [Zhu, 2013]. TDA can also help in time series data analysis such as in [Umeda, 2017]. Recently, TDA has also been used to shed light about the workings of convolutional neural networks through works such as [Gabrielsson and Carlsson, 2018; Carlsson and Gabrielsson, 2018] wherein the authors perform TDA on the weight matrices of the deep networks to study what is being learnt at each step of training. In a recent blog post [Gunnar, 2018], it is also mentioned that TDA on weight matrices of CNNs can be used to understand dataset variability, correlation between accuracy and persistence barcodes, etc. However, we have not come across works that use TDA and PH to understand bias due to various attributes in a dataset, which is the focus of this work.

2.2 AI safety

The safety of AI models could be compromised in several ways—a decision may not be ethically justified [Bostrom and Yudkowsky, 2018], there could be bias in the system [Srivastava and Rossi, 2018], the system could cause hazardous effects [Pettigrew *et al.*, 2018], or the privacy and security of individuals may be at stake [Al-Rubaie and Chang, 2018]. Several works have studied the impact of AI based decisions in safety critical applications such as healthcare [Challen1 *et al.*, 2019], judiciary [Kleinberg *et al.*, 2018], transport [Stilgoe, 2017], finance [FSB, 2017], amongst others. These studies have examined the impact AI based decisions can have on various consumers (such as doctors, patients, judges, etc.), and how the AI model can be made explainable to address the needs of these consumers. In addition, several other excellent works [Kurutach *et al.*, 2018; Goodfellow *et al.*, 2014; Ramakrishan and Shah, 2016] have explored explainable AI methods to cater to the needs of AI scientists in understanding the underpinnings of various models. However, there is a pressing need to address AI safety from the perspective of a software engineer. This paper is one such effort.

2.3 Bias and AI accessibility

Recognizing the need to develop tools that are accessible across a broader set of users, IBM released AIFairness360 which is an excellent tool to compute bias along various metrics [Bellamy *et al.*, 2018]. Accenture also released a similar fairness assessment tool [Peters, 2018]. It was also reported that Microsoft is creating an oracle to catch biased AI algorithms [Knight, 2018]. Google introduced AI bias visualization with the What-If tool and TensorBoard [Wexler, 2018]. There have also been several academic works in this area. In a recent paper, MIT researchers detailed what they call as a toolbox for helping machine learning engineers figure out what questions to ask of their data in order to diagnose why their systems may be making unfair predictions [Chen *et al.*, 2018]. Guidelines have also been proposed to reduce the potential for bias in AI. These include "factsheets for datasets" from IBM, and "Datasheets for Datasets", an approach for sharing essential information about datasets used to train AI models [Gebru *et al.*, 2018]. In this work, we propose a complementary perspective of analyzing bias using topological data analysis. This method offers the advantage of being applicable across sparse datasets and can be used for efficient feature representations and visualizations. Furthermore, a software engineer does not have to tune parameters or deal with metrics of evaluation, thereby enhancing accessibility.

3 Methodology

We begin by reviewing some necessary definitions. More details about the same can be found in any book on algebraic topology such as [Weintraub, 2014].

3.1 Definitions

- *Point cloud:* A point cloud is often defined as a finite set of points in some Euclidean space, but may be taken to be any finite metric space.
- *Simplex:* A simplex is a generalization of a triangle or a tetrahedron to their higher dimensional counterparts.
- *Simplicial complex:* A simplicial complex is a combination of simplexes such that any face (subset) of a simplex from K is also in K, and the intersection of any two simplices in K is either empty or shares faces.
- *Vietoris-Rips complex:* Also known as the Rips complex, this is a simplicial complex with radius r that consists of the set of all points (and simplicial complexes) such that the largest Euclidean distance between any of its points is at most $2r$.

A natural question may then arise as to what is the best value of r to use for a dataset. The answer is provided by persistence homology, which is defined next.

- *Persistence Homology (PH):* This is a method for computing topological features of a space at different spatial resolutions. Such topological features could include clusters, holes and voids in a dataset. More persistent topological features are detected over a wide range of spatial scales and are deemed more likely to represent true topological features of the underlying space rather than artifacts of sampling, noise, or other factors. For example, in a 2D space, as the radius r is gradually increased, points that are initially disconnected could get connected and higher order topological features such as holes could appear. The holes could later disappear as the entire space gets connected. The process of varying the radius is referred to as *"filtration"*. The best value of r is one that can reveal persistent topological features in the dataset. This value is automatically computed by PH softwares.
- *Persistence diagrams:* The appearance and disappearance of clusters, holes and other such topological features can be captured by means of persistence diagrams (bottom right in Figure 2) and persistence barcodes. Persistence diagram is a plot of the birth time (i.e., the value of the radius at which a topological feature appears) and death time (i.e., the value of the radius at which a topological feature disappears) of a topological feature as the the radius is varied. Any point on the diagonal of this plot is insignificant as it does not persists long enough (i.e. it disappears soon after it appears). Points above the diagonal are topological features that persist.
- *Persistence barcodes:* This captures the interval between the birth and death of a topological feature. It is another way of looking at the persistence diagram.

With the above background, we are now in a position to understand the intuition behind using PH in analyzing dataset bias.

3.2 Intuition

Persistent barcodes as computed by PH is a collection of intervals along various dimensions. In dimension=0, the barcode output reflects the decomposition of the dataset into clusters or components. In clustering, a threshold is chosen, and any two points are connected by an edge if their distance is less than this threshold. As the threshold grows, more points will be connected, and there will be fewer clusters. The barcode is a way of tracking this behavior.

The author in [Gunnar, 2018] nicely illustrates the intuition behind barcodes. We leverage a similar example to explain the concept of barcodes. Consider some toy data as shown in Figure 1. On the left, we see a dataset that consists of two clusters close to each other and on the right we see another dataset that consists of two clusters which are relatively farther apart. The corresponding barcodes beneath each dataset represents two lines, one longer than the other. The presence of two lines indicates that there are two clusters in both these datasets. However, we notice that in the left dataset, the lines are shorter compared to the ones on the right. This is because, the clusters on the left are closer to each other, as a result, the two initial clusters get merged into a single large cluster and there is only one line (top one) after the merger happens. For the dataset on the right, this merger happens little later at a larger value of radius r as the clusters are farther apart.

The aforementioned illustration is for dimension =0, wherein PH captures connected components or clusters. The length of the barcode is indicative of how well connected the clusters are, and the number of barcodes is indicative of the

number of such connected components. For higher dimensions, the barcodes captures the presence of holes, voids, etc. Further, lengths of barcodes are used as indicators of variations in training data. For example, the authors in [Carlsson and Gabrielsson, 2018] use the length of barcodes as indicators of the accuracy of convolutional neural networks.

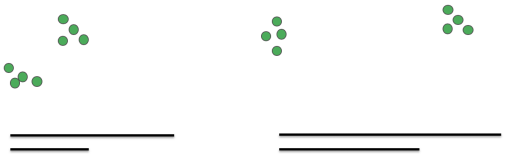

Figure 1: Intuition behind barcodes

3.3 Hypothesis

Now, imagine a 2D point cloud of predictors (attributes) and target variables, i.e., each predictor (say age) constitutes the x-coordinate and the y-coordinate is the target variable (say loan defaulting). We will use the words attributes and predictors interchangeably in the paper. In dimension 0, the barcode captures the connected components of the age variable with respect to loan defaulting. So, if the data consists of a particular age group of people who defaulted, it will be evident in the form of an isolated cluster, not connected to other age groups. In other words, if there is bias in the dataset, barcodes provide a visualization of the same. A barcode that is significantly longer than others is indicative of bias due to that predictor. If all barcodes are of same length, there is no bias due to that predictor.

3.4 Validation

The aforementioned hypothesis can be verified by means of statistical hypothesis tests. Since the distribution of topological features has not been well characterized yet, statistical inference on persistent homology must be non-parametric tests [Wadhwa *et al.*, 2018]. For our purposes, we use non-parametric permutation test.

If we define a function T that returns the persistent homology of a point cloud, then given two point clouds, C and D, we can use a permutation test to conduct statistical inference with the following null and alternative hypotheses:

$$H_A : T(C) = T(D) \tag{1}$$

$$H_0 : T(C) \neq T(D) \tag{2}$$

We then use the Wasserstein distance (Earth-mover's distance) as a similarity metric between persistent homologies of two point clouds [Vallender, 1974].

Going back to the example considered, suppose we have two point clouds corresponding to two predictors, say age and gender, with respect to loan defaulting. We can now run a permutation test on the two point clouds to confirm that the persistent homologies of the two are, in fact, distinct. We set the null hypothesis that the two persistent homologies are not distinct. The resulting p value from the test indicates whether the null hypothesis can be rejected or not. Typically if $p < \alpha$, the null hypothesis is rejected. The paramaeter α is known as the significance value of the test and a standard value of 0.05 is typically chosen for α. We use off the shelf TDAstats package to test the hypothesis [Wadhwa *et al.*, 2018].

3.5 Algorithm

The aforementioned procedure can be summarized as follows:
1. Create point clouds of individual predictors and the target variable.
2. Compute Rips complex for each of the point clouds created in step 1.
3. Compute persistance homology for the Rips complexes created in step 2 and plot persistance barcodes.
4. For each PH, the length of the longest barcode is a way of visualizing the bias due to individual predictors. If all barcodes are of same length, then there is no bias. If there are barcodes that are considerably longer than others, then there is bias.
5. Compute p values from permutation tests setting the null hypothesis that the resulting PHs from the two point clouds under consideration are not distinct. A rejection of null hypothesis indicates that the two PH are indeed distinct and thus there is bias in the attribute which has a long barcode.

4 Results

We demonstrate the method on German credit dataset [Dua and Graff, 2019]. This dataset contains 1000 data points wherein the goal is to predict loan defaulting based on twenty predictors such as credit history, savings, checking account status, property, housing, job, etc. "Age" and "gender" are the protected attributes with "old" and "male" being previlaged attributes and "young" and "female" being unprevilaged attributes.

As described earlier, different topological features can be detected at different dimensions. Dimension 0 reveals the existence of clusters or connected components. Figure 2 provides the persistence barcodes of age with respect to loan defaulting in dimension 0. The x-axis represents the variation of the radius parameter r. The y axis does not have a physical interpretation, it represents the set of all connected components. We see that there are several individual clusters when the radius is 1, these merge and there are two clusters until $r = 2$. Beyond $r = 2$, there is a single large cluster that persists. The length of the longest barcode is 4.

Now, consider the plot of persistence barcodes of gender with respect to defaulting as shown in Figure 3. We see a few clusters which persist upto $r = 1$, thus the length of the longest barcode is 1. From Figures 2 and 3, it can be inferred that the persistence homology due to age and gender are distinct. We can objectively validate the hypothesis that the two PHs are different by means of permutation test as described earlier. We obtained a p value of 0, thus leading to the rejection of the null hypothesis that the two PHs are not distinct. There is a single large cluster as evident by the long barcode in the PH of age, thus the bias due to age is significant. Furthermore, if we consider the length of the longest barcode as

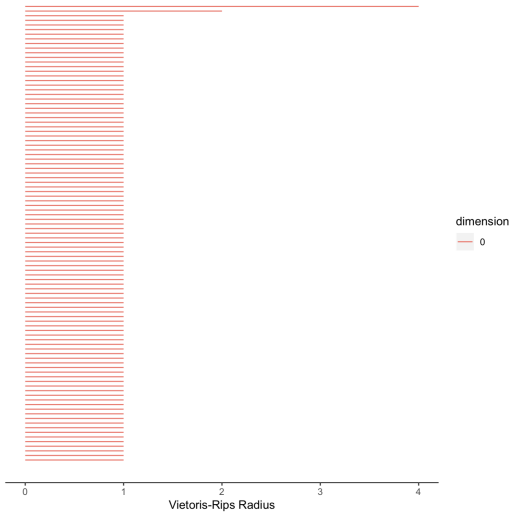

Figure 2: Persistence barcode of age with respect to defaulting

an indicator of the bias in the dataset, then the bias due to age is four times the bias due to gender (if there is any due to gender). In fact, since there is no single persistent bar in the PH of gender, we can conclude that there is no significant bias due to gender. In dimension 1, we did not observe a statistically significant difference between the two PHs. However, to detect bias, it suffices if there is a statistically significantly difference between the PHs in any one dimension.

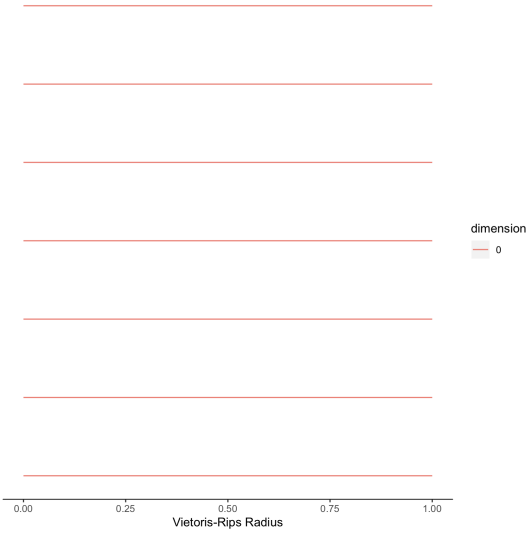

Figure 3: Persistence barcode of gender with respect to defaulting

The aforementioned result can also be validated using the AIfairness 360 tool as can be observed from Figure 4. AIfairness 360 also shows that there is no bias due to gender. Four out of the five metrics (statistical parity difference, equal opportunity difference, disparate impact, and average odds difference) show bias with respect to age. Theil's index, however does not show any bias. The presence of multiple metrics and varying amount of bias across those metrics might be little confusing to a software engineer not aware of the details of these metrics. The suitability of a particular metric may be dependent on the type of data amongst other factors. The burden of choosing a suitable metric might thus fall on the software engineer, who may not necessarily be equipped with the knowledge to do so. On the other hand, persistence homology based bias visualization method provides the software engineer with a universal tool that is applicable across datasets without having to choose any parameter. Intrigued by

Figure 4: Bias due to age: Visualizations from IBM AI360 tool

how PH of other attributes compare with respect to protected attributes, we also plotted the PH of attribute "job" with respect to the target variable. The PH of "job" was the same as that of gender indicating that there is no bias due to job as can be observed from Figure 5.

5 Accessibility for the software engineer

Engineering trust-worthy software architectural pipelines is an integral aspect of AI safety. There is a pressing need to create accessible AI interfaces and tools to ensure that software engineers who may not necessarily possess technical depth in AI are able to appropriately pre-process data, select the right AI model, and tune it. In this paper, we described how TDA can aid software engineers in understanding bias in datasets, a pre-processing step that is very important in the context of AI safety.

In this section, we summarize what a software engineer needs to know to leverage this method and how they can use the same. The mathematical background discussed earlier may give an impression that this method is not simple enough to be comprehended by a software engineer. However, the nice thing about TDA is that those details are not necessary to actually detect bias. Furthermore, the software engineer is not burdened to choose a bias metric. Below, we enlist some common questions and simple answers to further enhance the accessibility of this method.

- *What to know about TDA and PH:* Topology is the study of shapes. These shapes may be viewed as generalizations of triangle in higher dimensions and are referred to as simplicial complexes. Different simplicial complexes (triangles, tetrahedrons, etc.) appear based on the resolution at which the data is analyzed. PH is a method for computing topological features of a space at different spatial resolutions. Topological features could include clusters, holes and voids in a dataset.
- *How can TDA and PH help in detecting bias:* If there is bias due to an attribute in the dataset, it will be evident in the form of an isolated topological feature such as a cluster or hole that persists for a considerable interval.

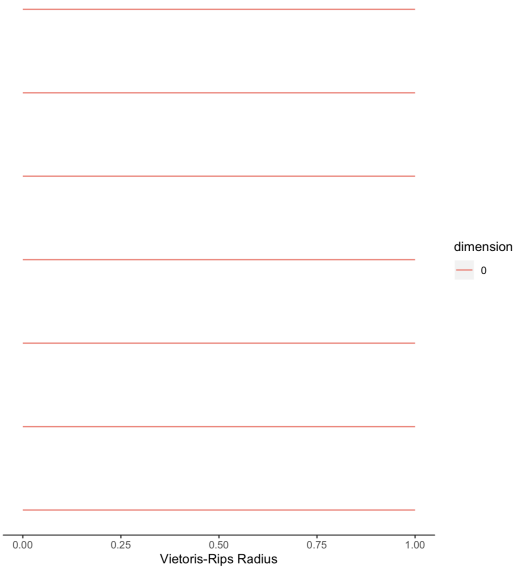

Figure 5: Persistence barcode of job with respect to defaulting

- *How to choose a method to construct simplicial complex:* This is based on data type. For example, Rips complex is typically chosen for point clouds, Morse complex is chosen for images, etc.
- *How to prepare data:* For categorical datasets, create point clouds of individual predictors and the target variable.
- *How to choose radius parameter :* The radius parameter is automatically chosen by PH based software to detect topological features of interest.
- *How to compute PH:* PH is visualized in terms of persistence diagrams and barcodes. Persistence diagram is a plot of the birth time (i.e., the value of the radius at which a topological feature appears) and death time (i.e., the value of the radius at which a topological feature disappears) of a topological feature. Persistence barcodes capture the interval between the birth and death of a topological feature. There are several off-the-shelf software packages available to compute PH like GHUDHI, R-TDA, Dipha, etc [Pun *et al.*, 2018] which can be chosen based on the language of preference.
- *How to interpret the barcodes for bias detection:* If there is a single long barcode, there is bias. If all barcodes are of same length, there is no bias. See Figure 6.
- *How to validate bias:* Use non-parametric permutation test to show that the PH of the predictor contributing to bias is distinct from other predictors.

6 Conclusions

Topological data analysis offers promising alternate feature representation techniques. In this work, we described a novel way of quantifying bias in datasets. Specifically, we used persistence homology to determine bias due to different attributes in the German credit dataset and validated the same using non-parametric statistical permutation tests. The proposed visualization can serve as a useful pre-processing tool for software engineers to understand which attributes need to be accounted for and mitigated when ensuring fairness in classification.

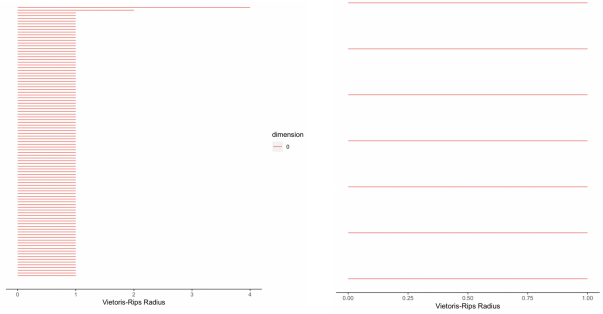

Figure 6: Right: An instance of no bias since all barcodes are of same length. Left: An instance of bias due to the presence of long barcodes at the top

References

[Al-Rubaie and Chang, 2018] Mohammad Al-Rubaie and Morris Chang. Privacy preserving machine learning - threats and solutions. *IEEE Security and Privacy Magazine*, 2018.

[Bellamy *et al.*, 2018] Rachel Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, Seema Nagar, Karthikeyan Natesan Ramamurthy, John Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Ramazon Kush, and Yunfeng Zhang. Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. 2018.

[Bostrom and Yudkowsky, 2018] Nick Bostrom and Eliezer Yudkowsky. The ethics of artificial intelligence. *Cambridge Handbook of Artificial Intelligence*, 2018.

[Carlsson and Gabrielsson, 2018] Gunnar Carlsson and Richard Gabrielsson. Topological approaches to deep learning. *ArXiv*, 2018.

[Challen1 *et al.*, 2019] Robert Challen1, Joshua Denny, Martin Pitt, Luke Gompels, Tom Edwards, and Krasimira Tsaneva-Atanasova1. Artificial intelligence, bias and clinical safety. *BMJ Quality and Safety*, 2019.

[Chen *et al.*, 2018] Irene Y. Chen, Fredrik D. Johansson, and David Sontag. Why is my clasifier discriminatory. *NeurIPS*, 2018.

[Dua and Graff, 2019] D. Dua and C. Graff. Uci machine learning repository. *University of California, School of Information and Computer Science, Irvine*, 2019.

[FSB, 2017] FSB. Artificial intelligence and machine learning in financial services: Market developments and financial stability implications. *Technical Report: Financial Stability Board*, 2017.

[Gabrielsson and Carlsson, 2018] Richard Gabrielsson and Gunnar Carlsson. Exposition and interpretation of the topology of neural networks. *ArXiv*, 2018.

[Gebru *et al.*, 2018] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumeé III, and Kate Crawford. Datasheets for datasets. *Proceedings of the 5 th Workshop on Fairness, Accountability, and Transparency in Machine Learning*, 2018.

[Goodfellow *et al.*, 2014] I. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *ICLR*, 2014.

[Gunnar, 2018] Carlsson Gunnar. Going deeper: Understanding how convolutional neural networks learn using tda. *Ayasdi Blog*, 2018.

[Kleinberg *et al.*, 2018] Jon Kleinberg, Himabindu Lakkaraju, Jure Leskovec, Jens Ludwig, and Sendhil Mullainathan. Human decisions and machine predictions. *The quarterly journal of economics*, 2018.

[Knight, 2018] Will Knight. Microsoft is creating an oracle for catching biased ai algorithms. *MIT Technology Review*, 2018.

[Kurutach *et al.*, 2018] Thanard Kurutach, Aviv Tamar, Ge Yang, Stuart Russell, and Pieter Abbeel. Learning plannable representations with causal infogan. *NeurIPS*, 2018.

[Li *et al.*, 2014] C. Li, M. Ovsjanikov, and F. Chazal. Persistence-based structural recognition. *CVPR*, 2014.

[Nguyen *et al.*, 2018] D. D. Nguyen, Z. X. Cang, K. D. Wu, M. L. Wang, Y. Cao, and G. W. Wei. Mathematical deep learning for pose and binding affinity prediction and ranking in d3r grand challenges. *ArXiv*, 2018.

[Pachauri *et al.*, 2011] D. Pachauri, C. Hinrichs, M.K. Chung, S.C. Johnson, and V. Singh. Topology-based kernels with application to inference problems in alzheimer's disease. *IEEE Transactions on Medical Imaging*, 2011.

[Peters, 2018] Adele Peters. This tool lets you see–and correct–the bias in an algorithm. *Fast Company*, 2018.

[Pettigrew *et al.*, 2018] Simone Pettigrew, Lin Fritschi, and Richard Norman2. The potential implications of autonomous vehicles in and around the workplace. *Int J Environ Res Public Health.*, 2018.

[Pun *et al.*, 2018] Chi Pun, Kelin Xia, and Si Lee. Persistent-homology-based machine learning and its applications – a survey. *ArXiv*, 2018.

[Ramakrishan and Shah, 2016] R. Ramakrishan and J. Shah. Towards interpretable explanations for transfer learning in sequential tasks. *AAAI Spring Symposium Series*, 2016.

[Srivastava and Rossi, 2018] B. Srivastava and F. Rossi. Towards composable bias rating of ai systems. *AIES*, 2018.

[Stilgoe, 2017] Jack Stilgoe. Machine learning, social learning and the governance of self-driving cars. *Social Studies of Science*, 2017.

[Umeda, 2017] Y. Umeda. Time series classification via topological data analysis. *Information and Media Technologies*, 2017.

[Vallender, 1974] S.S. Vallender. Calculation of the wasserstein distance between probability distributions on the line. *Theory of Probability and its Applications*, 1974.

[Wadhwa *et al.*, 2018] Raoul Wadhwa, Drew Williamson, Andrew Dhawan, and Jacob Scott. Tdastats: R pipeline for computing persistent homology in topological data analysis. *Journal of open source software*, 2018.

[Wang *et al.*, 2011] B. Wang, B. Summa, V. Pascucci, and M. Vejdemo-Johansson. Branching and circular features in high dimensional data. *IEEE Transactions on Visualization and Computer Graphics*, 2011.

[Weintraub, 2014] Steven Weintraub. Fundamentals of algebraic topology. *Springer*, 2014.

[Wexler, 2018] James Wexler. The what-if tool: Code-free probing of machine learning models. *Google AI blog*, 2018.

[Zeppelzauer *et al.*, 2018] M. Zeppelzauer, B. Zielinski, M. Juda, and M. Seidl. A study on topological descriptors for the analysis of 3d surface texture. *CVIU*, 2018.

[Zhou *et al.*, 2017] Z. Zhou, Y. Z. Huang, L. Wang, and T. N. Tan. Exploring generalized shape analysis by topological representations. *Pattern Recognition Letters*, 2017.

[Zhu, 2013] X. J. Zhu. Persistent homology: An introduction and a new text representation for natural language processing. *IJCAI*, 2013.