On Adversarial Machine Learning and Homology

Joshua Ackerman

October 30, 2020

1 Introduction

Deep learning is increasingly being studied and deployed in the context of numerous critical industries such as healthcare, cybersecurity and many others [10][4]. While deep learning may enable an unprecedented level of automation in these industries, it also creates a new class of security vulnerabilities that malicious actors can use to disrupt them [13]. However, the concept of adversarial input, is of course not unique to deep learning models, and are frequently studied in traditional computer systems e.g., buffer overflow attacks, SQL injection etc. One school of thought to address this issue in the context of traditional computer systems, is that of *language theoretic security* (LANGSEC). In short, LANGSEC is the idea that the "internet insecurity epidemic" is caused by the improper handling of input, and that the path towards creating trust worthy software involved modeling expected inputs as a formal language [1].

We support a similar approach to securing machine learning (ML) systems. In ML, the structure of the data is often formalized and rationalized not by formal language theory, but by topology. Works like [5] and [9] suggest a strong connection between topological properties of the data, and the architecture required to properly classify points. Further, works such as [7][11] have begun to explore how to leverage data manifold information perspective in the context of adversarial machine learning. They propose the strategy of manifold based defenses, which roughly speaking involves learning the data manifold, and pulling new points onto the manifold. In [8], the authors explore the question of how much this defense relies on understanding the topology of the manifold. Their findings suggest that a topology-aware model is needed, and in fact improves the effectiveness of a particular type of manifold-based adversarial defense.

We are broadly interested in the application of these topological methods towards creating more robust machine learning methods. However our interest stems from security applications where we wish to reliably classify or generate syntactic data that is, in principle, derived from an unknown formal grammar, but nevertheless is given as points in space. To apply traditional machine learning to this task, the data must be numerically encoded, giving rise to an interesting, but somewhat muddy situation that is not clearly addressed by any of the aforementioned frameworks or papers. Consequently, in this project we wish to investigate topology-aware generative models in the context of datasets derived from formal languages. While we likely will not have time in this particular project to explore the adversarial motivation of this, our overall goal is to perform a series of exploratory experiments to enable this in future work. In the next section, we will briefly introduce lesser known concepts needed to understand our proposal, and supply a more detailed plan in the next.

2 Background

Two central types of machine learning models are those of discriminative and generative models. Discriminative models attempt to derive a model $p(\mathbf{y}|\mathbf{x})$ from a dataset $\mathcal{X} = \left\{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\right\}_{i\in[n]}$, where $\mathbf{x}$ is a point from the data distribution $p(\mathbf{x})$, and $\mathbf{y}$ is a label (class) corresponding to that data point. In our context, generative models, learn the data distribution from a collection of points $\mathcal{X}$ i.e., $p(\mathbf{x})$. With these basic principles, we can define the aforementioned manifold-based defense, which is more precisely called *invert and classify* (INC). Given a classifier C, a generative model of the data G, and a novel datapoint to classify $\mathbf{x}$, INC first finds the projection of $\mathbf{x}$ into the range of G,

$$\mathbf{z}^* = \arg\min_{\mathbf{z}} ||\mathbf{x} - G(\mathbf{z})||$$

and then simply outputs $G(C(\mathbf{z}^*))$ [7].

In practice, the choice of G and C are extremely relevant, as basic architecture choices affect the expressiveness of the model and how the model generalizes to novel points. We choose G to be a particularly nice generative model called a *normalizing flow*. Normalizing flows, model $p(\mathbf{x})$ by learning a sequence of simple and invertible transformations $f = f_1 \circ f_2 \circ \cdots \circ f_D$ from $p(\mathbf{x})$ to a tractable density function $p_Z(\mathbf{z})$ such as the multivariate Gaussian distribution, $\mathcal{N}(\mathbf{z}; 0, \mathbf{I})$. We can use the change of variables formula to easily get a tractable optimization expression for $p(\mathbf{x})$,

$$\begin{aligned}\log p(\mathbf{x}) &= \log p_Z(\mathbf{z}) + \log\left(\left|\det \frac{\partial \mathbf{z}}{\partial \mathbf{x}}\right|\right) \\ &= \log p_Z(\mathbf{z}) + \sum_{i\in[D]} \log\left(\left|\det \frac{\partial f_i}{\partial f_{i-1}}\right|\right)\end{aligned}$$

which can be optimized with traditional maximum likelihood estimation. A common design for f_i, introduced by Dinh et al. [3], is the affine layer which partitions a given input $\mathbf{x} = [\mathbf{x}_a \ \mathbf{x}_b]$ and computes

$$\begin{aligned}\mathbf{z}_a &= \mathbf{x}_a \\ \mathbf{z}_b &= \mathbf{s}(\mathbf{x}_a) \odot \mathbf{x}_b + \mathbf{t}(\mathbf{x}_a) \\ \mathbf{z} &= \mathbf{z}_a || \mathbf{z}_b\end{aligned}$$

where $\mathbf{s}, \mathbf{t}$ are neural networks, $||$ denotes concatenation, and $\odot$ is element-wise multiplication. For further details see [CITE]. Since our data is intrinsically discrete, we dequantize it with uniform noise $\mathbf{u} \sim \text{Unif}[0,1]^n$ as $\mathbf{x} \leftarrow \mathbf{x}+\mathbf{u}$ [12]. Time permitting for C, we will choose a generic Long short-term memory network (LSTM) [6].

Moving on, we assume general knowledge of formal grammars, as these are often taught in undergraduate theory of computation courses. One less common class of languages, that is central to our investigation are Dyck languages.

Definition 1 (Dyck Languages) *Given a bipartite set of characters $(P, \bar{P})$, the Dyck language, $\mathcal{D}_P$, is defined by the set,*

$$\mathcal{D}_P = \left\{x \in (P \cup \bar{P})^* \mid x \text{ is a well balanced set of parenthesis}\right\}.$$

Although these languages seem simple or niche, the Chomsky–Schützenberger representation theorem suggests a strong connection between these languages and context-free ones. The relationship is formalized via the notion of a *homomorphism*, which for two alphabets Σ, Δ, is a map $h : \Sigma^* \mapsto \Delta^*$ such that $h(\epsilon) = \epsilon$, and for any $x, y \in \Sigma$, $h(xy) = h(x)h(y)$ [1].

Theorem 1 (Chomsky–Schützenberger Representation Theorem) *A language L over Σ is context-free if and only if there is a bipartite set of characters $(P, \bar{P})$, a regular language R over $(P, \bar{P})$, and a homomorphism $h : (P, \bar{P})^* \mapsto \Sigma^*$ such that $L = h(\mathcal{D}_P \cap R)$.*

3 Plan

The quality of the underlying generative model G is paramount for this defense to be effective. Follow up work have managed to find vulnerabilities in practical deployments of this approach [7], and as we mentioned earlier [8] argues that topological awareness of the model is critical to effectively use this defense. First and foremost, we would like to empirically study the homology of Dyck languages using persistent homology to gain a sense of the topological richness of a general class of syntactic data. We will use dimension reduction methods like Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP) to more effectively use persistent homology. Next, we will perform a study on architecture selection in normalizing flows analogous to [5]. We generate toy datasets with increasing homological complexity, and vary flow parameters such as the flow depth, and the hyperparameters of the neural networks parameterizing the affine coupling layers. With any time permitting, we will explore how this knowledge can improve invert and classify defenses. However, given the time limitations we expect this to be the subject of future work.

[1] This paragraph is derived from a previous survey paper of mine [2]

References

[1] Langsec: Language-theoretic security "the view from the tower of babel". `http://langsec.org`. Accessed: 2020-10-30.

[2] Joshua Ackerman and George Cybenko. A survey of neural networks and formal languages, 2020.

[3] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation, 2015.

[4] Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. A guide to deep learning in healthcare. *Nature Medicine*, 25(1):24–29, 2019.

[5] William H. Guss and Ruslan Salakhutdinov. On characterizing the capacity of neural networks using algebraic topology. *CoRR*, abs/1802.04443, 2018.

[6] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[7] Andrew Ilyas, Ajil Jalal, Eirini Asteri, Constantinos Daskalakis, and Alexandros G. Dimakis. The robust manifold defense: Adversarial training using generative models. *CoRR*, abs/1712.09196, 2017.

[8] Uyeong Jang, Susmit Jha, and Somesh Jha. On the need for topology-aware generative models for manifold-based defenses, 2020.

[9] Gregory Naitzat, Andrey Zhitnikov, and Lek-Heng Lim. Topology of deep neural networks, 2020.

[10] Thanh Thi Nguyen and Vijay Janapa Reddi. Deep reinforcement learning for cyber security. *CoRR*, abs/1906.05799, 2019.

[11] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *CoRR*, abs/1805.06605, 2018.

[12] L. Theis, A. Oord, and M. Bethge. A note on the evaluation of generative models. *CoRR*, abs/1511.01844, 2016.

[13] Han Xu, Yao Ma, Haochen Liu, Debayan Deb, Hui Liu, Jiliang Tang, and Anil K. Jain. Adversarial attacks and defenses in images, graphs and text: A review, 2019.

Maxwell Carmichael
COSC049
Professor Hsien-Chih Chang
October 30, 2020

Using Topological Features of Tonnetz for Music Analysis

What is Tonnetz?

Tonnetz is a lattice diagram representing tonal space, specifically of equal-tempered music, the form of virtually all Western music. The faces of each triangle represent either a major or minor chord, the vertices the notes of the chord. Any triange always shares two vertices with its adjacent faces, with the remaining vertex of the adjacent face completing the face's triad, either a minor chord if adjacent to a major face or a major chord if adjacent to a minor face. In the planar model, Tonnetz extends indefinitely in all directions.

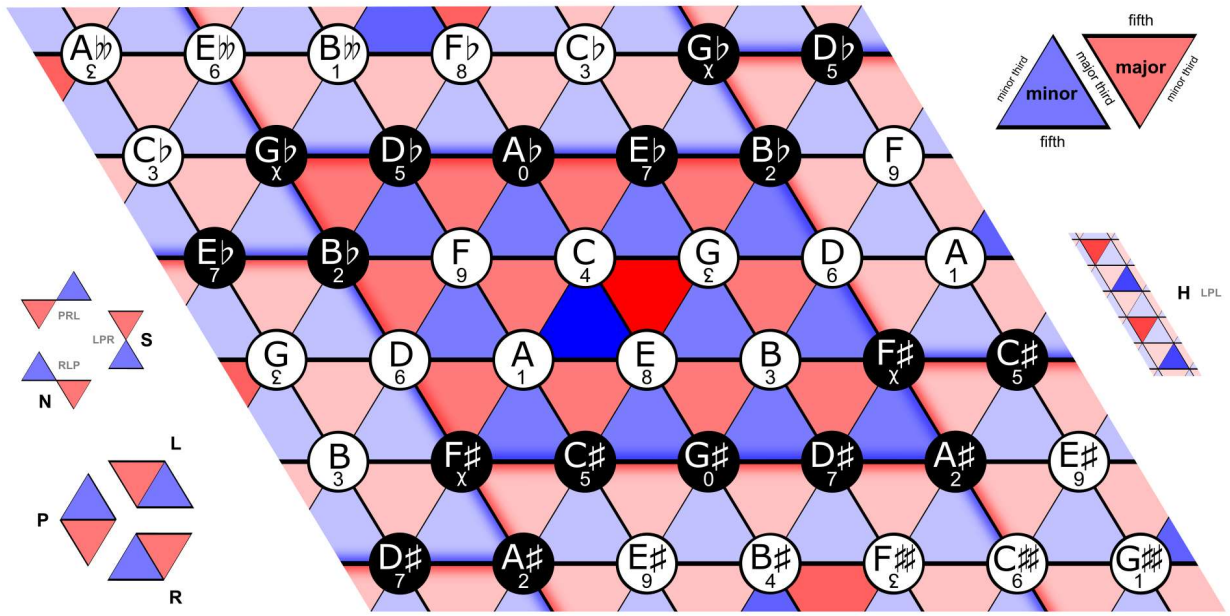


Image Source: Wikipedia

Geometric Modeling of Preferred Directions using Tonnetz

Given a particular piece or excerpt of a piece, we can use geometry to model the preferred directions of the piece, which, in musical terms, can be defined as the core concepts representing the main ideas of a piece. We can weight each vertex by the frequency it appears in an excerpt and then generate a (bounded) Tonnetz graph with an added z dimension of frequency, like the graph below.

Image Source: Dynamic and Topological Tools for (Modern) Musical Analysis. A portion of the planar Tonnetz is represented on a plane.

Topological Features and Project Goals

We can generate "preferred subcomplices" for any given piece or excerpt, which represent the fundamental domain of a piece, a portion of the edges of Tonnetz. With these preferred subcomplices, we can use them to classify compositions. There are limitations to this approach, as it prevents us from distinguishing the characteristics of two pieces which may have the same preferred subcomplex, which (Bergomi 2015) attempts to overcome using persistent homology. However, in this project I am more interested in analyzing the topological features of the subcomplices from a compositional standpoint rather than analyzing audio from existing pieces. Particularly, which topological features are associated with calm vs. exciting, happy vs. sad, majestic vs. playful, etc.

Another potential thing to consider along the same lines is the dual graph of Tonnetz, with chords as vertices (instead of faces). With cyclical chord structures, we could find paths (with a little well-thought-out manipulation) in the dual graph with a direction, and perhaps we could find interesting features in the Gauss codes or other characterizations of the path. The most ambitious potential findings could range from being able to categorize pieces by our emotional perceptions to discovering that the chord structure is irrelevant to our emotional perceptions, which could rely solely on rhythm, tempo, instrumentation, dynamics, etc. This approach could take advantage of the fact that Tonnetz can be represented on a torus, and explore features of toroidal Gauss codes.

Literature

Dysart, Paul. *Tonnetz*, 2017, www.dysartp.com/.

Mattia Giuseppe Bergomi. Dynamical and Topological Tools for (Modern) Music Analysis. Computer Science [cs]. Università degli Studi di Milano; Université Pierre et Marie Curie, 2015. English. fftel01265574f

Random curves

Dylan Fridman

October 2020

Abstract

Two main proposals are detailed, both related to random curves. The first project idea is to try to come up with an efficient algorithm that given a maximum number of intersections returns a random valid planar curve (as a signed Gauss code) from a uniform distribution. The second project idea is to try to figure out what can we say about the surface in which a random curve is embedded just from the curve, with the curve given as a random signed Gauss code. Lastly, I mention some other interesting related questions.

1 Generating random curves uniformly

The idea is straightforward: let's try to find a reasonably efficient algorithm (hopefully $O(n)$?) to generate a random planar curve given an upper bound n on the number of intersections from a uniform distribution. The sample space of curves could be the set of signed Gauss codes of length less than or equal to $2n$. There are many related problems we could consider:

- Same but for the torus.
- Same but in any particular surface.
- Same but the sample space being of all orientable surfaces.
- Same but for a fixed number of intersections.
- Same but the sample space is made out of unsigned Gauss codes.

1.1 Relevant work

There is a lot of work done on how to generate random regular graphs, so that's a great place to start. It's not enough to consider an algorithm that randomly generates a regular graph in which every vertex has degree 4 because it could correspond to a multicurve instead of a curve. However, seems like a path that might be helpful to explore. Here are some relevant papers and surveys:

- Models of random regular graphs

- A sequential importance sampling algorithm for generating random graphs with prescribed degrees
- Generating random regular curves

1.2 Ideas on how to attack the problem

One possible approach is to understand the algorithms that generate random regular graphs and modify them so as to avoid getting graphs that correspond to multicurves. I can't make a value judgement of the approach because of my current ignorance on the topic.

The first thing I was planning on doing was writing a code that spits out the signed Gauss codes that correspond to planar curves up to some length, and try to see if there is some kind of structure.

Also, I was thinking on reading up on algorithms that generate random elements uniformly, because I haven't studied an algorithm of this kind before.

2 Guessing the surface from a random curve

Suppose you are standing on an orientable surface but you have no idea what the genus of the surface is. You are approached by a Sphinx that tells you that she is going to eat you unless you figure out what's the surface you are living in. She then picks a number n bigger than the genus of the surface randomly and after that a random signed Gauss code with length $2n$ realizable on the surface you are in (from a uniform distribution). She gives you the random signed Gauss code, in order to help you out. What can we say about the probability distribution of the genus? In particular, what should you tell the Sphinx? How good are your odds of surviving?

3 Other related questions

3.1 Is that a valid planar curve?

Given a natural number n, if we take a random string that has two copies of each symbol from a list of n different symbols (i.e. a random valid Gauss code), what's the probability that we can realize that curve on the plane? And on a torus? And on a genus-k orientable surface? What if we also give a random valid signing to the Gauss code? For the unsigned version of the problem, we could try using the conditions for planarity from Homework 1.

3.2 Are those regularly homotopic?

Given a natural number n and two random Gauss codes of length at most $2n$, what is the probability that the two curves are regularly homotopic?

Project Proposal

Yixin (Kathy) Lin

1 Goal

The primary goal of this project would be to understand the paper 'Min-Cost Flow in Unit-Capacity Planar Graphs' written by Adam Karczmarz and Piotr Sankowski [5], which give an $\tilde{O}((nm)^{\frac{2}{3}} \log C)$ time algorithm for computing min-cost flow (or min-cost circulation) in unit capacity planar multigraphs where edge costs are integers bounded by C.

2 Relevant Work

There are immense number of works on flows and min-cost flows, so this project will only concentrate on the ones that are relevant to the sparse and planar graph case.

Here, the fastest algorithms are based on planar graph duality and reduce the problem to shortest path computations. The undirected s, t-flow problem can be solved in $O(n \log \log n)$ time [3], whereas the directed s, t-flow problem can be solved in $O(n \log n)$ time [1, 2]. Even for the case with multiple source and sinks, a nearly-linear time algorithm us known [4].

3 Plan

The first step of this project would be to go through the paper 'Min-Cost Flow in Unit-Capacity Planar Graphs' [5] in details, and understand all the proves and algorithms. After that, go through relevant work, which are the another 4 papers listed in the references. Finally, try to come up with alternative proofs or algorithms.

The textbook 'Introduction to Graph Theory' by West might also be used in this project.[6]

References

[1] G. BORRADAILE AND P. N. KLEIN, *An $o(n \log n)$ algorithm for maximum st-flow in a directed planar graph*, J. ACM, 56 (2009).

[2] J. ERICKSON, *Maximum flows and parametric shortest paths in planar graphs*, In Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, (2010), pp. 794–804.

[3] P. S. GIUSEPPE F. ITALIANO, YAHAV NUSSBAUM AND C. WULFF-NILSEN, *Improved algorithms for min cut and max flow in undirected planar graphs*, In Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC, (2011), pp. 313–322.

[4] S. M. Y. N. GLENCORA BORRADAILE, PHILIP N. KLEIN AND C. WULFF-NILSEN, *Multiple-source multiple-sink maximum flow in directed planar graphs in near-linear time*, SIAM J. Comput., 46 (2017), pp. 1280–1303.

[5] A. KARCZMARZ AND P. SANKOWSKI, *Min-cost flow in unit-capacity planar graphs*, (2019).

[6] D. B. WEST, *Introduction to Graph Theory*, Pearson, 2015.

Computational Topology project proposal
What is the Kolmogorov complexity of a manifold?
Jason D. Linehan

0.0.1 Background

The Kolmogorov complexity $K(x)$ of a binary string x is the length of the shortest program p for a universal machine U such that $U(p) \downarrow = x$.

While Kolmogorov complexity was originally defined only for finitely-representable objects, it has been extended to real numbers [6] by defining a basis of representation, for example the rationals or arbitrary precision floating point numbers, and defining the Kolmogorov complexity of a real number to be the limit of the Kolmogorov complexities of a sequence of numbers in that basis of representation which converge to the real.

This was taken a step farther by Farjudian [2], who defined a notion of Kolmogorov complexity for a continuous real function, by defining the Kolmogorov complexity of the function to be the limit of the Kolmogorov complexities of a sequence of 'enclosures' – polynomials bounding the function – which converged to the function.

0.0.2 Complexity of a manifold?

Given these positive results, we might wonder if it's possible to find a natural-enough definition of the Kolmogorov complexity of a manifold?

I can imagine a couple of approaches to answering this question and providing a definition. First would be to restrict the question to a simple set of manifolds. The Kolmogorov complexity of the real line [4] has been studied. I can imagine doing similar for a simplicial complex.

Next, the idea would be to follow the heuristic set out by Farjudian, and select a finitely-representable sequence of objects which converge to the manifold, and then define the Kolmogorov complexity of the manifold as the limit of the Kolmogorov complexities of the objects in that sequence.

Alternatively, there may be folklore results on this already. Finding them would be interesting (to me, at least).

0.0.3 Why might this be interesting?

It might be handy to have a theoretical notion of the complexity of a manifold due to the fact that manifold learning is more or less pervasive in machine learning today, and the established and growing links between algorithmic information theory – where Kolmogorov complexity is a key tool – and both compression and learning.

One example of something where I would personally feel happier with a notion of Kolmogorov complexity of a manifold is when thinking about the manifold hypothesis, which is the statement that high-dimensional data collected from "real-world" observations tend to lie near a low-dimensional manifold embedded within the high-dimensional space.

The truth of the manifold hypothesis is seen to underpin the success of various dimensionality reduction techniques in machine learning, and it kind of reminds me of a sentiment expressed by Kolmogorov in his "Three Approaches to the Quantitative Definition of Information," [5] namely,

> "... what real meaning is there, for example, in asking how much information is contained in "War and Peace"? Is it reasonable to include this novel in the set of "possible novels," or even to postulate some probability distribution for this set?"

The manifold hypothesis seems to hold that yes, there is a set (a topological space, even) of possible novels, and that to the extent that one can train a machine to classify novels written by Tolstoy using, e.g., `word2vec`, then there is a probability distribution on this set, which concentrates its measure on the sub-manifold near which all of Tolstoy's novels lie.

So then, relating this again to Kolmogorov complexity, would a predicate P that a binary classifier learns, plus some additional information, be a description of a low-dimensional manifold? How are the complexities of an effective procedure for computing P, the ambient feature space, the novels or data points in question, and this low-dimensional manifold, related?

I would expect that the Kolmogorov complexity of any Euclidean space is rather low. I would also expect that the Kolmogorov complexity of the low-dimensional manifold picked out by some "natural" predicate would also be low-ish. For example, spheres, torii, and Klein bottles, all of which arise in practice [1]. In such cases, it would seem that the Kolmogorov complexity of a point on the manifold could be studied using the two-part description $K(M) + \log(|M|)$, where M is the manifold. However this is obvious nonsense – $|M|$ is infinite. A related question, then: is there an analogy to the two-part description for the extended definition of Kolmogorov complexity on the reals, which is not obvious nonsense?

I could imagine that if there were such an analogy, then it would become possible to argue using the two-part description, and the notion of Kolmogorov complexity of the manifold, for an explanation of at least *why* the manifold in the manifold hypothesis should be of a certain complexity, if not a lower dimension. Perhaps it is natural that certain kinds of manifold have low Kolmogorov complexity and that if the data did *not* lie on a manifold, or near such a one, then they would need to be algorithmically random. Something like this.

0.0.4 Conclusion

The extension of the definition of Kolmogorov complexity to real manifolds seems possible (if not already out there in folklore somewhere). It could be useful for various theoretical arguments of more or less interest, but the definition either way seems certain to involve discrete structures of interest in computational topology and in that sense may be educational.

As a bonus, might allow messing around with the "strange" correspondence given in [3] which connects inequalities of Kolmogorov complexity with volumes of space and the Cauchy-Schwarz inequality (interesting short read).

References

[1] Carlsson, G. Topology and data. *Bulletin of The American Mathematical Society - BULL AMER MATH SOC 46* (04 2009), 255–308.

[2] Farjudian, A. On the kolmogorov complexity of continuous real functions. In *Models of Computation in Context* (Berlin, Heidelberg, 2011), B. Löwe, D. Normann, I. Soskov, and A. Soskova, Eds., Springer Berlin Heidelberg, pp. 81–91.

[3] Hammer, D., and Shen, A. A strange application of kolmogorov complexity. *Theory Comput. Syst. 31* (02 1998), 1–4.

[4] J. Cai, J. H. The complexity of the real line is a fractal. *Proceedings. Structure in Complexity Theory Fourth Annual Conference* (1989), 138–146.

[5] Kolmogorov, A. N. Three approaches to the quantitative definition of information. *International Journal of Computer Mathematics 2*, 1-4 (1968), 157–168.

[6] Staiger, L. The kolmogorov complexity of real numbers. *Theoretical Computer Science 284*, 2 (2002), 455 – 466.

Since the first problem, especially because I enjoyed working on the planar version of the problem, I became really interested in the final section of question 1: Can you provide such characterization for Gauss codes encoding curves on the torus? I think it's a really interesting problem especially because it feels like a completely novel approach would be necessary because the Dehn Crossing strategy feels fundamentally incompatible with this version of the problem.

Obviously, to get started on it I would want to read all of literature about the original problem as well as the literature which exists about toroidal curves and whatever invariants exist in that space. For instance, I found a doctoral thesis on the subject that I have begun exploring (Laat, D. de.) I hope for the project would be to solve it, obviously, but perhaps less ambitiously I would like to understand all existing work on the problem and feel like I've made some progress either towards a solution or in related invariants.

One strategy that Dylan and I have been considering was adjusting your focus from the curve being a distortion of the trivial loop to the curve being a path taken and viewing it as a process of making new crossings instead of a finished object. I have to keep her step would be to understand exactly what types of features the torus loop adds, obviously it removes the need for many of the restrictions we had such as the parity of the gaps between like terms stuff Taurus has the handle which can simply act as a bridge to skip any crossing.

Works Cited

Laat, D. de. *Contractibility and self-intersections of curves on surfaces*. Diss. Faculty of Science and Engineering, 2009.

Goal: Conduct a survey of and perhaps produce a small contribution to the application of **algebraic topology to deep learning**.

Bastien Rieck's blog here is a great resource for modern work in this domain: https://bastian.rieck.me/blog/ . Of particular interest is Rieck et al's paper on "**Neural Persistence**." Their initial observation is that in order to make any real sense of the structure of a trained neural network, our only real approach is to "interrogate" the network with input data. As we all know, most specialized neural nets are playing with a severe drought of labelled data to begin with, and even if we were able to perform this interrogation we won't have obtained any theoretical generality.

Enter neural persistence, "a novel measure for characterizing neural network structural complexity. In doing so, we adopt a new perspective that integrates both network weights and connectivity while not relying on interrogating networks through input data." They show that this measure correctly identifies networks that employ best practices such as dropout and batch normalization. . Personally I find this extremely interesting because deep learning is always seen as a blackbox, hyperparameter tuning is just exhaustive searching for something that works, and other major architecture decisions often seem like lucky random guesses. To actually have a measure we can efficiently compute that can identify effective architecture decisions is kind of amazing.

So why is this relevant to this class? Neural persistence is built upon persistent homology, and the paper begins with a quick recap of simplicial complexes and persistent homology. Things become quite a bit more complicated from there, meaning that a proper deep dive on this paper is required to really understand how everything works, and I think that makes a survey of this paper and its related works a solid project. If such a survey is conducted with reasonable time remaining, there are suggested future works in the discussion section that we might at least be able to throw a few ideas at.

I for one will obtain a lot of value from fully unravelling this paper, but there are some other algebraic topology papers for deep learning that can be found around Rieck's blog, some of the more interesting ones might be Topological Autoencoders (https://arxiv.org/abs/1906.00722), Topological Approaches to Deep Learning (https://arxiv.org/pdf/1811.01122.pdf), Topological Machine Learning with Persistence Indicator Functions (https://bastian.rieck.me/research/TopoInVis2017_Persistence_Indicator_Functions.pdf), I could go on. There's a lot of these, and since I think most grad students use deep learning for one thing or another, I imagine I won't be alone in wanting to dissect these papers. And even the students who don't choose this project will benefit when we present our survey of these works.

Representing Mueller Matrices as Geometric Transformations on the Poincaré Sphere

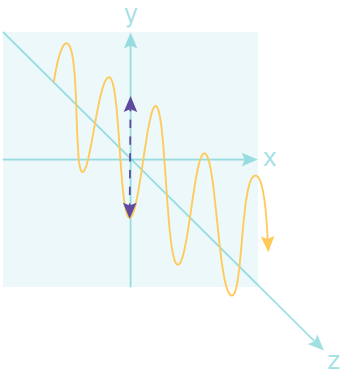
Kate Salesin

October 30, 2020

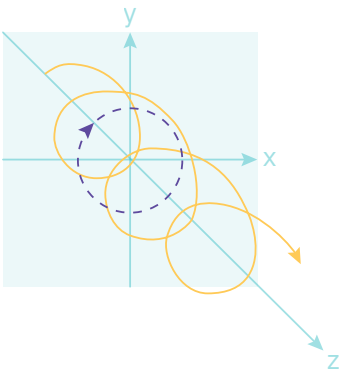
1 Goal

A fascinating and oft-overlooked property of light is its polarization. The polarization state of light may be represented mathematically as a Stokes vector, and may be transformed to another polarization state by a Mueller matrix. These matrices represent interactions of light with real-world materials, such as light passing through a linear polarizer, bouncing off a surface, or scattering within a turbid medium. These Stokes vectors, once power-normalized, may be visualized as vectors on and within the Poincaré sphere. The goals of this project is to investigate how Mueller matrices correspond to geometric transformations of Stokes vectors on the Poincaré sphere.

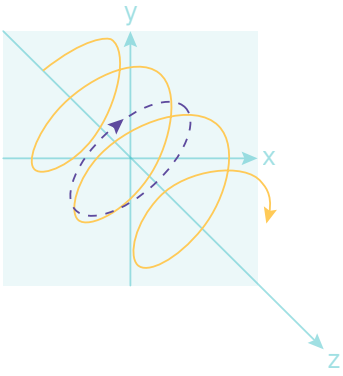
2 Background



Linear polarization



Circular polarization



Elliptical polarization

The polarization state of a beam of light is totally encompassed by a 4-component Stokes vector:

$$\vec{S} = \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{bmatrix} = \begin{bmatrix} S_0 \\ H - V \\ 45^\circ - 135^\circ \\ R - L \end{bmatrix} = \begin{bmatrix} S_0 \\ \leftrightarrow - \updownarrow \\ \nearrow - \searrow \\ \circlearrowleft - \circlearrowright \end{bmatrix}, \tag{1}$$

where S_0 is the total radiance, S_1 is the amount of light which is horizontally vs. vertically polarized, S_2 is 45° vs. 135° polarized, and S_3 is right-hand vs. left-hand circularly polarized. Each of S_1, S_2, and S_3 lies

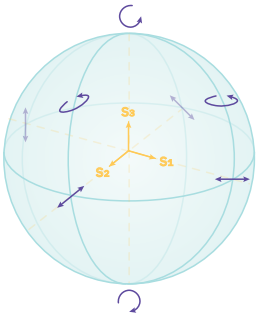
in the domain $[-S_0, S_0]$, where $S_0 \geqslant 0$. The Stokes vector of light may be easily measured experimentally by passing the light through various polarizers.

The degree of polarization of the light is $P = \frac{\sqrt{S_1^2+S_2^2+S_3^2}}{S_0}$. P ranges from 0 to 1, where 0 is completely unpolarized light, 1 is completely polarized light, and between 0 and 1 is partially polarized light. Any 4-vector that produces $P > 1$ is not a physically valid Stokes vector. Note that completely polarized light may be linearly polarized, circularly polarized, or elliptically polarized (the general case).

If we normalize the Stokes vector by the total radiance, we are left with a 3-coordinate vector which may be plotted as Cartesian coordinates in 3D space:

$$\vec{s} = \begin{bmatrix} S_1/S_0 \\ S_2/S_0 \\ S_3/S_0 \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix}. \tag{2}$$

The space of $\vec{s}$ spans a solid unit sphere known as the the *Poincaré sphere*. When $|\vec{s}| = 1$, the light is fully polarized and the point lies on the boundary of the sphere. When $|\vec{s}| < 1$, the light is partially polarized and the point lies within the sphere.



Any point along the equator of the sphere corresponds to light that is linearly polarized at some angle in $[0°, 180°]$, where $0°$ corresponds to horizontal polarization and $90°$ to vertical polarization. The North Pole of the sphere is light that is right circularly polarized and the South Pole is light that is left circularly polarized. Any point on the boundary of the sphere not on the equator or two poles is elliptically polarized.

3 Methods

From here, we ask how does a Mueller matrix transform an incoming $\vec{s}$ into an outgoing $\vec{s}'$? What geometric effects do different types of Mueller matrices have (e.g. linear polarizer, wave retarder, diffuse surface)?

Past studies[1] have mapped the outer boundary of the Poincaré sphere to the ellipsoid produced by a Mueller matrix, but the treatment is highly technical and only discusses the properties of the resulting ellipsoids, not the geometric transformations producing them (e.g. scale, rotation).

In this study, we hope to first replicate their results and discuss them in more familiar terms. Then, we hope to conduct further studies on the effects of each matrix comprising the polar decomposition of a Mueller matrix. We may also investigate the effects of the Mie (suspended sphere) Mueller matrix, which has complex polarization properties and many sharp peaks and valleys.

[1] Poincaré sphere mapping by Mueller matrices. J. Opt. Soc. Am. A. 2013. https://doi.org/10.1364/JOSAA.30.002291

COSC 149: Project Proposal

Manuel Stoeckl

October 30, 2020

Project Goal

To prove that there exists a "blind" human strategy to acquire a puppy in the chasing puppies game that works for all simple closed curves, or to find a family of paths on which there is no such strategy.

In the "chasing puppies" game, a human and puppy are constrained to a simple closed curve γ in $\mathbb{R}^2$. The human location is parameterized by $h \in S^1$, and the puppy location by $p \in S^1$. The human may move arbitrarily along the curve; in response to any infinitesimal human motion, the puppy always moves along the curve in the direction of decreasing Euclidean distance to the human, until it either reaches the human or gets stuck at a point on curve where any motion would increase the distance to the human. The goal of the game is, given some initial γ, h, p, for the human to make a pattern of moves so that the puppy returns to it.

A blind strategy for the human is a pattern of relative motions in S^1 for which, no matter the curve γ, or initial values of h and p, at the end of the pattern, the human will have recovered the puppy.

Relevant Work

The "chasing puppies" problem was studied by Erickson et al., `https://www.youtube.com/watch?v=Ysk0yhO4jVk`; in the talk, Erickson conjectures that a blind human strategy of making two clockwise traversals of the curve, followed by two counterclockwise traversals, always works.

It is similar to pursuit-evasion games, which are perhaps best remembered by the popular article "How the Lion Tamer was Saved"[1], but have been studied since the 1950s.[2] Since e.g. fixed ratios of speeds between players are often involved, these generally have geometric and casework heavy solutions. (In comparison, the puppy of the chasing puppies game instantaneously reacts to any human motion; furthermore, since the puppy is deterministic, the chasing puppies problem isn't technically a game.)

[1] `http://www.appliedprobability.org/content.aspx?Group=ms&Page=MS61`

[2] See e.g. Isaacs, Differential Games, John Wiley and Sons, 1965.

Plan

I do not yet have reason to believe that this problem cannot be solved using basic topological and geometric techniques, so right now I only plan to read a bit more about the properties and classification of simple closed curves.

There are a number of approaches, and sub-tasks to try to solve:

- Empirical approach: Look at all possible equivalence classes of curves, and determine what strategies work for them. It is also necessary to determine when to treat curves as equivalent; for example, does it suffice to check if the human diagrams have the same orderings of loops and path reversals? Is there a way to produce a curve matching a valid puppy diagram? This may require writing a bit of code, but the human/puppy diagram transformations given in the talk are straightforward to implement.

- Understanding behavior on non-smooth curves, and whether every smooth curve has a polygonal approximation on which the strategies are the same. Being able to reduce arbitrary curves to structured ones should be able to remove the "geometry" portions of the problem.

- Erickson mentioned that polygonal curves with axis-aligned segments had a blind strategy. Can this be reproved, and extended to e.g. having any three valid directions for edges, instead of just up/down and left/right?

- Test a number of examples, to see if one can generalize the ratchet/star/zigzag-like example to work in two directions. Does nesting such structures work? Is there a way to simplify curves so as to remove curve structures (i.e, a double zigzag) while preserving the nonexistence of a blind strategy?

0.1 Project Goal

This project is a survey of a specific topic and related ideas. In an ideal world, it would also be a novel research project, solving an interesting open problem, but I have no illusion that we will be able to do that with this specific problem in the next few weeks.

0.2 Relevant Problem

In 1911, Toeplitz ([5]) posed the Square Peg Problem: given any continuous Jordan curve γ, does it contain four points which form the vertices of a square? There is no restriction on this square; the vertices don't need to be in the same order on the square and γ, and the edges don't need to be on the interior or exterior of γ. This question is still open.

In 1929, Shnirleman ([4]) proved the Square Peg Problem in the case where γ is smooth. Since then, despite relaxing some regularity requirements (C^2 suffices, for example, instead of smoothness), no one has made serious progress on the continuous case. Despite the fact that any continuous curve is the limit of smooth curves, and each of these must contain a square, no one can deal with the problem that these squares might degenerate to a point in the limit.

Looking for generalizations or variants then, has led to progress in the last five years. The Rectangular Peg Problem: given any continuous closed curve γ and rectangle R, are there four points on γ which form the vertices of a rectangle similar to R? A positive answer would prove the Square Peg Problem.

In 2020, Josh Greene and Andrew Lobb proved the Rectangular Peg Problem ([1]), in the case where γ is smooth, in the affirmative. This was after recent progress by Hugelmeyer ([2],[3]).

0.3 My Plan

First off, we intend to provide a survey of the history of the Square and Rectangular Peg Problems, corresponding results for triangles and pentagons, and how the perspective on this problem has changed throughout time.

Second, the recent solution of the Rectangular Peg Problem is not constructive. It constructs a function whose self-intersections correspond to inscribed rectangles, then relies on an old result that the function cannot be an embedding, so it has a self-intersection. And, as a result, γ has an inscribed rectangle.

There is an open question about finding such rectangles: given a γ and a rectangle R, can we write an algorithm to find vertices of a rectangle similar to R on γ?

This is similar to Sperner's lemma, which took a nonconstructive theorem and provided a way to construct the solution; reproving the theorem along the way.

With no intentions of progress, we would love to consider this open question, and see if there are any avenues of attack.

Furthermore, we will analyze the complexity of the Square Peg Problem.

Bibliography

[1] Joshua Evan Greene and Andrew Lobb. The rectangular peg problem, 2020.

[2] Cole Hugelmeyer. Every smooth jordan curve has an inscribed rectangle with aspect ratio equal to $\sqrt{3}$, 2018.

[3] Cole Hugelmeyer. Inscribed rectangles in a smooth jordan curve attain at least one third of all aspect ratios, 2019.

[4] Lev Shnirleman. On some geometric properties of closed curves (in russian), 1929.

[5] Otto Toeplitz. Ueber einige aufgaben der analysis situs, 1911.

Andrew White

October 30 2020

Applications of Topology to Economics

With this project, I hope to apply some topological tools to economics. Specifically, I will endeavor to prove two facts that are taken as given in most of the econ courses here. The first is that there exists a price equilibrium in a competitive economy. The second is that there exists a Nash-Equilibrium in n-player games where mixed (probabilistic) strategies are allowed.

Both of these results were proved relatively recently, with Arrow proving the former in 1954, and Nash the latter in 1950. See:

https://www.jstor.org/stable/pdf/1907353.pdf

https://www.pnas.org/content/pnas/36/1/48.full.pdf.

The original articles are quite terse, so I intend to use a few other sources for these. I will begin by exploring the relevant sections in Ghrist's *Elementary Applied Topology*. See 4.27, 5.11:

https://www.math.upenn.edu/~ghrist/EAT/EATchapter4.pdf

https://www.math.upenn.edu/~ghrist/EAT/EATchapter5.pdf

Both of these results can be proved from Brouwer's fixed-point theorem, though it may be useful to explore a generalization, the Kakutani fixed-point theorem, for the existence of Nash Equilibrium:

https://en.wikipedia.org/wiki/Kakutani_fixed-point_theorem

This is the method Nash used in his proof. However, other proofs exist from first principles such as https://www.cs.ubc.ca/~jiang/papers/NashReport.pdf.

I envision the report component of this project to be theory-intensive, while the presentation portion would focus on some fun applications. A wealth of fascinating game-theoretic models can be found in Harrington's *Games, Strategies, and Decision Making*. Naturally, because the existence of a price-equilibrium is a necessary assumption for most economic models, I could provide applications from several textbooks.

Background

One topic of interest in computational topology is the efficient computation of the homology groups of a chain complex. Naive implementations of homology algorithms over arbitrary rings can have a running time that is exponential in the dimension of the chain complex. In special cases, however, we can do much better.

For example, in [2], Kaczynski et al. demonstrate an algorithm to compute the homology of a chain complex over a field in $O(n^3)$ time. While not explicitly defined this way, their algorithm can be thought of as an operation on edge-labeled directed graphs. Given a chain complex (C, ∂) over a field $\Bbbk$, we associate to it a graph $G = (V, E)$. Here, V is a set of generators for C and E contains an edge (x, y) labeled c iff cy is a summand of $\partial(x)$. Edges labeled 0 are thought of as not present in G. In the case that $\Bbbk = \mathbb{Z}/2\mathbb{Z}$, we can define G even more simply as "the graph with adjacency matrix $[\partial]$". Their algorithm can then be thought of as iterating an operation on G that "cancels" an edge, removing two vertices each repetition, until there are no edges remaining and thus $|V| = \dim H_*(C)$.

This idea of thinking of chain complexes as directed graphs with extra structure is rather common, and (coincidentally?) is a fundamental part of how some knot theorists work with and visualize chain complexes. In [4], for example, Zhan summarizes already well-known generalizations of the above algorithm to much more esoteric topological tools like Type D structures and DA-bimodules. More relevant to this project is [3], in which Levine describes a simple refinement of the algorithm to compute the pages of the spectral sequence associated to a filtered chain complex.

Proposal

My proposal centers around the generalization of the chain complex reduction algorithm to filtered complexes mentioned above. While it appears to be common knowledge in certain communities, as far as I know it is only ever referenced in passing, and not much is known about its exact behavior.

My specific questions are:

1. When reducing a filtered chain complex, the algorithm cancels edges in an arbitrary order. Can we characterize the effect that different edge cancellation orders has on the output graph?

2. Given the classical data of a spectral sequence, i.e. chain complexes (E^n, ∂^n) for all n, with isomorphisms $H_*(E^n, \partial^n) \to E^{n+1}$, can we compute the output of the reduction algorithm more efficiently?

To give a specific example, Khovanov homology and knot Floer homology are two knot invariants that are currently very popular to study. In [1] Dowlin constructs a filtered chain complex such that its associated spectral sequence has E^2 page isomorphic to the Khovanov homology of a knot, and E^∞ page isomorphic to the knot Floer homology of the same knot. It would be interesting if one could express the knot Floer homology of a knot as the differential of a chain complex with underlying module isomorphic to the Khovanov homology of the knot, and for this to all be "canonical" in some sense.

Bibliography

[1] Nathan Dowlin. A spectral sequence from khovanov homology to knot floer homology, 2018.

[2] T. Kaczyński, M. Mrozek, and M. Ślusarek. Homology computation by reduction of chain complexes. *Computers & Mathematics with Applications*, 35(4):59 – 70, 1998.

[3] Adam Simon Levine. Knot doubling operators and bordered heegaard floer homology. *Journal of Topology*, 5(3):651–712, Sep 2012.

[4] Bohua Zhan. Combinatorial proofs in bordered heegaard floer homology. *"Algebraic & Geometric Topology"*, 16(5):2571–2636, Nov 2016.

How complex is mbira music?

Project Proposal
By Linford J. Zirangwa ‘19
CS49 Fall ‘20

Project Goal

The mbira is a traditional instrument from Zimbabwe. The mbira is a lamellaphone, part of the plucked idiophone family of musical instruments. The mbira is played in religious ceremonies and festivities for the Shona people. There are artists from Zimbabwe, such as Jah Prayzah, Chiwoniso Maraire and Thomas Mapfumo who have managed to play the mbira commercially. By ear, the music compositions of the mbira do not seem complex, but usually they are played by an orchestra not one person. The goal of the project is to use persistent homology, to compare the complexities of classical compositions and mbira compositions to determine just how complex these mbira compositions are.

Relevant Work

This project is inspired by the paper, *Quantifying Music Complexities Using Topological Data Analysis,* by Kira Wencek (https://emerging-researchers.org/projects/20106/). In this paper, they used a persistent homology inspired algorithm to compare the complexities of music compositions.

Current Plan, Materials to read

Things I will need to read about:
MusicXML formats, R script, persistent homology, Vietoris-Rips Complex,
Some work related to this project:

- Edelsbrunner, H., Harer, J. (2009). Computational Topology: An Introduction. American Mathematical Society.
- Perea, J., Harer, J. (2013). Sliding Windows and Persistence: An Application of Topological Methods to Signal Analysis. eprint arXiv:1307.6188.
- Schuijer, M. (2008). Analyzing Atonal Music: Pitch-Class Set Theory and Its Contexts.
- University of Rochester Press. Zhu, X. (2013). Persistent Homology: An Introduction and a New Text Representation for Natural Language Processing. IJCAI 2013. 1953-1959.

I will have to find music compositions. The classical composition pieces will be easy to find. But the mbira music compositions might be hard to find. In the event, I don’t find the mbira compositions, I will look for other lamellaphone instrument compositions.