The Evolutionary Maps of Data

by

Abhinav Tamaskar

a dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

New York University

May, 2021

Bhubaneswar Mishra - Advisor

Dedicated to

My dearest brother Apoorva

Acknowledgments

The journey has been a long and hard one, there are many to thank for helping me finish this quest.

My advisor, Professor Bud Mishra, has played the most crucial role of being the guiding light in these past few years. He has been an amazing scientific mentor and has taught me the values and joys of interdisciplinary science. It was a joy working with a person brimming with ideas and knowledge across multiple domains of science. Moreover, he has also been an irreplaceable human guide in my life. When harsh realities have complicated life, it was through his encouragement and support that I have been able to trudge through these difficulties. I cannot thank him enough for all that he has done for me in the past few years.

From a scientific perspective, I would like to thank Professors Sylvain Cappell, Raul Rabadan and Oded Regev who have imparted invaluable skills for bringing this thesis to fruition. The varied set of interdisciplinary skills that I

have gained from each of them have been enormously helpful in tying together the vastly different areas of science. Their teachings play a central role in making this document readable.

This journey is never an easy one and along the way there are going to be many lows that can leave us feeling hopeless. Having someone to rely and and have as a support can make the most crucial difference. I'd like to my most special thank my younger brother, Apoorva, for being the pillar of emotional support who has held me together when the going got tough. His constant love, care and encouragement have made the last few years a lot less arduous.

Of course, no journey is complete without having a few friends along the way, some old and some new. I'd like to thank my friends Apurv, Drumil, James, Nikhil and Rijul for entertaining me during the long days and longer nights and making the last few years a blast.

Contents

List of figures

1

Introduction

In the recent years, evolutionary models for modeling populations[36,122] through interactive individuals have seen a rise in popularity, primarily spurred by the enormous increase in hardware capabilities and data collection efforts. Such models have seen considerable use in a wide variety of fields across the board such as advertisement recommendations[107], language evolutions[98], population genetics[80], single cell models for tumor growth[49], etc. And yet the primary tools used in these areas are still limited to the classical techniques from statistical learning like, graphical recommendation models, and ordinary[16,12], partial or stochastic differential equation(ODE/PDE/SDE) models[131], and have not yet seen a similar rise in geometric models for data as those for the deep learning and manifold learning fields.

This thesis presents a geometric and topological view of data science with a focus on population models, specifically extended to scenarios where the population is an interactive one and where the evolution of the population depends upon the information content of each individual. In this chapter, we go through some of the major challenges and questions that are still pending to model interactive populations with rich internal information. We give a brief overview to some of the techniques currently used for modelling interactive populations and show their strengths and shortcomings. We then go over the contributions of this thesis and outline how it tries to extend and enhance the current models using techniques from Topological Data Analysis and Manifold Learning for leveraging the geometric nature of data embeddings.

1.1 Modeling a population

Population modeling is a broad area with a rich history and several different types of models. Some of the models include

1. Differential equation models, including ODE, PDE, SDE, etc. These have extensively been used for modelling population growth[131].

2. Automata models, like Conway's Game of Life. These have been historically used to model interactive cellular systems but have seen a recent decline in popularity[109].

3. Recommendation systems. These have seen a rise in popularity due to their simple nature and ability to focus on a specified information content[16,12]. They have seen frequent use for ad recommendations in Google, movie recommendations in Netflix, Hulu, etc., buying recommendations in online stores, such as Amazon.

4. Deep neural networks. These are now being explored as replacements for some of the above systems but have not yet seen extensive independent use in any focused scenario[1,92].

In this section, we go through the details of two of the preceding models.

1.1.1 Differential Equation models

Using DEs for population modeling primarily focuses on analyzing the size of the population[131,75]. In such a scenario we study the size of a population mod-

eled using a proportional growth rate, $r \in \mathbb{R}_{\geq 0}$, along with a saturation limit, $k \in \mathbb{R}_{\geq 0}$, which can be seen as:

$$\dot{X} = rX\left(1 - \frac{X}{k}\right), \quad X(0) = 1,$$

where $X = X(t)$ is the population with respect to time and $\dot{X}$ is the rate of growth.

More complicated models used for investigating multiple subpopulations include the competition model[54], with two sub-populations X and Y, which has two parameters $a_{xy}, a_{yx} \in \mathbb{R}_{\geq 0}$ to represent the population pressure, i.e. the proportion of the populations that can change from X to Y and Y to X, respectively. Larger values of a_{xy}, a_{yx} implies a higher pressure that one subpopulation exhibits towards the other.

$$\dot{X} = r_x X\left(1 - \frac{X + a_{xy}Y}{k}\right), \quad \dot{Y} = r_y Y\left(1 - \frac{Y + a_{yx}X}{k}\right).$$

Even more complicated models utilize stochastic models such as one which addresses the accumulation of somatic mutations during the embryonic stage[52]. Frank, et al.[52], introduce a mathematical framework which examines two specific recessive mutations which are involved in colon cancer by considering the 4 population subtypes depending on the accumulation and the likelihood of a cell moving across a subtype during division.

DE models are good at scrutinizing the overall types in a population where there exist distinguished subtypes but they fail to model scenarios where we

wish to probe for information based on properties present in a population. These are useful when we are exploring very specific and well defined scenarios and are mostly used when we know the exact conditions when we travel across subtypes. For example, if we wish to model a scenario where we have n mutations and wish to categorize subpopulations based on what mutations are present, we would have to deal with 2^n different subpopulations and each having growth, suppression parameters with respect to other types. The fact that it is unknown which of the 2^n subtypes are feasible and should be probed leaves us with a very large parameter space. This shows a lack of extensibility of differential equation models when we have to deal with parametrized populations.

1.1.2 Automata models

Automata models are one of the oldest mathematical models, as well as one of the most extensively studied ones[109]. These models depend on a spatial "grid" where each node (traditionally called "cells") can represent an individual or a subpopulation.

Traditional examples of such models includes the famous Conway's Game of Life where the game is played on a 2-D grid with the following rules:

1. Each cell has two states: "dead" or "live".
2. Any live cell with two or three live neighbours survives.
3. Any dead cell with three live neighbours becomes a live cell.

4. All other live cells die in the next generation. Similarly, all other dead cells stay dead.

More complex examples exist, such as one introduced to model brain tumor growth[69], which uses a complex 3-D cellular automaton model, accounting for proliferative and non-proliferative cells, an isotropic lattice as well as an adaptive grid lattice.

Unfortunately, spatial models become too complex to represent and efficiently simulate without doing abstractions of the grid out to remove the spatial component. Such models also depend on specific states of each individual cell and cannot account for more complex representations of each cell.

1.1.3 Recommender Models

The models are particularly useful in scenarios where you have a population that responds to external stimuli, like buyers in an online shop such as Amazon, which respond to events such as sales, holidays and even get influenced by celebrities on TV or social media. In such cases, it is beneficial to model the population as a two separate populations, one of which is the recommender, which could be the seller, who would recommend items based on the history of others who it thinks is "similar" to the current buyer, or it could be the social media platform such as Facebook or Twitter, who recommend pages and celebrities to follow based on their perception of what the user enjoys. The second population are the buyers or the users, each of which has an internal state of their own, not directly changeable by the recommender and who take ac-

tion based on what their decision criterion is. It is this "decision criterion" that the recommender wants to learn so that they can leverage it to maximize their profit.

Such recommender systems have seen widespread use, one of the most famous of which is the Netflix Prize competition, won by Bell, et al.[14]. The competition was for developing a better movie recommendation system for Netflix, an online movie platform. The award winning technique used an ensemble of 107 different recommendation systems to create a final recommendation list. Some of the techniques involved in a recommendation systems include Boltzmann machines, collaborative filtering algorithms[114,84] and dynamic k-nearest neighbor models[103].

These are different from traditional population models, in that the population does not directly interact with each other to the same extent that they do in a free floating environment. The interactions take place through the recommender which acts as a man-in-the-middle, and can thus act on the information to manipulate it. Such examples of manipulation have been extensively seen on social media, the most studied of which are the effects on the political spectrum[22,132,15].

1.2 Key limitations

We see that some of the key limitations of current models for evolutionary popuations include:

- Populations with rich internal information are hard to capture.

Internal information blows up concretized stratification of a population and leads to bottlenecks in analysis based on features.

- Information on evolutionary trajectories is lost.
  Keeping a detailed history of large populations is complex unless reducing it to smaller metadata signatures.

- Populations have extreme representations with respect to information content.
  Models either reduce population representations to subpopulation analysis (macro representation) or to single individual representations (micro representation). There needs to be a flexible set of signatures of a population which can be changed by hyperparameters to get multiple levels of detail of a population.

Due to the said limitations of current models, this thesis tries to breach the gaps between topological intuition and data science, and help develop a more rigorous toolset.

1.3 Thesis Contributions

One of the key contributions of this thesis is the geometric and topological tools developed for analyzing evolutionary models. We develop tools which extend current techniques in Topological Data Analysis, Manifold Learning and Graph Theory and combine them together to develop a coherent workflow for using them in concrete settings.

This thesis contributes the following points to the field of machine learning and population modelling:

1. We develop a *Bayesian echo chamber* model for modelling interactive populations with internal information which evolve over time. Modeling an interactive population is made hard due to a multitude of problems:

   (a) Imprecise internal knowledge of individuals.

   (b) Unstructured representation of internal data.

   (c) Low quality data, in the amount of data as well as the time scale.

   We showcase a model using the Bayesian Echo Chamber where we observe political and linguistic data from multiple online communities in Reddit across several years and use it for predicting unobserved inter community interactions. We achieve this goal by using the strength of topological similarity metrics on the persistence diagrams of each community and using deep neural networks to achieve a metric space embedding of each users' internal language using *word2vec*.

2. A new method to perform transfer learning for the cancer survival forecasting problem using Suppes Bayes Causal Networks. Here we use Suppes Causal Networks and quantify similarities between cancer subtypes for patients. Several metrics are proposed and contrasted. We extend an agony based pseudo-metric to the space of causal networks to model similarities between subtypes and use these to boost transfer learning. The

largest contribution of this technique is the generality and extendability to other problems as the underlying techniques are not restricted to cancer survival forcasting.

3. A new method for modelling interactive evolutionary processes based on Digraphons. Creating a refutable hypothesis and giving tools for rejecting a hypothesis is an essential utility in any tool box. This method is a generative model for temporal causal networks which allows us to consider separate populations and gives a similarity metric to get a sense of distance between the evolutionary trajectories of said populations. This allows us to create baseline models for several standard evolutionary models and to apply refutable hypothesis testing by allowing us to use standard probabilistic tools in a temporal setting.

1.4 Thesis Outline

This thesis is organized as follows. In Chapter 2, we give a literature review of the mathematical prerequisites for understanding this thesis. This includes a preliminary introduction to Topological Data Analysis through Persistent Homology, a brief glance over Suppes Bayes causal networks for modeling temporal causal relationships and an overview of Graphons and their extensions to Digraphons and techniques involved in using these for performing predictions. In Chapter 3, we use Persistent Homology and Deep Neural Networks to show how we can model language acquisition and learning in an interactive population in a Bayesian Echo Chamber model using time series data from Reddit. In Chap-

ter 4, we show how to boost learning for cancer survival prediction using transfer learning by using data from different cancers by using a similarity score for causal networks. In Chapter 5, we develop a theory for Digraphons for creating a generative model for causal network learning, adopt the theory for graphons to causal networks and show an efficient algorithm for modeling large evolutionary populations using evoution by duplication. We end the thesis in Chapter 6 with concluding remarks and possible extensions for the work presented here.

2

Mathematical Prerequisites

2.1 Introduction

The recent advances in mathematics and computer science have had tremendous impact on the fields of Topological Data Analysis and Manifold Learning and have culminated in the advent of geometric tools in understanding the workings of machine learning models. This development has also shaped new theories in the underlying structures of data science and has had far reaching effects, even onto Deep Neural Networks, which now are developing a more geometric intuition to understand the black boxes on DNNs.

In this chapter we go over the mathematical prerequisites for understanding the research in data science with extensions to geometric embedding and manifold theory that are presented in this thesis. More specifically, the chapter is structured as follows. We start with a brief introduction to Topological Data Analysis with elements of Persistent Homology and extensions to similarity matrices and data approximation. Then we give a brief overview of the Suppes Bayes causal networks and its learning algorithms. Finally, we go over the theory of graphons and its extensions to digraphons while also introducing similarity metrics on the space of (di)graphons.

2.2 Topological Data Analysis

Topological Data Analysis (TDA) is the study of data as a topological space and to use topological, geometric and statistical tools together to find structure in data. One of the classical ways to understand TDA is in the context of clus-

tering. This is the view that the data points are drawn at random from a population and that analyzing the dataset is going to give an accurate approximation for the underlying distribution. This view is in contrast to computational geometry, which employs similar techniques, but the dataset is assumed to be fixed and we instead focus on performing optimizations for data representation.

One of the fundamentals of TDA is the field of persistent homology[26] and barcodes. Persistent homology employs a multi-scale approach for studying the dataset, where we stratify the dataset at intervals, fig. 2.3, to understand how the shape changes. For most researchers TDA and persistent homology are synonymous, as this is the area of TDA that receives the most attention.

2.2.1 Simplicial complexes and Homology

We start with the basic definition of a **simplex**, which is the first component in building the **simplicial complex**[63]. Intuitively, a simplex is a k-dimensional tetrahedron, fig. 2.1. Simplices are used to create **simplicial complex**, which are the building block of our topological structures. Intuitively, a simplicial complex is a "proper" gluing of simplices such that all faces are properly "aligned".

Definition. 1 (Simplicial Complex)**.**—A set of simplices K is called a *simplicial complex* if

1. for all simplices T in K, all the faces T are in K.

2. The intersection of any two simplices in K is a face of both of them.

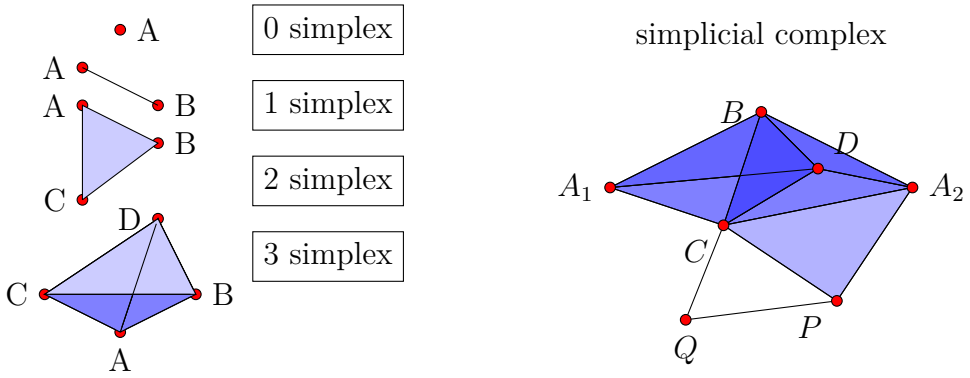

Figure 2.1: Example of 0-3 dimensional simplices and a simplicial complex. A simplex is the equivalent of a multi-dimensional tetrahedron. A simplicial complex is a "valid" attachment of multiple simplices via "gluing". To create the simplicial complex, first take two copies of the 3-simplex and "glue" them along the BCD face. to get a double sided-prism A_1BCDA_2. Next glue a 2-simplex via an edge to the current simplex at A_2C. and then add two 1-simplices to get the final shape.

A subset L of K is a subcomplex of K if it is a simplicial complex by itself. This fact is one of the most important parts of TDA using persistent homology. We are going to be dependent on the fact that over time, we will be "growing" our simplex, and the fact that our old simplices are a subset of our evolved simplex, allows us to view topological properties as a changing "feature" and in fact allows us to map which "features" have changed.

2.2.2 Persistence and barcodes

The first element when we want to introduce persistence is the notion of a *filtered simplicial complex*. This is a simplicial complex built piece by piece, by attaching new simplices, but keeping track of the whole history, giving rise to a *filtration* across the growth.

Definition. 2 (Filtered simplicial complex)**.**—A filtration of a simplicial com-

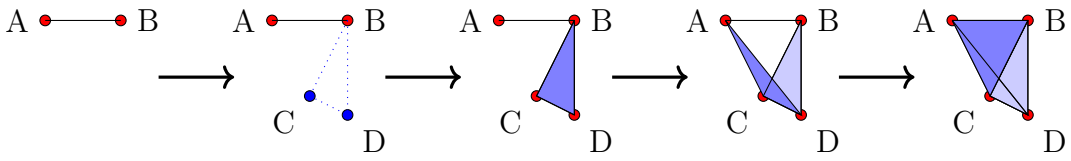

Figure 2.2: A filtered complex with newly added simplices highlighted. At each step the blue highlighted faces are the added simplices. We start with a 1 simplex AB and in the first attachment, glue three 1-simplices, BC, BD, CD. Next we attach a 2 simplex BCD. Then we attach another 2 simplex ACD. In the last step, we attach another 2 simplex ABC, to finally arrive at a hollow tetrahedron with one open face.

plex K is represented as a nested subsequence of complexes $\phi = K^0 \subseteq K^1 \subseteq \cdots \subseteq K^m = K$.

For generality, we let $K^i = K^m$ for all $i \geq m$. We call K a *filtered complex*, e.g. fig. 2.2.

The generalization of a filtered complex is a *persistence complex* which organizes maps across a chain of complexes.

Definition. 3 (Persistence Complex)**.**—A persistence complex C is a family of simplicial complexes $\{C^i\}_{i\geq 0}$, together with homomorphisms $f_i : C^i \to C^{i+1}$.

A filtered complex with the natural inclusion maps is a persistence complex.

Intuitively, a persistence complex is designed to give a growing picture of a complex. Wherein, we look at the local topology of a point cloud data, we try to slowly grow our region of interest and start mapping to progressively larger neighborhoods, fig. 2.3.

The key observation is that as we increase and decrease the radius of the balls (which are used to connect the points to create the simplex), new topological features will appear and old ones will disappear[135]. For example, when $r = 0$,

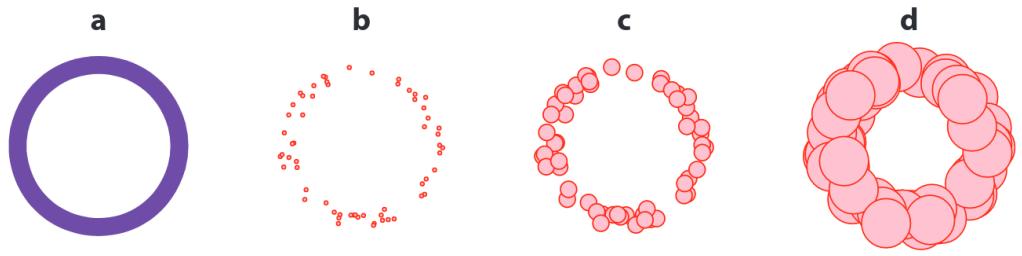

Figure 2.3: Constructing a persistence complex by growing balls at sample points.
We draw a sample dataset from a circle and start growing balls around each point. We see that when we have a sufficiently large radius for the balls, (d), we can topologicaly recover the shape of the original space, image from [126].

we have n connected components, one for each point in the space, that is n disconnected balls. As r increases, some of the components may start "dying", i.e. they will start merging with other components, until at a point where only one component remains.

Similarly, at a certain radius, fig. 2.3 (d), we see that the our complex has an inner "hole". Now if we were to keep increasng the radius of each ball to the radius of the hole, the hole will get filled and it will "die".

Thus each feature has a "birth" and a "death" time associated with them [135]. This process allows us to calculate the *persistence barcodes* of this filtered complex, as a series of intervals which represent features that are born across the timeline of the complex and die when moving to a later time as they are filled with the appearance of a new simplex.

This discussion brings us to the notion of persistence diagrams, which are a way of representing the barcodes such that we can use them as signatures of a space for comparisons, such as getting a distance metric for a notion of similar-

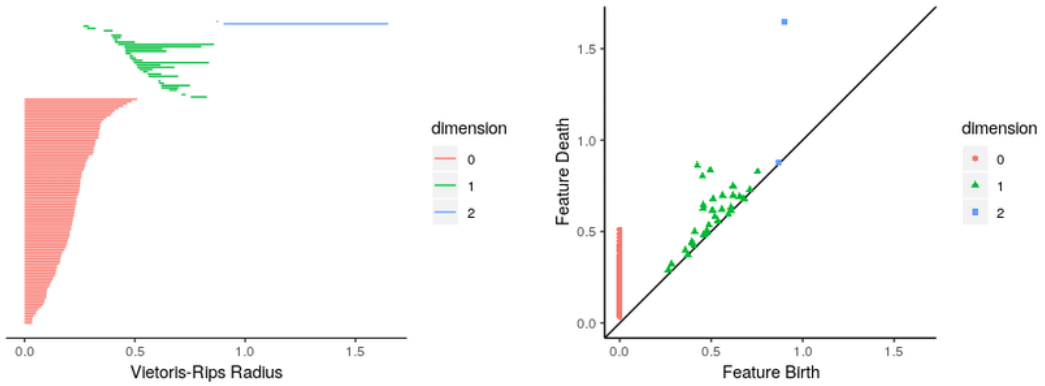

Figure 2.4: Barcode of a persistence complex.
Each "hole" is 'born' at some value of the neighborhood radius and 'dies' when the neighborhoods become large enough to cover it. These 'birth'-'death' time intervals are the *barcode* of that hole or 'feature '. We can then also visualize these barcodes as points in $\mathbb{R}^2_{\geq 0}$ which form the *persistence diagram*. There are many advantages of the persistence diagram, the most important of which is a distance metric, definition 5, which allows us to check for similarities between these diagrams. Image from [124].

ity, fig. 2.4.

Definition. 4 (Persistence Diagram)**.**—A *persistence diagram* is the a 2-D grid plotting points (x_i, y_i) which are the persistence intervals for a persistence complex, e.g. fig. 2.4.

For technical purposes, for a persistence diagram, X, of the points $\{(a_i, b_i)_{i\in\{1\ldots n\}}\}$, we will also include the points on the line $x = y$, and think of $X = \{(a_i, b_i)_{i\in\{1\ldots n\}}\} \cup \{(x, x)\}_{x\in\mathbb{R}_{\geq 0}}$. This inclusion of extra points is needed for allowing comparisons between persistence diagrams with different numbers of actual features, definition 5.

Persistence diagrams are an intuitive representation of the birth-death notion presented above. There are many advantages to the persistence diagrams as they allow for better visualization and a more intuitive understanding of the

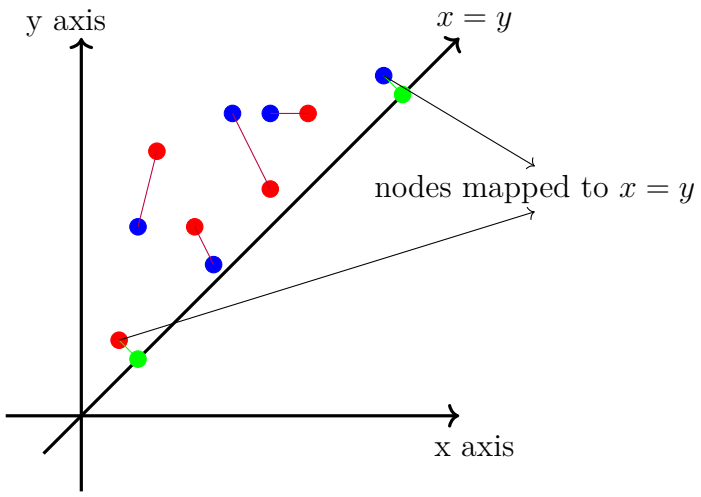

Figure 2.5: Bottleneck distance between persistence diagrams. The points in the first diagram (blue) are mapped to the nearest points in the second diagram (red) or to the line $x = y$ if it reduces total distance or no other points exist.

underlying topological features. The noise in the dataset due to real world constraints warns us that there may be misleading features in the TDA analysis. We see that this has a very simple fix in TDA terms, as most noisy features are those which are very short lived. In the space of persistence diagrams, this translates to points which are close to the $x = y$ line. This observation allows us to rectify our persistence diagrams with a hyperparamter for truncating our diagrams by shifting the line for viewing only features which have a significant lifetime.

2.2.3 Metrics on Persistence Diagrams

The importance of persistence diagrams stems from the fact that they can be used as a signature for a topological space. In the sense that they have a metric defined on them and we can use that as a similarity measure between two

spaces. The standard metric used on the persistence diagrams is the *bottleneck distance*, defined as follows.

Definition. 5 (bottleneck distance)**.**—Given two persistence diagrams X, Y, we define the bottleneck distance between, $d_B(X, Y)$ as

$$d_B(X, Y) = \inf_{\gamma \in \text{Bij}(X,Y)} \sup_{x \in X} \|x - \gamma(x)\|_\infty$$

We are guaranteed that a bijection will always exist as both persistence diagrams have been artificially inflated to contain an infinite number of points.

To generalize this distance, we can also define the bottleneck metric for other p-norms and change the definition to $\inf_{\gamma \in \text{Bij}(X,Y)} \sup_{x \in X} \|x - \gamma(x)\|_p$. For most practical purposes, using the ∞-norm and the Euclidean norm suffice.

The advantage of the bottleneck distance is the relative ease of computation. The problem can be reduced to a **minimum weight bipartite matching** on a graph, as evidenced by creating a bipartite graph with the two sets representing points in X and Y with the edge weights being the distance betweent the two points.

A better generalization is the Wasserstein distance defined as follows.

Definition. 6 (Wasserstein distance)**.**—Given two persistence diagrams X, Y, the $p - q$ Wasserstein distance is defined by

$$W_{p,q}(X, Y) = \inf_{\gamma \in \text{Bij(X, Y)}} \left(\sum_{x \in X} \|x - \gamma(x)\|_q^p \right)^{\frac{1}{p}}$$

The advantage of the Wasserstein distance is that we can recover the bottleneck distance by taking $p = q \to \infty$. Hence theoretical techniques can focus on the analysis of the Wasserstein distance, which in itself is also a very useful function, which has seen wide use as in statistical tools, as an alternative to the asymmetric KL-divergence. For many practical applications it is enough to consider the case where $p = q$. We have the added benefit of knowing that the information content of the Wasserstein distance decreases with the value of p as referenced by the following lemma.

Lemma. 2.2.1. For two persistence diagrams X, Y and $p < p' \in \mathbb{R}_{\geq 1}$, we have that

$$W_{p'}(X, Y) \leq W_p(X, Y)$$

If we wish to use a metric on the persistence diagrams, which is as close to the Gromov-Hausdorff distance as possible, we would want to use a smaller value of p. Unfortunately, a similar result of stability for the Wasserstein distance does not exist. The closest current results introduce error terms which are non-trivially large and not feasible for geometric analysis[113].

Stability of Persistence Diagrams

One of the key points of the distance functions defined apriori is the notion of *stability*. On one hand we have two topological spaces and on the other hand we have two persistence diagrams. The natural distance function defined on the two topological spaces is the **Gromov-Hausdorff** distance function, which uses isometric embeddings to make the two spaces belong to a single parent space.

Definition. 7 (Gromov-Hausdorff distance)**.**—Given two compact metric spaces X, Y, then we define the *Gromov-Hausdorff* distance, $d_{GH}(X, Y)$ as

$$d_{GH}(x, Y) = \inf_{e_x, e_y} d_H(e_x(X), e_y(Y))$$

where e_x and e_y are two isometric embeddings of X and Y into a shared metric space M and d_H represents the standard Hausdorff distance.

This is the standard distance for comparing two abstract spaces, but as is clear, it is made enormously difficult by the presence of all possible isometric embeddings. We instead use the fact that we have a persistence signature present and we have that the bottleneck distance respects the Gromov-Hausdorff distance.

We first define the notion of a monotone function f, over a simplicial complex, $\mathcal{K}$, such that f takes values for each simplex in $\mathcal{K}$. And here f is monotone over the inclusion map over the simplices, i.e. $\sigma \subset \sigma' \implies f(\sigma) \leq f(\sigma')$. This notion naturally extends to a function over a persistence complex. We are specifically interested in the distance function d_k, which maps each simplex to its diameter in the metric space defined by d_k.

The importance of this notion comes as a lower bound for the Gromov-Hausdorff distance as evidenced by the following theorem.

Theorem. 2.2.1 (Stability of the bottleneck distance[32])**.** For two metric spaces

(X, d_x) and (Y, d_y), and for each $n \in \mathbb{N}$, we have that

$$d_B(D_n(X, d_x), D_n(Y, d_y)) \leq d_{GH}((X, d_x), (Y, d_y))$$

where D_n represents the n'th dimentional persistence diagram for a space.

This lower bound is tight in the sense that many real world examples do achieve this bound. A simplified example is the case where X is a set of two points with distance 2 and Y is a set of two points with distance $2 + 2\epsilon$. They can both be isometrically embedded into $\mathbb{R}$ with the representations, $X = \{0, 2\}, Y = \{-\epsilon, 2 + \epsilon\}$, to get $d_{GH}(X, Y) = \epsilon$. We have that the 0-dimensional barcodes of X for its Vietoris Rips complex are $(0, +\infty), (0, 1)$, as the two balls touch when they have radius 1, and for Y are $(0, +\infty), (0, 1 + \epsilon)$. These have a bottleneck distance of ϵ, thereby achieving our desired lower bound.

2.2.4 Real World Approximations

In the real world, a lot of the times, the data size is very cumbersome to work with for all calculations. Especially with topological analysis where we have to deal with pairwise distances, we always have a non-trivial lower bound of $O(n^2)$. It is possible to reduce it to a more practical scenario with k-D trees and fast local approximate nearest neighbor (FLANN) algorithms, it is not desirable to introduce such complexity for a preliminary task.

The practical answer to this scenario is the introduction of the *witness complex*, which can be thought of as an approximation to the Delaunay triangula-

tion, while removing the exponential time complexity (curse of dimensionality) associated with the triangulation.

In simple terms, the witness complex is a simplicial complex built from a dataset D and a subset $L \subseteq D$, termed the *landmarks*. This means that we will be restricting our attention to the distance matrix $M = L \times D$, with any suitable metric being used. In later sections we will introduce more involved metrics on the space of digraphs and digraphons, which will allow us to adapt this theory to the space of digraphons.

In addition to the standard witness complex, $W(M)$, also called the parameter-free witness complex, we also have parametrized versions $W(M; r, v)$, where $v = 0, 1, 2$, and r is the *feature-size* hyper-parameter, which dictates the maximum sized balls to have around our data points. As any such family can be used to create a filtration, we can hence subsume them into persistent filtrations to get our desired topological signatures.

This approach while not having concrete footholds in theoretical results simplifies a lot of problems relating to computation costs and has had significant real world success. Some of the key points in favor of the witness complex include

- The computational matrix is orders of degrees smaller, which enables us to get signatures with larger 'feature-length'. This simplification allows us to gain better insight and do a better analysis into the topological features.

- Other than the number of and choice of landmarks, no other parameters

need to be set, unless upper bounding the feature length.

- A randomized sampling of landmarks allows us to filter noise in the dataset, especially related to low length barcodes. Multiple landmark sets sampled using a *maxmin* criterion, where we choose successive landmarks using a *max of min* distance optimization relative to current landmarks, which enables us to get a very clear picture of the topology.

The witness complex, $W(M_{l\times n})$, is constructed through an intermediary *strict witness complex*, $W_\infty(M)$.

1-skeleton . The edge $[a, b]$ belongs to $W_\infty(M)$, iff there exists a data point w, such that (a, w) and (b, w) are the two smallest entries in the w'th column of M, in some order. Here w is called the *witness* for the 1-simplex $[ab]$.

p-skeleton . Inductively, the p-simplex, $[a_0a_1 \dots a_p]$ belongs to $W_\infty(M)$, iff there exists a *witness* datapoint w such that, the entries, $(a_0, w), (a_1, w), \dots, (a_p, w)$ are the $p + 1$ smallest entries of the w'th column.

This computation is a bit cumbersome due to the inductive hypothesis, and we define the simplified $W(M)$ as the smallest complex $W_\infty(M) \subseteq W(M)$, such that

- $W(M)$ has the same 1-skeleton as $W_\infty(M)$.
- $[a_0a_1 \dots a_p]$ belongs to $W(M)$, if all of its edges belong to $W(M)$.

This relaxation allows us to get a better computational complexity and obtain a more robust pipeline for analysis.

2.3 Suppes Bayes Causal Networks

For many years, the dominant paradigm for understanding cancer has been one of continual stochastic mutation and selection. In this process some mutations — called *driver* mutations — are culpable in continued proliferation and the acquisition of phenotypic *hallmarks*[61] while others — so called *passengers* — are mere incidental alterations that are preserved via clonal expansion. In cancer genomics two problems go in to progression modeling. The first is the identification of driver mutations from the whole of genetic information. This has been addressed by efforts such as The Cancer Genome Atlas (TCGA) and the Catalogue Of Somatic Mutations In Cancer (COSMIC). The second task, and the one with which we concern ourselves, is the task of reconstructing the history in which driver mutations were acquired.

For the purposes of this thesis, we focus on a particular progression model where the partial order is due to a particular *selective advantage relation.* This relationship is called *prima facie* causality because it was originally developed as such in the philosophical literature by Patrick Suppes[116]. For this reason we keep the term causality for our progression models but remark that there is a close correspondence between this relation and the evolutionary accumulation of mutations that is of interest to us.

Definition. 8 (Prima-Facie Causality)**.**—An event c is a *prima facie cause* of another event e if the following two conditions hold:

- (Temporal Priority) $t_c < t_e$, where t_* is the time of observation of an

event.

- (Probability Raising) $\Pr[e|c] > \Pr[e|\neg c]$.

Two remarks are in order. The first is that the notion of prima facie causality here is different from the type of causality more commonly considered in the computer science literature[102] using counter factual possible worlds or do-calculus or structural equations . However, it is not entirely distinct and the probability raising condition maps to the same counterfactual intuitions that underlie clinical trials and the "do calculus." An additional and critical similarity is that we are able to construct causal graphs, a topic to which we return shortly.

The second remark is that this approach to causality mirrors the biological processes of the accumulation of driver genes[106]. Suppose at some time t_1 a patient undergoes a mutation in *KRAS* causing the tumor to grow rapidly. As this process continues the cells in the mass will begin to experience *hypoxia*, which necessitates an alteration of behavior to angiogenesis or to metastasis. In this condition a mutation in, for example, *VEGF*, at time $t_2 > t_1$ would lead to necessary metabolic changes (e.g. angiogenesis) to ensure continued growth. Observe also that without the initial mutation in *KRAS* the *VEGF* mutation would not have provided any advantage so it is presumably less likely to occur.*

*We acknowledge that rigorously stating this would require specifying base-rate mutations in *VEGF* the tumor. The thrust of the argument is that there is selective pressure on the tumor as a whole to survive hypoxia and as such the clones that survive and are biopsied are those which have such an advantageous mutation.

Definition. 9 (Suppes Bayes Causal Network (alpha))**.**—A Suppes Bayes Causal Network (alpha[†]) (SBCN) is a DAG where for every edge $v_i \to v_j$, Suppes conditions for *prima facie causation* hold, that is:

$$\Pr[v_i] > \Pr[v_j] \quad \text{and} \quad \Pr[v_j|v_i] > \Pr[v_j|\neg v_i]$$

For our purposes the input to learning an SBCN is cross-sectional patient data, that is a binary matrix $D \in \mathbb{Z}_2^{n\times m}$ where n is the number of patients and m is the number of genes that are being considered. The learning of an SBCN structure from data can be implemented efficiently using open source software[37]. For the purposes of this chapter, we consider only point mutations do not account for the variant type (e.g. missense, nonsense).

While we have chosen to work with this particular notion of causality, our methods in this chapter are agnostic as to the semantics of the underlying progression model. We chose Suppes causation because it has previously been applied in biology and is computationally tractable for large data sets[105].

2.3.1 Real World Optimizations

In real world setting, where we have to deal with noise and incomplete datasets, Suppes conditions are not enough to get a robust and noise free model. The model will have a lot of false positives as there is not enough evidence (number of events) to justify is difference in probabilities. Hence, even though the

[†]This is our preliminary definition which we will further enhance with Bayesian optimization techniqes.

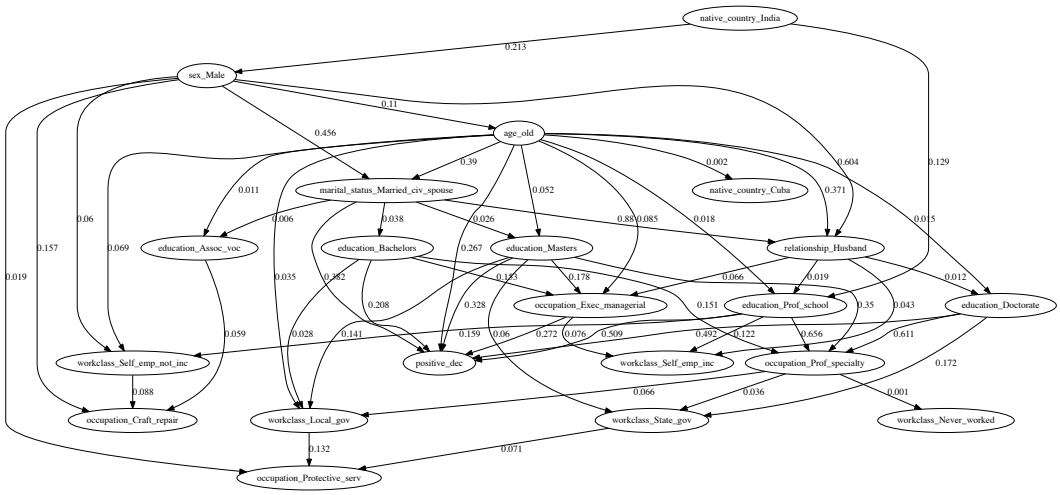

Figure 2.6: An example of an SBCN extracted from the Adult dataset, UCI. The graph edges denote causal relations satisfying the *prima facie* causality, definition 8, while the edge weights denote the increase in the probabilistic causation. Image taken from [18].

network may satisfy Suppes constraints, there will be "spurious" edges. To account for this, in general a structural conditions are used to guarantee simple networks. An alternative to that is the use of the Bayesian Information Criterion (BIC) as a regularizer for the likelihood score, which prioritized simpler networks to be used.

For a given graph G and dataset D, with s samples, select a subset $E' \subseteq E$, which maximizes

$$score_{BIC}(D, G) = \text{LogLikelihood}(D|G) - \frac{\log s}{2} dim(G)$$

where $dim(G)$ is the number of parameters of the dataset.

Thus we see that the regularizer term $\frac{\log s}{2} dim(G)$ prioritizes sparser graphs,

in terms of number of edges. Note that the log likelihood implicitly depends on the number of points in the graph, so we do not need to worry about optimizing number of vertices separately.

Given an SBCN we can now look at the confidence score as an output of the relationships between nodes. Using the conditional probabilities for every pair of nodes in the graph, connected by an edge, we expect to have several observations of any possible combination of the variables. For this reason, we can simplify the estimate for the node probabilities by counting the observations in the data. And we then use this to define the confidence of the edge.

In particular, for each edge $(v, u) \in E^*$, involving a relationship between two nodes $u, v \in V$, we define the confidence score, $W(u, v) = P(v|u)P(v|\neg u)$, which intuitively, tries to estimate how many the the observations contribute to the event where the cause u is followed by its effect v, that is $\Pr(v|u)$, and the ones where this is not observed due to the lack of the prior cause, $\Pr(v|\neg u)$, because of imperfect causal regularities. Note that, by the constraints discussed above, we have that $\Pr(v|u) > P(v|\neg u)$ and, thus, each weight is positive and no larger than 1, i.e., $W : E^* \to [0, 1]$. Combining all the previous observations, we define the generalized SBCN.

Definition. 10 (Suppes Bayes Causal Network)**.**—Given an input dataset D of m Bernoulli random variables and s samples, and given a partial order r of the variables, the Suppes Bayes Causal Network $S = (V, E^*, W)$ is a weighted DAG which satisfies the following constraints

1. **[SBCNa]** The graph S is an SBCNa, that is, it satisfies all Suppes con-

straints.

2. [**Simplification**] If E' is the set of all edges which satisfy Suppes constraints (which is a superset of E^*), then E^* should be the one which maximizes the BIC score

$$E^* = \underset{E \subseteq E', G=(V,E)}{\arg\max} LL(D|G) - \frac{logs}{2} dim(G)$$

3. [**Score**] For each edge $u, v \in E^*$, define the score

$$W(u, v) = \Pr(u|v) - \Pr(u|\neg v)$$

We present the hill climbing algorithm for learning the SBCN in algorithm 1, which is an iterative approach for doing optimization along the manifold of all valid solutions.

The $StoppingCriterion()$ mentioned in the algorithm is the amalgamation of the two cases

1. We have exceeded the number of maximum iterations, which is controlled by a hyper parameter, typically set to a valid large enough number to ensure a wide search space coverage.

2. None of the neighbors of the current fitted G_f have a better BIC score than our current fit.

Algorithm 1 SBCN learning algorithm

function SBCN(Dataset D, Int m, Int s, Int r)
    # m - number of Bernoulli variables
    # s - number of samples
    # r - partial ordering of events
    for all pairs of variables u, v
        # **SBCNa**
        if $r(u) < r(v)$ and $\Pr(v|u) > \Pr(v|\neg u)$ then
            Add (u, v) to the SBCN
        end if
    end for
    # Maximize Log Likelihood by hill climbing
    Start with the empty fitting $G_f(V, E^*, W) = \phi$
    while !$StoppingCriterion()$
        Let G_* be set of neighbours of G_f, constructed by adding/removing a single edge from G_f.
        Prune G_* to only include graphs which satisfy *SBCNa*.
        Consider random neighbor G' in G_*
        if $score_{BIC}(D, G') > score_{BIC}(D, G_f)$ then
            $G_f = G'$
            for each edge $u, v \in E_f$

$$W_f(u, v) = \Pr(u|v) - \Pr(u|\neg v)$$

            end for
        end if
    end while
    return G_f
end function

2.4 Graphons and Digraphons

Let $\mathcal{W}$ denote the space of all symmetric, bounded, measurable functions $\mathcal{W} : [0,1]^2 \to \mathbb{R}$. This is defined as the set of all *kernels* , reminiscent of the kernels used in Support Vector Machines. If we restrict our attention to the set of functions $W \in \mathcal{W}_0$ such that $0 \leq W \leq 1$, we arrive at the space of *graphons*. If we drop the condition of symmetry, we arrive at the space of digraphons.

For our purposes, we do not distinguish between the functions which are equal almost everywhere, as with most analytical scenarios, it is not possible to distinguish between such functions. We will soon see that this is actually not the only equivalence we want to put on the functions, as we will need to also equate a larger class of functions to each other for the sake of **exchangeability**, which is important for statistical modeling.

The notion of graphons is an important one where we want to look at limits of graph sequences. Such graph sequences arise in a variety of different natural scenarios, such as social networks, recommendation systems for advertisements, shopping, etc., and also in biological scenarios, such as genetic mutations, evolutionary models, population dynamics. As a general rule of thumb, any scenario where we have a dynamic population with potential for growth in event space is a candidate for getting sequence of graphs.

Natural questions arise in how to analyze such a sequence. Does the growth follow a pattern? Are there noticeable features of this graph that are preserved across its growth? Does the graph sequence converge to any discernible end ob-

ject? The theory of graphons (and digraphons) tries to answer many such questions in a rigorous form.

For us to have a notion of convergence and similarity, we need to start with a notion of a distance between digraphs. There are many distances defined on the space of digraphs, the distances introduced by the L_p norms, nuclear norms, etc.; we will restrict our attention to the more interesting *cut distance*.

Definition. 11 (Cut Distance)**.**—For two directed graphs G, G', on the same set of vertices V, the cut distance is defined as

$$d_{\square}(G, G') = \max_{S,T \subseteq V} \frac{e_G(S,T) - e_{G'}(S,T)}{|V|^2}$$

where $e_G(S,T)$ denotes the number of edges between S and T in the graph G.

If we let $d_1(G, G')$ be the L_1 distance on the adjacency matrices of G and G' we get the inequality $d_{\square}(G, G') \leq d_1(G, G')$. Hence we see that the two distances give differing information. As an example, for two Erdos-Renyi graphs with $p = 1/2$, we get that $E[d_1(G, G')] = 1/2$ while $E[d_{\square}(G, G')] = \theta(1/\sqrt{n})$.

For unlabeled graphs on the same set of nodes, the intuitive extension to the cut distance is to define it via equivalence on node relabelings, which turns out to be the correct one. Let G, G' be two graphs on same number of nodes, then the cut distance is overloaded as

$$d_{\square}(G, G') = \min_{\phi \in \mathrm{Hom}(G,G')} d_{\square}(\phi(G), G').$$

Here we minimize over all homomorphims of G into G'. For generalized mea-

surable digraphons, we first define the *cut norm*

$$\|W\|_{\square} = \sup_{S,T \subseteq [0,1]} \int_{S \times T} W$$

Which can then be extended to the cut distance as $d_{\square} W, W' = \|W - W'\|_{\square}$. Similar to the finite case, we achieve the inequalities between norms

$$\|W\|_{\square} \leq \|W\|_1 \leq \|W\|_p \leq \|W\|_{\infty} \leq 1$$

Again, similar to the finite case of the cut distance, where we have "relabelings" via measure preserving homomorphisms of $\phi : [0,1] \to [0,1]$.

$$d_{\square}(W, W') = \inf_{\phi \in \mathrm{Hom}([0,1])} d_{\square}(\phi(W), W')$$

It is important to note that finite graphs can be represented as a specific case of a graphon by using step functions. For example, a digraph on $[n]$ with the adjacency matrix $A_{i,j}$ can be canonically viewed as a digraphon W, with $W(\frac{i}{n}, \frac{j}{n}) = A[i,j]$. This allows us to treat even finite graphs as a graphons and simplify our analysis, where we no longer have to distinguish between sequences of graphs vs sequence of graphons.

We see that any measure preserving transformation of a digraphon has zero cut-distance to the original. An important theorem states that the only digraphons which have cut distance zero are the ones under measure preserving transformations of the original or of one which is equal almost everywhere, termed as weakly isomorphic pairs.

Theorem. 2.4.1 (Weak isomorphism theorem[89])**.** Let W, W' be two digraphons, then $d_\square(W, W') = 0$ if and only if there exists a digraphon Z, such that $W = Z$ almost everywhere and W' is a measure preserving homomorphism of Z[‡].

This result is one of the most important ones for the analysis of digraphons, as this gives confidence to our sampling algorithms. Indeed, the proof of this theorem itself relies on the canonical digraphon and sampling state introduced. Then we generalize the distance for two digraphons W, W' by round robin chasing across the commutative diagram,

$$d_\square(W, \text{Sample}(W)) \leftrightarrow d_\square(W, W') \leftrightarrow d_\square(W', \text{Sample}(W'))$$

Lemma. 2.4.1 (Convergence in norm)**.** Let $W_n, n = 1, 2, \ldots,$ be a sequence of digraphons such that $\|W_n\|_\square \to 0$. Then for all dikernels Z, we have that $\|W_n Z\|_\square \to 0$.

2.4.1 Sampling

One of the important parts of digraphons are the guarantees that a finite sampling is going to converge to the underlying digraphon. The sampling works as follows.

Given a digraphon W and an ordered set $S = (x_1, \ldots, x_n), x_i \in [0, 1]$, we create a weighted digraph $H(S, W)$ on the node set $[n]$ with the edge weights $H(i, j) = W(x_i, x_j)$. Now from such an H we can create a random simple un-

[‡]In particular this also covers the case where $W = Z$ everywhere.

weighted digraph by trying to sample G and adding an edge $G(i,j)$ with probability $H(i,j)$.

For example, if W is the uniform function with $W(i,j) = p, 0 \leq p \leq 1$, we would get the standard Erdos-Renyi graphs with probability p. If $W = W_G$, the canonical digraphon for a digraph G, then if we sample k points from W_G, it is the "almost" the same as calculating a random subgraph of G. It is not the same as we might have sampled x_i, x_j from the same step in W_G. To have an exact subgraph sampling, we need to condition on the fact that x_i, x_j need to be from different steps in W_G. In particular, we are removing sequences$(x_1, \ldots, x_n)$, with repetitions, which has $\binom{k}{2}$ such sequences, and hence a measure of $\frac{\binom{k}{2}}{n}$. This gives us that the average distance between a randomly chosen subgraph, $R(k,G)$ and a randomly sampled digraphon$R(k, W_G)$ is

$$d(R(k,G), R(k,W_G)) \leq \binom{k}{2}\frac{1}{n}$$

Now that we know how to sample, we can start looking at how sampling helps in parameter estimation. For such a scenario, we want to start with a notion of "good" parameters, which we can hope to be estimable. As it will turn out, most of the real world scenarios are going to be good and can be estimated using sampling. We wish to achieve some notions similar to the limit theorems for classical statistics which will give us confidence on doing real world analytics using EM or MAP algorithms for estimations.

A *reasonably smooth* graph parameter is defined as a function $f(G([S]))$, which is to say of a sampling of a digraphon, which satisfies that $|f(G) - f(G')| <$

1, for two graphs G, G' on the same set of nodes, whose edges differ only for a single vertex. We then achieve a sample concentration theorem

Theorem. 2.4.2 (Sample concentration theorem for digraphons[87])**.**

Let f be a reasonably smooth graph parameter, and let W be a digraphon, $k > 1 \in \mathbb{N}$. Let $f_0 = \mathbf{E}[f(R(k, W_G))]$, then for all $t > 0$,

$$\Pr\left[f(R(k, W_G)) > f_0 + \sqrt{2tk}\right] < e^{-t}$$

This theorem gives credibility to the fact that our intuitive sampling algorithms are going to be working correctly on standard simulations. In fact, we can achieve an even better result which states.

Theorem. 2.4.3 (Cut distance confidence[87])**.** Let $k > 1 \in \mathbb{N}$ and let G be a digraph on k nodes. Then with probability at least $1 - \exp\left(-\frac{k}{2\log k}\right)$, we have that

$$d_{\square}(G, R(k, W_G)) \leq \frac{20}{\sqrt{\log k}}$$

Now with such a result, we can employ our standard statistical tools to carry out prior/posterior estimations and simulations using generative models. Indeed, we develop a maximum a posteriori (MAP) estimation algorithm for the Dirichlet priors on digraphon generative models and due to this result, we can be somewhat confident in the fact that we are achieving a good result, subject to proper maximum optimizations in the log likelihood estimation.

3

Language acquisition and learning through interaction

3.1 The Intuit language

*The mystery of language evolution and its (co-)evolution with learning continues to arouse intense debates. There are only a handful of conceptual frameworks for human languages that have found common acceptance: (i) Human language is a biological artefact, as opposed to a cultural artifact[81]. (ii) Human language builds on a hierarchical structure, whose depth is not upper-bounded[30]. (iii) Human language acquisition occurs over a surprisingly short period aided primarily by positive examples[24,83]. However, there are many other corollaries that seem to have found neither acceptance in theory nor utilization in tool-boxes that aim to automate natural language processing.

There are other similar questions in the biology of evolution: e.g., codon evolution and evolution of intercellular signaling, which are important in the emergence of cellularization and multi-cellular organisms, respectively[110]. The theoretical framework for them can be built on information-asymmetric games and their conventional Nash equilibria, and can be tested experimentally in artificial cells with unnatural bases (and the resulting codons), and in modified cells with chimeric receptors, for instance. There are few natural experiments that shed light on these processes, e.g., mitochondria and tumor cells, and they have also played an important role in our understanding of evolution of these systems[2,77].

These systems, like human language, can also be thought of encoding some form of inter-agent coordination (not necessarily faithfully)[121]. They also share few other traits: e.g., (i) Universality, (ii) Stability and (iii) Near Optimality

*Co-authored with R. Rinberg, S. Chakraborty, B. Mishra, arXiv:2102.12382

(with respect to suitably selected utility); we will call them USNO-theories. A rigorous theory for human languages may seek to build on similar traits: (i) A universal grammar (with some flexibility for parametrization)[34], (ii) Stability (with faithful acquisition using meager amount of positive stimuli)[44,97] and (iii) Near Optimality (as a solution to minimal design specifications)[47,27]. However, hypotheses related to physiology of a language organ or the genetics of linguistic phenotypes are not readily testable experimentally as human language is unique to humans thus imposing stringent ethical barriers against their experimental manipulation. Some analysis of bird-songs have been useful, but not very conclusive (for obvious reasons). In silico models that work reasonably in the context of machine learning and artificial intelligence have focused on large text corpora and semi-supervised learning (with massive number of counter-examples) that do not capture the human context and remain orthogonal to the biology of languages[33].

Interesting natural experiments that are thought to have lent support to USNO-theories are in the creolization process, where a group of individuals from Old World are assembled with no common human language to use for coordination, but who give rise to a second generation of New World speech community that invent a human language (Creole) with a new parametrization of the universal grammar, but also enjoying the stability and near-optimality, with respect to any communication criterion, that is common to already-existing human languages[111]. However, while Creole languages can be studied, their evolution remains poorly understood as there exists no data recording their historical

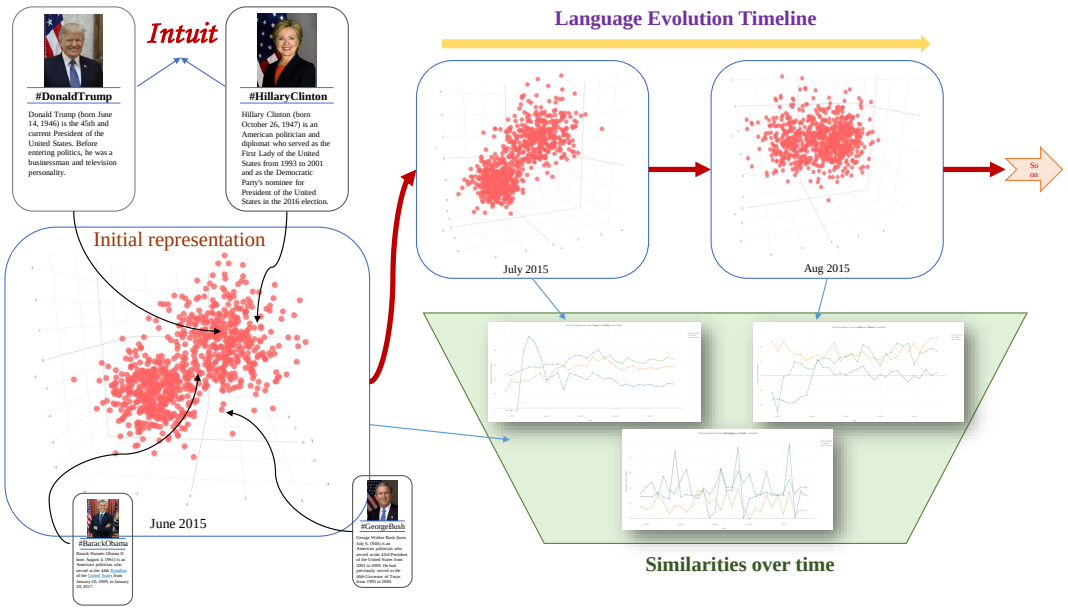

Figure 3.1: Overview of an intuit. A linguistic object `intuit` consists of an image, a hashtag and a short description of the intuit. A language starts with a small number of such intuits in a core germinal population and accrues additional users who add additional intuits, and use them to communicate. External to this dynamics, we can examine the time-stamped representations of the vocabulary of intuits and observe the evolution of the representation through time. This analysis will help us in understanding the social vs. inherent evolution of the representations and of language, based on the changes of the similarities of the intuits over time. Analyzing the data on a per user basis will give us hitherto unknown knowledge of the socio-cultural effects of interactions and community effects on dialects of the language. But, motivated to study the language as a whole, as opposed to just a pair of intuits and their similarities, we were led to novel mathematics to analyze topological differences in representations over time.

dynamics[66]. As Crick's Frozen Accident hypothesis and the Cambrian explosion have been used to explain codon evolution or multi-cellularity, there has been human language evolution's Pop hypothesis that suggests creolization would happen suddenly and freeze quickly, not thawing ever again[76]. The alternative experimentally-supported hypothesis suggesting emergence of a human language as a stable separating Nash equilibria of an information asymmetric game would be more explanatory and hence appealing[65,29].

Motivated thus, we have proposed using crowd-sourcing to create a super sized speech-community with a massively scalable socio-technological version of creolization. The elements of these systems would be **intuits** (with more details in later sections), and eventually a grammar that linearizes (or even planarizes) Intuits in a stable manner. We call this idea "Creolization of the Web" and here, we study various algorithmic issues related to machine learning, natural language processing and evolutionary processes to study the feasibility of such a creolization experiment(s). In particular, we focus on (i) definition of Intuits, the building blocks of the creolization combining images, hashtags, and short tweet-like (140 characters) description, (ii) their dynamic geometric representation and (iii) evolution of the representation via a Bayesian echo chamber. We illustrate the process with Reddit data involving political subreddits to identify evolutionary patterns that emerge in a dynamic population interaction model (fig. 3.1,[60]).

We realize that the ultimate system that combines elements of wiki, Twitter, emoticons, and Facebook could provide enormous utility in web-search, social

networking, and shared economy, possibly displacing English as the defacto intermediate language of the web. Creation of a suitable infrastructure for Intuits remains a secondary but critical goal.

3.2 Language evolution by interaction

We aim to build a database of a pictorial language called, **intuits**, which will help in the process of learning language evolution. The building blocks of this language, called an *intuit*, is a token for any word in the vocabulary, where the token contains richer information than just the word, by storing (1) a title (a hashtag unique identifier, (2) a brief (140 character) description of the title and (3) an image of the title. The presence of this database to track the change of meanings of the intuits over time will give important insights to the theory of language evolution.

In this chapter, we give a baseline minimal model, based on the *Bayesian Echo Chamber*[60], which is applicable to any evolutionary method and also has the flexibility to be individualized to any language using concrete grammars and objective semantics specific to that language. To experimentally verify the plausibility of such a model, we analyze real world data from Reddit , which is an online community of users – sufficiently active and engaged to model communication interactions in a population. Reddit is structured as a collection of "subreddits", which are communities dedicated to a particular topic, such as gaming , sports , technology , etc. Each user of Reddit is generally subscribed to a few of the subreddits, focusing on the content that the user generally browses and is

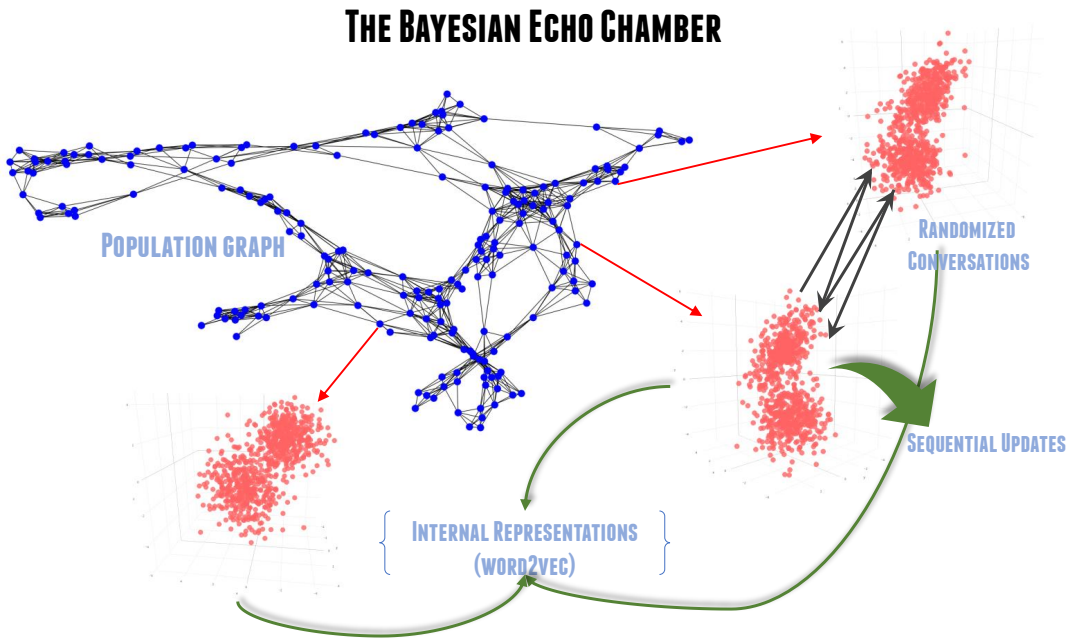

Figure 3.2: Overview of the Bayesian Echo Chamber (BEC). In the model of the Bayesian Echo Chamber, the population is represented as a graph of individuals, called (language) learners, where the edges denote interactions (conversations) between learners. Each learner has their own internal representation of the language, which they use in their conversations with their neighbors. The conversations happen based on a particular topic. And the words in the conversation are chosen based on the similarities of the words with the topic in the internal representation of the learner.

exposed to. The Reddit community has been frequently divided on many topics, most recent of which has been on the political spectrum. This discordance provides a very rich environment to measure the effects of social stratification of language due to dissenting views between communities. We started with a synthetic model of intuits for a large population interacting panmictically, i.e. by random interaction (or as determined by an expander population graph), as it provides a baseline for an idealized theoretical model and a null model for hypotheses testing.

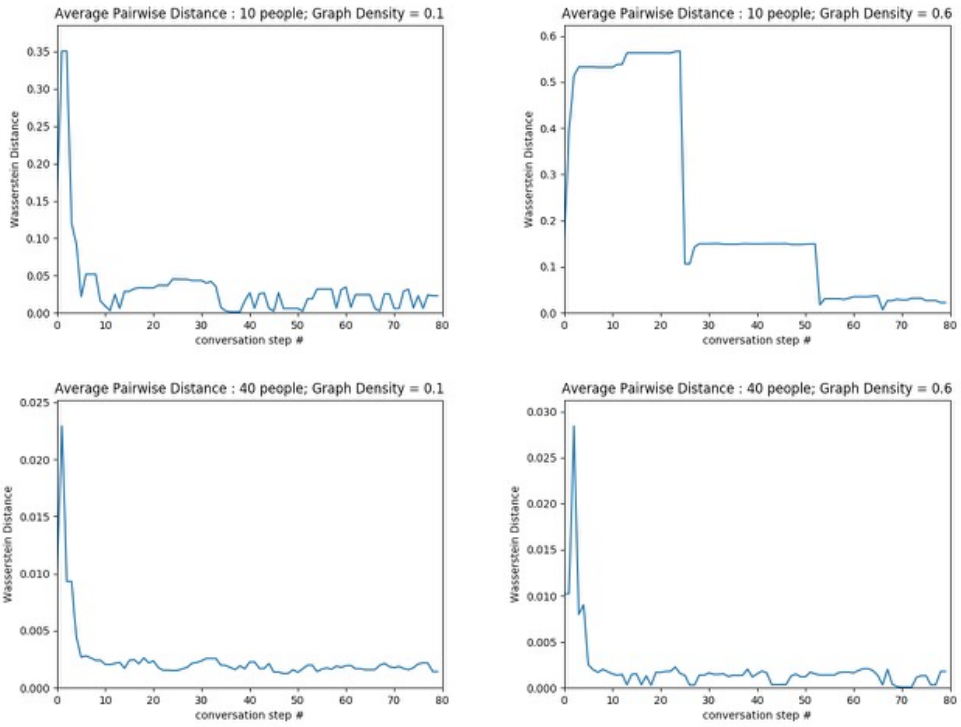

Figure 3.3: Synthetic simulations using word2vec models for small BECs with various parameters of connectivity and size. The simulations show a fast convergence in the representation of individuals and a small drift over time after convergence. These results agree with the accepted theories of language evolution which predict fast stabilization and small drifts in language representations [57,51].

The change in language is measured using computational tools (originally developed for Natural Language Processing, NLP), specifically `word2vec`, to get a feature rich, high dimensional embedding of the elements of a language associated with individual speakers. These embeddings can be thought of as the representation of the language for the individual and the difference in the representations gives us a measure of the dissimilarity between the interpretations of the language in the population. Each representation being a corpus of high dimensional points ("point clouds"), there is no standard notion of a distance between two such comparable representations. We propose to apply a topological metric using *persistent homology*[43,25], which is an emerging field of computational mathematics, quantifying a sense of difference between two representations. The advantages of using the topological metric is the rich information content, which provides insights into the local features of a space as well as measuring the global differences between two representations[6,112].

3.3 Analyzing Reddit communities

3.3.1 Confirmation of Echo Chambers in Reddit

The existence of echo chambers in any society can be manifested in many forms, such as the presence of dialects across the physical distribution of a population or the prevalence of accepted norms and ideologies in a community. The frequent divide in the political spectrum within a population, popularly described as the "left and right-wing extremisms", is an interesting part of language that can be harnessed to understand political ideologies in subreddits.

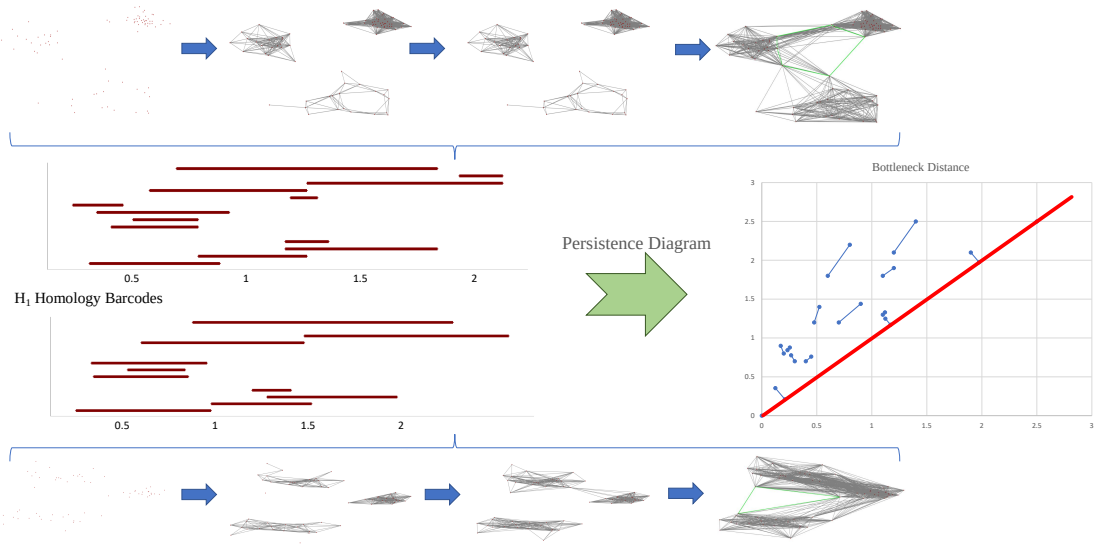

Figure 3.4: To define the distance between representations we start with a topological metric, as it gives information about the representation as well as the differences between two representations. We examine features in the word2vec embeddings of the vocabulary of a learner and calculate the distances based on the geometric embeddings. Two words will be close to each other in the word2vec space iff they are semantically nearly synonymous in the vocabulary of the learner. Now we can calculate the persistent homologies of the embedding and obtain the persistence diagrams of the space. This computation gives us the **bottleneck distance** between the two diagrams, and equips us with a sense of how dis-similar two embeddings are.

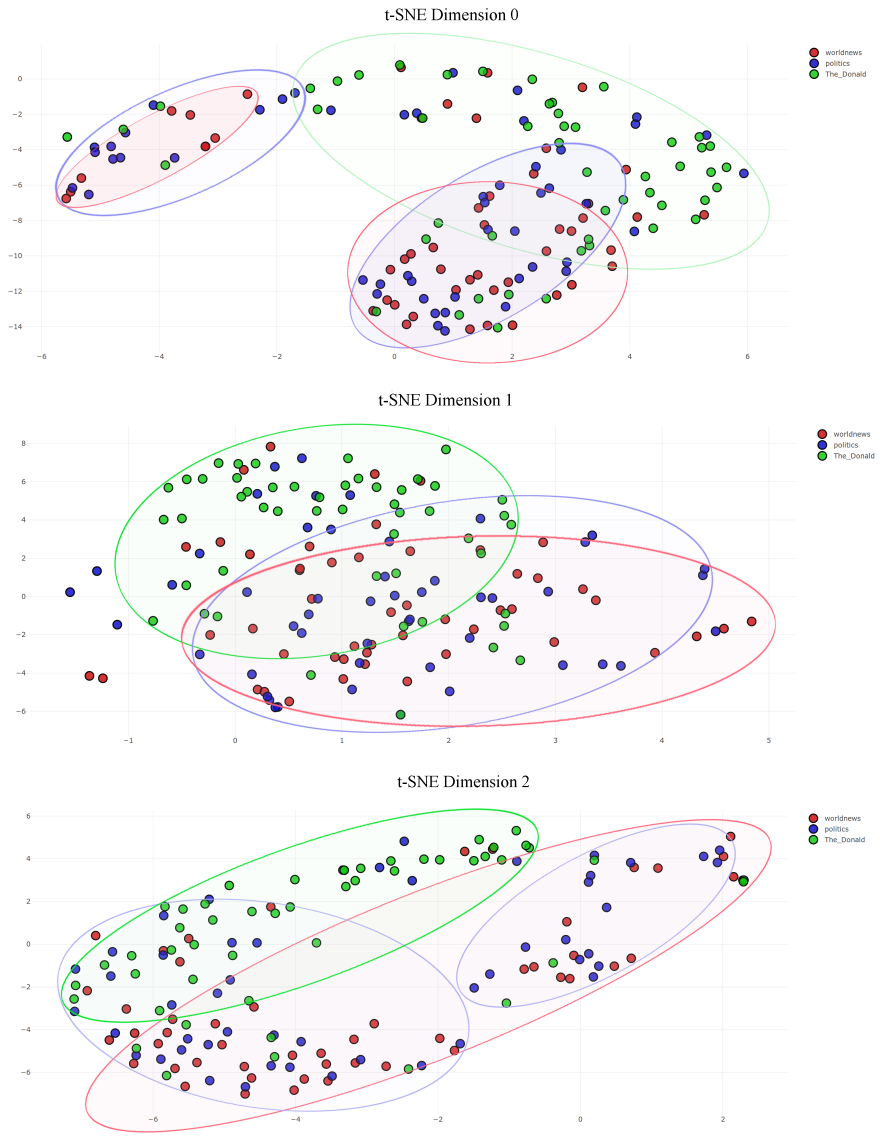

Figure 3.5: The embeddings of each user were generated using the state of the art `word2vec` models and their Reddit data from the June 2015 till November 2017; using which, we calculated the distances between each pair of users using the persistent homology metric. To visualize and quantify the clusters formed using this metric, we performed t-SNE in 2-D plane, as t-SNE gives higher probabilities to cluster pairs which have small distance while not clustering larger distance pairs.
The resulting clusters, visualized via a simple min-max similar to k-means, show a stark similarity between the users of /r/politics and /r/worldnews , while those of /r/The_Donald are clustered separately. This behavior is mimicked in all dimensions, though less clearly in dimension 2, suggesting that there is very little communication happening between the users of /r/The_Donald and others.

To examine this hypothesis explaining a spectrum in the communities, we proceeded to analyze the three most popular political subreddits which are widely believed to cater to different groups, namely /r/politics , /r/worldnews , /r/The_Donald . /r/politics is the subreddit focused on US politics; the user base of /r/politics has been thought to be largely liberal[59]. /r/worldnews focuses more on international news and has frequent discussions on international relations between countries. /r/The_Donald is another US politics focused group, which was founded in June 2015, and has a more republican user base.

We collected the top fifty most frequent and popular users from each subreddit, to infer a model of the user base of the subreddit. We took the Reddit data for each user over a period of two years from June 2015 to November 2017. Using this as a data corpus for the `word2vec` model we created word embeddings for each user to get a point cloud of the vocabulary of the user. Persistent homology was then used to calculate the *barcodes* of the `word2vec` embeddings of each user. Based on the barcodes of each user, the bottleneck distance metric provided a similarity score to every pair of users, which was used by t-SNE to get a low-dimensional clustering embedding of the population, fig. 3.5. The advantage of the t-SNE clustering is the ability to find highly probable clusters (i.e., with a large likelihood), while low probability clusters are ignored,[127].

Based on the t-SNE clusterings, we see a stark similarity between the users of /r/politics and /r/worldnews . This structure not only supports the hypothesis postulating existence of largely liberal user bases in the two subreddits, but also gives a clear method to find echo chambers across the whole Reddit com-

munity. The users of /r/The_Donald are shown to be hugely dissimilar to those of /r/politics and /r/worldnews as the political ideologies of republicans have many contrasting accepted notions than those of democrats.

The idea for using these embeddings and the topological similarity can also be extended to any other spatial model, such as the embeddings computed by `GloVE`, `fasttext`, `sense`, etc. `Sense` embeddings have the additional characteristic of being able to identify polysemy. Thus Topological Data Analysis (TDA) can take advantage of this feature to characterize measures of polysemy between different languages. Nonetheless, one needs to be careful, when considering the potential effects of prevalent topics in the subreddits and to ensure that secondary structures do not dominate the embedding criterion. This goal can be ensured by restricting the topic base to a particular subset so that the vocabulary of the topics remains largely consistent through the subreddits.

3.3.2 Comparison of subreddits gives details of divergence over time

One of the main reasons for performing temporal analysis of language in Reddit is to be able to identify the effects of communications (or lack thereof) between the population on the language of each community. To analyze this effect, we took the most popular topics from each month, from June 2015 till November 2017, in each subreddit and made an incremental `word2vec` model. This incremental model presented to us a highly dynamic picture of each subreddit through time, which we used as an input to the persistent homology toolbox to rigorously quantify the changing similarities over time fig. 3.6.

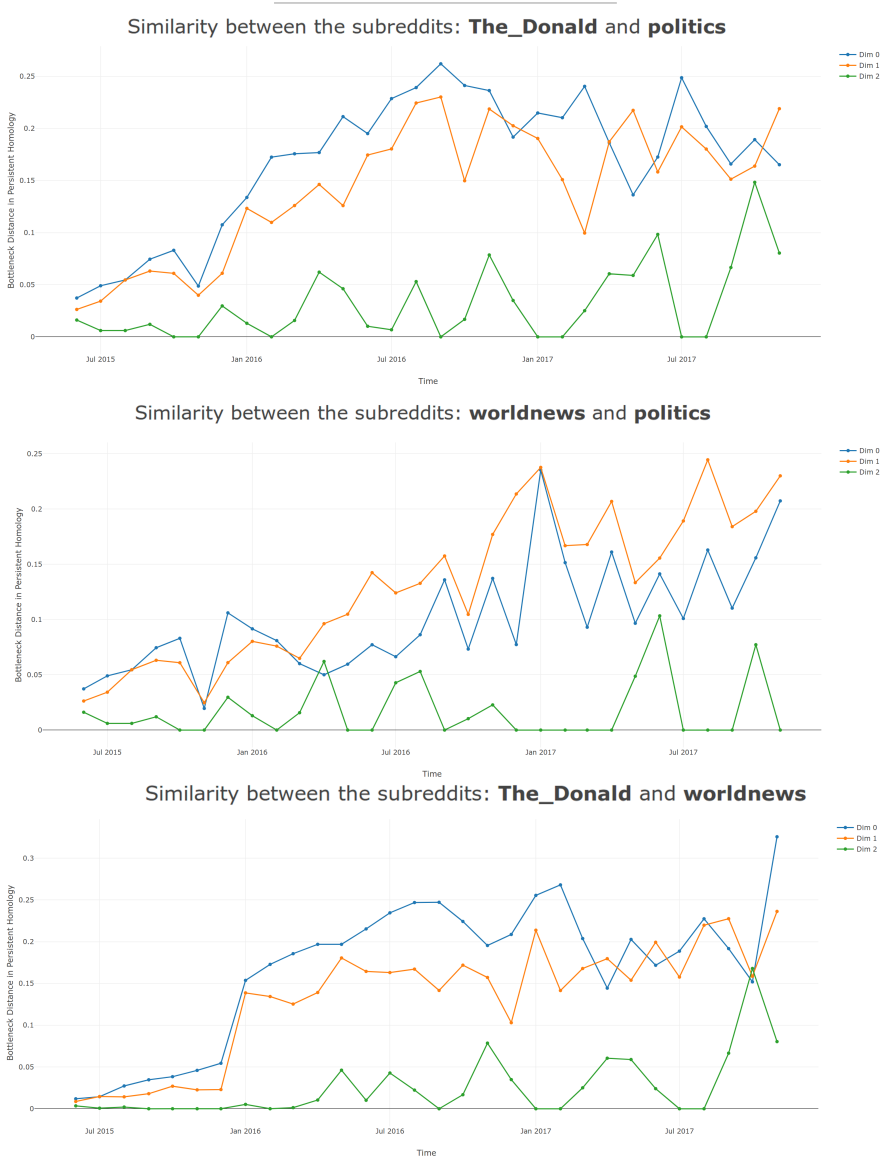

Figure 3.6: Bottleneck distance between subreddits. We collect the most popular posts from every month in each subreddit to build a temporal model for language representation. Using the bottleneck distance of persistence diagrams we can calculate the distance between the language representation over time and see the effects of the community structure. The consistent increase in the distance between the representations confirms the hypothesis of echo chambers in subreddits, leading to a divergence in representations and topic focus between the subreddits.

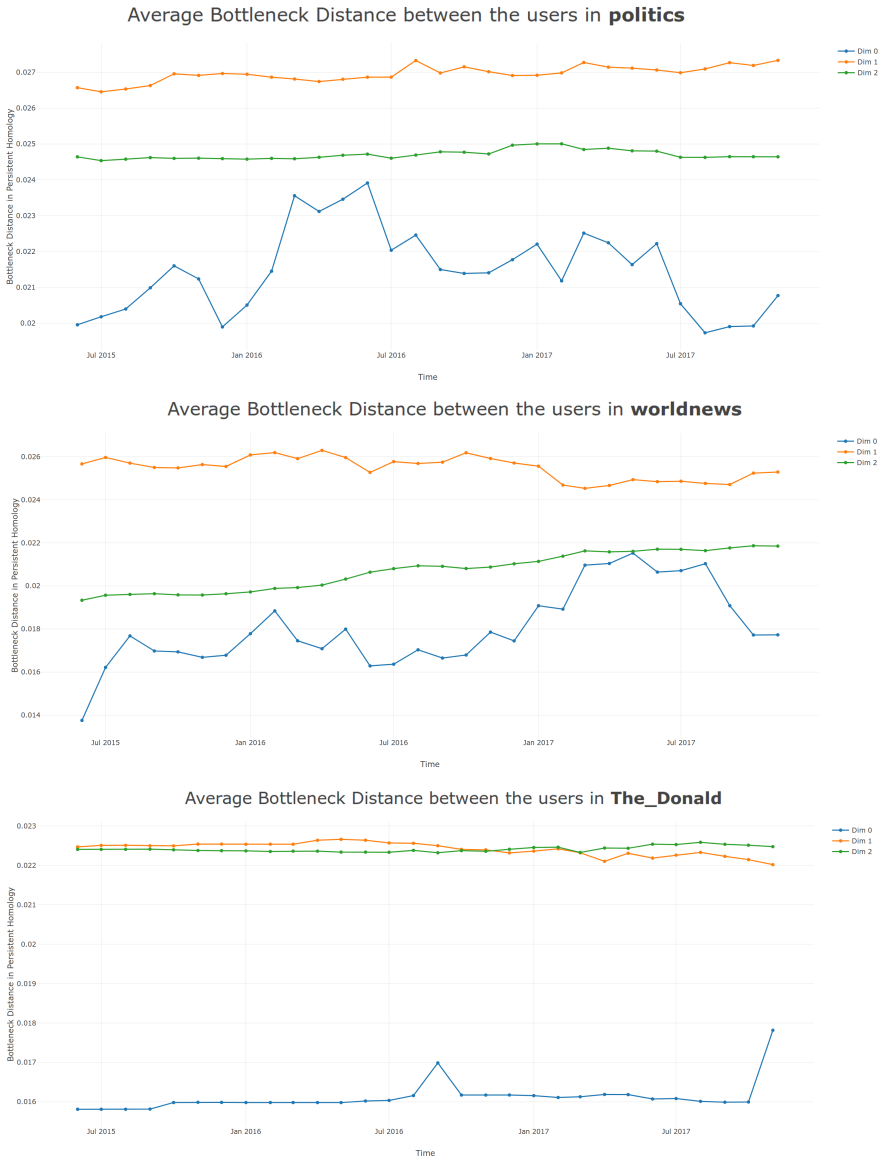

Figure 3.7: Intra-subreddit distance. The individualized embeddings inside each subreddit can help us understand the convergence of language over time and the stability of the language after convergence. The current user distances remain stable over a period of two years suggesting a stable distribution of language representations, where the divergence observed *a priori* is an effect of the drift in language due to shifts in the topic focus over time.

We observe that there is a consistent increase in the relative pair-wise distances of the subreddits. This dispersion corresponds to the formation of communities and how the nascent communities differ in interpreting semantic nature and sentiments of words in the subreddits. The increase in the bottleneck distances can be seen as one effect of the widening division in the population based on political creeds and affiliations.

3.3.3 Non-isotropy of language embeddings

Language isotropy has been thought of as a reason for the robustness of the `word2vec` models and any embedding tool in general. Isotropy in a geometric sense is the measure of uniformity of the word embeddings across the inherent embedding space. The core idea that is assumed to support the word embeddings (and approaches based on them) is as follows: All natural languages must be able to describe all concepts in the language model using minimal combinations of words. This property is facilitated as the words become uniformly distributed across the space[5].

Persistent homology offers an easy way to measure the isotropy of any word embedding model by looking at the point cloud of the embeddings. The presence of holes in the embedding space can be thought of as parts of the space which are poorly described using the current geometry and for which news words should either be introduced or words can be remapped to new meanings, reminiscent of Moran processes in evolution and linguistics[120].

We took the subreddit data from each of the three political subreddits and

calculated the embeddings of the word corpus to get a representation of the words at the end of 2017. We observed the presence of multiple large homology groups suggesting inconsistencies with the hypothesis of isotropy of `word2vec` embeddings. Our observation, albeit in a limited context provokes additional analysis of `word2vec` models and their effectiveness. Another potential investigation is the location of the homologies and identifying the regions of space contributing to the homologies. This strategy may lead to a tool for analyzing a text corpus and identifying topics which can be misrepresented. Such a tool can point to potential pitfalls of the embeddings and also new approaches to avoid them.

3.3.4 Using user data to find similarities of subreddits

One of the reasons for conducting the experiment on a per user basis is to be able to identify the communities from population data and minimal structural information. This new individualized data prompted us to re-perform the previous analysis of subreddit distance based on only the user data. We took the word corpus for each user and made an incremental `word2vec` model to get temporal embeddings of the each user from June 2015 till November 2017. Using these embeddings, we calculated the average distance between each pairs of users in the subreddits to observe changes in the language representations.

The average user distance between the subreddits remained largely unchanged throughout the time period of analysis, painting a different picture than the more robust analysis from the overall subreddit data. This discrepancy prompts

a more detailed analysis of using personalized data to gather succinct information to compare communities. This approach also faces a problem in identifying communities based on individualized data, where no proper means of learning the underlying population graph exists. In a setting where conversations take place with multiple users, the problem of inferring the communication hypergraph is a harder problem[72].

3.3.5 Intra-subreddit language drift using users

To observe the drift in language over time we examine the distance between the representations of each user over time (fig. 3.6). The user data has many limitations, namely, initialization process is slow; vocabulary remains limited; length of conversations is typically short; and most importantly, the best existing data corpus is inadequately small. Due to these limitations, any kind of user based analysis of subreddits has proven difficult. We notice a small pattern of increasing distance, reminiscent of the subreddit distance metric. But the fluctuations in first two homologies show the effect of lack of data on the bottleneck distance.

One way of getting around this limitation is to have robust user data to construct good individual representations of the language. The design of `intuits` is such that the crowd-sourced natural experiments can yield better individual representations, each of which can be tracked over time to get drift of the language and observe the community effects on the representation. Collecting more focused data, such as the ones to be gathered by the `intuit` project, will help

reveal much more about various linguistic hypotheses – ranging from origins of the language to its universality and stability.

3.4 Extensions of the model

We conclude that design and launch of `intuit`'s large-scale crowd-sourced creolization experiment constitutes a feasible project – *proviso*, serious attention is given to language's convergence properties (and subsequent stability). Our computational simulation of Bayesian Echo Chamber and the mathematical analysis of convergence to equilibria within it appear promising for the following reasons: (i) by providing the right tools to a crowd-sourced wiki-like public effort, it seems conceivable to creolize a natural language more suitable for the world-wide web and (ii) furthermore, by not ignoring the effects of naturally occurring population (graph) structures (e.g., reddit), it seems possible to avoid certain natural limitations, usually exhibited as disparate Echo Chambers, coexisting, but in fundamental disagreement with one another. Thus there must be significant efforts to bridge the differences between the idealized theoretical model and extant empirical models, which may be achieved by simply prompting conversations among key individuals, who could facilitate rapid mixing in the population graph. Theories of random graphs, expander graphs and algebraic analysis of graphs provide powerful mathematical tools to achieve these goals algorithmically.

We hypothesize further that a properly designed `intuit` experiment will parametrize the universal grammar (assuming and validating its existence) common to nat-

ural languages; it will quickly converge to a highly stable Nash equilibrium; and it will optimize certain information-theoretic utility functions for the utterer-hearer pairs. These hypotheses are, separately and together, refutable. The data collected from this natural experiment will shed important light on the biological mechanisms responsible for the emergence of human languages, while spurring the emergence of a new wave of language creation.

The experiment also raises additional questions:

How will the intuit language relate to the ongoing research in Artificial Intelligence? Currently there is much interest in using deep learning for natural language processing, especially for language translation, text-tagging, captioning images, etc. – all relying on some form of `word2vec` embeddings based on large corpora from multiple languages. There is a lack of a proper theory in deep learning explaining its spectacular successes and intriguing failures (e.g., adversarial perturbations,[96]) that this version of AI (sub-symbolic, black-boxes) exhibits. Our work on the signalling-game-theoretic models, as initiated here, could be useful in injecting robustness to the future AI research. A particularly colorful example of a confusing experiment in AI involves Microsoft's Tay, which was effortlessly hijacked by a millenials' echo chamber.

How will the intuit language relate to the current thinking in Mathematical Data Science? We have shown here that topological analysis of point-cloud-data provides a powerful tool that could be widely applicable. Some applied works on evolutionary studies in virology and oncology have been influential, but wider applications remain unexplored, especially in the context of the evo-

lution of languages, social norms, social contracts, social institutions, etc., – all topics of immense importance as intelligence/information technologies have begun to disrupt long-standing, hitherto stable institutions in unpredictable manners. Creolization's deeper relations to topological data analysis (TDA), Manifold Learning, Information Geometry, Game Theory etc. are thus important topics of future research.

How will the intuit language relate to the current thinking in Biology? Our experiments anticipate support for the usefulness of distributional methods of representing semantics in a language. Our approach is supported by the analysis by Arora et al.[5], who were able to identify a semantically-relevant low-dimensional shared representation of fMRI responses. Their experiments and analysis were conducted in an unsupervised fashion and involved views of multiple subjects watching the same natural movie stimulus. These studies point to some fundamental questions about the biology of languages and how it evolved in a relatively short period. Our analysis using `intuits` – with its multimodal emoji like structures – is hoped to raise more challenges and resolve ancient mysteries.

Last but not least, *how will the intuit language relate to the current thinking in Linguistics?* Noam Chomsky and his followers have played a dominant role in shaping the current theories of language, but in isolation from other evolutionary researchers and their theories, such as cellularization (codons), endosymbiosis, multi-cellularity, speciation,etc. However, human spoken language is hypothesized to be a biological artefact (postulating a yet-to-be identified lan-

guage organ; related to the so-called I-language; and supporting distributional semantics), but leads to theories that are unexperimentable ("not-even-wrong"). The existence of WWW and crowd-sourcing drastically changes the situation by enabling scalable and experimental inventions of new artificial natural languages using large number of communicating human learners.

However, our biggest challenges will remain in the engineering of the `intuit` Linguistic System, focusing on how the data should be collected and how it should be analyzed. We can use existing efforts developed in cloud computing (e.g., BigTable, BigQuery, etc.), enabling construction of such a system with relatively small man-power. But given that internet is already affecting how younger generations communicate (with hashtags, emojis, acronyms, etc.), the window of opportunity for the natural experiments based on `intuit` may be closing soon, particularly as the field gets crowded by powerful monolithic corporations, namely, the so-called unicorns e.g. Twitter (tweets), Facebook (identity systems) and Google (Language Translations).

4

Efficient Agony Based Transfer Learning for Survival Forecasting

4.1 Causation and Progression

*Cancer progression modeling is a mature subfield of cancer informatics[41]. The desirable models seek to recapitulate or forecast the accumulation of genomic events in the course of a patient's disease. Given these purposes, progression models often take the form of hierarchical combinatiorial structures such as phylogenetic trees[3 67] or various forms of Bayesian networks[13,50]. In this chapter we consider a cancer progression model (CPM) to be a directed acyclic graph (DAG) defined over a collection of (epi)genomic events. This view encompasses the structures of both phylogenetic trees and Bayesian networks and is agnostic with respect to probabilistic assumptions or interpretations of any particular CPM.

Research on CPMs often focuses on accurately recreating an underlying ground truth. Most research pipelines, involve the presentation of a new algorithm and showing empirically or rigorously that this algorithm reconstructs a latent CPM correctly. These methods are, for the most part, retrospective and are predicated on the theory that understanding the course of evolution of a particular patient population will shed insight into the nature of that particular cancer and, hopefully, its treatment.

To our knowledge CPMs have not yet made this final clinical leap. In particular while similarities between CPMs have been explored via edit distances[70], the similarity between progression models across different cancer types have not been exploited. As cancer progression seems to correlate with phenotype it fol-

*Co-authored with J. Bannon, B. Mishra, bioRxiv:2021.02.24.432695

lows that patients with similar disease progressions will be similar in terms of clinical presentation.

This chapter contains two main contributions that address these issues. In order to compare progression models across different cancer types we *first* introduce a notion of similarity based on the graph theoretic concept of "agony." This is an alternate notion of distance that measures the preservation of structural (involving driver genes) relationships in a progression model. Thus the semantics of looking at agony directly correspond to the orderings of events given by two CPMs.

The *second* contribution of this chapter involves using this notion of distance to automate transfer learning, with specific experiments directed towards survival forecasting. In transfer learning one seeks to leverage the learned capability to perform *source task* to improve the ability to perform a *target task*. Usually the choice of source and target based on the fact that they are in some ways similar. Here we assume that similar progression models correspond to similar phenotypes. We fix the target task as forecasting survival time for a particular cancer and then choose the source task as predicting survival from the cancer which has the smallest agony based distance from the target task, fig. 4.2. We show empirically that the comparison metric introduced corresponds to meaningful biological similarities and that agony-guided transfer learning significantly improves performance in some cases.

The rest of the chapter is structured as follows. In section 4.1, we review the necessary background material. In particular we review progression models, sur-

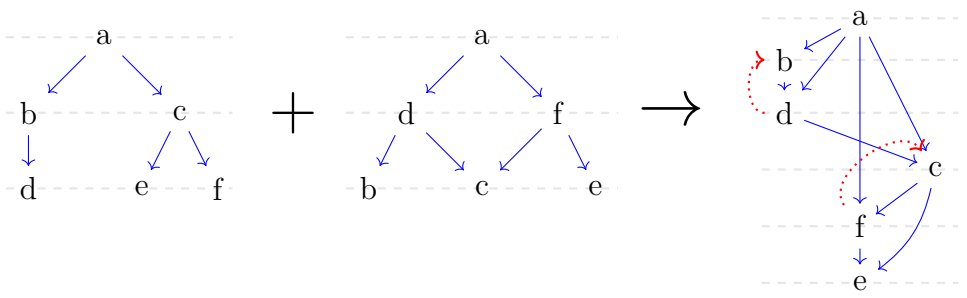

Figure 4.1: The grey lines represent the minimal rank function, and the red arrows are the edges contributing to the *agony*. In this case the agony distance is 4.

vival forecasting, and transfer learning and we introduce the notion of *agony* a way of measuring the similarity of two progression models. Then in section 4.2, we report on two experiments in using agony for bioinformatic purposes. The first experiment consists of clustering patients in different cancers using pairwise agony dissimilarity. The second experiment automates source task selection in transfer learning using minimum-agony distance as defined in section 4.1. Finally in section 4.3 we provide a discussion and pointers to future work. We include an online appendix where we include all technical details [†].

4.1.1 Graph Agony

Agony[53] is one of many measures assessing hierarchies in directed graphs, some of these are known to be computationally intractable[42]. Agony, however, is computable in polynomial time. In particular, agony is a measure of the degree to which a directed graph is acyclic[117,118]. To our knowledge it is the only computationally tractable method for comparing pairs of directed acyclic graphs,

[†]https://github.com/epsilon-0/pan-cancer

and yet approximates as close as other graph distance functions.

Definition. 12 (Rank function)**.**—A rank function on a graph $G = (V, E)$ is a map $r : V \to \{1, \ldots, |V|\}$.

Intuitively the rank function can be thought of as specifying an ordering on the nodes of the graph. In the context of a progression model the rank function could be thought of as a hypothesized temporal ordering or as assigning certain mutations to *levels* in a hierarchy.

Definition. 13 (Agony of a Graph With Respect to a Rank Function)**.**—For a graph $G = (V, E), V = \{v_1, \ldots, v_n\}$ and a rank function $r : V \to \{1, \ldots, n\}$ the agony of G, with respect to r, is defined as

$$\mathfrak{A}(G, r) = \sum_{v_i \to v_j \in E} max(0, r(v_j) - r(v_i) + 1) \tag{4.1}$$

Clearly from eq. (4.1) the larger the difference in rank between a parent node and its child, the larger the agony. In practice a given rank function r is not available *a priori* which is an issue because the value of $\mathfrak{A}(G, r)$ is highly dependent on r. In light of this, we define the general agony of a graph as follows:

Definition. 14 (Agony)**.**—For a graph G we define the *agony of the graph* G to be

$$\mathfrak{A}^*(G) = \min_{r:V \to \{1,\ldots,n\}} \mathfrak{A}(G, r) \tag{4.2}$$

For the rest of this chapter, whenever we refer to the *agony of a graph* we mean it as given in definition 14 unless otherwise stated. It is not obvious from

the form of eq. (4.2) that $\mathfrak{A}^*$ is computationally tractable. However, if one constructs the dual problem to eq. (4.2) one arrives at the maximal Eulerian subgraph problem, which can be solved efficiently. We refer the reader to[117] for details.

Definition. 15 (Agony Between Graphs)**.**—For two graphs G_1, G_2 on the same set of nodes, V, let G' be the union of the two graphs, without duplicate edges. The agony between G_1, G_2 is defined as

$$\mathfrak{A}^*(G_1, G_2) = \mathfrak{A}^*(G') \tag{4.3}$$

In the case where G_1, G_2 are progression models then eq. (4.3) can be thought of as a measure of mutual inconsistency. If there are conflicting or contradicting paths in G_1 and G_2 then $\mathfrak{A}^* > 0$, fig. 4.1. For the case where G_1, G_2 are in fact progression models, eq. (4.3) possesses properties useful for comparing them.

Indeed, the algorithm to compute the agony goes about by finding the smallest cycles to remove from the combined graph, until what remains is a valid progression model.

Lemma. 4.1.1 (Agony Pseudometric)**.** The *agony* between graphs can be used as a pseudometric on the space of *directed acyclic graphs*. That is for any two directed acyclic graphs G_1, G_2 the following hold:

$$\mathfrak{A}^*(G_1, G_2) = 0 \text{ if } G_1 = G_2 \tag{4.4}$$

$$\mathfrak{A}^*(G_1, G_2) = \mathfrak{A}^*(G_2, G_1) \tag{4.5}$$

In general, the triangle inequality — necessary for the status of being a full metric — does not hold. However, most of the generally occurring cases have the triangle inequality to be valid, which we leverage for using this as a fast approximation for a proper metric.

As a consequence of lemma 4.1.1 we can use eq. (4.3) to compare two progression models derived from different patient populations. We can then investigate whether or not two patient populations that have similar progression models are phenotypically similar. In this chapter, we look at one highly relevant phenotype: disease aggressiveness as measured by forecasted patient survival time.

4.1.2 Survival Forecasting

A common problem in clinical oncology — and clinical care in general — is forecasting the time until a meaningful change in the patient's condition occurs. For example the time until death, the time until disease progression, or the time until the acquisition of a particular hallmark. Data of this type consists of a duration, which is usually measured from the beginning of an observation period (e.g. a clinical trial) until the event of interest occurs. In general a time-to-event data set $\mathcal{D}$ for a survival forecasting problem involves observations of the form

$$\{x_i, t_i, \delta_i\}_{i=1}^{n}$$

where $x_i = (x_{i_1}, \ldots, x_{i_m})$ is a vector of covariates, t_i is the time of the event or censorship, and δ_i is an indicator variable marking that the event was truly observed ($\delta_i = 1$) or if censorship has occurred ($\delta_i = 0$). There are many ap-

proaches to survival analysis – most either involve learning a function $f(x)$ that returns a predictive survival time or compute a hazard ratio for a specific value of x. The most well-known approach is Cox' proportional hazard model[35] which assumes that the probability that the event occurs at time t follows an exponential distribution. We refer the reader to the two monographs[79,94] for a thorough treatment.

The machine learning literature has also attempted to address the survival problem with standard methods such as the support vector machine[130,48] and deep neural networks[133,133]. For our experiments we used a deep neural network model called SurvivalNet as our survival forecasting method. We chose this method because the software had pre-implemented several important neural network techniques such as drop-out and Bayesian optimization for model hyperparameters[133‡].

The most common method for evaluation of survival forecasting is the *concordance measure*[62] which compares the relative risk assigned to patients by the model to the order in which they actually died. Correctly ordered pairs are rewarded and incorrectly ordered pairs are penalized. We evaluate our survival forecasting models by concordance on a test set and prove the translational utility of the system.

‡Unlike the blackbox models from SVM and DNN, SBCN can provide an evolutionary/causal explanation for the cancer prognostics.

4.1.3 Transfer Learning

Transfer learning[100,31] is a technique in machine learning where information from a *source task* is used to improve performance on a *target task*. For example if the task is to recognize cars in natural images a transfer learning approach might be to train a system to recognize trucks and then apply it to images looking for cars. Clearly part of the consideration here is that trucks and cars are, in some ways, similar. While research in transfer learning focuses on many different aspect of the *process* of transferring knowledge[100], the source and target tasks are usually treated as given. For clinical oncology, source and target tasks should be similar in a way that is *biologically significant*. For example, in[133], the authors augment the training data for BRCA cancers with patient data from OV and UCEC because they are all hormone-driven tumors.

We generalize this by performing experiments where we augment the data by choosing the cancer with the closest (least agony) progression model. Specifically we begin with a *target task* T which is the survival forecasting problem for a particular cancer $\mathcal{C}$, and then choose as the source task the data from the cancer $\mathcal{C}'$ which has the smallest agony distance from $\mathcal{C}$.

4.2 Analysis of TCGA pan-cancer dataset

Our aim is to be able to evaluate the utility of agony as a (dis)similarity measure between different cancer (sub)types. To evaluate how well agony is capturing biological information we first performed two clustering experiments,

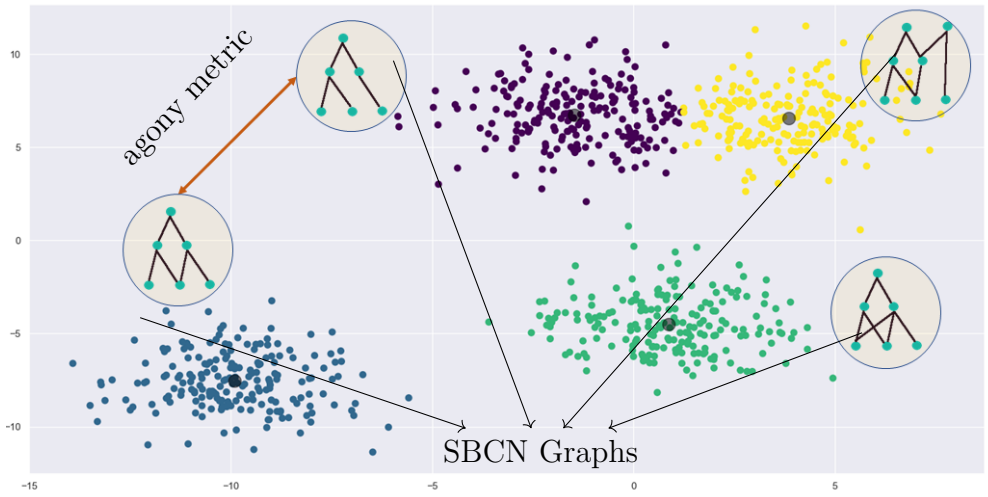

Figure 4.2: Pipeline for transfer learning with patient clusters with their Suppes Bayes Causal Network(SBCN). Each cluster has its own progression model, which are then used to calculate the distance between clusters. This in turn is used to automate transfer learning by selecting low agony pairs to boost the dataset information content for enhancing the survival forecasting toolbox.

detailed in section 4.2.1. In order to make our approach *translational* we performed transfer learning experiments between low agony and high agony cancers, and show empirically that low agony transfer learning improves performance more than high agony. We report on these results in section 4.2.2.

Data Sources and Preprocessing

For our experiments we used the data from the TCGA 2018 PanCancer Atlas accessed via cBioPortal. This data consists of approximately 11,000 genes spread across 33 different tumor types. Before all experiments the mutation data from each cancer was processed into a binary matrix $\mathbb{D} \in \mathbb{Z}_2^{n \times m}$ where each row corresponded to one of the n patients. The m genes were filtered to be only those that were either included in both tiers of COSMIC or those considered to

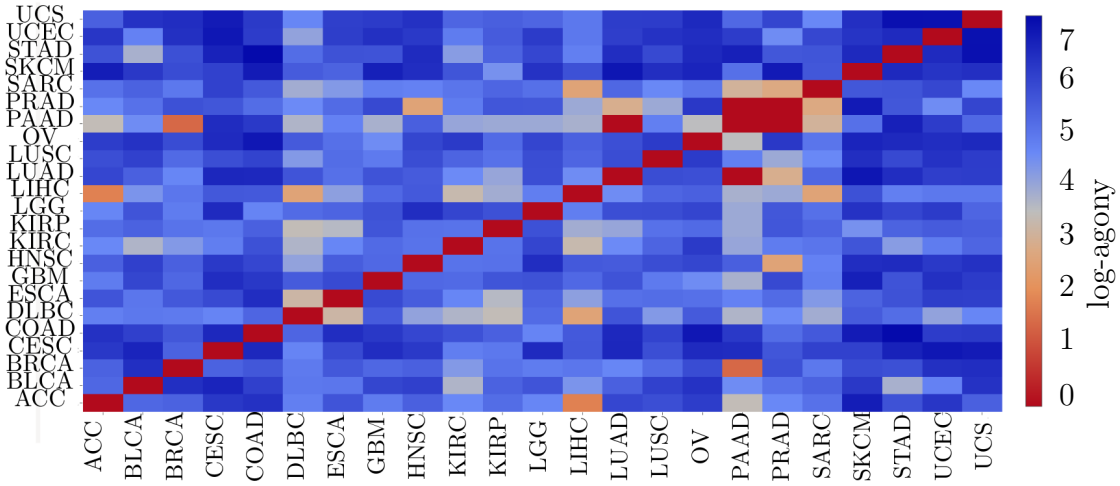

Figure 4.3: Heatmap for agony distance between different cancer types. We plot the $\ln(\text{agony})$ between the cancer types, with the red representing lower agony measure.

be driver genes by TCGA.

4.2.1 Agony Recovers Biologically Meaningful Dissimilarity

Agony across clusters shows biologically significant similarities

The first experiment was to explore the similarities between cancers based on their respective progression models. The first step involved fitting, for each cancer type $\mathcal{C}$, a SBCN $\mathbb{G}_{\mathcal{C}}(V, E)$. For each cancer type the top 100 most frequently mutated genes were selected and a SBCN was constructed using point mutations in these genes as the nodes. Each SBCN had approximately 1,000 edges. For each pair of graphs we calculated the *agony* distance between the pair in accordance with eq. (4.3). In fig. 4.3 we plot the pairwise agony values for each cancer type on a logarithmic scale, and show a clustering of the data using a multi-dimensional scaling (MDS) embedding of the data into two dimensional

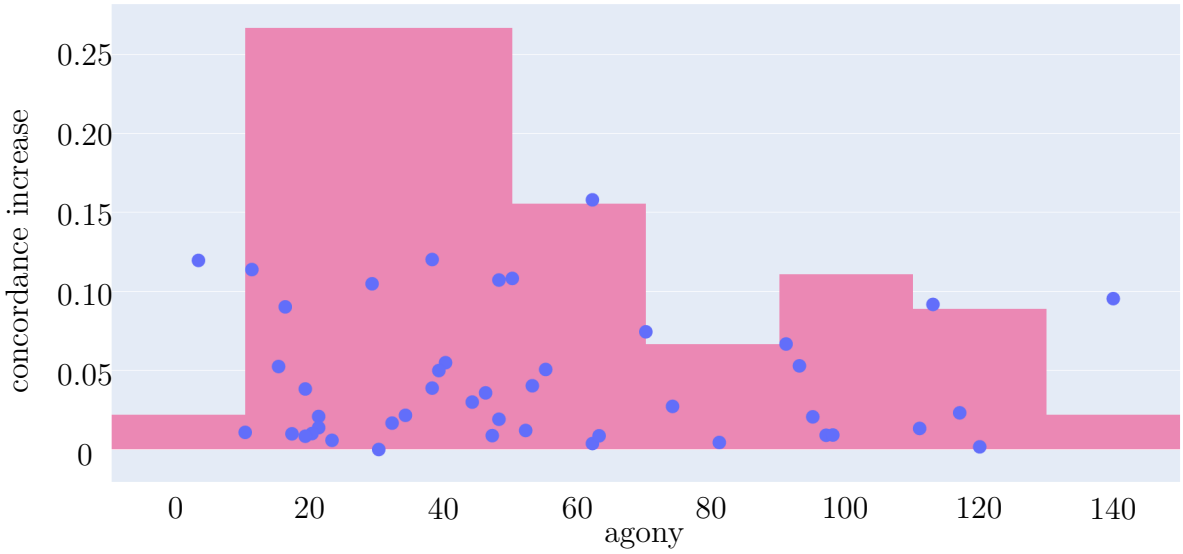

Figure 4.4: Concordance as a function of agony distance. Out of the chosen low-agony pairs, we plot the **increase in concordance** vs **agony** and we notice a higher concentration of pairs with a larger increase in concordance for low agony measure. The histogram shows a decrease in density for pairs which can do transfer learning to increase concordance, validating our hypothesis.

space.

We observe that skin cutaneous melanoma has a high dissimilarity to other cancers, which may be hypothesized to be due to the central role played by *BRAF* in controlling growth and apoptosis. Further analysis, such as using phenotype matching, would be necessary to validate this hypothesis. We also note that in the MDS clustering the hormone-driven cancers UV, BRCA, and UCEC are farther from the other subtypes, which suggests the agony distance has recovered this distinction. However we expected these three to be close to each other in agony distance, which is not the case. Our hypothesis is that since BRCA has well delineated subtypes[71], due to Simpsons' paradox[125] the overall BRCA population is not representative and that combining them loses informa-

tion. We check this in the next subsection.

Agony across BRCA subtypes captures known dissimilarities

To see if agony could capture well-known subtypes, and to assess if Simpson's paradox played a roll in the overall dissimilarity of BRCA to other cancers, we stratified the TCGA PanCancer BRCA patients into the five subtypes given in the data: Luminal A (LumA, $n = 499$), Luminal B (LumB, $n = 197$, Her2-Enriched (Her2, $n = 78$), Basal-like (Basal, $n = 171$), and Normal/Untyped (Normal, $n = 36$).

In fig. 4.5 we present the pairwise agony distances across the BRCA subtypes. From the data in fig. 4.3 and fig. 4.5 we hypothesize that agony is recovering meaningful discrepancies in progression models and as well as meaningful *phenotypic differences.* This hypothesis motivated our use of agony as a metric for selecting the source task in transfer learning, which we try to validate by observing the boost (or lack thereof) in the concordance obtained by black box learning.

4.2.2 Transfer Learning

In the following experiments we split the data into train (60%), validation (20%), and test (20%) sets. Survival net was run with default values including automatic procedures for regularization and hyperparameter fitting. In our transfer learning experiments we did the actual transferring by adding the data from the source cancer to the training data.

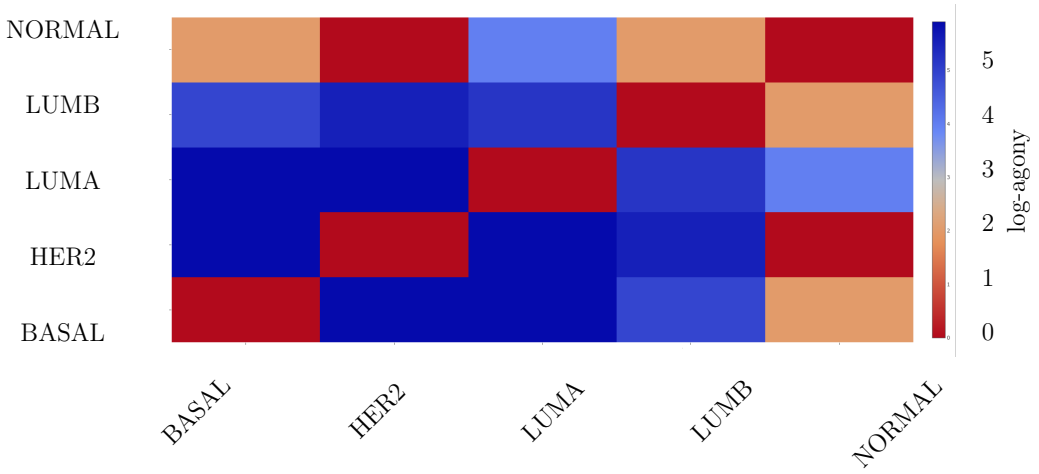

Figure 4.5: Heatmap for agony distances between the BRCA subtypes.

Low *agony* identifies similar cancer types, by improving concordance

We refer to fig. 4.3(b) to infer that most of the high concordance increase happens in the region of low agony metric, giving a confidence boost to the fact that lower agony distance is a good indicator of the accuracy gained from transfer learning.

For showcasing actual cancer pairs we chose LUAD as the dataset to be augmented. For LUAD two cancers which have a low agony distance are PRAD and MESO. These have 566, 494, and 87 patients, respectively. We performed two transfer learning experiments on these three cancers. First we evaluated SurvivalNet on LUAD alone, achieving a mean concordance 0.552 and a median concordance of 0.554 over 93 rounds of training and testing. We then augmented the LUAD training data with the data from PRAD (100 rounds), which lead to an increase in mean and median concordance to 0.735 and 0.7324 re-

Data	Mean Concordance	Median Concordance	Log Agony
LUAD	$0.5522, [0.5421, 0.5620]$	$0.5542, [0.5424, 0.5663]$	—
LUAD+PRAD	$0.7353, [0.7263, 0.7440]$	$0.7324, [0.7225, 0.7389]$	3.16
LUAD+MESO	$0.6317, [0.6204, 0.6432]$	$0.6395, [0.6524, 0.6251]$	0
STAD	$0.5714, [0.5574, 0.5855]$	$0.5759, [0.5680, 0.5926]$	—
PAAD	$0.6120, [0.5966, 0.6278]$	$0.6239, [0.6524, 0.6066]$	—
STAD+PAAD	$0.5748, [0.5672, 0.5826]$	$0.5780, [0.5671, 0.5920]$	7.11891

Table 4.1: Confidence intervals and statistics for agony based dissimilarity.
We report the mean and median concordances for our experiments along with bootstrap 95% confidence intervals, across 60 runs. A single cancer name represents running survival net on that data alone. Other rows take the form of TARGET+SOURCE. For transfer learning experiments we give the pairwise log agony distance.

spectively. The difference in distribution means was statistically significant (Wilcoxon p =2e-16). To check that this was not simply the result of an increased amount of training data, we performed the same experiment with MESO (83 rounds) as the source. This analysis shows a modest increase in mean (0.631) and median (0.639) ($p = 6.025\text{e-}16$). A figure demonstrating the improvement is given in the appendix.

High *agony* identifies vastly different types does not improve concordance

It is possible that simply combining *any* two datasets might yield increased concordance. To test this we performed the same experiments as above but with a *high agony* pair. Specifically we performed survival forecasting on STAD ($n = 440$) and PAAD ($n = 184$) and then a transfer-learning with PAAD as the source. In this context survival net performed well on STAD (54 rounds) and PAAD (60 rounds) individually but in the transfer learning context (44

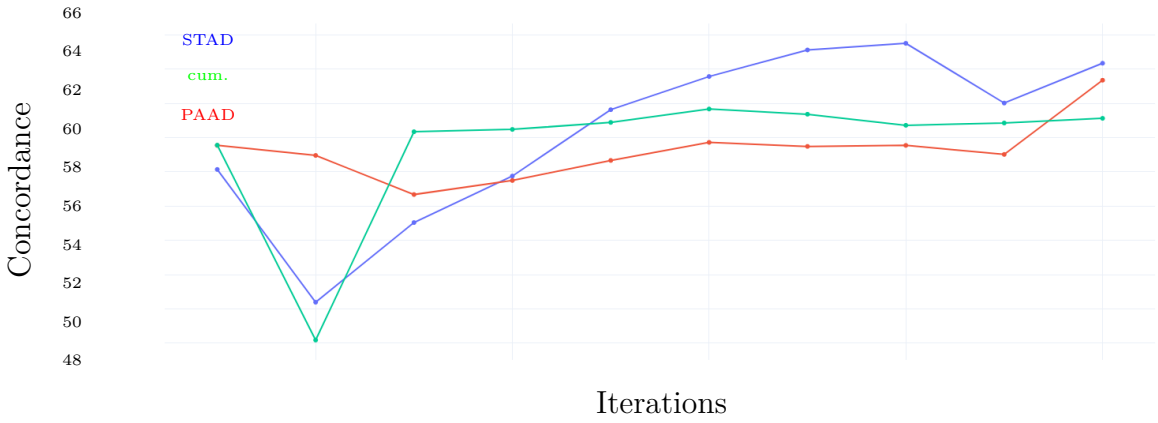

Figure 4.6: Transfer learning with large *agony* for cancer types Pancreatic Adenocarcinoma (PAAD) and Stomach Adenocarcinoma (STAD) shows no improvements in concordance score.

rounds) performance the concordance did not meaningfully change. A Wilcoxon test on the concordance of STAD and STAD+PAAD has a p-value of 0.7659, so we fail to reject the null hypothesis that the distributions are the same. We show the lack of improvement citing the results in fig. 4.3(b) and table 4.1 for STAD+PAAD. The regression curves show a distinct decrease in concordance gain with the increase in the agony metric, in addition to which the showcased example, STAD+PAAD, shows a statistically insignificant change in the accuracy values.

While these results are not definitive, they suggest that agony is capturing a meaningful distinction *phenotypically* and could be used to guide further transfer learning experiments. Such information is useful for important clinical applications, which include drug repurposing[104], early intervention[7] and immunotherapy[55].

4.3 Extensions of the model

We have proposed agony as a novel method of quantifying the (dis)similarity between progression models by discovering conflicts in their hierarchical relationships. We have shown empirically that this measure recovers known biological similarities and differences in cancer types. Finally we showed the potential for clinical utility by using agony to automate the choice of a source task in transfer learning experiments. To our knowledge this is the first biological attempt to automatically solve the source-selection problem, which is of research interest in the artificial intelligence community. Our experiments showed a correspondence between low agony distance and increased task performance.

Our approach can be easily generalized. Agony clustering is agnostic to the semantics of the underlying CPM but, since it measures pairwise inconsistency, it directly accounts for the semantics of whichever CPM is chosen. Also, our transfer learning methodology is amenable to any machine learning technique. An obvious next step is to use agony to compare different progression models. Another option is to vary the machine learning task in question. One can even generalize beyond machine learning to investigate whether populations with similar progression models are similar in *any* interesting phenotypic characteristic.

In the near term we hope to expand on the theoretical foundations of agony to large graph limits and to graphons[87]. This generalization would allow us to bring large sample theory to bear on the techniques presented in this chapter,

and potentially allow us to generalize SBCNs to cases of continuous variables, e.g. gene expression. Graphons have also received growing attention from the machine learning literature[4,45]. We believe that even its current form graph agony can be a valuable tool for both clinical and research cancer bioinformaticians.

5

Efficient Evolutionary Models with Digraphons

5.1 Introduction

Graphical models are one of the most important tools used in machine learning[74] and arise in most applications which involve pairwise interactions, such as mutations in cancer evolution[134], protein networks[58,129], hierarchical network models[39], influence in social networks[91,108,56], population dynamics[17] and many more. In machine learning, there are various techniques which forgo the use of these graphs and instead employ more algebraic representations to take advantage of the underlying theories, such as latent variable models[86], network or dynamic models[82], deep neural networks (DNN)[123,93], clustering models[23,28]. The key advantage of the later techniques is the reliance of the abundance of techniques developed in linear algebra and optimized algorithms for doing fast implementations to get efficient real world analysis.

One of the main motivations of this study is the case of evolutionary populations, where the evolution is modeled as interactions between individuals of the populations, such as mutations, genotypic variations and phenotypic selection. In such cases, evolution of the gene regulatory network (GRN) or the protein-protein interaction (PPI) network happens by specific events, such as insertion, deletion, duplication, point mutations, translocation and inversion[68,46]. Of these the insertion, deletion and duplication events have the most noticeable effect on the networks and have an easily observable effect on the phenotype. If we think of genes as nodes in a graph and gene interactions as edges, these events can be thought of as an edge or node insertion, deletion or duplication.

When our network starts evolving and growing in size, we naturally think about what the outcome of such a process would be. As our network keeps on increasing in size, we need to extend our definition of a graph and naturally arrive at the intuition of a *limit graph*, i.e. a graph on an infinite number of nodes, which we analyze not by looking at properties on non-empty subsets of the graph. These dynamics can be represented in terms of limit networks with the help of *digraphons*. A *digraphon* is measurable function $G : [0,1]^2 \to [0,1]$. Given a digraphon G, there is a corresponding countably infinite exchangeable, definition 16, graph $\mathcal{G}(\mathbb{N}, G)$, with the adjacency matrix $(\mathcal{G}_{ij})_{i,j\in\mathbb{N}}$ defined by the generative model

$$U_i \overset{\text{iid}}{\sim} \text{Uniform}[0,1] \quad \forall\ i \in \mathbb{N}$$
$$\mathcal{G}_{ij}|U_i, U_j \overset{\text{ind}}{\sim} \text{Bernoulli}(G(U_i, U_j))$$

Thus the digraphons are an ideal object to use for a generative model for evolutionary networks. Any useful tool which is to be used for real world analysis must also support *hypothesis refutability*, which necessitates the notion of similarity between digraphons. The space of digraphons have many norms defined on it, such as the l_p norm

$$\|G\|_p = \left(\int_{(i,j)\in[0,1]^2} (G_{ij})^p \right)^{\frac{1}{p}}$$

or the more interesting, *cut norm*, which can be thought of as the maximum

dissimilarity in a bounded region,

$$\|G\|_{\square} = \sup_{S,T \subseteq [0,1]} \int_{S \times T} G$$

The cut metric is quite complex to calculate and does not yield the most intuitive results for causal models. One of the key points which we wish to capture between two causal models, is a notion of *contradictory information* among them. We wish to penalize *larger* contradictions more than *smaller* ones. Edges which are a part of larger cycles should be given a higher penalty than as the presence of a large cycle would imply that many of the edges could be spurious, which is a bad position for a causal model to be in. The *agony* heuristic[53], definition 14, helps us bypass this by using an error metric dependent on the length of the cycle.

These techniques for analyzing digraphons have recently been developed and have yet to see a wider use in conjunction with the standard Bayesian statistical tools prevalent in machine learning. The notions of limit graphs and asymptotic behaviours of evolutionary models are very important in using generative models from Bayesian statistics as the above model for digraphons seems to suggest an intuitive method for reasoning an approximation of the parameters.

The current black box learning from DNNs falls short of generating an explainable hypothesis which is needed for refutability. Indeed such scenarios have previously been observed such as the universal adversarial perturbations, which have then been leveraged to design adversarial networks. But these still fall short of an ideal answer.

The importance of these tools has been seen in many places, such as those used for signaling games[78], population dynamics and biomolecular networks section 1.1, mesh network topologies[38], 3-D neural imaging reconstruction[64], etc. These techniques work directly in complement to the notions from deep neural networks by providing an explainable AI which can then be used as a hyperparameter in designing the DNNs by affecting their layer hierarchies, activation functions, dropout scenarios and memory length estimates.

5.1.1 Contributions

We present two main contributions. We show a generative model for digraphons using a finite basis of subgraphs, which is representative of biological networks with evolution by duplication[99]. We then show a MAP estimate on the Bayesian non parametric model using the Dirichlet Chinese restaurant process representation, with the help of a Gibbs sampling algorithm to infer the prior. We discuss how this can be generalized to other priors due to the simplicity and extensibility of the model.

Next we show an efficient implementation to do simulations on finite basis segmentations of digraphons. This implementation is used for developing fast evolutionary simulations with the help of an efficient 2-D representation of the digraphon using dynamic segment-trees with the square-root decomposition representation. We further show how this representation is flexible enough to handle changing graph nodes and can be used to also model dynamic digraphons with the help of an amortized update representation to achieve an efficient time

complexity of the update at $O(\sqrt{n}\log n)$, where n is the number of nodes in the digraph.

5.2 Background

Starting in 2006, in a series of papers by László Lovász, et al.,[19,88,20,21] the theory of large graphs and graph limits has developed an elegant and delightful theory unifying parts of topology and analysis for graph theory. This theory has been developed for dense graphs and has connections to graph homomorphisms, graph property testing, extremal graph theory, Szemeredi's regularity lemma, etc.

Intuitively, a graph property (characterized by presence of a subgraph H in G) is testable if, as G grows larger, the ratio of copies of H in G also converges. This notion is important in evolutionary models, as biomolecular networks from by duplication, where duplication happens by *preferential attachment*, which very closely resembles the aforementioned model.

Such limit graphs depend on the notion of *exchangeability*, which is important for ensuring that the order of sampling our initial conditions does not affect our final outcome.

Definition. 16 (Exchangeability)**.**—Let $\{X_i\}_{i\in\mathbb{N}}$ be a sequence of binary random variables. They are *exchangeable* if

$$\Pr\left(X_i = e_i \;\forall\; i \in [1\ldots n]\right) = \Pr\left(X_i = e_{\pi(i)} \;\forall\; i \;\in [1\ldots n]\right)$$

for all n and all permutations π, $e_i \in \{0, 1\}$.

This is used in the famous *De Finetti's theorem*, which says

Theorem. 5.2.1 (De Finetti[40,73])**.** Let $(X_i)_{i\in\mathbb{N}}$ be a binary, exchangeable sequence, then

1. $X_\infty = \lim_{n\to\infty} \frac{X_1 + \ldots + X_n}{n}$, exists with probability 1.
2. There is a probability distribution η on $[0, 1]$, given by $\eta(S \subseteq [0, 1]) = \Pr(X_\infty \in S)$ and

$$\Pr(X_i = e_i \ \forall \ i \in [1 \ldots n]) = \int_0^1 x^s (1 - x)^{n-s} \, \eta(\mathrm{d}x)$$

   where $s = \sum_{i=1}^n X_i$.

The crux of this theorem is to allow us to observe events, not necessarily independent, in a random order and still be able to give an estimate of the limit sequence. This is also known as *De Finetti's Strong Law of Large Numbers*. This is crucial in evolutionary networks where we are almost never going to have independence.

For graphical models, aka directed graphs, the notion of exchangeability is the same as saying that the distribution is invariant under vertex re-orderings,

$$(G_{i,j})_{i,j\in\mathbb{N}} \sim (G_{\pi(i),\pi(j)})_{i,j\in\mathbb{N}}$$

And is also equivalent to saying that the permutation is invariant under all finite re-orderings of vertices.

Definition. 17 (Digraphons and dikernels)**.**—A digraphon is a bounded, measurable function : $[\mathbf{0},\mathbf{1}]^{\mathbf{2}} \to [\mathbf{0},\mathbf{1}]$.

Dropping the condition on the image space, a dikernel is a digraphon where the image space is $\mathbb{R}$, $\mathbf{K} : [0,1]^2 \to \mathbb{R}$.

For our purposes, we will be focusing on digraphons as they have enough information to model evolutionary networks. It can be shown that all directed graphs (in our case, evolutionary models) can be obtained from such digraphon limits by theorem 5.2.2.

Theorem. 5.2.2 (Diaconis–Janson[40])**.** Every exchangeable random, countably infinite, directed graph can be obtained as a mixture of $G(\mathbb{N}, \mathbf{W})$, for some random digraphon $\mathbf{W}$.

5.2.1 Evolution by Duplication

Graph and network evolution has a rich history, from a variety of real world applications, such as social networks, recommendation systems, language modeling, ad systems, etc. The history can be traced back to the works of Erdös and Rényi, and a number of other models have been developed since then.

Biological networks have important non-intuitive properties, such as clustering, small world property (small degrees of separation)[128] and pronounced groups, aka "hubs", where most of the interactions take place[11]. Any network model for evolution is expected to capture these properties in the model. Even more complex information theoretic asymmetry can be incorporated when using game theory to model these networks[101,8,115].

Small World models

Erdös and Rényi graphs have a low tendency for clustering, where clustering is calculated from the degree distribution of the graph. For ER graphs, the degree distribution, for edge probability p and n graph nodes, is given by the Poisson distribution in the limit:

$$\begin{aligned} \Pr(\deg(v) = k) &= \binom{n-1}{k} p^k (1-p)^{(n-k-1)} \\ &\sim \exp(-\lambda)\frac{\lambda^k}{k!}, \qquad\qquad \lambda = pn, n \to \infty \end{aligned}$$

Which results in a clustering coefficient of $\frac{p}{n}$. This is a very low clustering coefficient, which is not exhibited by the real world biomolecular networks[85]. To rectify this, the *small world models* were introduced[128]. Such a graph is constructed as a *ring lattice*, in a two step process, for some hyper parameter K

1. *wire*: connect every node to $K/2$ nodes on either side of the ring.
2. *rewire*: for every edge to a node, add another edge with probability p.

The *wire* step ensures presence of local clusters, while the *rewire* step ensures the presence of small degrees of separation. This results in a average of $(1+p)nK$ edges in the graph and a clustering coefficient of $\frac{3(K-2)}{4(k-1)}$, which is independent of the graph size. This solves two of the three major points needed for real world models but leaves out the last one, hubs or high degree nodes.

Scale free networks with preferential attachment

It is hypothesized that most biomolecular networks have scale-free properties[11,9], i.e. the number of nodes, n_k, of degree k is independent of the size of the graph and is instead inversely proportional to k,

$$n_k \propto k^{-\beta}$$

for some $\beta > 1$, called the *coefficient characteristic* of the network, with the value of $\beta \in [2, 3]$ for eukaryotic organisms. In scale free networks, there are nodes with high degrees with a relatively larger probability, termed as *fat tail* distributions. These distributions are better than exponentially decaying distributions found from small world models and allow for the emergence of *hubs* with an inverse polynomial probability. Power law distribution is more common in social interaction networks, such as internet and phone call maps and interpersonal collaboration networks.

Preferential Attachment.

Preferential attachment models are a specific subtype of scale free networks, in which evolution occurs by duplication at a local scale[10]. The evolution happens using local *growth rules* which lead to global characteristics from the small world and scale free models as a consequence of the power law distribution.

Given a graph G_0 and a probability p, the preferential attachment can happen in two ways

1. *vertex step*: Add a new vertex v and an edge (u, v) by randomly selecting

the u proportional to its degree

2. *edge step*: Add a new edge (u, v), by selecting u and v proportional to their degrees.

Thus at each step the number of nodes increases with probability p and the number of edges always increases. Thus after time t, $e_t = t + 1$ and expected number of nodes is $\mathbf{E}[n_t] = 1 + pt$.

It can be shown that this leads to a scale-free network with $\beta = 2 + \frac{p}{2-p}$. These networks also have the *hierarchicity*, i.e. the local clustering coefficient is proportional inversely proportional to k, the size of the cluster,

$$C(k) \propto k^{-\alpha}$$

where α is called the *hierarchy coefficient*.

These distributions suggest the presence of small dense subgraphs which are connected to each other via "hubs". In other words, there is a lot of clustering, more than random graphs, but at a smaller scale than generic scale free models. Hence these models have a much higher error tolerance, which is similarly exhibited by biological networks.

5.3 Evolution by Duplication

In many scenarios, preferential attachment happens at a larger scale than a single vertex, where multiple parts, aka clusters, of the network get duplicated. This process is hard to model in a one shot setting where we have to duplicate

the parts one at a time. For added robustness, we expand the definition of *preferential attachment* to allow duplicating subgraphs.

We specifically want to model graphs for evolutionary events, such as graphical models and causal networks. Each node represents an event that can take place, while each edge (potentially weighted) represents the *influence* from an event to another. We call this the *event graph* and represent this as a weighted adjacency matrix. An example of such a network that is the Suppes Bayes causal network (SBCN).

We define evolution by duplication as an extension of the preferential attachment model, where we allow extending the graph by duplicating a larger subgraph. This subsumes the original case where preferential attachment of a single node and allows for a larger, more robust evolutionary model.

Definition. 18 (Evolution by Duplication)**.**—Let G be a directed graph, and let $X = [a_i, \ldots, a_j] \times [b_k, \ldots, b_l] \subset [1, \ldots, n]^2$. A **preferential attachment** of X on G, with the weight function θ, is the new digraphon G', defined by

$$G' = (1 - \theta(X)) \cdot G + \theta(X) \cdot G|_X$$

Typically, we want to randomly select segments which are going to be attached. This is carried out with the help of a weight function θ, which is inversely proportional to the measure of the attached segment, $\theta(X) \propto \frac{1}{\int_X G} = \frac{1}{\sum_{\substack{x \in [a_i, \ldots, a_j] \\ y \in [b_k, \ldots, b_l]}} w(a_x, b_y)}$. This weighting implies that smaller segments are easier to attach than larger segments, which mimics the biological characteristics, wherein

it is easier to duplicate smaller, simpler networks over complex multi-path networks.

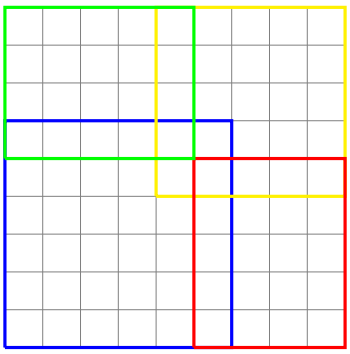

Figure 5.1: Segment basis for a digraph.
The colored rectangles denote *segments* for a digraph, which can be duplicated by *preferential attachment*. It is not necessary for the segments to be non-intersecting, the overlapped regions will be attached multiple times through different segments. Each segment also has an associated weight, which influences the evolutionary priority of duplicating that segment.

Due to the exponentially large size of the segment space, it is preferred to have a smaller finite collection of segments of interest. In biological networks, such as those for cancer somatic mutations, we typically want to restrict our attention to either driver genes or those known to interact with cancer mutations up to some extent. For example, The Cancer Genome Atlas (TCGA) program or the Catalogue of Somatic Mutations in Cancer (COSMIC) databases are the main places which give information about important genes.

Definition. 19 (Segmented Digraph)**.**—A *segmented* digraph, fig. 5.1, is a digraph with a finite collection of weighted segments (X_i, w_i), $X_i = [a_i, b_i] \times [c_i, d_i]$, with weights $w_i \in \mathbb{R}_+, \sum w_i = 1$.

We can also allow dynamic sized vertices by allowing an attachment to introduce a node, thereby adding a row or a column to our adjacency matrix. For

our current analysis, we restrict ourselves to static sized attachments, which only affect weights of the graph.

It is important to note that this is an exchangeable process; relabeling the segments, X_i, does not lead to a different simulation, we only depend upon the weights of the segments. Thus we can leverage all the theoretical properties of digraphons, such as asserting that any such evolutionary process, wherein we do simulations either by attachments to make larger graphs, or by performing weighted boosting, is going to converge to a final digraphon.

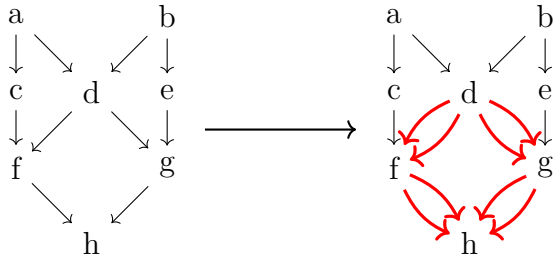

Figure 5.2: Evolution by subgraph attachment via duplication.
Preferential attachment of the segment $[d, f, g, h]^2$. The same effect can be seen for more than one segment attachment, as evidenced by preferentially attaching $[d, f, g] \times [f, g, h]$.

5.4 Finite Modeling and Implementations

To model such an evolutionary process is non trivially complex, as the graphs can have large number of nodes and edges. Naive algorithms for doing *preferential attachment* of a segment $[i, j] \times [k, l]$, have a time complexity of $(j - i) \cdot (k - l) = O(n^2)$. In addition to that, to do a weighted sample, we need to get the current weight of a segment, which would also take $O(n^2)$. This strategy is feasible in cases of small segments, but has an undesirable asymptotic behaviour,

which we improve upon. In this section, we detail an efficient data structure for simulating the preferential attachment framework on a digraph.

There are two key operations that we want our model to access:

1. Get the current weight of a segment X.
2. *Preferentially Attach* a segment, X, with a particular weight, $\theta(X)$.

We note that we can perform (2) in two steps - (a) Multiply the whole graph by $(1 - \theta(X))$, (b) Multiply the segment X by $\frac{1}{1-\theta(x)}$.

Hence, our data structure needs to support the following operations:

1. Multiply a segment $[a, b] \times [c, d]$ by some value c.
2. Return the sum of all elements in a segment $[a, b] \times [c, d]$.

   For a finite model implementation, we assume that the size of the digraph is fixed.

We represent the digraph G using the **square-root decomposition** along the rows and a **segment tree** along the columns for each group of $\sqrt{n}$ rows, fig. 5.3.

Each T_i represents a collection of $\sqrt{n}$ rows with a segment tree. An update of a segment $X = [a, b] \times [c, d]$ can span across segment trees. The square root decomposition facilitates fast updates in $O(\sqrt{n} \log n)$ time.

For each segment tree, it is possible to do *range sum* together with *range multiply* using *lazy propagation* in $O(\log n)$, algorithm 2 and algorithm 3.

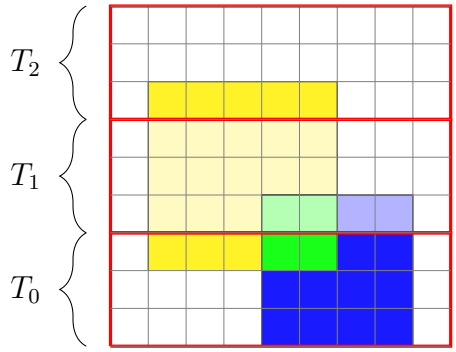

Figure 5.3: Graph data structure for preferential attachment.
We break the graph into $\sqrt{n}$ groups along the rows and create a segment tree, T_i, for each of these groups. We see that any preferential attachment that is done can be broken down into $\sqrt{n}$ parts, where we do updates for each tree, with a total of $\sqrt{n}$ updates. The overlaps are handled efficiently using lazy propagation in the segment tree, which allow us to do the weight updates in $O(\log n)$ for each individual tree.

To prove that the whole data structure works in the time complexity described above, we break the analysis into two parts. First, we show that in an individual segment tree, we can solve range sum and range multiplication in $O(\log n)$. Then we show how to extend this across rows by grouping multiple trees together.

5.4.1 Segment Tree - Lazy Multiply and Sum

The per row simplification of this problem boils down to

Problem. 5.4.1 (Lazy Multiply and Sum)**.** Given an array of numbers, $[a_1, \ldots, a_n]$, perform the following operations in $O(\log n)$

1. Find the sum of all numbers in a contiguous range $[i, j]$.

2. Multiply all numbers in a contiguous range $[i, j]$ by some value k.

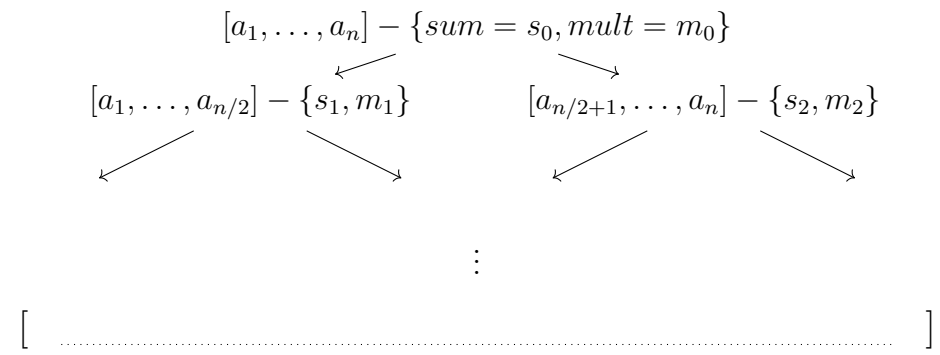

Figure 5.4: Segment tree for lazy propagation.
For each node in the segment tree, we store the subtree range along with two additional parameters, current sum and propagated multiplicand. The multiplicand is not propagated fully until a subrange is updated in a subsequent query or until a subrange weight is queried. And even in those cases, we only propagate it optimally with lazy propagation.

We solve this by creating a segment tree of nodes, fig. 5.4 ,with each node containing the following metadata:

- *low, high* - the lower and upper end points of the range of the node
- *sum* - current sum of all values in the range of the node
- *mult* - current multiplicand not yet propagated to lower nodes
- *left, right* - left and right children nodes of current node

The lazy multiplication algorithm, algorithm 2, works by manipulating the *mult* parameter to keep track of the accumulated multiplication and only propagating it when intersecting ranges are updated.

Theorem. 5.4.1 (Lazy Multiply)**.** The running time of algorithm 2 is $O(\log n)$ for all ranges $[l, r]$.

Algorithm 2 Segment Tree - Lazy Multiply

```
1: function LazyMultiply(Segment Tree Node node, int l, int r, int c)
2:     # Multiplies the part of the range [l, r] contained inside node by c
3:     # Time complexity = O(log n)
4:     if node.low > r or node.high < l then
5:         # No intersection, nothing to do
6:         return
7:     elseif node.low ≥ l and node.high ≤ r then
8:         # Fully contained, multiply current multiplicand by c
9:         node.mult *= c
10:        node.sum *= c
11:    else
12:        LazyMultiply(node.left, node.low, node.high, node.mult)
13:        LazyMultiply(node.right, node.low, node.high, node.mult)
14:        node.mult = 1
15:        LazyMultiply(node.left, l, r, c)
16:        LazyMultiply(node.right, l, r, c)
17:        node.sum = node.left.sum + node.right.sum
18:    end if
19: end function
```

Proof. Notice that the recursion stops at any node at which the range of the node is full contained inside the range to be updated. For example, if our tree is on $[1, \ldots, 8]$, we will have 15 total nodes,

$$[1,8], [1,4], [5,8], [1,2], [3,4], [5,6], [7,8], [1,1], \ldots, [8,8]$$

. If we wish to do a range multiplication on $[4, 7]$, the algorithm will stop at the top most nodes possible which unify up to our desired range, which in this case - $[4,4], [5,6], [7,7]$, which is strictly smaller than 4, the range to be updated.

Notice that we will only ever update a maximum of two nodes of the same length, and they will never be neighbours, as the recursion would instead stop at the parent node. Hence, to represent our range $[l, r]$, as a unification of k ranges - $[l, r] = \cup_{i=1}^{k} [x_i, y_i]$, we can only have $k < 2 \log n$, as the maximum size of a node's subrange is $n/2$.

This argument shows that we will only ever visit $O(\log n)$ nodes, proving our running time proposition. □

We can reach the desired worst case of $\sim 2 \log n$ for a tree of range $[1, 2^n]$ and a range update for $[2, 2^n - 1]$.

Theorem. 5.4.2 (Lazy Sum)**.** The running time of algorithm 3 is $O(\log n)$ all ranges $[l, r]$.

The proof is identical to that of the lazy multiplication, where we only look at the top level nodes.

Algorithm 3 Segment Tree - Lazy Sum

```
1: function LazySum(Segment Tree Node node, int l, int r)
2:     # Returns the sum of the part of the range [l, r] in node
3:     # Time complexity = O(log n)
4:     if node.low > r or node.high < l then
5:         # No intersection
6:         return 0
7:     elseif node.low ≥ l and node.high ≤ r then
8:         # Fully contained
9:         return node.sum
10:    else
11:        LazyMultiply(node.left, node.low, node.high, node.mult)
12:        LazyMultiply(node.right, node.low, node.high, node.mult)
13:        node.mult = 1
14:        return LazySum(node.left, l, r) + LazySum(node.right, l, r)
15:    end if
16: end function
```

5.4.2 Lazy Attach

To utilize the previous algorithms for a 2-D structure, we have to make certain modifications.

We first create a segment tree for each row and then group them together in groups of size $\sqrt{n}$. For each group, create a parent segment tree, with the same metadata, except that for each node, store the **parent.sum** $= \sum_{i \in \textbf{rows}}$ **row.sum**. The parent tree stores the aggregate information across all rows, which allows us to do range queries across the whole group in $O(\log n)$.

It is vital to note that this is only for the whole group and not a subset of the group. To query for a subset of the group, we have to go to each individual row and query its segment tree. Due to the fact that each group is of size $\sqrt{n}$,

we are guaranteed that each operations only touches a maximum of $\sqrt{n}$ groups, out of which only two groups need to ever do individual row updates, as show in algorithm 4.

Algorithm 4 Preferential Attachment on a Digraph

```
1: function PreferentialAttach(Digraph G, Segment X, Weight k)
2:     # Modifies G in place by attaching the 2-D segment X = [a, b] × [c, d]
3:     # Time complexity = O(√n log n)
4:     for T_i ∈ [1, √n]
5:         LazyMultiply(T_i.root, 1, n, 1 − k)
6:     end for
7:     for T_i ∈ [a, b]
8:         # These trees are fully inside the range and can be updated as a
   group
9:         LazyMultiply(T_i.root, c, d, k)
10:    end for
11:    for boundary trees T_a and T_b
12:        # These two trees have partial intersection with the range [a, b] and
   must be updated manually
13:        # Each individual row is also represented as a segment tree
14:        for R_i ∈ T_a, T_b and i ∈ [a, b]
15:            LazyMultiply(R_i.root, c, d, 2)
16:        end for
17:    end for
18: end function
```

Real World Optimizations

There are some factors that can be considered for optimizations.

- If it is known that segment sizes to be updated are within a certain bounded width w, then it is possible to create segment groups of size w instead of $\sqrt{n}$, this implies, that there will only ever be $O(w \log n)$ maximum time.

- For really small width segments, it can also be possible to use data structures such as a Quad Tree or a k-D tree.

- If working with large segments, but a small number of such segments, it is beneficial to look at binary space partition (BSP) trees and pre-partition segments into non intersecting parts. This can quickly become complex, with a growth in number of segments. BSPs are used in computer vision to do segmentation, which allows us to use highly optimized implementations, if such a scenarios is feasible.

5.5 Learning and Inference

Sampling of a digraphon is done by the Chinese Restaurant Process(CRP), which is a staple tool to model the Dirichlet distribution[119,90]. The CRP process aims to model how people are assigned tables when sitting at a shared seating restaurant. With higher probability, people wish to sit next to others for a more pleasant experience, while with a lower probability, they wish to get a new table. This is formalized as follows.

Let $\alpha \in (0, 1)$ be a hyperparameter. At any point of time n, let us have k groups of size $[g_1, \ldots, g_k]$, which are a partition of $[1, \ldots, n]$. At time $n + 1$, we wish to assign a group to the element $n + 1$ which is done as follows:

- With probability $\frac{(1-\alpha)|g_i|}{n+1}$, $n + 1$ is assigned to group i.

- With probability $\frac{\alpha}{n+1}$, $n + 1$ is assigned to a new group g_{k+1}

As we scale $n \to \infty$, we achieve a distribution of the set $\mathbb{N}$, which is the Dirichlet distribution with scaling parameter α. The CRP representation is very useful convenient when performing finite sampling and parameter estimation. Another advantage of the CRP representation is the ease of computation and the fast simulations, which are important for larger models.

5.5.1 Sampling a Digraphon

Similar to Bayesian statistical models, we need a generative model for a digraphon to be able to get insights using parameter estimation. The most common generative models used are the Dirichlet prior, which do leverage using the CRP model.

Let α be the hyper parameter affecting the CRP model to get the clustering assignment of the vertices and let β be another hyper parameter for the standard Dirichlet process, such as the gamma representation.

The generative model for the digraphon is as follows, where we generate a digraph on n vertices:

1. Draw clustering assignments, ζ, for each vertex,

$$\zeta \sim CRP(\alpha)$$

2. Draw weights for the edges, for each pair of groups $r \neq s$

$$\eta_{r,s}|\beta \sim \text{Dirichlet}(\beta)$$

3. Set the edge using the measure of the partition

$$\mathcal{G}_{ij} = \text{Categorical}(\eta_{\zeta_i,\zeta_j})$$

We see that this is an exchangeable process as well, as the clustering assignments are generated irrespective of the actual labels of the vertices. This reasoning implies that as the number of vertices $n \to \infty$, the generated digraph converges to the digraphon.

5.5.2 MAP inference

Let $\mathcal{G}$ be the final digraph generated from the digraphon generative model. Our aim is to infer the weights $\eta_{r,s}$.

The likelihood that $\mathcal{G}$ is sampled is given by

$$\Pr(\mathcal{G}|\zeta) = \prod_{r \neq s} (\zeta_{r,s})^{m_{r,s}}$$

where $m_{r,s}$ denotes the number of edges from cluster r to cluster s and we assume no self loops for simplicity.

$$\Pr(\zeta|\alpha) = \frac{\prod_{r \neq s} (\zeta_{r,s})^{(\alpha-1)}}{\mathbf{B}(\alpha)}$$

where $\mathbf{B}(\omega) = \frac{\prod_i \Gamma(\omega_i)}{\Gamma(\sum_i \omega_i)}$ is the multivariate beta function.

$$\begin{aligned}
\Pr(\mathcal{G}|\alpha) &= \int \Pr(\mathcal{G}|\zeta) \cdot \Pr(\zeta|\alpha) \\
&= \int \frac{\prod_{r\neq s}(\zeta_{r,s})^{m_{r,s}} \prod_{r\neq s}(\zeta_{r,s})^{(\alpha-1)}}{\mathbf{B}(\alpha)} \\
&= \frac{1}{\mathbf{B}(\alpha)} \prod_{r\neq s} \int \zeta_{r,s}^{m_{r,s}+\alpha-1} \, \mathrm{d}\zeta_{r,s} \\
&= \frac{1}{\mathbf{B}(\alpha)} \prod_{r\neq s} \mathbf{B}(m_{r,s}+\alpha)
\end{aligned}$$

We can remove the constant $\frac{1}{\mathbf{B}(\alpha)}$ and maximize the negative log likelihood $\sum_{r\neq s} \log \mathbf{B}(m_{r,s}+\alpha)$ at

$$\zeta_{r,s} = \frac{m_{r,s}+\alpha}{\binom{n}{2} \cdot \alpha + \sum_{r\neq s} m_{r,s}}$$

5.6 Discussion

As final concluding remarks, we have seen how to use digraphons as generative models for directed graphs for evolutionary models. We have further developed a robust modeling data structure for fast simulations which is easily extendable for other evolutionary mechanisms, as the data structure allows for generic operations which can be used for other world models. This opens up further testing grounds for hypothesis checking by comparing simulations to real world dynamics.

There are many extensions that have yet to be explored using the theory of

digraphons, of which an important one is the use of the various distance metrics on the digraphon space. There are many metrics, such as the ones induced by the L_p norm , nuclear norm, and the more interesting cut-distance, which are used. Some of the more interesting questions are presented below.

(1) Can we use the metric on the digraphon space to measure similarity of models for two distinct populations?

An example of such a scenario would be, given data for two distinct populations, we wish to know if they have evolved from a common ancestor population. If so, how far back did they diverge? Can we quantify the divergence of populations? Even more thoroughly, can we find the evolutionary pathway used by the populations to reach the current state?

(2) Do the metrics induce a EM-type convergence for learning algorithms?

Given multi-dimensional data for a population, we wish to stratify it into subpopulations with individuals having a similar evolutionary patterns. This problem is reminiscent of k-means clustering where we wish to use the digraphon metric to perform the clustering. There has been extensive work in the analysis of Euclidean norms for showing convergence (if only to a local minima for the error function), which would be important to translate to the digraphon spaces. The work by Lovász, et al.,[89] has shown many convergence results which may be helpful for such scenarios.

6

Conclusions and Extensions

In this thesis, we established a general pipeline to use of topology and geometry as tools for understanding and analyzing high dimensional point cloud data for evolving populations, using persistence homology and limit digraphons. We further showed a simple learning model for digraphons using Dirichlet processes and gave an efficient implementation to model large evolutionary populations. Our goal was to create an intuitive and reusable pipeline for leveraging rigorous mathematical tools to gain insights into biological mechanisms. As a case study, we implemented these techniques and used them in the context of language evolution and cancer progression.

Our pipelines use the inherent geometric nature of point clouds to analyze relationships between populations clusters. Most current models only analyze similarities between individuals, while we are working at a higher level to understand the relationships between populations. In particular, we also look at the temporal nature of the point cloud data to understand how two different populations are evolving.

The current work in using topological metrics and geometric embeddings is very rudimentary and has a multitude of avenues yet unexplored. We present some of the more important questions below with our current insights and hypothesized steps to tackle the problems.

1. Given the temporal point cloud data of an evolving population, where it consists of multiple (unknown) communities, can we recreate the communities?

   This question naturally arises in terms of cancer progressions models.

Given temporal data for a cancer type, can we find the cancer subtypes?

This is a very important question hitherto left unanswered. Recreating cancer subtypes is currently done using phenotype information and important genetic information is lost in the translation. In chapter 4, we showed how to use transfer learning to great effect to boost information content of 'similar' cancers. This prompts us to theorize that a **k-means** type learning algorithm, where we use the *agony* distance or the *cut* distance for clustering may produce similar results. As an example of adapting the classical *k-means* to the agony distance, we present algorithm 5 which we hope to use for reconstructing the unknown clusters.

Algorithm 5 Probabilistic K-means with Agony

```
function AGONY-K-MEANS(Dataset D, Int k)
    # k is a hyperparameter for number of clusters
    Π^0 = initial partition D into k random sets
    for t = 0 to max iterations (or no change)
        # Similar to centroid calculation, we calculate the SBCN of each
cluster
        G_i^t = SBCN(Π_i^t)  ∀ 1 ≤ i ≤ k
        # Recalculate log-likelihood of each point for all SBCNs
        # and reassign point to cluster with highest LL
        Π^{t+1}(j) = arg max_{1≤i≤k} LL(D_j|G_i^t)  ∀ 1 ≤ j ≤ |D|
    end for
end function
```

2. Can we use the digraphon models to find the divergent points for two populations? Can we do early predictions of events using the Digraphons/SBCNs? This problem arises in terms of finding cancer subtypes, where we wish to have a robust knowledge of different subtypes. We discussed how to

reconstruct subtypes, which themselves are important for personalized medicine. Knowing the full trajectories of cancer types and their divergent branches, informs us about the temporal parameters yet to be taken into account.

3. Can we reconstruct the topology of the fitness landscape based on the paths of the populations? Can we identify key bottlenecks in traversing the landscape?

   Our setting of using an evolutionary population is a rich playground for using ideas related to the mapping of the landscape. The topology of the landscape is dynamic, due to the diverse and ever changing nature of the environment, population individuals, change in both genotypic and phenotypic information. Such problems force us to only do small scale models as full computational simulations are very expensive. Our work on the digraphon model implementation allows us to circumvent a few of these problems, yet many still remain unsolved.

   Some important points worth noting about the landscape which have direct translations to topological terms include the notion of a (a) saddle point, where our populations halt for an extended period of time and (b) evolutionary bottlenecks, where the population has reached a local minima and needs a certain threshold of variance before it starts to re-evolve. Another interesting extension is the tying of pheontypic information to the bottleneck points in the landscape by using the topological information of the evolutionary paths manifested by our sample population.

4. In the context of the Bayesian echo chamber, given a population, is it possible to identify individuals with high 'influence' and 'information flow'? Can we identify 'trending' topics? Knowing such information, how easy is it to manipulate the network to make artificial information flow? Can we identify if there has been artificial tampering with the information network?

   Information flow in social networks is a very important topic due to the prevalence of large online social communities with very low moderation and high participation. Such scenarios lead to the spread of 'fake news' and more grossly incorrect information, which is not vetted by any authoritative third party.

Given many of the open problems above, there is ample space to expand upon the use of these topological and geometric tools to further the study of evolutionary models. Topological and geometric tools are still new to the field of machine learning, with manifold learning being the closest technique in similarity. The tools developed in this thesis venture to show a general and usable pipeline robust enough for most applications and fast enough for large scale simulations. Our belief is that these tools will help us get a more human readable information from these models and develop an explainable AI as opposed to the multitude of black box learning which is currently prevalent.

Appendix A - Persistence Homology

In this appendix we detail the mathematial foundations and algorithms for calculating persistence homology.

Persistence

First we start with algebraically defining a *simplex*.

Definition. 20.—Given a set $T = \{a_0, \cdots, a_k\}$ of *affinely independent* points in $\mathbb{R}^n$, a k-**simplex**, denoted by σ_T, is the convex-hull of T. The simplices σ_U, $U \subset T$ are called the **faces** of T. k is called the **dimension** of σ_T.

We first define an orientation of[95] a simplex $\sigma = \{a_0, \cdots, a_p\}$ is the equivalence class of permutations which have the same sign.

Given a simplicial complex K, let C_p be the free group generated by the set of oriented p-simplices of K, with $[\alpha] = -[\beta]$ if α and β are the same simplex with opposite orientations.

Hence $C_p = \left\{ \sum_i n_i \alpha_i \mid \alpha_i \in K,\ dim(\alpha_i) = p \right\}$, is a free group.

We have a homomorphism $\delta_p : C_p \to C_{p-1}$ which is defined on the generators as

$$\delta([a_0, \cdots, a_p]) = \sum_{i=0}^{p} (-1)^i [a_0, \cdots, a_{i-1}, a_{i+1}, \cdots, a_p]$$

This is called the boundary operator as it sends each simplex to a sum of its faces.

It has the property that $\delta_{p-1}\delta_p = 0$.

We define subgroups of C_p which will be used to define the homology groups. Let $Z_p = Ker(\delta_p)$ be the p-cycle group and $B_p = Im(\delta_{p+1})$ be the p-boundary group.

Due to the previous observation we have that $B_p \subseteq Z_p \subseteq C_p$.

We define the p-homology group as $H_p = Z_p/B_p$. We can change the base ring of computations R (which as of now is $\mathbb{Z}$) to define C_p as a module over this ring instead of a free group. This allows us to define the homology groups over non-trivial rings.

If the base ring is a PID then the structure theorem of modules over PID (a generalization of the structure theorem for finite abelian groups) gives us the composition of the module.

Over a field of the kind $\mathbb{R}, \mathbb{C}, \mathbb{Z}_p$, this module does not have a torsion component and is instead a vector space, which is fully described by its rank, β_p, which is called the p-th betti number of K.

We can calculate the homology of a simplicial complex by representing the boundary operator as a matrix and calculating the rank of the operator.

Represent δ_p in the standard bases of C_p, C_{p-1} as a matrix M_p of order $r_{p-1} \times$

r_p. The null space of M_p is Z_p and the image space is B_{p-1}. If we can calculate rank of Z_p and B_{p-1} for all p, we have $\beta_p = rank(Z_p) - rank(B_p)$.

Using elementary row(column) operations of the form

- exchange row(column) i and j
- multiply row(column) i by -1
- replace row(column) i by $i + q \cdot j$(column) $(i \neq j)$

we can reduce M_p to its Smith normal form

$$\left[\begin{array}{c|c} \begin{array}{ccc} a_1 & & \\ & \ddots & \\ & & a_{l_p} \end{array} & \begin{array}{c} \\ 0 \\ \\ \end{array} \\ \hline \begin{array}{ccc} & 0 & \end{array} & \begin{array}{c} 0 \end{array} \end{array}\right]$$

where $a_i \geq 1$, $a_i | a_{i+1}$.

Here we have that $\text{rank}(M_p) = l_p = \text{rank}(B_{p-1})$ and $\text{rank}(Z_p) = r_p - l_p$. Hence $\beta_p = r_p - l_p - l_{p+1}$. This reduction can be done in $O(n^3)$ time (n p-simplices in K), hence we can find all the homologies of a simplicial complex in $O(m^3)$ where we have m simplices in K (a gross over estimate).

Definition. 21.—Given a set $T = \{a_0, \cdots, a_k\}$ in $\mathbb{R}[n]$ the ϵ-**Vietoris-Rips** complex of T, denoted by $VR_\epsilon(T)$ is defined as

$$\{U \subset T \mid d(a_i, a_j) \leq \epsilon \ \forall \ a_i, a_j \in U\}$$

We see that if $\epsilon < \epsilon'$ then $VR_\epsilon(T) \subset VR_{\epsilon'}(T)$. This gives us a filtration with varying values of ϵ. Also note that at a large value of ϵ the complex does not change as it is consists of all possible subsets of T.

Definition. 22.—Given an open cover of $S, U = \{U_i\}_{i \in I}$, the **nerve** of U denoted by N, is given by

1. $\phi \in N$.
2. if $\cap_{j \in J} U_J \neq \phi$, then $J \in N$.

A cover is called good if all the sets $\cap_{j \in J} U_j$, where J is finite, are contractible.

Lemma. 6.0.1. The underlying space of the nerve of a good cover is homotopy equivalent to the union of the sets in the cover.

Definition. 23.—A **filtered simplicial complex** is a chain of subcomplexes

$$\phi = K_0 \subset K_1 \subset \cdots \subset K_m = K$$

Given a filtered simplicial complex we have a chain of homology groups

$$0 = H_p(K_0) \to H_p(K_1) \to \cdots \to H_p(K_n) = H_p(K)$$

for each dimension p, induced by the inclusion from $K_i \to K_j$ for $i \leq j$.

Call the maps $f_p^{i,j} : H_p(K_i) \to H_p(K_j)$.

Definition. 24.—The **p-persistent homology groups** $H_p^{i,j} = Im(f_p^{i,j})$, the images of the homomorphisms. The **p-persistent** Betti numbers are the ranks of the corresponding homology groups, $\beta_p^{i,j} = Rank(H_p^{i,j})$.

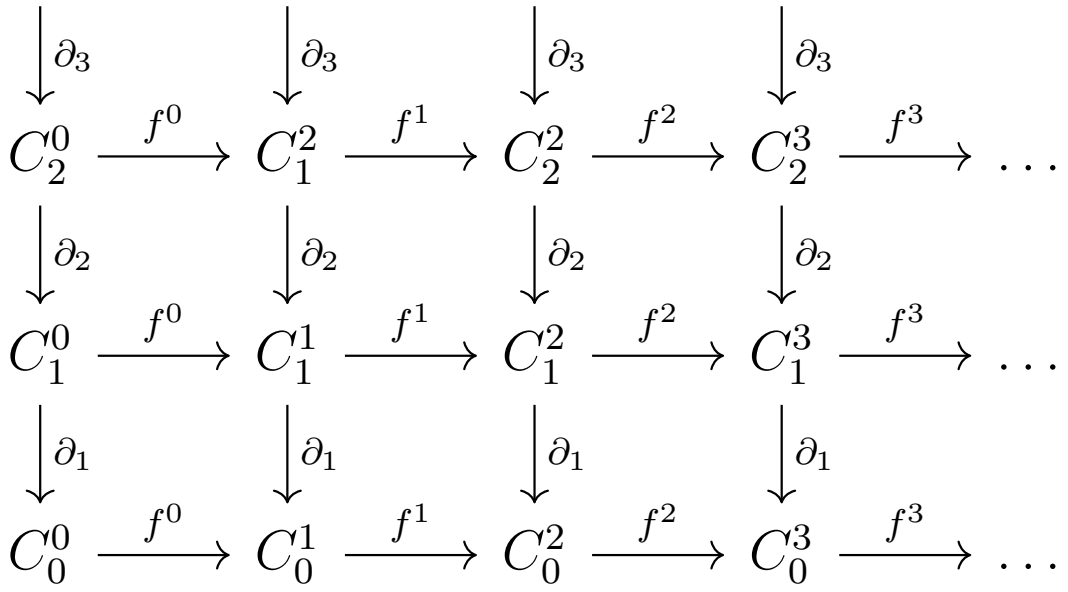

Figure 6.1: C_d^k represents the d-dimensional chain complex of the k'th persistence module in our filtration. In the iterative growth setting, each column represents the state of the current simplicial complex at a designated radius. As we are growing our complex, one step at a time, each simplicial complex is contained inside the one succeeding it, giving rise to the natural inclusion maps, which we can use in lieu of the morphisms f^k.

Definition. 25.—A persistence complex $\mathcal{C}$ is a chain of complexes $\{\mathrm{C}_*^i\}_{i\geq 0}$ over a ring R, e.g. fig. 6.1,

$$C_*^0 \xrightarrow{f^0} C_*^1 \xrightarrow{f^1} \cdots$$

Definition. 26.—A persistence module is a chain of R modules

$$M^0 \xrightarrow{\phi^0} M^1 \xrightarrow{\phi^1} \cdots$$

Definition. 27.—A persistence complex (module) is of finite type if each component complex (module) is a finitely generated R module and the maps f^i (ϕ^i) are isomorphisms for $i \geq m$ for some m.

Definition. 28.—A $\mathcal{P}$-interval is an ordered pair $(i, j), 0 \leq i < j \in \mathbb{Z}\infty = \mathbb{Z} \cup \{\infty\}$

Lemma. 6.0.2. There is an equivalence of categories between the category of finitely generated non-negatively graded $R[t]$ modules and the category of persistence modules of finite type over R.

Lemma. 6.0.3. There is an equivalence of categories between the category of finite sets of $\mathcal{P}$-intervals and the category of finitely generated non-negatively graded $R[t]$ modules.

These representation allows us to translate the homology groups from modules over $R[t]$ to $\mathcal{P}$-intervals. Hence we can restructure our problem to finding the $\mathcal{P}$-intervals of a given persistence complex.

We shall use the following simplicial complex as an example and do the computations over $\mathbb{Z}_2$.

Let α_i, β_i be the standard bases for C_k and C_{k-1} We assign degrees to each simplex in the complex which represent the time at which this simplex came into the complex. The persistence complex we are dealing with has the following degrees

a	b	c	d	ab	bc	ac	bd	ad	abc	abd
0	0	1	1	1	2	2	2	2	3	4

The representation of the boundary matrix of δ_k, say M_k, is characterized by having

$$deg(\beta_i) + deg(M_k[i, j]) = deg(\alpha_j)$$

Here we have

$$M_1 = \left[\begin{array}{c|ccccc} & ab & bc & ac & bd & ad \\ \hline d & 0 & 0 & 0 & t & t \\ c & 0 & t & t & 0 & 0 \\ b & t & t^2 & 0 & t^2 & 0 \\ a & t & 0 & t^2 & 0 & t^2 \end{array}\right] \tag{6.1}$$

To calculate the homology of the persistence complex we need to first represent the boundary matrix relative to the bases of C_k and Z_{k-1} and reducing the matrix to a column echelon form. We find the representation of the matrix inductively as follows. For the base case we have the standard representation of M_1, as shown above. Now suppose we have a representation M_k of δ_k relative to the standard basis $\{\alpha_i\}$ of C_k and a homogeneous basis $\{\beta_i\}$ of Z_{k-1}. What we want is a homogeneous basis of Z_k and to represent δ_{k+1} relative to this basis.

First we sort β_i in reverse order of degree, as above. Then reduce M_k to column echelon form $\tilde{M}_k$, as shown.

$$\left[\begin{array}{cccc|ccc} \lambda_{11} & 0 & & & \cdots & 0 \\ & \lambda_{22} & 0 & & \cdots & 0 \\ \vdots & \vdots & \vdots & & & \vdots \\ & & \lambda_{ij} & 0 & \cdots & 0 \\ & & \vdots & & & 0 \end{array}\right]$$

Every λ_{ij} represents a **pivot** and the corresponding row(column) is a pivot row(column). The basis elements for non-pivot columns are the desired basis

of Z_k, in our case after reducing to column echelon form we have that

$$\tilde{M}_1 = \left[\begin{array}{c|ccccc} & bd & bc & ab & z_1 & z_2 \\ \hline d & t & 0 & 0 & 0 & 0 \\ c & 0 & t & 0 & 0 & 0 \\ b & t^2 & t^2 & t & 0 & 0 \\ a & 0 & 0 & t & 0 & 0 \end{array}\right] \tag{6.2}$$

where $z_1 = ac - bc - t \cdot ab$ and $z_2 = ad - bd - t \cdot ab$.

We can get the column echelon form by using elementary column operations similar to the ones in Gaussian elimination.

Lemma. 6.0.4. The pivots in the column echelon form are the same as the diagonal elements in the normal form. Moreover, the degree of the basis elements on pivot rows is the same in both forms.

What we want from this is the following corollary which tells us how to get the $\mathcal{P}$-intervals from the description of the matrix.

Corollary..—Let $\tilde{M}_k$ be the column echelon form for δ_k relative to the bases α_i of C_k and β_i of Z_{k-1}. If row i has pivot t^n, then it contributes a $\mathcal{P}$-interval $(deg(\beta_i), deg(\beta_i) + n)$, else it gives $(deg(\beta_i), \infty)$, in the description of H_{k-1}.

In our case we have $\tilde{M}_1[2,2] = t$ the element contributes a $\mathcal{P}$-interval $(1,2)$ to H_0.

So to represent δ_{k+1} in terms of the homogeneous basis that we have for Z_k we first make a few observations. We have that $M_k M_{k+1} = 0$ as $\delta_k \delta_{k+1} = 0$.

As we have only performed column operations on M_k to get $\tilde{M}_k$ we have that $\tilde{M}_k M_{k+1} = 0$. We see that we can get the basis representation by removing the rows of M_{k+1} which correspond to non-zero pivot columns in $\tilde{M}_k$.

Lemma. 6.0.5. To represent δ_{k+1} relative to the bases of C_{k+1} and Z_k, we delete the rows corresponding to non-zero pivot columns in $\tilde{M}_k$.

The second boundary map, for δ_2 is

$$M_2 = \left[\begin{array}{c|cc} & abc & abd \\ \hline bc & t & 0 \\ ad & 0 & t^2 \\ ac & t & 0 \\ bd & 0 & t^2 \\ ab & t^2 & t^3 \end{array}\right] \tag{6.3}$$

To represent it in terms of Z_2 we need to remove the last three rows to get

$$\tilde{M}_2 = \left[\begin{array}{c|cc} & abc & abd \\ \hline z_1 & t & 0 \\ z_2 & 0 & t^2 \end{array}\right] \tag{6.4}$$

Which is the representation that we need.

References

[1] Deepdpm: Dynamic population mapping via deep neural network. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01).

[2] Ashley Acevedo, Leonid Brodsky, and Raul Andino. Mutational and fitness landscapes of an RNA virus revealed through population sequencing. *Nature*, 505(7485):686, 2014.

[3] Nuraini Aguse, Yuanyuan Qi, and Mohammed El-Kebir. Summarizing the solution space in tumor phylogeny inference by multiple consensus trees. *Bioinformatics*, 35(14):i408–i416, 07 2019.

[4] Edo M Airoldi, Thiago B Costa, and Stanley H Chan. Stochastic blockmodel approximation of a graphon: Theory and consistent estimation. In *Advances in Neural Information Processing Systems*, pages 692–700, 2013.

[5] Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. A latent variable model approach to PMI-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399, 2016.

[6] Sanjeev Arora, László Lovász, Ilan Newman, Yuval Rabani, Yuri Rabinovich, and Santosh Vempala. Local versus global properties of metric spaces. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 41–50. Society for Industrial and Applied Mathematics, 2006.

[7] Stuart G Baker. Identifying combinations of cancer markers for further study as triggers of early intervention. *Biometrics*, 56(4):1082–1087, 2000.

[8] Jeffrey S Banks et al. *Signaling games in political science*, volume 46. Psychology Press, 1991.

[9] Albert-László Barabási. Scale-free networks: a decade and beyond. *science*, 325(5939):412–413, 2009.

[10] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[11] Albert-László Barabási and Eric Bonabeau. Scale-free networks. *Scientific american*, 288(5):60–69, 2003.

[12] Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger. Research-paper recommender systems: A literature survey. *International Journal on Digital Libraries*, pages 1–34, 07 2015.

[13] Niko Beerenwinkel, Nicholas Eriksson, Bernd Sturmfels, et al. Conjunctive bayesian networks. *Bernoulli*, 13(4):893–909, 2007.

[14] Robert M. Bell, Yehuda Koren, and Chris Volinsky. The bellkor solution to the netflix prize. 2007.

[15] Emily Bienvenue. Computational propaganda: political parties, politicians, and political manipulation on social media. *International Affairs*, 96(2):525–527, 2020.

[16] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez. Recommender systems survey. *Knowledge-Based Systems*, 46:109–132, 2013.

[17] Rob J de Boer. Modeling population dynamics: A graphical approach, 2012.

[18] Francesco Bonchi, Sara Hajian, Bud Mishra, and Daniele Ramazzotti. Exposing the probabilistic causal structure of discrimination. *International Journal of Data Science and Analytics*, 3(1):1–21, 2017.

[19] Christian Borgs, Jennifer Chayes, László Lovász, Vera T Sós, Balázs Szegedy, and Katalin Vesztergombi. Graph limits and parameter testing. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 261–270, 2006.

[20] Christian Borgs, Jennifer T Chayes, László Lovász, Vera T Sós, and Katalin Vesztergombi. Convergent sequences of dense graphs i: Subgraph frequencies, metric properties and testing. *Advances in Mathematics*, 219(6):1801–1851, 2008.

[21] Christian Borgs, Jennifer T Chayes, László Lovász, Vera T Sós, and Katalin Vesztergombi. Convergent sequences of dense graphs ii. multiway cuts and statistical physics. *Annals of Mathematics*, pages 151–219, 2012.

[22] S Bradshaw and P Howard. Troops, trolls and troublemakers: A global inventory of organized social media manipulation. 2017.

[23] Åke Brännström and David JT Sumpter. The role of competition and clustering in population dynamics. *Proceedings of the Royal Society B: Biological Sciences*, 272(1576):2065–2072, 2005.

[24] Ted Briscoe. *Linguistic evolution through language acquisition*. Cambridge University Press, 2002.

[25] Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

[26] Gunnar Carlsson. Topological pattern recognition for point cloud data. *Acta Numerica*, 23:289–368, 2014.

[27] John Case and Samuel E Moelius. Optimal language learning. In *International Conference on Algorithmic Learning Theory*, pages 419–433. Springer, 2008.

[28] Chibiao Chen, Eric Durand, Florence Forbes, and Olivier François. Bayesian clustering algorithms ascertaining spatial population structure: a new computer program and a comparison study. *Molecular Ecology Notes*, 7(5):747–756, 2007.

[29] Carol Chomsky. Stages in language development and reading exposure. *Harvard Educational Review*, 42(1):1–33, 1972.

[30] Morten H Christiansen and Nick Chater. Toward a connectionist model of recursion in human linguistic performance. *Cognitive Science*, 23(2):157–205, 1999.

[31] Dan C Cireşan, Ueli Meier, and Jürgen Schmidhuber. Transfer learning for latin and chinese characters with deep neural networks. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6. IEEE, 2012.

[32] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete and Computational Geometry*, 37(1):103–120, 2007.

[33] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

[34] Vivian Cook and Mark Newson. *Chomsky's universal grammar*. John Wiley & Sons, 2014.

[35] David R Cox. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202, 1972.

[36] James F. Crow. *Basic concepts in population, quantitative, and evolutionary genetics*. 1986.

[37] Luca De Sano, Giulio Caravagna, Daniele Ramazzotti, Alex Graudenzi, Giancarlo Mauri, Bud Mishra, and Marco Antoniotti. Tronco: an r package for the inference of cancer progression models from heterogeneous genomic data. *Bioinformatics*, 32(12):1911–1913, 2016.

[38] Vin De Silva and Robert Ghrist. Coverage in sensor networks via persistent homology. *Algebraic & Geometric Topology*, 7(1):339–358, 2007.

[39] Walter Dempsey, Brandon Oselio, and Alfred Hero. Hierarchical network models for exchangeable structured interaction processes. *Journal of the American Statistical Association*, pages 1–43, 2021.

[40] Persi Diaconis and Svante Janson. Graph limits and exchangeable random graphs. *Rendiconti di Matematica e delle sue Applicazioni. Serie VII*, pages 33–61, 2008.

[41] Ramon Diaz-Uriarte and Claudia Vasallo. Every which way? on predicting tumor evolution using cancer progression models. *BioRxiv*, page 371039, 2019.

[42] Irit Dinur and Samuel Safra. On the hardness of approximating minimum vertex cover. *Annals of mathematics*, pages 439–485, 2005.

[43] Herbert Edelsbrunner and John Harer. Persistent homology-a survey. *Contemporary mathematics*, 453:257–282, 2008.

[44] Frank Eisner and James M McQueen. Perceptual learning in speech: Stability over time. *The Journal of the Acoustical Society of America*, 119(4):1950–1953, 2006.

[45] Justin Eldridge, Mikhail Belkin, and Yusu Wang. Graphons, mergeons, and so on! In *Advances in Neural Information Processing Systems*, pages 2307–2315, 2016.

[46] Douglas H Erwin and Eric H Davidson. The evolution of hierarchical gene regulatory networks. *Nature Reviews Genetics*, 10(2):141–148, 2009.

[47] Paola Escudero. *Linguistic perception and second language acquisition: Explaining the attainment of optimal phonological categorization*. Netherlands Graduate School of Linguistics, 2005.

[48] Ludger Evers and Claudia-Martina Messow. Sparse kernel methods for high-dimensional survival data. *Bioinformatics*, 24(14):1632–1638, 2008.

[49] Jean Fan, Kamil Slowikowski, and Fan Zhang. Single-cell transcriptomics in cancer: computational challenges and opportunities. *Experimental and Molecular Medicine*, 52(9):1452–1465, 2020.

[50] Hossein Shahrabi Farahani and Jens Lagergren. Learning oncogenetic networks by reducing to mixed integer linear programming. *PloS one*, 8(6):e65773, 2013.

[51] John L Fischer. Social influences on the choice of a linguistic variant. *Word*, 14(1):47–56, 1958.

[52] Steven A Frank and Martin A Nowak. Problems of somatic mutation and cancer. *Bioessays*, 26(3):291–299, 2004.

[53] Alan M. Frieze and Ravi Kannan. Quick approximation to matrices and applications. *Combinatorica*, 19(2):175–220, 1999.

[54] Robert A Gatenby and Thomas L Vincent. An evolutionary model of carcinogenesis. *Cancer research*, 63(19):6212–6220, 2003.

[55] Luca Gattinoni, Daniel J Powell, Steven A Rosenberg, and Nicholas P Restifo. Adoptive immunotherapy for cancer: building on success. *Nature Reviews Immunology*, 6(5):383–393, 2006.

[56] Amit Goyal, Francesco Bonchi, and Laks VS Lakshmanan. Learning influence probabilities in social networks. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 241–250, 2010.

[57] Joseph H Greenberg. A quantitative approach to the morphological typology of language. *International journal of American linguistics*, 26(3):178–194, 1960.

[58] Marco Grzegorczyk. Extracting protein regulatory networks with graphical models. *Proteomics*, 7(S1):51–59, 2007.

[59] Anna Guimaraes, Oana Balalau, Erisa Terolli, and Gerhard Weikum. Analyzing the traits and anomalies of political discussions on reddit. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 13, pages 205–213, 2019.

[60] Fangjian Guo, Charles Blundell, Hanna Wallach, and Katherine Heller. The bayesian echo chamber: Modeling social influence via linguistic accommodation. In *Artificial Intelligence and Statistics*, pages 315–323, 2015.

[61] Douglas Hanahan and Robert A Weinberg. Hallmarks of cancer: the next generation. *cell*, 144(5):646–674, 2011.

[62] Frank E Harrell, Robert M Califf, David B Pryor, Kerry L Lee, and Robert A Rosati. Evaluating the yield of medical tests. *Jama*, 247(18):2543–2546, 1982.

[63] Allen Hatcher. *Algebraic topology*. 2001.

[64] Xiaoling Hu, Li Fuxin, Dimitris Samaras, and Chao Chen. Topology-preserving deep image segmentation. *arXiv preprint arXiv:1906.05404*, 2019.

[65] Xuedong Huang, Alex Acero, Hsiao-Wuen Hon, and Raj Reddy. *Spoken language processing: A guide to theory, algorithm, and system development*, volume 1. Prentice hall PTR Upper Saddle River, 2001.

[66] Dell H Hymes. *Pidginization and creolization of languages*. CUP Archive, 1971.

[67] Katharina Jahn, Jack Kuipers, and Niko Beerenwinkel. Tree inference for single-cell data. *Genome biology*, 17(1):86, 2016.

[68] David Juan, Florencio Pazos, and Alfonso Valencia. Co-evolution and co-adaptation in protein networks. *FEBS letters*, 582(8):1225–1230, 2008.

[69] Anuraag R Kansal, S Torquato, GR Harsh, EA Chiocca, and TS Deisboeck. Simulated brain tumor growth dynamics using a three-dimensional cellular automaton. *Journal of theoretical biology*, 203(4):367–382, 2000.

[70] Nikolai Karpov, Salem Malikic, Md Rahman, S Cenk Sahinalp, et al. A multi-labeled tree edit distance for comparing" clonal trees" of tumor progression. In *18th International Workshop on Algorithms in Bioinformatics (WABI 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[71] Hagen Kennecke, Rinat Yerushalmi, Ryan Woods, Maggie Chon U Cheang, David Voduc, Caroline H Speers, Torsten O Nielsen, and Karen Gelmon. Metastatic behavior of breast cancer subtypes. *Journal of clinical oncology*, 28(20):3271–3277, 2010.

[72] Chiheon Kim, Afonso S Bandeira, and Michel X Goemans. Community detection in hypergraphs, spiked tensor models, and sum-of-squares. In *Sampling Theory and Applications (SampTA), 2017 International Conference on*, pages 124–128. IEEE, 2017.

[73] Werner Kirsch. An elementary proof of de finetti's theorem. *Statistics and Probability Letters*, 151:84–88, 2019.

[74] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[75] Natalia L Komarova, Erin Urwin, and Dominik Wodarz. Accelerated crossing of fitness valleys through division of labor and cheating in asexual populations. *Scientific reports*, 2:917, 2012.

[76] Eugene V Koonin and Artem S Novozhilov. Origin and evolution of the genetic code: the universal enigma. *IUBMB life*, 61(2):99–111, 2009.

[77] Kirill S Korolev, Joao B Xavier, and Jeff Gore. Turning ecology and evolution against cancer. *Nature Reviews Cancer*, 14(5):371–380, 2014.

[78] Travis LaCroix. Evolutionary explanations of simple communication: Signalling games and their models. *Journal for General Philosophy of Science*, 51(1):19–43, 2020.

[79] Jerald F Lawless. *Statistical models and methods for lifetime data*, volume 362. John Wiley & Sons, 2011.

[80] Carol Eunmi Lee. Evolutionary genetics of invasive species. *Trends in Ecology and Evolution*, 17(8):386–391, 2002.

[81] Eric H Lenneberg. The biological foundations of language. *Hospital Practice*, 2(12):59–67, 1967.

[82] Simon A Levin. Population dynamic models in heterogeneous environments. *Annual review of ecology and systematics*, pages 287–310, 1976.

[83] David CS Li. Li wei, three generations, two languages, one family: Language choice and language shift in a chinese community in britain.(multilingual matters, 104.) clevedon (uk) & philadelphia (pa): Multilingual matters, 1994. pp. viii, 221. hb£ 49.00, 99.00; *pb*£16.95,34.95. *Language in Society*, 25(1):147–151, 1996.

[84] G. Linden, B. Smith, and J. York. Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, 2003.

[85] Wei Liu, Matteo Pellegrini, and Xiaofan Wang. Detecting communities based on network topology. *Scientific reports*, 4(1):1–7, 2014.

[86] John C Loehlin. *Latent variable models*. hillsdale, nj: erlbaum, 1987.

[87] László Lovász. *Large networks and graph limits*, volume 60. American Mathematical Soc., 2012.

[88] László Lovász and Balázs Szegedy. Limits of dense graph sequences. *Journal of Combinatorial Theory, Series B*, 96(6):933–957, 2006.

[89] László Lovász and Katalin Vesztergombi. Nondeterministic graph property testing. *arXiv preprint arXiv:1202.5337*, 2012.

[90] Steven N MacEachern and Peter Müller. Estimating mixture of dirichlet process models. *Journal of Computational and Graphical Statistics*, 7(2):223–238, 1998.

[91] Winter A Mason, Frederica R Conrey, and Eliot R Smith. Situating social influence processes: Dynamic, multidirectional flows of influence within social networks. *Personality and social psychology review*, 11(3):279–300, 2007.

[92] Risto Miikkulainen, Jason Liang, Elliot Meyerson, Aditya Rawal, Daniel Fink, Olivier Francon, Bala Raju, Hormoz Shahrzad, Arshak Navruzyan Nigel Duffy, and Babak Hodjat. Chapter 15 - evolving deep neural networks. In Robert Kozma, Cesare Alippi, Yoonsuck Choe, and Francesco Carlo Morabito, editors, *Artificial Intelligence in the Age of Neural Networks and Brain Computing*, pages 293–312. Academic Press, 2019.

[93] Risto Miikkulainen, Jason Liang, Elliot Meyerson, Aditya Rawal, Daniel Fink, Olivier Francon, Bala Raju, Hormoz Shahrzad, Arshak Navruzyan, Nigel Duffy, et al. Evolving deep neural networks. In *Artificial intelligence in the age of neural networks and brain computing*, pages 293–312. Elsevier, 2019.

[94] Rupert G Miller Jr. *Survival analysis*, volume 66. John Wiley & Sons, 2011.

[95] Nikola Milosavljević, Dmitriy Morozov, and Primoz Skraba. Zigzag persistent homology in matrix multiplication time. In *Proceedings of the twenty-seventh annual symposium on Computational geometry*, pages 216–225. ACM, 2011.

[96] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017.

[97] Johanna Nichols. *Linguistic diversity in space and time*. University of Chicago Press, 1992.

[98] Martin A. Nowak, Natalia L. Komarova, and Partha Niyogi. Computational and evolutionary aspects of language. *Nature*, 417(6889):611–617, 2002.

[99] Susumu Ohno. *Evolution by gene duplication*. Springer Science & Business Media, 2013.

[100] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[101] Jeffrey Pawlick, Edward Colbert, and Quanyan Zhu. Modeling and analysis of leaky deception using signaling games with evidence. *IEEE Transactions on Information Forensics and Security*, 14(7):1871–1886, 2018.

[102] Judea Pearl. *Causality: models, reasoning and inference*, volume 29. Springer, 2000.

[103] Francois Petitjean, Germain Forestier, Geoffrey I. Webb, Ann E. Nicholson, Yanping Chen, and Eamonn Keogh. Dynamic time warping averaging of time series allows faster and more accurate classification. In *2014 IEEE International Conference on Data Mining*, pages 470–479, 2014.

[104] Sudeep Pushpakom, Francesco Iorio, Patrick A Eyers, K Jane Escott, Shirley Hopper, Andrew Wells, Andrew Doig, Tim Guilliams, Joanna Latimer, Christine McNamee, et al. Drug repurposing: progress, challenges and recommendations. *Nature reviews Drug discovery*, 18(1):41–58, 2019.

[105] Daniele Ramazzotti, Giulio Caravagna, Loes Olde Loohuis, Alex Graudenzi, Ilya Korsunsky, Giancarlo Mauri, Marco Antoniotti, and Bud Mishra. Capri: efficient inference of cancer progression models from cross-sectional data. *Bioinformatics*, 31(18):3016–3026, 2015.

[106] Daniele Ramazzotti, Alex Graudenzi, Giulio Caravagna, and Marco Antoniotti. Modeling cumulative biological phenomena with suppes-bayes causal networks. *Evolutionary Bioinformatics*, 14:1176934318785167, 2018.

[107] Chhavi Rana and Sanjay Kumar Jain. A study of the dynamic features of recommender systems. *Artificial Intelligence Review*, 43(1):141–153, 2015.

[108] Garry Robins and Philippa Pattison. Random graph models for temporal processes in social networks. *Journal of Mathematical Sociology*, 25(1):5–41, 2001.

[109] Inés Santé, Andrés M. García, David Miranda, and Rafael Crecente. Cellular automata models for the simulation of real-world urban processes: A review and analysis. *Landscape and Urban Planning*, 96(2):108–122, 2010.

[110] Paul M Sharp and Giorgio Matassi. Codon usage and genome evolution. *Current opinion in genetics & development*, 4(6):851–860, 1994.

[111] Jeff Siegel. Mixing, leveling, and pidgin/creole development. *The structure and status of pidgins and creoles*, pages 111–149, 1997.

[112] Gurjeet Singh, Facundo Mémoli, and Gunnar E Carlsson. Topological methods for the analysis of high dimensional data sets and 3d object recognition. In *SPBG*, pages 91–100, 2007.

[113] Primoz Skraba and Katharine Turner. Wasserstein stability for persistence diagrams. *arXiv preprint arXiv:2006.16824*, 2020.

[114] Brent Smith and Greg Linden. Two decades of recommender systems at amazon.com. *IEEE Internet Computing*, 21(3):12–18, 2017.

[115] Joel Sobel. Signaling games. *Complex Social and Behavioral Systems: Game Theory and Agent-Based Models*, pages 251–268, 2020.

[116] Patrick Suppes. A probabilistic theory of causality. 1973.

[117] Nikolaj Tatti. Faster way to agony. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 163–178. Springer, 2014.

[118] Nikolaj Tatti. Tiers for peers: a practical algorithm for discovering hierarchy in weighted networks. *Data mining and knowledge discovery*, 31(3):702–738, 2017.

[119] Yee Whye Teh. Dirichlet process., 2010.

[120] Michael Tiefelsdorf. *Modelling spatial processes: the identification and analysis of spatial relationships in regression residuals by means of Moran's I*, volume 87. Springer, 2006.

[121] Arne Traulsen, Christoph Hauert, Hannelore De Silva, Martin A Nowak, and Karl Sigmund. Exploration dynamics in evolutionary games. *Proceedings of the National Academy of Sciences*, 106(3):709–712, 2009.

[122] Peter Turchin. *Complex Population Dynamics*. 2003.

[123] Suneetha Uppu and Aneesh Krishna. Tuning hyperparameters for gene interaction models in genome-wide association studies. In *International Conference on Neural Information Processing*, pages 791–801. Springer, 2017.

[124] Raoul R Wadhwa, Drew FK Williamson, Andrew Dhawan, and Jacob G Scott. Tdastats: R pipeline for computing persistent homology in topological data analysis. *Journal of open source software*, 3(28):860, 2018.

[125] Clifford H Wagner. Simpson's paradox in real life. *The American Statistician*, 36(1):46–48, 1982.

[126] Larry Wasserman. Topological data analysis, 2016.

[127] Martin Wattenberg, Fernanda Viégas, and Ian Johnson. How to use t-sne effectively. *Distill*, 1(10):e2, 2016.

[128] Duncan J Watts. Networks, dynamics, and the small-world phenomenon. *American Journal of sociology*, 105(2):493–527, 1999.

[129] Adriano V Werhli, Marco Grzegorczyk, and Dirk Husmeier. Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical gaussian models and bayesian networks. *Bioinformatics*, 22(20):2523–2531, 2006.

[130] Achmad Widodo and Bo-Suk Yang. Machine health prognostics using survival probability and support vector machine. *Expert Systems with Applications*, 38(7):8430–8437, 2011.

[131] Dominik Wodarz and Natalia L. Komarova. *Dynamics of Cancer: Mathematical Foundations of Oncology*. World Scientific Publishing Co., Inc., USA, 1st edition, 2014.

[132] Liang Wu, Fred Morstatter, Kathleen M. Carley, and Huan Liu. Misinformation in social media: Definition, manipulation, and detection. *Sigkdd Explorations*, 21(2):80–90, 2019.

[133] Safoora Yousefi, Fatemeh Amrollahi, Mohamed Amgad, Chengliang Dong, Joshua E Lewis, Congzheng Song, David A Gutman, Sameer H Halani, Jose Enrique Velazquez Vega, Daniel J Brat, et al. Predicting clinical outcomes from large scale cancer genomic profiles with deep survival models. *Scientific reports*, 7(1):11707, 2017.

[134] Haitao Zhao and Zhong-Hui Duan. Cancer genetic network inference using gaussian graphical models. *Bioinformatics and biology insights*, 13:1177932219839402, 2019.

[135] Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.