UNIVERSITAT DE BARCELONA

TFG - Grau de Matemàtiques

An analysis of metric and topological estimators of generalization in deep learning

January 2021

Authored by

Xavier Arnal Clemente

Directed by

Sergio Escalera Guerrero
Carles Casacuberta Vergés
and
Meysam Madadi
Ciprian Corneanu

Contents

Abstract

One of the most important notions in deep learning (or in statistical learning) is that of *generalization* — the capacity of a model trained on some data to perform well on the latent source of truth from which its training data has been drawn. Though some ways exist to estimate a model's capacity to generalize from its internal structure or performance during training, it is generally not well understood what distinguishes the models that generalize from the ones that do not. In this text, we formalize, expand on and give some basic results regarding a previously proposed way to extract a finite metric space from a neural network. This is done with the intention of using this finite metric space as a starting point to perform techniques from topological data analysis, and use the data obtained about the latent topology of this metric space to predict the capacity to generalize of a model. Moreover, we compare the performance of several estimators with other known predictors of generalization, and with estimators based solely on the metric information obtained from the model.

Resumen

Uno de los conceptos más importantes en el aprendizaje profundo (o en la regresión estadística) es la *generalización* — la capacidad de un modelo entrenado sobre unos ciertos datos de dar buenos resultados en la fuente de verdad subyacente a estos datos. Aunque existen algunos métodos para estimar la capacidad de generalizar de un modelo conociendo solo su estructura interna o comportamiento durante el entrenamiento, en general no se conoce con seguridad qué distingue los modelos que generalizan de aquellos que no lo hacen. En este texto, formalizamos, extendemos y damos algunos resultados básicos sobre una manera previamente propuesta de extraer un espacio métrico finito de una red neuronal. Esto se hace con la intención de utilizar este espacio métrico finito como punto de partida para aplicar técnicas de análisis de datos topológico, y usar los datos obtenidos sobre la topología latente de este espacio métrico para predecir la capacidad de generalización de un modelo. Además, comparamos los resultados obtenidos de estas predicciones con las estimaciones dadas por otros predictores conocidos de la generalización, y con predictores basados directamente en la información métrica del modelo.

Resum

Un dels conceptes més importants en l'aprenentatge profund (o a la regressió estadística) es la *generalització* — la capacitat d'un model entrenat amb certes dades de donar un bon resultat a la font de veritat subjacent d'on s'han extret aquestes dades. Encara que hi ha maneres d'estimar la capacitat d'un model de generalitzar coneixent només la seva estructura interna o comportament durant l'entrenament, en general no es coneix bé què distingeix els models que generalitzen d'aquells que no ho fan. En aquest text, formalitzem, estenem, i donem alguns resultats bàsics sobre una manera prèviament proposada d'extreure un espai mètric finit d'una xarxa neuronal. Això es fa amb la intenció d'emprar aquest espai mètric finit com a punt de partida per a utilitzar tècniques d'anàlisi de dades topològica, i fer servir les dades obtingudes sobre la topologia latent d'aquest espai mètric per predir la capacitat de generalització d'un model. A més, comparem els resultats d'aquestes prediccions amb les estimacions donades per altres predictors coneguts de la generalització, i amb predictors basats directament en la informació mètrica del model.

Chapter 1

Introduction

Deep neural networks are, in essence, a very powerful form of statistical regression. Arguably not the most important, but the most obvious difference between DNNs and classical classifiers is the sheer amount of parameters DNNs employ — for instance, the fairly recent GPT3 model's weights number in the hundreds of billions. However, it is not difficult to endow a classical classifier with a similarly ludicrous amount of parameters and watch it excel at classifying its learning data — and fail with anything outside of the training dataset. The actual strengh of DNNs lies in their (comparatively higher) resistance to overfitting when trained properly.

This begs the question: What can a deep neural network do with a billion extra parameters that a classical classifier cannot? Presumably some part of the answer lies in the use of the word "deep" — there is empirical evidence ([8]) that intermediate layers in DNN models serve to detect increasingly complex patterns in the input. However, the fact that DNNs *can* do this does not mean that they *do*: like any regression model, they are susceptible to overfitting. One might guess, then, that how well a network *generalizes* (that is, how well it performs on real, unseen data with respect to its performance during training) has to do with how well it has managed to build these internal representations of complex patterns that are significant in the input data.

The generalization problem, then, has to do with predicting how well a network generalizes, given no information about its performance on unseen data — just from looking at the network itself and at the training process. It is not hard to imagine why this is an important problem in practice — but, if our previous guess is true, how should one go about figuring out whether a DNN has "learned" any complex patterns?

There exist numerous approaches to this ([15, 21, 27]). We follow the ideas laid out by Corneanu et al [7] — namely, building a finite metric space out of a neural network, where the vertices are nodes and their distances are determined by some measure of how functionally related these nodes are. Importantly, this does not explicitly carry any structural information about the network, and yet it has significant predictive power regarding generalization.

The building of a finite metric space allows for the usage of methods from *Topological Data Analysis*, a fairly recent field concerned with associating (in a principled way) topological objects to point clouds and finding topological features of these objects, in that way inferring topological features of some hypothetical unkown space from where these point clouds were sampled. TDA has found extensive application, particularly in biology and medicine ([11, 18, 23, 28]), as it can provide nuanced information about the structure of data.

The text is structured as follows:

Chapter 2 contains the theoretical background for the rest of the text. In Section 1 we build up a rigorous definition of neural network, and give some explanation of the basic concepts at play in deep learning. In Section 2 we go through the usual pipeline of Topological Data Analysis: from its building blocks in the form of simplices and filtrations, to the construction of these filtrations from point clouds, to the definition of and some of the intuition behind the persistent homology of these filtrations.

Chapter 3 is concerned with the process of extracting a finite metric space from a neural network. We closely follow the steps of [7], building upon this paper to provide a few alternative formulations, prove their metricity, and give some results on the convergence of these metric spaces with respect to sampling from an underlying dataset, as well as on the relationship between the few different metrics proposed.

Chapter 4 defines all the estimators of generalization that we include results about, as well as providing some results about the stability of the ones proposed, and expanding on some prior results regarding the stability of some other estimators.

Chapter 5 is about the setup of the experiments performed and the results obtained. It explains the sets of models on which tests were performed and the measures of the relationship between estimators, as well as going through the conclusions drawn from the results of the experiments.

1.1 Notation

- $\#A$ is the cardinality of a set A

- $[a...b]$ is the integer range $[a,b] \cap \mathbb{Z}$

- If $\mathbf{x}$ is some tuple of integers $\mathbf{x} = (x_1, ..., x_n)$, then $[1...\mathbf{x}]$ is the set of integer tuples $[1...x_1] \times \cdots \times [1...x_n]$

- $\mathcal{T}(\mathbf{x};\mathbb{K})$ is the set of tensors over a field $\mathbb{K}$ of shape $\mathbf{x}$

- $\mathbb{R}_{\neq 0}$ denotes the nonzero reals $\{x \in \mathbb{R} \mid x \neq 0\}$

- $\mathbb{R}_{\geq 0}$ denotes the nonnegative reals $\{x \in \mathbb{R} \mid x \geq 0\}$

- sgn is the sign function $sgn(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$

- $x^{\perp}$ is the space orthogonal to $x \in V$, $\{y \in V \mid \langle x, y \rangle = 0\}$

- For $f : A \to B; x \mapsto f(x)$ and $g : C \to D; x \mapsto g(x)$, $f \times g : A \times C \to B \times D; (x,y) \mapsto (f(x), g(y))$ is the Cartesian product of f and g.

- $\mathfrak{P}(X)$ is the power set of X, $\{Y \subseteq X\}$

Chapter 2

Preliminary theory

This chapter provides a brief overview of the two main topics underlying this text, namely, deep learning and topological data analysis. These are intended to both serve as an introduction to either topic and provide a proper mathematical grounding for their usage in the rest of the text.

2.1 Deep learning

2.1.1 Fitting

Let X, Y be two sets and $\mathcal{D} \subseteq X \times Y$, and define $\mathcal{D}_x = p_X(\mathcal{D})$. Loosely speaking, a fitting problem is to do with finding some reasonable mapping $f \colon X \to Y$ such that $\{(x, f(x)) \mid x \in \mathcal{D}_x\}$ is in some sense similar to the "data" $\mathcal{D}$. Often, $\mathcal{D}$ is understood to be of the form $\{(x, g(x))\}$, perhaps with some inaccuracy. In this way, the process of finding f can be understood as approximating an unknown mapping g. A complete definition of a fitting problem requires two more pieces:

1. A way to measure how good (or bad) f is at fitting $\mathcal{D}$. Usually called loss, this takes the form of a mapping $\mathcal{L} : Y^X \to \mathbb{R}$, higher the worse its input is at fitting $\mathcal{D}$. We will assume $\mathcal{L}(f)$ can be written as $\frac{1}{|\mathcal{D}|} \sum_{x,y \in \mathcal{D}} \ell(f(x), y)$ (wherein ℓ measures how far f is off at a particular datapoint (x, y)), as this is very often the case.

2. A parametrized set $F \subset Y^X$ from which we intend to pick out our solution. This defines exactly what we mean by a "reasonable mapping", as otherwise it is trivial to select an f perfectly fitting $\mathcal{D}$ as long as p_X is injective over $\mathcal{D}$.

A classic example of a fitting problem is LLS, where $X = \mathbb{R}^n$, $Y = \mathbb{R}^m$, $F = \{f : X \to Y \,|\, f \text{ is linear}\}$, and $\mathcal{L}$ is defined through $\ell(x, y) = ||x - y||_2$. It is well known that in this setting, the ideal solution $argmin_{f \in F}\, \mathcal{L}(f)$ is attainable. However, this is not usually the case: it is too naive to say that the solution to an arbitrary fitting problem is finding $argmin_{f \in F}\, \mathcal{L}(f)$, or even getting loss arbitrarily close to $min_{f \in F}\, \mathcal{L}(f)$. Indeed, a "solution" to a fitting problem is usually a reliable way to find some $f \in F$ with sufficiently small $\mathcal{L}$.

A prototypical example of such a solution is **gradient descent**. To define it, we need some specific parametrization of F, say $F = \{f_\theta \,|\, \theta \in \Theta\}$ with Θ an open subset of $\mathbb{R}^n$. Then, $\theta \mapsto \mathcal{L}(f_\theta)$ is a function from $\Theta \subseteq \mathbb{R}^n$ to $\mathbb{R}$. In more concrete settings, it is often easy to see that $\theta \mapsto \mathcal{L}(f_\theta)$ is differentiable. This enables us to take its gradient, $\nabla_\theta \mathcal{L}(f_\theta)$, which will be the direction of steepest ascent of loss. Thus, given some sufficiently small scaling constant λ and any $\gamma_0 \in \Theta$, the iterative update rule

$$\gamma_{i+1} = \gamma_i - \lambda \nabla_\theta \mathcal{L}(f_\theta)(\gamma_i)$$

will yield a series $\{\gamma_i\}_{i \in \mathbb{N}}$ such that $\{f_{\gamma_i}\}_{i \in \mathbb{N}}$ converges (in the topology induced by the parametrization) to some local minimum of $\mathcal{L}$.

Gradient descent has a number of weaknesses. Among others,

- It is slow to compute for very large $\mathcal{D}$.
- It can and will get stuck on local minima.
- It is often quite slow — each iteration only displaces γ a small amount in Θ.

There is a rich family of methods based on gradient descent that tackle these and other weaknesses. State-of-the-art methods nowadays are quite robust, in particular to the usage of Fs with very high-dimensional parametrizations, as this is their most common use case. The next section will proceed to describe one very important possible form of F.

2.1.2 Computation graphs

In a general sense, a computation graph is a description of a certain type of function and a certain way to compute it, with an emphasis placed on parametrizing these types of functions. Structurally, it is defined as a directed graph that admits a linear order, with each node of this graph representing a variable bound to some expression of the variables from its incident nodes. In this way, one can start with some values at some set of input nodes, *propagate* these values using the expressions through the directed graph, and obtain the value of certain output nodes. A precise definition needs to be more specific than this, starting with the meaning of graph:

A **directed graph** of vertex set V and edge set E is a tuple (V, E) such that $\#V < \infty$ and $E \subseteq V \times V$. Note that this excludes infinite graphs or multigraphs, as they will not be of any use in this text.

Let $G = (V, E)$ be a directed graph. For any $e \in E$, we denote by e^0 and e^1 the two vertices from V such that $e = (e^0, e^1)$. Then, a **path** in G is a tuple $(e_1, ..., e_n) \in E^n$ such that $e_i^1 = e_{i+1}^0$ for all $i \in [0...n-1]$. In particular, this path is a *cycle* if $e_n^1 = e_1^0$. A directed graph is **acyclic** if it contains no cycles.

As $E \subseteq V \times V$, E can be interpreted as a relation over V. We use $\prec_E$ (or just $\prec$, if E is clear from the context) to denote the transitive closure of E. Then, if G is acyclic, $\prec_E$ is a strict partial order — this is a well known property of directed acyclic graphs.

Definition 2.1. *A **computation graph schema** is a tuple (V, E, r, σ) such that*

i) *(V, E) is a directed acyclic graph.*

ii) *S_I and S_O (occasionally I and O) are the **input nodes** and **output nodes** of S, defined as the minimal and maximal sets of V with respect to $\prec_E$.*

iii) *r can be written as $r = \{r_v\}_{v \in V}$, and $\forall v \in V$, $r_v = \mathcal{T}(...;\mathbb{R})$ (as a real vector space).*

iv) *σ can be written as $\sigma = \{\sigma_v\}_{v \in V \setminus I}$, and $\forall v \in V$, $\sigma_v : r_v \to r_v$.*

Note that for a specific schema S, (V, E, r, σ) might be written as $(S_V, S_E, S_r, S_\sigma)$. For any $v \in V$, the shape of the tensors in r_v will be written as $shape(r_v)$.

In the definition above, r_v represents the values one particular node might take. The meaning of this, as well as the purpose of σ, will be made clear in the next few definitions. Nonetheless, it is worth noting that the choices regarding the restrictions put on r_v are by no means canonical: For instance, for the most part we are only interested in r_v being finite-dimensional real vector spaces with some notion of a canonical basis. Modifying these concepts as applicable can yield more complex families of computation graphs (e.g., Grassmann Networks [13]) which are not considered here.

Definition 2.2. *A **transition set** for a computation graph schema S is an S_E-indexed set $w = \{w_e\}_{e \in S_E}$ such that w_e, at times written w_{e^0, e^1}, is a function $r_{e^0} \to r_{e^1}$.*

Definition 2.3. *A **bias set** for a computation graph schema S is an S_V-indexed set $b = \{b_v\}_{v \in S_V \setminus S_I}$ such that $\forall v \in S_V$, $b_v \in r_v$.*

Definition 2.4. *A **computation graph** is a tuple (S, w, b) such that S is a computation graph schema and w, b are transition and bias sets for S.*

2.1.3 Propagation

This section is concerned with how a computation graph behaves in a dynamic sense, that is, how data gets from the input to the output. For this, it is necessary to be precise about the meaning of input and output: in this context, an input for some computation graph $G = (S, w, b)$ is of the form $x = \{x_i\}_{i \in S_I}$, with $x_i \in (S_r)_i$, and likewise output $y = \{y_o\}_{o \in S_O}$. Importantly, more often than not there will only be one input and output node, that is, $S_I = \{i\}$ and $S_O = \{o\}$. In these cases, to avoid cumbersome notation, input and output may just be represented by elements of $(S_r)_i$ and $(S_r)_o$, respectively.

Definition 2.5. *The **input space** ($\mathcal{I}_G$) and **output space** ($\mathcal{O}_G$) of a computation graph G, or of its underlying computation graph schema, are the sets of inputs and outputs of G, respectively, as described in the paragraph above.*

To connect the input and the output, it is necessary to involve the rest of the graph. In particular, as both inputs and outputs are mappings from input or output nodes to values, a mapping from each node to a value is defined:

Definition 2.6. *A* ***state*** *of a computation graph $G = (S, w, b)$, with $S = (V, E, r)$, is a V-indexed mapping $s = \{s_v\}_{v \in V}$ such that $\forall v \in V$, $s_v \in r_v$.*

Definition 2.7. *A* ***consistent state*** *of a computation graph $G = (S, w, b)$, with $S = (V, E, r)$, is a state s of G such that*

$$\forall v \in V \setminus I,\, s_v = \sigma_v\big(b_v + \sum_{v' \in a(v)} w_{(v',v)}(s_{v'})\big).$$

Then, connection of the input with the output (or rather, the rest of the nodes in the graph), is defined by a certain mapping P_G:

Definition 2.8. *The* ***propagation function*** *of a computation graph G is a function P_G that maps any input x to the unique state $P_G(x)$ satisfying:*

i) *$P_G(x)$ is consistent.*

ii) *$\forall i \in I$, $P_G(x)_i = x_i$.*

Of course, it is not obvious that this is well defined. For this, we must show that for any computation graph $G = (S, w, b)$, with $S = (V, E, r)$, and any input x:

1) There exists a consistent state s with $s_i = x_i$ for all $i \in I$.

Proof. Consider any topological ordering q of $V \setminus I$, that is, a $[1...\#V - \#I]$-indexing of $V \setminus I$, written as $V \setminus I = \{q_i\}_{i \in [1...\#V - \#I]}$, such that $q_i \prec_E q_j \implies i < j$. Then, one may simply create a state s with all the input values as in x, as necessary, and construct for each i in $[1...\#V - \#I]$, in order, set $s_{q_i} = \sigma_{q_i}\big(b_{q_i} + \sum_{(v,q_i) \in E} w_{(v,q_i)}(s_v)\big)$. This is always defined, as each v appearing in the sum is such that $(v, q_i) \in E \implies v \prec_E q_i \implies v = q_j$ with $j < i$, meaning all s_v are defined beforehand. □

2) This state is unique.

Proof. Let s and s' be two consistent states satisfying the definition of $P_G(x)$, and assume for the sake of obtaining a contradiction that they are different. Then, there must be a set $U \subseteq V$ where s and s' disagree. Let v be any minimal element of U with respect to $\prec_E$. Then, either $v \in I$, which would mean that at least one of s, s' does not satisfy the definition of $P(x)$, or

$$\begin{aligned} s_v \neq s'_v &\implies \sigma_v\big(b_v + \sum_{(v,v') \in E} w_{(v',v)}(s_{v'})\big) \neq \sigma_v\big(b_v + \sum_{(v',v) \in E} w_{(v',v)}(s'_{v'})\big) \\ &\implies \sum_{(v,v') \in E} w_{(v',v)}(s_{v'}) \neq \sum_{(v',v) \in E} w_{(v',v)}(s'_{v'}) \\ &\implies \exists (v', v) \in E \mid s_{v'} \neq s'_{v'} \implies \exists v' \in U \mid v' \prec_E v. \end{aligned}$$

This is a contradiction, as we wanted to show. □

Note that while the definition of P_G itself is not very helpful for implementation, **(1)** does provide a basic description of the computation process. Moreover, **(2)** inadvertently proves the less-obvious fact that the result of **(1)** does not depend on the linear ordering used. In any case, this is enough to actually use the computation graph for computation:

Definition 2.9. *The **functional representation** (or just **function**) of a computation graph $G = (S, w, b)$ is the function f_G mapping an input x to the output $\{P_G(x)\}_{o \in S_O}$.*

2.1.4 Parametrization and backpropagation

We are now interested in using computation graphs in the way suggested in Section 2.1.1, that is, given some loss function $\mathcal{L}$, defining F as a certain family of computation graphs from which we intend to pick out one with sufficiently small $\mathcal{L}$. For this, we parametrize some such set of computation graphs, and endow it with some differentiable structure to allow the application of gradient descent:

Definition 2.10. *A **parametrized computation graph** G is a tuple (S, Θ, b, w) such that*

i) *S is a computation graph schema;*

ii) *$\Theta \subseteq \mathbb{R}^n$ is an (open) parameter set;*

iii) *b and w are functions mapping $\theta \in \Theta$ to b_θ and w_θ, where $\forall \theta \in \Theta$ b_θ and w_θ are bias and transition sets for S.*

Then, for any $\theta \in \Theta$, $G_\theta := (S, w_\theta, b_\theta)$, that is, the specific computation graph obtained by setting the biases and transitions at b_θ and w_θ.

Definition 2.11. *A **differentiably parametrized computation graph** is a parametrized computation graph (S, Θ, b, w) such that*

i) *$\forall v \in S_V$, $\theta \mapsto (b_\theta)_v \in \mathcal{C}^1(\Theta, r_v)$;*

ii) *$\forall (v, u) \in S_E$, $\forall \theta \in \Theta$, $x \mapsto (w_\theta)_{v,u}(x) \in \mathcal{C}^1(S_{r_v}, S_{r_u})$;*

iii) *$\forall (v, u) \in S_E$, $\forall x \in S_{r_v}$, $\theta \mapsto (w_\theta)_{v,u}(x) \in \mathcal{C}^1(\Theta, S_{r_u})$.*

The application on gradient descent, then, relies on the existence of $\frac{\partial \mathcal{L}}{\partial \theta}$. Showing that this differential does indeed exist, however, is extremely cumbersome in our notation, owing to the fact that it is significantly more natural to associate a *variable* to each vertex $v \in S_V$, rather than working with a purely functional approach. Borrowing this convention, then: for any parametrized computation graph $G = (S, \Theta, b, w)$, with $S = (V, E, r, \sigma)$, $\forall v \in S_V$, let x_v be a variable associated to v, living in r_v, constrained to $\forall v \in S_V \setminus S_I$, $x_v = (b_\theta)_v + \sum_{v' \in a(v)} (w_\theta)_{(v',v)} (\sigma_{v'}(s_{v'}))$. Note that this is not completely analogous to Definition 2.7 — $(P_G(x))_v$ would correspond to $\sigma_v(x_v)$ rather than x_v directly. Then, taking advantage of the differentiability conditions imposed in Definition 2.11 to do some routine computation (in point-free form for simplicity, though it is an abuse of notation in this case):

$$\forall v \in S_V \setminus S_O, \quad \frac{\partial \mathcal{L}}{\partial x_v} = \frac{\partial \sigma_v(x_v)}{\partial x_v} \frac{\partial \mathcal{L}}{\partial \sigma_v(x_v)} = \nabla \sigma_v \sum_{(v,u) \in E} \frac{\partial \mathcal{L}}{\partial x_u} \frac{\partial x_u}{\partial x_v}$$

$$\frac{\partial x_u}{\partial x_v} = \frac{\partial\Big((b_\theta)_u + \sum_{u' \in a(u)} (w_\theta)_{(u',u)}\big(\sigma_{u'}(s_{u'})\big)\Big)}{\partial x_v} = \frac{\partial\Big((w_\theta)_{(v,u)}\big(\sigma_v(x_v)\big)\Big)}{\partial x_v} = \nabla\big((w_\theta)_{(v,u)} \circ \sigma_v\big).$$

Thus, eschewing the point-free style:

$$\forall v \in S_V \setminus S_O, \quad \frac{\partial \mathcal{L}}{\partial x_v}(x_v) = \nabla\sigma_v(x_v) \cdot \sum_{(v,u)\in E} \left(\frac{\partial \mathcal{L}}{\partial x_u}(x_u) \cdot \nabla\big((w_\theta)_{(v,u)} \circ \sigma_v\big)(x_v)\right).$$

This is similar to the expression in Definition 2.7, but instead of defining x_v in terms of the x_u for $u \in S_V$ before v in $\prec_E$, starting with S_I, it defines $\frac{\partial \mathcal{L}}{\partial x_v}$ in terms of the $\frac{\partial \mathcal{L}}{\partial x_u}$ for $u \in S_V$ after v in $\prec_E$, starting at S_O. As this is a similar propagation process in the opposite direction, it is known as **backpropagation**.

What we have right now, however, is not what we needed for gradient descent (namely, $\nabla_\theta \mathcal{L}$). However, it easy to see that

$$\forall v \in V \setminus I, \quad \frac{\partial \mathcal{L}}{\partial (b_\theta)_v} = \frac{\partial \mathcal{L}}{\partial x_v},$$

$$\forall (u,v) \in E, \quad \frac{\partial \mathcal{L}}{\partial (w_\theta)_{(u,v)}(\sigma_u(x_u))} = \frac{\partial \mathcal{L}}{\partial x_v}.$$

Therefore, provided that all of the x_v are known, and $\frac{\partial \mathcal{L}}{\partial x_v}$ can be obtained through backpropagation, these last two equations let us compute $\nabla_\theta(\mathcal{L})$. This shows that gradient descent is a viable technique for using differentiably parametrized computation graphs in fitting problems.

It should be noted that, more often than not, w_θ, b_θ, and the parametrization thereof are nowhere near as complicated as this section allows them to be. Very commonly $(b_\theta)_v$ is simply some set of coordinates of θ in the shape of a tensor. Similarly, $(w_\theta)_e$ is often just multiplication by some tensor parametrized in the same way as was just described, although more complicated linear operations (convolutions, in particular) are frequent as well.

2.1.5 Neural networks

Though computation graphs (differentiably parametrized ones, in particular) are clear stand-ins for non-recurrent traditional neural networks, the fact that they are named differently is deliberate. In a way, neural network is more of a semantic term that is not wholly captured by the definition of computation graph — for instance, adequately modelling the behaviour of Radial Basis Function Networks [5] with computation graphs is hardly doable with the definitions given: it would require associating *two* different r_v's with each v, and letting $\sigma_v : r_v \to r'_b$. However, RBFNs clearly *are* still neural networks. In the same vein, our definition does not allow for the parametrization of σ_v, but this is a fairly common practice (such as with Swish [25] or several variants of ReLu).

The distinction being made here is largely analogous to the difference between a Turing machine and a programming language: though they have significant overlap and might even be equivalent in a theoretical sense, their purpose is different. Indeed, computation graphs are only being defined rigorously to lend credence to the rest of definitions in this text that will depend on computation graphs. With the same analogy as before — much like a good theoretical computer scientist need not make a good software designer, the realm of neural networks in practice (and theory) is much richer than this very brief introduction would suggest. We will briefly step into this practical realm to define some concepts that might be relevant to this text.

First of all — for the sake of readability:

Definition 2.12. *A **neural network** is a differentiably parametrized computation graph.*

Specific "instances" of a neural network, that is, G_θ for specific values of θ, may also be described as a neural network.

A **multilayer perceptron** (MLP) with $h \in \mathbb{N}$ hidden layers of widths $(w_1, ..., w_h) \in \mathbb{N}^h$, input width w_0 and output width w_{h+1}, with activation function $\sigma\colon \mathbb{R} \to \mathbb{R}$, weights $\{\mathbf{w}_i \in \mathcal{T}(w_{i-1}, w_i; \mathbb{R})\}_{i \in [1 \dots h+1]}$, and biases $\{\mathbf{b}_i \in \mathcal{T}(w_i; \mathbb{R})\}_{i \in [1 \dots h+1]}$, is the computation graph (S, f, b), with $S = (V, E, r, \overline{\sigma})$, where:

- (V, E) forms the following directed graph:
  $$I = L_0 \xrightarrow{\qquad} L_1 \longrightarrow L_2 \longrightarrow \cdots \longrightarrow L_{h-1} \longrightarrow L_h \xrightarrow{\qquad} L_{h+1} = O$$
- $f_{(L_{i-1}, L_i)}\colon x \mapsto \mathcal{T}(w_i; \mathbb{R}) \cdot x$
- $b_{L_i} = \mathbf{b_i}$
- $r_{L_i} = \mathcal{T}(w_i; \mathbb{R})$
- For every vertex v, $\overline{\sigma}_v$ simply maps every coordinate of r_v through the σ provided.[1]

Multilayer perceptrons can also be seen as (differentiably) parametrized computation graphs, with h, w and σ fixed, as they are parametrized by $(\mathbf{b_1}, ..., \mathbf{b_{h+1}}, \mathbf{w_1}, ..., \mathbf{w_{h+1}})$.

To avoid confusion with the use of "parameter" in "parametrized computation graph", values like h or w_i that define the structure of a neural network are usually called **hyperparameters**. These two examples are *structural* hyperparameters — in contrast to *training* hyperparameters, which are predefined values that tune the behaviour of the training algorithm (e.g., learning rate).

Neural networks have found plenty of use in computer vision. A practically universal feature of NNs applied to this field is an internal representation of images. Note that,

[1] Usually the activation function on the last layer is something else — commonly *softmax* or some other differentiable surrogate of argmax.

intuitively, the values that will be held in the nodes of a multilayer perceptron are vectors (vectors in the list-of-scalars sense) — for some node v to hold an image, we would require that $r_v = \mathcal{T}(width, height, \#channels; \mathbb{R})^2$.

Consider then some network having this capacity to have internal representations of images. It is common for transforms between "image nodes" to be (discrete) convolutions — note that, given a fixed kernel size, these are easily parametrizable linear transformations. This explains the name commonly given to these "image networks": **convolutional neural networks** (CNNs).

Generalization

Recall that a dataset $\mathcal{D}$ is a set of pairs (x, y), and fitting this dataset requires finding some mapping satisfying $x \mapsto y$ for all pairs, to a reasonable degree of accuracy. Say then we are training a model on some (training) dataset $\mathcal{D}_{train}$. Loosely speaking, for this model to *overfit* would mean that it memorized $\mathcal{D}_{train}$, rather than learning any meaningful pattern in this dataset.

Insofar as polynomial interpolation is a form of fitting, it can also be a prototypical example of overfitting. In fact, it is possible for a polynomial to perfectly fit some dataset but get arbitrarily worse as more data is provided — this would be an example of *Runge's phenomenon*, depicted in Fig. 2.1.

The opposite of overfitting, then, would be *generalization* — intuitively, a model's ability to learn from a dataset rather than memorize it. This is measured by comparing its accuracy on this training dataset to its accuracy on some unseen ("validation") dataset assumed to be some unbiased representation of the latent source of the training data. If these two datasets are $\mathcal{D}_{train}$ and $\mathcal{D}_{val}$, the **generalization gap** of a model f is

$$gen(f) := acc(f, \mathcal{D}_{train}) - acc(f, \mathcal{D}_{val}), \text{ where } acc(f, \mathcal{D}) := \frac{\#\{(x,y) \in \mathcal{D} \mid f(x) = y\}}{\#\mathcal{D}}.$$

At times, differences in the accuracy of f in $\mathcal{D}_{train}$ and $\mathcal{D}_{val}$ can be explained through differences between $\mathcal{D}_{train}$ and $\mathcal{D}_{val}$ — such as when $\mathcal{D}_{train}$ has some bias not present in the validation dataset. We assume this not to be the case, and thus, by definition, a large generalization gap can only be explained by f treating data it has seen before differently from data it has not — meaning that the model has learned about the specific data provided, as opposed to the patterns underlying it.

This only serves as intuition, but it informs our (and many other) approaches to estimating the generalization gap without knowledge of any validation dataset — by trying to detect

[2] The usage of $\mathbb{R}$ might sound dubious, as an actual computer representation of an image would be more along the lines of $\mathcal{T}(width, height, \#channels, [0...255])$, ignoring the fact that [0...255] is not a field. However, the usage of $\mathbb{R}$ doesn't stop a node from holding values from $\mathcal{T}(width, height, \#channels, [0...255])$ — replacing [0...255] with $\mathbb{R}$ simply endows "image space" with the continuity that NNs benefit from.

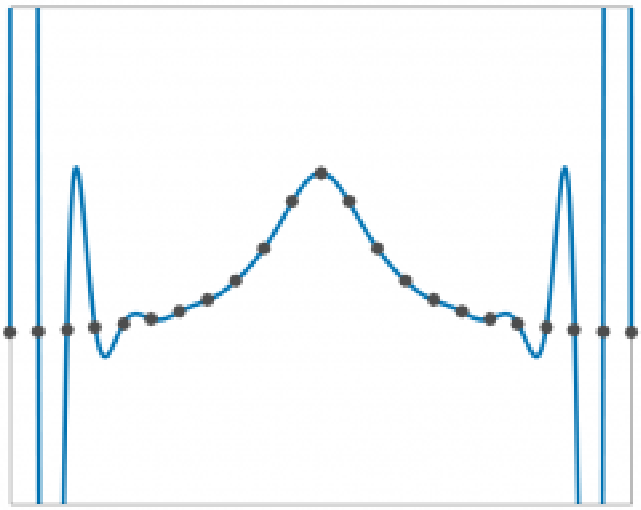


Figure 2.1: An illustration of *Runge's phenomenon*; specifically, the unbounded error of successive interpolations of $f(x) = 1/(1+25x^2)$ from points uniformly sampled from $[-1,1]$ (i.e., a dataset sampled from the latent source $f(x)$).

structures within the model indicating learning as opposed to overfitting. In particular, we are interested in topological information about some geometric representation of the model — this information is obtained through Topological Data Analysis, explained in the following section.

2.2 Topological Data Analysis

2.2.1 Simplicial complexes

Simplices are typically imagined as a concept from affine geometry — namely, an n-simplex is the convex hull of $n+1$ affinely independent points. As a geometric object, simplices are a particular family of polytopes — thus it makes sense to speak of their *faces*. A *simplicial complex*, then, is an arrangement of non-intersecting simplices in some affine space, where this condition of non-intersection does not proclude different simplices from sharing a face, edge, vertex, etc.

Simplices are, as their name suggests, a remarkably simple geometric object. For this reason, simplicial complexes are quite easy to describe combinatorially. For instance, given prior knowledge about the points **a** through **g**, the simplicial complex in Fig. 2.2 can be described by

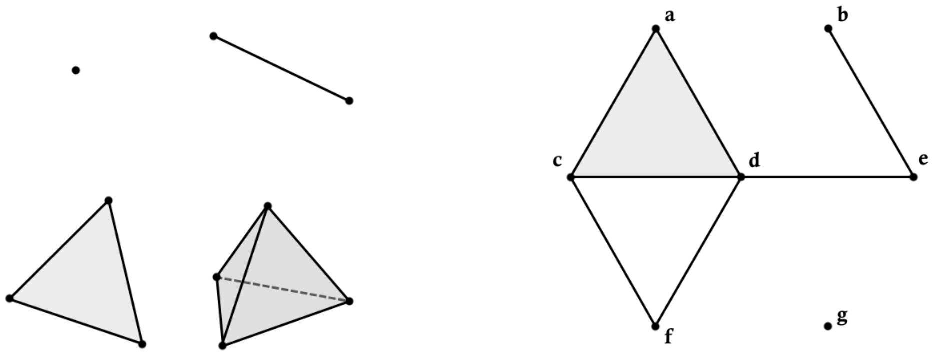


Figure 2.2: Left: n-simplices for n in 0–3. Right: a simplicial complex in two dimensions

$$\{\mathbf{acd}, \mathbf{ac}, \mathbf{ad}, \mathbf{be}, \mathbf{cd}, \mathbf{cf}, \mathbf{de}, \mathbf{df}, \mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}, \mathbf{f}, \mathbf{g}\},$$

where **acd** is the triangle of vertices $\mathbf{a}, \mathbf{c}, \mathbf{d}$, **de** is the line (1-simplex) from **a** to **c**, etc. The inclusion of 1-simplices such as **ac** or 0-simplices such as **a** may seem redundant, as they are included in other higher-order simplices (**acd**, for instance), but it will be in line with the following few definitions.

A simplicial complex, then, can be described by two sets:

1) A set of vertices belonging to some affine space.

2) A set of affinely independent subsets of the vertex set, each corresponding to the simplex defined by their convex hull.

We will concern ourselves with *abstract simplicial complexes*, which are, roughly speaking, simplicial complexes after having lost interest in information about any specific set of vertices.

An **abstract simplicial complex** is a set S of finite non-empty sets such that $\sigma \in S \;\wedge\; \rho \subseteq \sigma \;\wedge\; \rho \neq \varnothing \implies \rho \in S$. The σ in S are **abstract simplices**, and are said to be **faces** of S. Moreover, if σ, ρ are faces of S and $\sigma \subset \rho$, σ is also said to be a face of ρ.

It should be noted that this is not the most abstract definition of abstract simplicial complexes — instead of treating the simplices as sets of vertices, one can forget about the vertices altogether and say that an abstract simplicial complex is a set of elements with some relation playing the role of $\subseteq$. However, we do not adopt a definition along these lines, as this would make a good deal of the following definitions more complicated (or simply unnecessary).

The **vertex set** of an abstract simplicial complex S is $V_S := \bigcup S$. All of the abstract simplicial complexes treated will be assumed to be finite — meaning $\#S < \infty$, or equivalently,

$\#V_S < \infty$. The **dimension** of an abstract simplex σ is $dim(\sigma) := \#\sigma - 1$. Likewise, the dimension of an abstract simplicial complex S is $dim(S) := \max_{\sigma \in S} dim(\sigma)$.

Simplicial complexes as a topological space

In forgetting the actual positions of the vertices in the vertex set, we forgot any geometric information about simplicial complexes. However, an important thing to note is that the topology up to homeomorphism of a simplicial complex does not depend on the positions of its vertices, as long as they ensure that the non-intersection condition is satisfied. Therefore, we define for each simplicial complex S an associated topological space $|S|$, corresponding to the topology S would have if it was a specific simplicial complex living in affine space.

For this we need to introduce the concept of **ordered abstract simplicial complex** — a pair $(S, \prec)$, with S a simplicial complex and $\prec$ a total order on V_S. Then, for any $\sigma \in S$, σ_i denotes the i-th element of σ when ordered according to $\prec$, that is, $\sigma = \{\sigma_1, ..., \sigma_{\#\sigma}\}$, and $i < j \implies \sigma_i \prec \sigma_j$. Notice the abuse of notation in that σ_i does not show the dependence on $\prec$.

Now, let $\Sigma = (S, \prec)$ be an ordered abstract simplicial complex. Importantly, the requirement that S is finite implies V_S is enumerable — particularly enumerable in a way agreeing with $\prec$. Thus, let $\{v_i\}_{i \in [1...\#V_S]} = V_S$, like before, satisfying $i < j \implies v_i \prec v_j$.

Then, for each $\sigma \in S$, define

$$\Delta_\sigma^\Sigma := \Big\{ \sum_{v_i \in \sigma} t_i e_i \Big| \sum_{v_i \in \sigma} t_i = 1, \forall v_i \in \sigma, \, t_i \geq 0 \Big\},$$

where e_i is the i-th vector of the standard basis of $\mathbb{R}^{\#S_V}$. Note that this does not depend on $\prec$ (up to isometry): Let $\Sigma = (S, \prec)$ and $\Sigma' = (S, \prec')$ be two different ordered abstract simplicial complexes on the same underlying complex S, and $\{v_i\}_{i \in [1...\#V_S]}$, $\{v'_i\}_{i \in [1...\#V_S]}$ be orderings of V_S agreeing with $\prec$ and $\prec'$, respectively. Then

$$\Delta_\sigma^\Sigma = T(\Delta_\sigma^{\Sigma'}),$$

where T is the linear transform satisfying $v'_i = v_j \implies e_i \mapsto e_j$ for each $1 \leq i, j \leq \#V_S$. As this is a permutation on an orthonormal basis, T is an isometry. Now, let

$$\Delta_\Sigma = \bigcup_{\sigma \in S} \Delta_\sigma^\Sigma,$$

and likewise for $\Delta_{\Sigma'}$. With T as before, $\Delta_\Sigma = T(\Delta_{\Sigma'})$ — thus, if Δ_Σ and $\Delta_{\Sigma'}$ are interpreted as topological spaces (with the subset topology from $\mathbb{R}^{\#S_V}$), these spaces are homeomorphic. The **geometric realization** of an abstract simplicial complex S, denoted $|S|$, is Δ_Σ as a topological space, with Σ being any ordered abstract simplicial complex with S as an underlying complex.

Note that it is possible to extend this definition to non-finite abstract simplicial complexes — in fact, it is easy to avoid enumerating V_S altogether, by eschewing any immersions into an ambient space and simply gluing the disjoint union of all the Δ_σ^Σ along the faces accordingly. Moreover, as far as immersions into ambient space go, $\#V_S$ often overshoots the necessary dimensions quite a bit — it is actually not very hard to check that any abstract simplicial complex S can be embedded in $\mathbb{R}^{2\,dim(S)+1}$ without bending any of the Δ_σ^Σ.

A simplicial complex S, then, serves as a combinatorial way to describe a topological space $|S|$. It is natural to be concerned about what spaces admit such a combinatorial description — the spaces that are homeomorphic to $|S|$ for some S are said to be **triangulable**. A particular homeomorphism is a **triangulation**.

2.2.2 Simplicial homology

We are interested, then, in some topological characteristics of a space X, as told through some triangulation $\phi : |S| \to X$. For this we will use the concept of the *simplicial homology*[3] of an ordered abstract simplicial complex. To convince oneself that this will indeed give us topological information, it is important to note that:

1) The simplicial homology of $\Sigma = (S, \prec)$ agrees with the *singular homology* of $|S|$, and thus it makes sense to speak of the simplicial homology of S, as opposed to Σ ([12, Theorem 2.27]).

2) Singular homology (and thus, simplicial homology) is a homotopical invariant ([12, Corollary 2.11]).

Note, however, that simplicial homology of a triangulation *does not* determine the homotopy class of the space it is triangulating. Loosely speaking, the actual information conveyed by simplicial homology is the amount of n-dimensional holes in some space, for each $n = 1, 2, ...$, together with the amount of connected components. We will now build up a proper definition.

Let $\Sigma = (S, \prec)$ be an ordered abstract simplicial complex, and $p \in \mathbb{N}$. Then, the ***p*-th chain group** of Σ, $C_p(\Sigma)$, is the free abelian group with basis $\{\sigma \in \Sigma \mid dim(\sigma) = p\}$. Elements of $C_p(\Sigma)$ are ***p*-chains** of Σ.

Note that this last definition is equivalent to saying $C_p(\Sigma)$ is the free G-module over $\{\sigma \in \Sigma \mid dim(\sigma) = p\}$, with $G = \mathbb{Z}$. This is not the only choice for G — in fact, it is not really necessary to make a choice of G at all, so long as it is a nontrivial abelian group. It is worth noting that a common convention, especially among applications of TDA, is $G = \mathbb{F}_2$ — this significantly simplifies a few definitions (in fact, it avoids the need for an ordering on S_V altogether). Moreover, it provides a correspondence $C_p(\Sigma) \cong \mathfrak{P}(\{\sigma \in \Sigma \mid dim(\sigma) = p\})$, though this is best not relied upon for intuition.

[3] We will not define singular homology in this text. The interested reader is encouraged to look into [12].

Then, the **boundary operator** is a (series of) homomorphism(s) $\partial_p : C_P(\Sigma) \to C_{p-1}(\Sigma)$. It is sufficient to define their image on a basis of $C_p(S)$:

$$\partial_p(\sigma) = \sum_{i=1}^{p+1} (-1)^i \big(\sigma \setminus \{\sigma_i\}\big)$$

A very important property of ∂ is that $\partial_{p-1} \circ \partial_p = 0$. This is straightforward to check:

$$\begin{aligned}
\partial_{p-1} \circ \partial_p(\sigma) &= \partial_p \sum_{i=1}^{p+1} (-1)^i \big(\sigma \setminus \{\sigma_i\}\big) = \sum_{i=1}^{p+1} (-1)^i \partial_p (\sigma \setminus \{\sigma_i\}) \\
&= \sum_{i=1}^{p+1} (-1)^i \Big(\sum_{j=1}^{i-1} (-1)^j \big(\sigma \setminus \{\sigma_i, \sigma_j\}\big) + \sum_{j=i+1}^{p+1} (-1)^{j-1} \big(\sigma \setminus \{\sigma_i, \sigma_j\}\big)\Big) \\
&= \sum_{i=1}^{p+1} \sum_{j=1}^{i-1} (-1)^{i+j} \big(\sigma \setminus \{\sigma_i, \sigma_j\}\big) - \sum_{i=1}^{p+1} \sum_{j=i+1}^{p+1} (-1)^{i+j} \big(\sigma \setminus \{\sigma_i, \sigma_j\}\big),
\end{aligned}$$

where the third equality follows from enumerating the terms in $\partial_p\big(\sigma \setminus \{\sigma_i\}\big)$. Then, notice that the values i and j take in the last two sums are the same for both sums if i and j are flipped — but because the expression within both sums is invariant with respect to permutation of i and j, both sums are equal, and their difference is 0 — as we wanted to verify.

In a sense, it might be more fair to call $\partial_{p-1} \circ \partial_p = 0$ a defining property of ∂_p, as the actual definition of $\partial_p(\sigma)$ is just a linear combination of the maximal faces of σ with the coefficients chosen to satisfy $\partial_{p-1} \circ \partial_p = 0$. A discussion of this is available in [9].

The ***p*-boundaries** and ***p*-cycles** of Σ are the elements of $\mathrm{Im}(\partial_{p+1})$ and $\mathrm{Ker}(\partial_p)$, respectively. We have just seen that all p-boundaries are p-cycles — however, not all p-cycles are p-boundaries. Loosely speaking, for some p-cycle c, the non-existence of some $c' \in C_{p+1}(\Sigma)$ satisfying $\partial_{p+1}(c') = c$ (and thus $\partial_p \circ \partial_{p+1}(c') = 0$) implies at least some of the "interior" of c is missing in Σ. More precisely, the ***p*-th homology group** of Σ is

$$H_p(\Sigma) = \mathrm{Ker}(\partial_p) / \,\mathrm{Im}(\partial_{p+1}),$$

and the ***p*-th Betti number** of Σ, written $\beta_p(\Sigma)$, is the rank of $H_p(\Sigma)$. Thus, p-dimensional voids in Σ can be understood as generators in $H_p(\Sigma)$, and consequently $\beta_p(\Sigma)$ is often described as counting the amount of p-dimensional voids in Σ (in particular, $\beta_0(\Sigma)$ is the number of connected components).

As was said before, the simplicial homology of $\Sigma = (S, \prec)$ depends only on $|S|$, that is, on S. Therefore, we may write $H_p(S)$ or $\beta_p(S)$ instead of using Σ.

2.2.3 Filtrations

It is worth pointing out that in the phrase "topological data analysis", the adjective "topological" is modifying "data analysis", not "data" by itself — that is, the data that we start with is not necessarily topological in nature. In this subsection we explain one of the forms the starting data can take, and how the tools from the the two prior subsections can be leveraged to understand this data.

For a metric space $\mathcal{M}$, $\mathcal{M}_X$ and $d_{\mathcal{M}}$ will denote the underlying set and distance function of $\mathcal{M}$, respectively. Then:

Definition 2.13. *A **point cloud** is a pair $(P, \mathcal{M})$, where $\mathcal{M}$ is a metric space and P is a finite subset of $\mathcal{M}_X$.*

In general, one assumes that P is sampled from some subspace of $\mathcal{M}_X$ with a non-trivial topology (perhaps with some inaccuracy), and the objective is to recover some information about the homology class of this subspace. This is not necessarily always the case, but it informs the next few definitions.

For some metric space $\mathcal{M}$, let $B_r^{\mathcal{M}}(p)$ be the r-radius open ball centered at p in $\mathcal{M}$.

Definition 2.14. *The **r-dilation** of a point cloud $PC = (P, \mathcal{M})$ is the subspace $D_r(PC) = \bigcup_{p \in P} B_r^{\mathcal{M}}(p)$.*

Definition 2.15. *The **Čech complex** with radius r of a point cloud $PC = (P, \mathcal{M})$ is the abstract simplicial complex*

$$\check{C}ech_r(PC) = \{\sigma \subseteq P \mid \bigcap_{p \in \sigma} B_r^{\mathcal{M}}(p) \neq \varnothing\}.$$

Hence, $\{B_r^{\mathcal{M}}(p)\}_{p \in P}$ is an open cover of $D_r(PC)$, and $\check{C}ech_r(PC)$ is the **nerve** of this open cover (note that its definition depends only on the $B_r^{\mathcal{M}}(p)$, that is, the elements of the open cover). We will need a **nerve lemma**, which is the name commonly given to results relating a topological space to the geometric realization of the nerve of some open cover of this space, given some conditions on this open cover — one such nerve lemma could be Theorem 2.4 in [10]. Applying this result requires verifying that each $\bigcap_{p \in \sigma} B_r^{\mathcal{M}}(p)$ is contractible — this is straighforward to verify for a suitable range of r for all the metric spaces we will use later on. However, this will not be really necessary, as we will later justify switching over to a different kind of complex that is not guaranteed to be topologically equivalent to $D_r(PC)$.

In any case, assuming it is indeed true that each $\bigcap_{p \in \sigma} B_r^{\mathcal{M}}(p)$ is contractible, the singular homology of $D_r(PC)$ and the simplicial homology of $\check{C}ech_r(PC)$ agree. The tools in Section 2.2.2, then, will allow us to study the singular homology of $D_r(PC)$ through $\check{C}ech_r(PC)$ — but this fails to clarify two things:

- What relevant information should one expect to find in the singular homology of $D_r(PC)$?

- For which values of r should one investigate the singular homology of $D_r(PC)$?

We try to answer the first question by way of example. Let $PC = (P, \mathbb{E}^2)$, where $\mathbb{E}^2$ is 2-dimensional Euclidean space and P is a specific set of points sampled from a couple of joined annuli. A few of the resulting dilations/complexes are shown in Fig. 2.3.

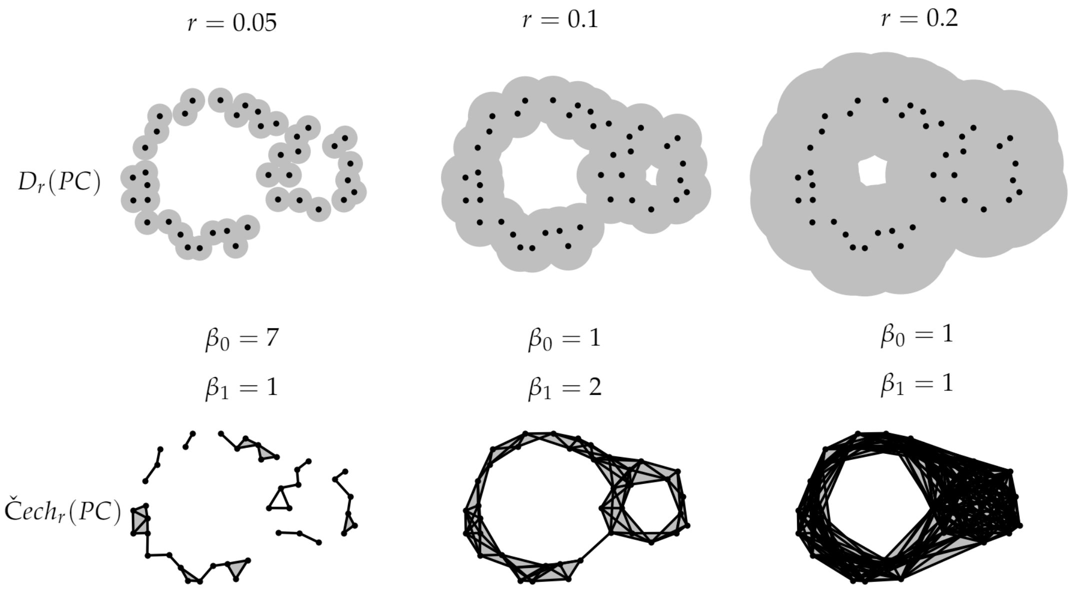


Figure 2.3: Dilations and resulting Čech complexes for different values of r, and the Betti numbers of these complexes

Regarding the second question, Fig. 2.3 might give the impression that the solution is to find some r such that all the relevant features of the shape underlying P are captured. While this is an option, it is not necessarily possible that such an r exists, nor is it obvious how to go about finding it. In truth, there is an interplay in choices of r: too small clearly risks introducing noise for small $\#P$, but too large can completely overlook smaller topological features. The solution, then, is to avoid choosing an r at all. For this, let $PC = (P, \mathcal{M})$ be a point cloud, and notice that for any r, r' from $\mathbb{R}$

$$r \leq r' \implies \check{C}ech_r(PC) \subseteq \check{C}ech_{r'}(PC).$$

An indexed set of simplicial complexes $S_\bullet = \{S_i\}_{i \in I}$ with $\subseteq$ agreeing with the order on I (i.e., ordered by inclusion: the index set will always be assumed to be equipped with some total order $<$) is called a **filtration**. Defining a notion of homology for filtrations is the subject of the following section.

2.2.4 Persistent homology

Now, let $S_\bullet = \{S_i\}_{i \in I}$ be a filtration. It follows that there is an inclusion map $i^{j,j'} : S_j \to S_{j'}$ for any pair $j \leq j'$. Moreover, for any $j \leq j' \leq j''$, $i^{j',j''} \circ i^{j,j'} = i^{j,j''}$, and $i^{j,j} = Id_{S_j}$.

For any abstract simplicial complex S, let S_p denote $\{\sigma \in S \mid dim(\sigma) = p\}$, and let $i_p^{j,j'}$ be the restriction of $i^{j,j'}$ to $(S_j)_p$, which is a map $(S_j)_p \to (S_{j'})_p$. Now, for any index j, let

$$\begin{aligned} h_p^j \colon (S_j)_p &\to H_p(S_j) \\ \sigma &\mapsto [\sigma], \end{aligned}$$

where σ on the RHS is understood as an element of $C_p(S_j)$, and $[\sigma]$ is its class in $H_p(S_j)$. Then, let $hi_p^{j,j'}$ be the linear map such that

$$\begin{tikzcd} (S_j)_p \arrow[r, "i_p^{j,j'}"] \arrow[d, "h_p^j"] & (S_{j'})_p \arrow[d, "h_p^{j'}"] \\ H_p(S_j) \arrow[r, "hi_p^{j,j'}"] & H_p(S_{j'}) \end{tikzcd}$$

commutes. It follows then that for any $j \leq j' \leq j''$, $hi_p^{j',j''} \circ hi_p^{j,j'} = hi_p^{j,j''}$, and $hi_p^{j,j} = Id_{H_p(S_j)}$. Therefore, the pair

$$(\{H_p(S_j)\}_{j>0}, \{hi_p^{j,j'}\}_{j'>j>0})$$

forms a **persistence module** — namely, the ***p*-th persistent homology** of S. Intuitively, the existence of $hi_p^{j,j'}$ lets one associate a p-dimensional hole in S_j to some other hole in $S_{j'}$, in effect having this hole "persist" from the index j to j'. With this we can extend the definition of Betti numbers to filtrations — namely, the ***p*-th (i,j)-persistent Betti number** of S is $\beta_p^{i,j}(S) := \text{rank}(hi_p^{i,j})$, and it counts the amount of holes that persist from index i to index j. Indeed, this is an extension of Betti numbers, as $\beta_p^{i,i}(S) = \beta_p(S_i)$. Note that, unlike with the p-th persistent homology of S, one can no longer recover elements of S from this definition.

Recall that we assume that I comes equipped with an order simply written as $\leq$. Let $\Delta_I^+ = \{(i,j) \in I^2 \mid i \leq j\}$. We define a (partial) order on Δ_I^X: $(i,j) \leq (i',j') \iff i < i' \wedge j' < j$. Clearly, $(i,j) \leq (i',j') \implies \beta_p^{i,j} \leq \beta_p^{i',j'}$. Moreover, if both S and I are finite, then for each $p \in \mathbb{N}$ there is an uniquely defined multiset $PD_p(S)$ such that for all $i,j \in \Delta_I^X$

$$\beta_p^{i,j} = \#\{(i',j') \in PD_p(S) \mid (i',j') \leq (i,j)\}.$$

The multiset $PD_p(S)$ is the ***p*-th persistence diagram** of S, and the (i,j) in $PD_p(S)$ are **birth-death pairs** or **persistence pairs**. The existence and uniqueness of this follows from the main result in [29, Section 3].

Notice that in particular the requirement that I, the set indexing S, be finite. This clashes with how $\check{C}ech_r(PC)$ (for some point cloud $PC = (P, \mathcal{M})$, with $\#P < \infty$) is defined for every $r \in \mathbb{R}_{>0}$, which suggests the use of $\mathbb{R}_{>0}$ as an index set. However, since $\check{C}ech_r(PC)$ can only change at a finite amount of different values of r (as the $\check{C}ech_r(PC)$ are ordered by inclusion and contained in a finite set), one can simply restrict the values of r to this finite set, together with $r = \infty$, and lose no information in the process. We will assume all indexing sets I are of this form, i.e., $I \subseteq \mathbb{R}_{>0} \cup \{\infty\}$ and $\#I < \infty$. This ensures that for any persistence diagram PD, any $(b, d) \in PD$ can be regarded as an pair in $\mathbb{R}^2$ — except for some points in $(\mathbb{R} \cup \{\infty\})^2$, which in general will be considered separately.

2.2.5 Other filtrations

An alternative to the Čech complex is the *Vietoris-Rips complex*. Let $PC = (P, \mathcal{M})$ be a point cloud and $r \in \mathbb{R}_{>0}$. Then, the r-radius **Vietoris-Rips complex** of PC is the abstract simplicial complex

$$VR_r(PC) = \{\sigma \subseteq P \mid \forall v, u \in \sigma,\ \ d_{\mathcal{M}}(v, u) < r\}.$$

There are a few things to note about this definition:

- It does not depend on any part of $\mathcal{M}$ other than the pairwise distances between elements of P — a distance matrix is enough to determine it.
- In fact, this can be refined to simply the knowledge of what pairwise distances are smaller than r, meaning $VR_r(PC)$ is completely determined by the simplices of dimension ≤ 1 in $VR_r(PC)$.
- The condition imposed on σ is much easier to check than its counterpart for Čech complexes — for this reason, it is significantly easier to compute the persistent homology of Vietoris-Rips filtrations than that of Čech filtrations.

In short, VR complexes are simpler and faster to treat computationally, which is why they are the complexes we have used in practice. However, it is important to note that they can be significantly different from Čech complexes, and moreover, there is no result analogous to a nerve lemma for VR complexes — though more geometrically minded requirements for topological accuracy do exist, such as in [2]. Nonetheless, VR and Čech complexes are nowhere near unrelated. For instance, it is easy to see that

$$\check{C}ech_r(PC) \subseteq VR_r(PC) \subseteq \check{C}ech_{2r}(PC).$$

In fact, a tighter bound for the latter inclusion can be obtained through geometric reasoning when $\mathcal{M} = \mathbb{E}^d$ ([26, Theorem 2.5]).

Chapter 3

Metrization of a Neural Network

In this chapter, we intend to extract a (finite) metric space that in some sense encodes the behaviour of a neural network $G = (S, w, b)$. The elements of this metric space could correspond to elements of S_V, or combinations of these, or subsections thereof. The distances between them will in some way encode how much the firing patterns of these elements have to do with each other. Our particular approach closely follows the ideas laid out in [7]. This yields a very specific definition for our metric space, though this in turn makes a number of results about this metric space readily available. In this chapter we will go through this definition and some of the results.

3.1 Ciprian spaces

3.1.1 Indices

Recall that, for a tuple of integers $\mathbf{x} = (x_1, ..., x_n)$, $[1...\mathbf{x}]$ denotes the set of integer tuples in $[1...x_1] \times \cdots \times [1...x_n]$. Note that for some tensor T of shape s, the set $[1...s]$ is precisely the set of valid indices referring to entries of T, with this association being denoted with contrapositive indices: T^i, $i \in [1...shape(T)]$. Then,

Definition 3.1. *The* ***index set*** *of a computation graph schema S, written $\mathbb{I}_S$, is the set $\{(v, i) \mid v \in S_V, i \in [1...shape(S_{r_v})]\}$.*

For a (not necessarily differentiably parametrized) computation graph G, $\mathbb{I}_G$ may be used to refer to the index set of the underlying schema. These index sets will index the vertices in the metric space.

3.1.2 Elements

Let $G = (S, w, b)$ be a computation graph and $x \in \mathcal{I}_G$. Then, $\{\left(P_G(x)\right)_v^i \mid (v, i) \in \mathbb{I}_G\}$ is precisely the set of all entries in tensors in the computation graph produced from the input x. For brevity, with $j = (v, i) \in \mathbb{I}_G$, we will write $P_G(x)_j$ instead of $P_G(x)_v^i$.

We will be interested in how $\big(P_G(x)\big)_i$ varies with respect to $x \in \mathcal{I}_G$, for some fixed $i \in \mathbb{I}_G$, and then comparing the joint behaviour of $\big(P_G(x)\big)_i$ and $\big(P_G(x)\big)_j$ for $i \neq j$ to obtain the distance between i and j.

To be precise about what is meant by "behaviour", however, one needs a dataset. As is common practice, a dataset will be modeled as a probability distribution $\mathcal{D}$ on $\mathcal{I}_G$. This requires a measure structure on $\mathcal{I}_G$ — but $\mathcal{I}_G$ is simply the product of a finite amount of real tensor spaces, so it can easily be endowed with the standard Lesbegue measure on $\mathbb{R}^n$ for some probably pretty large n. Alternatively, a measure with finite support might be preferable, as this allows for modelling actual finite datasets.

Then, one can take samples from $\mathcal{D}$ — these are random vectors, if they are flattened. With this in mind:

Definition 3.2. *The **activation distribution** of an entry $i \in \mathbb{I}_G$ of a computation graph G with respect to a dataset $\mathcal{D}$ is the random variable $P_G(D)_i$, where D is a sample from $\mathcal{D}$.*

This being a random variable or not depends on $x \mapsto P_G(x)_i$ being a measurable function. It is not difficult to show that this must be the case if all the transition functions in G are measurable, which we will assume to be the case. Moreover, we will assume that $\mathcal{D}$ has a bounded support, as this ensures that all $P_G(X)_i$ have a finite expectation.

The set of vertices in the metric space, then, will be $V_G(\mathcal{D}) := \{P_G(D)_i\}_{i \in \mathbb{I}_G}$. We will also be interested in finite samples of D used to approximate these distributions. Thus, let $n \in \mathbb{N}$ and $D_1, ..., D_n$ be i.i.d. random variables following the distribution of $\mathcal{D}$ — then, $V^n_G(\mathcal{D}) := \{(P_G(D_1)_i, ..., P_G(D_n)_i\}_{i \in \mathbb{I}_G}$.

3.1.3 Correlation

A cumbersome limitation of computers (or the real world, for that matter) is that they rarely ever allow infinite amounts of anything. This is particularly relevant to our case, as the most principled way of defining the metric space we are interested in involves distributions lacking a finite support — but of course only finite approximations of these are computable. It will be important to understand to what extent this gap can be bridged. Because of this, this subsection will need to draw attention to the (common) distinction between sample estimators and statistics. Starting with sample estimators:

The **sample mean** of an $\mathbb{R}$-tuple $v = (v_1, ..., v_n)$ is $\overline{v} = \frac{1}{n} \sum_{i=0,...,n} v_i$.

The **sample variance** of an $\mathbb{R}$-tuple $v = (v_1, ..., v_n)$ is $var^s(v) = \frac{1}{n} \sum_{i=0,...,n} (v_i - \overline{v})^2$.

The **sample correlation** of a pair of $\mathbb{R}$-tuples, $v = (v_1, ..., v_n)$, $u = (u_1, ..., u_n)$ is

$$corr^S(v, u) = \frac{1}{n} \frac{\sum_{i=0,...,n} (v_i - \overline{v})(u_i - \overline{u})}{\sqrt{var^s(v) \cdot var^s(u)}}.$$

Note that *biased* sample variance is used. Substituting this for the unbiased version, however, is completely possible and does not have a lot of effect on its usage later on. Continuing, we have the continuous versions:

The **mean** of a random variable X with underlying probability space $\mathcal{P}$ is $\mathbb{E}_{\mathcal{P}}(X) = \int_{\mathcal{P}} X$.

The **variance** of a random variable X as before is $var_{\mathcal{P}}(X) = \mathbb{E}_{\mathcal{P}}\big((X - \mathbb{E}_{\mathcal{P}}(X))^2\big)$.

The **correlation** of two random variables X, Y with underlying probability space $\mathcal{P}$ is

$$corr_{\mathcal{P}}(X,Y) = \mathbb{E}_{\mathcal{P}}\left(\frac{(X - \mathbb{E}_{\mathcal{P}}(X))(Y - \mathbb{E}_{\mathcal{P}}(Y))}{\sqrt{var_{\mathcal{P}}(X) \cdot var_{\mathcal{P}}(Y)}}\right).$$

All these integrals (and thus, the statistics themselves) depend on the probability spaces underlying X and Y. $\mathbb{E}$, *var* and *corr* may be used without subscript if the underlying probability space is clear from context. Note also that correlation here is defined as the standard Pearson correlation coefficient.

In practice, correlation (sample and continuous) is the only estimator we are interested in, as it is what will be used as a starting point for creating distance functions. We will now connect these two:

Proposition 3.3. *Let X and Y be random variables with the same underlying probability space $\mathcal{P}$, both with non-null variance, and $\{X_i\}_{i\in\mathbb{N}}$, $\{Y_i\}_{i\in\mathbb{N}}$ be sequences of i.i.d. random variables with the same distributions as X and Y, respectively (i.e., samplings from X and Y's distributions). Then,*

$$corr^S\big((X_1, ..., X_i), (Y_1, ..., Y_i)\big) \longrightarrow corr(X,Y).$$

Here, $\longrightarrow$ denotes convergence of a sequence of random variables to a constant (in distribution or probability, as they are equivalent in this case).

Proof. Algebraic manipulations on the expressions of *corr* yields

$$corr(X,Y) = \frac{\mathbb{E}(X \cdot Y) - \mathbb{E}(X) \cdot \mathbb{E}(Y)}{\sqrt{\mathbb{E}(X \cdot X) - \mathbb{E}(X) \cdot \mathbb{E}(X)}\sqrt{\mathbb{E}(Y \cdot Y) - \mathbb{E}(Y) \cdot \mathbb{E}(Y)}}.$$

In particular, the factors of the denominator correspond to $\sqrt{var(X)}$ and $\sqrt{var(Y)}$. Likewise for $corr^S$:

$$corr^S(x,y) = \frac{\overline{x \cdot y} - \overline{x} \cdot \overline{y}}{\sqrt{\overline{x \cdot x} - \overline{x} \cdot \overline{x}}\sqrt{\overline{y \cdot y} - \overline{y} \cdot \overline{y}}}$$

This expression of $corr^S(x,y)$ can be interpreted as a continuous map of $(\overline{x}, \overline{y}, \overline{x \cdot x}, \overline{y \cdot y}, \overline{x \cdot y})$, as long as the denominator (the product of root variances) is nonzero. Moreover, if this map evaluated at $(\mathbb{E}(X), \mathbb{E}(Y), \mathbb{E}(X^2), \mathbb{E}(Y^2), \mathbb{E}(XY))$, this yields the lower expression of $corr(X,Y)$.

Now, let $\overline{X}_i = \overline{(X_1, ..., X_i)}$, and likewise for $\overline{Y}_i$, $\overline{X^2}_i$, $\overline{XY}_i$ and $\overline{Y^2}_i$. It follows that the vector $(\overline{X}_i, \overline{Y}_i, \overline{X^2}_i, \overline{XY}_i, \overline{Y^2}_i)$ converges[1] to $(\mathbb{E}(X), \mathbb{E}(Y), \mathbb{E}(X^2), \mathbb{E}(Y^2), \mathbb{E}(XY))$, and that the denominator of $corr(X, Y)$ is non-zero, as the variances being nonzero is part of the hypothesis. Therefore, by the continuous mapping theorem:

$$corr^S\big((X_1, ..., X_i), (Y_1, ..., Y_i)\big) \longrightarrow corr(X, Y). \qquad \square$$

3.1.4 Distances

Let G be a computation graph, $\mathcal{D}$ a dataset in $\mathcal{I}_G$, and D a random sample from this dataset. Recall that the objects of our metric space are random variables $P_G(D)_i$, indexed by $i \in \mathbb{I}_G$. Then, the distance function used in [7] is

$$d_{PD}(X, Y) = 1 - |corr(X, Y)|.$$

Strictly speaking, this is closer to the statistical notion of a *similarity*, as it does not obey the triangle inequality. However, it is in some sense equivalent to some of the other distances presented — this will be discussed in Section 3.4. Moreover, all the distance functions presented that do obey the triangle inequality are actually **pseudometrics** — meaning that $d(x, y) = 0 \not\Rightarrow x = y$. The implications of this will be discussed in Section 3.5.1.

First of all, we introduce two pseudometric alternatives to d_{PD}:

$$d_{PS}(X, Y) = arccos(|corr(X, Y)|),$$

$$d_{PE}(X, Y) = \sqrt{2\Big(1 - |corr(X, Y)|\Big)}.$$

These three distances, d_{PD}, d_{PS}, and d_{PE}, will be described as **projective**. This refers to the fact that if $corr(X, Y) = -1$, then $d_{PD}(X, Y) = d_{PS}(X, Y) = d_{PE}(X, Y) = 0$. This is the same ranking that they would give $corr(X, Y) = 1$ — this is fairly transparent from the usage of $|\cdot|$ in the definitions above. One can also gleam, then, that the maximum distance between X and Y is attained only when $corr(X, Y) = 0$. In contrast, in the next three counterparts to d_{PD}, d_{PS}, and d_{PE}, the furthest distance is attained when $corr(X, Y) = -1$:

$$d_D(X, Y) = 1 - corr(X, Y),$$

$$d_S(X, Y) = arccos(corr(X, Y)),$$

$$d_E(X, Y) = \sqrt{2\Big(1 - corr(X, Y)\Big)}.$$

[1] Convergence of each coordinate is due to the law of large numbers. In this case, this also implies convergence as a random vector, as each of the components converges to a constant.

Note that these distances are undefined if X or Y have null variance. We will assume that this is not the case for any $X \in V_G(\mathcal{D})$. The reliability and practical implications of this will be discussed in Section 3.5.2. In any case, we have four pseudometric spaces, namely

$$\mathcal{C}^G_{PS} := (V_G(\mathcal{D}), d_{PS}), \quad \mathcal{C}^G_{PE} := (V_G(\mathcal{D}), d_{PE}), \quad \mathcal{C}^G_{S} := (V_G(\mathcal{D}), d_{S}), \quad \mathcal{C}^G_{E} := (V_G(\mathcal{D}), d_{E}).$$

Definition 3.4. *A* ***Ciprian space*** *of a computation graph G with respect to a dataset $\mathcal{D}$ is either one of the four spaces defined above, or $\mathcal{C}^G_D := (V_G(\mathcal{D}), d_D)$ and $\mathcal{C}^G_{PD} := (V_G(\mathcal{D}), d_{PD})$.*

The next few sections will give some basic fundamental results regarding these spaces.

3.2 Pseudometricity

To prove that the spaces we just defined are indeed metric, it will be necessary to introduce an "approximate" form of them. To be precise: let G be a computation graph, and $\mathcal{D}$ a dataset over $\mathcal{I}_G$.

Definition 3.5. *The* ***sample Ciprian space*** *with $n \in \mathbb{N}$ samples of a Ciprian space $\mathcal{C} = (V_G(\mathcal{D}), d)$ is the metric space, denoted $(\mathcal{C})^n$, with vertex set $V^n_G(\mathcal{D})$, and with the metric denoted $(d)^n$, obtained by substituting corr by $corr^S$ in d.*

The $(\mathcal{C})^n$, then, serve as approximations of $\mathcal{C}$. In this section, we will show that $(\mathcal{C}^G_S)^n$ and $(\mathcal{C}^G_E)^n$ are pseudometric spaces for any $n \in \mathbb{N}$. In Section 3.3, we will explain that the $(\mathcal{C})^n$ converge, in some sense, to $\mathcal{C}$, and why this implies that $\mathcal{C}$ is also a pseudometric space.

Importantly, $(\mathcal{C})^n$ is defined as a random element, but for most of this section we will talk about it as if though it is a particular value of this random element. We do not make the distinction explicitly, as this is notationally cumbersome — it is easy to intuit when $(\mathcal{C})^n$ is regarded as a probabilistic object (namely, in Section 3.3).

The proofs in this section are, for the most part, obtained through straightforward algebraic manipulation. The less interesting ones will be omitted.

An **affine transform** of linear term $a \in \mathbb{R}$ and constant term $b \in \mathbb{R}$ of a vector $x = (x_1, ..., x_n)$ is $ax + b := (ax_1 + b, ..., ax_n + b)$. Notice that this disagrees with the usual meaning of affine transform, where a is allowed to be a matrix and b a vector. We say an affine transform is **strictly positive** and **non-null** when $a > 0$ and $a \neq 0$, respectively.

Now, let $x \in \mathbb{R}^n$, and $ax + b$ be an affine transform of x. It is easy to check that

$$\overline{ax + b} = a\overline{x} + b, \quad var^S(ax + b) = |a|\, var^S(x). \tag{3.1}$$

Similarly, if x and y are two vectors, and $ax + b$ is a non-null affine transform of x, then $corr^S(ax + b, y) = sgn(a) \cdot corr^S(x, y)$. It is easy to see that correlation is symmetric, meaning that this can apply to both of the terms in $corr^S(x, y)$ — at once.

We say that a vector x is **normal** if it satisfies $\overline{x} = 0$, $var^S(x) = 1$. The **normalized form** of a vector x with non-null variance is $\tilde{x} := (x - \overline{x})/var^S(x)$. Notice that the normalized form of x is a strictly positive affine transform of x. With (3.1), the next few results are straightforward to obtain:

- For any vector x with non-null variance, $\tilde{x}$ is a normal vector.
- Let x and y be two normal vectors from $\mathbb{R}^n$. Then, $corr^S(x,y) = \langle x,y\rangle/n$.
- Let $x \in \mathbb{R}^n$ with $\overline{x} = 0$. Then, $var^S(x) = \sqrt{n}||x||_2$.
- Let $x \in \mathbb{R}^n$ be a normal vector. Then, $||x||_2 = \sqrt{n}$.
- Let x be any vector with non-null variance. Then, $corr^S(x,x) = 1$.

Proposition 3.6. *Let x and y be two vectors from $\mathbb{R}^n$ with non-null variance. Then, the following four statements are equivalent:*

i) $corr^S(x,y) = 1$.

ii) $\forall z \in \mathbb{R}^n$, $corr^S(x,z) = corr^S(y,z)$.

iii) $\exists a, b \in \mathbb{R} \mid y = ax + b$, *and* $ax + b$ *is a strictly positive affine transform of* x.

iv) $\tilde{x} = \tilde{y}$.

Proof. The following five implications are enough:

$(\mathbf{iv}) \implies (\mathbf{iii})$:

$\tilde{x} = \tilde{y} \implies \frac{x-\overline{x}}{Var^S(x)} = \frac{y-\overline{y}}{Var^S(y)} \implies x = \frac{Var^S(x)}{Var^S(y)}y + \left(\overline{x} - \frac{Var^S(x)}{Var^S(y)}\overline{y}\right)$.

Moreover, $Var^S(x) > 0 \wedge Var^S(y) > 0 \implies \frac{Var^S(x)}{Var^S(y)} > 0$.

$(\mathbf{iii}) \implies (\mathbf{ii})$:

With a, b as in **(iii)**, $\forall z \in \mathbb{R}^n, corr^S(y,z) = corr^S(ax+b,z) = corr^S(x,z)$.

$(\mathbf{iii}) \implies (\mathbf{i})$:

With a, b as in **(iii)**, $corr^S(x,y) = corr^S(x, ax+b) = corr^S(x,x) = 1$.

$(\mathbf{i}) \implies (\mathbf{iv})$:

$corr^S(x,y) = 1 \implies corr^S(\tilde{x},\tilde{y}) = 1 \implies \langle \tilde{x},\tilde{y}\rangle = n = \sqrt{n}\sqrt{n} = ||\tilde{x}||_2||\tilde{y}||_2$.

This is the equality case of the Cauchy-Schwarz inequality, meaning that $\tilde{x}$ and $\tilde{y}$ are linearly dependent. As they have the same modulus, this means that $\tilde{x} = \tilde{y}$.

$(\mathbf{ii}) \implies (\mathbf{iv})$:

$$\begin{aligned}\forall z \in \mathbb{R}^n, corr^S(x,z) = corr^S(y,z) &\implies \forall z \in \mathbb{R}^n, corr^S(\tilde{x},z) = corr^S(\tilde{y},z)\\ &\implies \forall z \in \mathbb{R}^n, \langle \tilde{x},z\rangle/n = \langle \tilde{y},z\rangle/n\\ &\implies \forall z \in \mathbb{R}^n, \langle \tilde{x}-\tilde{y},z\rangle = 0\end{aligned}$$

As $\langle\cdot,\cdot\rangle$ is a non-degenerate inner product, we have that $\tilde{x} - \tilde{y} = 0 \implies \tilde{x} = \tilde{y}$. □

Definition 3.7. *Two vectors x, y are **equivalent**, denoted $x \sim y$, if they satisfy any (all) of the conditions in Proposition 3.6.*

We will now fix an $n \in \mathbb{N}$ such that all vectors from here on out are from the same $\mathbb{R}^n$. Define:

$$\mathbb{R}^n_* := \{x \in \mathbb{R}^n \mid Var^S(x) \neq 0\}$$

$$\mathbb{H}' := \{x \in \mathbb{R}^n \mid x \text{ is a normal vector}\}$$

$$\mathbb{H} := \mathbb{H}'/\sqrt{n}$$

It is obvious from Proposition 3.6.iv that $\sim$ is an equivalence relation on $\mathbb{R}^n_*$, particularly one where $\mathbb{H}'$ is a suitable set of representatives. It is easy to check that $\mathbb{H}$ also fulfills this role (consider the $\mathbb{H} \to \mathbb{H}'$ map defined by $x \mapsto x\sqrt{n}$). Moreover, note that for any x, y from $\mathbb{H}$:

$$corr^S(x, y) = \langle \tilde{x}, \tilde{y} \rangle / n = \langle x\sqrt{n}, y\sqrt{n} \rangle / n = \langle x, y \rangle.$$

Therefore, from here on out we will only deal with vectors $x \in \mathbb{H}$, with the understanding that they represent their equivalence class $[x] = \{y \in \mathbb{R}^n_* \mid \tilde{y} = x\sqrt{n}\}$. Since we will only deal with these vectors x through their correlation with other vectors, Proposition 3.6.ii ensures that it will be well-defined to always assume that $corr^S(x, y) = \langle x, y \rangle$.

Now, elements in $\mathbb{H}$ are those satisfying that $\overline{x} = 0$ and $Var^S(x) = 1/\sqrt{n}$. This first condition is just a linear equation — i.e., $\mathbb{H} \subseteq (1, ..., 1)^\perp$, and as $\overline{x} = 0$, the second condition can be read as $||x||_2 = 1$. It follows that $\mathbb{H}$ is isometric to S^{n-2}, an $(n-2)$-dimensional unit-radius hypersphere.

We are now equipped to show the pseudometricity of $(\mathcal{C}^G_E)^n$ and $(\mathcal{C}^G_S)^n$. Note, however, that these proofs may read as if they were written backwards, as indeed they were — here we show that a distance function on a set indexed by $\mathbb{I}_G$ must be a pseudometric, as it is indeed a pseudometric when included into the larger space $\mathbb{R}^n_*$, but, unsurprisingly, the actual definition of these metrics came about by noticing the structure of $\mathbb{H}$ and its relationship with correlation, defining two obvious metrics on $\mathbb{H}$, and "restricting" these to $\mathbb{I}_G$. Now then:

The n-sample Euclidean Ciprian space, $(\mathcal{C}^G_E)^n$:

First of all, define:

$$\begin{aligned} d:\quad & \mathbb{R}^n_* \times \mathbb{R}^n_* \to \mathbb{R} \\ & (x, y) \mapsto \sqrt{2\Big(1 - corr^S(x, y)\Big)} \end{aligned}$$

Notice then that $(d_E)^n$ is simply a restriction of d. It follows that we only need to show that d is a pseudometric on $\mathbb{R}^n_*$, and therefore it will be sufficient to show that it is a metric on $\mathbb{H}$. Now, for any x, y from $\mathbb{H}$:

$$\begin{aligned} d(x,y) &= \sqrt{2\Big(1 - corr^S(x,y)\Big)} \\ &= \sqrt{2\Big(1 - \langle x,y\rangle\Big)} \\ &= \sqrt{\langle x,x\rangle - 2\langle x,y\rangle + \langle y,y\rangle} \\ &= \sqrt{\langle x-y, x-y\rangle} \\ &= ||x-y||_2. \end{aligned}$$

This is a metric on $\mathbb{R}^n$, and thus also on $\mathbb{H} \subseteq \mathbb{R}^n$. It follows, then, that d is a pseudometric on $\mathbb{R}^n_*$, and therefore $(\mathcal{C}^G_E)^n$ is a pseudometric space.

The n-sample spherical Ciprian space, $(\mathcal{C}^G_S)^n$:
First of all, define:

$$\begin{aligned} d: \quad & \mathbb{R}^n_* \times \mathbb{R}^n_* \to \mathbb{R} \\ & (x,y) \mapsto arccos(corr^S(x,y)) \end{aligned}$$

Exactly like in the prior case, we only need to show that d is a metric on $\mathbb{H}$. Now, let $\alpha(x,y)$ be the angle between two vectors x and y. It is well known that if $||x||_2 = ||y||_2 = 1$, then $\langle x,y\rangle = cos(\alpha(x,y))$. Then, $\forall (x,y) \in \mathbb{H} \times \mathbb{H}$, it is easy to see that $d(x,y) = \alpha(x,y)$. This is the orthodromic metric, and by the same chain of implications as before it follows that $(\mathcal{C}^G_S)^n$ is a pseudometric space.

Showing that this orthodromic metric d is indeed a metric on $\mathbb{H}$ is tantamount to showing that a certain manifold (say, $\mathbb{H}$) is geodesically complete. As $\mathbb{H}$ is isometric to S^{n-2}, we will use the latter. We will sketch a proof of geodesic completeness, as it will be useful later on (notice that the metric underlying $(d_E)^n$ also is a geodesic metric — particularly, that from flat Euclidean space):

1) Show that all geodesics are great circle segments. One can limit oneself to arclength-parametrized geodesics, which must then be of the form $t \mapsto x\cos(t) + y\sin(t)$, with x and y orthogonal unit vectors. This is easy using the local uniqueness (and existence) theorem for geodesics, as all geodesics in S^{n-2} are easy to find given only a starting point and direction.

2) Show that the exponential map is well-defined. Like the previous point, this is easy provided that geodesics are easy to write out explicitly. Then, the Hopf-Rinow theorem ensures that S^{n-2} is a complete geodesic space.

3) Show that the distances in this complete geodesic space actually correspond to the d we defined previously. For this, simply enumerate all geodesics from x to y, find the shortest one, and express its arclength in terms of $\langle x,y\rangle$.

This leaves $(\mathcal{C}^G_{PS})^n$ and $(\mathcal{C}^G_{PE})^n$. The pseudometricity of these will not be proved, as it is largely a repetition of the proofs for $(\mathcal{C}^G_S)^n$ and $(\mathcal{C}^G_E)^n$, with small changes. Namely, $corr^S$ becomes $|corr^S|$ and positive affine transforms become non-null affine transforms. The last section is slightly trickier: Before, the "underlying metrics" to $(d_S)^n$ and $(d_E)^n$ were

$$d_S(x,y) = ||x-y||_2 \quad d_E(x,y) = \alpha(x,y).$$

For $(d_{PS})^n$ and $(d_{PE})^n$, they become:

$$d_{PS}(x,y) = min(||x-y||_2, ||x+y||_2) \quad d_{PE}(x,y) = min(\alpha(x,y), \alpha(x,-y)).$$

Here, the triangle inequality is less obvious. Perhaps the most straightforward way to go about proving it is noticing that:

$$d_{PS}(x,y) = min_{x'=\pm x, y'=\pm y}(d_S(x',y'))$$
$$d_{PE}(x,y) = min_{x'=\pm x, y'=\pm y}(d_E(x',y'))$$

This is the quotient pseudometric of d_S and d_E with respect to the equivalence relation $x \sim -x$, which indeed must always be a pseudometric. This also explains why d_{PS}, d_{PE} and d_{PD} are described as "projective", as $x \sim -x$ and $x \sim \lambda x$ are the same relation when considered on S^m.

3.3 Convergence

Let G be a computation graph, and $\mathcal{D}$ a dataset over $\mathcal{I}_G$. Let D be a random variable following the distribution of $\mathcal{D}$, and $D_{n n\in\mathbb{N}}$ also be i.i.d. samples from $\mathcal{D}$. Now, for any $i \in \mathbb{I}_G$, $X_i = P_G(D)_i$, and $x_i^n = (P_G(D_1)_i, ..., P_G(D_n)_i)$. The main result from Section 3.1.3 tells us that:

$$corr^S(x_i^n, x_j^n) \longrightarrow corr(X_i, X_j) \quad \text{as } n \to \infty,$$

where $\longrightarrow$ represents convergence in probability. As all of the pseudometrics introduced so far are continuous functions of $corr(X_i, X_j)$, it follows from the continuous mapping theorem that

$$\forall d \in \{d_D, d_S, d_E, d_{PD}, d_{PS}, d_{PE}\} \quad (d)^n(i,j) \longrightarrow d(i,j) \quad \text{as } n \longrightarrow \infty.$$

So clearly there is some sense in which $(\mathcal{C})^n \longrightarrow \mathcal{C}$. We will need some definitions to clarify the notion of convergence in metric spaces.

The **directed Hausdorff distance** of two sets X, Y living in a metric space (M, d) is $d'_H(X,Y) = \sup_{x\in X} \inf_{y \in Y} d(x,y)$. The **Hausdorff distance** of the same two sets is

$$d_H(X,Y) = \max(d'_H(X,Y), d'_H(Y,X)).$$

The **Gromov-Hausdorff distance** of two metric spaces $X = (M_X, d_X)$, $Y = (M_Y, d_Y)$, is

$d_{GH}(X,Y) = inf(\{d_H(f_X(X), f_Y(Y))\})$, where the infimum is taken over f_X and f_Y being isometric embeddings of X and Y into some common metric space (M, D).

To be precise, the Gromov-Hausdorff distance is only a metric on compact metric spaces modulo isometry (put another way, it is also a pseudometric). Moreover, it is in the same sense well-defined for pseudometrics, which is convenient in our case. In general, the definition is not very treatable, but there are many simpler phrasings for finite X and Y. For example, a consequence of [6, Theorem 7.3.25] is that if $\#M_X = \#M_Y$,

$$d_{GH}(X,Y) = \frac{1}{2} \inf_{p \in \mathcal{B}(M_X, M_Y)} \Big(\sup_{(a,b) \in M_X \times M_X} \big| d_X(a,b) - d_Y(p(a), p(b)) \big| \Big),$$

where $\mathcal{B}(A,B)$ is the set of bijections between A and B.

The statement in [6] is quite a bit stronger — it treats (a class of) relations instead of bijections, and thus can be used for X and Y with $\#M_X \neq \#M_Y$. In our case, however, $\mathcal{C}$ and $(\mathcal{C})^n$ are defined on the same set. We can simply take p to be the identity to obtain an upper bound on the distance:

$$d_{GH}(\mathcal{C}, (\mathcal{C})^n) \leq \sup_{i,j \in \mathbb{I}_G \times \mathbb{I}_G} \big| d(i,j) - (d)^n(i,j) \big|.$$

There is no reason to believe that this prior inequality is not actually an equality, but it is not troublesome to leave it as $\leq$. In any case, since $\mathbb{I}_G \times \mathbb{I}_G$ is finite and $(d)^n(i,j) \longrightarrow d(i,j)$, we have that $d_{GH}(\mathcal{C}, (\mathcal{C})^n)$ converges in probability to 0, and thus $(\mathcal{C})^n$ converges in probability in the Gromov-Hausdorff sense to $\mathcal{C}$. Moreover, the completeness of the Gromov-Hausdorff metric ([24, Proposition 11.1.8]) implies that $\mathcal{C}$, being the limit of $(\mathcal{C})^n$, is itself a pseudometric space[2].

Note that it is not immediately clear that this notion of probabilistic convergence in the Gromov-Hausdorff sense is well defined. We give a brief clarification: For the restriction of d_{GH} spaces below some specific cardinality, these spaces can be understood as elements of a finite-dimensional vector space (the strict upper distance matrices), i.e., elements of a subset of some $\mathbb{R}^d$. This subset of $\mathbb{R}^d$ is known as the *metric cone*, and since it is defined by a finite set of inequalities (namely, each element being positive and the triangle inequality for every triplet of indices), it is measurable and can be endowed with the subset measure in $\mathbb{R}^d$. By a similar process we can restrict ourselves to a suitable (measurable) set of representatives with respect to isometry. We make no claim about whether similar statements can be made about d_{GH} for general metric spaces.

[2] Notice that it was an abuse of notation to use $\mathcal{C}$ as an argument of d_{GH} before we knew it was a pseudometric space — it would have been more correct to define some $\overline{d_{GH}}$ (for general "distance matrices" that need not obey the triangle inequality) such that d_{GH} is the restriction of $\overline{d_{GH}}$ to proper pseudometric spaces.

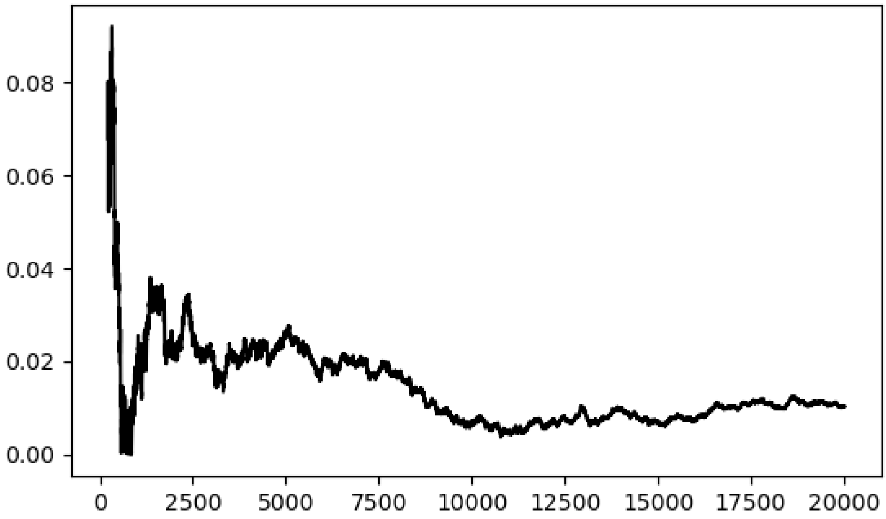


Figure 3.1: The relative error of some specific value of $(d)^n$ with respect to d, as n grows. Notice even for $n = 20000$ there is close to 0.01 error.

3.4 Equivalence

Notice that for any values of x and y, $corr^S(x, y) \in [-1, 1]$, and thus likewise for $corr(x, y)$. This means that

$$\begin{aligned} \operatorname{Im}(d_S) &\subseteq [0, \pi] & \operatorname{Im}(d_{PS}) &\subseteq [0, \pi/2] \\ \operatorname{Im}(d_E) &\subseteq [0, 2] & \operatorname{Im}(d_{PE}) &\subseteq [0, \sqrt{2}] \\ \operatorname{Im}(d_D) &\subseteq [0, 2] & \operatorname{Im}(d_{PD}) &\subseteq [0, 1] \end{aligned}$$

Now, notice that

$$\begin{aligned} d_S &= m_{DS} \circ d_D & m_{DS}(t) &= arccos(1 - t) \\ d_E &= m_{SE} \circ d_S & m_{SE}(t) &= \sqrt{2(1 - cos(t))} \\ d_D &= m_{ED} \circ d_E & m_{ED}(t) &= (t^2/2) \end{aligned}$$

Moreover, m_{DS} is monotonically increasing from $\operatorname{Im}(d_D)$ to $\operatorname{Im}(d_S)$, and similarly for m_{SE} and m_{ED}. Composing these functions, then, can yield a monotonic function relating any pair of distances in $\{d_D, d_S, d_E\}$. For the d's that are metric, this means that they are topologically equivalent, meaning that they generate the same topological space — in fact, treating d_D as a metric and using it for a topological space would also yield this same space.

This topological equivalence is not very relevant to us, though, as topological spaces lose a lot of metric information. Indeed, it is fairly easy to come up with two metric spaces that are topologically equivalent but not "monotonically related", meaning the connection between the distances of $\{d_D, d_S, d_E\}$ is more significant than this. A fairly important consequence of it is that:

Proposition 3.8. *Let $\mathcal{M}$ and $\mathcal{M}'$ be two finite metric spaces over the same vertex set V, with distance functions d and d' satisfying $d' = m \circ d'$ for some monotonic function m. Let $m^2(x, x) = (m(x), m(x))$. Then, for any $p \in \mathbb{N}$,*

$$PD_p(VR(\mathcal{M})) = m^2\Big(PD_p(VR(\mathcal{M}'))\Big).$$

Proof. By the definition of m,

$$VR_r(\mathcal{M}) = VR_{m(r)}(\mathcal{M}').$$

Recall that to compute the persistent homology, we select the finite sets of values of r at which $VR_r(\mathcal{M})$ changes. Let I and I' be these index sets for $VR_\bullet(\mathcal{M})$ and $VR_\bullet(\mathcal{M}')$ — then, the equation above implies that $I' = m(I)$, and, moreover, for each $r \in I$, $VR_r(\mathcal{M}) = VR_{m(r)}(\mathcal{M}')$. This means that $VR(\mathcal{M})$ and $VR(\mathcal{M}')$ are the same filtration, just indexed by a different set — and thus $PD_p(VR(\mathcal{M})) \subseteq I \times I$ will be in bijective correspondence with $PD_p(VR(\mathcal{M}') \subseteq I' \times I'$, in particular by m^2. □

It is easy to obtain an analogous result for Čech filtrations. Importantly, this shows that the persistence diagrams obtained from $\{\mathcal{C}^G_D, \mathcal{C}^G_S, \mathcal{C}^G_E\}$ carry the same information.

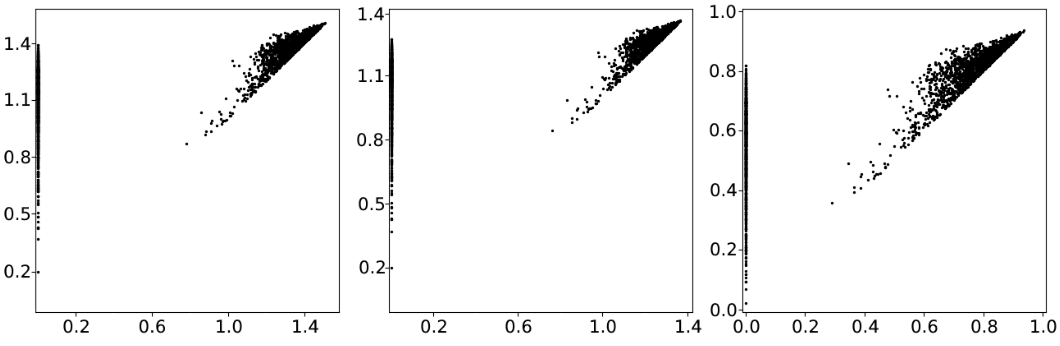


Figure 3.2: Persistence diagrams in three equivalent metrics. Left to right: d_{PS}, d_{PE}, d_{PD}. Note the different scales.

The same three monotonic functions also relate $\{d_{PD}, d_{PS}, d_{PE}\}$ in the same way, meaning the same result also holds.

3.5 Other considerations

3.5.1 Metricity

Strictly speaking, a Ciprian space $\mathcal{C}$ (of a computation graph G with a dataset $\mathcal{D}$) can only be ensured to be pseudometric (as opposed to metric). One can observe that $\mathcal{C}$ *not* being a metric space would require the existence of $i, j \in \mathbb{I}_G^2$ such that $corr(P_G(D)_i, P_G(D)_j) = \pm 1$. While this seems extremely unlikely, it *does* happen in practice — at least to a degree of precision that is indistinguishable from equality (because of numerical error). However, this is indifferent, as VR complexes are perfectly well defined for finite pseudometric spaces — moreover, $VR(\mathcal{M})$ for some pseudometric space $\mathcal{M}$ is homology-equivalent to $VR(\mathcal{M}')$, where $\mathcal{M}'$ is the quotient of $\mathcal{M}'$ with respect to $x \sim y :\equiv d_{\mathcal{M}}(x, y) = 0$.

3.5.2 Constant vectors

As was noted before, basing our metric spaces on correlation makes the assumption that correlation is always well-defined, i.e., that we will never find elements of the space with null variance. In practice this does actually happen a fair amount — the usage of $ReLu$ as an activation:

$$ReLu(x) = max(0, x)$$

makes it so that there is a nonzero possibility that there is an element of $V_G^n(\mathcal{D})$ with all its coordinates set to 0. These elements are simply removed from analysis — this is because in the networks we use, a neuron always being set to 0 is tantamount to this network having no effect on any computation, i.e., the network behaviour would be exactly the same without it. A less black-and-white case can be made for removing non-zero neurons with null variance, though this did not happen in any experiment.

Chapter 4

Estimators of Generalization

For a model M, let $g(M)$ denote the generalization gap of M. Strictly speaking, an **estimator** is simply a function $f(M)$. A *good* estimator is a function that can in some sense approximate $g(M)$ — ideally, we should be interested in estimators satisfying

$$sgn(g(M) - g(M')) = sgn(f(M) - f(M')),$$

i.e., those that can accurately rank models based on generalization gap, not necessarily with knowledge of the exact gap — or equivalently, those f that are a monotonic transform away from g. We will detail how these estimators are ranked in Section 5.

In this chapter, we will go through all of the estimators of generalization that will be tested in Section 5. These include:

- The topological summaries introduced in [7], as well as a number of new summaries based on other representations of persistent homology.

- Some "metric summaries", meaning scalar descriptors of finite metric spaces. These are applied to the Ciprian spaces introduced in Section 3.4 in an attempt to tease out what information is gained through topological analysis, as opposed to readily available after the metrization.

We will also give some basic results on some of these estimators, particularly regarding stability.

Note that these are not all the estimators that *appear* in Section 5 — we also include a number of other known generalization estimators, chosen mostly as benchmarks according to their performance in [16]. In particular, we chose the margin, path norms, various estimators based on spectral norms, and gradient noise.

4.1 Topological estimators

In Section 2.2 we gave the definition of persistence diagram. We will recall an intuitive understanding of this definition and, with persistence diagrams as a starting point, introduce a number of alternate representations of persistent homology. For each of these we will introduce some "summaries" — real-valued functions expected to capture some important aspect of the persistent homology of a filtration. These will later be used for Section 5.

4.1.1 Persistence diagrams

Recall that the p-th persistence diagram of a filtration S is a multiset $PD_p(S)$ of pairs, such that each pair (b,d) represents a p-dimensional hole in S that is born at filtration step b and dies at filtration step d. If the original filtration S or the dimension p are irrelevant, one may just refer to "a persistence diagram PD" without specifying either.

It is common to include points in persistence diagrams with infinite coordinates — for instance, $(0,\infty)$ might represent a connected component born at the index 0 that never dies. We will often assume these infinite points are removed.

Possibly the most common metric used among persistence diagrams is the *Wasserstein distance*. For this, we need a preliminary definition — a **matching** $\sim$ between two sets A and B is a subset of $A \times B$ (where we write $a \sim b$ and say a and b are *matched* iff $(a,b) \in\sim$) such that each $a \in A$ is matched to at most one $b \in B$, and viceversa. Then, $A\backslash \sim$ is the set of $a \in A$ such that there is no $b \in B$ with $a \sim b$, i.e., the *unmatched* elements of A, and likewise for $B\backslash \sim$. We use $A \sim B$ to denote the set of matchings $\sim$ between A and B. We extend this definition to include the possibility of matchings between multisets — this can be done by simply regarding a multiset as a disjoint union of some amount of normal sets with the required copies of each element (i.e., regarding each copy of some element as distinct from the other copies). Then, the **q-Wasserstein distance** of two persistence diagrams PD and PD' is

$$W_q(PD,PD') = \left(min_{PD\sim PD'} \Big(\sum_{x\sim x'} \|x-x'\|_\infty^q + \sum_{x\in(PD\backslash\sim)\cup(PD'\backslash\sim)} ||x-\pi_\Delta(x)||_\infty^q \Big) \right)^{\frac{1}{q}},$$

where $\pi_\Delta(\cdot)$ is orthogonal projection to the diagonal $\{(x,x)\}$. Taking the ∞-norm here is in a sense arbitrary — any other vector p-norm would result in a valid persistence diagram metric, and indeed these are often also described as Wasserstein metrics. We will not be considering this, as it would have little effect on any of the results. In particular, the $q=\infty$ case is known as the **bottleneck distance**. It is worthwhile to note that the Wasserstein metric(s) induce complete and separable metric spaces [20].

As with Section 3.4, we follow in the steps of [7]. Let (b,d) be a persistence pair — then,

$l(b,d) := d - b$ is its **life**[1], and $ml(b,d) := \frac{b+d}{2}$ its **midlife**. Then, the summaries defined therein are as follows:

Definition 4.1. *The* ***average life*** *of a persistence diagram PD is*

$$l(PD) := \frac{1}{\#PD} \sum_{e \in PD} l(e).$$

Definition 4.2. *The* ***average midlife*** *of a persistence diagram PD is*

$$ml(PD) := \frac{1}{\#PD} \sum_{e \in PD} ml(e).$$

More commonly they are used with a threshold on l, i.e.:

$$PD^{\epsilon} := \{e \in PD \mid l(e) \geq \epsilon\} \qquad l_{\epsilon}(PD) := l(PD_{\epsilon}) \qquad ml_{\epsilon}(PD) := ml(PD_{\epsilon}).$$

We note that these are unstable with respect to W_1.

Proof. We divide the estimators in question in two categories.

l, ml) For $\delta > 0$, define $PD_{\delta} = \{(0,1),(0,\delta)\}$, and $PD = \{(0,1)\}$. Note that $W_1(PD, PD_{\epsilon}) \leq \frac{\delta}{2}$, as obtained by taking a $\sim = \{((0,1),(0,1))\}$. However, $l(PD) = 1$ and $ml(PD) = \frac{1}{2}$, but $l(PD_{\delta}) = \frac{1+\delta}{2}$, and $ml(PD_{\delta}) = \frac{1+\delta}{4}$. Taking the limit $\delta \to 0$, then, yields:

- $W_1(PD, PD_{\delta}) \to 0$
- $l(PD) - l(PD_{\delta}) \to \frac{1}{2}$
- $ml(PD) - ml(PD_{\delta}) \to \frac{1}{4}$

This shows that both l and ml are unstable with respect to W_1, as we wanted.

l_{ϵ}, ml_{ϵ}) For any $\delta \in \mathbb{R}$, define PD_{δ} = $PD_{\delta} = \{(0, 1+\epsilon), (0, \delta+\epsilon)\}$. Now force $\epsilon > \delta > 0$, and note that $l_{\epsilon}(PD_{\delta}) = \epsilon + \frac{1+\delta}{2}$, and $ml_{\epsilon}(PD_{\delta}) = \frac{\epsilon}{2} + \frac{1+\delta}{4}$. On the other hand, $l_{\epsilon}(PD_{-\delta}) = \frac{1+\epsilon}{2}$, and $ml_{\epsilon}(PD_{-\delta}) = \frac{1+\epsilon}{4}$. Moreover, $W_1(PD_{\delta}, PD_{-\delta}) \leq 2\delta$. Taking the limit $\delta \to 0$, then, yields:

- $W_1(PD_{\delta}, PD_{-\delta}) \to 0$
- $l_{\epsilon}(PD_{\delta}) - l_{\epsilon}(PD_{-\delta}) \to \frac{1}{2}\epsilon$
- $ml_{\epsilon}(PD_{\delta}) - ml_{\epsilon}(PD_{-\delta}) \to \frac{1}{4}\epsilon$

The $\epsilon = 0$ case corresponds to l, ml. Then, assuming that $\epsilon > 0$, this shows that both l_{ϵ} and ml_{ϵ} are unstable with respect to W_1. □

Note that these proofs do not depend on taking persistence pairs with arbitrarily high life — meaning that the results hold even in settings where the life of persistence pairs is bounded (e.g., the PDs are from Čech filtrations in some metric space with finite diameter). However, we note that l is in some sense close to stability. Let

[1] This is often referred to as *persistence*.

$$l^*(PD) = \sum_{e \in PD} l(e).$$

Then l^*, which is just $(\#\cdot)l(\cdot)$, is stable with respect to W_1.

Proof. We will treat persistence pairs as elements of $\mathbb{E}^2$, as a vector space. In particular, note that $l(e) = \langle(-1,1), e\rangle$, where $\langle\cdot,\cdot\rangle$ is the standard $\mathbb{E}^2$ product. Then, let PD, PD' be two persistence diagrams, and $\sim$ be the matching between PD and PD' that minimizes the expression in W_1. Then,

$$\begin{aligned}
l^*(PD) - l^*(PD') &= l^*(PD) - l^*(PD') \\
&= \sum_{e \in PD} l(e) - \sum_{e \in PD'} l(e) \\
&= \Big[\sum_{e \sim e'} l(e) - l(e')\Big] + \Big[\sum_{e \in (PD \setminus \sim)} l(e)\Big] - \Big[\sum_{e \in (PD' \setminus \sim)} l(e)\Big] \\
&\overset{*}{=} \Big[\sum_{e \sim e'} l(e) - l(e')\Big] + \Big[\sum_{e \in (PD \setminus \sim)} l(e) - \pi_\Delta(e)\Big] - \Big[\sum_{e \in (PD' \setminus \sim)} l(e) - \pi_\Delta(e)\Big] \\
&= \Big[\sum_{e \sim e'} \langle(-1,1), e - e'\rangle\Big] \\
&\quad + \Big[\sum_{e \in (PD \setminus \sim)} \langle(-1,1), e - \pi_\Delta(e)\rangle\Big] \\
&\quad - \Big[\sum_{e \in (PD' \setminus \sim)} \langle(-1,1), e - \pi_\Delta(e)\rangle\Big],
\end{aligned}$$

where the marked equality comes from the fact that $\pi_\Delta(e) \in \Delta$, and thus $l(\pi_\Delta(e)) = 0$. With this,

$$\begin{aligned}
|l^*(PD) - l^*(PD')| &\leq ||(-1,1)||_2 \Big[\sum_{e \sim e'} ||l(e) - l(e')||_2 + \sum_{e \in (PD \setminus \sim) \cup (PD' \setminus \sim)} ||l(e) - \pi_\Delta(e)||_2\Big] \\
&\leq \sqrt{2}\sqrt{2}[\sum_{e \sim e'} ||l(e) - l(e')||_\infty + \sum_{e \in (PD \setminus \sim) \cup (PD' \setminus \sim)} ||l(e) - \pi_\Delta(e)||_\infty\Big] \\
&\leq 2 \cdot W_1(PD, PD').
\end{aligned}$$

□

Note that, when comparing l and l^*, even with this result of stability (and the fact that l^*, unlike l, is defined for all persistence diagrams) there is no reason a priori to expect one to perform better than the other — and indeed, we made some experiments that empirically indicated that l performs better as an estimator of generalization than l^*. This is likely because l is more dependent on features with low-persistence, which seem to carry more information about generalization.

4.1.2 Persistence landscapes

Let $e = (b, d)$ be some persistence pair. Then, the **tent function** associated to e is $T_e(t) = \max(0, \min(t - b, d - t))$. Now, given a persistence diagram PD, the **persistence landscape** associated to this diagram (and to the underlying persistence module) is

$$PL(x;k) = \begin{cases} kmax(\{T_e(x) \mid e \in PD'\}) & \text{if } k \leq \#PL \\ 0 & \text{otherwise,} \end{cases} \tag{4.1}$$

where

- $kmax(S)$ is the k-th largest element in a multiset S,
- PD' is PD without infinities (i.e., $\{(b,d) \in PD \mid \{b,d\} \cap \{-\infty, +\infty\} = \varnothing\}$),
- $\#PL := \#PD'$.

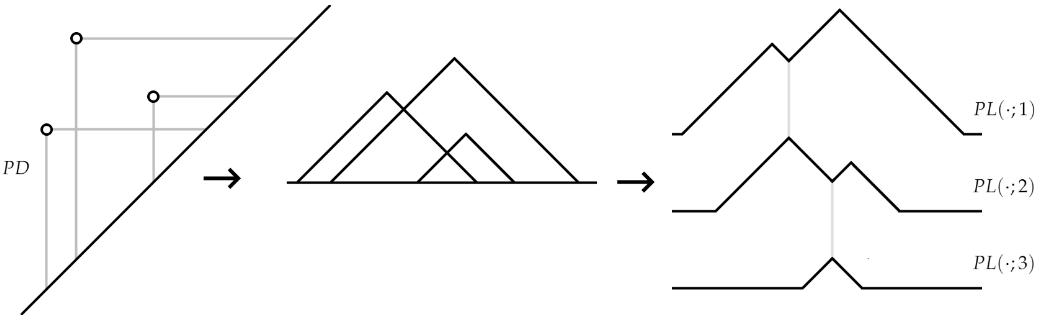


Figure 4.1: A persistence diagram and the resulting landscape, divided into three non-zero envelopes

For each $k \in \mathbb{N}$, $PL(\cdot;k)$ is the **k-th envelope** of PL. Each envelope is a bounded integrable function. Since $k > \#PL \implies PL(\cdot,k) = 0$, PL as a $\mathbb{R} \times \mathbb{N} \to \mathbb{R}$ function is integrable with respect to the measure product of the counting measure on $\mathbb{N}$ and the standard Lebesgue measure on $\mathbb{R}$. Now, since each envelope is bounded,

$$\langle PL, PL' \rangle := \sum_{k \leq \min(\#PL, \#PL')} \int_{\mathbb{R}} PL(x,k) \cdot PL(x,k) \, dx$$

is a valid inner product. Along the same lines, the **landscape p-norm** is

$$||PL||_p = \sum_{k \in [1 \ldots \#PL]} \left(\int_{\mathbb{R}} PL(\cdot;k)^p \right)^{\frac{1}{p}},$$

and each of the summands is the **k-th envelope p-norm** of PL. A more detailed treatment of these concepts can be found in [3].

Theorems 12[2] and 13 in [3] provide stability results for the persistence landscape, but they are both results using the supremum norm. While using this to obtain a stability result of

[2] Note that applying Theorem 12 to Čech filtrations requires expressing them as *sublevel set filtrations*

the landscape p-norm, it would be necessary to include a term to account for the measure of $\{PL > 0\}$.

Note that the naïve implementation of 4.1 is (for any purpose that involves evaluating PL more than once) very inefficient. However, the simple nature of PL makes PL easy to express combinatorially — for instance, any envelope of PL can be described by its critical points alone. Some efficient algorithms for obtaining these simplified representations are available in [4] — from these simplified representations, it is fairly easy to analytically compute landscape products or norms.

4.1.3 Persistence images

Persistence images are an idea introduced in [1]. As the title explains, they serve as a stable vector representation of persistent homology. A few steps are needed to build up to the definition:

- $T(b, d) = (b, d - b)$ is the birth-death to birth-persistence coordinate transform.
- ϕ is a differentiable probability distribution on $\mathbb{R}^2$ with mean 0. Moreover, $\phi_u := \phi(x - u)$.
- f is a $\mathbb{R}^2 \to \mathbb{R}_{\geq 0}$ bounded differentiable weighting function, to be applied to birth-persistence pairs, such that $f(\cdot, 0) = 0$.

Now, let PD be a persistence diagram. Then, the **persistence surface** of PD (or the underlying persistence module) with distribution ϕ and weighting function f is

$$\begin{aligned} \rho \colon \mathbb{R}^2 &\to \mathbb{R} \\ x &\mapsto \sum_{e \in T(PD)} f(e) \cdot \phi_e(x). \end{aligned}$$

Theorem 1 in [1] ensures that persistence surfaces are stable (in the supremum norm) with respect to W_1. Persistence images are then defined as discretizations of persistence surfaces, and their stability follows automatically. The term "discretizations" is intentionally vague — for instance, for some set of points $(p_1, ..., p_n) \subseteq \mathbb{R}^2$ and a persistence surface ρ, one might just consider $(\rho(p_1), ..., \rho(p_n))$ as a suitable finite stable representation of ρ (and thus the underlying persistence module). The approach followed in the original paper is more principled than this:

Let $P := (p_1, ..., p_n)$ be an ordered set of *pixels*, i.e., a set of boxes in $\mathbb{R}^2$ encompassing some square region, such that for $i \neq j$, $p_i \cap p_j$ is a null measure set. Then, the **persistence image** of ρ over P is $(\overline{p_1}, ..., \overline{p_n})$, where $\overline{p} := \int_p \rho$.

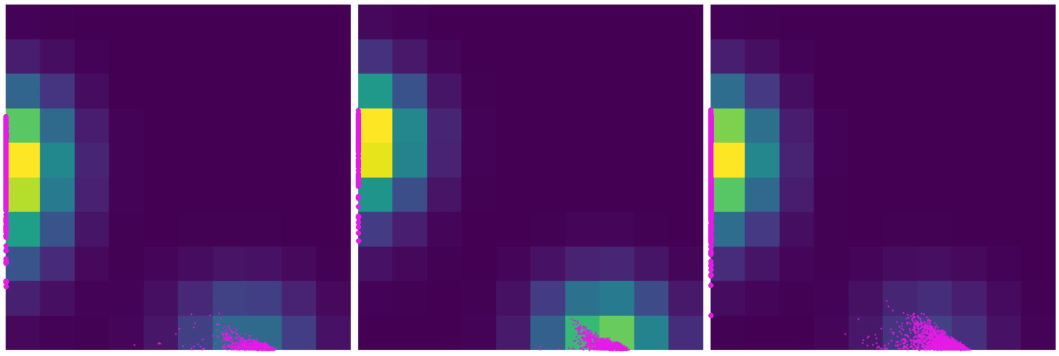


Figure 4.2: Some persistence images (in green), with the image through T of the corresponding diagram overlaid in magenta

Efficient computation

Note that each p admits an expression as $p = [x_-, x_+] \times [y_-, y_+]$ (or something measure-equivalent), and therefore

$$\begin{aligned}
\overline{p} &= \int_{x^-}^{x^+} \int_{y^-}^{y^+} \sum_{e \in T(PD)} w(e) \phi_e(u)\, du \\
&= \sum_{e \in T(PD)} w(e) \int_{x^-}^{x^+} \int_{y^-}^{y^+} \phi_e(u)\, du \\
&= \sum_{e \in T(PD)} w(e) \int_{x^- - e_x}^{x^+ - e_x} \int_{y^- - e_y}^{y^+ - e_y} \phi_0(u)\, du.
\end{aligned}$$

Now, $w(e)$ is either easy to compute or precomputable, and one can store a large grid of integrals of ϕ_0 over sections some domain[3] D in such a way that computing the integrals in the last expression is $O(1)$, making the computation of persistence images tractable.

Restrictions on the weighting function

We provide a small addition to the results in [1]. Namely, we show that if $\phi(0) \neq 0$, the requirements of continuity and piecewise differentiability of f can be relaxed to to Lipschitz continuity, and show that this condition (together with being zero on $\{x, 0\}$) is actually necessary.

The fact that Lipschitz continuity is sufficient is actually proved in the original paper — it

[3] D has to be large enough to contain $(x^- - e_x, x^+ - e_x) \times (y^- - e_y, y^+ - e_y)$ for any e in the underlying persistence diagram DP — assuming it is a rectangular domain, this would be $(x^- - max(e_x), x^+ - min(e_x)) \times (y^- - max(e_y), y^+ - min(e_y))$, where the maxima/minima are taken over $e \in DP$.

is the only fact that is used about the weighting function f (in Lemma 1). We only need to show that Lipschitz continuity is necessary, or equivalently, that the stability result does not hold for a non-Lipschitz weighting function f.

Proposition 4.3. *Let PD and PD' be persistence diagrams, and ρ, ρ' be the persistence surfaces of PD, PD' with some distribution ϕ satisfying $\phi(0) \neq 0$ and weighting function f. If f is not Lipschitz, then there is no $K \in \mathbb{R}$ independent of PD and PD' such that $||\rho - \rho'||_\infty \leq K \cdot W_1(PD, PD')$.*

Proof. Let $\{p_i\}_{i\in\mathbb{N}}$ and $\{q_i\}_{i\in\mathbb{N}}$ be sequences from $\mathbb{R}^2$ such that $\{K_i\}_{i\in\mathbb{N}} := |f(p_i) - f(q_i)|/||p_i - q_i||_2 \to \infty$ as $i \to \infty$, which must exist since f is Lipschitz. Moreover, as $\mathbb{E}^2$ is path-connected and length-metric, we can select $\{p_i\}_{i\in\mathbb{N}}$ and $\{q_i\}_{i\in\mathbb{N}}$ such that $\{D_i\}_{i\in\mathbb{N}} := ||p_i - q_i||_2 \to 0$ as $i \to \infty$, as well as $K_i \to \infty$.

Then, let PD_i be the persistence diagram $\{T^{-1}(p_i)\}$, and likewise with PD'_i and $T^{-1}(pq_i)$. Note that T^{-1} is linear, in particular, $T^{-1}(x, y) = (x, x + y)$, and thus $||T^{-1}(x) - T^{-1}(x')||_\infty \leq 2||x - x'||_\infty$. It follows that $W_1(PD_i, PD'_i) \leq 2||p_i - q_i||_\infty \leq 2\sqrt{2}||p_i - q_i||_2$. Now, let ρ_i and ρ'_i be the persistence surfaces of PD_i and PD'_i respectively, with distribution ϕ and weighting function f. Then,

$$\begin{aligned}\rho_i(p_i) - \rho'_i(p_i) &= \phi(0)f(p_i) - \phi(p_i - q_i)f(q_i)\\ &= \phi(0)\Big(f(p_i) - f(q_i)\Big) + f(q_i)\Big(\phi(0) - \phi(p_i - q_i)\Big).\end{aligned}$$

Now, if divided by $W_1(PD_i, PD'_i)$, the fact that $W_1(PD_i, PD'_i) \leq 2\sqrt{2}||p_i - q_i||_2$ implies that the first summand is bounded below by $\phi(0)K_i$, which, as $\phi(0) \neq 0$, must go to infinity. The second summand, on the other hand, is bounded by $||f||_\infty||\nabla\phi||_\infty$, which is finite. This shows that $\rho_i(p_i) - \rho'_i(p_i)/W_1(PD_i, PD'_i)$ goes to infinity — and thus $||\rho_i - \rho'_i||/W_1(PD_i, PD'_i)$ is unbounded, as we wanted. □

4.2 Metric estimators

There is an observation to be made that the persistence diagrams obtained from the point clouds in our experiments very rarely carry any high-persistence features (except the necessary H_0 generators). One might conjecture, then, that the topological information to be gained from these point clouds is more local rather than global, and question whether this information can be obtained by observing the finite metric space directly without building a persistence module. This section introduces two simple estimators aimed at doing just that: namely, estimating the dimension of a point cloud, as task is fairly straightforward and is an obvious example of a local topological feature. This is a rather naïve approach, but its intent is perhaps more to get an idea of *how much* information is being contributed by TDA.

We will recall two common definitions of dimension, and loosely translate these to expressions that can be evaluated for finite point clouds.

4.2.1 Correlation dimension

Let $\mathcal{M}$ be a metric space equipped with a measure μ on its underlying set $\mathcal{M}_X$. Then, $\mu \otimes \mu$ is a measure on $\mathcal{M}_X^2$. This is enough to define $c(r) = (\mu \otimes \mu)(\{d_{\mathcal{M}}(x, y) \leq r\})$. Then, the **correlation dimension** of $\mathcal{M}$ is the limit as $r \to 0$ of $\frac{\log(c(r))}{\log(r)}$.

There is a large amount of literature on the properties of the correlation dimension and ways to approximate it. We adopt a rather simple approach: Let X be a finite set of points, with pairwise distances $d : X \times X \to \mathbb{R}_{\geq 0}$. The assumption here is that the $x \in X$ (and their distances) are sampled from a larger metric space $\mathcal{M}$ that we are trying to study. Then, let $c(r) = \#\{(x, y) \in X \times X \mid d(x, y) \leq r\}$.

The reasoning behind the limit in the definition of the correlation dimension was that $c(r)$ was assumed to be roughly of the form r^D for some D for $r \approx 0$, and the limit was a form of recuperating this D. In the finite case, clearly $\frac{\log(c(r))}{\log(r)} \to 0$ — thus, we define $a(r_1, r_2) = \frac{\log(c(r_1)) - \log(c(r_2))}{\log(r_1) - \log(r_2)}$, and instead of $\lim_{r_1, r_2 \to 0} a(r_1, r_2)$, we construct a matrix $M_{ij} = a(r_i, r_j)$, where $\{r_i\}$ is a set of logarithmically spaced values close to 0. We select the estimate of the dimension of $\mathcal{M}$ (and thus the dimension of X) to be the 90th percentile of the values of M.

4.2.2 Minkowski dimension

Let $d \in \mathbb{N}$ and $X \subseteq \mathbb{E}^d$. Now, define $c(r)$ to be the amount of d-dimensional hypercubes X intersects, from some grid of hypercubes of side length r tesselating $\mathbb{E}^d$. The **box-counting dimension** of X, then, is the limit as $r \to 0$ of $-\frac{\log(c(r))}{\log(r)}$. This is often also referred to as the *Minkowski dimension* — however, we will use this expression to refer to a close cousin of the box-counting dimension:

Let $\mathcal{M}$ be some metric space, and define $c(r)$ to be the minimum amount of r-radius balls required to cover $\mathcal{M}$. Then, the **Minkowski dimension** of X is the limit as $r \to 0$ of $-\frac{\log(c(r))}{\log(r)}$.

The discretization, like before, requires a finite set X with defined pairwise distances $d : X \times X \to \mathbb{R}_{\geq 0}$. In principle, c should be defined as

$$c(r) = \min_{C \in Cov_r(X)} \#C, \text{ where } Cov_r(X) = \{C \subseteq X \mid \forall x \in X \exists c \in C d(x, c) < r\}.$$

In practice, however, we simply estimate this with a greedy algorithm to find a cover of X. Then, the definition of $a(r_1, r_2)$ and estimation of the dimension proceed exactly like in the prior section, except $a(r_1, r_2)$ having the opposite sign.

Chapter 5

Results

In this chapter we go over how the experiments were performed and what results were obtained, along with some discussion of these results and how they motivate further experiments.

5.1 Context

5.1.1 Model datasets

A number of tests were performed on tasks 1 and 2 of PGDL — a NeurIPS competition on prediction of the generalization gap. Descriptions of both of these sets of models can be found at [14]. For the purpose of understanding how much the accuracy of these results depended on the complexity of the network structure, we designed a set of simpler models, all of them MLPs. To define this set of models, note that the structure (i.e., the underlying computation graph schema) of an MLP can be parametrized by:

- The input and output width, w_i and w_o. These will be determined by the dataset we attempt to fit.
- The amount of hidden layers, h.
- The width of each hidden layer, $(w_1, ..., w_h)$.

This leaves out activations — we use softmax at the output layer and ReLu for every intermediate layer. Then, we consider the set of functions $[0,1] \to \mathbb{R}$ parametrized by (a, b, c):

$$f_{a,b,c}(x) = a\Big(\frac{1}{(x+1)^b} + c\Big).$$

Now, one can reparametrize $f_{a,b,c}$, writing $f_{\alpha,\beta,\gamma}$, such that $f_{\alpha,\beta,\gamma} = \beta f_{\alpha,1,\gamma}$ (meaning β controls the scale), $||f_{\alpha,1,\gamma}||_1 = 1$, and $f_{\alpha,\beta,\gamma}(1) = \gamma\beta$ (note that the 1-norm here is $\int_0^1$, not $\int_{\mathbb{R}}$). This reparametrization is straightforward to obtain analitically. Now, for some fixed w_i, w_o, h, consider some $n \in \mathbb{N}$ understood to be the desired amount of nodes in a network we are building. Let $m = n - w_i - w_o$. Then, it suffices to set:

$$w = \Big(f_{\alpha, m/h, w_o/m}(\frac{1}{h}) \Big)_{i \in [1 \ldots h]}$$

for some shape parameter α. In truth, this definition does not guarantee that the MLP defined by w (and w_i, w_o, h) has exactly n nodes, owing to the fact that $\sum_{i=1}^{h} f(i/h)/h \neq \int_0^1 f$. However, it is good enough to obtain a set of reasonable, diverse MLPs to test on. Fig. 5.1 shows a few of the networks obtained, particularly highlighting the effect of the shape parameter.

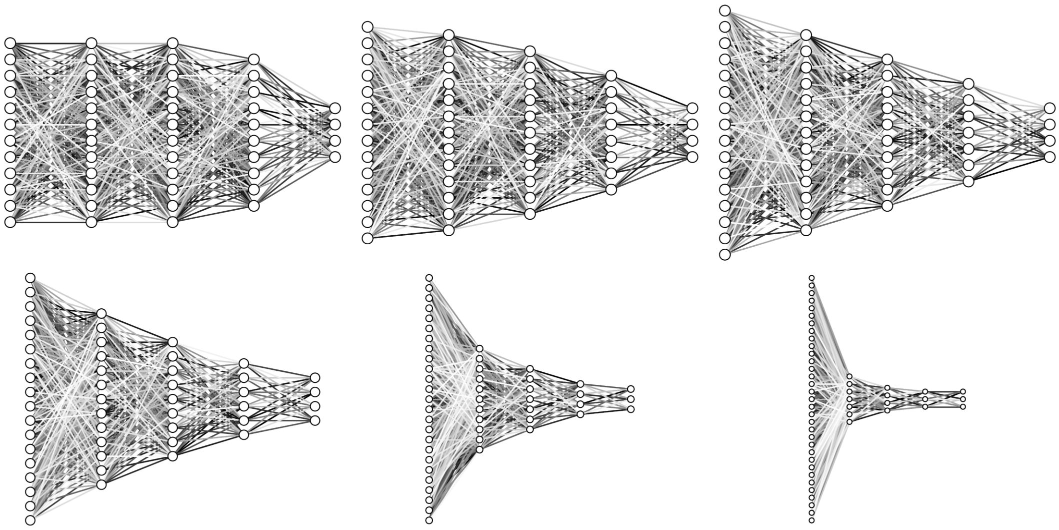


Figure 5.1: MLPs following the distribution described, with $\gamma = 0.1$ and 50 neurons. α varies from -8 to 8. The shading is for visibility only.

We created two sets of 60 MLP structures: One set to be trained on the **IMDB**[1] dataset, and another on **CIFAR10**[2]. Desired number of neurons is fixed at 1000. γ is always set so that the width of the last layer matches the width of the output layer. α is varied between -5 and 5. The width of the first and last layers are determined by the problem. The networks to be trained on **CIFAR10** include a flattening layer after the input, and the **IMDB** dataset was fed through a word2vec process before using it as input for training data. Each network is saved in an underfitted and overfitted state, aswell as a median state of "nice fitting". This makes for a total of 180 models for each dataset. We will refer to these sets of models as **CIFAR10** and **IMDB**, as well as using **Task 1** and **Task 2** to reference the model sets from PGDL.

[1] Text-based sentiment analysis problem — IMDB reviews associated to a positive/negative review score [19].

[2] Image classification problem, with images divided into ten separate categories [17].

5.1.2 Estimators

The analysis of an estimator (or a group of estimators) will always depend on some (finite) set of models S. Of course this means that this analysis will be subject to bias in the choice of S — this will be discussed when it is relevant. In any case, given an S, an **estimator** is simply a mapping $e\colon S \to \mathbb{R}$. Note that g, the generalization gap, is an estimator[3].

Comparing estimators will be done through:

- Their sample correlation, $CORR(e_1, e_2)$ (as $\mathbb{R}$-vectors, indexed by S). In general this is not a very principled way to compare estimators, but it is simple enough to yield easily interpretable results.
- Their mutual information, $MI(e_1, e_2)$. This is defined as the mutual information between V_{e_1} and V_{e_2}, (where V_e is a random variable defined on the uniform probability space over S^2 with $V_e(m_1, m_2) = sgn(e(m_1) - e(m_2))$), over the entropy of e_2. This means MI is *not* symmetric, although only if $\mathbb{P}(V_{e_1} = 0 \text{ or } V_{e_2} = 0) \neq 0$. Note that this is significantly different from the mutual information used in [16], because we consider the possibility that $e(m1) = e(m2)$ (meaning $V_e(m_1, m_2) = 0$), and because the sets of models that we use are not nicely parametrizable enough that we can consider the performance of an estimator when varying along one particular hyperparameter axis.

5.1.3 Sampling

In Section 3.4 we introduced alongside each space $\mathcal{C}$ a sequence of approximations $(\mathcal{C})^n$, corresponding to approximating each distance between elements in $\mathcal{C}$ by computing the correlation of the activations for only n elements from the dataset. Since computing $\mathcal{C}$ is either unfeasible or impossible, we must make a choice of n. We usually set $n = 2000$, as this is a fair bit into the point of diminishing returns in terms of accuracy, and we cannot set n much higher because of memory limitations.

We also have to consider downsampling in $\mathcal{C}$ directly, that is, selecting a subspace $\mathcal{M} \subseteq \mathcal{C}$ under some preset size. This is a requirement imposed directly by the usage of TDA, as the state of the art packages for computation of persistent homology struggle with $\#\mathcal{M} \geq 1000$. For the original experiments with the model sets from PGDL, $\#\mathcal{M}$ was set at 200 — particularly low to allow for computation of features of higher dimension. For later experiments $\#\mathcal{M} = 1000$, which is a decently representative sample of the Ciprian spaces corresponding to the models described above.

[3] g, in particular, needs information about the training and validation datasets. Only estimators serving as a benchmark can depend on the validation dataset, but a lot of estimators depend on the training data. This is not considered in the above definition of estimator, as it pays no mind to how any of the estimators are computed or defined outside of S.

5.2 TDA-based estimators

Recall that in Section 3.4 we defined six Ciprian spaces for a network, namely:

$$\mathcal{C}_D, \mathcal{C}_S, \mathcal{C}_E, \mathcal{C}_{PD}, \mathcal{C}_{PS}, \mathcal{C}_{PE}.$$

As they are finite, for each of these $\mathcal{C}$ we can consider the corresponding Vietoris-Rips filtration, and thus the persistence diagram of this — namely, $PD(VR(\mathcal{C}))$. For any $p \in \mathbb{N}$ we can consider only p-dimensional features, i.e., points from $PD_p(VR(\mathcal{C}))$, though for computational reasons we never go above $p = 3$. This can give us a total of 24 diagrams to consider, with several summaries to be extracted from each diagram. We intend to reduce this number to make the analysis somewhat tractable.

5.2.1 Choice of metric

Experimentally, two main things were found:

- Euclidean, sphere and direct correlation metrics/similarities all perform comparably, and likewise for their projective variants. Fig. 5.2 shows this for the projective metrics on **IMDB**. Note the similar correlations with generalization for each metric, as well as the bright diagonal lines indicating consistently > 0.98 correlation between the same estimators in different metrics.

  This is not surprising, considering the comments from Section 3.4. With this in mind, we can comfortably restrict the analysis to sphere correlation metrics ($\mathcal{C}_S$, $\mathcal{C}_{PS}$). This specific choice is because their metricity (as this makes more stability results applicable) — in particular, ($\mathcal{C}_S$, $\mathcal{C}_{PS}$) are chosen over ($\mathcal{C}_E$, $\mathcal{C}_{PE}$) because, while both are geodesic spaces, those of ($\mathcal{C}_S$, $\mathcal{C}_{PS}$) coincide with the quotient space induced by the relation described in Section 3.2, which makes it a more natural choice.

- Projective metrics perform better than non-projective metrics — not by a large margin, but fairly consistently. This is visible, for both TDA-based and simple metric estimators, in Fig. 5.3. We conjecture that, rather than being something intrinsic about the relation between these metrics and generalization, this is due to the fact that the ambient space is smaller in projective metrics — thus, $\mathcal{C}_{PS}$ is more saturated than $\mathcal{C}_S$ with the same amount of points, leading to more accurate readings about the structure of the metric space.

This justifies restricting ourselves to $\mathcal{C}_{PE}$, which reduces six-fold the amount of TDA-based estimators to consider.

5.2.2 Dimension of features

Persistent homology is very costly to compute. Moreover, this cost increases very fast with the dimension of the features one is interested in[4]. For this reason, we intend to note some of the relationships between estimators based on features of different dimensions,

[4] See [22] for information on the computation (and time complexity) of persistent homology.

with the intention of finding a reasonable justification for restricting the analysis to a low maximum dimension. Fig. 5.4 shows some scatter plots, illustrating the relation between average midlife of features in different dimensions in the model sets from PGDL. The main takeaway is that there is a (not very strong, but visible) linear relation between estimators based on features of dimension 0, 2 or 3 — but dimension 1 seems to have a non-linear relation to the others. However, these relationships are different in other model sets (Fig. 5.5, Fig. 5.6). For this reason, we restrict the dimension p of computation of homology to $p \leq 2$ (as opposed to $p \leq 1$).

5.2.3 Persistence landscapes

We used norms of envelopes as estimators. This means that we have a set of estimators parametrized by 3 integers: The dimension d of features we are looking at, the order p of the norm we will take, and the index i of the envelope we will take the norm of in a certain landscape. This large and nicely parametrizable set of estimators is one of the reasons for how much there is to understand about the use of landscapes as estimators. Importantly, they have significantly different behaviour depending on the context:

- The results for **CIFAR10**, attached in Fig. 5.7, show a correlation to generalization ranging from -0.5 to -0.6 for dimensions 1 and 2 when taking the 1-norm of the first envelope. This correlation slowly decays when taking higher order norms or further envelopes. Dimension 0 landscapes have almost no predicting power for generalization.
- The results for **Task2** (Fig. 5.9), on the other hand, perform significantly worse, with correlation below 0.2, showing very little dependence on the envelope or norm chosen.
- The results for **IMDB**, attached in Fig. 5.8, show correlation weaker than the results for **CIFAR10**, but visibly increasing with respect to both norm order and envelope index for landscapes of dimension 1 features.

Experimentally, landscape-based estimators perform, at best, on par with simple PD-based estimators (Figs. 5.3, 5.10, 5.11, 5.12). This includes the model sets where, such as with **CIFAR10** and **Task 2**, it seems likely that maximum correlation with generalization from a landscape-based estimator has been attained. This leads us to conjecture that this is the case *in general*, that is, estimators based on landscape norms provide at best the same accuracy as PD-based estimators. This is supported by (Figs. 5.3, 5.11) — note that landscape estimators with high correlation with generalization always also exhibit high correlation with PD-based estimators, meaning they do not add information to the analysis.

The fact that persistence landscapes of 0-dimensional features are not very informative is to be expected — a property of Vietoris-Rips (and Čech) filtrations is that all dimension-0 features are born at time 0, and thus lie on the vertical axis of their persistence diagram. It is easy to see that this implies that the k-th envelope of their persistence landscape is entirely determined by one single feature (namely, the feature with the k-th highest persistence).

5.3 Non-TDA estimators

5.3.1 Metric estimators

Comparisons of metric estimators and TDA-based estimators can be seen at Figs. 5.3, 5.10, and 5.11. There are two main observations to be made:

- Correlation dimension generally performs better than Minkowski dimension, though not by a large margin.
- Metric estimators generally have a high correlation with TDA-based estimators, at least when these TDA-based estimators have a decent correlation with generalization. While this is not very surprising, it is worth noting that they generally have a higher correlation with estimators based on features of dimension > 0, rather than $= 0$.

As estimators of generalization by themselves, they provide very strong results on occasion, but are significantly less reliable than TDA-based estimators.

5.3.2 Other estimators

Among the other, non-topological non-metrical estimators of generalization tested against TDA-based estimators, two interesting relationships came up:

- With estimators based on spectral norms (see Fig. 5.12). Surprisingly, although correlation between spectral methods and TDA methods was fairly high for some pairs of estimators (0.8, sometimes close to 0.9), this seems to be more common among estimators that perform *badly* at the task of predicting generalization. Indeed, spectral methods themselves got mediocre scores on our model sets.
- Figs. 5.13 and 5.14 show that methods based on path norms do not exhibit many strong correlations with TDA-based methods, but have some (comparatively) significant mutual information. This suggests there is a less evident (or at the very least non-linear) relationship between the path norm and TDA — this is particularly noteworthy considering the path norm is defined from structural information of a network, while the metric spaces we perform TDA on carry no explicit structural information whatsoever.

5.4 Conclusions

In general, TDA-based estimators of generalization are at the very least competitive. They can achieve a correlation with generalization as high as 0.75, and even higher in some contexts — but this, of course, has to be taken with a grain of salt. The bulk of the tests were performed on models designed precisely to be simple, and we have seen that the more impressive examples of predictive power happen in these simple model environments.

This, of course, does not change the fact that TDA methods in deep learning show promise. Even the less impressive results are significantly above the baseline, and, moreover, in the contexts that have TDA-based estimators strongly correlate with generalization, this does not imply strong correlation with some other well-known estimators of generalization — meaning that these methods bring new information to the table, as opposed to rephrasing well-known older methods.

On the theoretical side, we have grounded the work of Ciprian et al in [7], showing that the spaces we consider are indeed metric. This — with well as the result of the convergence with respect to sampling — can allow the application of the theoretical machinery of topological data analysis, such as to obtain results on the stability of the topological summaries with respect to sampling.

There are several sections of this work that are visibly open for refinement and expansion. There is little exploration done of Ciprian spaces before focusing on the topological information, but the brief experimentation with dimension estimators already shows that (in some situations) a decent amount of the information we obtain through TDA does not necessarily need TDA to be obtained. On the TDA end of things, there is an ever-growing amount of topological summaries that could provide us with novel information.

Our analysis did not include any of these things — it had a fairly narrow focus, with the intention of providing a solid foundation for the application of TDA to the estimation of generalization in deep learning. Our results are more concerned with the viability of this (as told by the relationship between TDA and the current methodology for generalization estimation) than with providing an estimator on par with the current state of the art. Indeed, we only worked with (implicit) single-parameter linear models, even when in several contexts it was clear that a multivariate or nonlinear approach would be beneficial. Moreover, the differences in the relationships between estimators in different model sets, and the closely tied topic of the interpretability of these estimators, are not yet explored. This, as well as every other open door mentioned thus far, is left to future work.

Bibliography

[1] Henry Adams, Sofya Chepushtanova, Tegan Emerson, Eric Hanson, Michael Kirby, Francis Motta, Rachel Neville, Chris Peterson, Patrick Shipman, and Lori Ziegelmeier, *Persistence images: A stable vector representation of persistent homology*, arXiv e-prints (2015), arXiv:1507.06217.

[2] Dominique Attali, André Lieutier, and David Salinas, *Vietoris–Rips complexes also provide topologically correct reconstructions of sampled shapes*, Computational Geometry **46** (2013), no. 4, 448 – 465, 27th Annual Symposium on Computational Geometry (SoCG 2011).

[3] Peter Bubenik, *The persistence landscape and some of its properties*, arXiv e-prints (2018), arXiv:1810.04963.

[4] Peter Bubenik and Paweł Dłotko, *A persistence landscape toolbox for topological statistics*, Journal of Symbolic Computation **78** (2017), 91 – 114, Algorithms and Software for Computational Topology.

[5] M. D. Buhmann, *Radial basis function networks*, pp. 823–827, Springer US, Boston, MA, 2010.

[6] Dmitri Buragi, Yuri Burago, and Sergei Ivanov, *A Course in Metric Geometry*, American Mathematical Society, 2001.

[7] Ciprian Corneanu, Meysam Madadi, Sergio Escalera, and Aleix Martinez, *Computing the testing error without a testing set*, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020).

[8] Dumitru Erhan, Y. Bengio, Aaron Courville, and Pascal Vincent, *Visualizing higher-layer features of a deep network*, Technical Report, Univeristé de Montréal (2009).

[9] Robin Forman, *Morse theory for cell complexes*, Advances in Mathematics **134** (1998), no. 1, 90 – 145.

[10] R. Ghrist, *Elementary Applied Topology*, CreateSpace Independent Publishing Platform, 2014.

[11] Chad Giusti, Eva Pastalkova, Carina Curto, and Vladimir Itskov, *Clique topology reveals intrinsic geometric structure in neural correlations*, Proceedings of the National Academy of Sciences **112** (2015), no. 44, 13455–13460.

[12] Allen Hatcher, *Algebraic Topology*, Cambridge University Press, Cambridge, 2002. MR 1867354 (2002k:55001)

[13] Zhiwu Huang, Jiqing Wu, and Luc Van Gool, *Building deep networks on Grassmann manifolds*, arXiv e-prints (2016), arXiv:1611.05742.

[14] Yiding Jiang, Pierre Foret, Scott Yak, Daniel M. Roy, Hossein Mobahi, Gintare Karolina Dziugaite, Samy Bengio, Suriya Gunasekar, Isabelle Guyon, and Behnam Neyshabur, *NeurIPS 2020 Competition: Predicting Generalization in Deep Learning*, arXiv e-prints (2020), arXiv:2012.07976.

[15] Yiding Jiang, Dilip Krishnan, Hossein Mobahi, and Samy Bengio, *Predicting the generalization gap in deep networks with margin distributions*, International Conference on Learning Representations, 2019.

[16] Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio, *Fantastic generalization measures and where to find them*, CoRR **abs/1912.02178** (2019).

[17] Alex Krizhevsky, *Learning multiple layers of features from tiny images*, University of Toronto (2012).

[18] Li Li, Wei-Yi Cheng, Benjamin S. Glicksberg, Omri Gottesman, Ronald Tamler, Rong Chen, Erwin P. Bottinger, and Joel T. Dudley, *Identification of type 2 diabetes subgroups through topological analysis of patient similarity*, Science Translational Medicine **7** (2015), no. 311, 311ra174–311ra174.

[19] Andrew Maas, Raymond Daly, Peter Pham, Dan Huang, Andrew Ng, and Christopher Potts, *Learning word vectors for sentiment analysis*, 01 2011, pp. 142–150.

[20] Yuriy Mileyko, Sayan Mukherjee, and John Harer, *Probability measures on the space of persistence diagrams*, Inverse Problems **27** (2011), no. 12, 124007.

[21] Parth Natekar and Manik Sharma, *Representation based complexity measures for predicting generalization in deep learning*, arXiv preprint arXiv:2012.02775 (2020).

[22] Nina Otter, Mason A. Porter, Ulrike Tillmann, Peter Grindrod, and Heather A. Harrington, *A roadmap for the computation of persistent homology*, EPJ Data Science **6** (2017), no. 1, 17.

[23] Jose A. Perea, Anastasia Deckard, Steve B. Haase, and John Harer, *Sw1pers: Sliding windows and 1-persistence scoring; discovering periodicity in gene expression time series data*, BMC Bioinformatics **16** (2015), no. 1, 257.

[24] Peter Petersen, *Riemannian Geometry*, Springer International Publishing, Cambridge, 2016.

[25] Prajit Ramachandran, Barret Zoph, and Quoc V. Le, *Searching for activation functions*, arXiv e-prints (2017), arXiv:1710.05941.

[26] Vin Silva and Robert Ghrist, *Coverage in sensor networks via persistent homology*, Algebraic & Geometric Topology **7** (2007), no. 1, 339–358.

[27] Scott Yak, Javier Gonzalvo, and Hanna Mazzawi, *Towards task and architecture-independent generalization gap predictors*, arXiv preprint arXiv:1906.01550 (2019).

[28] Yuan Yao, Jian Sun, Xuhui Huang, Gregory R. Bowman, Gurjeet Singh, Michael Lesnick, Leonidas J. Guibas, Vijay S. Pande, and Gunnar Carlsson, *Topological methods for exploring low-density states in biomolecular folding pathways*, The Journal of Chemical Physics **130** (2009), no. 14, 144115.

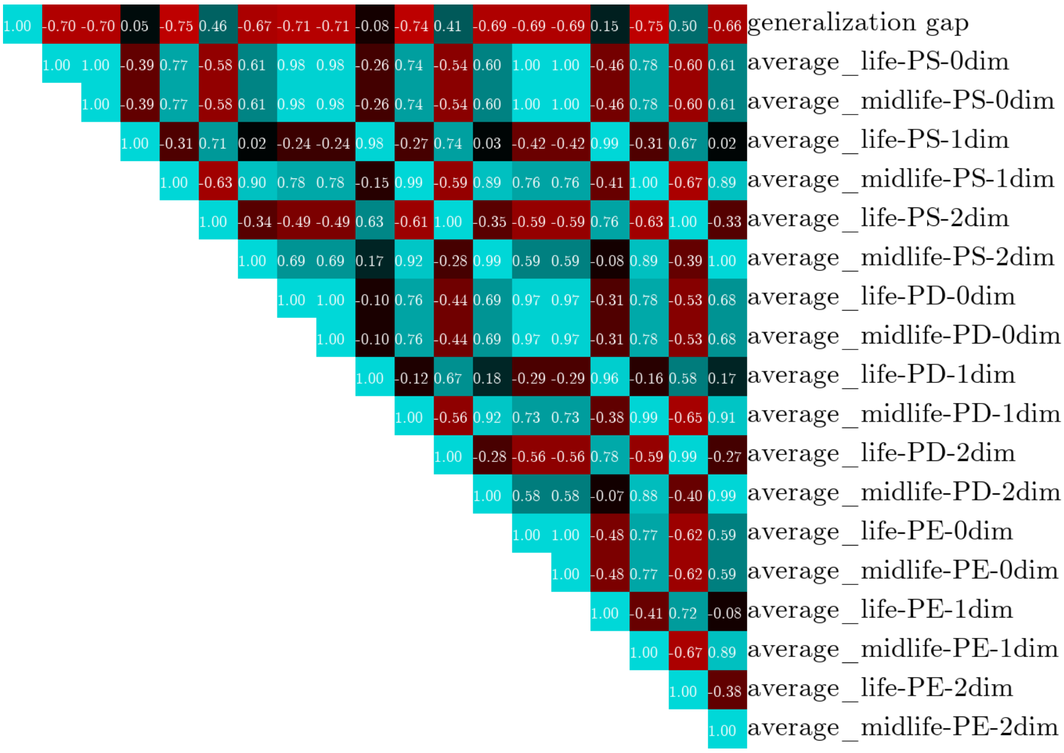
[29] Afra Zomorodian and Gunnar Carlsson, *Computing persistent homology*, Discrete and Computational Geometry **33** (2005), 249–274.

Appendix: Figures from results

A large amount of the graphics in this section are correlation or mutual information matrices. Colors are added to make it easier to navigate the images (i.e., easily detect strong relationships). The names of the estimators corresponding to each row are written on the side. Those corresponding to columns are omitted, but they are in the same order as the rows. Some abbreviations are used to reduce the size of the legend:

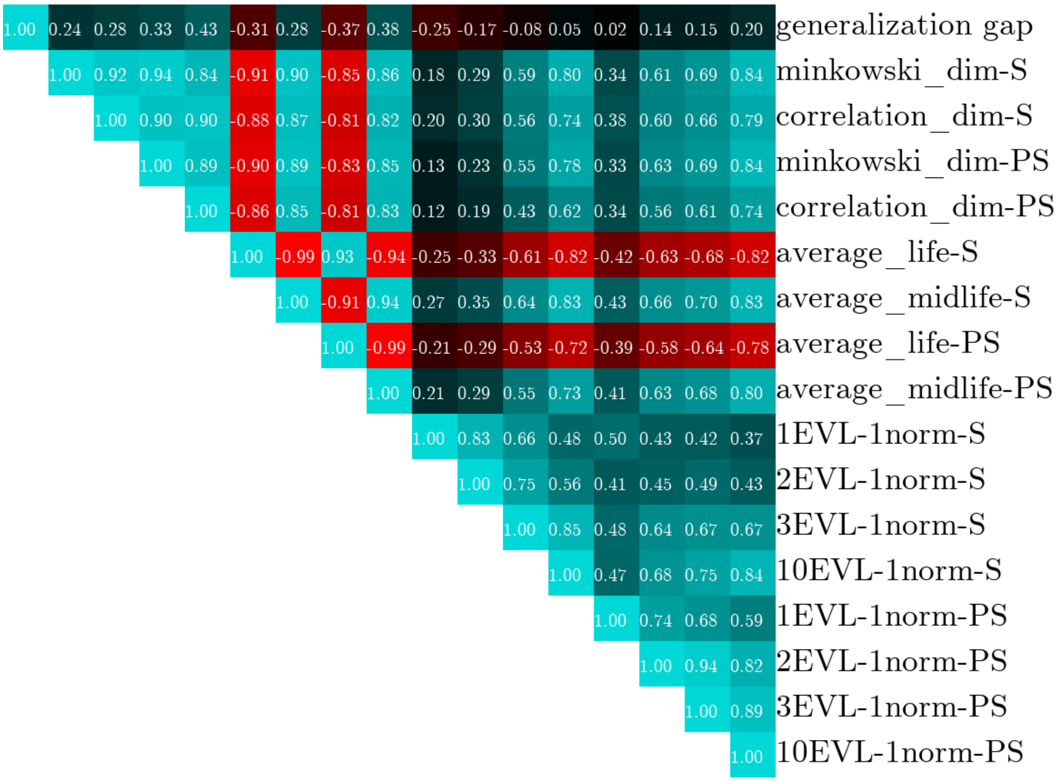
- Metrics are abbreviated with the same codes as in the text (D, S, E, PD, PS, PE)
- "EVL" stands for "envelope" — for instance, "2EVL-1norm-PS-4dim" is the 1-norm of the 2nd envelope of the persistence landscape of the 4-th persistence diagram of the Vietoris-Rips filtration obtained from $\mathcal{C}_P S$
- If the dimension p is absent in an estimator that would usually depend on p, it is derived from the union of all the p-th persistence diagrams, that is, $\bigcup_{p \in [0 \dots 3]} PD_p$
- The estimators used from [16] follow the same naming scheme as the paper, with the exception of the path norm. We use path_norm_p to denote the p-th path norm.

Correlation matrices are symmetric, and thus entries below the diagonal are not shown. Mutual information need not be symmetric in general, so the entire matrix is shown.



1.00	-0.70	-0.70	0.05	-0.75	0.46	-0.67	-0.71	-0.71	-0.08	-0.74	0.41	-0.69	-0.69	-0.69	0.15	-0.75	0.50	-0.66	generalization gap
	1.00	1.00	-0.39	0.77	-0.58	0.61	0.98	0.98	-0.26	0.74	-0.54	0.60	1.00	1.00	-0.46	0.78	-0.60	0.61	average_life-PS-0dim
		1.00	-0.39	0.77	-0.58	0.61	0.98	0.98	-0.26	0.74	-0.54	0.60	1.00	1.00	-0.46	0.78	-0.60	0.61	average_midlife-PS-0dim
			1.00	-0.31	0.71	0.02	-0.24	-0.24	0.98	-0.27	0.74	0.03	-0.42	-0.42	0.99	-0.31	0.67	0.02	average_life-PS-1dim
				1.00	-0.63	0.90	0.78	0.78	-0.15	0.99	-0.59	0.89	0.76	0.76	-0.41	1.00	-0.67	0.89	average_midlife-PS-1dim
					1.00	-0.34	-0.49	-0.49	0.63	-0.61	1.00	-0.35	-0.59	-0.59	0.76	-0.63	1.00	-0.33	average_life-PS-2dim
						1.00	0.69	0.69	0.17	0.92	-0.28	0.99	0.59	0.59	-0.08	0.89	-0.39	1.00	average_midlife-PS-2dim
							1.00	1.00	-0.10	0.76	-0.44	0.69	0.97	0.97	-0.31	0.78	-0.53	0.68	average_life-PD-0dim
								1.00	-0.10	0.76	-0.44	0.69	0.97	0.97	-0.31	0.78	-0.53	0.68	average_midlife-PD-0dim
									1.00	-0.12	0.67	0.18	-0.29	-0.29	0.96	-0.16	0.58	0.17	average_life-PD-1dim
										1.00	-0.56	0.92	0.73	0.73	-0.38	0.99	-0.65	0.91	average_midlife-PD-1dim
											1.00	-0.28	-0.56	-0.56	0.78	-0.59	0.99	-0.27	average_life-PD-2dim
												1.00	0.58	0.58	-0.07	0.88	-0.40	0.99	average_midlife-PD-2dim
													1.00	1.00	-0.48	0.77	-0.62	0.59	average_life-PE-0dim
														1.00	-0.48	0.77	-0.62	0.59	average_midlife-PE-0dim
															1.00	-0.41	0.72	-0.08	average_life-PE-1dim
																1.00	-0.67	0.89	average_midlife-PE-1dim
																	1.00	-0.38	average_life-PE-2dim
																		1.00	average_midlife-PE-2dim

Figure 5.2: Correlation among (and with generalization gap) of PD-based estimators on the **IMDB** model set for the three different projective distance functions.



1.00	0.24	0.28	0.33	0.43	-0.31	0.28	-0.37	0.38	-0.25	-0.17	-0.08	0.05	0.02	0.14	0.15	0.20	generalization gap
	1.00	0.92	0.94	0.84	-0.91	0.90	-0.85	0.86	0.18	0.29	0.59	0.80	0.34	0.61	0.69	0.84	minkowski_dim-S
		1.00	0.90	0.90	-0.88	0.87	-0.81	0.82	0.20	0.30	0.56	0.74	0.38	0.60	0.66	0.79	correlation_dim-S
			1.00	0.89	-0.90	0.89	-0.83	0.85	0.13	0.23	0.55	0.78	0.33	0.63	0.69	0.84	minkowski_dim-PS
				1.00	-0.86	0.85	-0.81	0.83	0.12	0.19	0.43	0.62	0.34	0.56	0.61	0.74	correlation_dim-PS
					1.00	-0.99	0.93	-0.94	-0.25	-0.33	-0.61	-0.82	-0.42	-0.63	-0.68	-0.82	average_life-S
						1.00	-0.91	0.94	0.27	0.35	0.64	0.83	0.43	0.66	0.70	0.83	average_midlife-S
							1.00	-0.99	-0.21	-0.29	-0.53	-0.72	-0.39	-0.58	-0.64	-0.78	average_life-PS
								1.00	0.21	0.29	0.55	0.73	0.41	0.63	0.68	0.80	average_midlife-PS
									1.00	0.83	0.66	0.48	0.50	0.43	0.42	0.37	1EVL-1norm-S
										1.00	0.75	0.56	0.41	0.45	0.49	0.43	2EVL-1norm-S
											1.00	0.85	0.48	0.64	0.67	0.67	3EVL-1norm-S
												1.00	0.47	0.68	0.75	0.84	10EVL-1norm-S
													1.00	0.74	0.68	0.59	1EVL-1norm-PS
														1.00	0.94	0.82	2EVL-1norm-PS
															1.00	0.89	3EVL-1norm-PS
																1.00	10EVL-1norm-PS

Figure 5.3: Correlation among (and with generalization gap) topological/metric estimators on **Task 2** from PGDL, comparing projective and non-projective variants of the sphere correlation metric. Note that, although their predictive power for generalization is similar, landscapes behave differently in either metric.

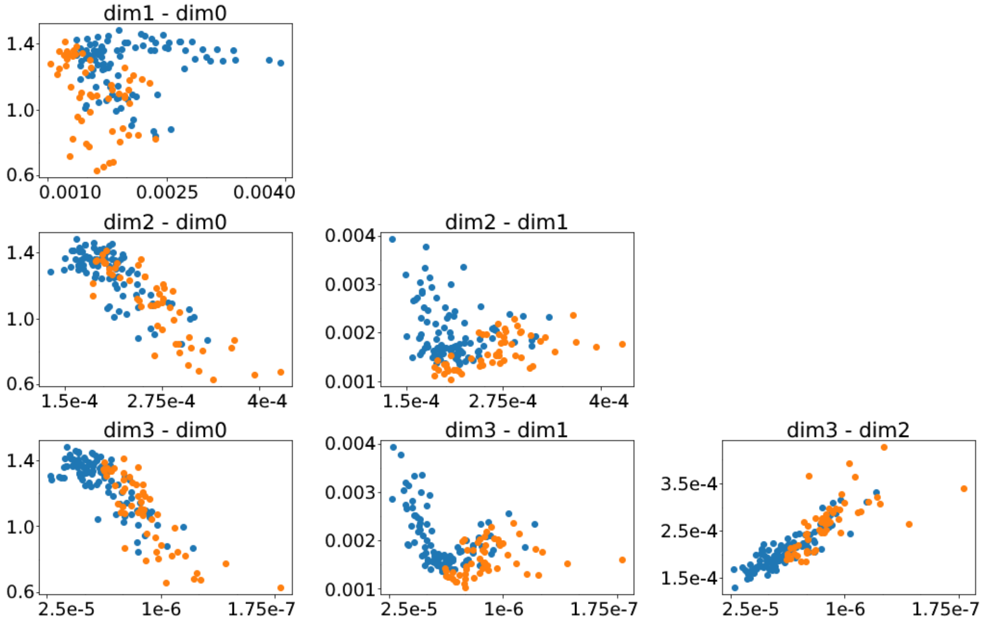


Figure 5.4: Scatter plots of average midlife for each model, segregated by dimension. Each scatter plot titled dim*i* - dim*j* shows the average midlife of *i*-dimensional features against the average midlife of *j*-dimensional features in each model. Blue points represent models from **Task 1** in PGDL, orange points represent models from **Task 2**.

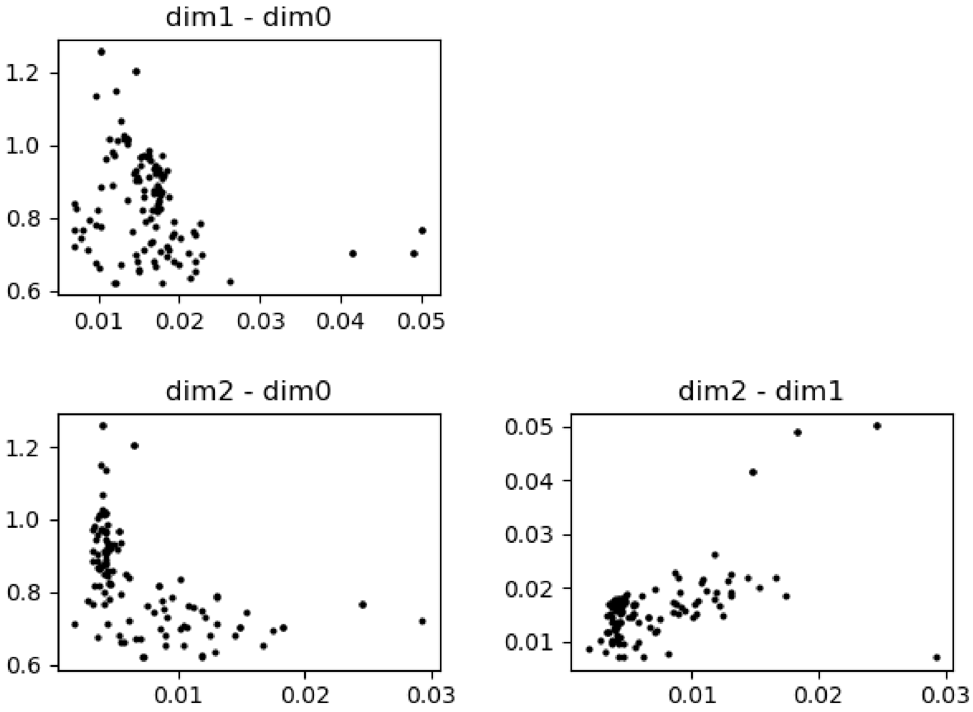


Figure 5.5: Scatter plots of average life for each model in **IMDB**, segregated by dimension. Each scatter plot titled $i - j$ plots the average life of i-dimensional features against the average life of j-dimensional features in each model.

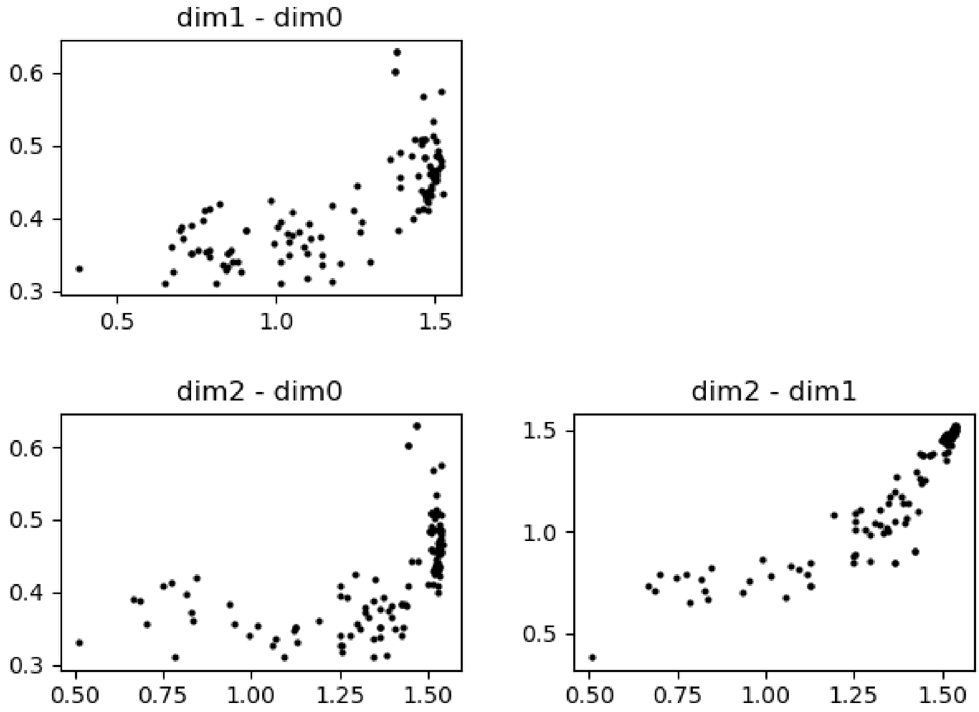


Figure 5.6: Scatter plots of average midlife for each model in **IMDB**, segregated by dimension. Each scatter plot titled $i - j$ plots the average midlife of i-dimensional features against the average midlife of j-dimensional features in each model.

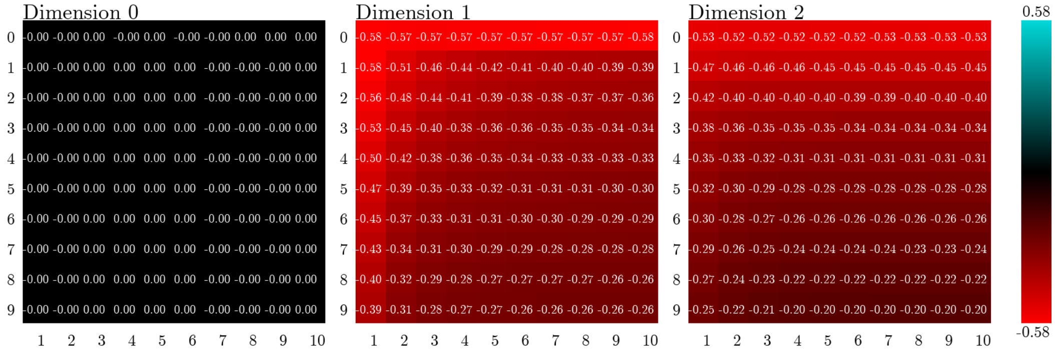


Figure 5.7: Correlation with generalization of the i-th norm of the j-th envelope of the p-dim landscape of each model in the **CIFAR10** model set. i and j vary horizontally and vertically, respectively. Note that envelopes are 0-indexed.

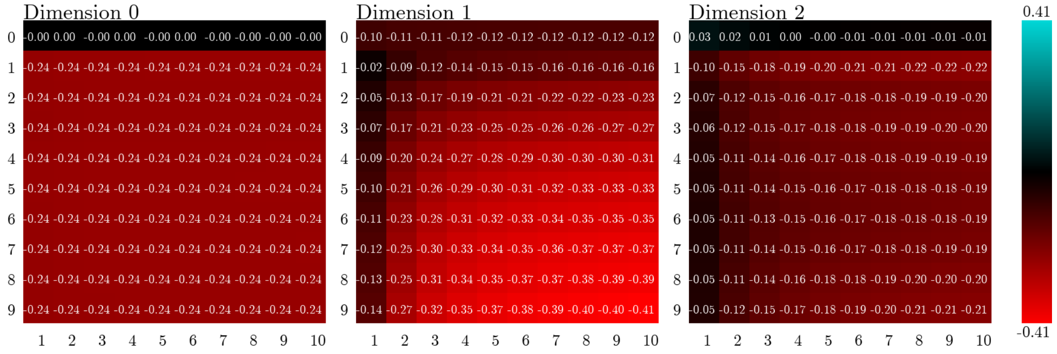


Figure 5.8: Correlation with generalization of the i-th norm of the j-th envelope of the p-dim landscape of each model in the **IMDB** model set. i and j vary horizontally and vertically, respectively. Note that envelopes are 0-indexed.

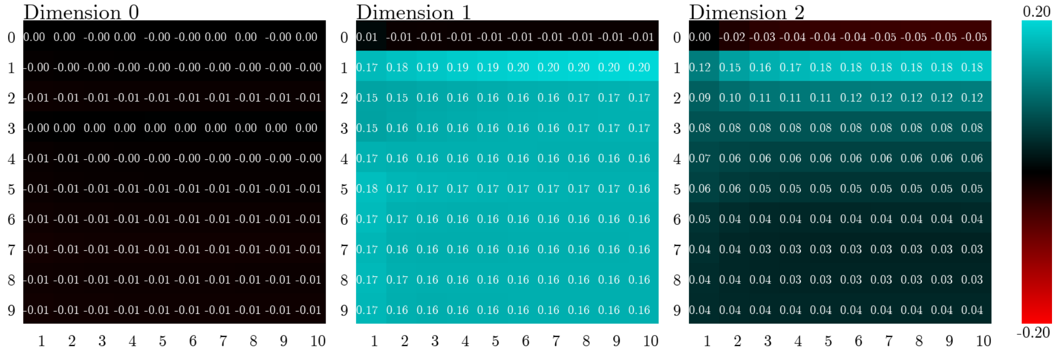
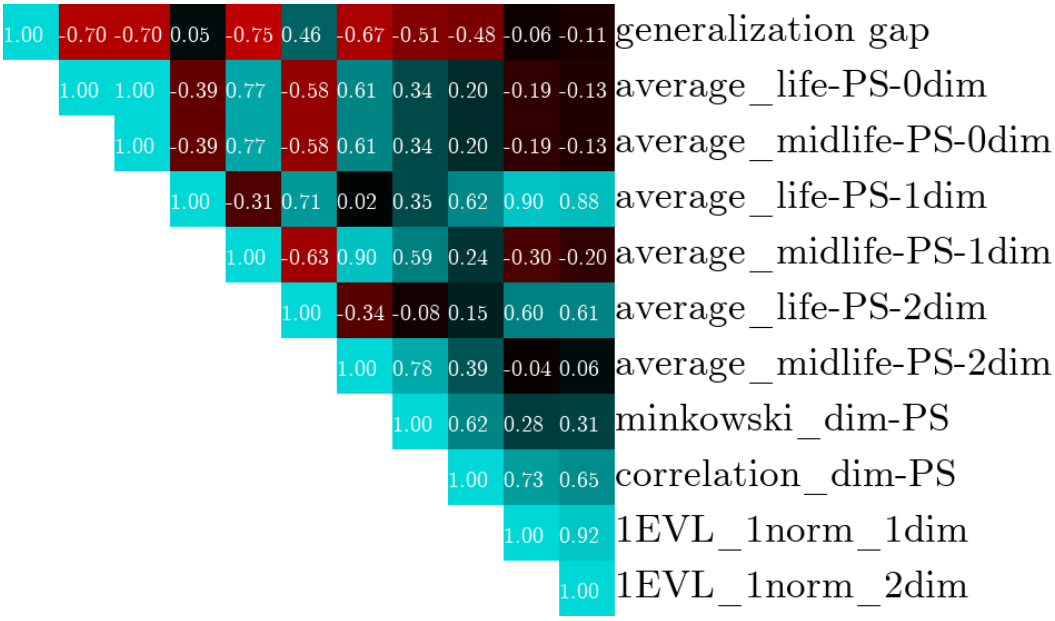
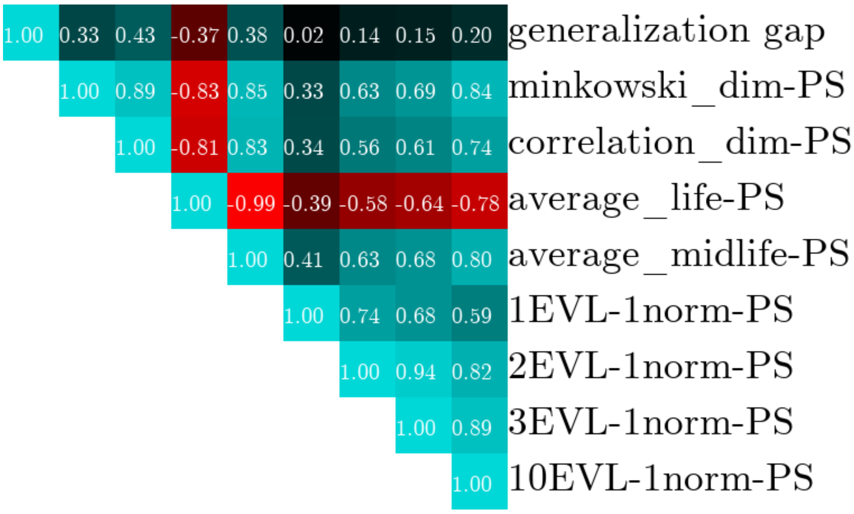


Figure 5.9: Correlation with generalization of the i-th norm of the j-th envelope of the p-dim landscape of each model in **Task 2** from PGDL. i and j vary horizontally and vertically, respectively. Note that envelopes are 0-indexed.



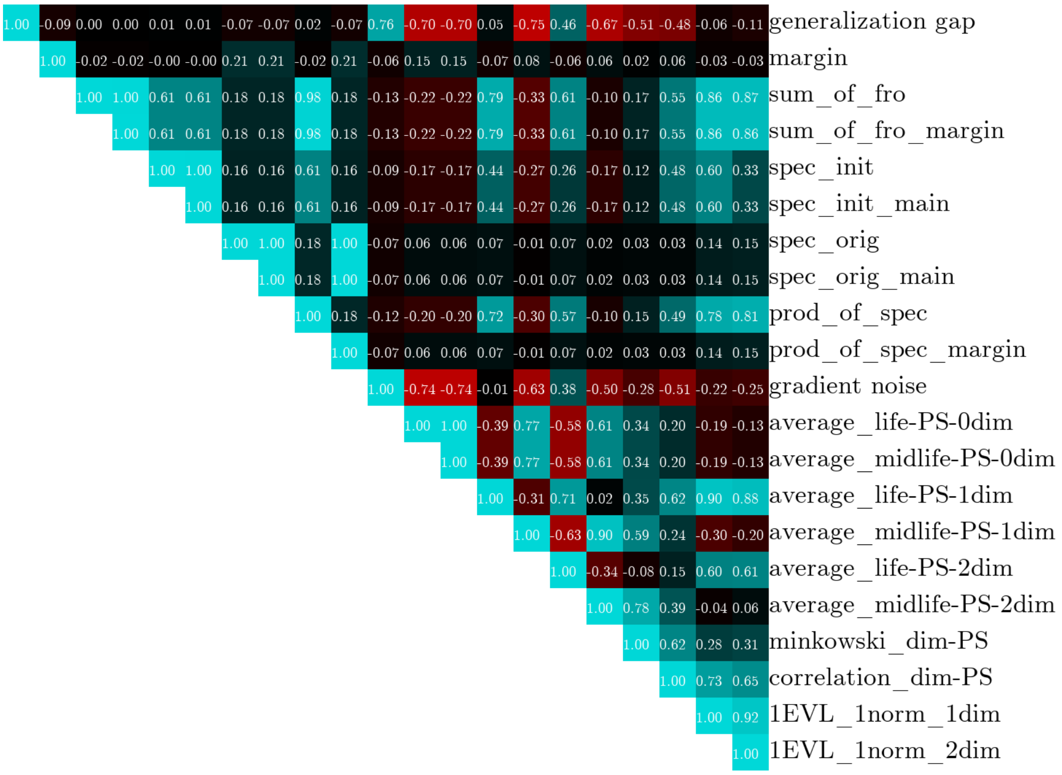
1.00	-0.70	-0.70	0.05	-0.75	0.46	-0.67	-0.51	-0.48	-0.06	-0.11	generalization gap
	1.00	1.00	-0.39	0.77	-0.58	0.61	0.34	0.20	-0.19	-0.13	average_life-PS-0dim
		1.00	-0.39	0.77	-0.58	0.61	0.34	0.20	-0.19	-0.13	average_midlife-PS-0dim
			1.00	-0.31	0.71	0.02	0.35	0.62	0.90	0.88	average_life-PS-1dim
				1.00	-0.63	0.90	0.59	0.24	-0.30	-0.20	average_midlife-PS-1dim
					1.00	-0.34	-0.08	0.15	0.60	0.61	average_life-PS-2dim
						1.00	0.78	0.39	-0.04	0.06	average_midlife-PS-2dim
							1.00	0.62	0.28	0.31	minkowski_dim-PS
								1.00	0.73	0.65	correlation_dim-PS
									1.00	0.92	1EVL_1norm_1dim
										1.00	1EVL_1norm_2dim

Figure 5.10: Correlation matrix with generalization, TDA-based methods, and dimension estimators, for **IMDB**.



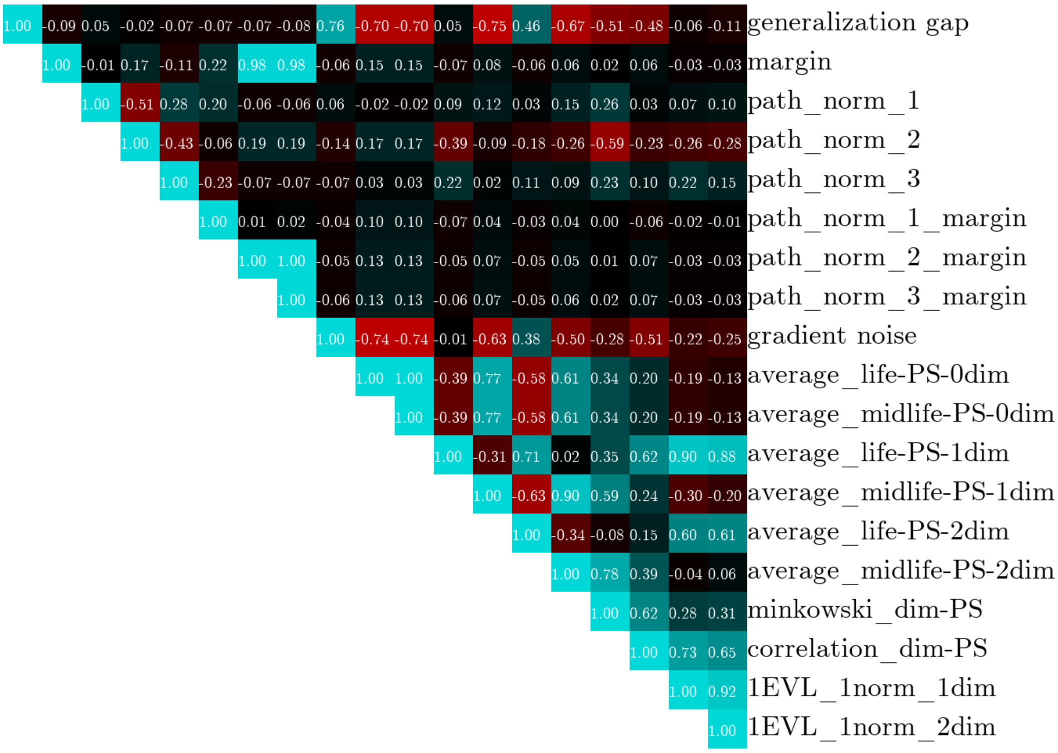
1.00	0.33	0.43	-0.37	0.38	0.02	0.14	0.15	0.20	generalization gap
	1.00	0.89	-0.83	0.85	0.33	0.63	0.69	0.84	minkowski_dim-PS
		1.00	-0.81	0.83	0.34	0.56	0.61	0.74	correlation_dim-PS
			1.00	-0.99	-0.39	-0.58	-0.64	-0.78	average_life-PS
				1.00	0.41	0.63	0.68	0.80	average_midlife-PS
					1.00	0.74	0.68	0.59	1EVL-1norm-PS
						1.00	0.94	0.82	2EVL-1norm-PS
							1.00	0.89	3EVL-1norm-PS
								1.00	10EVL-1norm-PS

Figure 5.11: Correlation matrix with generalization, TDA-based methods, and dimension estimators, for **Task 2**. TDA estimators are taken across dimensions 0–2.



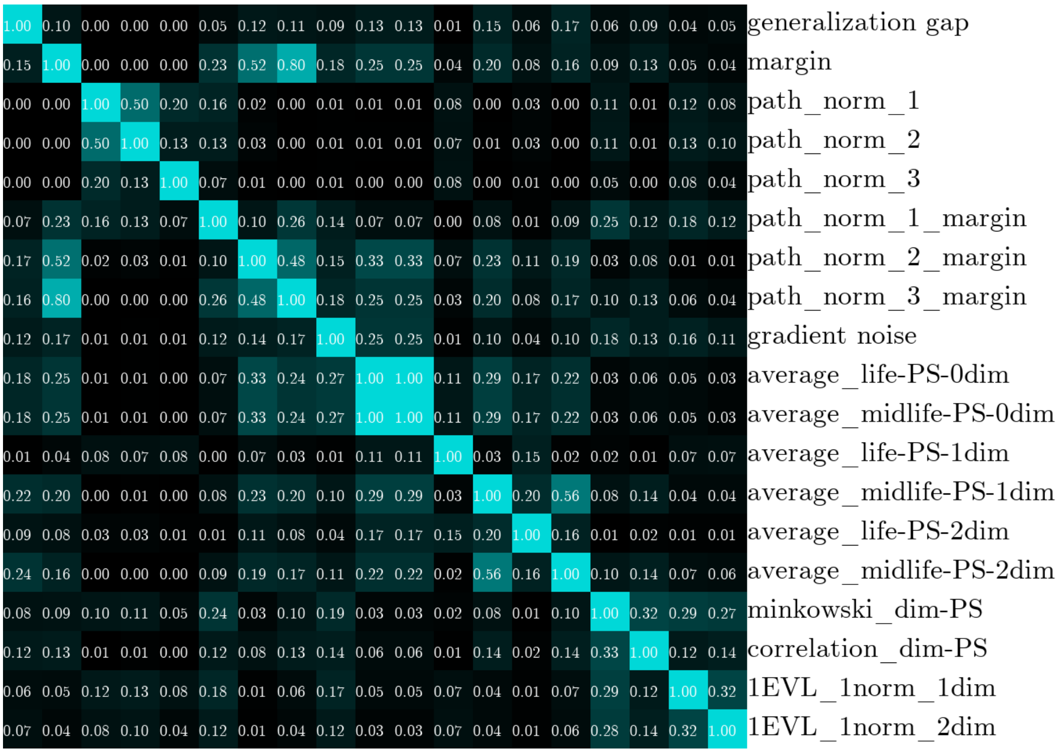
1.00	-0.09	0.00	0.00	0.01	0.01	-0.07	-0.07	0.02	-0.07	0.76	-0.70	-0.70	0.05	-0.75	0.46	-0.67	-0.51	-0.48	-0.06	-0.11	generalization gap
	1.00	-0.02	-0.02	-0.00	-0.00	0.21	0.21	-0.02	0.21	-0.06	0.15	0.15	-0.07	0.08	-0.06	0.06	0.02	0.06	-0.03	-0.03	margin
		1.00	1.00	0.61	0.61	0.18	0.18	0.98	0.18	-0.13	-0.22	-0.22	0.79	-0.33	0.61	-0.10	0.17	0.55	0.86	0.87	sum_of_fro
			1.00	0.61	0.61	0.18	0.18	0.98	0.18	-0.13	-0.22	-0.22	0.79	-0.33	0.61	-0.10	0.17	0.55	0.86	0.86	sum_of_fro_margin
				1.00	1.00	0.16	0.16	0.61	0.16	-0.09	-0.17	-0.17	0.44	-0.27	0.26	-0.17	0.12	0.48	0.60	0.33	spec_init
					1.00	0.16	0.16	0.61	0.16	-0.09	-0.17	-0.17	0.44	-0.27	0.26	-0.17	0.12	0.48	0.60	0.33	spec_init_main
						1.00	1.00	0.18	1.00	-0.07	0.06	0.06	0.07	-0.01	0.07	0.02	0.03	0.03	0.14	0.15	spec_orig
							1.00	0.18	1.00	-0.07	0.06	0.06	0.07	-0.01	0.07	0.02	0.03	0.03	0.14	0.15	spec_orig_main
								1.00	0.18	-0.12	-0.20	-0.20	0.72	-0.30	0.57	-0.10	0.15	0.49	0.78	0.81	prod_of_spec
									1.00	-0.07	0.06	0.06	0.07	-0.01	0.07	0.02	0.03	0.03	0.14	0.15	prod_of_spec_margin
										1.00	-0.74	-0.74	-0.01	-0.63	0.38	-0.50	-0.28	-0.51	-0.22	-0.25	gradient noise
											1.00	1.00	-0.39	0.77	-0.58	0.61	0.34	0.20	-0.19	-0.13	average_life-PS-0dim
												1.00	-0.39	0.77	-0.58	0.61	0.34	0.20	-0.19	-0.13	average_midlife-PS-0dim
													1.00	-0.31	0.71	0.02	0.35	0.62	0.90	0.88	average_life-PS-1dim
														1.00	-0.63	0.90	0.59	0.24	-0.30	-0.20	average_midlife-PS-1dim
															1.00	-0.34	-0.08	0.15	0.60	0.61	average_life-PS-2dim
																1.00	0.78	0.39	-0.04	0.06	average_midlife-PS-2dim
																	1.00	0.62	0.28	0.31	minkowski_dim-PS
																		1.00	0.73	0.65	correlation_dim-PS
																			1.00	0.92	1EVL_1norm_1dim
																				1.00	1EVL_1norm_2dim

Figure 5.12: Correlation matrix with generalization, TDA-based methods, and spectral norm estimators, for **IMDB**.



1.00	-0.09	0.05	-0.02	-0.07	-0.07	-0.07	-0.08	0.76	-0.70	-0.70	0.05	-0.75	0.46	-0.67	-0.51	-0.48	-0.06	-0.11	generalization gap
	1.00	-0.01	0.17	-0.11	0.22	0.98	0.98	-0.06	0.15	0.15	-0.07	0.08	-0.06	0.06	0.02	0.06	-0.03	-0.03	margin
		1.00	-0.51	0.28	0.20	-0.06	-0.06	0.06	-0.02	-0.02	0.09	0.12	0.03	0.15	0.26	0.03	0.07	0.10	path_norm_1
			1.00	-0.43	-0.06	0.19	0.19	-0.14	0.17	0.17	-0.39	-0.09	-0.18	-0.26	-0.59	-0.23	-0.26	-0.28	path_norm_2
				1.00	-0.23	-0.07	-0.07	-0.07	0.03	0.03	0.22	0.02	0.11	0.09	0.23	0.10	0.22	0.15	path_norm_3
					1.00	0.01	0.02	-0.04	0.10	0.10	-0.07	0.04	-0.03	0.04	0.00	-0.06	-0.02	-0.01	path_norm_1_margin
						1.00	1.00	-0.05	0.13	0.13	-0.05	0.07	-0.05	0.05	0.01	0.07	-0.03	-0.03	path_norm_2_margin
							1.00	-0.06	0.13	0.13	-0.06	0.07	-0.05	0.06	0.02	0.07	-0.03	-0.03	path_norm_3_margin
								1.00	-0.74	-0.74	-0.01	-0.63	0.38	-0.50	-0.28	-0.51	-0.22	-0.25	gradient noise
									1.00	1.00	-0.39	0.77	-0.58	0.61	0.34	0.20	-0.19	-0.13	average_life-PS-0dim
										1.00	-0.39	0.77	-0.58	0.61	0.34	0.20	-0.19	-0.13	average_midlife-PS-0dim
											1.00	-0.31	0.71	0.02	0.35	0.62	0.90	0.88	average_life-PS-1dim
												1.00	-0.63	0.90	0.59	0.24	-0.30	-0.20	average_midlife-PS-1dim
													1.00	-0.34	-0.08	0.15	0.60	0.61	average_life-PS-2dim
														1.00	0.78	0.39	-0.04	0.06	average_midlife-PS-2dim
															1.00	0.62	0.28	0.31	minkowski_dim-PS
																1.00	0.73	0.65	correlation_dim-PS
																	1.00	0.92	1EVL_1norm_1dim
																		1.00	1EVL_1norm_2dim

Figure 5.13: Correlation matrix with generalization, TDA-based methods, and path norms, for **IMDB**.



1.00	0.10	0.00	0.00	0.00	0.05	0.12	0.11	0.09	0.13	0.13	0.01	0.15	0.06	0.17	0.06	0.09	0.04	0.05	generalization gap
0.15	1.00	0.00	0.00	0.00	0.23	0.52	0.80	0.18	0.25	0.25	0.04	0.20	0.08	0.16	0.09	0.13	0.05	0.04	margin
0.00	0.00	1.00	0.50	0.20	0.16	0.02	0.00	0.01	0.01	0.01	0.08	0.00	0.03	0.00	0.11	0.01	0.12	0.08	path_norm_1
0.00	0.00	0.50	1.00	0.13	0.13	0.03	0.00	0.01	0.01	0.01	0.07	0.01	0.03	0.00	0.11	0.01	0.13	0.10	path_norm_2
0.00	0.00	0.20	0.13	1.00	0.07	0.01	0.00	0.01	0.00	0.00	0.08	0.00	0.01	0.00	0.05	0.00	0.08	0.04	path_norm_3
0.07	0.23	0.16	0.13	0.07	1.00	0.10	0.26	0.14	0.07	0.07	0.00	0.08	0.01	0.09	0.25	0.12	0.18	0.12	path_norm_1_margin
0.17	0.52	0.02	0.03	0.01	0.10	1.00	0.48	0.15	0.33	0.33	0.07	0.23	0.11	0.19	0.03	0.08	0.01	0.01	path_norm_2_margin
0.16	0.80	0.00	0.00	0.00	0.26	0.48	1.00	0.18	0.25	0.25	0.03	0.20	0.08	0.17	0.10	0.13	0.06	0.04	path_norm_3_margin
0.12	0.17	0.01	0.01	0.01	0.12	0.14	0.17	1.00	0.25	0.25	0.01	0.10	0.04	0.10	0.18	0.13	0.16	0.11	gradient noise
0.18	0.25	0.01	0.01	0.00	0.07	0.33	0.24	0.27	1.00	1.00	0.11	0.29	0.17	0.22	0.03	0.06	0.05	0.03	average_life-PS-0dim
0.18	0.25	0.01	0.01	0.00	0.07	0.33	0.24	0.27	1.00	1.00	0.11	0.29	0.17	0.22	0.03	0.06	0.05	0.03	average_midlife-PS-0dim
0.01	0.04	0.08	0.07	0.08	0.00	0.07	0.03	0.01	0.11	0.11	1.00	0.03	0.15	0.02	0.02	0.01	0.07	0.07	average_life-PS-1dim
0.22	0.20	0.00	0.01	0.00	0.08	0.23	0.20	0.10	0.29	0.29	0.03	1.00	0.20	0.56	0.08	0.14	0.04	0.04	average_midlife-PS-1dim
0.09	0.08	0.03	0.03	0.01	0.01	0.11	0.08	0.04	0.17	0.17	0.15	0.20	1.00	0.16	0.01	0.02	0.01	0.01	average_life-PS-2dim
0.24	0.16	0.00	0.00	0.00	0.09	0.19	0.17	0.11	0.22	0.22	0.02	0.56	0.16	1.00	0.10	0.14	0.07	0.06	average_midlife-PS-2dim
0.08	0.09	0.10	0.11	0.05	0.24	0.03	0.10	0.19	0.03	0.03	0.02	0.08	0.01	0.10	1.00	0.32	0.29	0.27	minkowski_dim-PS
0.12	0.13	0.01	0.01	0.00	0.12	0.08	0.13	0.14	0.06	0.06	0.01	0.14	0.02	0.14	0.33	1.00	0.12	0.14	correlation_dim-PS
0.06	0.05	0.12	0.13	0.08	0.18	0.01	0.06	0.17	0.05	0.05	0.07	0.04	0.01	0.07	0.29	0.12	1.00	0.32	1EVL_1norm_1dim
0.07	0.04	0.08	0.10	0.04	0.12	0.01	0.04	0.12	0.03	0.03	0.07	0.04	0.01	0.06	0.28	0.14	0.32	1.00	1EVL_1norm_2dim

Figure 5.14: Normalized mutual information matrix with generalization, TDA-based methods, and path norms, for **IMDB**.