Topological Tools in Data Analysis

Tadas Temčinas

University of Oxford

A dissertation submitted for the degree of

MMathPhil in Mathematics and Philosophy

Hilary Term 2018

Acknowledgements

First and foremost, I would like to thank my supervisor Vidit Nanda, who has been extremely helpful, approachable and supportive during the course of this dissertation. I would like to thank my family for their love and support, as well as my friends Michelle Liu, Mantas Pajarskas, Andrius Ovsianas, Tadas Kriščiūnas, Aidas Kilda, and Tomas Vaškevičius for their encouragement and countless conversations. There is no doubt that without these people this work would not be in its current state, for which I am very grateful.

Contents

Notation

$\mathbf{C}^{op}$ Oopposite category of the category $\mathbf{C}$

$\mathbf{Ab}$ Category of abelian groups and abelian group homomorphisms

$\mathbf{Ch(A)}$ Category of chain complexes in an abelian category $\mathbf{A}$ and chain maps

$\mathbb{Z}[A]$ Free $\mathbb{Z}$-module with basis A

$H_\bullet$ Homology of a top. space with integral coefficients or a chain complex

$\tilde{H}_\bullet$ Reduced homology

$H^\bullet$ Cohomology of a top. space with integral coefficients or a chain complex

$H_\bullet(X, A)$ Relative homology of a pair of topological spaces

$H_\bullet(X; G)$ Homology of a top. space X with coefficients in an abelian group G

$\overline{A}$ Closure of a set A in a topological space

∂A Boundary of a set A in a topological space

$\oplus$ Direct sum

$\otimes$ Tensor product

$v_{\text{mathematics}}$ Word vector of the word 'mathematics' with respect to a word embedding

$|A|$ Cardinality of a set A

$\mathcal{P}(A)$ Powerset of a set A

Chapter 1

Introduction

It is not hard to believe that today the amount of data the world generates is larger and more complex than ever before. With that much information comes a natural question of how to use it. In the last 10-15 years mathematicians have realised that topology might be a useful framework for the analysis of complex data, making the term 'applied topology' no longer an oxymoron [11, 4, 3].

There are a few intuitive reasons why topological data analysis (TDA) can be useful. Topology provides a framework which is:

- Agnostic to a metric being used (only the induced topology matters).[1]
- Robust and not as susceptible to bounded noise.
- Concerned with qualitative properties.
- Well studied and understood by pure mathematicians.

Since TDA has become a very active and broad area of research, by no means does this dissertation aim to cover all (or even most) of TDA. Tools that will be discussed in this work are those leveraging local topological structure, more concretely – local homology. On the practical side, implementations of such tools very often can easily take advantage of the power of distributed computing. On the more theoretical side, we will see that those techniques can help detect singularities in data, dimensionality of the dataset as well as cluster the datapoints according to decomposition of the space into manifold-like pieces (called strata).

[1] This is particularly useful in the case, which happens often in practice, when the datapoints lie in $\mathbb{R}^n$ since any two metrics on $\mathbb{R}^n$ are equivalent and hence induce the same topology.

Word embedding is a collective term for ways to represent words of a natural language as vectors in a high-dimensional real vector space. We will see in the following chapters that datasets coming from word embeddings have interesting topology. Hence natural language processing (NLP) seems to be a natural domain of TDA applications. Despite this fact, only a few attempts at using TDA techniques to analyse language data have been published [30, 33, 19, 28]. One of the aims of this dissertation is to contribute to closing the gap between TDA and NLP by applying TDA techniques to data from NLP.

Intuitively, stratification is a decomposition of a topological space into manifold-like pieces. When thinking about stratification learning and word embeddings, it seems intuitive that vectors of words corresponding to the same broad topic would constitute a structure, which we might hope to be a manifold. Hence, for example, by looking at the intersections between those manifolds or singularities on the manifolds (both of which can be recovered using stratification learning algorithms) one might hope to find vectors of homonyms like 'bank' (which can mean either a river bank, or a financial institution) or vectors of words with very different meanings like 'cancer' (which can refer to the Cancer constellation or to the illness). This, in turn, has potential to help solve the word sense disambiguation (WSD) problem in NLP, which is pinning down a particular meaning of a word used in a sentence when the word has multiple meanings.

In this dissertation we will:

- Introduce TDA in a self-contained fashion with a focus on tools based on local homology and stratification.
- Apply the discussed tools to word embedding data and present the results.
- Provide an implementation of some of the discussed tools.

In this dissertation we assume that the reader is familiar with some basic category theory, e.g. [18, Chapters 1 and 2], basic homological algebra, e.g. [31, Chapter 1] as well as algebraic topology, e.g. [15, Chapters 0, 2, 3]. In a few places we mention persistent homology, particularly when speaking of future work but since it is not central to the main part of the work, we do not include the definition. For those who are not familiar, an introduction aimed at general computational scientists can be found in [25, 26].

The main body of this work is comprised of four chapters (Chapters 2-5):

- In Chapter 2 we will briefly present the mathematics that we will use throughout the work.
- In Chapter 3 we will briefly present word embeddings, focusing on Word2Vec (the skip-gram version) and GloVe models.
- In Chapter 4 we will discuss a clustering algorithm based on local homology, its implementation and computational results on datasets coming from word embeddings.
- In Chapter 5 we will discuss how this algorithm can be extended to a stratification learning algorithm, and how stratification might be helpful to solve WSD.

Chapter 2

Background mathematical material

2.1 Local homology

Let us remind ourselves that given a CW complex X we can define a partial order on its cells. By definition of a CW complex we know that if σ, τ are cells such that $\sigma \cap \overline{\tau} \neq \emptyset$, then we have $\sigma \subseteq \overline{\tau}$ and we say that σ is a face of τ, and τ is a co-face of σ. Write $\sigma \leq \tau$ iff σ is a face of τ. This relation defines a partial order on the set of cells of X. Now we can define the star and link of a cell by $\mathrm{st}(\sigma) = \{\, \tau \in X \mid \sigma \leq \tau \,\}$ and $\mathrm{lk}(\sigma) = \{\, \tau \in X \mid \exists \rho \in X.\, \tau,\, \sigma \leq \rho \wedge \nexists \rho' \in X.\, \rho' \leq \tau,\, \sigma \,\}$ respectively, examples of which can be seen in Figures 2.1, 2.2.

Definition 2.1. Let X be a CW complex and let σ be a cell in X. The local homology of σ in X is $H_\bullet^\sigma = H_\bullet(\overline{\mathrm{st}(\sigma)}, \partial\, \mathrm{st}(\sigma))$. Rank of the free part of H_n^σ is called the n-th local betti number of σ in X.

Definition 2.2. A CW complex X is called regular iff the attaching map for each cell is a homeomorphism.

Figure 2.1: The star (green, right) of a vertex (yellow, left) in a simplicial complex. Taken from [7].

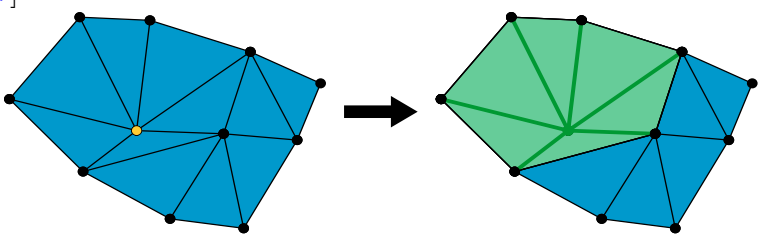
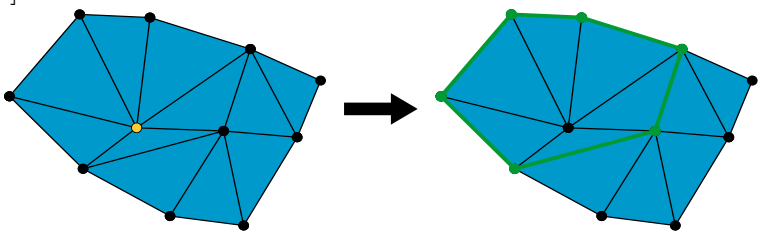


Figure 2.2: The link (green, right) of a vertex (yellow, left) in a simplicial complex. Taken from [6].



The lemmas that we prove in this section are in the setting of finite simplicial complexes since in the end when we compute homology for the applications, we use simplicial complexes. However, the analogous lemmas hold in a more general setting of finite regular CW complexes.

We remind ourselves that given a finite simplicial complex X we can orient it by defining a total order on the vertices $\{v_0, v_1, \ldots, v_N\}$.[1] We can specify an n-simplex σ by an $(n+1)$-tuple of vertices $(v_{\sigma_0}, v_{\sigma_1}, \ldots, v_{\sigma_n})$ where $\sigma_i < \sigma_{i+1}$. Also, let $(v_{\sigma_0}, v_{\sigma_1}, \ldots, \hat{v_{\sigma_i}}, \ldots, v_{\sigma_n})$ be the face of σ specified by the vertices $\sigma \setminus \{v_{\sigma_i}\}$. Now to calculate local homology of a simplex, we can take star of the simplex, pretend that it is a simplicial complex and then calculate its simplicial homology. This is formulated more precisely in the following lemma:

Lemma 2.1. *Take a simplex σ in a finite simplicial complex X. Let $S_n(\sigma) = \{\, \tau \subseteq \sigma \mid |\tau| = n+1 \,\}$. Let $C_n^\sigma = \mathbb{Z}[S_n]$ and define $\partial_n : C_n^\sigma \to C_{n-1}^\sigma$ on the basis elements: $\partial_n(\tau) = \sum_{i=0}^{n} (-1)^i (v_{\tau_0}, v_{\tau_1}, \ldots, \hat{v_{\tau_i}}, \ldots, v_{\tau_n})$. Set $(v_{\tau_0}, v_{\tau_1}, \ldots, \hat{v_{\tau_i}}, \ldots, v_{\tau_n}) = 0$ iff the simplex specified by $(v_{\tau_0}, v_{\tau_1}, \ldots, \hat{v_{\tau_i}}, \ldots, v_{\tau_n})$ is not in S_{n-1}. Then we have that $(C_\bullet^\sigma, \partial)$ is a chain complex and $H_\bullet(C_\bullet^\sigma) = H_\bullet^\sigma$.*

Proof. By definition, 2.1: $H_\bullet^\sigma = H_\bullet(\overline{\mathrm{st}(\sigma)}, \partial\,\mathrm{st}(\sigma))$. Clearly $\partial\,\mathrm{st}(\sigma)$ is a non-empty closed subspace of $\overline{\mathrm{st}(\sigma)}$ that is a deformation retract of a neighbourhood in $\overline{\mathrm{st}(\sigma)}$. Hence, by [15, Proposition 2.22] we have that $H_\bullet(\overline{\mathrm{st}(\sigma)}, \partial\,\mathrm{st}(\sigma)) = \tilde{H}_\bullet(\overline{\mathrm{st}(\sigma)}/\partial\,\mathrm{st}(\sigma))$. Now $\overline{\mathrm{st}(\sigma)}/\partial\,\mathrm{st}(\sigma) = \mathrm{st}(\sigma)$ and hence the result follows.

We do not need to worry that in the equation $H_\bullet(\overline{\mathrm{st}(\sigma)}, \partial\,\mathrm{st}(\sigma)) = \tilde{H}_\bullet(\overline{\mathrm{st}(\sigma)}/\partial\,\mathrm{st}(\sigma))$ one side has reduced homology and the other not because we will have non-trivial 0th local homology only when σ is a 0-simplex and $\mathrm{st}(\sigma) = \{\sigma\}$, in which case the result is trivial. □

[1] We do this when calculating simplicial homology.

Remark. Instead of taking the usual homology in the Definition 2.1 we could take homology with coefficients in an abelian group G and then we would get a definition of local homology with coefficients in G. In this setting the previous lemma holds just as well if we take $C_n^\sigma = \mathbb{Z}[S_n] \otimes G$.

The following lemma shows that if we think about a simplicial complex as a generalisation of a graph (on the basis that 1-skeleton of a simplicial complex is an undirected graph without self-loops or multiple edges between the same vertices), then we can think about local homology as a generalisation of a degree of a vertex. Hopefully this will provide some intuition of how local homology works.

Lemma 2.2. *Let X be a 1-dimensional finite simplicial complex. For any 0-simplex $v \in X$ let* $\deg(v) = |\{\, e \in X \mid v \in e \,\}|$. *Then for any 0-simplex v:*

$$H_n^v = \begin{cases} \mathbb{Z}^{\deg(v)-1} & n = 1 \\ 0 & \textit{otherwise} \end{cases}$$

Proof. Let us use Lemma 2.1 to do the calculation.

Pick any 0-simplex v and let $E(v) = \{\, e \in X \mid v \in e \,\}$. Then $\mathrm{st}(v) = \{v\} \cup E(v)$ so the associated chain complex is:

$$0 \to \mathbb{Z}[E(v)] \to \mathbb{Z}[\{v\}] \to 0$$

Now we can pick an orientation on the simplicial complex such that for all $e \in E(v)$ we have $\partial e = v$. Then we can clearly see that $\ker \partial_0 \cong \operatorname{im} \partial_1 \cong \mathbb{Z}$ and so we get $H_0^v = 0$. Also, by Rank-Nullity theorem, $\ker \partial_1$ has dimension $\deg(v) - 1$. Also, $\operatorname{im} \partial_2 = 0$ and so we have $H_1^v = \mathbb{Z}^{\deg(v)-1}$. All the other homology groups are clearly zero because the complex is non-zero only in degree 0 and 1. □

The following lemma characterises the local homology in terms of the link of a simplex instead of its star. We will use this viewpoint when analysing results in Chapter 4.

Lemma 2.3. *If X is a finite simplicial complex and σ is a k-simplex, then for $n \geq k+1$ we have $H_n^\sigma = \tilde{H}_{n-k-1}(\mathrm{lk}(\sigma))$.*

Proof. Pick any k-simplex $\sigma \in X$. Let $\mathrm{lk}_i(\sigma) := \{\, \tau \in \mathrm{lk}(\sigma) \mid |\tau| = i+1 \,\}$ and $\mathrm{st}_i(\sigma) := \{\, \tau \in \mathrm{st}(\sigma) \mid |\tau| = i+1 \,\}$.

For any $i \geq 0$ define a map $f_i : \mathrm{lk}_i(\sigma) \to \mathrm{st}_{i+k+1}(\sigma)$ by $x \mapsto x \cup \sigma$. Note that for $i < k$, $\mathrm{st}_i(\sigma) = \emptyset$ and $\mathrm{st}_k(\sigma) = \{\sigma\}$.

For all $i \geq 0$ the map f_i is well-defined: pick $x \in \mathrm{lk}_i(\sigma)$. By definition of the link, x and σ do not share a face and so $x \cap \sigma = \emptyset$. Therefore if x is an i-simplex, $x \cup \sigma$ is a $i + k + 1$-simplex, which is clearly in the star as $\sigma \subseteq x \cup \sigma$.

For all $i \geq 0$ the map f_i is injective: assume for $x, y \in \mathrm{lk}_i(\sigma)$ we have $f_i(x) = f_i(y)$. Then $x \cup \sigma = y \cup \sigma$. Now as $x \cap \sigma = \emptyset$ and $y \cap \sigma = \emptyset$ we have that $(x \cup \sigma) \setminus \sigma = x$ and $(y \cup \sigma) \setminus \sigma = y$. Therefore $x = (x \cup \sigma) \setminus \sigma = (y \cup \sigma) \setminus \sigma = y$.

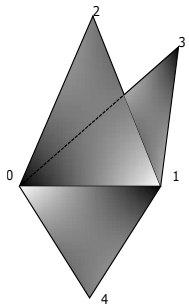
For all $i \geq 0$ the map f_i is surjective: pick any $y \in \mathrm{st}_{i+k+1}(\sigma)$. By definition of a star, $\sigma \subseteq y$ and so define $x := y \setminus \sigma$. Clearly x is an i-simplex and y is a co-face x and σ share. By construction of x, there cannot be any any common face of x and σ. Hence, $x \in \mathrm{lk}_i(\sigma)$ and $f_i(x) = y$.

From Lemma 2.1 we know that $H_\bullet(C^\sigma_\bullet) = H^\sigma_\bullet$. By linearly extending the bijections f_i, we get isomorphisms at the level of chain complexes for all $i \geq 0$: $C^\sigma_{i+k+1} = \mathbb{Z}[\mathrm{st}_{i+k+1}(\sigma)] \cong \mathbb{Z}[\mathrm{lk}_i(\sigma)] = C_i(\mathrm{lk}(\sigma))$ where $C_i(\mathrm{lk}(\sigma))$ is the group of i-th simplicial chains of the link. It is easy to see the induced isomorphisms commute with the boundary maps and hence induce isomorphism on the homology groups. The only thing is that we do not have a bijection from something in the link onto $\mathrm{st}_k(\sigma) = \{\sigma\}$ and therefore on the right hand side of $H^\sigma_n = \tilde{H}_{n-k-1}(\mathrm{lk}(\sigma))$ we have the reduced homology. □

Remark. Note that since we have bijection at the level of simplices, the above lemma also holds if we calculate local homology with coefficients in an abelian group G. It is only that instead of applying the usual chain functor to the maps f_i we need to apply the chain functor with coefficients in G.

Example. *Let X_k be a simplicial complex that is k triangles glued on a common edge. X_k is the simplicial closure of $\{\,\{0, 1, i+2\} \mid i \in \{0, 1, \ldots, k-1\}\,\}$. In Figure 2.3 we can see X_3. Let $\sigma := \{0, 1\}$ be the common edge. Then $\mathrm{lk}(\sigma) = \{\{2\}, \{3\}, \ldots, \{k+1\}\}$ consists of k 0-simplices. In this case it is easy to calculate the homology of the link - it has k connected components hence 0th homology will be $\mathbb{Z}^k$. Since there are no other simplices but the 0-dimensional ones, all the higher homology groups are trivial. Therefore by Lemma 2.3 we have $H^\sigma_2 = \mathbb{Z}^{k-1}$ and all the other local homology groups being trivial.*

Figure 2.3: X_3



Also, we will make use of the following definition later on.

Definition 2.3. Given a topological space X, a cone over X is a space $C(X) = (X \times [0,1])/(X \times \{0\})$ and an open cone over X is a space $OC(X) = (X \times [0,1))/(X \times \{0\})$.

2.2 Applied topology

The most basic but probably most important question in applied topology is how to construct a finite simplicial complex given a finite set of points in a metric space. We assume that the finite set of points is sampled from some topological space, and we aim to construct a simplicial complex that would recover some of the structure of the underlying topological space. In this section we discuss a few ways to construct such simplicial complexes. For a more detailed account we refer to [10, Chapter 3].

Definition 2.4. An undirected graph is a tuple $G = (V, E)$ where V is a countable set and $E \subseteq \{ X \in \mathcal{P}(V) \mid |X| = 2 \}$, and for all $v \in V$ we have $(v, v) \notin E$. We call V the set of vertices and we call E the set of edges.

Remark. Note that this definition disallows "self-loops" (i.e. edges from a vertex to itself) and more than one edge between a pair of vertices.

Example. *1-skeleton of any countable simplicial complex is an undirected graph. In fact, any graph can be realised as a 1-skeleton of a countable simplicial complex so the two notions are equivalent.*

In this section whenever we speak of a graph, we mean an undirected graph in the sense of Definition 2.4.

Definition 2.5. Given a graph $G = (V, E)$ a path between vertices v and w is a finite sequence of edges $e_1, e_2, \ldots, e_n$ such that $v \in e_1$, $w \in e_n$ and for all i, $|e_i \cap e_{i+1}| = 1$.

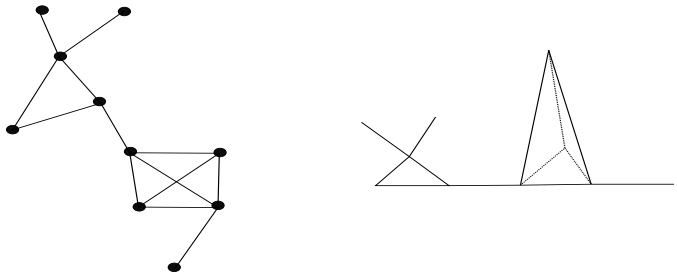
Definition 2.6. Given a graph G an n-clique in G is a subgraph $G' = (V', E')$ such that $|V'| = n$ and $E' = \{\, \{v, w\} \in V' \mid v \neq w \,\}$.

We have already seen that a graph is a 1-dimensional simplicial complex. So given a simplicial complex, it is easy to associate a graph to it - we can just take the 1-skeleton. The following definition shows how one can do the reverse: given a graph associate a simplicial complex to it.

Definition 2.7. Let $G = (V, E)$ be a graph. A clique complex associated to the graph is a simplicial complex $C(G)$ such that $\{v_0, v_1, \ldots, v_n\} \subseteq V$ is an n-simplex of $C(G)$ iff there is an $(n+1)$-clique (V', E') in G such that $V' = \{v_0, v_1, \ldots, v_n\}$.

Remark. Note that the 1-skeleton of $C(G)$ is the graph G itself.

Figure 2.4: A graph (left) and its clique complex (right). The triangles and the tetrahedron are filled in.



Now let us turn to the main topic of this section - constructing a simplicial complex out of a finite dataset, which lies in a metric space.

Definition 2.8. A finite subset of points in a metric space (M, d) is called a point cloud in (M, d).

Definition 2.9. Let S be a point cloud in some metric space (M, d) and fix $\epsilon \in \mathbb{R}_{\geq 0}$. An ϵ-radius neighbourhood graph associated to a point cloud S is a graph $G_\epsilon(S) = (V, E)$ where $V = S$ and $\{v, w\} \in E \iff v \neq w \wedge d(v, w) \leq \epsilon$.

Definition 2.10. We say that two points x, y in a point cloud S are ϵ-neighbours for some $\epsilon \in \mathbb{R}_{\geq 0}$ iff $x \neq y$ and $d(x, y) \leq \epsilon$ (i.e. there is an edge between x and y in $G_\epsilon(S)$).

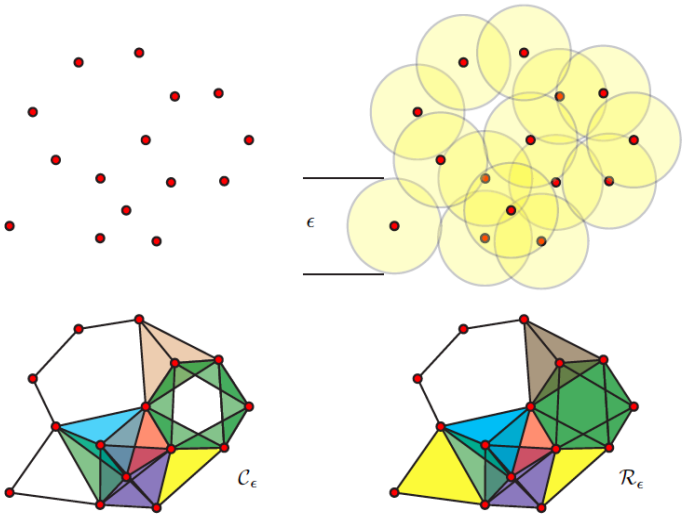
Definition 2.11. Given a point cloud S in (M, d), a Vietoris-Rips complex (VR-complex) associated to S and $\epsilon \in \mathbb{R}_{\geq 0}$ is the clique complex of the ϵ-radius neighbourhood graph of S. We denote this complex as $\mathrm{VR}_\epsilon(S)$.

There is another way to associate a simplicial complex to a point cloud, which is better from a theoretical point of view.

Given a point x in a metric space (M, d) we write $B_\epsilon(x) = \{\, y \in M \mid d(x, y) \leq \epsilon \,\}$.

Definition 2.12. Given a point cloud S in (M, d) and $\epsilon \in \mathbb{R}_{\geq 0}$, the Čech complex is defined as $\check{C}_\epsilon(S) = \left\{\, \sigma \subseteq S \;\middle|\; \bigcap_{x \in \sigma} B_{\frac{\epsilon}{2}}(x) \neq \emptyset \,\right\}$.

Figure 2.5: A point cloud (top left) and its Čech (bottom left) and VR (bottom right) complexes. Figure from [13].



The following two results justify VR and Čech complexes and their use in applications.

Theorem 2.1 (**Nerve Theorem**). *Assume a metric space (M, d) and $\epsilon \in \mathbb{R}_{\geq 0}$ are such that for any finite $X \subseteq M$ the set $\bigcap_{x \in X} B_\epsilon(x)$ is contractible. Then given a point cloud S, the Čech complex $\check{C}_\epsilon(S)$ is homotopy equivalent to $\bigcup_{s \in S} B_\epsilon(s)$.*

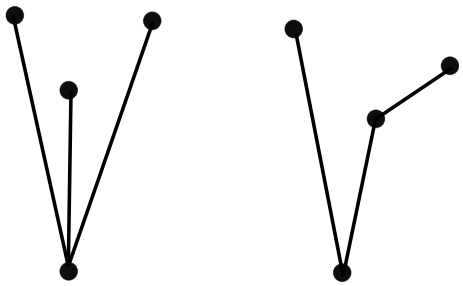
Proof. This is a special case of [15, Corollary 4G.3.]. □

Lemma 2.4 (Vietoris-Rips Lemma). *Assume (M, d) is a metric space and $\epsilon \in \mathbb{R}_{\geq 0}$. Then for any finite point cloud S in the metric space, we have the following inclusions:*

$$\check{C}_\epsilon(S) \subseteq VR_{2\epsilon}(S) \subseteq \check{C}_{2\epsilon}(S)$$

Remark. Note that Theorem 2.1 does not give any guarantee for local homology since it is possible for two spaces to have the same homotopy type but different local homologies of points. Consider two graphs: $G_1 = (\{0,1,2,3\}, \{\{0,1\},\{0,2\},\{0,3\}\})$ and $G_2 = (\{0,1,2,3\}, \{\{0,1\},\{0,2\},\{2,3\}\})$ as seen in Figure 2.6. Taking topological realisation of the two graphs regarded as simplicial complexes we get the corresponding spaces X_1 and X_2, both of which are contractible and hence have the same homotopy type. However, in G_1 we have one vertex with degree 3 and 3 vertices with degree 1 but in G_2 we have 2 vertices with degree 1 and 2 vertices with degree 2 and so by Lemma 2.2 the local homologies differ.

Figure 2.6: The graphs G_1 (left) and G_2 (right).



When thinking about implementation of algorithms, the VR-complex is computationally cheaper to construct but the Čech complex is carries a massive theoretical guarantee from Theorem 2.1. However, we can interpret Lemma 2.4 as a result saying that the VR-complex approximates the Čech complex[2] and so we can justifiably use the VR-complex, which is computationally cheaper.

When thinking from a homological point of view, a drawback that VR and Čech complexes have is that their dimension can be higher than the highest dimension in which the space we assume the point cloud is sampled from has non-trivial homology. So it is possible that VR or Čech complexes have non-trivial homology where it

[2]More precisely, speaking in terms of homology, we see that the map $H_n(\check{C}_\epsilon(S)) \to H_n(\check{C}_{2\epsilon}(S))$ induced by inclusion factors through $H_n(\mathrm{VR}_{2\epsilon}(S))$ for every n.

actually should be trivial. In practice, if we know the dimensionality of the space from which a point cloud is sampled n, we just look at the n-skeleton of VR or Čech complexes.

2.3 Sheaf theory

Intuitively, a sheaf over a topological space associates an object in a category to every open set of the space. Here, as usual, keeping our focus on applications, we introduce the theory of sheaves over a topological space as well as their "discrete counterpart" – cellular sheaves over CW complexes. For a more detailed introduction and background we refer to Curry's PhD thesis [9, Chapter 2, 4].

Let us remind ourselves that the notion of a category such that between any pair of objects there is at most one morphism and isomorphic objects are equal, and the notion of a partially ordered set are equivalent: there is a morphism $x \to y$ iff $x \leq y$.

Definition 2.13. If X is a CW complex, we write $\mathbf{Fc(X)}$ for the poset of cells of X with respect to the face relation,[3] viewed as a category.

Definition 2.14. A cellular sheaf over a CW complex X in a category $\mathbf{C}$ is a covariant functor $F : \mathbf{Fc(X)} \to \mathbf{C}$. Dually, a cellular cosheaf over a CW complex X in a category $\mathbf{C}$ is a contravariant functor $\hat{F} : \mathbf{Fc(X)}^{op} \to \mathbf{C}$.

Remark. Since a cellular sheaf (or a cosheaf) over X is nothing else but a functor between two categories, all cellular sheaves (or cosheaves) over X form a category which objects are cellular sheaves (or cosheaves) and morphisms are natural transformations between them.

Example. *Let X be a finite simplicial complex. Then $\mathbf{Fc(X)}$ is the poset of simplices of X. It is easy to see that if $\sigma, \tau \in X$ are such that $\sigma \subseteq \tau$ (i.e. σ is a face of τ), then we have $\mathrm{st}(\tau) \subseteq \mathrm{st}(\sigma)$, which induces a chain map $C^\tau_\bullet \to C^\sigma_\bullet$ between the complexes in Lemma 2.1. Therefore we have an induced map on local homology $H^\tau_\bullet \to H^\sigma_\bullet$, viewed as chain complexes of abelian groups. Also, this assignment extends to a functor and so gives an example of a cellular sheaf as defined in the following definition.*

[3] The face relation was introduced at the very beginning of this chapter.

Definition 2.15. Let X be a finite simplicial complex. The local homology cosheaf over X is a functor $H_\bullet^- : \mathbf{Fc(X)}^{op} \to \mathbf{Ch(Ab)}$ which associates to each simplex the local homology of the simplex viewed as a chain complex of abelian groups, and to each morphism $\sigma \leq \tau$ in $\mathbf{Fc(X)}$ the induced map on local homology $H_\bullet^\tau \to H_\bullet^\sigma$.

Let us now extend the notion of a cellular sheaf to a sheaf over a topological space.

Definition 2.16. Let X be a topological space. With the notion of topological space comes the set of open subsets of X, which can be partially ordered by the subset relation $\subseteq$. We will write $\mathbf{O(X)}$ for the poset of open subsets of X partially ordered by the subset relation, viewed as a category.

Definition 2.17. A presheaf over a topological space X in a category $\mathbf{C}$ is a functor $P : \mathbf{O(X)}^{op} \to \mathbf{C}$

For our purposes the notion of a sheaf taking values in the category of abelian groups is enough. Hence we define a sheaf taking values in $\mathbf{Ab}$ to make the introduction shorter but we note that it is possible to define sheaves in a general category. However, the definition is more elaborate and requires some other notions like that of a nerve corresponding to an open cover. For details we refer to [9, Chapter 2].

Definition 2.18. A sheaf over a topological space X in $\mathbf{Ab}$ is a presheaf $F : \mathbf{O(X)}^{op} \to \mathbf{Ab}$ satisfying the following axiom: for every open set U and open cover $\{U_i\}_{i\in I}$ of U, if a selection of elements $x_i \in F(U_i)$ has the property that for all $i, j \in I$ we have $F((U_j \cap U_i) \subseteq U_i)(x_i) = F((U_j \cap U_i) \subseteq U_j)(x_j)$, then there is a unique $x \in F(U)$ such that for all $i \in I$ we have $F(U_i \subseteq U)(x) = x_i$.

Example. *Fix an abelian group G and endow it with discrete topology. Fix a topological space X. For an open set $U \subseteq X$ let $F(U) := \{\, f : U \to G \mid f \text{ is continuous} \,\}$. It is possible to check that this assignment extends to a sheaf F over X. We call such a sheaf a constant sheaf with stalks G.*

Definition 2.19. Given a sheaf F over X and an open subset $U \subseteq X$, define a sheaf $F\big|_U$ on U as a topological space by setting $F\big|_U(V) = F(V)$ for every open $V \subseteq U$. Clearly this defines a sheaf on U, which we call a restriction of sheaf F on U.

Finally, we can define a notion that will be directly useful for us in the future:

Definition 2.20. A sheaf F on X is called locally constant iff for any $x \in X$ there is an open set $x \in U$ such that $F\big|_U$ is a constant sheaf.

Remark. Of course, each constant sheaf is locally constant but the converse does not hold.

For a reader that sees sheaves for the first time this might seem a little technical. But in fact we can think about constant and locally constant sheaves in a similar way to how we think about constant and locally constant functions in topology – this analogy is very helpful when building intuition.

2.4 Stratified spaces

Intuitively, a stratification of a topological space is a decomposition of the space into closed subspaces such that each of the closed subspaces are manifolds. Here we discuss the discrete counterpart of a stratification of a space: a stratification of a (finite, regular) CW complex. For a more detailed introduction to the theory of stratified spaces we refer to [9, Chapter 11].

Definition 2.21. A cellular stratification of an n-dimensional finite regular CW complex X is a sequence of closed subspaces of X

$$X_0 \subseteq X_1 \subseteq \ldots \subseteq X_n = X$$

such that each connected component of $X_d - X_{d-1}$ is a connected union of some cells. The connected components of $X_d - X_{d-1}$ are called d-dimensional strata and have to satisfy the following axioms:

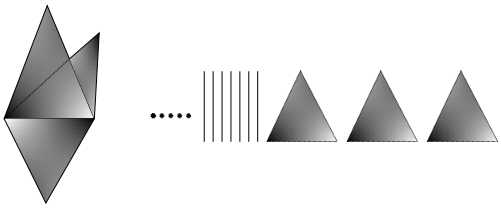
- **Frontier**: if there are strata σ, τ such that $\sigma \cap \overline{\tau} \neq \emptyset$ we then have $\sigma \subseteq \overline{\tau}$ as well as $\dim \sigma \leq \dim \tau$ and $\dim \sigma = \dim \tau \iff \sigma = \tau$. In such a situation we say that $\sigma \leq_f \tau$.

- **Link**: each d-stratum σ can be associated to a $(n-d-1)$-dimensional stratified CW complex $L(\sigma)_0 \subseteq \ldots \subseteq L(\sigma)_{n-d-1} = L(\sigma)$ (called its link) such that for all $x \in \sigma$ there is an open neighbourhood $x \in U \subseteq X$ such that for $i < d$ we have $U \cap X_i = \emptyset$ and for $d \leq i \leq n$ we have $OC(L(\sigma)_{i-d-1}) \times \mathbb{R}^d$ homeomorphic to $U \cap X_i$

Definition 2.22. If the link axiom in the definition above is dropped, then we say that a CW complex X is decomposed into pieces and the connected components of each $X_d - X_{d-1}$ are called pieces.

Example. *Note that given a finite regular CW complex X, the sequence $X_0 \subseteq X_1 \subseteq \ldots \subseteq X_n = X$ where X_i is the i-th skeleton of X defines a cellular stratification of X. Then strata correspond to cells and the link is given by the link of a cell. We call such stratification the skeletal stratification of X.*

Figure 2.7: A simplicial complex (left) and its skeletal stratification (right).



Remark. We note three things:

- $\leq_f$ defines a partial order on the strata of a given stratification.
- The link axiom ensures that each stratum is a manifold.
- In practice, when using just homological tools, we can never check if two topological spaces are homeomorphic but we can check if they have the same homology. As we are studying homological tools and their applications in this work, we should adapt the notion of a stratified space, which we do now.

Definition 2.23. We say a locally compact topological space X is an n-dimensional G-homology manifold for some abelian group G iff for all $x \in X$ we have

$$H_k(X, X \setminus \{x\}; G) = \begin{cases} G & k = n \\ 0 & \text{otherwise} \end{cases}$$

Example. *Every manifold is a G-homology manifold for any abelian group G.*

An example of a space that is a G-homology manifold for some G but not a manifold is rather complicated and not very enlightening for our purposes. We need this definition just to remind ourselves that when we use homological methods, we study spaces **not** up to homeomorphism or homotopy equivalence but up to isomorphism of homology groups.

Now we can define stratification of a CW complex with a more relaxed link axiom and in practice when we use computational homological methods, we can hope to recover only this relaxed notion of stratification.

Definition 2.24. [23, Definition 2.1] Fix an abelian group G. A cellular G-homological stratification of an n-dimensional finite regular CW complex X is a sequence of closed subspaces of X

$$X_0 \subseteq X_1 \subseteq \ldots \subseteq X_n = X$$

such that each connected component of $X_d - X_{d-1}$ is a gluing of some cells along their boundaries. The connected components of $X_d - X_{d-1}$ are called d-dimensional strata and have to satisfy the following axioms:

- **Frontier**: if there are strata σ, τ such that $\sigma \cap \overline{\tau} \neq \emptyset$ we then have $\sigma \subseteq \overline{\tau}$ as well as $\dim \sigma \leq \dim \tau$ and $\dim \sigma = \dim \tau \iff \sigma = \tau$. In such a situation we say that $\sigma \leq_f \tau$.
- **Link**: each d-stratum σ can be associated to a $(n-d-1)$-dimensional G-homologically stratified space $L(\sigma)_0 \subseteq \ldots \subseteq L(\sigma)_{n-d-1} = L(\sigma)$ (called its link) such that for all $x \in \sigma$ there is an open neighbourhood $x \in U \subseteq X$ such that for $i < d$ we have $U \cap X_i = \emptyset$ and for $d \leq i \leq n$ we have $W_i := OC(L(\sigma)_{i-d-1}) \times \mathbb{R}^d$ and $V_i := U \cap X_i$ such that $H_\bullet(\overline{W_i}, \partial W_i; G) \cong H_\bullet(\overline{V_i}, \partial V_i; G)$

Remark. Note that the link axiom ensures that each stratum is a G-homological manifold.

In the following discussion we do not make the distinction between G-homological stratification and stratification, and the ideas discussed can be applied to both notions.

Definition 2.25. Let $X_\bullet$ and $X'_\bullet$ be cellular stratifications of an n-dimensional finite regular CW complex X. We say that $X_\bullet$ is a coarsening of $X'_\bullet$ if each stratum in $X_\bullet$ is a connected union of strata in $X'_\bullet$.

Remark. Note that by our definition, each cellular stratification is a coarsening of the skeletal stratification.

When X is a finite regular simplicial complex, there is a finite number of stratifications on X. Let $\mathrm{Strat}(X)$ be the set of all stratifications on X.

Definition 2.26. Define a relation on $\mathrm{Strat}(X)$: $X'_\bullet \leq X_\bullet \iff X_\bullet$ is a coarsening of $X'_\bullet$.

Lemma 2.5. *$(\mathrm{Strat}(X), \leq)$ is a poset.*

Proof. Let us check all the axioms of a poset:

- Reflexivity: given $X_\bullet \in \mathrm{Strat}(X)$ we can regard each stratum as a connected union of itself.
- Antisymmetry: assume $X'_\bullet \leq X_\bullet$ and $X_\bullet \leq X'_\bullet$. Let σ be a stratum of $X_\bullet$, which is a connected union of strata in $\Sigma' = \{\sigma'_1, \ldots, \sigma'_k\}$. Here each σ'_i is a stratum in $X'_\bullet$ and in turn it has to be a connected union of elements of Σ_i, which is a set of strata in $X_\bullet$. But then by definition of a cellular stratification, we have to have $\Sigma_i = \{\sigma\}$ or else we would have $X_\bullet \neq X_\bullet$. Therefore, $\sigma = \sigma'$ and so $X'_\bullet = X_\bullet$.
- Transitivity: assume $X''_\bullet \leq X'_\bullet \leq X_\bullet$. Let σ be a stratum of $X_\bullet$, which is a connected union of strata in $\Sigma' = \{\sigma'_1, \ldots, \sigma'_k\}$. Each σ'_i is a stratum in $X'_\bullet$ and in turn it has to be a connected union of elements of Σ''_i, which is a set of strata in $X''_\bullet$. Therefore, σ is a connected union of strata in $\bigcup_i \Sigma''_i$ and so $X''_\bullet \leq X_\bullet$.

□

Definition 2.27. Since for a finite regular CW complex X, $\mathrm{Strat}(X)$ is a finite poset, it must have a maximal element. We call such an element a canonical stratification of X.

Lemma 2.6. *A canonical stratification of a finite regular CW complex is unique.*

Proof. [14, Section 4]. □

In fact, we can generalise the setting of stratification to decomposition of a space with respect to some sheaf. From a computational perspective this approach is taken in [2]. Assume that X is a finite simplicial complex and F is a sheaf on X taking values in **Ab**.

Definition 2.28. A sheaf F is called constructible with respect to a decomposition of X into disjoint pieces (in the sense of Definition 2.22) iff F restricted to each piece is locally constant.

Definition 2.29. A decomposition of X into disjoint pieces is an F-decomposition iff F is constructible with respect to this decomposition.

In an analogous way, one can then define a partial order on F-decompositions of X using the notion of a coarsening of a decomposition, and then prove the existence and uniqueness of the coarsest F-decomposition of X. Because our space is limited and the material is similar to the one on stratified spaces, we omit the full exposition and instead refer to [2].

Chapter 3

Word embeddings

Here we present word embeddings by concentrating on two models: Word2Vec (skip-gram version) and GloVe. It is by no means an extensive introduction to word embeddings, but serves as a brief exposition of the topic. This is so that we able to understand the motivation behind the choice of datasets in Chapter 4 and provide intuition for why topology seems to be a useful framework for studying word embeddings. For an intuitive discussion of word embeddings we refer to [24]. For an academic introduction into word embeddings (and much more), we refer to [5].

3.1 Introduction

Word embeddings seems to be an umbrella term, which refers to various (mostly neural-network based) models that take a corpus of text data as an input, and output a function from the set of words in the corpus to $\mathbb{R}^d$ for some $d \in \mathbb{N}$.[1]

For the purpose of this work a more formal framework for the study of word embeddings can be given by the following definitions:

Definition 3.1. Given a finite set Σ (called the alphabet), the set of all possible words in Σ is $\Sigma^* = \{ x_1 x_2 \dots x_n \mid x_i \in \Sigma \}$ – the set of all finite sequences taking values in Σ.

For the purpose of our discussion we can assume that Σ contains all lowercase and uppercase letters from the English alphabet.

Definition 3.2. A corpus C is a finite sequence of words $\{w_1, w_2, \dots, w_T\}$

[1] For the pre-trained models that we will be using, d is set to 300, which is quite typical.

Definition 3.3. Vocabulary V of a corpus C is the set of all unique words in C. More precisely, $V = \{ w \in \Sigma^* \mid w = w_i,\ w_i \in C \}$.

Definition 3.4. A word embedding with respect to vocabulary V is a function $E : V \to \mathbb{R}^d$.

Remark. According to our definition, distributional semantics models (DSM), which are more traditional ways to come up with a mapping from a set of words to $\mathbb{R}^d$ based on counting the co-occurrence of different words in a corpus, are also word embeddings. The reason is that in the context of this work, since we treat those models as a black box, there is no difference between DSMs and the more conventional notion of word embeddings. In fact, even if we do not treat them as black boxes, some of the word embedding models and DSMs are mathematically equivalent as shown in [17], questioning this distinction.

3.2 Word2Vec model

Word2Vec is a collection of similar models that given a corpus produce word embeddings. The models were developed in [22, 21]. Here we describe a particular variation from [21], which we later use in Chapter 4.

The idea behind this variation (called the skip-gram model) of Word2Vec is to come up with a word embedding that given a word w_i in a corpus C would predict well for some $n \in \mathbb{N}$ the words $w_{i-n}, w_{i-n+1}, \ldots, w_{i-1}, w_{i+1}, \ldots, w_{i+n}$ (i.e. the n words before w_i and the n words after w_i). More formally, we can think of this using the following definitions:

From now on, assume we have a corpus $C = \{w_1, w_2, \ldots, w_T\}$, its vocabulary V and a desired dimension of a word embedding we are trying to construct d.

Definition 3.5. Let functions $i : V \to \mathbb{R}^d$ and $o : V \to \mathbb{R}^d$ be called input and output functions respectively.

Definition 3.6. Given input and output functions i and o define the following quantity for any $v, w \in V$:

$$p(v|w) = \frac{\exp(o(v)^\top i(w))}{\sum_{w' \in V} \exp(o(w')^\top i(w))}$$

which will later be interpreted as the probability of observing a word v given that one has observed a word w.

Definition 3.7. Given input and output functions i and o as well as $n \in \mathbb{N}$ define the cost quantity:

$$\text{cost}_n = -\frac{1}{T}\sum_{t=1}^{T}\sum_{-n\leq j\leq n, j\neq 0} \log(p(w_{t+j}|w_t))$$

Definition 3.8. A skip-gram model, given an input function $i : V \to \mathbb{R}^d$ (usually initialised to some random function) and some $n \in \mathbb{N}$ constructs (using a particular neural network architecture with one input, one hidden and one output layer with no non-linear activation functions) an output function $o : V \to \mathbb{R}^d$ so that the cost quantity cost_n is minimised. This output function o is the function that one then uses as a word embedding.

In practice, when one implements an algorithm to produce a word embedding using the skip-gram model, one does not expect to find an exact solution to the optimisation problem but rather tries to numerically approximate a solution. However, practical implementation of the skip-gram model is beyond the scope of this work. A few methods of approximating different components of the model and speeding up the training process can be found in [21].

Let us think about the input of the model. Of course, we need to have a corpus, which is the dataset on which the produced word embedding is based. The model takes function i (and hence the dimension of a real vector space which we are embedding the vocabulary in) and a natural number n as an input. In practice i is set to a random function, or some simple pre-defined function. Input n is slightly more interesting – it defines the size of a "window" that we treat as a context of a particular word. That is, for each word we look at n words before it and after it, and sum up the quantities interpreted as the probability of the word given another word in the context. For details, see Definition 3.7. A very small n can render the produced embedding not as accurate, but a large n will make cost_n have more terms and hence be harder to optimise. In [21] n is set to 5 because it was found as the most optimal when thinking about the trade-off between training time and accuracy.

Intuitively speaking, the model is trying to come up with a word embedding that maximises the probability of each observed word given its context, which is defined as a window of size $2n$. What is interesting about this model is that although it seems that the neural network is learning something of little interest – to predict

the context of a given word – it ends up constructing a representation of the vocabulary in a d-dimensional real vector space with remarkable properties, which we will discuss at the end of this chapter.

3.3 GloVe model

GloVe is a model developed in [27]. It is known to be a model which is derived (almost) uniquely from the desired properties of a word embedding. Since we are using word embeddings merely as a tool, we do not discuss those properties here and instead refer to [27] for this discussion.

Speaking intuitively, GloVe leverages word co-occurrence information and is influenced by an observation that it is not the probability of observing a word w in the context[2] of a word v that carries the most relevant information but rather ratios of such probabilities, as seen in Figure 3.1.

Figure 3.1: Taken from [27]

Table 1: Co-occurrence probabilities for target words *ice* and *steam* with selected context words from a 6 billion token corpus. Only in the ratio does noise from non-discriminative words like *water* and *fashion* cancel out, so that large values (much greater than 1) correlate well with properties specific to ice, and small values (much less than 1) correlate well with properties specific of steam.

Probability and Ratio	k = *solid*	k = *gas*	k = *water*	k = *fashion*
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

Since we are looking into ratios of probabilities, if we want the ratio to be non-trivial, we need at least two different probabilities. Since each probability is defined by an ordered pair of words, we need two such pairs. If we want the ratio to be intuitively interpretable, a natural thing to do is to pick some context word x and look at $\frac{\mathbb{P}(x|w)}{\mathbb{P}(x|v)}$ for some words v, w. Note that this idea seems to implicitly distinguish between regular words and context words. There is no formal difference between them because both context and regular words come from the vocabulary of a corpus. However, this distinction is based on a role of a word in the situation. We will see

[2]Context of a word is yet to be defined but at present one can imagine the context of a particular occurrence of a word w_i in the corpus $\{w_j\}_{j\in\{1,\dots,T\}}$ being the set $\{w_{i-n}, w_{i-n-1}, \dots, w_{i-1}, w_{i+1}, \dots, w_{i+n-1}, w_{i+n}\}$ where $n \in \mathbb{N}$ is some parameter.

that instead of constructing one function from the vocabulary to $\mathbb{R}^d$, like in the skip-gram model case, this model will construct two such functions: one for regular words, and one for context words.

To formulate the model, assume we have a corpus $C = \{w_1, w_2, \ldots, w_T\}$, its vocabulary V and a desired dimension of a word embedding that we are trying to construct d.

Like in the skip-gram model case, we will have the parameter $n \in \mathbb{N}$ which will determine how big the context of a word is (i.e. how many words to the left and to the right of a particular word we should consider its context). In [27], when implementing the model $n = 10$ is chosen.

Definition 3.9. For any $v, w \in V$ let $c(v, w)$ be the number of times the word w appears in the context of v in the corpus. Note that by encoding each word by a natural number, we can put the quantities $c(v, w)$ into a matrix, which is usually called the word co-occurrence matrix.

Remark. It is possible to introduce some weights to the context quantity $c(-,-)$ described above to reflect the intuition that if we look into the context of a word w, words v which are further away from w are less significant. For simplicity we do not consider such weighted versions of $c(-,-)$.

Definition 3.10. Let $v_{\text{context}}, v_{\text{word}} : V \to \mathbb{R}^d$ be the context-vector and word-vector functions respectively.

Definition 3.11. Let $b_{\text{context}}, b_{\text{word}} : V \to \mathbb{R}$ be the context-bias and word-bias functions respectively.

Definition 3.12 ([27])**.** A function $f : \mathbb{R} \to \mathbb{R}$ is called a GloVe weighting function iff it satisfies the following properties:

- $f(0) = 0$ and f, viewed as a continuous function, satisfies $\lim_{x\to 0} f(x)\log^2(x) = 0$.
- f is non-decreasing.
- $f(x)$ is "relatively small for large values of x".[3]

[3]This property is obviously not mathematically well-defined but it is an intuitive requirement proposed in [27]

Remark. In [27] the following function is used as a GloVe weighting function:

$$f(x) = \begin{cases} (\frac{x}{100})^{\frac{3}{4}} & x < 100 \\ 1 & \text{otherwise} \end{cases}$$

Definition 3.13. Given $n \in \mathbb{N}$ and hence the notion of the context, as well as a GloVe weighting function f and functions $v_{\text{context}}, v_{\text{word}} : V \to \mathbb{R}^d$, and $b_{\text{context}}, b_{\text{word}} : V \to \mathbb{R}$ the cost quantity of the GloVe model is defined to be:

$$\begin{aligned} \text{cost}_{\text{GloVe}} = \sum_{w,w' \in V} & f(c(w,w'))[v_{\text{word}}(w)^\top v_{\text{context}}(w') \\ & + b_{\text{word}}(w) + b_{\text{context}}(w') - \log(c(w,w'))]^2 \end{aligned} \tag{3.1}$$

Definition 3.14. For fixed n, $d \in \mathbb{N}$ and a GloVe weighting function f, the GloVe model constructs $v_{\text{context}}, v_{\text{word}} : V \to \mathbb{R}^d$, and $b_{\text{context}}, b_{\text{word}} : V \to \mathbb{R}$ such that $\text{cost}_{\text{GloVe}}$ is minimised. One then uses $v_{\text{context}} + v_{\text{word}}$ as a word embedding.

Remark. In fact one could use v_{context} or v_{word} alone but it was noticed empirically in [27] that they do not differ much and combining them additively yields slightly better results probably because the model is less prone to overfitting.

Like in the Word2Vec case, in practice one does not expect to solve the optimisation problem associated with the model exactly but rather to approximate the solution. However, practical implementation is outside the scope of this work and for more details we refer to [27, Section 4].

When looking at the cost quantity of the GloVe model, it is far from obvious why such a definition is chosen and why we should expect the model to construct a word embedding with desired properties, whatever those properties might be. To account for the intuition and previous discussion of ratios of probabilities we say the following. A simplification of this model (disregarding the weighting function and the bias terms, which are important in practice but do not describe what the model does) is trying to construct functions v_{context} and v_{word} such that $v_{\text{word}}(w)^\top v_{\text{context}}(w') = \log(c(w,w'))$. Such an equation is derived from a desired property (along with a few other assumptions), postulated in the original paper, that

$$F((v_{\text{word}}(w_i) - v_{\text{word}}(w_j))^\top v_{\text{context}}(w_k)) = \frac{c(w_i,w_k)}{c(w_j,w_k)} \qquad [27, \text{ eq. } (3)]$$

Here F is some function that is specified as exp in the end and hence we see the log term in the final definition of the cost quantity. We note that this equation exactly captures the idea of looking at ratios of probabilities as discussed at the beginning of this section. For full derivation of the cost quantity we refer to [27].

3.4 Topology and word embeddings

Here we discuss the interesting properties that the word embeddings produced by GloVe and skip-gram models have and why topology might be a good framework for studying such embeddings, especially in the context of word sense disambiguation (WSD). We will see that word embeddings, even though trained on lexical data with no additional information of semantics and syntax, can learn semantical and syntactical information quite well.

Firstly, let us set the scene for thinking about the produced word embeddings. Let $f : V \to \mathbb{R}^d$ be a word embedding of a vocabulary V.[4] Naturally, we want to look at $f(V)$, which is a point cloud in $\mathbb{R}^d$. To apply topology, we need a structure of a topological space; in this setting, equipping $f(V)$ with a metric seems a natural way to obtain such structure. From a mathematical point of view, since $f(V)$ is a subset of $\mathbb{R}^d$, we might want to study it using the Euclidean distance. However, empirically we know that various semantical relationships are best captured by looking at the angle between different (linear combinations of) elements of $f(V)$ [20, 16, 27].

Motivated by this, we will map $f(V)$ into the $(d-1)$-dimensional unit sphere with the geodesic distance. We will obtain a new set $W = \left\{ \frac{v}{\|v\|} \in S^{d-1} \;\middle|\; v \neq 0,\, v \in f(V) \right\}$. Note that it is possible to have two vectors $v, w \in f(V)$ such that $\frac{v}{\|v\|} = \frac{w}{\|w\|}$ but we do not observe such situations in practice. Neither do we observe the zero vector in $f(V)$. We will write v_{word} for the vector in W corresponding to the word 'word'. After such procedure we end up with a point cloud in the metric space given by the following definition:

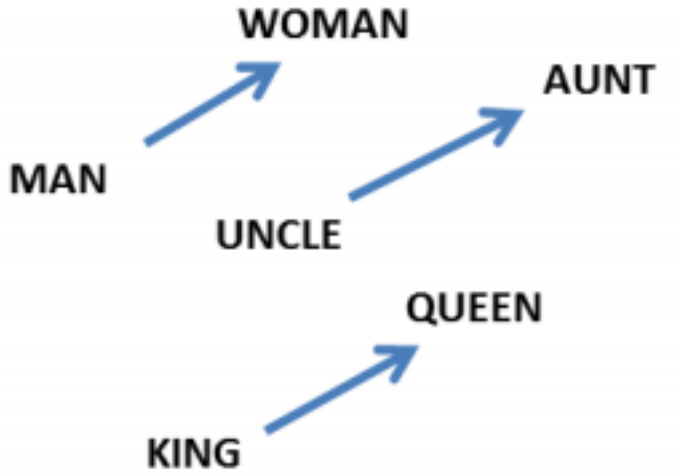
Definition 3.15. Let $S^{d-1} := \left\{ x \in \mathbb{R}^d \;\middle|\; \|x\| = 1 \right\}$ be the $(d-1)$-dimensional unit sphere embedded into the Euclidean space $\mathbb{R}^d$, which is equipped with the usual dot product. Define a distance function $d_{\text{geo}} : S^{d-1} \times S^{d-1} \to \mathbb{R}_{\geq 0}$ by setting for all $x, y \in S^{d-1}$ $d_{\text{geo}}(x, y) = \cos^{-1}(x \cdot y)$, where $x \cdot y$ is the usual dot product in $\mathbb{R}^d$. It is a well-known fact from basic topology that $(S^{d-1}, d_{\text{geo}})$ defines a metric space and we call it the unit sphere with geodesic distance.

The interesting properties of both skip-gram and GloVe models were observed on many occasions and a short summary of them can be found in [27, Subsections 4.1, 4.3]. The most relevant properties for our work is good performance on word

[4] The reader can assume f is produced by one of the models we have discussed. However, the following discussion holds for a larger class of models.

analogy and word similarity tasks. A word analogy task is of the form "a is to b like c is to what?". A typical example is "queen is to king like man is to what?" and the correct answer is 'woman'. A task "a is to b like c is to what?" can be answered by looking at the vector $v_a - v_b + v_c$ and then finding a vector $v_d \in W$ which is the closest to $v_a - v_b + v_c$ with respect to the geodesic distance. The word 'd' would be returned as the answer to the word analogy task. Such analogies can be semantic like the king-queen example or syntactic, for example, "fast is to fastest like quick is to what?". Obviously, the answer is 'quickest'. Depending on corpora that models are trained on and the set of analogies that is being used for testing, word embeddings can achieve accuracy of up to 81.9% on semantic and 69.3% on syntactic analogies [27, Table 2].

Figure 3.2: Taken from [20].



Perhaps a more important property of word embeddings from a topological point of view is that they map similar words to vectors that are close with respect to the geodesic distance. This can be tested by taking pairs of words and assigning them with scores, given by a human, on how similar the words in a single pair are, and then seeing how the scores predicted by the model (i.e. the (cosine of) geodesic distance of the pair) differ from the human-produced ones. Again, a high accuracy up to 83.6% is observed [27, Table 3].

To summarise, on the basis of the performance on word analogy and similarity tasks, we see that the datasets coming from word embeddings have a potential of exhibiting interesting topology.

Chapter 4

Local homology clustering

Here we present a clustering algorithm based on local homology and review the results on two datasets that we generate from embeddings of a certain set of words with respect to two different embeddings.

In the previous section we have seen that word embedding vectors of similar words end up close to each other when they are mapped into a sphere equipped with the geodesic distance. Hence, we hope that similar words comprise some sort of structure. Continuing this line of thought, we hope that by looking at a local neighbourhood of some word vectors we can distinguish words with vastly different meanings since they would lie at the intersection of two (or more) structures of word vectors of similar words. For example, we might hope that the vector of 'bank' would be in the structure of vectors of words related to finance and also in the structure of vectors of words related to rivers since 'bank' can mean either a certain financial institution or a river bank. Detecting such words would help to solve the word sense disambiguation problem in NLP. Local homology seems a natural tool and hence it is explored in this chapter.

4.1 Algorithm and pipeline

A broad idea of the algorithm we present here is to cluster datapoints that have a local homology preserving path in a simplicial complex associated to the point cloud. To make this more precise we define a pipeline of data that we are using and then the clustering algorithm itself.

The pipeline is as follows:

1. Start with a point cloud S in a metric space (M, d).
2. Pick ϵ and build the VR-complex associated to S and ϵ – $\mathrm{VR}_\epsilon(S)$.
3. Since each $s \in S$ is a 0-simplex in $\mathrm{VR}_\epsilon(S)$, the notion of the local homology of s is well-defined, so we compute $H_\bullet^s$ for each $s \in S$.
4. For each edge (i.e. 1-simplex) e in $\mathrm{VR}_\epsilon(S)$, we also compute $H_\bullet^e$.
5. Define a relation on S: $s \sim p$ iff $H_n^s \cong H_n^p$ for all $n \in \mathbb{N}$ and there is a path of edges $\{e_1, e_2, \ldots, e_k\}$ between s and p such that for any $n \in \mathbb{N}$ and for any $i, j \in \{1, 2, \ldots, k\}$ we have $H_n^{e_i} \cong H_n^{e_j} \cong H_n^s \cong H_n^p$. Note that $\sim$ defines an equivalence relation.
6. We output equivalence classes in S with respect to $\sim$ (together with their local homology groups) and call them clusters.

The reason we require a local homology preserving path between vertices to cluster them is because it is a necessary condition of them belonging to the same canonical stratum of $\mathrm{VR}_\epsilon(S)$ as proved in [23]. The extension of local homology clustering to stratification learning is discussed in Chapter 5.

We provide pseudocode for the less trivial parts of the pipeline. From now on assume that S is a point cloud in a metric space (M, d) and we have picked $\epsilon \in \mathbb{R}_{\geq 0}$ and d, which will serve as a cut-off point for the VR-complex.

The algorithms given in this chapter calculate local homology over $\mathbb{Z}/2\mathbb{Z}$ but this field can be replaced by any other field and the algorithms would still work. In fact, the algorithms can be easily extended to calculate local homology over $\mathbb{Z}$ but we would need to keep track of the torsion and not just the betti numbers as we do now. However, the same Smith Normal Form approach works.

The first step of the pipeline is to build the VR-complex. As remarked in Section 2.3, the VR-complex can have higher dimensionality than the ambient space that the data is coming from, so it is usual to restrict the complex and look at its d-skeleton. For computation of the VR-complex we use the incremental algorithm as described in [34, Section 4.2]. Assume we have a subroutine NeighbourhoodGraph() which accepts S together with ϵ and returns the ϵ-radius neighbourhood graph associated to S. Also, we assume that we have a total ordering of the vertices of such a graph. The ordering does not need to have any particular properties – an arbitrary total ordering will work. Then the incremental algorithm applied to our case is as follows:

Algorithm 1 Incremental Construction of the VR-complex

```
1: procedure LowerNbrs(V, E, u)                    ▷ (V, E) is a graph, u ∈ V
2:     return { v ∈ V | u > v, {u, v} ∈ E }          ▷ V is ordered
3: end procedure

4: procedure AddCofaces(V, E, k, τ, N, 𝒱)
5:     if dim(τ) ≥ k then                  ▷ τ is a simplex so dim(τ) is well-defined
6:         return
7:     else
8:         for all v ∈ N do
9:             σ := τ ∪ {v}
10:            M := N ∩ LowerNbrs(V, E, v)
11:            AddCofaces(V, E, k, σ, N, 𝒱)
12:        end for
13:    end if
14: end procedure

15: procedure IncrementalVR(S, ϵ, k)
16:    (V, E) := NeighbourhoodGraph(S, ϵ)
17:    𝒱 := ∅
18:    for all v ∈ V do
19:        N := LowerNbrs(V, E, v)
20:        AddCofaces(V, E, k, {v}, N, 𝒱)
21:    end for
22:    return 𝒱
23: end procedure
```

Theorem 4.1. [34, Section 4.2 Theorem 2] IncrementalVR(S, ϵ, d) *computes the* d*-skeleton of* $VR_\epsilon(S)$.

Now let us turn to an algorithm that computes k-th local homology with coefficients in $\mathbb{Z}/2\mathbb{Z}$. We assume that a simplicial complex that we are dealing with is totally ordered as a set – arbitrary order is sufficient. Also, in the following algorithm we use a notion of empty matrix. It is a degenerate case of a matrix – it has 0 rows and 0 columns as well as 0 entries. We also assume to have the following subroutines:

- SmithNF() – given a matrix M with entries in $\mathbb{Z}/2\mathbb{Z}$ it returns the Smith Normal Form of M.
- ZeroColumns() and NonZeroColumns() – given a matrix M, return the number of zero columns in M and the number of non-zero columns in M respectively.

Algorithm 2 Local Homology of a Simplex

1: **procedure** STAR(K, τ) ▷ K is a finite simplicial complex, $\tau \in K$
2: **return** $\{ \sigma \in K \mid \tau \subseteq \sigma \}$ ▷ Inherits ordering from K
3: **end procedure**

4: **procedure** GETBOUNDARYOPERATOR(K, k)
5: $B_{\text{domain}} := \{ \sigma \in K \mid |\sigma| = k+1 \} = \{d_1, \ldots, d_n\}$ ▷ Ordering inherited from K
6: $B_{\text{codomain}} := \{ \sigma \in K \mid |\sigma| = k \} = \{c_1, \ldots, c_m\}$ ▷ Ordering inherited from K
7: **if** $B_{\text{domain}} = \emptyset$ **then**
8: **return** empty matrix
9: **else if** $k = 0$ **then**
10: **return** $1 \times |B_{\text{domain}}|$ matrix with all zero entries
11: **else**
12: $M_{i,j} := \begin{cases} 1 & d_j \subseteq c_i \\ 0 & \text{otherwise} \end{cases}$
13: **return** $[M_{i,j}]_{m \times n}$
14: **end if**
15: **end procedure**

16: **procedure** LOCALHOMOLOGY(K, τ, k)
17: $s_\tau := \text{STAR}(K, \tau)$
18: $\partial_k := \text{GETBOUNDARYOPERATOR}(s_\tau, k)$
19: $\partial_{k+1} := \text{GETBOUNDARYOPERATOR}(s_\tau, k+1)$
20: $\dim(\ker \partial_k) := \text{ZEROCOLUMNS}(\text{SMITHNF}(\partial_k))$
21: $\dim(\text{im}\, \partial_{k+1}) := \text{NONZEROCOLUMNS}(\text{SMITHNF}(\partial_{k+1}))$
22: **return** $\dim(\ker \partial_k) - \dim(\text{im}\, \partial_{k+1})$
23: **end procedure**

Theorem 4.2. *Given a finite simplicial complex K and a simplex $\tau \in K$, the procedure* LOCALHOMOLOGY(K, τ, k) *computes* $\dim H_k^\tau$.

Proof. Since we are working over $\mathbb{Z}/2\mathbb{Z}$ and with finite simplicial complexes, the local homology groups are just finite-dimensional vector spaces over $\mathbb{Z}/2\mathbb{Z}$ and hence are characterised by their dimension.

Here we are just invoking Lemma 2.1. It is clear that $\partial_\bullet$ in lines 18-19 are the boundary operators from Lemma 2.1. It is a fact from basic algebra that the number of zero-columns in the Smith Normal Form of a matrix associated with a linear map f is the dimension of $\ker f$ and the number of non-zero columns in the dimension of $\text{im}\, f$. Hence $\dim(\ker \partial_k)$ and $\dim(\text{im}\, \partial_{k+1})$ are calculated correctly in lines 20, 21.

By Lemma 2.1, $H_k^\tau = \ker \partial_k / \operatorname{im} \partial_{k+1}$ and hence $\dim H_k^\tau = \dim(\ker \partial_k) - \dim(\operatorname{im} \partial_{k+1})$ as returned by the procedure. □

The final step of the pipeline is to cluster points in S according to the equivalence relation defined at the beginning of this subsection. Assume we have a subroutine $\mathrm{DFS}(V, E)$ – given a graph $G = (V, E)$ it returns the connected components of G using the Depth First Search algorithm.

Algorithm 3 Local Homology Clustering

1: **procedure** SKELETON(K, k) ▷ K - finite simplicial complex, $k \in \mathbb{N}$
2: **return** $\{ \tau \in K \mid |\tau| \leq k \}$
3: **end procedure**

4: **procedure** CLUSTER(S, ϵ, d)
5: $\mathcal{V} :=$ INCREMENTALVR(S, ϵ, d)
6: $\mathcal{V}_1 :=$ SKELETON$(\mathcal{V}, 1)$
7: **for all** $\tau \in \mathcal{V}_1$ **do**
8: **for all** $0 \leq k \leq d$ **do**
9: $H_k^\tau =$ LOCALHOMOLOGY$(\mathcal{V}, \tau, k)$
10: **end for**
11: **end for**
12: $V :=$ SKELETON$(\mathcal{V}, 0)$
13: $E := \left\{ \{v, w\} \in \mathcal{V}_1 \;\middle|\; H_\bullet^v \cong H_\bullet^w \cong H_\bullet^{\{v,w\}} \right\}$
14: **return** $\mathrm{DFS}(V, E)$
15: **end procedure**

Theorem 4.3. CLUSTER(S, ϵ, d) *computes the clustering of* S *defined at the beginning of the subsection.*

Proof. Since in line 14 we are removing all the edges that do not preserve the local homology between their endpoints, what we are left with are the edges that do preserve local homology. Therefore all paths in such a graph will be local homology preserving and so by definition of the equivalence relation, connected components of this graph will be exactly the required equivalence classes. □

4.2 Implementation

We implement this algorithm in Python 3.6 when coefficients in $\mathbb{Z}/2\mathbb{Z}$ are used to calculate the local homology groups. We implement everything from scratch except

the SNF() subroutine, which we take from [8] and adapt to our needs by making the implementation iterative rather than recursive. We use a popular linear algebra library called *numpy* to do the matrix calculations and the built-in Python library *multiprocessing* to implement distribution across multiple processors. The code can be found in a GitHub repository [29].

4.3 Results on data

We run the local homology clustering algorithm on datasets obtained by taking an embedding of 155 words that are either related to the topic of water and rivers or to the topic of finance. There are a few words that are not related to any of the two topics. The reason for this choice is because we hope to see interesting local structure near the word 'bank', which is a homonym, related to both rivers and finance, as discussed earlier.

We use two pre-trained word embeddings:

1. Embedding obtained by training the skip-gram model on "part of Google News dataset (about 100 billion words). The model contains 300-dimensional vectors for 3 million words and phrases" [32].

2. Embedding obtained by training the GloVe model on web crawl data obtained from Common Crawl. The result is 300-dimensional word embedding [12] and we use the spaCy library in Python to access it.

By taking the word vectors of the 155 words with respect to the first and the second embeddings, and regarding them as lying on the unit sphere with geodesic distance (Definition 3.15) we get respective point clouds $D_{\text{skip-gram}}$ and D_{GloVe}, which are used as an input to the pipeline.

Also, we note that we have performed calculations without mapping the word vectors to the unit sphere and using the Euclidean distance. However, no interesting results have been found; almost all datapoints end up having all homology groups being 0 or that of a point. As we will see, this is not the case when using the geodesic distance (Definition 3.15) and so this again suggests that the geodesic distance carries more interesting properties than the Euclidean distance, as seen in Chapter 3 as well.

Even though the two datasets exhibit different local structure and local homologies of vectors corresponding to the same words are very different, we see similar structure of clusters obtained:

- For higher values of ϵ there is a giant cluster with local homology all being zero, which we can think of the boundary of the dataset. All the other clusters mostly have only one point in them. Very rarely we see 2 or 3 points clustered together.
- For lower values of ϵ 2 or 3 point clusters are even more rare, there is no giant cluster, almost all clusters have only one point in them.

The reason for such clustering results is possibly not the similarity between the datasets but the fact that the requirement of local homology groups to be isomorphic is very limiting and in practice we observe that local homology groups are vastly different. Therefore, it is rare to see interesting clusters. This can be changed by relaxing the isomorphism requirement by looking at persistent local homology instead. This would take out the ϵ parameter but would introduce another parameter d for the distance between two barcodes in the barcode space with a distance one is using. Then we could say that persistent local homology groups are "isomorphic" (i.e. sufficiently similar to be clustered together) iff at each homological degree the distance between the barcodes is less than d. However, this is left as a part of future work.

Because the clusters do not have interesting structure, they are not discussed further. Now we will explore the local homology groups of different datapoints.

4.3.1 Results on $D_{\text{skip-gram}}$

On this dataset we have performed experiments for all $\epsilon \in \{40°, 41°, \ldots, 81°\}$.

Note that since we are calculating homology over $\mathbb{Z}/2\mathbb{Z}$, each homology group is completely characterised by its betti number. Hence, for simplicity purposes we denote (local) homology groups as a list of natural numbers. Writing $a_0\, a_1\, \ldots\, a_n$ for $a_i \in \mathbb{N}$ means that $H_i = (\mathbb{Z}/2\mathbb{Z})^{a_i}$ for $i \in \{0, 1, \ldots, n\}$ and $H_i = 0$ for $i > n$. Table 4.1 contains a selection of words, of which their word vectors have local homology that we find interesting.

Table 4.1: Words with interesting homology ($D_{\text{skip-gram}}$)

Word	Value of ϵ	Local homology
bank	80	0 0 1 5
bank	79	0 1 1 2
bank	78	0 1 1
bank	75	0 1 3
bank	74	0 1 2
bank	66	0 5
corporation	81	0 0 0 4 4
corporation	80	0 0 2 6
corporation	79	0 0 5 4
corporation	78	0 0 5
corporation	77	0 0 10
corporation	76	0 0 7
corporation	75	0 2 1
invest	81	0 0 0 7
invest	80	0 0 1 5
invest	79	0 0 5
invest	78	0 0 6
invest	74	0 2 1
invest	73	0 2 1
manufacturing	81	0 0 0 4 1
manufacturing	80	0 0 1 3
market	80	0 1 0 2
market	79	0 1 2 1
market	77	0 1 1
market	76	0 1 1
savings	80	0 0 3 1
savings	77	0 1 1
savings	75	0 1 1
transaction	81	0 2 2 1
transaction	79	0 1 4
transaction	78	0 1 4

Here we will use some notation for the sake of brevity. We will write S for the point cloud $D_{\text{skip-gram}}$, $G_\epsilon(S)$ for the graph in Definition 2.9, and $\text{VR}_\epsilon(S)$ for the VR-complex (Definition 2.11). Also, recall Definition 2.10.

4.3.1.1 Understanding 1st local homology

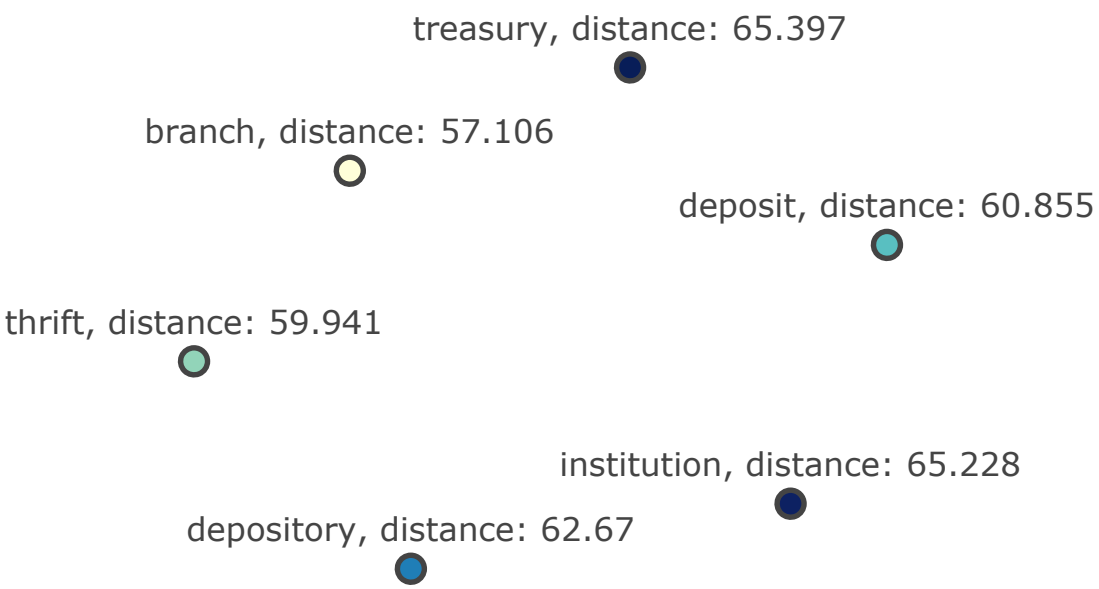
We saw earlier in Lemma 2.3 that the 1st local homology group of a vertex $\{x\} \in \mathrm{VR}_\epsilon(S)$ indicates the number of connected components in $\mathrm{lk}(\{x\})$. In practice this usually happens when a vertex x has an ϵ-neighbour that is not ϵ-neighbours with any of the ϵ-neighbours of x in $G_\epsilon(S)$. Hence we usually observe a big connected component of the link and then a few "lone" vertices, the existence of which is detected by $H_1^{\{x\}}$. However, there are rare occasions where we have more than one connected component which is not just a point.

Also, we notice that many word vectors have a range of values of ϵ for which the 1st local homology changes due to new points coming into the link that form "lone" connected components for a short range of ϵ-values before they connect to the main component.

The local structure around v_{bank} is a good example to look at in order to understand the 1st local homology.

For $\epsilon < 56°$ the vector v_{bank} has the local homology of a point since its link is empty. For $\epsilon \in \{56°, \ldots, 65°\}$ the only thing that changes is the 1st local homology, reflecting the fact that we have more word vectors coming into the link, which have no connections with each other. At $\epsilon = 66°$ the 1st local homology is $(\mathbb{Z}/2\mathbb{Z})^5$ and the 6 points in the link are vectors of the following words: 'deposit', 'branch', 'institution', 'treasury', 'depository', 'thrift', as seen in Figure 4.1.

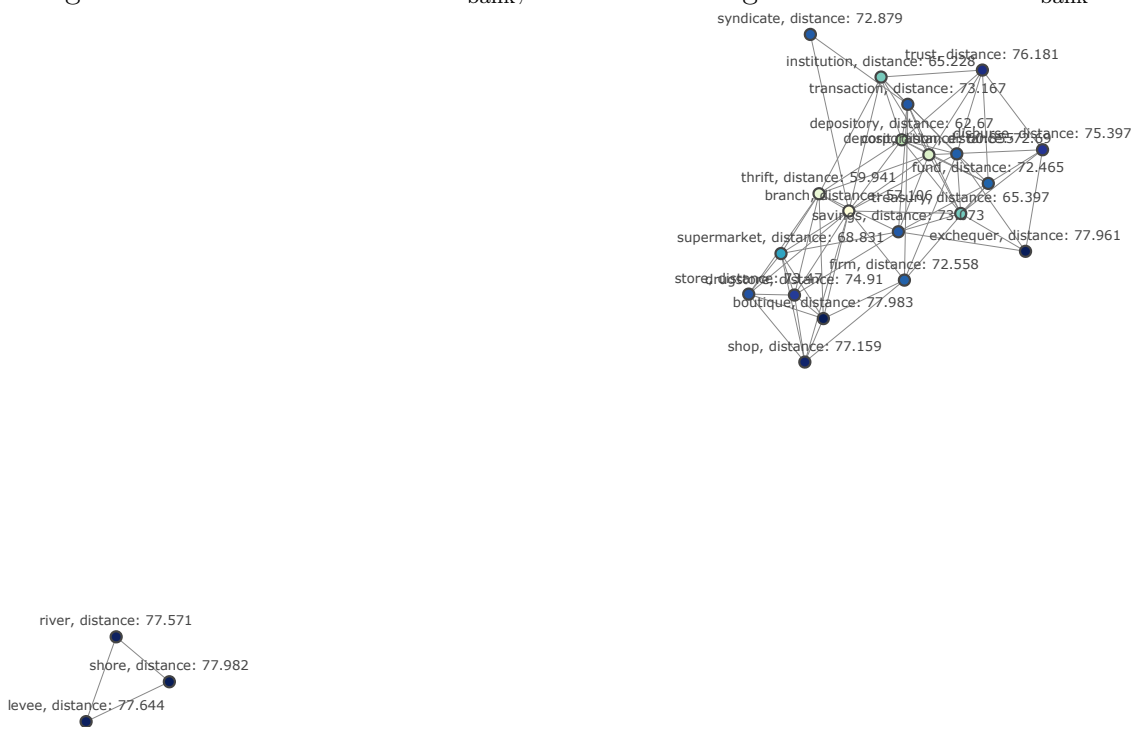
Figure 4.1: Link at $\epsilon = 66°$ of v_{bank}; distance is the geodesic distance to v_{bank}.



When $\epsilon \in \{67°, 68°\}$ the 1st local betti number of v_{bank} gets smaller reflecting the fact that the 6 words in the link start having connections between each other. At $\epsilon = 69°$ we have another word vector ('supermarket') coming into the link with no connections. At $\epsilon \in \{71°, 72°\}$ the 1st local homology being $\mathbb{Z}/2\mathbb{Z}$ reflects the fact that v_{thrift} has no connections but is in the link. At $\epsilon \in \{73°, 74°, 75°\}$ the 1st local homology being $\mathbb{Z}/2\mathbb{Z}$ reflects the fact that a new word vector ('syndicate') becomes part of the link with no connections to the biggest connected component.

When $\epsilon \in \{78°, 79°\}$ the 1st local homology being $\mathbb{Z}/2\mathbb{Z}$ reflects the fact that a new connected component – a 3-clique ('river', 'levee', 'shore') is formed in the link of v_{bank}. This is one of the rare cases when the smaller connected component of the link is not just a point. Also, the interesting thing is that the big connected component is composed of finance related words but the small ('river', 'levee', 'shore') is composed of nature related words, confirming our intuition that the fact that 'bank' has two unrelated meanings should be reflected by the local homology.

Figure 4.2: Link at $\epsilon = 78°$ of v_{bank}; distance is the geodesic distance to v_{bank}.



4.3.1.2 Understanding 2nd local homology

From Lemma 2.3, we know that the 2nd local homology of a vertex is isomorphic to the 1st homology of its link. Hence we can say that it counts cycles in the link which do not contain cliques. The reason cycles cannot contain cliques is because $\text{VR}_\epsilon(S)$ is a clique complex $G_\epsilon(S)$, and since simplices are contractible, they do not contribute to homology. Intuitively speaking, such cycles in the link appear when there are 4 (or more) words, say v_1, v_2, v_3, v_4, such that:

- v_1 and v_2 as well as v_2 and v_3 are used together in the corpus enough to form a connection at **that particular** ϵ **scale**, but not v_1 and v_3.
- v_1 and v_4 as well as v_4 and v_3 are used together in the corpus enough to form a connection at **that particular** ϵ **scale**, but not v_1 and v_3.

So v_1, v_2, v_3, v_4 form a "rectangle" is the graph. An intuitive example where we might see something like this is with words 'reservoir', 'reserve' and 'stash'. $v_{\text{reservoir}}$ and v_{reserve} are close as well as v_{reserve} and v_{stash} but not $v_{\text{reservoir}}$ and v_{stash}. Semanti-

cally, it makes sense because we stash things to reserve them and we use reservoirs to reserve water but we do not really stash water in reservoirs.

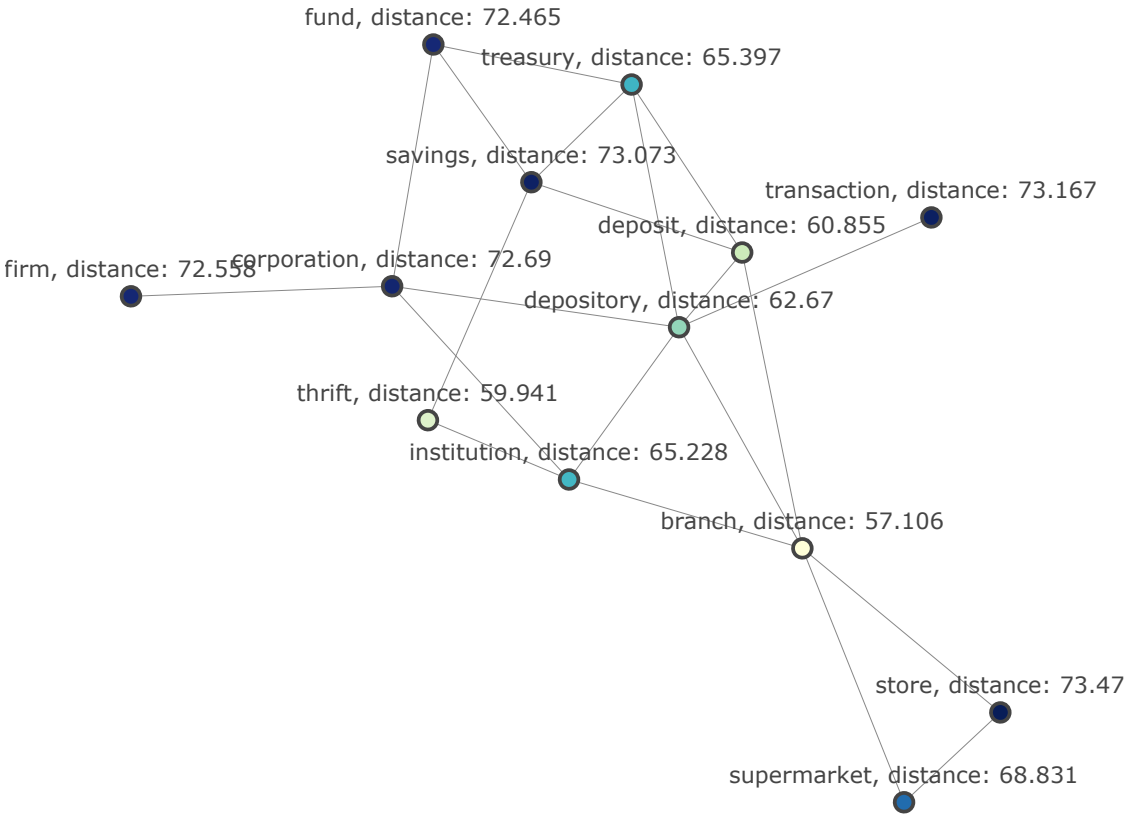
The local structures around v_{bank} and $v_{\text{corporation}}$ provide some insight into the 2nd local homology, so let us look into them now.

At $\epsilon = 74°$ v_{bank} has the 2nd local homology $(\mathbb{Z}/2\mathbb{Z})^2$, which reflects the existence of cycles composed of vectors of the following words:

- 'depository', 'treasury', 'savings', 'thrift', 'institution'.
- 'depository', 'treasury', 'fund', 'corporation', 'institution'.

This can be noted in Figure 4.3.

Figure 4.3: A part of the link of v_{bank} at $\epsilon = 74°$; distance is the geodesic distance to v_{bank}.

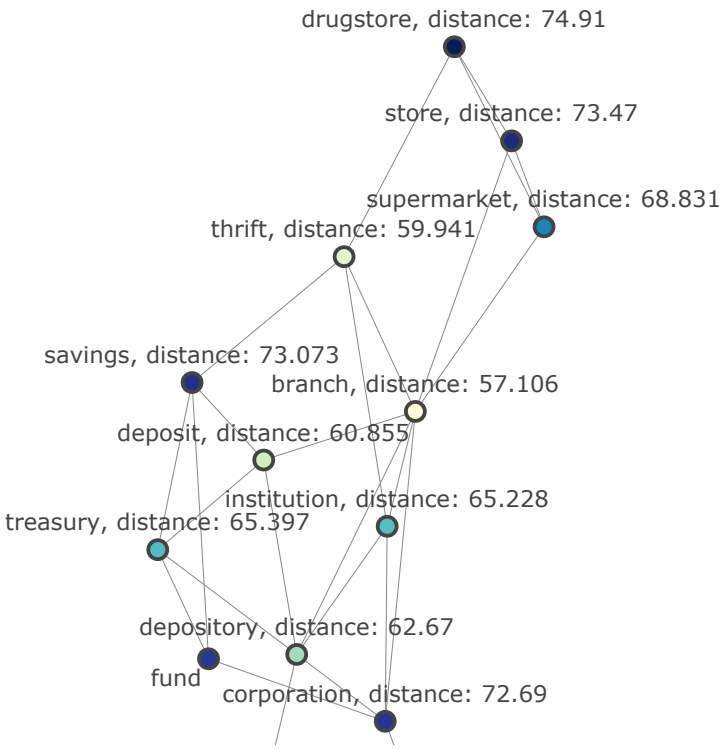


At $\epsilon = 75°$ we have the 2nd local homology of v_{bank} is $(\mathbb{Z}/2\mathbb{Z})^3$, generated by the following cycles:

- ‘depository’, ‘treasury’, ‘fund’, ‘corporation’.
- ‘thrift’, ‘branch’, ‘store’, ‘drugstore’.
- ‘depository’, ‘treasury’, ‘savings’, ‘thrift’, ‘institution’.

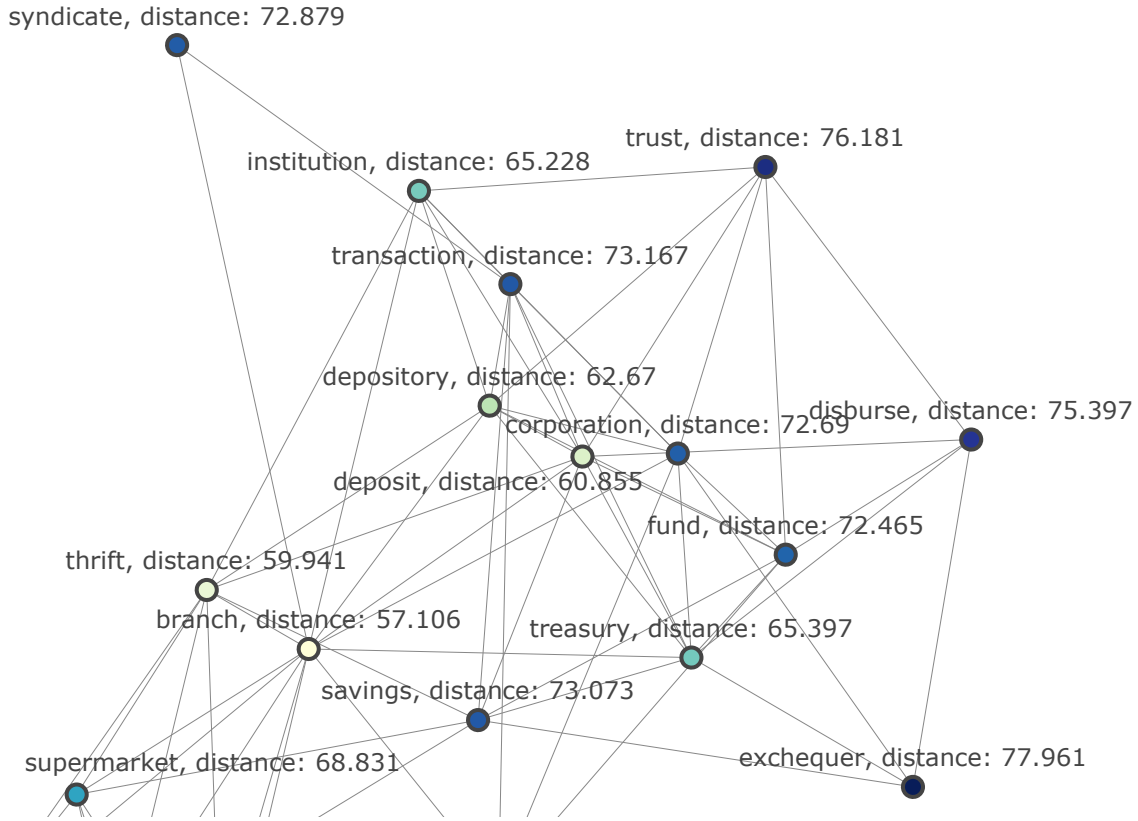
Of course, those representatives of cycles are not unique. In fact, at this scale the cycle ‘depository’, ‘treasury’, ‘savings’, ‘thrift’, ‘institution’ and ‘depository’, ‘deposit’, ‘savings’, ‘thrift’, ‘institution’ represent the same homology class because of the 3-cliques ‘depository’, ‘deposit’, ‘treasury’ and ‘deposit’, ‘treasury’, ‘savings’. All of this can be seen in Figure 4.4.

Figure 4.4: A part of the link of v_{bank} at $\epsilon = 75°$; distance is the geodesic distance to v_{bank}.



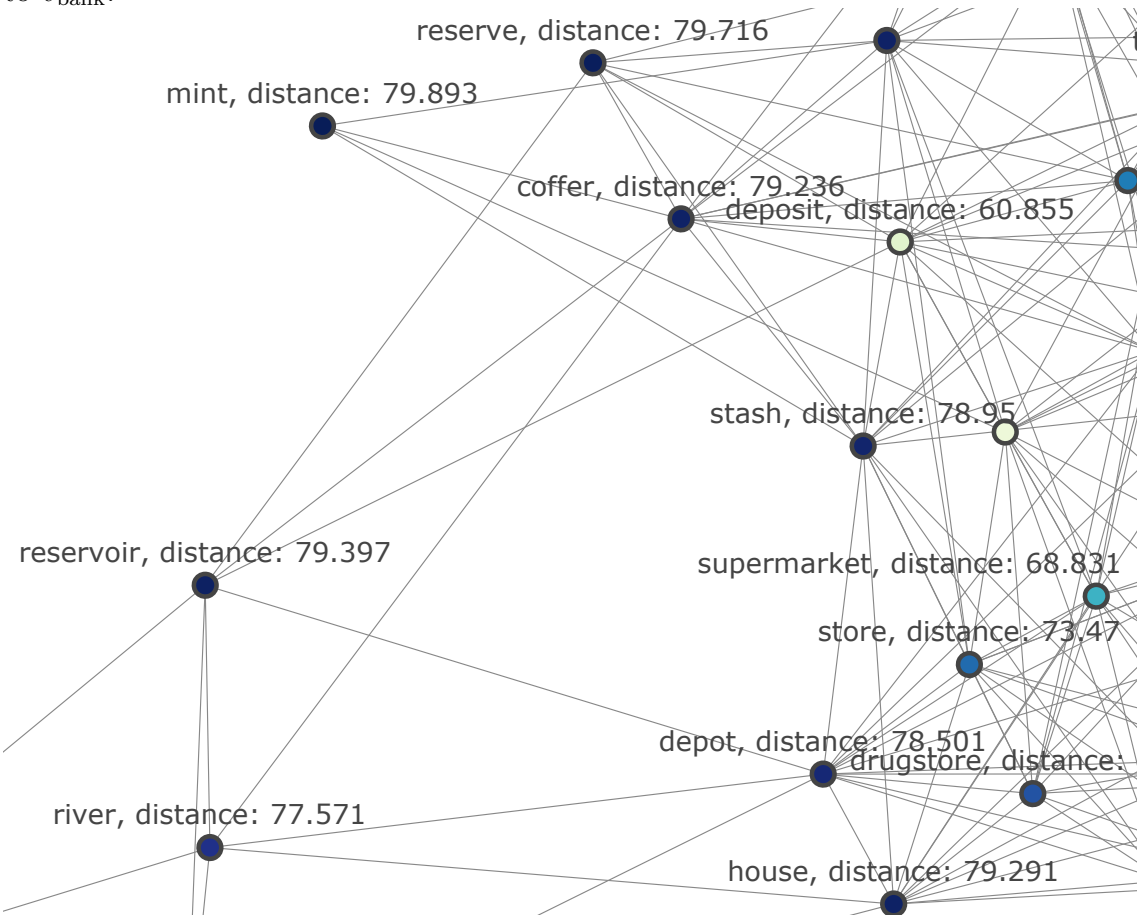
At $\epsilon \in \{78°, 79°\}$ we have the 2nd local homology of v_{bank} being $\mathbb{Z}/2\mathbb{Z}$, which reflects the existence of the cycle ‘transaction’, ‘syndicate’, ‘branch’, ‘deposit’. We saw the word vector of ‘syndicate’ earlier as a “lone” connected component in the link and now, still being relatively far away from a lot of points, at this scale it connects only to a couple points and hence creates a cycle which we capture by looking at local homology. This can be noted in Figure 4.5.

Figure 4.5: A part of the link of v_{bank} at $\epsilon = 78°$; distance is the geodesic distance to v_{bank}.



At $\epsilon = 80°$ the 2nd local homology being $\mathbb{Z}/2\mathbb{Z}$ indicates the existence of the cycle 'reservoir', 'reserve', 'stash', 'depot', as seen in Figure 4.6.

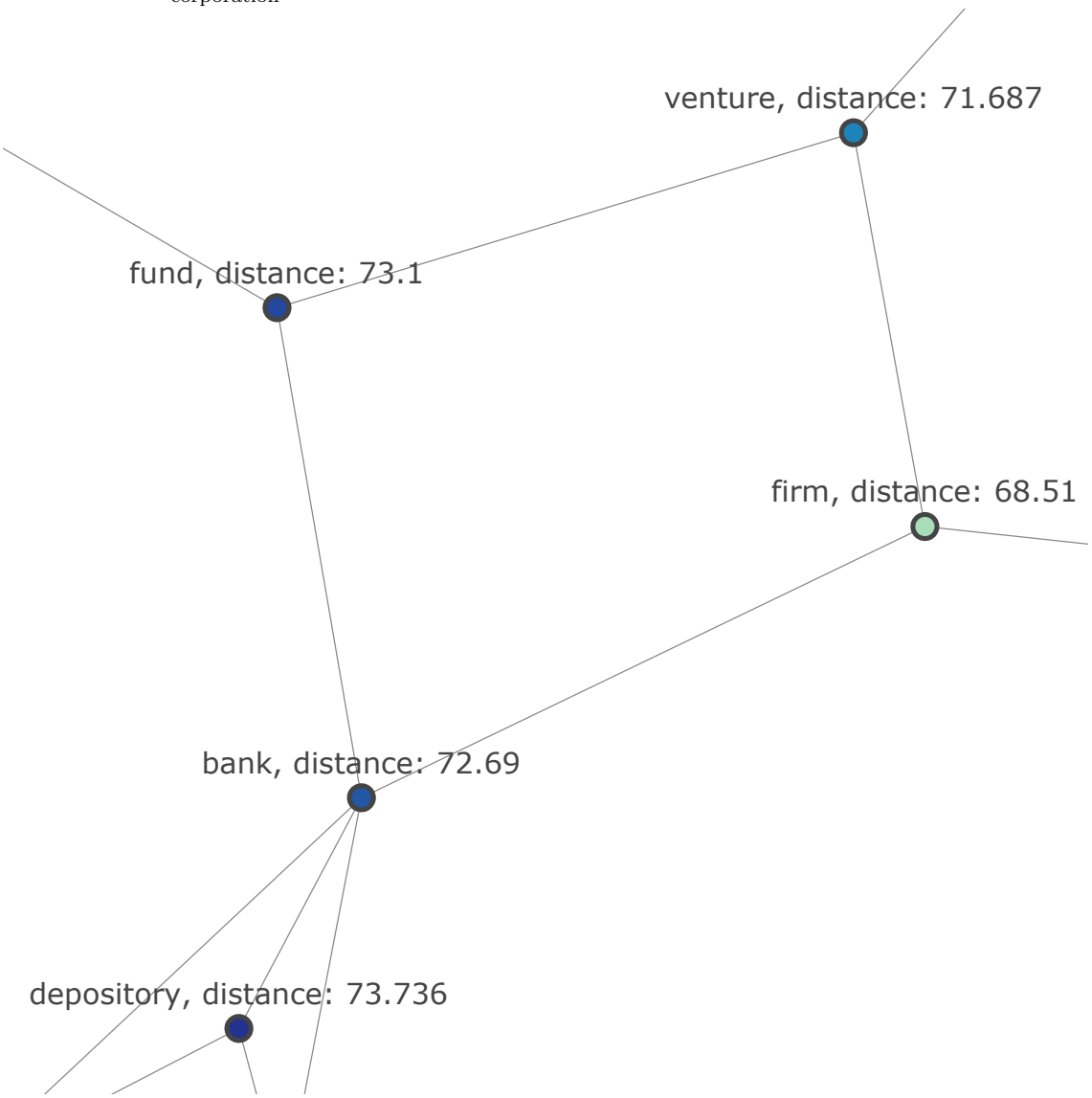
Figure 4.6: A part of the link of v_{bank} at $\epsilon = 80°$; distance is the geodesic distance to v_{bank}.



Now let us take a look at the vector $v_{\text{corporation}}$ and its local structure. We will concentrate on the homology when $\epsilon \in \{75°, 76°, 77°\}$.

When $\epsilon = 75°$, the 2nd local homology $\mathbb{Z}/2\mathbb{Z}$ generated by the cycle of vectors of 'fund', 'venture', 'firm', 'bank', as can be seen in Figure 4.7.

Figure 4.7: A part of the link of $v_{\text{corporation}}$ at $\epsilon = 75°$; distance is the geodesic distance to $v_{\text{corporation}}$.



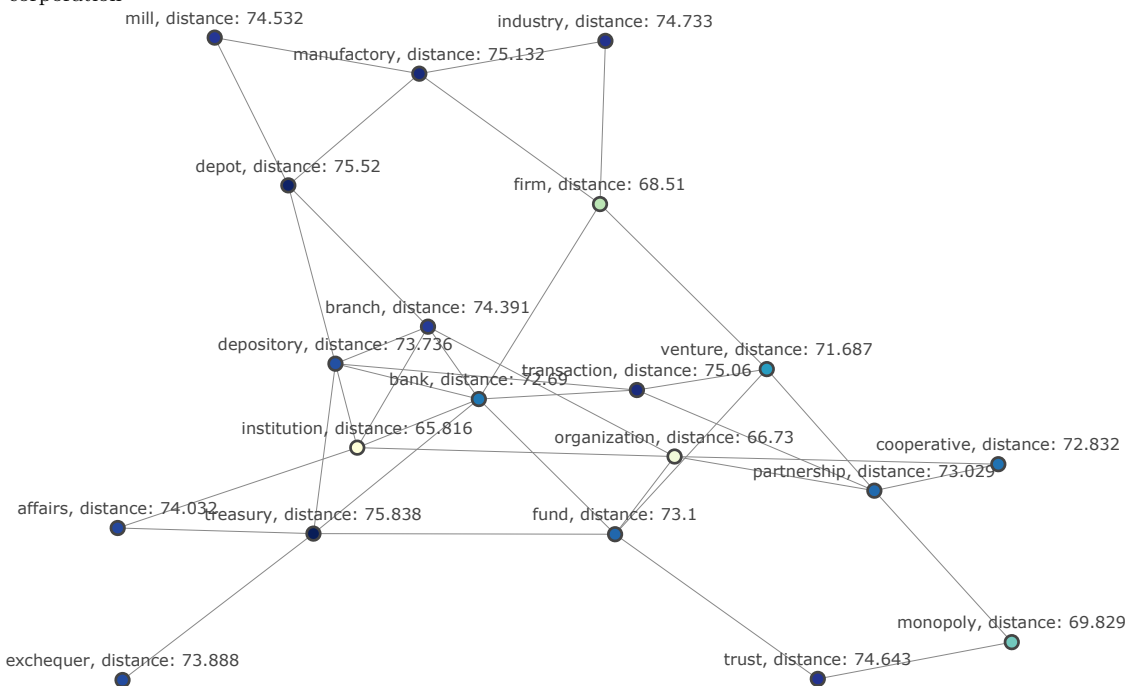
When $\epsilon = 76°$ we suddenly see the 2nd local homology being $(\mathbb{Z}/2\mathbb{Z})^7$. By looking at the link we see that the earlier mentioned cycle persists and 6 more cycles appear:

- 'depot', 'branch', 'bank', 'monopoly'.
- 'bank', 'transaction', 'partnership', 'organization', 'institution'.
- 'institution', 'depository', 'transaction', 'partnership', 'organization'.
- 'affairs', 'institution', 'bank', 'treasury'.

- ‘branch’, ‘organization’, ‘fund’, ‘bank’.
- ‘fund’, ‘organization’, ‘partnership’, ‘monopoly’, ‘trust’.

The cycles can be seen in Figure 4.8.

Figure 4.8: The link of $v_{\text{corporation}}$ at $\epsilon = 76°$; distance is the geodesic distance to $v_{\text{corporation}}$.

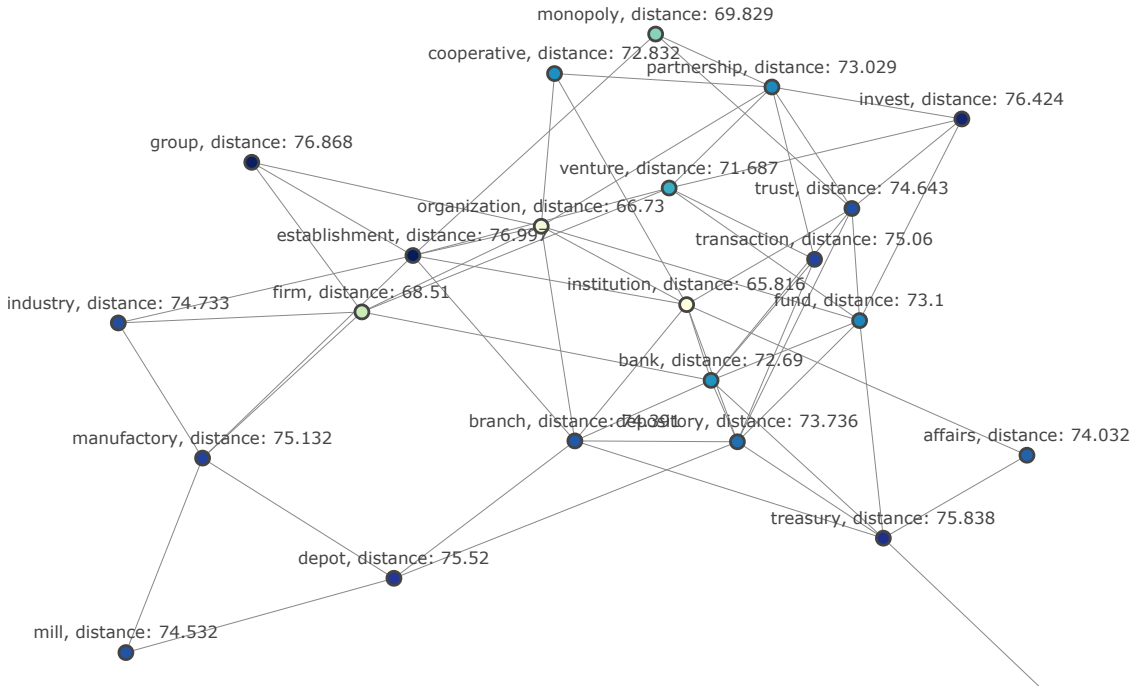


At $\epsilon = 77°$ the one cycle (‘fund’, ‘rganization’, ‘partnership’, ‘monopoly’, ‘trust’) gets filled in and hence dies but 4 more cycles come in and we have 10 cycles in total that give rise to the 2nd local homology $(\mathbb{Z}/2\mathbb{Z})^{10}$. The 4 new cycles are:

- ‘fund’, ‘trust’, ‘partnership’, ‘organization’.
- ‘manufactory’, ‘establishment’, ‘branch’, ‘depot’.
- ‘bank’, ‘institution’, ‘establishment’, ‘group’, ‘firm’.
- ‘monopoly’, ‘establishment’, ‘institution’, ‘trust’.

The cycles can be seen in Figure 4.9.

Figure 4.9: A part of the link of $v_{\text{corporation}}$ at $\epsilon = 77°$; distance is the geodesic distance to $v_{\text{corporation}}$.



Afterwards the cycles start dying out and at $\epsilon = 81°$ we observe the 2nd local homology of $v_{\text{corporation}}$ being trivial.

Higher local homology groups of a vertex correspond to higher homology groups of the link but are harder to interpret intuitively. Of course, the same principle of creating (higher order) cycles works.

4.3.2 Results on D_{GloVe}

On this dataset we have performed experiments for all $\epsilon \in \{40°, 41°, \ldots, 74°\}$.

In the same way as in the previous subsection, we present a table of words with interesting local homology (Table 4.2) and then look closer at a few words.

Table 4.2: Words with interesting homology (GloVe)

Word	Value of ϵ	Local homology
bank	74	0 0 1 1
bank	68	0 2 2
bank	66	0 1 4
bank	63	0 2 1
bank	62	0 2 1
corporation	64	0 1 2
factory	70	0 1 1
fund	68	0 1 0 1
house	68	0 1 1
invest	73	0 0 2 1
invest	72	0 0 2 1
invest	68	0 1 3 1
river	68	0 1 1
river	66	0 2 1
stock	74	0 1 1
trade	74	0 0 1 3
trade	73	0 0 1 1
trade	72	0 0 1 1
transaction	68	0 1 1
waste	72	0 1 3
waste	71	0 1 2
waste	68	0 1 3

Here as well we will use some notation for the sake of brevity. The same conventions from the last part hold but we will write S for D_{GloVe} and **not** $D_{\text{skip-gram}}$.

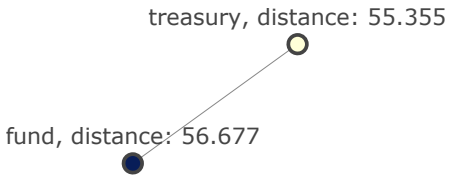
4.3.2.1 Understanding 1st local homology

The intuition behind the 1st local homology from the previous section still holds here. Without repeating ourselves we dive straight into a few examples.

Looking at v_{bank}, we see that in general we observe lower 1st local betti numbers compared to the previous dataset. In the previous dataset we first observed 6 "lone" connected components in the link before the vertices started connecting to each other but here it happens for lower ϵ values. Here, as seen in Figure 4.10, we observe a connected component in the link that is not a "lone" vertex for quite a low $\epsilon = 58°$, and before that we had only 2 "lone" vertices in the link, rather than 6. However, we still observe this phase of "lone" vertices coming in, generating 1st local homology

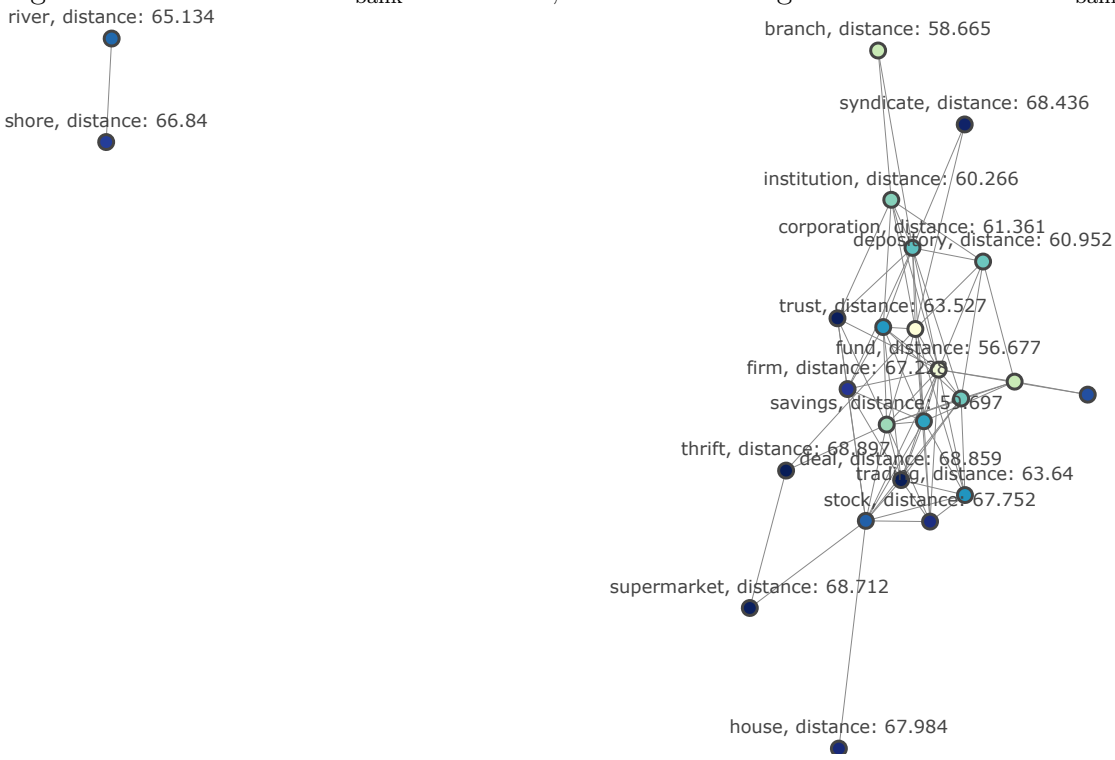
for a short period of time and then connecting to the main component. From our observations this generalises to other word vectors as well, not just v_{bank}. This phase usually encompasses a mid-range of ϵ-values.

Figure 4.10: The link of v_{bank} at $\epsilon = 58°$; distance is the geodesic distance to v_{bank}.



A more interesting observation is that at $\epsilon = 66°$ the link of v_{bank} has a connected component consisting of a vector corresponding to 'river', which is reflected in the 1st local homology being $\mathbb{Z}/2\mathbb{Z}$. This river related connected component persists longer than in the other dataset's case – until $\epsilon = 70°$ at which point the river related component connects to the main (finance related) one via the vector of 'house'. Again, we see the fact that 'bank' has two very different meanings is reflected in its local homology as well. From a persistence point of view, it is even more pronounced in this dataset.

Figure 4.11: The link of v_{bank} at $\epsilon = 69°$; distance is the geodesic distance to v_{bank}.



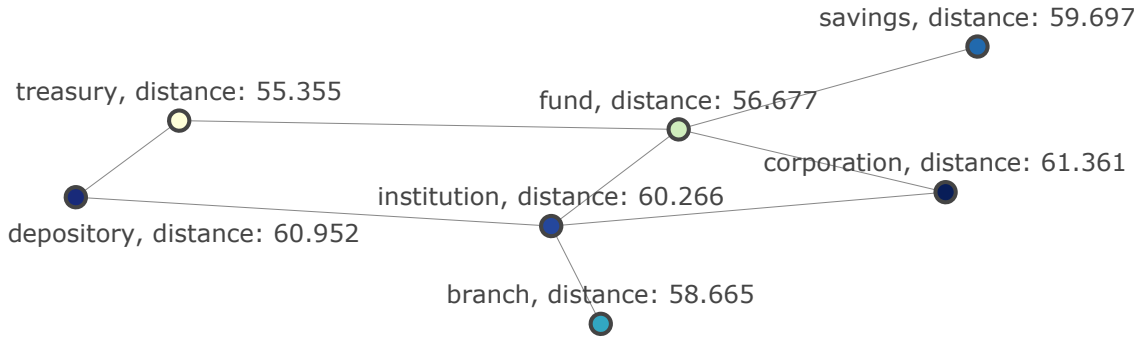
4.3.2.2 Understanding 2nd local homology

The intuition behind the 2nd local homology from the previous section still holds here so let us go straight into a few examples.

We observed earlier that the 1st local homology in this dataset is usually of lower dimension compared to the other dataset. We observe a similar thing in the 2nd local homology case as well, though perhaps a little less.

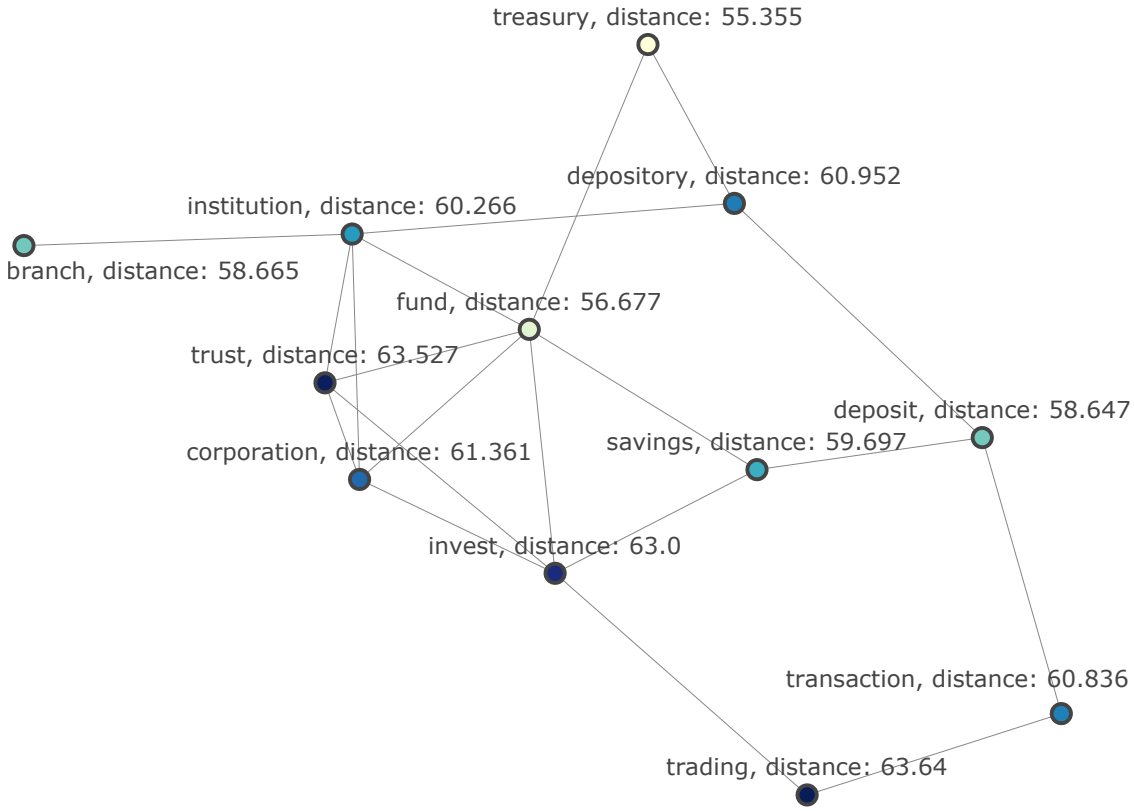
An interesting thing that we observe is a cycle that is common to both datasets. To see this, we can look again at v_{bank} when $\epsilon = 62°$. We have the 2nd local homology being $\mathbb{Z}/2\mathbb{Z}$ detecting the existence of the cycle 'depository', 'treasury', 'fund', 'institution' as seen in Figure 4.12, which actually persists for long – it dies at $\epsilon = 68°$. This cycle is homologous to 'depository', 'treasury', 'fund', 'corporation', 'institution' – a cycle of vectors of those words we also observed in the previous dataset when $\epsilon = 74°$. Also, by replacing 'corporation' with 'institution' in the original cycle, we obtain a cycle observed in the previous dataset at $\epsilon = 75°$.

Figure 4.12: A part of the link of v_{bank} at $\epsilon = 62°$; distance is the geodesic distance to v_{bank}.



Of course, there are cycles like 'invest', 'savings', 'deposit', 'transaction', 'trading' appearing at $\epsilon = 65°$, which look like nothing we have seen in the previous dataset – Figure 4.13. In fact, most cycles are "new" in the sense that they do not generate non-trivial elements of the local homology groups in the last dataset. Also, we do not see any point with extremely high 2nd local homology – like 'corporation' in $D_{\text{skip-gram}}$. In this dataset, the vector corresponding to 'corporation' has at most 2nd local betti number being 2 and the cycles that generate 2nd local homology are all different from the previous dataset.

Figure 4.13: The link of v_{bank} at $\epsilon = 65°$; distance is the geodesic distance to v_{bank}.



4.4 Discussion

As we can see, local homology does detect interesting structures in the datasets that we are using. Even though the generators of homology are vastly different in the two datasets, we see that the fact that 'bank' is a homonym can be seen in both of them by looking at the generators of 1st local homology. By no means is this conclusive but it does suggest that local homology carries relevant information to the problem of word sense disambiguation and therefore this approach is worth exploring further.

However, there still are a few problems with the algorithm. As it was noted earlier, the clusters have little structure. There can be a few reasons for this:

- The datasets are too small.
- the VR-complex is not the right one to use in this case.

- The algorithm is not robust to noise and data is noisy.

It would be useful to run the algorithm for bigger datasets but due to computational and time limitations we are unable to do so. However, this is definitely part of future work. Also, it would be very beneficial to better understand how the Vietoris Rips construction affects local homology and to compare how the results change if we use different constructions (e.g. the Čech complex). Again, due to computational limitations, we used the computationally cheaper construction, which is the VR-complex. Regarding robustness, we saw earlier (Figure 2.6) how one can have homotopy equivalent simplicial complexes that have different local homologies. In fact, given a value of ϵ, small perturbations can easily change the local homology of many simplices in $\mathrm{VR}_\epsilon(S)$. As it was suggested earlier, one could change the setting of the algorithm and look at persistent local homology instead.

Despite the limitations of this algorithm, we believe that it is beneficial to study word embedding problems using local topological constructions (e.g. local persistent homology) with the hope of coming up with a word-sense disambiguation algorithm.

Chapter 5

Stratification learning

In this section we discuss how the extension of the local homology clustering algorithm to a stratification learning algorithm can benefit us in connection to providing insight for the WSD problem. A stratification learning algorithm based on local homology is presented in [23] and when speaking of stratification learning, we will have the algorithm from [23] in mind. An alternative stratification algorithm using ideas from persistence exists and can be found in [1].

We note that the paper [23] develops and proves the correctness of the algorithm using local cohomology. However, since we are working over a field and with finite simplicial complexes, we have that local homology and cohomology groups are isomorphic. It is not too hard to see this and we will use the Universal Coefficient Theorem (UCT). See [31, Theorem 3.6.5] for a more general version of UCT or [15, Theorem 3.2] for a less general one. If $\mathbb{F}$ is any field and X is a finite simplicial complex, then by UCT we have for all $n \in \mathbb{N}$ that $H^n(X;\mathbb{F}) \cong \mathrm{Hom}(H_n(X;\mathbb{F}),\mathbb{F}) \oplus \mathrm{Ext}(H_{n-1}(X;\mathbb{F}),\mathbb{F})$. Now since we are working over a field, $\mathrm{Ext}(H_{n-1}(X;\mathbb{F}),\mathbb{F}) = 0$. Also, as X is finite, $H_n(X;\mathbb{F})$ is a finite-dimensional $\mathbb{F}$-vector space and hence is isomorphic to its first dual, which is $\mathrm{Hom}(H_n(X;\mathbb{F}),\mathbb{F})$. Therefore, we get that $H^n(X;\mathbb{F}) \cong H_n(X;\mathbb{F})$. This also applies for local homology and cohomology since Lemma 2.3 shows that local homology is isomorphic to regular homology of a simplicial complex (the link), and similar dual results can be proven for local cohomology.[1]

[1] If you recall the proof of Lemma 2.3 once we got the bijections f_i we could have applied the cochain functor rather than the chain functor, and it would have worked just as well.

5.1 Motivation

The main motivation to study stratification comes from manifold learning, which is concerned with the problem of, given a point cloud, detecting a manifold the point cloud was sampled from. However, if the point cloud is better modelled as a stratified space rather than a manifold, it might be more useful to firstly cluster the points by strata and then use each stratum (which is a manifold) as an input to manifold learning algorithms.

Hence we think that there are three motivating reasons to study stratification learning, as noted in [23]:

- One can possibly boost performance of manifold learning algorithms by using stratification learning as a pre-processing step.
- By knowing stratification of a space one knows the intrinsic dimensionality of different parts of the space (since dimensionality of each stratum is known).
- Lower-dimensional strata might be interpreted as somewhat anomalous and one might be interested in detecting that.

From the perspective of word embeddings, as was noted earlier, the hope is that once we stratify the point cloud of word vectors with respect to a word embedding, we would see words with many meanings being at the intersection of different strata (hence constituting a low-dimensional stratum) and each high-dimensional stratum would correspond to a topic or meaning.

5.2 Idea of the algorithm

The algorithm, the idea of which we present, computes the canonical stratification of a finite simplicial complex (or a finite regular CW complex) in the sense of Definition 2.27. Due to space limitations, we present a brief idea of the algorithm and mostly discuss its applications to word embeddings, expected results and improvements. For the full algorithm and proof of correctness the reader is referred to [23].

What we did with local homology clustering was a weaker version of stratification learning. In fact, given a point cloud S, the existence of a local homology preserving path in $G_\epsilon(S)$ is a necessary (though not sufficient) condition of the two points

belonging to the same canonical stratum of $\mathrm{VR}_\epsilon(S)$. To strengthen the condition and make it sufficient one needs to look at local homologies of n-simplices for $n > 1$ and not just vertices and edges.

5.3 Discussion

The starting point of this algorithm is a finite regular CW complex. However, in most applications one starts with a point cloud S. So a natural question arises in how to extend such an algorithm to one that takes a point cloud as an input. We would like to cluster points in S together iff they belong to the same canonical stratum with respect to some particular simplicial complex. One simple idea might be to use the standard constructions like VR or Čech complexes and just stratify them. However, we have seen that even given the Nerve Theorem, there is no theoretical guarantee to recover local homology but only for the "global" homology groups.

Despite these theoretical considerations, if we extend the algorithm in this way, we would get an algorithm that would cluster datapoints in an even finer grade than the local homology clustering algorithm from Chapter 4. This is because the clustering condition of the local homology clustering algorithm is a necessary condition for two points to lie in the same canonical stratum of $\mathrm{VR}_\epsilon(S)$, and to make the condition sufficient, more things would have to be enforced. So in practice, even fewer points would be clustered together. That is, to cluster the datapoints according to the canonical stratification of $\mathrm{VR}_\epsilon(S)$, we would need to take the clusters from local homology clustering and break them down even further by imposing some further conditions like the isomorphisms of local homology of higher dimensional simplices, not just edges. Hence, we see that at least on our datasets, we would get even less interesting structure in the clusters. Also, such an approach yields an algorithm which is not robust to noise: a small perturbation in the data for a given ϵ value can introduce or delete a simplex, which can change the local homology and make some local homology groups non-isomorphic hence changing the structure of clusters.

A more realistic approach, in our opinion, is to consider the persistent version of local homology and instead of checking for isomorphic local homology groups (of edges as well as higher-dimensional simplices), we would check for the barcodes to be close (closeness parametrised by some $d \in \mathbb{R}_{\geq 0}$) with respect to some distance (e.g. bottleneck or Wesserstein distance)). Then the algorithm would also be more robust with respect to noise.

An interesting generalisation of the stratification algorithm is to try and learn the coarsest F-decomposition of a finite simplicial complex X given a sheaf F (in the sense of Definition 2.29). Taking the sheaf F to be the local cohomology sheaf and then learning F-decomposition of X would not yield a (cohomological) stratification of X since the pieces do not have to be (cohomological) manifolds. However, there might still be interesting applications of such an approach, perhaps using sheaves other than the local cohomology one. For the development of a learning algorithm of the coarsest F-decomposition of a finite simplicial complex (together with the existence and uniqueness proofs of such decomposition) see [2].

For any reader wishing to look more closely into [23] and [2], we note that the three algorithms (local homology clustering (1), F-decomposition where F is the local cohomology sheaf (2), and stratification learning (3)) form a strict hierarchy. By looking at the clustering conditions, it is not hard to note that the clustering condition of (3) implies the clustering condition of (2), which implies the clustering condition of (1). But the converse implications do not hold. So we will always have that each cluster produced by (3) lies within a cluster produced by (2), which in turn lies within a cluster produced by (1). Due to space limitations, we have not introduced all three algorithms in full and hence have not proven the hierarchy.

Chapter 6

Conclusions and future work

In this work we have presented some basic constructions of topological data analysis (TDA) as well as basics of word embeddings and argued that TDA is a promising framework to study word embeddings. We have also presented and implemented a local homology clustering algorithm and tested it on two datasets coming from word embeddings. We have seen that even though some word vectors exhibit interesting local structure that is captured by local homology, the limiting requirement of isomorphic local homology groups render the algorithm susceptible to noise and produce almost trivial clustering (at least on the datasets considered). Moreover, we have seen a broad idea of how such a local homology approach can be extended to a stratification learning algorithm. However, stratifying the VR-complex of a point cloud suffers from similar drawbacks to the local homology algorithm. As previously discussed, we think that if we want to work directly with a point cloud rather than a finite simplicial complex, moving to persistent local homology seems promising.

Therefore, we would expect future work to include:

- Looking into relaxing the limiting isomorphism condition by considering persistent local homology or other local constructions.
- Seeing how relaxing the conditions carry over to stratification learning.
- Exploring how stable such an approach is with respect to noise.
- If results are successful on word embedding datasets, introducing this approach as a word sense disambiguation algorithm.

Bibliography

[1] Paul Bendich, Bei Wang, and Sayan Mukherjee. "Local homology transfer and stratification learning". In: *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics. 2012, pp. 1355–1370.

[2] Adam Brown and Bei Wang. "Sheaf-Theoretic Stratification Learning". In: *arXiv:1712.07734* (2017).

[3] Gunnar Carlsson. "Topology and data". In: *Bulletin of the American Mathematical Society* 46.2 (2009), pp. 255–308.

[4] Gunnar Carlsson et al. "Persistence barcodes for shapes". In: *International Journal of Shape Modeling* 11.02 (2005), pp. 149–187.

[5] Ronan Collobert et al. "Natural language processing (almost) from scratch". In: *Journal of Machine Learning Research* 12.Aug (2011), pp. 2493–2537.

[6] Wikimedia Commons. *Simplicial complex link*. 2016. URL: https://en.wikipedia.org/wiki/File:Simplicial_complex_link.svg (visited on 03/18/2018).

[7] Wikimedia Commons. *Simplicial complex star*. 2016. URL: https://en.wikipedia.org/wiki/File:Simplicial_complex_star.svg (visited on 03/18/2018).

[8] *Computing Homology*. 2014. URL: https://triangleinequality.wordpress.com/2014/01/23/computing-homology/ (visited on 02/26/2018).

[9] Justin Curry. "Sheaves, Cosheaves and Applications". University of Pennsylvania, 2013.

[10] Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction*. Providence, USA: American Mathematical Society, 2010.

[11] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. "Topological Persistence and Simplification". In: *Discrete & Computational Geometry* 28 (2002), 511—533.

[12] *English Vectors GloVe (medium) v1.0.0*. 2017. URL: https://github.com/explosion/spacy-models/releases/en_vectors_glove_md-1.0.0 (visited on 02/26/2018).

[13] Robert Ghrist. “Barcodes: the persistent topology of data”. In: *Bulletin of the American Mathematical Society* 45.1 (2008), pp. 61–75.

[14] Mark Goresky and Robert MacPherson. “Intersection homology II”. In: *Inventiones Mathematicae* 72.1 (1983), pp. 77–129.

[15] Allen Hatcher. *Algebraic Topology*. Cambridge, UK: Cambridge University Press, 2002.

[16] Omer Levy and Yoav Goldberg. “Linguistic regularities in sparse and explicit word representations”. In: *Proceedings of the eighteenth conference on computational natural language learning*. 2014, pp. 171–180.

[17] Omer Levy and Yoav Goldberg. “Neural word embedding as implicit matrix factorization”. In: *Advances in neural information processing systems*. 2014, pp. 2177–2185.

[18] Saunders Mac Lane. *Categories for the Working Mathematician*. 2nd ed. New York, USA: Springer, 1998.

[19] Paul Michel, Abhilasha Ravichander, and Shruti Rijhwani. “Does the Geometry of Word Embeddings Help Document Classification? A Case Study on Persistent Homology Based Representations”. In: *arXiv:1705.10900* (2017).

[20] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. “Linguistic regularities in continuous space word representations”. In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2013, pp. 746–751.

[21] Tomas Mikolov et al. “Distributed representations of words and phrases and their compositionality”. In: *Advances in neural information processing systems*. 2013, pp. 3111–3119.

[22] Tomas Mikolov et al. “Efficient Estimation of Word Representations in Vector Space”. In: *In Proceedings of Workshop at ICLR*. 2013.

[23] Vidit Nanda. “Local cohomology and stratification”. In: *arXiv:1707.00354* (2017).

[24] Christopher Olah. *Deep Learning, NLP, and Representations*. 2014. URL: http://colah.github.io/posts/2014-07-NLP-RNNs-Representations/ (visited on 02/26/2018).

[25] Nina Otter et al. “A roadmap for the computation of persistent homology”. In: *EPJ Data Science* 6.1 (2017), p. 17.

[26] Steve Y Oudot. *Persistence Theory: From Quiver Representations to Data Analysis*. Providence, USA: American Mathematical Society, 2015.

[27] Jeffrey Pennington, Richard Socher, and Christopher Manning. “Glove: Global vectors for word representation”. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[28] Ishrat Rahman Sami and Katayoun Farrahi. "A Simplified Topological Representation of Text for Local and Global Context". In: *Proceedings of the 2017 ACM on Multimedia Conference*. ACM. 2017, pp. 1451–1456.

[29] Tadas Temcinas. *Local Homology Clustering*. 2018. URL: https://github.com/temcinas/applied_topology_project (visited on 03/16/2018).

[30] Hubert Wagner, Paweł Dłotko, and Marian Mrozek. "Computational Topology in Text Mining". In: *Computational Topology in Image Context*. Ed. by Alexandra Bac and Jean-Luc Mari. Berlin: Springer, 2012.

[31] Charles A Weibel. *An Introduction to Homological Algebra*. Cambridge, UK: Cambridge University Press, 1994.

[32] *word2vec*. 2013. URL: https://code.google.com/archive/p/word2vec/ (visited on 02/26/2018).

[33] Xiaojin Zhu. "Persistent Homology: An Introduction and a New Text Representation for Natural Language Processing". In: *Proceedings of International Joint Conference on Artificial Intelligence*. 2013, pp. 1953–1959.

[34] Afra Zomorodian. "Fast construction of the Vietoris-Rips complex". In: *Computers & Graphics* 34.3 (2010), pp. 263–271.