Homology Localization by Hierarchical Blowups

Ahmed Abdelkader *

Abstract

Topological descriptors such as the generators of homology groups are very useful in the analysis of complex data sets. It is often desired to find the smallest such generators to help localize the interesting features. One interpretation of localization utilizes a covering of the underlying space and computes generators contained within these covers. A similar construction was later used to compute persistence homology for smaller subsets in parallel before gluing the results. In this presentation, we describe a more efficient version of this construction and discuss how it can be used to find generators within a large class of subspaces.

1 Introduction

Persistent Homology is a crucial device in computational topology and finds wide application in data analysis and as a core component of a variety of algorithms. Although the formalism of homology provides efficient means to detect the existence of topological features, it cannot directly locate them within the space. This is often required to reason about the embedding of the data set in the measurement space and to allow further processing, e.g., cleaning up noises introduced in data collection or detecting holes in sensor networks.

One approach to locating topological features is to find the smallest generator for a given homology group [3]. Another approach is *localized homology* [6], which utilizes a covering of the space and computes homology bases compatible with the bases of the local pieces defined by the cover. However, choosing an appropriate cover was left to the domain expert.

The idea of computing homology through a cover was later reused to devise a parallel algorithm based on a hierarchical decomposition of the domain [4]. The algorithm performs reduction on local pieces in parallel before gluing the results, which is inherently expensive. Potentially better ways to go about gluing are provided by spectral sequences [2, 5].

In this presentation, we take a different look at homology localization utilizing the model of computation developed for the parallel setting. Using a hierarchical space decomposition, we aim to quickly report the topological descriptors within nearly arbitrary covers by gluing partial precomputed results. Motivated by recent developments in approximate range queries [1] we anticipate similar notions in topological data analysis.

We start by revisiting the construction used in [4] to enable finer decompositions of the domain as required for range queries. Then, we discuss the anticipated notion of *topological range queries* and their applications.

2 Preliminaries

A topological space X may be represented by a *simplicial complex* K. A *filtration* is a nested sequence of simplicial complexes $K_0 \subseteq K_1 \subseteq \cdots \subseteq K_n = K$. Given a set of subcomplexes as a cover $\mathcal{C} = \{K^i\}_{i \in I}$, with $K = \cup\, \mathcal{C}$, the *Mayer-Vietoris Blowup Complex* $K^{\mathcal{C}} \subseteq K \times I$ is defined as:

$$K^{\mathcal{C}} = \cup_{J \subseteq I} \cup_{\sigma \in K^J} \sigma \times J, \quad \text{where } K^J = \cap_{j \in J} K^j. \quad (1)$$

Intuitively, the blowup complex creates one copy of each simplex $\sigma \in K$ for each of the covers containing it. This allows each cover to be processed independently. The blowup complex also includes additional copies of simplices where each subset of covers overlap. This marks the locations where these covers should be glued together to recover the original space.

Formally, the projection $\pi : K^{\mathcal{C}} \to K$ takes an element of the blowup to its first factor and induces a map on homology $\pi^* : H(K^{\mathcal{C}}) \to H(K)$. As π is a homotopy equivalence [6], π^* is an isomorphism. Then, by the Persistence Equivalence Theorem, the persistent homology of K can be computed from $K^{\mathcal{C}}$ [4].

3 The Hierarchical Blowup Complex

We start by formalizing the notion of the blowup complex for a hierarchical cover as introduced in [4].

Definition 1 *A Hierarchical Cover of height h is a system of covers $\mathcal{H} = \{\mathcal{C}_i\}_{i \in [h]}$, with $\mathcal{H} = \cup\, \mathcal{C}_i$ $\forall i$, such that $\forall\, c^i_\alpha \in \mathcal{C}_i$, there is a unique j-parent, $c^j_{p_j(\alpha)} \in \mathcal{C}_j$, $\forall j \in [i]$, satisfying $c^i_\alpha \subseteq c^j_{p_j(\alpha)}$, where $p_i(c^i_\alpha) = c^i_\alpha$.*

Consider a simplex $\sigma \in K$ lying in the intersection of two subcovers at the lowest level, i.e., $\sigma \in c^h_x \cap c^h_y$ for $x \neq y$. We track σ through the hierarchy until it possibly falls into a single subcover at a higher level. Note that $\sigma \in c^{i+1}_\alpha \implies \sigma \in c^i_{p_i(\alpha)}$.

*University of Maryland, College Park, akader@cs.umd.edu

Define the set of all levels where σ falls into a pairwise intersection as $\lambda = \{j \in [h] \mid p_j(x) \neq p_j(y)\}$. It follows that σ is contained in the intersection of $h + |\lambda|$ covers. Letting $\mathcal{F}_h = \cup_{i \in [h]} \mathcal{C}_i$, it follows from the product formula of the blowup complex that $K^{\mathcal{F}_h}$ will generate $2^{h+|\lambda|} - 1$ copies of σ. Although the cover sets form a hierarchy, they are distinct sets and $K^{\mathcal{F}_h}$ as defined in [4] does not seem to exploit their nesting structure. This makes it infeasible to work with deeper hierarchies where h grows as a function of $|K|$ as $|K^{\mathcal{F}_h}| = O(2^h)|K|$.

To remedy this, we propose a recursive construction. Given a hierarchical cover $\mathcal{H}$, we lift each level of the hierarchy to a decomposition of the blowup complex at the above level. We denote the lifted cover $\mathcal{C}_i$ as $\hat{\mathcal{C}}_i$.

Definition 2 *Given a hierarchical cover $\mathcal{H} = \{\mathcal{C}_i\}_{i \in [h]}$, the Hierarchical Blowup Complex $\{K^{\mathcal{H}_i}\}_{i \in [h]}$ is defined recursively as $K^{\mathcal{H}_1} = K^{\mathcal{C}_1}$ and*

$$K^{\mathcal{H}_{i+1}} = {K^{\mathcal{H}_i}}^{\hat{c}_{i+1}}. \qquad (2)$$

Definition 3 *A lifted cover $\hat{\mathcal{C}}_i = \{\hat{c}^{i+1}_\alpha \mid c^{i+1}_\alpha \in \mathcal{C}_i\}$ where $\hat{c}^1_\alpha = c^1_\alpha$ and*

$$\hat{c}^{i+1}_\alpha = \{(\sigma, J_1, \ldots, J_i) \in K^{\mathcal{H}_i} \mid \sigma \in c^{i+1}_\alpha \wedge p_i(\alpha) \in J_i\}.$$

Now, for $\sigma \in c^h_x \cap c^h_y$ and $i = \min \lambda$, $K^{\mathcal{H}_i}$ generates

$$\{(\hat{\sigma}_{i-1}, p_i(x)), (\hat{\sigma}_{i-1}, p_i(y)), (\hat{\sigma}_{i-1}, \{p_i(x), p_i(y)\})\},$$

where $\hat{\sigma}_{i-1}$ is the lifted copy of σ in the common parent $\hat{c}^{i-1}_{p_{i-1}(x)}$. Further decomposing at level $j > i \in \lambda$, lifted simplices with only $p_{j-1}(x)$ or $p_{j-1}(y)$ get one new copy while mixed simplices again get three. Letting $s(j)$ be the number of copies of σ at level j, we get $s(i) = 3$ and $s(j) = s(j-1) + 2$ for $j > i$. Summing over all levels, $K^{\mathcal{H}_h}$ will only create $h + O(|\lambda|^2)$ copies of σ. If $\mathcal{H}$ only has pairwise intersections at any level, $|K^{\mathcal{H}_h}| = O(h^2)|K|$. For typical decompositions $h = O(\log |K|)$ and we get only a polylogarithmic expansion.

Finally, we define the projections $\pi_1 : K^{\mathcal{H}_1} \to K$ and $\pi_i : K^{\mathcal{H}_{i+1}} \to K^{\mathcal{H}_i}$. Again, these maps are homotopy equivalences [6] and it follows that the induced maps on the homology of the hierarchical blowup complex are isomorphisms. We get the following:

Theorem 1 *A filtration $K_1 \subseteq \cdots \subseteq K_i \subseteq \cdots \subseteq K$ induces filtrations of all levels of a hierarchical blowup complex $K_1^{\mathcal{H}_j} \subseteq \cdots \subseteq K_i^{\mathcal{H}_j} \subseteq \cdots \subseteq K^{\mathcal{H}_j}$. Passing to homology, we get a sequence of homology groups connected by isomorphisms at each level. By the Persistence Equivalence Theorem, the persistence pairs in all levels are the same.*

$$\begin{array}{ccccc}
H(K_1) \longrightarrow \ldots \longrightarrow & H(K_i) & \longrightarrow \ldots \longrightarrow & H(K_n) \\
\uparrow \pi^*_{1,1} & \uparrow \pi^*_{1,i} & & \uparrow \pi^*_{1,n} \\
H(K_1^{\mathcal{H}_1}) \to \ldots \to & H(K_i^{\mathcal{H}_1}) & \to \ldots \to & H(K_n^{\mathcal{H}_1}) \\
\uparrow \pi^*_{2,1} & \uparrow \pi^*_{2,i} & & \uparrow \pi^*_{2,n} \\
H(K_1^{\mathcal{H}_2}) \to \ldots \to & H(K_i^{\mathcal{H}_2}) & \to \ldots \to & H(K_n^{\mathcal{H}_2})
\end{array}$$

In this example, for a filtration with a 2-level blowup complex, the diagram commutes. Specifically, the persistence pairs of $\mathcal{C}_j$ encoded into $K^{\mathcal{H}_j}$ may be computed by gluing the results for subcovers at any level.

4 Topological Range Queries

We envision queries of the form $(\mathcal{Q}^r, \alpha, \epsilon)$, where $\mathcal{Q}^r$ is a *parameterized range* and α is a *filtration parameter*. For example, given a set of points $\mathcal{P} \subset \mathbb{R}^d$ the query can be defined over an *implicit* Vietoris-Rips Complex $\mathcal{R}^\alpha(\mathcal{P} \cap \mathcal{Q}^r)$, with approximations of both $\mathcal{Q}^r$ and $\mathcal{R}^\alpha$ within ϵ. The query can then ask for the Betti numbers at a given (r, α) or possibly the persistence barcodes for a range of parameter settings, e.g., $[r_0, r_1]$ or $[\alpha_0, \alpha_1]$.

In $\mathbb{R}^d$, ranges are often simple geometric primitives, e.g., hyperrectangles and hyperspheres. To support a larger class of embeddings, it would be interesting to define corresponding notions of ranges in more general metric spaces, for which a variety of algorithmic results in the Euclidean setting were adapted. Another direction is to consider ranges defined directly on the complex, e.g., geodesic balls, utilizing tools from graph theory like recursive separators.

References

[1] S. Arya, D. M. Mount, and E. Park. Approximate Geometric MST Range Queries. In *31st International Symposium on Computational Geometry (SoCG 2015)*, pages 781–795, 2015.

[2] U. Bauer, M. Kerber, and J. Reininghaus. Distributed computation of persistent homology. In *Proceedings of the Meeting on Algorithm Engineering & Experiments*, pages 31–38, 2014.

[3] T. K. Dey, A. N. Hirani, and B. Krishnamoorthy. Optimal homologous cycles, total unimodularity, and linear programming. *SIAM Journal on Computing*, 40(4):1026–1044, 2011.

[4] R. Lewis and D. Morozov. Parallel computation of persistent homology using the blowup complex. In *Proceedings of the 27th ACM Symposium on Parallelism in Algorithms and Architectures*, pages 323–331, 2015.

[5] D. Lipsky, P. Skraba, and M. Vejdemo-Johansson. A spectral sequence for parallelized persistence. *CoRR*, abs/1112.1245, 2011.

[6] A. Zomorodian and G. Carlsson. Localized homology. *Computational Geometry*, 41(3):126 – 148, 2008.