LOCAL HOMOLOGY DIMENSION AS A NETWORK SCIENCE MEASURE

Cliff Joslyn, Brenda Praggastis, Emilie Purvine, Arun Sathanur (Pacific Northwest National Laboratory)
Michael Robinson (American University)
Stephen Ranshous (North Carolina State University)

SIAM Workshop on Network Science 2016
July 15-16 · Boston

Summary

While topological methods are gaining prominence in other areas of data analytics [1], there is only sporadic attention to the topological roles played by specific graph structures, including vertices, edges, and cliques [3, 4]. We describe a class of measures on the clique (flag) complex of a network based on the local topological structure surrounding the multidimensional faces, and the role they play within the the overall graph structure. This representation of a graph as a topological complex admits an Alexandroff topology, and the dimensions of the local homology groups of neighborhoods of faces are computable as measures on nodes, edges, or any other higher-order cliques. Their propertries are considered and compared to other network measures both on the corresponding contracted neighborhoods in the graph, and in consideration of the planarity of the underlying graph. Examples are provided for both standard test and random graphs. We conclude with a couple of analytical results.

Local Homology on Flag Complexes of Graphs

For an undirected graph $\mathcal{G} = \langle V, E\rangle$ on a finite set of vertices V with edge set $E \subseteq V^2$, its flag complex $\mathcal{X}$ is a collection of nonempty subsets $F \subseteq V$ where $F \in \mathcal{X}$ if and only if F is a clique in $\mathcal{G}$. Each $F \in \mathcal{X}$ is a k-dimensional *simplex* or *face* in $\mathcal{X}$, where $k = |F| - 1$. $\mathcal{X}$ has the *Alexandroff topology*, whose open sets are arbitrary unions of "stars" $\star(F) = \{G \in \mathcal{X} : F \subseteq G\}$ for $F \in \mathcal{X}$. We also have the closure $\mathrm{cl}(F) = \{G \in \mathcal{X} : G \subseteq F\}$. In general for any set of faces, $Y \subseteq \mathcal{X}$, the star, $\star(Y)$, is the subset of $\mathcal{X}$ containing Y and the star of each of its elements. We similarly define the closure of any set of faces. For each set of faces $Y \subseteq \mathcal{X}$, define the 0-neighborhood of Y as $N_0(Y) = \star(Y)$ and for each $k > 0$, the k-neighborhood as $N_k(Y) = \star(\mathrm{cl}(N_{k-1}(Y))$. For every open subset, $Y \subseteq \mathcal{X}$, we introduce the measure $LH_j(Y) := \dim(H_j(\mathcal{X}, \mathcal{X}\backslash Y))$. For $j \geq 0$ $LH_j(Y)$ is the j'th local Betti number of the space (a cell complex) produced by taking the quotient of $\mathcal{X}$ by $\mathcal{X} \setminus Y$, collapsing everything outside of Y to a single point.

Fig. 1 shows an example, where on the left a graph on 7 nodes has the flag complex $\mathcal{X}$ shown with a tetrahedron, two triangles, and an edge as maximal faces. $N_0(\mathrm{cl}(\{A, K\}))$ is shown in green (the red points and edges are excluded), and the quotient space, $\mathcal{X}/(\mathcal{X} \setminus N_0(\mathrm{cl}(\{A, K\})))$ is on the right. Here we have $LH_1(N_0(\mathrm{cl}(\{A, K\}))) = 1$: focusing on $N_0(\mathrm{cl}(\{A, K\}))$ yields a single loop (shown), in that the points E and C are separated from V, T_1, and T_2. This is *not* the case, for example, with $LH_1(N_0(\{A, V\})) = 0$, since it sits on the boundary and does not divide the space.

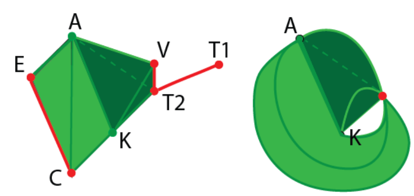

Figure 1: Example (left) flag complex of a graph and (right) the cell complex produced by focusing on the 0-neighborhood of the $\mathrm{cl}(\{A, K\})$.

Fig. 2 shows an example over a graph of USA borders: two states are connected when they have an adjoining border. As a flag complex, there is one maximal tetrahedron and one maximal edge, otherwise maximal triangles. The top and bottom show LH_1 and LH_2 of the 0-neighborhoods of all faces, as identified. We observe that LH_1 serves to identify cut faces and cuttable regions; while LH_2 serves to identify the border, including the four corners, which is measured as part of the border due to its high internal connectivity.

Observational Comparison With Network Measures

While comparing LH with vertex and edge measures used in network science is straightforward, to do so for higher dimensional faces we build a new face contracted graph,

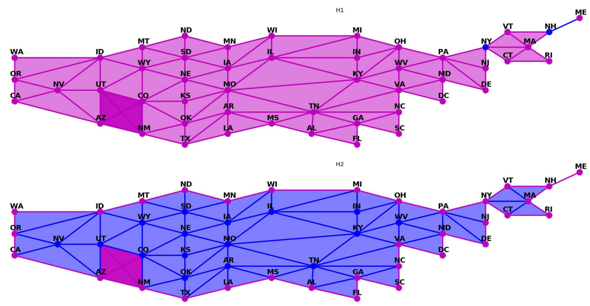

Figure 2: LH_k measures over the 0-neighborhoods of an example flag complex. (Top) $LH_1 = 0$ (magenta), $LH_1 = 1$ (blue); (Bottom) $LH_2 = 0$ (magenta), $LH_2 = 1$ (blue).

$\mathcal{G}^F$, in which vertices of $F \in \mathcal{X}$ are replaced by a single vertex, v_F, which will be adjacent to the union of the sets of neighbors of vertices in F that remain in $\mathcal{G}^F$. For any graph measure g (e.g. centrality), we let $g(F) := g(v_F)$ as computed in $\mathcal{G}^F$, noting that this will not alter the computation of any network measures on vertices.

We used NetworkX [2] and wrote a Python library [5] to calculate $LH_1(N_0(\text{cl}(F))$ on all faces, F, of (1) the Zachary Karate Club social network (34 vertices, 78 edges); (2) a synthetic Erdős-Rényi (ER) graph (40 vertices, 146 edges); and (3) a synthetic Barabási-Albert (BA) preferential attachment graph (40 vertices, 144 edges). Fig. 3 shows select scatter plots comparing LH_1 with centrality measures and clustering coefficient (CC) for the Karate club. We observed strong positive correlation between LH_1 and a number of centrality measures, and strong negative correlation with the local CC.

Analytical Results

We also have some analytical results which bolster our observations above. First, we have proved [4, Thm. 12] that when $\mathcal{X}$ is an abstract simplicial complex (as all flag complexes are), and connected, then for a face with $N_0(F) \subseteq \mathcal{X}$, $LH_1(N_0(F)) + 1$ is an upper bound on the number of connected components of $\mathcal{X} \setminus N_0(F)$; and when $H_1(\mathcal{X})$ is trivial (the usual, global homology), that upper bound is attained. Thus we have $LH_1(N_0(\{v\})) = C - 1$, where C is the number of connected components in the subgraph of $\mathcal{G}$ induced by the neighbors of v (not including v itself). This is visible in Fig. 2 for the NY and NH vertices and the NH-ME edge: with $LH_1(N_0(F)) = 1$, their removal splits the local vicinity into $1 + 1 = 2$ connected components; and these reach the upper bound since the (global) $H_1 = 0$.

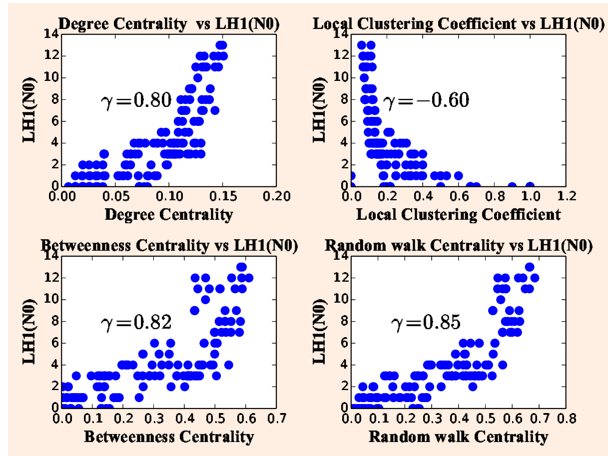

Figure 3: Scatter plots comparing network measures with LH_1 for the Karate network.

We have also found a functional relationship between CC and $LH_1(N_0(\{v\}))$. If $\mathcal{G}$ is planar, then the number of triangles incident to v is bounded by a linear function of the number of neighbors of v. This allows us to prove

$$\begin{aligned} d_v - 1 - \tfrac{d_v(d_v-1)CC(v)}{2} &\leq LH_1(N_0(\{v\})) \\ &\leq d_v - 1 - \tfrac{d_v(d_v-1)CC(v)}{6}, \end{aligned}$$

where d_v is the degree of v and $CC(v)$ is its clustering coefficient. The lower bound is true for any simple graph (i.e., not necessarily planar), but a similar upper bound cannot be shown for all simple graphs. This tells us that $LH_1(N_0(\{v\}))$ is bounded above by the pointwise maximum of a set of negatively sloped linear functions in $CC(v)$ sweeping out an "L" shaped curve. This is reflected in our experiments depicted in Figure 3 in which the data points are following the predicted upper bound. Though our graphs are not planar, the neighborhoods generally are which is sufficient for this result to hold.

References

[1] R. Ghrist. Barcodes: The persistent topology of data. *Bulletin of the American Mathematical Society*, 45:1:61–75, 2007.

[2] A. Hagberg, D. Schultz, and P. Swart. Exploring network structure, dynamics, and function using NetworkX. In *SciPy2008*, pages 11–15, Pasadena, CA USA, Aug. 2008.

[3] J. Jonsson. *Simplicial Complexes of Graphs*. Springer-Verlag, Berlin, 2008.

[4] M. Robinson. Analyzing wireless communication network vulnerability with homological invariants. *GlobalSIP 2014*, pages 900–904, Atlanta, GA, 2014. IEEE.

[5] M. Robinson, B. Praggastis, and C. Capraro. Python Sheaf Library `https://github.com/kb1dds/pysheaf`, 2016