Betti Numbers of Graphs with an Application to Anomaly Detection

David A. Johannsen David J. Marchette

Abstract

This paper describes an application of research that sits at the intersection of commutative algebra and combinatorics: Betti numbers of graphs. In particular, we describe a correspondence between simple undirected graphs and a class of ideals in a polynomial ring. We then briefly introduce some of the algebraic invariants that can be associated to the ideal and the relation of these invariants to the existence of induced subgraphs in the original graph. We discuss a novel application of the theory to a problem in anomaly detection – detection of a local non-homogeneity in a graph. We describe some variants of these ideas designed to make the computations more tractable.
Keywords: algebraic geometry; graph theory; anomaly detection; social network analysis.

1 Introduction

Consider the problem of detecting a region of anomalous activity in a graph. There are many ways one could define "anomalous activity"; informally, we mean that the structure of a small region (a small set of interconnected vertices) in the graph is fundamentally different from other regions. To make this explicit, consider a graph defined by the communications between entities: each vertex is an entity and there is an edge between two entities if they have communicated (called, sent email, met) within a given time window. We seek to find small sets of vertices that are communicating *differently* than other regions: they may have more/fewer edges than the norm; there may be a single central vertex through which nearly all communications pass; the communications may form a tree (indicating a hierarchical structure to their social network).

Several approaches to this problem are possible, depending on how much information one has on what types of anomaly are interesting, whether the graph is static or dynamic (a time series of graphs), what kinds of assumptions one can make on the structure of the graph or graphs, et cetera. See [1, 2, 3, 4]. The basic approach we consider here is to compute graph invariants, in this case a sequence of Betti numbers, and use these to detect the anomaly. These invariants can detect local anomalies when they are computed on neighborhoods

of vertices. We will first give some basic terminology and definitions, and then discuss the specific approach.

A graph is a pair $G = (V, E)$, where $V = V(G)$ is a finite set $\{v_1, \ldots, v_n\}$, the vertices, and $E = E(G)$ is a set of pairs of vertices, the edges. We will usually write vw for the edge $\{v, w\}$. We assume all graphs are simple: edges are pairs of distinct vertices; there are no edges from a vertex to itself. The number of vertices of a graph is called the *order* of the graph. The number of edges is called the *size*.

An *induced subgraph* of a set of vertices $W = \{v_{i_1}, \ldots, v_{i_k}\}$ is the graph (W, E'), where E' is the subset of E consisting of those pairs containing only elements of W: $E' = \{w_i w_j \in E | w_i, w_j \in W\}$. Thus, the induced subgraph contains all the edges between its vertices that exist in the original graph.

A *cycle* is a set of distinct vertices $\{v_{i_1}, \ldots, v_{i_k}\}$ with $i_k > 2$, such that $\{v_{i_1}v_{i_2}, v_{i_2}v_{i_3}, \ldots, v_{i_k}v_{i_1}\} \in E$. We will also refer to the corresponding graph as a cycle, and denote it C_n. A cycle has a chord if the subgraph induced by the vertices of the cycle has an edge between two non-adjacent vertices (non-adjacent as members of the cycle). The complete graph K_n is the graph on n vertices such that $uv \in E(K_n)$ for all $u \neq v \in V(K_n)$. The complete bipartite graph on n, m vertices is the graph for on the disjoint union of two sets ($V = X \cup Y$), where $|X| = n$ and $|Y| = m$, with all possible edges between the sets ($E = \{xy | x \in X \text{ and } y \in Y\}$). We denote the graph as $K_{n,m}$.

Definition 1.1. *A graph is* chordal *if all induced cycles have at least one chord.*

Definition 1.2. *The* open neighborhood *of a vertex v, denoted $N(v)$, is the set of vertices $\{w \in V | vw \in E\}$. That is, it consists of all neighbors of v. Note that $v \notin N(v)$. The* closed neighborhood *of v, denoted $N[v]$ is the set $N(v) \cup \{v\}$.*

Definition 1.3. *The* trivial graph *is the graph with no edges; $E = \emptyset$.*

Definition 1.4. *Given a graph G, define* star(G) *to be the graph defined by adding a single vertex v to G, with edges from v to each vertex of G. We will denote by* star(n) *the star on the trivial graph on n vertices. Thus star$(n) = K_{1,n}$ has order $n + 1$ and contains one vertex of degree n and n vertices of degree 1.*

Definition 1.5. *A vertex v is* simplicial *if $N(v) = K_n$ for some n.*

2 Commutative Algebra

Suppose we have a graph G on n vertices $\{v_1, \ldots, v_n\}$. Let k be a field (we may assume for the purposes of this paper that $k = \mathbb{C}$). Let $S = k[x_1, \ldots, x_n]$, the ring of polynomials in n variables with coefficients in k. Our notation for edges, $v_i v_j$ is evocative of monomials, and we make use of this observation in the definition of the edge ideal of a graph.

Definition 2.1. *The* edge ideal *of G is the ideal of S generated by the monomials $\{x_i x_j | v_i v_j \in E\}$. We write $\mathcal{I}(G)$ for the edge ideal of G.*

Although there is a one-to-one correspondence between the vertices v_i and the variables x_i, we will keep to the usual naming convention so that it is clear when we are referring to elements of the graph and when we are referring to elements of the ring. For more information on edge ideals see [5], [6], [7], [8], [9].

We can now use the tools of commutative and algebraic geometry to learn about a graph by studying its edge ideal. One such tool, the one we will be focused on in this paper, is the minimal free resolution (MFR).

Definition 2.2. *An* augmented free resolution *of an S-module M is an exact sequence of the form*

$$0 \longrightarrow F_m \longrightarrow F_{m-1} \cdots \longrightarrow F_1 \longrightarrow F_0 \longrightarrow M \longrightarrow 0$$

where each F_i is a free S-module (a direct sum of β_i copies of S). The image of F_i in the sequence is called the i^{th} syzygy module. *Such a resolution is* minimal *if m is minimal over all such, and each β_i is minimal. We define the* minimal free resolution *of the edge ideal to be the minimal free resolution of $I = \mathcal{I}(G)$, in which case the free resolution becomes:*

$$0 \longrightarrow S^{\beta_m} \xrightarrow{\phi_m} S^{\beta_{m-1}} \xrightarrow{\phi_{m-1}} \cdots \xrightarrow{\phi_2} S^{\beta_1} \xrightarrow{\phi_1} S^{\beta_0} \xrightarrow{\phi_0} I \longrightarrow 0.$$

The β_i are called the Betti numbers. *The length of the resolution is m. We will refer to these as* total Betti numbers *to distinguish them from their graded versions discussed below.*

It is well known that minimal free resolutions always exist, and are unique up to isomorphism ([9]). Thus, the length of the minimal free resolution of an edge ideal is well defined.

Definition 2.3. *The* projective dimension *of an edge ideal is the length of the minimal resolution.*

Note that $\beta_0 = \text{size}(G)$. There is a natural $\mathbb{N}^n$ grading on the ring $k[x_1, \ldots, x_n]$, which gives a natural grading on the resolution.

$$0 \longrightarrow \bigoplus_j S(-j)^{\beta_{m,j}} \longrightarrow \cdots \longrightarrow \bigoplus_j S(-j)^{\beta_{1,j}} \longrightarrow \bigoplus_j S(-j)^{\beta_{0,j}} \longrightarrow I \longrightarrow 0,$$

where $S(-j)$ is the shifted module obtained by shifting the degrees by j, so that the corresponding maps remain degree 0. We will be concerned with methods for computing the graded Betti numbers $\beta_{i,j}$. The total Betti number β_i is the sum over j of $\beta_{i,j}$.

3 Splitting

Definition 3.1. *An edge uv is a* splitting edge *if $N[u] \subset N[v]$ or $N[v] \subset N[u]$. If uv is a splitting edge, we will assume that the vertices are ordered so that $N[u] \subset N[v]$.*

Theorem 3.1 ([10]). *If uv is a splitting edge of G, then for all $i \geq 1$ and $j \geq 0$*

$$\beta_{i,j}(\mathcal{J}(G)) = \beta_{i,j}(\mathcal{J}(G \setminus \{uv\})) + \sum_{k=1}^{i} \binom{n}{k} \beta_{i-1-k,j-2-k}(\mathcal{J}(H)), \tag{1}$$

where $n = |N[v]| - 2$, $H = G \setminus N[v]$, $\beta_{-1,0} = 1$ and $\beta_{-1,j} = 0$ for $j > 0$. Recall that we are using the convention that uv is ordered so that $N[u] \subset N[v]$.

Proof. See [10]. □

Lemma 3.1. *A graph G has the property that all (non-trivial) induced subgraphs H contain a splitting edge if and only if G is chordal. Here "non-trivial" refers to the condition that H contain at least one edge.*

Proof. ($\Rightarrow$) Assume G is not chordal. Then there is an induced cycle C_n with $n > 3$ with no chord. But it is easy to see that any such cycle does not have a splitting edge.

($\Leftarrow$) If G is chordal, then it contains a simplicial vertex u. Since $N(u)$ is a clique, uv is a splitting edge for any $v \in N(u)$. Since any induced subgraph of a chordal graph is chordal, we have the result. □

Lemma 3.2. *For any chordal graph G there is a splitting edge e such that $G \setminus \{e\}$ is chordal. In fact, any edge incident on a simplicial vertex may be chosen for e.*

Proof. The only way removing an edge from G can make it non-chordal is if it opens up an induced C_4 with no chord. So in particular, we want to avoid removing chords. Let v be a simplicial vertex. Any edge e incident to v is a splitting edge. Further, it is not the chord of any cycle external to $N[v]$. Since $N[v]$ is complete, removing e cannot result in an induced cycle without a chord. Thus, any edge incident to a simplicial vertex can be removed. □

We call a function s that takes a graph containing splitting edges and returns one splitting edge a *splitting edge selection strategy*, or simply a *strategy*. We will say that Equation (1) is *recursive* for a class of graphs if there is a strategy for which it is recursive. We do not require that any arbitrary strategy will work, only that there is at least one. The following theorem has been stated elsewhere (see [11]), but is usually stated without explicit proof.

Theorem 3.2. *Under the strategy which selects splitting edges incident on simplicial vertices, Equation (1) is recursive for a graph G if and only if G is chordal.*

Proof. This is immediate from the two lemmas and the fact that induced subgraphs of chordal graphs are chordal. □

This shows that the algorithm to apply Equation (1) recursively always works to compute the minimal resolution of G whenever G is chordal, provided the splitting edges are chosen appropriately. It is not the case, though, that it

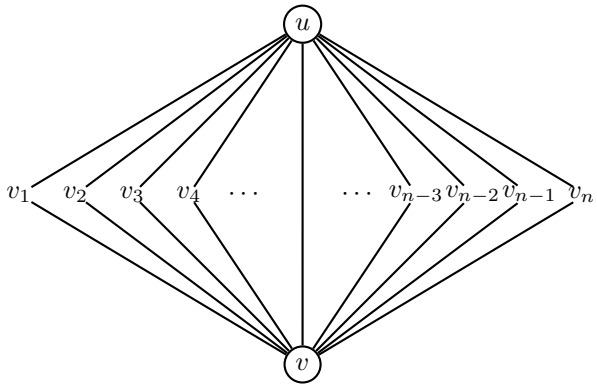

Figure 1: star(star(n)).

is recursive no matter what splitting edge is chosen at each step. The simplest example of this is shown below. It is clear that all edges in this graph are splitting and that this is a chordal graph. Further, if we remove any edge from the outside cycle, the resulting graph is chordal. However, if we remove the diagonal, the resulting graph is not chordal, and furthermore, does not contain any splitting edges. The two degree two vertices are simplicial, while the vertices on which the diagonal are incident are not.

The graph above is a specific case of a general class of graphs, $\{G = \text{star}(\text{star}(n))\}$, for $n > 1$, shown in Figure 1. G is not chordal. In fact there are a number of copies of C_4 in G. Note that every edge in star(star(n)) is splitting. However, all edges are not equal. If we choose uv, then $G \setminus uv$ contains no splitting edges, and Equation (1) fails to recurse. However, if we select any other edge (uv_i or vv_j) and continue to select such edges until only uv remains, Equation (1) recurses. At each step, H is the empty graph, since $N[u] = N[v] = V(G)$.

The strategy that works is to use a perfect vertex elimination scheme ([12] also called a perfect (vertex) elimination ordering (PEO) ([13]). This is an ordering of the vertices such that each vertex is simplicial in the graph induced by the remaining vertices. If one uses the strategy of selecting edges incident on vertices in a PEO, the algorithm recurses.

Now consider the graph in Figure 1 with the edge uv removed: $K = G \setminus \{uv\}$ (we will call this a *mace* (or a *mace head*) because of its similarity to the head of a type of mace; see below).

None of the edges is splitting, so we cannot even start the recursion. However, as we have noted, there is a strategy for using Equation (1) reflexively on G. Thus, we can turn Equation (1) around and recursively compute the Betti numbers for K using those of G:

$$\beta_{i,j}(\mathcal{I}(K)) = \beta_{i,j}(\mathcal{I}(K \cup \{uv\})) - \sum_{k=1}^{i} \binom{n}{k} \beta_{i-1-k,j-2-k}(\mathcal{I}(H)). \tag{2}$$

Thus, if an edge uv is missing from a graph, but would be splitting if added to the graph, we can use Equation (2). We will say that we "add a splitting edge" when we mean an edge which is splitting in the resultant graph. Note that the subgraph H is computed on the augmented graph $K \cup \{uv\}$ using $N[v] \subset V(K \cup \{uv\})$. In computing the first summand of the right hand side, we need not select uv as the splitting edge (and in fact we must not, since this would bring us back to our starting point). Instead, for the equation to be recursive, the addition of uv must induce other edges to be splitting in the resultant graph. This is the case in our example of $K = \text{star}(\text{star}(n)) \setminus \{uv\}$. If the resultant graph is chordal, then our PEO strategy will select the appropriate edges, and we can recurse to the solution.

Thus if adding a splitting edge results in a chordal graph, then we can recurse. Similarly, if there is a sequence of splitting edges we can add for which the final graph is chordal, then we can recurse. Note that algorithms that produce PEOs on chordal graphs can be modified (if necessary) to produce partial PEOs: a set of vertices each of which is simplicial in the graph induced by the vertices in the rest of the list plus all vertices not in the list. If the graph is non-chordal (and contains at least one simplicial vertex), we can apply our recursion to this partial PEO until we have processed all the splitting edges. Other methods must be employed to compute the Betti numbers for the remaining subgraphs. At this point, one could attempt to apply Equation (2). It is unclear, from a practical standpoint, whether it is better to employ Equation (2) up front, on the original graph, or only once we hit the end of the partial PEO.

There are three ways to add a splitting edge to a graph: one can connect an isolated vertex, resulting in a pendant; one can connect a pendant, resulting in a triangle; one can add a diagonal to a square (4-cycle). The first two don't change the "chordal status" of the graph: they cannot make a non-chordal graph chordal. The third requires that one of the vertices of the square have the property that its neighborhood is contained in the neighborhood of its opposite. Thinking of the mace head depicted in Figure 1 (with the edge uv removed), the vertex can be the "point" of the mace, resulting in the added splitting edge being uv. Note that this vertex is not simplicial, even after adding the edge, and so the added edge will not be selected by the algorithm until all the other edges of the mace have been selected. Note that we could have added $v_i v_j$ for any $i \neq j$, but this would not have resulted in a chordal graph (we would have to add many such edges). Thus, the strategy is to select splitting edges which make the graph "most chordal": fill in the maximum number of squares.

4 Approximations

The above algorithm works if and only if the graph is chordal. If it is not, then at some point in the recursion a graph is reached which has no splitting edges. At this point, one has three choices: push the graph off to a non-recursive algorithm, for example by a call to the symbolic algebra package *Singular* ([14]) or other algebraic geometry software; give up and fail to return any answer; use some method to approximate the Betti numbers so that what is returned is approximately correct (in some sense) rather than simply failing to return anything. In this section we discuss one method of approximation well suited to the recursive algorithm.

Given that the graph contains no splitting edges, theorem 3.2 cannot apply, as it only applies to splitting edges. It is possible, however, that if we chose a non-splitting edge, that the recursion would still be valid. It is easy to design a graph for which this will not work no matter which edge is chosen, however it is also known that in some cases choosing a non-splitting edge will still satisfy the recursion and result in the correct minimal free resolution. This is the basis of the approximate algorithm we have implemented.

1. While a splitting edge exists, proceed as according to the splitting edge algorithm above.
2. If no splitting edge exists, choose an edge to use in place of the splitting edge. This choice may be made:
   (a) At random.
   (b) So that the graph $G \setminus e$ with the edge removed contains the largest number of splitting edges (ties broken arbitrarily, and if no edge results in a subgraph with any splitting edges, choose the edge arbitrarily or at random).
3. Proceed as if the chosen edge were splitting.

5 Scan Statistics

Another approach to processing large graphs is to give up on the goal of producing a minimal free resolution of the entire graph, and instead consider the resolutions of subgraphs. One could cluster the graph, reducing to a graph defined by the clusters, but instead we consider scan statistics.

For each vertex v, define the scan subgraph $s(G, v)$ to be the induced subgraph of the closed neighborhood of v. We then define the scan MFR of G to be the collection of MFRs of all scan subgraphs of v. We can then define the scan statistic of the Betti numbers to be the maximum across all scan MFR's:

$$\beta_{ij}^s = \max_v \beta_{ij}(s(G, v)). \tag{3}$$

6 Anomaly Detection

We now consider the problem of detecting an anomaly in a graph. In this case we define "anomaly" to mean a small region (set of vertices) in the graph which have a greater number of communications amongst them than is "typical" in the remainder of the graph. We make this precise in the following.

The Erdös-Renyí random graph $\mathrm{ER}(n,p)$ is the random graph on n vertices whose edges are present independently with probability p. The $\kappa(n,p,m,q)$ random graph is a model for a graph with an anomaly: a small subset of vertices "communicating" amongst themselves at a higher rate. Here there are two disjoint sets of vertices, $V = \mathcal{K} \cup \mathcal{E}$, with an edge between vertices i and j independently with probability:

- p if either of $i, j \in \mathcal{K}$.
- q if both $i, j \in \mathcal{E}$.

It is our hypothesis that the Betti numbers, and in particular the scan MFR, can provide statistics for detecting this type of anomaly with higher power than other approaches. Note that the size of the graph is already known to have some power for detecting this type of anomaly ([1, 15, 16, 17]). It is reasonable, given that the Betti numbers essentially count the number of certain subgraphs, that they would have power superior to size, and perhaps other commonly used invariants, for problems of this type. To investigate this hypothesis, we present some simulations in the next section.

7 Examples and Simulation

Figure 2 shows a well-known graph called the Petersen graph. It's MFR is depicted in Figure 3. Here we display the total Betti numbers as a curve in the upper panel, with two data images below. In these, the values of the total (top) and graded (bottom) Betti numbers are depicted as gray-scale squares, with small values in black, large values in increasing whiteness. Zero graded Betti numbers are completely white. The numerical values are depicted in the squares. For large graphs with very large Betti numbers, we drop the numerical values from the plot.

We illustrate the timing improvement of the recursive algorithm over the algorithm in *Singular* in Table 1. For each order, we randomly generated 100 chordal graphs, and computed the time to calculate the MFR using each of the two algorithms. As can be seen, the recursive algorithm on these random chordal graphs is vastly faster than the algorithm in *Singular*, particularly for graphs of order 15. Figures 4 and 5 show timings of the recursive algorithm on larger graphs, both random chordal graphs and Erdös-Renyí random graphs. In the latter, *Singular* is called for those graphs that have no splitting edges, but these are typically small enough in these simulations that there is not an excessive time cost as a result.

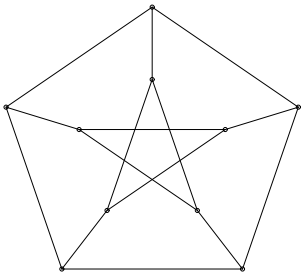

Figure 2: The Petersen graph.

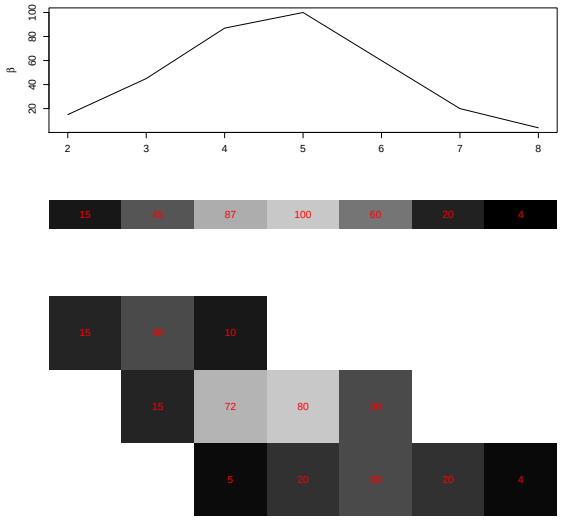

Figure 3: The minimal free resolution of the Petersen graph. The top plot shows the total Betti numbers β_i, with the x-axis corresponding to the index i. The image below shows the graded Betti numbers β_{ij} with the horizontal and vertical axes corresponding to the grading.

Table 1: Timings (in seconds) for the chordal algorithm as compared to *Singular* on a selection of chordal graphs.

Recursive Algorithm						
n	Minimum	1st Quartile	Median	Mean	3rd Quartile	Maximum
10	0.011	0.017	0.022	0.022	0.028	0.037
11	0.014	0.021	0.028	0.029	0.035	0.051
12	0.016	0.029	0.040	0.040	0.048	0.073
13	0.015	0.029	0.048	0.048	0.064	0.098
14	0.019	0.041	0.063	0.063	0.081	0.124
15	0.023	0.060	0.087	0.084	0.110	0.159
Singular						
n	Minimum	1st Quartile	Median	Mean	3rd Quartile	Maximum
10	0.014	0.032	0.070	0.102	0.12	0.415
11	0.014	0.05625	0.188	0.368	0.52	1.848
12	0.023	0.2333	1.154	1.875	2.84	7.816
13	0.024	0.417	3.084	6.999	11.5	42.42
14	0.038	3.646	25.86	52.09	74.97	275.4
15	0.090	40.86	174.1	320.5	476.4	1450

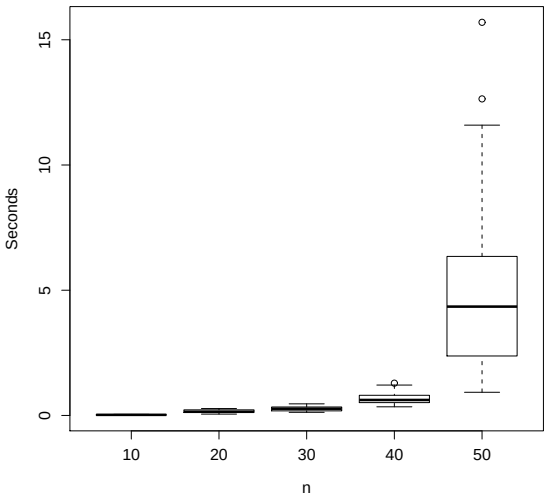

Figure 4: Timings for random chordal graphs.

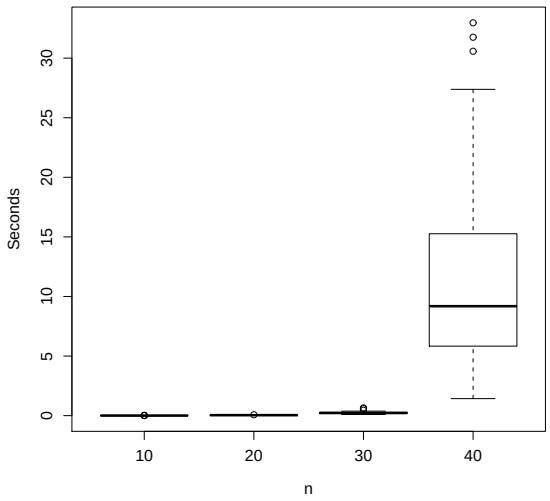

Figure 5: Timings for Erdös-Renyí random graphs.

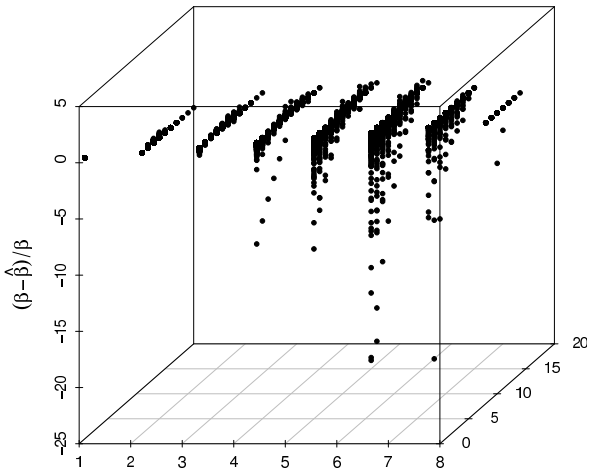

Figure 6: Percent difference between true Betti numbers and approximate Betti numbers using the approximation discussed in the text.

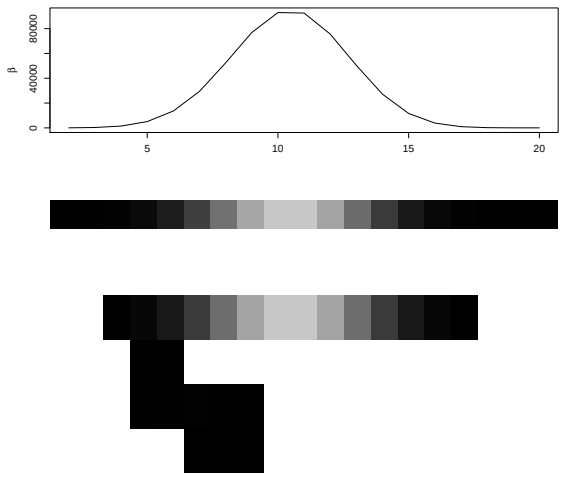

Figure 7: Scan statistic for an ER$(100, 0.1)$ graph. The logarithm of the values of the graded Betti numbers are displayed to better show the structure.

Figure 6 shows the percent difference between the true Betti numbers and approximate Betti numbers for 1000 ER$(20, .1)$ graphs.

Figures 7 and 8 show the scan MFR for homogeneous (ER) and inhomogeneous (κ) graphs. As one can see from the figures, the Betti numbers tend to be higher for the inhomogenious graph, indicating that these are likely to be useful for detecting this type of inhomogeneity. Since each graded Betti number corresponds to a count of the number of subgraphs of certain types, it is possible, at least in principle, to use Betti numbers to design tests to detect very specific types of deviation from homogeneity. This is a subject for future work.

8 Conclusion

We have discussed the proof of Hà and Van Tuyl's theorem that Equation (1) is recursive on chordal graphs, and used this, with the notion of a perfect elimination ordering (PEO) to provide an explicit strategy for applying this theorem. A trivial observation allows us to apply the same algorithm to slightly more general graphs than chordal graphs.

More generally, the idea of using a partial PEO can reduce the computation of the Betti numbers of an edge ideal in large graphs to that of computations on smaller graphs. This, combined with "filling in the squares" and other algorithms, or with specific formulas for different classes of graphs, can make the calculation of the Betti numbers of relatively large graphs practical.

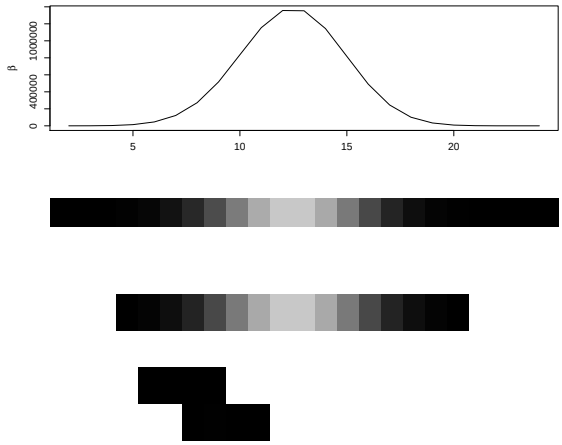

Figure 8: Scan statistic for a $\kappa(100, 0.1, 10, 0.8)$ graph. The logarithm of the values of the graded Betti numbers are displayed to better show the structure.

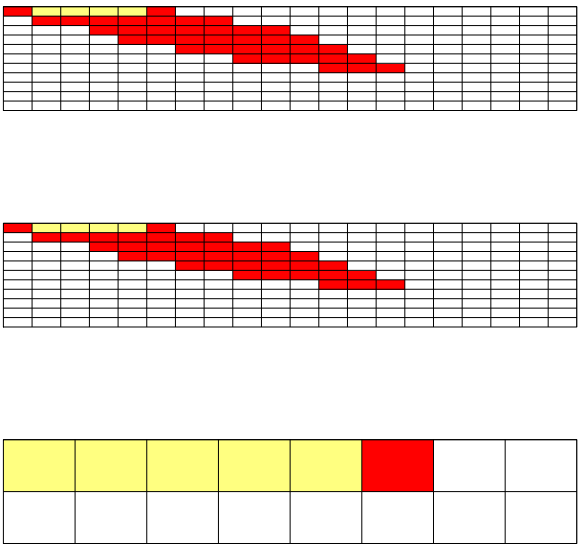

Figure 9: Powers for the true Betti numbers (top), approximate Betti numbers (middle) and scan Betti numbers (bottom) for a simulation from ER$(50, 0.05)$ versus $\kappa(50, 0.05, 6, 0.8)$. Yellow corresponds to high power, red to low, white to power ≤ 0.1. Here $\alpha = 0.05$. Here the upper left corner corresponds to the power of the size of the graph, $\beta_{1,1}$.

We have also discussed ways to approximate the Betti numbers for graphs that are not chordal, and to compute related statistics, the scan MFR, which can be computed relatively efficiently for many graphs using these algorithms. We demonstrated empirically that some of the Betti numbers have high power for detecting certain types of anomalies (non-homogeneities) in graphs. Much work is still needed to develop the theory of the scan MFR, and to explore applications of the Betti numbers of graphs.

Acknowledgments

This work was funded in part by the Office of Naval Research under the In-House Laboratory Independent Research program.

References

[1] C. E. Priebe, J. M. Conroy, D. J. Marchette, and Y. Park, "Scan statistics on Enron graphs," *Computational and Mathematical Organization Theory*, vol. 11, pp. 229–247, 2005.

[2] N. A. Heard, D. J. Weston, K. Platanioti, and D. J. Hand, "Bayesian anomaly detection methods for social networks," *Annals of Applied Statistics*, vol. 4, pp. 645–662, 2010.

[3] A. Rukhin and C. E. Priebe, "A comparative power analysis of the maximum degree and size invariants for random graph inference," *Journal of Statistical Planning and Inference*, vol. 141, pp. 1041–1046, 2011.

[4] E. Arias-Castro, E. J. Candès, and A. Durand, "Detection of an anomalous cluster in a network," *The Annals of Statistics*, vol. 39, 2011.

[5] S. Jacques, *Betti numbers of graph ideals*. PhD thesis, 2004.

[6] S. Jacques and M. Katzman, "The betti numbers of forests," 2005.

[7] R. P. Stanley, *Combinatorics and Commutative Algebra*. Basel: Birkhuser, second ed., 1996.

[8] R. H. Villarreal, *Monomial Algebras*, vol. 238 of *Monographs and Textbooks in Pure and Applied Mathematics*. New York: Marcel Dekker, Inc., 2001.

[9] E. Miller and B. Sturmfels, *Combinatorial Commutative Algebra*, vol. 227 of *Graduate Texts in Mathematics*. New York: Springer, 2005.

[10] H. T. Hà and A. Van Tuyl, "Splittable ideals and the resolutions of monomial ideals," 2006.

[11] H. T. Hà and A. Van Tuyl, "Monomial ideals, edge ideals of hypergraphs, and their minimal graded free resolutions," 2006.

[12] R. Balakrishnan and K. Ranganathan, *A Textbook of Graph Theory*. New York: Springer, 2000.

[13] D. B. West, *Introduction to Graph Theory*. Upper Saddle River, NJ: Prentice Hall, 2001.

[14] G.-M. Greuel and G. Pfister, *A Singular Introduction to Commutative Algebra*. Berlin: Springer, 2002.

[15] A. Rukhin, *Asymptotic analysis of various statistics for random graph inference*. PhD thesis, 2009.

[16] J. Grothendieck, C. E. Priebe, and A. L. Gorin, "Statistical inference on attributed random graphs: Fusion of graph features and content," *Computational Statistics and Data Analysis*, vol. 54, pp. 1777–1790, 2010.

[17] C. E. Priebe, Y. Park, D. J. Marchette, J. M. Conroy, J. Grothendieck, and A. L. Gorin, "Statistical inference on attributed random graphs: Fusion of graph features and content: An experiment on time series of Enron graphs," *Computational Statistics and Data Analysis*, vol. 54, pp. 1776–1776, 2010.