On the Spectra of Syntactic Structures

Isabella Senturia
Yale University
isabella.senturia@yale.edu

Robert Frank
Yale University
robert.frank@yale.edu

Abstract

This paper explores the application of spectral graph theory to the problem of characterizing linguistically significant classes of tree structures. As a case study, we focus on three classes of trees, binary, X–bar, and asymmetric c–command extensional, and show that the spectral properties of different matrix representations of these classes of trees provide insight into the properties that characterize these classes. More generally, our goal is to provide another route to understanding the structure of natural language, one that does not come from extensive definitions and rules taken by extrapolating from the syntactic structure, but instead is extracted directly from computation on the syntactically–defined graphical structures.

1 Introduction

In order to explore properties of natural and artificial language, the choice of representation is extremely important, as one is constrained to work within the tools existent for that representation. Motivated by immediate consituency theory, tree–structured graphical representations are the overwhelming favorite of syntacticians, capturing the multidimensionality inherent in the hierarchical structures of grammar. Modern graphical representations of syntax utilize binary trees: rooted tree graphs where each node branches into 0, 1 or 2 new nodes.

Syntacticians ask what constraints exist on tree structures by deriving properties of the acceptable structure and extrapolating from those potential rules and axioms governing natural language structures. All syntactic trees are rooted and downward branching. The most basic of restrictions syntacticians have imposed on a syntax tree is the branching factor of the nodes: it is widely assumed that syntactic trees are binary branching.

Another attempt by syntacticians to constrain permissible tree structures which accurately model natural language is *X–bar theory*: all phrases require the template of XP branching into specifier SpecXP and X′, and X′ branching into head X and complement CompX, as in Figure 1. SpecXP and CompX are optional—if they do not exist, neither do the edges connecting them to the structure (denoted by the dashed lines). If they do exist, they themselves have to follow the same structural guidelines of X–bar theory.

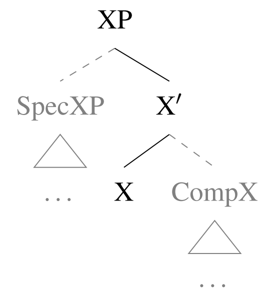


Figure 1: The requisite underlying structure of a phrase XP.

Kayne (1994) develops another restriction on possible tree structures by means of the Linear Correspondence Axiom (LCA), which states that the asymmetrical c–command relationship is a strict linear order (i.e. irreflexive, transitive, and asymmetric). Well–formed versus ill–formed trees can then be characterized as a result of the hierarchy (by way of the LCA and asymmetric c–command).

Frank and Vijay-Shanker (2001) suggest a partial order defined by a c–command relation as a primitive relation and that which should determine the hierarchy of syntactic tree structures (as opposed to dominance, by deriving dominance using the c–command relation). Frank and Kuminiak (2000) extended this idea to *asymmetric c–command*, suggesting that asymmetric c–command is a primitive relation, defining trees using this relation and arguing that this class is very similar to X–bar trees. Kuminiak (1999) considers classes of trees that are uniquely definable by some relation—more specifically, those that are uniquely defined by their asymmetric c–command relation, i.e. *asymmetric*

Proceedings of the Society for Computation in Linguistics (SCiL) 2023, pages 199-211.
Amherst, Massachusetts, June 15-17, 2023

c–command extensional (ACC).

Much of the work studying constraints on syntactic structures that accurately reflect properties of natural language has been done in a vein similar to the aforementioned work, by way of thinking about which structures are syntactically valid, and then attempting to generalize these properties. This paper provides an alternate route, one which directly studies syntactic classes from a mathematical perspective. While many properties are not derivable directly from the graphical structure, the aforementioned work demonstrates some which are. This paper explores the three previously–defined classes of trees—binary, X–bar, and ACC—from the vantage point of spectral graph theory.

Spectral graph theory (SGT) maps graphs to various matrix representations and analyzes spectral properties of these matrices. [1] Simple eigenvalue/eigenvector properties of a graph's matrix can be linked to properties of the graph that are often of high importance to the mathematician/computer scientist, such as graph–coloring and graph isomorphism (Wilf, 1967; Hoffman, 1970; Spielman, 2019; Chung, 1997; Godsil and Royle, 2001).

Researchers explore the distribution of eigenvalues of various graphs across the real numbers and concrete bounds on these distributions. A host of work explores whether graphs can be determined or distinguished by their spectra: cf. van Dam and Haemers (2003), Haemers and Spence (2004).

The notion of a tree has long existed within the mathematical subfield of graph theory, and trees have been extensively studied within both graph theory and spectral graph theory. Jacobs et al. (2021) study the distribution of eigenvalues of tree graphs. Dadedzi (2018) analyzes the spectra of various classes of trees, developing bounds on multiplicities of eigenvalues. Work has been done studying the spectrum of $k-$ary trees, trees where every non-leaf node has *branching factor*, i.e. degree, of $k \in \mathbb{N}$, and each leaf has degree 1 (He et al., 2000; Wang and Xu, 2006).

With respect to linguistic questions, Chowdhury et al. (2021) demonstrates an application of SGT to phylogenetic trees involving different graph isomorphism techniques. Ortegaray et al. (2021) use eigenvectors of the Laplacian matrix to detect relations between various vectors of syntactic parameter values.

SGT has not, however, been used to explore graphical properties of linguistic classes of tree structures. This paper demonstrates the utility in doing just that. It presents natural spectral properties of these trees that distinguish desirable classes of syntactic structures, exploring the extent to which these classes can be characterized by properties of their spectra.

The paper is structured as follows. Section 2 introduces the formal mathematical tools necessary: graph theory, matrix theory, and spectral graph theory. Section 3 explores spectral properties of the undirected graphs, before pivoting to those properties of directed graphs in section 4. Section 5 concludes.

2 Mathematical preliminaries

We present the mathematical notations and concepts of the paper, beginning with graph theory.

2.1 Graph Theory

Formally, we define a graph $G = (V(G), E(G))$, where $V(G) = \{v_1, v_2, \ldots, v_n\}$ is a set of n *vertices*, $E(G) = \{\{v_a, v_b\}, \ldots, \{v_p, v_q\}\}$ is a set of m *edges*.[2] We often abbreviate this notation to $G = (V, E)$, and label a set of k nodes with integers 1 through k. If the edges are undirected, the edge pair $\{v_i, v_j\}$ is unordered, whereas if the edge is directed, the edge pair is ordered $\{start, end\}$.

The *degree* d_v of a vertex v is the number of edges connected to that node. For directed graphs, we use *outdegree*, the number of edges leaving that node. We denote the set of (out)degrees of a graph G as $\mathcal{D}(G)$. A *leaf* is a node of degree 1 (or, in the case of a *directed graph*, i.e. digraph, outdegree 0). Two *adjacent* vertices are connected by a single edge. A *quasipendant* vertex is a vertex adjacent to a leaf. A *path* from some vertex v_i to another v_j is the sequence of edges connecting adjacent nodes between v_i and v_j. A graph is *connected* if there is a path from every node to every other node.

Graphs are often divided into *classes*. Graphs in a given class have one or more (often structural) unifying characteristics. The class of *trees* is the class of connected *acyclic graphs* $T = (V, E)$ defined by the existence of exactly one path connecting any two given vertices $v_1, v_2 \in V$—that is, they have no loops. A *directed tree* is a tree with directed edges. A *rooted* tree is a tree for which a

[1] We thus interchangeably refer to the spectra of a matrix representing a graph as the spectra of the graph.

[2] We follow the presentation of graph theory of Bondy and Murty (1976).

specific node has been designated as the root, and is graphed with this root at the top or bottom. Any directed tree, i.e. directed acyclic connected graph, will have a root: the node that has no edges entering it.

2.2 Spectral Graph Theory

Mathematicians have explored different ways to represent graphs, outside of the canonical picture of nodes and edges. Spectral graph theory, exploring algebraic representations of graphs by mapping graphs to various matrix representations, provides an approach to both explore what sorts of graphical properties (already observable through the graph–theoretic depiction) can be captured algebraically, and what new otherwise–unperceived properties emerge by virtue of the algebraic representation.

Spectral graph theory explores the link between algebra and graph theory by examining algebraic properties of matrix representations of graphs and how they reflect or represent combinatorial properties of these graphs.[3] We construct a mapping from a graph $G = (V, E)$ to a matrix $M \in \mathbb{F}^n \times \mathbb{F}^n$, where $\mathbb{F}$ is the field over which the entries of M are defined[4] and m_{ij} contains information about v_i, v_j, or the edge connecting them. Shifting between two different mathematical representations, a graph and a matrix, of the same mathematical object, allows both graphical/combinatorial and algebraic exploration of this object, permitting discovery of connections across these subfields that can be used to capture otherwise unascertainable properties of the graph.

A number of possible matrix representations are available for graphs, including the adjacency matrix A_G and diagonal matrix D_G (McKay, 1977).

Definition 2.1. *Given a graph $G = (V, E)$, we define the entries of the* adjacency matrix $A_G \in \mathbb{N}^{|V|} \times \mathbb{N}^{|V|}$ *as follows:*

$$a_{ij} = \begin{cases} 1 & \text{if } \{v_i, v_j\} \in E \\ 0 & \text{otherwise} \end{cases}$$

In the case of undirected graphs, the adjacency matrix will be symmetric (as $\{v_i, v_j\} \in E \iff \{v_j, v_i\} \in E$), whereas digraphs' adjacency matrices are not symmetric.

Definition 2.2. *Given $G = (V, E)$, let the* diagonal matrix $D_G \in \mathbb{N}^{|V|} \times \mathbb{N}^{|V|}$ *be defined as:*

$$d_{ii} = \sum_{j \in |V|} \mathbb{1}(\{v_i, v_j\}),$$

where the indicator function $\mathbb{1}(\{v_i, v_j\})$ is 1 when the edge $\{v_i, v_j\}$ exists, and 0 otherwise.

These d_{ii} values indicate the degree d_{vi} of each node v_i. So intuitively, D_G records the degree of each v_i in the i^{th} diagonal.

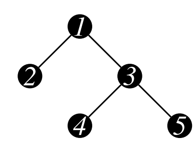
Given these two matrix representations of a graph, we can now define the Laplacian.

Definition 2.3. *Let $G = (V, E)$ be a graph with adjacency matrix A_G and diagonal matrix D_G. The* Laplacian *is defined as*

$$L_G = D_G - A_G$$

In the following example, we give an undirected binary tree with five nodes and construct its adjacency, diagonal and Laplacian matrix representations.

Example 2.4. *Consider the undirected rooted binary tree $G = (V, E)$ with $V = \{1, 2, 3, 4, 5\}$, $E = \{\{1, 2\}, \{1, 3\}, \{3, 4\}, \{3, 5\}\}$:*



$$A_G = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \ D_G = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$L_G = D_G - A_G = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 3 & -1 & -1 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{bmatrix}$$

An (uncommon) variation on the Laplacian, the signless Laplacian, is also relevant to this paper.

Definition 2.5. *Let $G = (V, E)$ be a graph with adjacency matrix A_G and diagonal matrix D_G. The* signless Laplacian *is defined as*

$$\hat{L}_G = |L_G| = D_G + A_G$$

After mapping the graph to a matrix representation, such as the Laplacian, we have all the tools of linear algebra at our disposal.

[3] Spielman (2019), Chung (1997) and Godsil and Royle (2001) form the basis of the following discussion.

[4] In this paper, we deal with the field of real numbers $\mathbb{R}$.

2.3 Spectral Theory

Spectral graph theory is based in *eigentheory*, the theory of eigenvalues and eigenvectors of matrices.

Definition 2.6. *A vector* $\psi \in \mathbb{R}^n$ *is an* eigenvector of matrix $M \in \mathbb{R}^n \times \mathbb{R}^n$ with eigenvalue $\lambda \in \mathbb{R}$ *if it is nonzero and if*

$$M\psi = \lambda\psi$$

For any matrix M and vector v (of the proper dimensions), the product Mv indicates M acting as a linear transformation via scaling and rotation. However, for all eigenvalues λ (a scalar) of M and their corresponding eigenvectors ψ, the equation $M\psi = \lambda\psi$ signals the ψ are those vectors for which M does not rotate but only scales by a factor of λ.

A matrix of dimension n has n (not necessarily unique) eigenvalues. We follow the convention of denoting this set of eigenvalues of a graph G's matrix representation M_G, known as the *spectrum* of M_G, as $\Lambda(M_G) = \{\lambda_1, ..., \lambda_n\}$, where the eigenvalues $\lambda_1, ..., \lambda_n$ are ordered from smallest to largest (that is, $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$). The *multiplicity* of an eigenvalue λ in the spectrum of M, denoted $\mu_M(\lambda)$, is the number of times that λ occurs. Within $\Lambda(M)$, an eigenvalue λ with multiplicity k is represented as λ^k.

Obviously any matrix representation of a graph changes with node labeling, as the node labels determine the position of node information in the matrix. However, the spectrum is *invariant* under permutation of the rows and columns of the matrix, meaning any permutation of the rows and (the same) columns of M yielding M' has the property that $\Lambda(M) = \Lambda(M')$. Thus, the spectrum of a graph is a useful way to explore properties of a graph as *isomorphic* graphs (graphs which are identical with a relabeling of nodes) have the same spectrum.

Spectral graph theory explores properties of these eigenvalues which have been extracted from the matrix of a graph to uncover combinatorial properties of the graph.

3 Spectral properties of undirected syntactic structures

This paper concerns the spectral properties of three classes of potentially syntactically–relevant graphs: binary trees, X–bar trees, and ACC trees.

Because the mathematics of undirected trees has been more widely studied, we begin with studying syntactic structures as undirected graphs. This ignores a crucial aspect of the tree structure assumed in linguistics—namely, the presence of a root node, and the ordered relationship between pairs of nodes (i.e. dominance). We completely ignore the issue of precedence among nodes so that trees are encoded entirely on the basis of their hierarchical relationships.

3.1 Generating classes

First, we define the three classes of graphs representing the three syntactic classes of binary, X–bar and ACC trees. Let `bin_base` be the smallest non–empty binary tree with three nodes, i.e. the three–noded path graph where $d_v = 2$ for the root v. Let $(T_\alpha, T_\beta) \uparrow \texttt{bin\_base}$ denote the simultaneous substitution of the trees T_α and T_β into the left and right leaves of `bin_base`, respectively. In what follows, we assume the trees to be unordered.

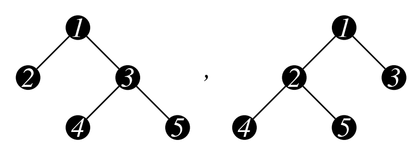
`Bin(n)` is the class of all binary (branching) trees with $n = 2k + 1$ nodes defined recursively as

$$\texttt{Bin(2k+1)} = \bigcup_{i=1}^{k-1} (T_\alpha, T_\beta) \uparrow \texttt{bin\_base}$$

over $T_\alpha \in \texttt{Bin(2i+1)}, T_\beta \in \texttt{Bin(2k-2i-1)}$, where T_1 is `single_node`, the single–noded tree.

Example 3.1. *Let* $T_\gamma = \texttt{single\_node}$ *and* $T_\delta = \texttt{bin\_base}$. *So* $\texttt{Bin(1)} = \{T_\gamma\}, \texttt{Bin(3)} = \{T_\delta\}$, *and*

$$\texttt{Bin(5)} = \{(T_\gamma, T_\delta) \uparrow T_\delta, (T_\delta, T_\gamma) \uparrow T_\delta\}$$

$= \{$  $\}$

`Xbar(n)` is the class of all X–bar trees with $n = 3k$ nodes. Define the base `xbar` tree as the path graph with three nodes:[5]

$$\texttt{xbar} = (\{v_1, v_2, v_3\}, \{\{v_1, v_2\}, \{v_2, v_3\}\}).$$

Define two new substitution operations specific to this syntactic class, $T_\chi \uparrow^*_{spec} \texttt{xbar}$ and $T_\chi \uparrow^*_{comp} \texttt{xbar}$ as inserting T_χ into the specifier or complementizer of the base `xbar` tree by connecting the root of T_χ to the top/root (XP) node or middle (X′) node of `xbar`, respectively, with a new edge.

[5]This can be understood from Figure 1 as the path with nodes XP, X′, and X. As SpecXP and CompX are both empty, the edges denoted by dashed lines in 1 are also absent.

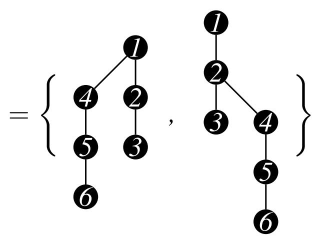
We denote by $(T_\chi, T_\rho) \uparrow^* \texttt{xbar}$ the simultaneous insertion of T_χ and T_ρ into the specifier and complementizer, respectively, of xbar.

$$\texttt{Xbar(3k)} = \bigcup_{i=1}^{k-1} (T_i, T_j) \uparrow^* \texttt{xbar}$$

for $T_i \in \texttt{Xbar(3i)}, T_j \in \texttt{Xbar(3(k - i))}$.

Example 3.2. $\texttt{Xbar(3)} = \{\texttt{xbar}\}$ *and*

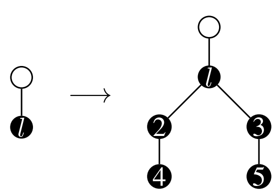
$$\texttt{Xbar(6)} = \{\texttt{xbar} \uparrow^*_{spec} \texttt{xbar}, \texttt{xbar} \uparrow^*_{comp} \texttt{xbar}\}$$



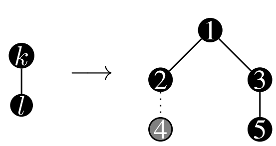
The natural interpretation of these are a single XP with a specifier of a single XP, and no complementizer, and a single XP containing a single complementizer of a single XP, and no specifier.

As presented by Kuminiak (1999), the asymmetric c–command extensional trees (i.e. those uniquely determined by their asymmetric c–command relation) can be generated by two types of insertion.

1. Add: Add two non–branching quasipendant vertices to any leaf.



2. Replace: For any nonbranching quasipendant node k, replace k with the five–noded structure below, with or without the left leaf node (4), i.e.



Then we can define the class ACC. Note we index families of trees from this class with number of *insertions*, as opposed to the number–of–node indexing we used previously, because each operation adds a variable number of nodes to the graph. We specify performing the Add (1) or Replace (2) operations at leaf node l (or in the case of the Replace operation (2), at l's quasipendant vertex, removing l altogether) as as $T_\alpha \uparrow^m_l T_\beta$, where T_0 is the empty tree, as

$$\texttt{ACC(k)} = \bigcup_{i=0}^{k} T_\alpha \uparrow^m_l T_\beta$$

for $T_\alpha \in \texttt{ACC(i)}, T_\beta \in \texttt{ACC(k - i)}, l \in L(T_j), m \in \{1, 2\}$.

Finally, we note a simple but important fact about the three defined classes.

Proposition 3.3. *For $n > 3$, the three classes are disjoint.*

When looking at the spectra of large trees from the three classes, this idea is useful in that it guarantees that the three tree sets are non–overlapping. So, it would be important for the spectra to reflect this fact.

3.2 Spectra of the three classes

It is known that the signless Laplacian spectrum and the Laplacian spectrum are identical for bipartite graphs (Abdian et al., 2018).[6] We additionally note that the magnitude of the eigenvectors of L_G and $\hat{L}_G$ are equal—the only difference stems from differences in sign in some of the entries of the vectors. Thus, we have the following proposition.

Proposition 3.4. *For any undirected rooted tree graph $T = (V, E)$ where $T \in$ BIN, XBAR, or ACC,*

$$\Lambda(L_G) = \Lambda(\hat{L}_G).$$

Further, the eigenvectors of L_G and $\hat{L}_G$ are identical modulo sign.

Now, we compare the spectra of the three classes of syntactic graphs by randomly generating three equal–sized sets (corresponding to the three syntactic classes) of high–dimensional[7] n–noded graphs, map them each to a matrix representation of dimension n, and graph their spectra in order of increasing value according to their percentile rank with the coordinates $(i \cdot \frac{100}{n}, \lambda_i)$.The trees are high–dimensional so the shape of the spectra is visible.

Each of the three graphs in Figure 2 demonstrate that each syntactic class has a unique spectrum distinct from the others: binary trees have the highest

[6]As such, the graph of $\hat{L}_G$ is omitted from this paper.

[7]We use "dimensional" to refer to the number of nodes in the graph, as the number of nodes in a graph corresponds to the number of dimensions of its matrix representations.

multiplicity of eigenvalues 0 and 1, followed by X–bar trees, while ACC trees are smoothest.

There are a couple of facts that help analyze the distribution of the spectrum. Let $l(T)$ be the number of leaves of a given tree, and $q(T)$ be the number of quasipendant vertices.

Corollary 3.5 (Nosal, 1970; Smith, 1970; Cvetkovic et al., 1980 p. 258.[8]). *The multiplicity of the eigenvalue* 0 *in the adjacency spectrum of a tree* T *is at least* $l(T) - q(T)$.

The same fact can be said of the eigenvalue 1 in the Laplacian spectrum:

Corollary 3.6 (Nosal, 1970; Smith, 1970; Cvetkovic et al., 1980 p. 258). *The multiplicity of the eigenvalue* 1 *in the Laplacian spectrum of a tree* T *is at least* $l(T) - q(T)$.

It turns out that the multiplicity of eigenvalue 1 in the Laplacian spectrum, $\mu_L(1)$, is a tight lower bound for all three classes. For the binary trees, experimentation with randomly generated trees points to the number $l(T) - q(T)$ as either *exactly* $\mu_L(1)$, or 1 or 2 less than $\mu_L(1)$.[9] The few trees T generated experimentally whose multiplicity of eigenvalue 1 in the Laplacian is *not* equal to $l(T) - q(T)$ share in common having a *maximal full binary tree* subgraph—that is, it is symmetric and every leaf at a given depth branches until the lowest level. This is stated in the following conjecture.

Conjecture 3.7. *For any rooted binary tree* $T = (V, E)$ *with* $|V| = n$, $\mu_{L_T}(1) = l(T) - q(T)$ *unless there is some subgraph* U *of* T *where, given the maximum possible* k *where* $n > 2^k - 1$, U *is a full binary tree of size* $2^k - 1$ *or* $2^{k-1} - 1$. *In this case,* $l(T) - q(T) + 1 \leq \mu_{L_T}(1) \leq l(T) - q(T) + 2$.

On the other hand, with respect to the XBAR and ACC trees, $l(T) = q(T)$ (every quasipendant vertex branches exactly once), and thus $l(T) - q(T) = 0$. Experimentation has shown that $\mu_{L_T}(1) = 1$ for all $T \in \mathtt{XBAR(n)} \cup \mathtt{ACC(m)}$, meaning that

$$\mu_{L_T}(1) = l(T) - q(T) + 1$$

for every tree in this class.

So for all three syntactic classes, the lower bound provided by Corollary 3.6 is extremely tight.

We can directly connect this to the syntactic constraints from which we defined these graphs. From

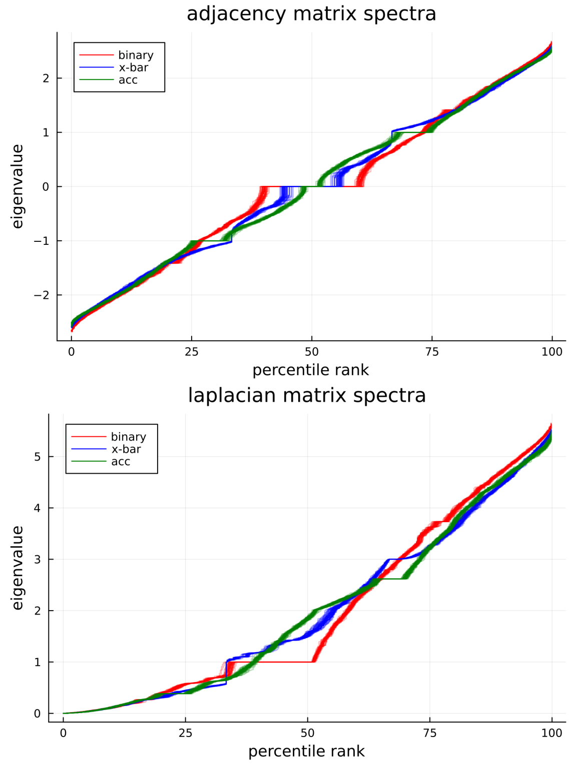


Figure 2: The adjacency and Laplacian spectra of a random sample of 50 trees from each of the three classes `Bin(501)`, `Xbar(501)`, `ACC(170)`.

the graphical/syntactic perspective, the multiplicity of the eigenvalue 1 in the Laplacian spectrum of these trees indicates an integral part of the syntactic classes' distinction: whether or not the *syntactic constraints* mandate binary branching at quasipendent vertices.

It is known that eigenvalues with high multiplicity within the spectrum of a graph can indicate the existence of a *motif*, i.e. repeated subgraph, in the graph (Banerjee and Jost, 2009). Recall that Corollary 3.6 linked the multiplicity of eigenvalue 1 in the Laplacian spectra of binary tree graphs to the number of quasipendant vertices branching into two leaves. We can then reframe the discussion around Corollary 3.6 as $\mu_L(1)$ in binary tree graph spectra being potentially indicative of the motif `bin_base` at the leaves of the binary trees.

We now move to discussing the general shape of the eigenvalue graphs and explore potential reasons the spectral graphs preserve class distinctions.

He et al. (2000) observe that the Laplacian spectrum of $k-$ary trees resemble a Cantor step function. The binary branching trees are $3-$ary trees for all non–leaf nodes except the central/root node

[8] Useful discussion provided by Dadedzi (2018).

[9] This is significant given that these trees have over 500 nodes (and subsequently, ove 500 eigenvalues), and yet the multiplicity of eigenvalue 1 is so close to exactly the quantity $l(T) - q(T)$.

branching $2 = k-1$ times, so this substantiates the observation that the Laplacian spectrum of the class `Bin(501)` resembles the Cantor step function.

The Cauchy Interlacing Theorem describes properties of spectrum of submatrices of matrices in relation to the matrix, and can be used to understand properties of the spectrum of subgraphs of graphs as a function of the graph.

Theorem 3.8 (Cauchy Interlacing Theorem, Haemers, 1995)**.** *Let A be an $n \times n$ hermitian matrix (i.e. $A = \overline{A}^T$: it is equal to its conjugate transpose, which is true for any symmetric matrix over the field $\mathbb{R}$) with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$, and B be an $m \times m$ submatrix obtained from A by deleting $n-m$ rows and columns of the same index. Suppose B has eigenvalues $\beta_1 \geq \beta_2 \geq \cdots \geq \beta_m$, then*

$$\lambda_i \geq \beta_i \geq \lambda_{n-m+i}, \text{ for } i = \{1, 2, \ldots, m\}.$$

In other words, the eigenvalues of any submatrix of a matrix (where the submatrix is formed by deleting corresponding rows and columns) are interleaved with the eigenvalues of the matrix. Thus, we can generalize this to adjacency matrices of graphs. [10]

Proposition 3.9. *Let $G = (V, E)$ be a graph with $|V| = n$, adjacency matrix A_G and corresponding spectra $\Lambda(A_G) = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$. Let $H = (V', E')$ be a subgraph of G with $|V'| = m$, adjacency A_H and spectrum $\Lambda(A_H) = \{\mu_1, \mu_2, \ldots, \mu_m\}$. Then*

$$\lambda_i \geq \mu_i \geq \lambda_{n-m+i}, \text{ for } i = \{1, 2, \ldots, m\}.$$

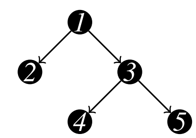
So the eigenvalues of the adjacency matrix of any subgraph of a graph should be interleaved with the eigenvalues of the adjacency matrix of that graph. Recalling that these trees are built off of recursively combining smaller subtrees, this helps give intuition towards the consistent distinctness of the spectra as you increase the size of the tree—given a large tree, the eigenvalues of a subtree of it are distributed amongst the eigenvalues of the tree, preserving the shape, so inductively this is true as you decrease the size of the tree.

[10] Laplacian matrices are more difficult, as the Laplacian of a subgraph of a graph is not immediately a submatrix of the Laplacian of the graph: deleting rows and columns results in a decrease of the degrees reported along the diagonal.

4 Spectral properties of directed syntactic structures

We now consider what happens when we incorporate more traditional assumptions concerning syntactic structure and represent syntactic structures as directed graphs. As above, we explore the spectra of the three classes `BIN`, `XBAR`, and `ACC` as digraphs. Consider the following tree in `Bin(5)` from example 2.4 but with directed edges.

Example 4.1. *The directed rooted binary tree $G = (V, E)$ where $V = \{1, 2, 3, 4, 5\}, E = \{\{1, 2\}, \{1, 3\}, \{3, 4\}, \{3, 5\}\}$:*



First, as in Example 2.4, we calculate the Laplacian of the above digraph.

Example 4.2. *Let $G = (V, E)$ be given as in example 4.1.*[11] *Then*

$$L_G = D_G - A_G = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Observe that both A_G and L_G are *upper triangular* matrices–that is, all the entries below the diagonal are 0. In fact, A_G is *strictly upper-triangular*, as its diagonal too is all 0.[12] We state the following well–known fact about upper triangular matrices.

Proposition 4.3. *Let $M \in \mathbb{R}^n \times \mathbb{R}^n$ be an upper triangular matrix. Then its eigenvalues are the diagonal entries of the matrix.*

The following is derived from Proposition 4.3.[13]

Proposition 4.4. *Let M be an $n \times n$ strictly upper triangular matrix. Then it has one distinct eigenvalue 0 with $\mu_M(0) = n$.*

[11] Note that we say a directed edge $\{v_i, v_j\}$ exists if there is an edge *from v_i to v_j*, and not vice–versa, and recall that the degrees of the nodes here are calculated by using the outdegree.

[12] These graphs are acyclic and thus loopless, so there is never an edge $\{v_i, v_i\}$ from a vertex to itself.

[13] Note that Proposition 4.4 can also be derived by the fact that a strictly upper triangular matrix is *nilpotent*, i.e. for a nilpotent $n \times n$ matrix N there exists a $k \in \mathbb{N}$ such that $N^k = \mathbf{0}$, the $n \times n$ zero matrix. It is a well–known fact that all nilpotent matrices have spectra containing one unique eigenvalue, 0 (with multiplicity equal to the dimension of the matrix).

So we can calculate the eigenvalues of these matrices simply by extracting their diagonal entries. Thus, $\Lambda(A_G) = \{0^5\}$ and $\Lambda(L_G) = \{0^3, 2^2\}$.

This leads us to the following theorem.[14]

Theorem 4.5. *Given a rooted tree digraph $T = (V, E)$ where $|V| = n$, the spectrum of its adjacency matrix A_T is $\{0^n\}$ and the spectrum of its Laplacian matrix L_T is equal to the outdegree of each of its nodes (in particular, $\mu_{L_T}(0) = l(T)$). That is,*

$$\Lambda(A_T) = \{0^n\} \text{ and } \Lambda(L_T) = \mathcal{D}(T).$$

So for any rooted tree digraph, we need only track of the outdegree of each node in order to know the spectrum of its Laplacian. Then we have the following.

Theorem 4.6. *Let $T = (V, E)$ be a directed binary tree with $|V| = n$. Then the spectrum of its Laplacian is $\Lambda(L_T) = \{0^{\frac{n+1}{2}}, 2^{\frac{n-1}{2}}\}$.*

Next, we state analogous theorems for XBAR and ACC. The proofs are left to the reader—factors to consider are included in the proof of the previous theorem.

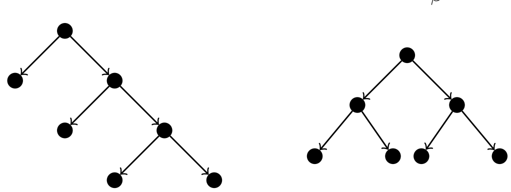
Theorem 4.7. *Let $G = (V, E)$ be an X–bar tree with $|V| = n$. Then the spectrum of its Laplacian is $\Lambda(L_G) = \{0^{\frac{n}{3}}, 1^{\frac{n}{3}+1}, 2^{\frac{n}{3}-1}\}$.*

Theorem 4.8. *Let $T = (V, E) \in \texttt{ACC(m)}$ with $|V| = n$. Then $\Lambda(L_T) = \{0^{m+1}, 1^{n-(2m+1)}, 2^m\}$.*

Given the important role that the spectrum of a graph plays in determining what class it falls in, we might ask the question of whether the spectrum uniquely determines a specific graph G modulo vertex relabeling. For the case of the spectrum of the Laplacian of a directed tree (where the eigenvalues are the degrees) the answer is no, as the following example illustrates.

Example 4.9. *Consider the following graphs.*

Tree T_α: *Tree T_β:*



$T_\alpha, T_\beta \in \texttt{Bin(7)}, \mathcal{D}(T_\alpha) = \mathcal{D}(T_\beta)$*, but* $T_\alpha \neq T_\beta$.

On the other hand, does the spectra of the Laplacian of these families of graphs, i.e. the outdegrees of the nodes, uniquely determine whether a tree belongs to a specific syntactic class? The answer is *yes* with respect to any family of binary trees—in fact, in general for any n–ary trees (where each node has outdegree of either n or 0).

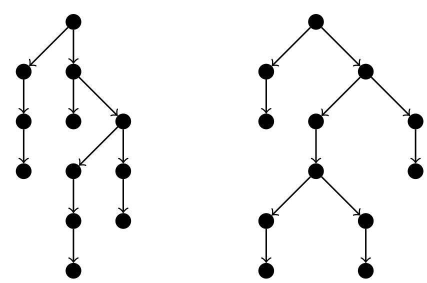
Proposition 4.10. *Let $\mathcal{T}_n$ be a family of $n-ary$ trees, where every non–leaf has an outdegree of n. For total number of nodes N in the tree T, $\Lambda(T) = \{0^{\frac{N+1}{n}}, n^{\frac{N-1}{n}}\}$ if and only if $T \in \mathcal{T}_n$.*

This does not hold for any class non–n–ary trees, i.e. any tree with more than two distinct outdegrees. Whenever more than one non-zero branching factor is allowed, spectral uniqueness is lost.[15]

For instance, XBAR and ACC, two examples of tree families with three distinct eigenvalues/outdegrees (0, 1 and 2), are not uniquely defined by their outdegrees/spectra.

Example 4.11. *Consider the following graphs.*

Tree T_α: *Tree T_β:*



It is clear that $T_\alpha \in \texttt{XBAR(12)}, T_\beta \in \texttt{ACC(3)}$, *while* $T_\beta \notin \texttt{XBAR(12)}, T_\alpha \notin \texttt{ACC(3)}$. *However,* $\Lambda(T_\alpha) = \mathcal{D}(T_\alpha) = \mathcal{D}(T_\beta) = \Lambda(T_\beta)$. *So although T_α and T_β are members of distinct syntactic classes, their Laplacian spectra are identical.*

Though the spectra of a rooted tree digraph does not definitively classify it to a particular syntactic class (besides n–ary trees), we can say something interesting about spectra of graphs in tree languages generated by (directed) regular tree grammars.

Definition 4.12. A regular tree grammar *is a tuple $G = (N, \Sigma, R, S)$. N is a finite set of nonterminals and Σ is a ranked alphabet of terminals such that $\Sigma \cap N =$, $S \in N$ is the initial nonterminal, and R is a finite set of rules of the form $A \rightarrow t$ with $A \in N$ and $t \in T_\Sigma(N)$. The tree language generated by G, denoted $L(G)$, is defined as $L(H)$ where H is the context–free grammar $(N, \Sigma \cup \{[,]\}, R, S)$.*

---

[14] All proofs are contained in the appendix.

[15] Given any tree with at least two nodes with distinct, nonzero branching numbers, you can swap their location (along with the subtrees that they each are the root of) in the tree and come up with a new, distinct tree from the original with the same spectrum.

Assuming that the graphs $t \in T_\Sigma(N)$ comprising the right side of the rules are *directed*, we can state the following theorem.

Theorem 4.13. *Suppose $G = (N, \Sigma, R, S)$ is a (directed) regular tree grammar. Define the set of outdegrees of any rule $A \to t \in R$ for $t \in T_\Sigma(N)$, $\mathcal{OD}(A)$, as the set of outdegrees of the graph structure t excluding any nodes labeled by nonterminals. Then the spectrum of any tree $T \in L(G)$ generated by G is the union of the spectra of the rules used to generate T, i.e. the union of the set of outdegrees of each rule. So, for $\mathcal{R}(T)$ as the set of rules applied to generate T,*

$$\Lambda(L_T) = \bigcup_{R \in \mathcal{R}(T)} \mathcal{OD}(R)$$

In other words, one can directly compute the spectrum of a tree T generated by a directed regular tree grammar by simply taking the union of the outdegrees of the rules used to generate T (excluding any nonterminals, which end up being replaced by nodes of graphs of other rules).

Thus far, our discussion has been focused on the eigenvalues of a matrix representation of a graph. Included in the set of spectral properties of a matrix are its eigenvectors. We now briefly consider the eigenvectors of matrix representations of the syntactic classes we have concerned ourselves with.

With respect to adjacency, Laplacian and signless Laplacian matrix representations, the eigenvectors of all three undirected graph classes all contain both positive and negative signed entries. In comparison, for the directed versions of all of these tree graphs, there is an eigenvector for each eigenvalue whose non–zero entries are all the same sign.[16]

Theorem 4.14. *For any directed rooted tree graph $T = (V, E)$ where $T \in$ `BIN`, `XBAR`, or `ACC`, for every eigenvalue λ of L_T there exists an eigenvector ψ such that every entry of ψ has the same sign.*

5 Conclusion

This paper presents a novel way to explore differences in syntactic structure. We give the first results connecting properties of spectra to syntactically relevant classes of trees. The case study in this paper considers three specific classes of tree structures and shows structural syntactic differences are perceivable at the spectral level, with a variety of properties of these trees (which class they belong to, whether they are directed or undirected, etc.) reflected in the spectra and eigenvectors.

At present, we have only considered a limited set of syntactic classes. This leaves a wide variety of other potentially syntactically relevant graphs, including those that limit leftward branching, or non-tree structure graphs allowing multidominance. Our results leave open further exploration of other classes of trees that are uniquely characterized by the spectra of directed or undirected graphs. We leave this for future work.

One especially exciting result in the current work concerns the degree to which spectra of a class can be derived from a regular tree grammar that generates the class. Just as the Parikh mappings of strings can be derived from the underlying string CFG, so too can the Laplacian spectra of directed syntactic tree graphs be derived from the underlying graph rules. We leave it as an open question to look at richer classes of tree grammars and alternative matrix representations.

Our motivation in this work is to identify novel mathematical tools with which we can look beneath surface representations of linguistic structures and explore more fundamental features of their linguistic essence. The current work has utilized spectral graph theory as one mathematical tool to do just this, examining the reflection of certain syntactic features and properties in the spectra. This paper demonstrates SGT is a way to peel back the surface combinatorial graphical structure we see, and attempt to understand deeper, more inherent features of the syntactic structures. The goal of future work would be to take this one step further—not only understanding the ways in which spectra can reflect syntactically relevant properties, but further developing the spectral studies of these graphs in order to use the spectra to identify fundamental properties about syntactic structure that are inaccessible or hidden from view based on the surface combinatorial structure of these graphs.

Acknowledgements

The authors would like to thank Daniel Spielman for his invaluable guidance and insights.

References

Ali Zeydi Abdian, Afshin Behmaram, and Gholam Hossein Fath-Tabar. 2018. Graphs determined by sign-

[16] As eigenvectors define a linear space, each eigenvector defines a set of all multiples of that eigenvector by all real numbers. So this is equivalent to saying an eigenvector's entries do not change signs.

less Laplacian spectra. *AKCE International Journal of Graphs and Combinatorics*.

Anirban Banerjee and Jürgen Jost. 2009. Graph spectra as a systematic tool in computational biology. *Discrete Applied Mathematics*, 157(10):2425–2431.

J. A. Bondy and U. S. R. Murty. 1976. *Graph Theory with Applications*. Elsevier, New York.

Koel Chowdhury, Cristina España-Bonet, and Josef Genabith. 2021. Tracing Source Language Interference in Translation with Graph-Isomorphism Measures. In *Proceedings of Recent Advances in Natural Language Processing*, pages 375–385.

Fan Chung. 1997. *Spectral Graph Theory*. American Mathematical Society.

Dragos M Cvetkovic, Michael Doob, and Horst Sachs. 1980. *Spectra of Graphs : Theory and Application*. Academic Press.

Kenneth Dadedzi. 2018. *Analysis of tree spectra*. Ph.D. thesis, Stellenbosch University.

Robert Frank and Fero Kuminiak. 2000. Primitive Asymmetric C-Command Derives X-Theory. In *Proceedings of the 30th Annual Meeting of the North East Linguistics Society*, volume 1, pages 203–217, University of Massachusetts, Amherst.

Robert Frank and K. Vijay-Shanker. 2001. Primitive C-Command. *Syntax*, 4:164–204.

Chris Godsil and Gordon Royle. 2001. *Algebraic Graph Theory*, volume 207 of *Graduate Texts in Mathematics*. Springer.

Willem H. Haemers. 1995. Interlacing eigenvalues and graphs. *Linear Algebra and its Applications*, 226–228:593–616.

Willem H. Haemers and Edward Spence. 2004. Enumeration of cospectral graphs. *European Journal of Combinatorics*, 25(2):199–211.

Li He, Xiangwei Liu, and Gilbert Strang. 2000. Laplacian Eigenvalues of Growing Trees. In *Conf. on Math. Theory of Networks and Systems*.

A. J. Hoffman. 1970. On Eigenvalues and Colorings of Graphs. In *Graph Theory and Its Applications*, volume 175, pages 79–92. Academic Press, New York.

David P. Jacobs, Elismar R. Oliveira, and Vilmar Trevisan. 2021. Most Laplacian eigenvalues of a tree are small. *Journal of Combinatorial Theory, Series B*, 146:1–33.

Richard Kayne. 1994. *The Antisymmetry of Syntax*. MIT Press, Cambridge, MA.

Fero Kuminiak. 1999. Formal properties of asymmetric c-command in tree structures. Manuscript, Johns Hopkins University, Baltimore, MD.

Brendan McKay. 1977. On the spectral characterisation of trees. *Ars Combin*, 3:219–232.

Eva Nosal. 1970. Eigenvalues of graphs. Master's thesis, University of Calgary.

Andrew Ortegaray, Robert Berwick, and Matilde Marcolli. 2021. Heat Kernel Analysis of Syntactic Structures. *Mathematics in Computer Science*, 15.

John H Smith. 1970. Some properties of the spectrum of a graph. *Combinatorial Structures and their applications*, pages 403–406.

Daniel A. Spielman. 2019. Spectral and Algebraic Graph Theory. Manuscript.

Edwin R. van Dam and Willem H. Haemers. 2003. Which graphs are determined by their spectrum? *Linear Algebra and its Applications*, 373:241–272.

Wei Wang and Cheng-Xian Xu. 2006. On the spectral characterization of T-shape trees. *Linear Algebra and its Applications*, 414(2):492–501.

Herbert S. Wilf. 1967. The Eigenvalues of a Graph and Its Chromatic Number. *Journal of the London Mathematical Society*, s1-42(1):330–332.

A Appendix: Proofs of Theorems

Theorem 4.5. *Given a rooted tree digraph $T = (V, E)$ where $|V| = n$, the spectrum of its adjacency matrix A_T is $\{0^n\}$ and the spectrum of its Laplacian matrix L_T is equal to the outdegree of each of its nodes (in particular, $\mu_{L_T}(0) = l(T)$). That is,*

$$\Lambda(A_T) = \{0^n\} \text{ and } \Lambda(L_T) = \mathcal{D}(T).$$

Proof of Theorem 4.5. Suppose $T = (V, E)$ is a rooted tree digraph with $|V| = n$. There exists an enumeration of the vertices such that $i = e(v_i) \leq e(v_j) = j$ for natural numbers i, j iff v_i is the parent of v_j. Then, for all directed edges $\{v_i, v_j\}, i \leq j$. As (i, j) corresponds to the indices of the adjacency matrix A_T of G, this yields a strictly upper-triangular adjacency matrix A_T: all edges from i to j will set $a_{ij} = 1$, above the diagonal, and $a_{ji} = 0$, below the diagonal.[17]

Given that the adjacency matrix is strictly upper triangular, its spectrum is

$$\Lambda(A_T) = \{0^n\}$$

.

The diagonal entries of the Laplacian are determined by D_G, which correspond to the outdegree

[17]As there are no self-loops in G, meaning no edges $\{v_i, v_i\}$ for all $i \leq n$, $a_{ii} = 0$ for all $i \leq n$.

of each of the n nodes. L_G is upper triangular: $L_G = D_G - A_G$.[18] Thus, by Proposition 4.3, the eigenvalues of L_G are equal to the diagonal D_G, which is the outdegree of each of the n nodes.

□

Theorem 4.6. *Let $T = (V, E)$ be a directed binary tree with $|V| = n$. Then the spectrum of its Laplacian is $\Lambda(L_T) = \{0^{\frac{n+1}{2}}, 2^{\frac{n-1}{2}}\}$.*

Proof of Theorem 4.6. We can think about a given binary tree as a construction starting with the smallest possible binary tree, the 3–noded binary tree, and then recursively substituting that same binary tree with 3 vertices to the leaves of the first tree. Any binary tree with $n = 2k + 1$ vertices can be constructed by inserting $(n - 3)/2$ copies of this base binary tree, root–to–leaf (i.e. the root of the tree being inserted inserts into one of the current leaves) including the initial starting tree, or $(n - 1)/2$ copies of the base binary tree total.

We prove this by induction. For $k = 1$ with $n = 3$ we have T_3 = `bin_base`, which has one outdegree of 2 (the root/branching node) and two outdegrees of 0 (the leaves). So $\Lambda(T_3) = \{0^2, 2^2\} = \{0^{\frac{3+1}{2}}, 2^{\frac{3-1}{2}}\} = \{0^{\frac{n+1}{2}}, 2^{\frac{n-1}{2}}\}$.

Suppose we have performed $k-2$ insertions into this tree T_{2k-1} (for a total of $k - 1$ copies of the binary tree). At each insertion of a new base binary tree T_3 to one of the leaves of the current binary tree, two additional nodes are gained. The first tree T_3 begins with 3 nodes, and each subsequent insertion of a new copy of T_3 yields two more nodes (inserting root-to-leaf does not add a count to the node with the root node, but it does with the two new leaves). So $n = |V| = 2(k - 1) + 1 = 2k - 1$. Assume $\Lambda(T_{2k-1}) = \{0^{\frac{n+1}{2}}, 2^{\frac{n-1}{2}}\} = \{0^{\frac{2k-1+1}{2}}, 2^{\frac{2k-1-1}{2}}\} = \{0^k, 2^{k-1}\}$.

To construct T_k, we insert a new copy of the base tree T_3 to one of the leaves of T_{2k-1}. This insertion forces that leaf to branch, turning its outdegree from 0 to 2, and then adds two new outdegrees of 0, the two new leaves, resulting in a net gain of one leaf. We have gained one node with outdegree 2, the formerly-leaf-turned-binary-branch. Thus, the insertion of a copy of T_3 into T_{2k-1} has a net degree gain of one 2-degree and one 0-degree. Note the total number of nodes here is two more than $2k - 1$, $2k + 1$. Thus

$$\Lambda(T_{2k+1}) = \{0^{k+1}, 2^{k-1+1}\} = \{0^{k+1}, 2^k\}$$

$$= \{0^{\frac{2k+2}{2}}, 2^{\frac{2k}{2}}\} = \{0^{\frac{(2k+1)+1}{2}}, 2^{\frac{(2k+1)-1}{2}}\}$$

$$= \{0^{\frac{n+1}{2}}, 2^{\frac{n-1}{2}}\}.$$

The proof of the class `XBAR` is identical in structure: we only need observe that substituting `xbar` to the specifier or complementizer positions adds three nodes to the graph (as we create a new edge) and increases the node counts by one new outdegree 2, one outdegree of 1 and one outdegree of 0.

To prove the case of `ACC`, we are forced to consider the multiplicity of eigenvalues as a function of both the number of insertions and the number of nodes. This is due to the variability in the number of nodes gained through each different operation. Operation (1) above creating the five–noded structure results in a net gain of one outdegree 0, two outdegrees of 1, and one outdegree of 2. Operation (2), replacing the quasipendant node and its leaf with the four– or five–noded structure results in a net gain of one outdegree 0, one outdegree of 1, and one outdegree of 2 or one outdegree 0 and one outdegree 2. This optionality of which structure you insert, as well as the ambiguity of indexing this class by number of insertions as opposed to node number (for this very reason) results in the variation of multiplicity of eigenvalue 1.

□

Proposition 4.10. *Let $\mathcal{T}_n$ be a family of n – ary trees, where every non–leaf has an outdegree of n. For total number of nodes N in the tree T, $\Lambda(T) = \{0^{\frac{N+1}{n}}, n^{\frac{N-1}{n}}\}$ if and only if $T \in \mathcal{T}_n$.*

Proof of Proposition 4.10. Given an $n-$ary rooted directed tree $T = (V, E)$ with $|V| = n$, any non–leaf branches exactly n times by definition. So every node either has n children or is a leaf. Thus by Theorem 4.5, $\Lambda(\mathcal{T}_n) = \{0^{\frac{N+1}{n}}, n^{\frac{N-1}{n}}\}$.

On the other hand, suppose we are given a rooted tree digraph $T = (V, E)$ with spectrum $\Lambda(T) = \{0^{\frac{N+1}{n}}, n^{\frac{N-1}{n}}\}$ for N nodes and $n \in \mathbb{N}$. Since any rooted tree digraph has eigenvalues corresponding directly to its outdegrees means (by Theorem 4.5) T must have $\frac{N+1}{n}$ leaves and $\frac{N-1}{n}$ nodes with outdegree n. Thus $T \in \mathcal{T}_n$. □

[18] Both D_G and A_G are upper triangular, and the sum/difference of two upper triangular matrices is upper triangular.

Theorem 4.14. *For any directed rooted tree graph $T = (V, E)$ where $T \in$ BIN, XBAR, or ACC, for every eigenvalue λ of L_T there exists an eigenvector ψ such that every entry of ψ has the same sign.*

Proof of Theorem 4.14. Recall that an eigenvector of any matrix is, by definition 2.6, nonzero. We provide the intuition behind the class BIN, as the other two are similar.

Let $T = (V, E)$ where $T \in \mathtt{BIN(n)}$ is a directed rooted tree graph where $n = 2k + 1$ for some $k \in \mathbb{N}$. We know the Laplacian L_T is upper triangular. It will have $k + 1$ rows/columns of zeros, corresponding to each of the $k + 1$ leaves.As each binary tree with $n = 2k + 1$ nodes has k binary–branching nodes, L_T has k rows/columns with 2 on the diagonal (i, i) and two entries of -1, at (i, j_1) and (i, j_2), where $j_1, j_2 > i$. For any (nonzero) eigenvector

$$\psi = [\psi_1, \psi_2, \ldots, \psi_n]^T$$

where

$$L_T\psi = \lambda\psi,$$

the zero rows of L_T indexed by integers $l_1, l_2, \ldots, l_{k+1}$ give rise to $k + 1$ equations of the form

$$0 = \lambda\psi_{l_i}.$$

On the other hand, the k nonzero rows give rise to equations of the form

$$2\psi_i - \psi_{i+c} - \psi_{i+c+d} = \lambda\psi_i$$

for nonzero numbers $c, d \in \mathbb{N}$.

It is useful in building intuition to connect the occurrences of each $\psi_i \in \psi$ in the system of equations given by the equation

$$L_T\psi = \lambda\psi$$

to the behavior of the node in the graph enumerated with label i.

Let $\mathcal{L}$ be the set of integers corresponding to the labels of the leaves of the tree. For all $l \in \mathcal{L}$, ψ_l exists as a variable with coefficient -1 in exactly one equation of the second form, that is,

$$2\psi_i - \psi_j - \psi_l = \lambda\psi_i,$$

as every leaf node necessarily is connected to one binary–branching node, as well as in one equation of the first form,

$$0 = \lambda\psi_l.$$

Every non–leaf, non–root node with label m exists in two equations, both of the second form: one with coefficient 2, that is,

$$2\psi_m - \psi_i - \psi_j = \lambda\psi_m,$$

and one with the coefficient -1,

$$2\psi_i - \psi_j - \psi_m = \lambda\psi_i.$$

The root node r exists in exactly one equation, the equation of the second form, with coefficient 2:

$$2\psi_r - \psi_i - \psi_j = \lambda\psi_r.$$

In what follows, we assume the matrix has been permuted into the form of the first k rows being the nonzero rows, that is, 2 in the diagonal followed later in the row with two entries of -1 (i.e. the rows corresponding to the binary–branching nodes) and then $k + 1$ rows of zeros. Schematically, the matrix is of the form:

$$A = \begin{bmatrix} 2 & a_{12} & & \cdots & & a_{1n} \\ 0 & \ddots & & \cdots & & \vdots \\ & & 2 & a_{k,(k+1)} & \cdots & a_{k,n} \\ \vdots & & \ddots & 0 & \cdots & 0 \\ & & & & \ddots & \vdots \\ 0 & & \cdots & & 0 & 0 \end{bmatrix}$$

A is not only an upper triangular matrix, but also the last $k + 1$ rows is an all–zeros rectangle of dimension $(k + 1) \times n$.

There are two categories of eigenvectors, those which pertain to eigenvalue 2 and those pertaining to eigenvalue 0.

Case 1: Suppose $\lambda = 2$.

Then there are $k + 1$ equations of the form

$$0 = 2\psi_{l_i},$$

yielding

$$\psi_{l_i} = 0.$$

By the form of the vector above, the final nonzero row of the matrix L_T, row k, with a 2 in position (k, k), gives the equation

$$2\psi_k - \psi_{j_1} - \psi_{j_2} = 2\psi_k$$

will subsequently have

$$\psi_{j_1} = \psi_{j_2} = 0.$$

Intuitively, we can understand this row as representing a binary–branching node in the tree which branches into two non–branching leaf nodes. There is, necessarily, at least one of these existing in any given tree. Then this results in

$$2\psi_k - 0 - 0 = 2\psi_k,$$

yielding ψ_k being a free variable (where k is not the label of the root node, assuming $k > 1$, that is, $2k + 1 > 3$).

It is necessary for $\psi_k = 0$, and subsequently for $\psi_j = 0$, in the second equation containing ψ_k,

$$2\psi_h - \psi_k - \psi_j = 2\psi_h.$$

For the leaf nodes, then, it is easy to see that the fact that for every $l \in \mathcal{L}$, $\psi_l = 0$ results in free variables for the k binary–branching nodes, which all exist in a second equation with coefficient -1, except for the root node. The reader can verify that then for every ψ_i in at least two equations, that is, every entry of the eigenvector except for the first (which correlates to the root node and is only in equations of the second form with coefficient 2), $\psi_i = 0$. As eigenvectors must be nonzero, then, this first entry must assume a nonzero value. So every eigenvector of eigenvalue 2 must have eigenvector of the form $c \cdot e_1$ for the first basis vector e_1 and $c \in \mathbb{R}$. As 2 has multiplicity k, there are k eigenvectors of this form.

Case 2: Suppose $\lambda = 0$.

Then equations of the first form are

$$0 = 0\psi_l$$

and the second form are

$$2\psi_k - \psi_{j_1} - \psi_{j_2} = 0.$$

This means that every $l \in \mathcal{L}$, ψ_l becomes a free variable. The reader can verify that in order for a given eigenvector to have all entries of either the same sign or 0, exactly one ψ_l can be nonzero. Not only this, but for $\psi_1 = c$ for $c \in \mathbb{R}$ and root with label 1, for each node i on the path from root to leaf l with nonzero ψ_l,

$$\psi_i = 2^m c$$

for m being the length of the path from root to i. This comes from the equations of the second form

$$2\psi_k - \psi_{j_1} - \psi_{j_2} = 0$$

as, without loss of generality, if $j_1, j_2 \in \mathcal{L}$ and $\psi_{j_1} = 0$,[19] then

$$2\psi_k = \psi_{j_2}.$$

Each nonzero entry of a given eigenvector thus corresponds to the labels of nodes which form a directed path from root to leaf for a chosen nonzero ψ_l corresponding to label of a leaf l.

Therefore, each of the $k + 1$ eigenvectors of eigenvalue 0 correspond to each possible nonzero choice of ψ_l for $l \in \mathcal{L}$, and each of these eigenvectors have nonzero entries ψ_i for every label i on the path from the root to l for the given nonzero ψ_l.

In the case where $T \in \texttt{XBAR(n)}$ or $T \in \texttt{ACC(k)}$, note that we have the additional row/column case where there is 1 in the diagonal and thus nonzero, non–two rows are of the form $\psi_i - \psi_{i+c} = 1\psi_i$, yielding ψ_i as a free variable and $\psi_{i+c} = 0$. $\square$

[19]The case where both $\psi_{j_1} = \psi_{j_2} = 0$ results in $\psi_k = 0$, which percolates into the equation where ψ_k has coefficient -1 and the same scenario is repeated.