Kirchoff's Matrix Tree Theorem

Jordan Wheeler
Nebraska Wesleyan University
jwheeler@nebrwesleyan.edu

December 13, 2017

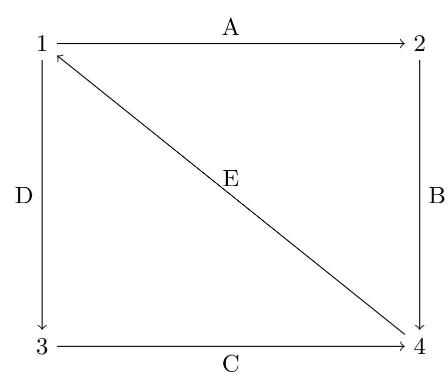
Abstract

The Matrix Tree Theorem, often referred to as the Kirchhoff?s Theorem, is a theorem in graph theory that finds the number of spanning trees in an undirected graph. We will explore the basic definitions that are required to setup the proof and then see how Gustav Kirchhoff used linear algebra in polynomial time to help provide a proof for this theorem in combinatorics. The proof will show how one can utilize different fields of mathematics, in this case linear algebra, to assist in prove theorems within combinatorics and graph theory. [1]

1 Preliminaries

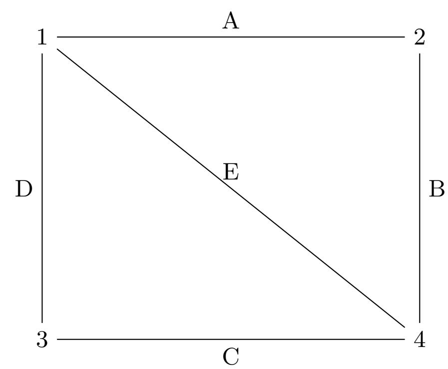
1.1 Directed and Undirected Graphs

The interesting thing about Kirchoff's Matrix Tree Theorem is the application of Linear Algebra to Combinatorics and Graph Theory. Before diving deep into the proof of the Matrix Tree Theorem, there are a few preliminaries we need to go through. First, it will be important to know the difference between directed graphs and undirected graphs. A **directed graph** is a graph that each edge is assigned a direction.



If you notice here, there are arrows on each edge, pointing from one vertex to another. It is also important to note that the **head** of a directed edge is the vertex which the arrow is pointing to and the **tail** of a directed edge is the vertex which the arrow is coming from. For further clarification, looking at the graph above, for edge A, the head is vertex *2* and the tail is vertex *1*.

The other type of graphs we will need to know is undirected graphs. An **undirected graph** is a graph which no edges are assigned a direction. This means that for an undirected graph, there are no heads and no tails to each edge.



As you can see above, this graph is the same graph as the directed one, which was previously shown. The only difference is there are no arrows points from one vertex to another.

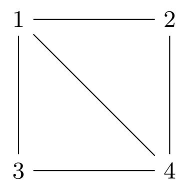
1.2 Trees and Spanning Trees

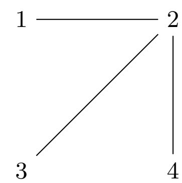
Now that we have gone over the different types of graphs and what they look like, we need to go over trees and spanning trees, which is what we will end up counting from the Matrix Tree Theorem. Trees are going to be a subcollection of edges from a graph that hit all the vertices of the graph. More formally:

Definition 1. *A connected simple graph G that satisfies either (and therefore both since they are equivalent) of the following conditions is called a* ***tree****. (Note: all trees on n vertices have exactly $n - 1$ edges)*

- *G is a minimally connected, meaning if any edge of G is removed, then the obtained graph, G', will not be connected.*
- *G does not contain a cycle. [1]*

It is important to note that a tree with n vertices has to have $n-1$ edges. Below you will see two graphs, on the left, you have a graph that is not a tree. This is because the graph has 4 vertices and 5 edges. On the right, you have a graph that is a tree. This is because the graph has 4 vertices ($n = 4$) and the graph has 3 edges ($n - 1 = 3$), which means its minimally connected and there are no cycles.

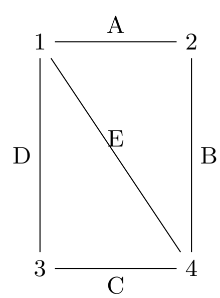


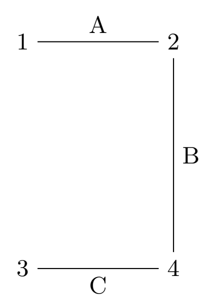


A sub group of trees are spanning trees. Spanning trees are actually what we will be counting with the Matrix Tree Theorem. A spanning tree is formally defined by:

Definition 2. *If G is a connected graph, we say that T is a **spanning tree** of G, if G and T have the same vertex set and each edge of T is also an edge of G. [1]*

Below on the left we have a graph, let's say G, with vertices $\{1, 2, 3, 4\} \in G$ and the edges $\{A, B, C, D, E\} \in G$. Below on the right we have a spanning tree, let's say T, of G. We know that T is a spanning tree of G, because it contains the same vertices as G and edges $\{A, B, C\} \in T$ are also in G. Also, T does not have any cycles and it is minimally connected.



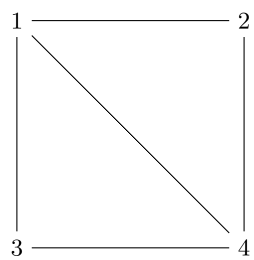


1.3 Adjacency Matrix

Now that we have gone over the basics needed for the Graph Theory part of the Matrix Tree Theorem, we need to cover the Linear Algebra part. We can create matrices for each graph that we are given based on its vertices and edges. These matrices are important because they tell us information about the graphs and will help us count the spanning trees of each graph, which is our end goal. For undirected graphs, we will need to know what the adjacency matrix is and how we can find it for any given graph. The formal definition of the adjacency matrix follows:

Definition 3. *Let G be an undirected graph on n labeled vertices and define an $n \times n$ matrix $A = A_G$ by setting $A_{i,j}$ equal to the number of* *edges between vertices i and j. Then A is called the **adjacency matrix** of G.*

Below you will notice there is a graph and next to it is its adjacency matrix.



$$\begin{array}{c c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

Each row and column of the adjacency matrix is labeled with a vertex from the graph. You will notice along the diagonal there will always be 0's because there are no loops connecting a vertex to itself. You will then see that if two vertices are connected by an edge, then it receives an 1 in the corresponding location of the matrix.

1.4 Incidence Matrix

The other types of matrix that will tell us information about a graph is the incidence matrix. These matrices are for directed graphs and will tell us even more about the graph than the adjacency matrix. These matrices will tell us not only what vertices are connected, but since its for directed graphs, it will tell us what vertices are the head and tail of each edge. The formal definition is as follows:

Definition 4. *Let G be a directed graph. Let $\{v_1, v_2, ..., v_n\}$ denote the vertices of G and let $\{e_1, e_2, ..., e_m\}$ denote the edges of G. Then the **incidence matrix** of G is the $n \times m$ matrix A defined by:*

- $a_{i,j} = 1$ *if v_i is the head of e_j*
- $a_{i,j} = -1$ *if v_i is the tail of e_j*
- $a_{i,j} = 0$ *otherwise [1]*

Below you will see the graph that we have been using for our examples and next to it is a spanning tree of that graph.

$$
\begin{tikzcd}
1 \arrow[r, "A"] \arrow[d, "D"'] & 2 \arrow[d, "B"] \\
3 \arrow[r, "C"'] & 4 \arrow[ul, "E"]
\end{tikzcd}
\qquad
\begin{tikzcd}
1 \arrow[d, "D"'] & 2 \arrow[d, "B"] \\
3 & 4 \arrow[ul, "E"]
\end{tikzcd}
$$

Now if we look more specifically at the graph below, we will see its incidence matrix to the right of it.

$$
\begin{tikzcd}
1 \arrow[d, "D"'] & 2 \arrow[d, "B"] \\
3 & 4 \arrow[ul, "E"]
\end{tikzcd}
\qquad
\begin{array}{c c}
 & \begin{array}{ccc} B & D & E \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{pmatrix} 0 & -1 & 1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & -1 \end{pmatrix}
\end{array}
$$

As you see from the incidence matrix, it tells us a lot about the graph. We can see that the rows represent the vertices of the graph and the columns represent the edges of the graph. Also, each column only has a single 1 and a single -1 and the rest are 0's. This is because an edge can only be a head and tail once.

2 Spanning Trees for Directed Graphs

Now that we have gone through all the preliminaries we can go over the two proofs. The first proof we will go over is the proof that will help make Kirchoff's Matrix Tree Theorem possible. This proof involves the number of spanning trees of a directed graph, where as Kirchoff's Matrix Tree Theorem involves the number of spanning trees of an undirected graph. The theorem of the amount of spanning trees of a directed graph is as follows:

Theorem 1. *Let G be a directed graph without loops and let A be the incidence matrix of G. Remove any row from A and let A_0 be the remaining matrix. Then the number of spanning trees of G is det $A_0 A_0^T$. [1]*

Before diving into the proof, we must understand that we will be looking at each possible subgraphs of the graph G that contain all the vertices, n, and have $n-1$ edges. It's almost like we are given the vertices and we throw all the edges of G in a bag and randomly select $n-1$ edges and check to see if it creates a spanning tree. We will do this until we have seen all possibilities of subgraphs. This seems very redundant and inefficient, but following the proof will show how we can do this in a slick manner. The proof is as follows:

Proof.
Assume, without loss of generality, that the last row of A is removed and call that A_0. Let B be an $(n-1)\times(n-1)$ submatrix of A_0. Now we will claim that the det $B=1$ if and only if the subgraph G' corresponding to the columns of B is a spanning tree. Otherwise, det $B=0$.

Let's assume there is a vertex $v_i(i \neq n)$ of degree one in G'. Then the ith row of B contains exactly one nonzero element, which is either 1 or -1. When we expand the det B by this row, the claim follows: G' is a spanning tree of G if and only if $G'-v_i$ is also a spanning tree of $G-v_i$. Therefore det $B=1$.

Now, let's assume that G' has no vertices of degree one. Then G' is not a spanning tree. Since G' has $n-1$ edges and it is not a spanning tree, there must be a vertex in G' that has degree 0. If this vertex is not v_n, then B has a zero row and the det $B=0$. If this vertex is v_n, then each column of B are linearly dependent and the det $B=0$.

So, we now know that det $B=1$ if the subgraph G' corresponding to the columns of B is a spanning tree and det $B=0$ otherwise. The Binet-Cauchy formula states:

$$\sum (detB)^2 = detA_0A_0^T$$

where the summation ranges over all $(n-1)\times(n-1)$ submatrices B of A_0. Thus, det $A_0A_0^T$ equals the number of spanning trees of G. □

As you can see, we found a way to have the determinant equal 1 or -1, which does not matter because it will end up being squared and equalling 1, when the randomly selected graph is a spanning tree. We also found a way to have the determinant equal 0 when the randomly selected graph is not a spanning tree. Then we used a seemingly irrelevant formula from Linear Algebra, the Binet-Cauchy formula, to easily check all of the possible subgraphs, which at first seemed inefficient and redundant.

3 Kirchoff's Matrix Tree Theorem

3.1 Theorem

Now on to the main proof of the paper. This theorem states that we can find the number of spanning trees of an undirected graph. It is as follows:

Theorem 2. *Let U be a simple undirected graph. Let $\{v_1, v_2, ..., v_n\}$ be the vertices of U. Define the $(n-1) \times (n-1)$ matrix L_0 by:*

$$l_{i,j} = \begin{cases} \textit{the degree of } v_i \textit{ if } i = j \\ -1 \textit{ if } i \neq j, \textit{ and } v_i \textit{ and } v_j \textit{ are connected} \\ 0 \textit{ otherwise} \end{cases}$$

where $1 \leq i$, $j \leq n-1$. Then U has exactly det L_0 spanning trees. [1]

This proof is a little bit shorter than the previous one but still uses the main ideas of the previous proof. It is important to notice that we are going to be slick and turn the undirected graph into a directed graph by giving each edge two directed edges, one going in each direction. This will cause our method of counting of spanning trees to be doubled from the previous theorem. The proof is as follows:

Proof.
First, turn U into a directed graph G, by each edge going in both directions. Let A_0 be the incidence matrix of G. Notice the entry of $A_0 A_0^T$ in position (i, j) is the scalar product of ith and jth row of A_0.

If $i = j$, then every edge that starts or ends at v_i contributes 1 to the product. Therefore, the entry of $A_0 A_0^T$ in position (i, i) is the degree of v_i in G, which is twice the degree of v_i in U.

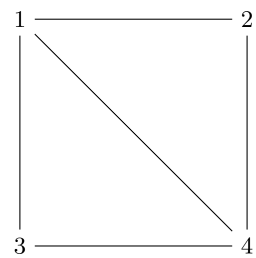
If $i \neq j$, then every edge that starts at v_i and ends at v_j and every edge that starts at v_j and ends at v_i contributes -1 to the product. Therefore, the entry of $A_0 A_0^T$ in position (i, j) is -2 if $v_i v_j$ is an edge of U and 0 otherwise.

So $A_0 A_0^T = 2L_0$. This implies that 2^{n-1} det L_0 = det $(A_0 A_0^T)$. Note that each spanning tree of U are 2^{n-1} different spanning trees of G. Thus, it follows that U has exactly det L_0 spanning trees. □

As you can see, everything was doubled since we made the undirected graph a directed graph by putting two edges, one going in either direction, where only one edge existed in the undirected graph. This caused the det $(A_0 A_0^T)$ to be doubled $n-1$ times. That is why when we divide by 2^{n-1} to get det L_0 alone, we actually have the amount of spanning trees, which happens to be det L_0.

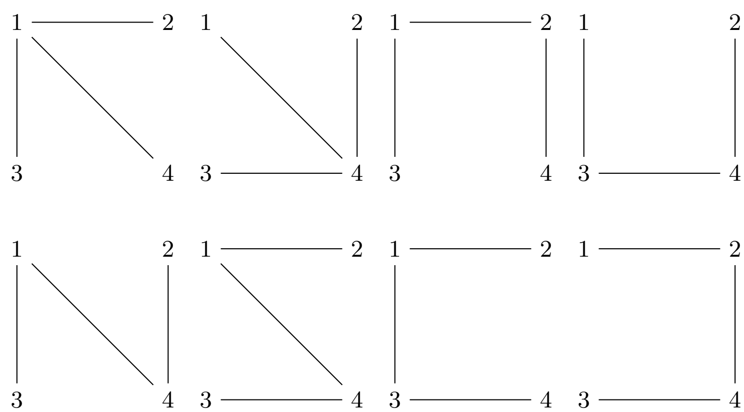
3.2 Example

For a quick example of this theorem, we will use the graph that we have been looking at throughout this paper.



$$\begin{array}{c c} & \begin{array}{cccc} \scriptstyle 1 & \scriptstyle 2 & \scriptstyle 3 & \scriptstyle 4 \end{array} \\ \begin{array}{c} \scriptstyle 1 \\ \scriptstyle 2 \\ \scriptstyle 3 \\ \scriptstyle 4 \end{array} & \begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 2 & 0 & -1 \\ -1 & 0 & 2 & -1 \\ -1 & -1 & -1 & 3 \end{pmatrix} \end{array}$$

We can remove any row and any column to get an $(n-1) \times (n-1)$ matrix, which in this case will be a 3×3 matrix. No matter which row you choose to eliminate, the resulting graph will have a determinant of 8. This means that there are eight spanning trees for this specific graph. These eight spanning trees are as follows:



References

[1] M. Bóna, *A walk through combinatorics: An introduction to enumeration and graph theory*, 3rd ed., New Jersey: World Scientific, 2017.