18.312: Algebraic Combinatorics — Lionel Levine

Lecture 20

Lecture date: Apr 26, 2011 — *Notes by: Josh Alman*

1 Plan for the Remainder of the Course

Today: Applications of the Matrix-Tree Theorem

- Hamming Cube
- Eulerian Tours

Next Class: De Bruijn sequences, Polya Theory

In-class final on Thurs, May 5^{th}

Last week: Abelian Sandpile Group

2 Hamming Cube

Our first application of the Matrix-Tree Thorem will be to find the number of spanning trees in the Hamming Cube.

Definition 1 *The* Hamming Cube of dimension n *is the undirected graph* $H_n = (V, E)$*, where* $V = \{0, 1\}^n$ *is the set of binary strings of length* n*, and* $x, y \in V$ *are adjacent iff* $x_i = y_i$ *for all but one index* $i \in [n]$.

Example 2 *For* $n = 3$*, we have that* H_3 *looks like:*

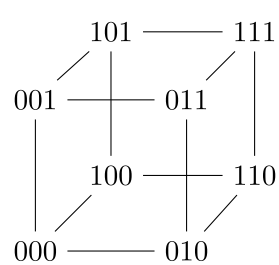

Figure 1: The Hamming Cube of dimension 3.

Question 3 *How many spanning trees are there in H_n?*

Answer: By the Matrix-Tree Theorem, the number of spanning trees is given by:

$$\kappa(H_n) = \frac{\lambda_1 \cdot \lambda_2 \cdots \lambda_{N-1}}{N}, \tag{1}$$

where $N = |V| = 2^n$ is the number of vertices, and $\lambda_1, ...\lambda_{N-1}$ are the non-zero eigenvalues of the Laplacian matrix:

$$L = nI - A_n,$$

where A_n is the adjacency matrix for H_n. Indeed, each vertex has degree n since it is adjacent to the vertex which differs from it only in the i^{th} position for all $i \in [n]$.

Now, we are going to find the eigenvalues of A_n by finding the eigenvalues of A_1 and writing the eigenvalues of A_n as sums of these. Notice that:

$$H_n = H_1 \times H_1 \times \cdots \times H_1 \qquad (n \text{ times}).$$

This gives us exactly that two vertices in H_n are adjacent if they differ in exactly one position, as we want. Thus, we get that:

$$A_n = A_1 \otimes I \otimes I \otimes \cdots \otimes I + I \otimes A_1 \otimes I \otimes \cdots \otimes I + \cdots + I \otimes I \otimes I \otimes \cdots \otimes I \otimes A_n. \tag{2}$$

One might think that we actually get $A_n = A_1 \otimes A_1 \otimes \cdots \otimes A_1 \quad (n \text{ times})$, but this would actually correspond to the graph where two binary strings are adjacent if they differ in every position. We can see that (2) is correct.

Now, H_1 looks like:

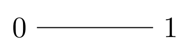

Figure 2: The Hamming Cube of dimension 1.

This has adjacency matrix:

$$A_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Letting $v_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ and $v_2 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$, we can see that:

$$A_1 v_1 = v_1, \text{ and } A_1 v_2 = -v_2,$$

so A_1 has eigenvectors v_1 and v_2 with eigenvalues 1 and -1, respectively. Thus, by (2), we get that the eigenvectors of A_n are the vectors of the form:

$$v = v_{i_1} \otimes v_{i_2} \otimes \cdots \otimes v_{i_n}, \qquad \text{where } i_r \in \{1, 2\} \ \forall r \in [n].$$

The corresponding eigenvalue for this eigenvector of A_n is simply the sum of the eigenvalues of the 'v_{i_r}'s, which is:

$$\lambda_v = \sum_{i_r=1} 1 + \sum_{i_r=2} (-1)$$
$$= n - 2 \cdot \#\{r \mid i_r = 2\}.$$

Thus, since there are $\binom{n}{k}$ ways to pick k of the n 'i_r's to equal 2, we have that A_n has the eigenvalue $n - 2k$ with multiplicity $\binom{n}{k}$ for each $k = 0, 1, ..., n$; this is all the 2^n eigenvalues of A_n. Recalling that:

$$L_n = nI - A_n,$$

we get that L_n has eigenvalue $n-(n-2k) = 2k$ with multiplicity $\binom{n}{m}$ for each $k = 0, 1, ..., n$. We can finally evaluate (1), keeping in mind that we do not include zero eigenvalues in our product:

$$\kappa(H_n) = \frac{\prod_{k=1}^{n} (2k)^{\binom{n}{k}}}{N} = \frac{\prod_{k=1}^{n} (2k)^{\binom{n}{k}}}{2^{\binom{n}{1}}} = \prod_{k=2}^{n} (2k)^{\binom{n}{k}} \tag{3}$$

Thus, the number of spanning trees in H_n is $\boxed{\prod_{k=2}^{n} (2k)^{\binom{n}{k}}.}$

Surprisingly, despite the form of this answer, finding a combinatorial proof of it is an open problem. One promising approach involves corresponding spanning trees in H_n to phototropic trees in H_n - trees where the sun is placed at the origin, and edges "grow toward the light." More formally:

Definition 4 *A directed spanning tree T_n of H_n, rooted at the origin, 0^n, is called* phototropic *if whenever the arc $(x, y) \in T_n$, we have that $x_i \geq y_i$ $\forall i \in [n]$.*

Example 5 *Here are directed spanning trees in H_2:*

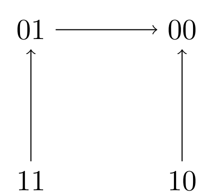

Figure 3: A phototropic tree in H_2

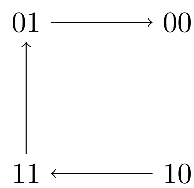

Figure 4: A tree in H_2 that is not phototropic, since there is an arc between 10 and 11 in the wrong direction.

Counting phototrpphic trees is simple: we can construct them by choosing an outward arc for each vertex other than the origin. The arc from a vertex x must go to a vertex identical to x except that it has a zero in one position where x has a one, and so if there are a_x ones in x, then there are a_x choices for that arc. The number of phototropic trees in H_n is thus:

$$\#\{\text{phototropic trees in } H_n\} = \prod_{x \in \{0,1\}^n \setminus 0^n} a_x = \prod_{x \in \{0,1\}^n \setminus 0^n} \left(\sum_{i=1}^{n} x_n \right) = \prod_{k=1}^{n} k^{\binom{n}{k}}$$

The similarity of this answer to (3) leads to the idea of corresponding phototropic trees to spanning trees to produce a combinatorial proof of our earlier result.

3 Eulerian Tours

3.1 Introducing the problem

In this section, we will use a tricky application spanning trees to find the number of Eulerian tours in a directed graph.

Definition 6 *If $G = (V, E)$ is a finite directed graph, with $n = |V|, m = |E|$, then a* (directed) Eulerian tour *of G is a path $t_0, t_1, ..., t_m = t_0$ with each $t_i \in V$, such that each directed edge is used exactly once, meaning, $\{(t_i, t_{i+1})\}_{i=0}^{m-1} = E$.*

Example 7 *Here are some Eulerian tours on $\overleftrightarrow{K_3}$:*

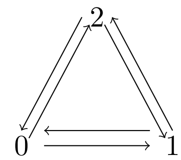

Figure 5: The complete directed graph on 3 vertices.

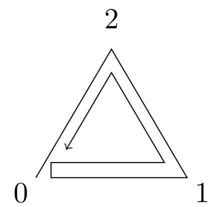

Figure 6: The Eulerian Tour 021012

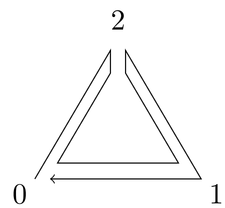

Figure 7: The Eulerian Tour 020121

Question 8 *Which directed graphs have Eulerian tours?*

One might recall that an undirected graph has an Eulerian tour if every vertex has even degree. However, here, we are dealing with directed graphs, so the condition will require that our graph be balanced.

Definition 9 *In a directed graph $G = (V, E)$, a vertex $v \in V$ is said to be* balanced *if it has equal indegree and outdegree;*

$$indeg(v) = outdeg(v).$$

Definition 10 *A directed graph $G = (V, E)$ is said to be* balanced *if each of its vertices is balanced.*

We can then state our condition on a directed graph having an Eulerian tour:

Answer 11 *A directed graph G=(V,E) has an Eulerian tour iff it is balanced.*

It is clear that a directed graph needs to be balanced to have an Eulerian tour. The converse ends up being true as well. We will see this, but we also want to count the number of Eulerian tours in a directed graph that has any.

Definition 12 *If $G = (V, E)$ is a directed graph, with $e \in E$ an arc in the graph, then we write:*

$$\tau(G, e) := \#\{\textit{Eulerian tours of } G \textit{ starting with the edge } e\}.$$

Since an Eulerian tour goes over all arcs in a graph, and we can cycle our tours to start at any edge we want, we can see that $\tau(G, e)$ is actually independent of e, so we can similarly define $\tau(G) := \tau(G, e)$ for any $e \in E$. Recalling that $\kappa(G, v_0)$ is the number of oriented spanning trees in directed graph G rooted at vertex v_0, we can then state our theorem for counting Eulerian tours in directed graphs:

Theorem 13 *If $G = (V, E)$ is a directed graph, then G has an Eulerian tour iff it is balanced. If it is balanced, then for any $e = (v_0, v_1) \in E$,*

$$\tau(G) = \tau(G, e) = \kappa(G, v_0) \cdot \prod_{v \in V} (deg(v) - 1)!. \tag{4}$$

We quickly introduce some new notation:

Definition 14 *If $e \in E$ is an arc in directed graph $G = (V, E)$, and $e = (v, w)$, then we write:*

$$e^- := v,$$
$$e^+ := w.$$

Then, for a vertex $x \in V$, we write E_x for the set of arcs coming out of x:

$$E_x := \{f \in E | f^- = x\}.$$

We want to interpret the $(deg(v) - 1)!$ terms in (4) as cyclic permutations of the edges in E_v. We thus define the following:

Definition 15 *Given an Eulerian tour* $\gamma = (t_0, t_1, ..., t_{m-1})$ *in directed graph* $G = (V, E)$, *for any vertex* $v \in V$, *let* $i_1, ..., i_d$ *be the indices such that* $t_{i_r} = v, r = 1, ..., d = outdeg(v)$. *Then, for each* r, *write* $e_r = (v, t_{i_r+1})$; *the '*e_r*' are the edges in* E_v *in the order that they are crossed in* γ. *Then, we can define the cyclic permutation* $c_v : E_v \to E_v$ *as:*

$$c_v(e_r) = \begin{cases} e_{r+1} & \text{if } 1 \leq r < d \\ e_1 & \text{if } r = d \end{cases}$$

To use this new cyclic permutation, we are going to use some properties of Rotor Walks.

3.2 Rotor Walks

Definition 16 *A* rotor configuration *on a directed graph* $G = (V, E)$ *is a map* $\rho : V \to E$ *such that* $\rho(v) \in E_v \ \forall v \in v$; *it assigns an outgoing edge to each vertex.*

Definition 17 *Given a directed graph* $G = (V, E)$ *and associated* ρ *and a* c_v *for each* $v \in V$, *and a starting vertex* V_0, *define a* rotor walk *as a sequence of vertices in* V, $v_0, v_1, v_2, ...$, *as follows: let* $\rho_0 = \rho$, *then for each* $i = 1, 2, ...$, *let:*

$$\rho_i(w) = \begin{cases} \rho_{i-1}(w) & \text{if } w \neq v_{i-1} \\ c_{v_{i-1}}(\rho_{i-1}(v_{i-1})) & \text{if } w = v_{i-1}. \end{cases}$$

Then, let $v_i = \rho_i(v_{i-1})^+$.

Basically, at each step, the rotor of the vertex we are looking at goes to the next edge coming out of that vertex in its cyclic permutation of edegs, and we move along that new edge. We illustrate this with an example on $\overleftrightarrow{K_3}$:

Example 18 *We give the initial conditions in ρ_0; at each step, the arcs in the current ρ are shown:*

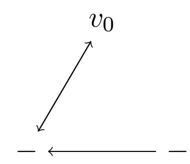

Figure 8: ρ_0

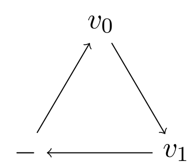

Figure 9: ρ_1

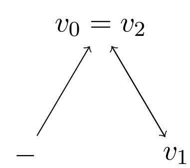

Figure 10: ρ_2

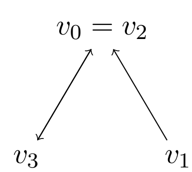

Figure 11: ρ_3

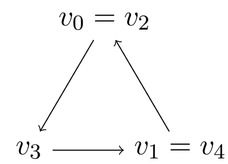

Figure 12: ρ_4

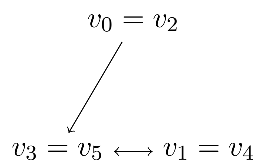

Figure 13: ρ_5

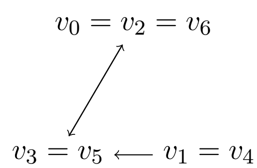

Figure 14: ρ_6

Since $\rho_0 = \rho_6$ and $v_0 = v_6$, we are now going to repeat; rotor walking is deterministic. Notice that the vertices traversed on the path, $(v_0, v_1, v_2, v_3, v_4, v_5) = (2, 1, 2, 0, 2, 1)$, form an Eulerian path of the graph! It turns out that the reason for this is that ρ_0 formed a spanning tree of the graph.

3.3 Finding the number of directed Eulerian tours

Lemma 19 *If directed graph $G = (V, E)$ is balanced, and the graph $T = (V, \rho(V - \{v_0\}))$ is a tree, then there exists an Eulerian tour $t_0, t_1, ..., t_{m-1}$ such that the rotor walk $v_0, v_1, v_2, ...$ induced by ρ travels around the tour, namely:*

$$\begin{aligned} v_0 &= t_0, \ldots, v_{m-1} = t_{m-1} \\ v_m &= t_0, \ldots, v_{2m-1} = t_{m-1} \\ &\cdots \\ v_i &= t_{i+m} \qquad \forall i \geq 0. \end{aligned}$$

Proof: Let $e = (v_i, v_{i+1})$ be the first edge travelled twice by the rotor walk; i is the smallest index such that $\exists j < i$ such that $v_j = v_i$ and $v_{j+1} = v_{i+1}$. This is well-defined since there are finitely many edges and the path is infinite. Let $d = deg(v_i)$. Since each outgoing edge of v_i had to be traversed before e could be traversed a second time, as the rotor goes over all these edges, after step i, there have been $d + 1$ exits from v_i. This means there were $d + 1$ entrances to v_i. But, since v is balanced, there are only d edges coming into it, so one of these edges were already traversed twice, a contradiction, unless $v_i = v_0$, as we start there. Thus, the first edge travelled twice started from v_0.

Now, we call a vertex w full if all its incoming edges have been used before time i, meaning, for each edge e such that $e^+ = w$, there exists a $j < i$ such that $e = (v_j, v_{j+1})$. Note that v_0 is full, since it had d entrances other than at the beginning, and d incoming edges, and none of these were traversed twice.

Notice that if a vertex w is full, and (u, w) is an edge of our initial spanning tree T, then u is also full if it is not v_0. Indeed, $(u, w) \in T$ means that $\rho_0(u) = (u, w)$, meaning (u, v) is the last edge traveled out of u in the rotor walk. But, since w is full, (u, w) was used, so all the edges out of u were used. Thus, there were $deg(u)$ entrances to u, and since u is not v_0, all incoming edges were used, so u is full.

We have that the root of the spanning tree is full, and every non-root vertex adjacent to a full vertex in the tree is full. Since T is a spanning tree, all the vertices are full. Thus, each edge was used exactly once. □

We can now prove our theorem! We recall what we wanted to show:

Theorem 20 *If $G = (V, E)$ is a directed graph, then G has an Eulerian tour iff it is balanced. If it is balanced, then for any $e = (v_0, v_1) \in E$,*

$$\tau(G) = \tau(G, e) = \kappa(G, v_0) \cdot \prod_{v \in V} (deg(v) - 1)!. \tag{5}$$

Proof: We define a bijection:

$$f : \{\text{Eulerian tours of } G \text{ starting with edge } e = (v_0, v_1)\} \to$$

$$\to \{\text{oriented spanning trees of } G \text{ rooted at } v_0\} \times \prod_{v \in V} \{\text{cyclic permutations of } E_v\},$$

where the products in the right hand side are cartesian products of sets. Proving this will clearly imply (5).

We can define f^{-1} as: $f^{-1}(\rho, (c_v)_{v \in V}) \to$ rotor walk $(v_0, v_1, ...)$, where the rotor walk is exactly what we defined in the lemma. Notiice that an oriented spanning tree does not actually give us a value for $\rho(v_0)$, but we can define it as $\rho(v_0) = c_{v_0}^{-1}(v_0, v_1)$.

We can define f as: $f(t_0, t_1, ..., t_{m-1}) = (\rho, (c_v)_{v \in V})$, where the '$c_v$' are the orders of the edges taken in the tour, and $\rho(u)$ is the last edge travelled out of u in the tour; $\rho(u) = (u, w)$, where $w = v_{M+1}$, where $M = max\{i < m | v_i = u\}$. With ρ defined in this way, $(V, \rho(V - \{v_0\}))$ is a tree because given $S \subset V - \{v\}$, letting $N = max\{i < m | v_i \in S\}$, we have that (v_N, v_{N+1}) is an edge of T and $v_{N+1} \notin S$. Thus, S does not form an oriented cycle, and so T is a tree.

It is clear that these are inverses. □

It is worth noting that it is not clear that the number of oriented spanning trees rooted at v is independent of v; this is problem PF6 on the practice final.