A Random Walk Proof of Matrix Tree Theorem

Larissa Richards
CUMC 2013

Based on joint work with Michael Kozdron (Regina) and Dan Stroock (MIT).
Preprint available from `arXiv:1306.2059`.

Outline

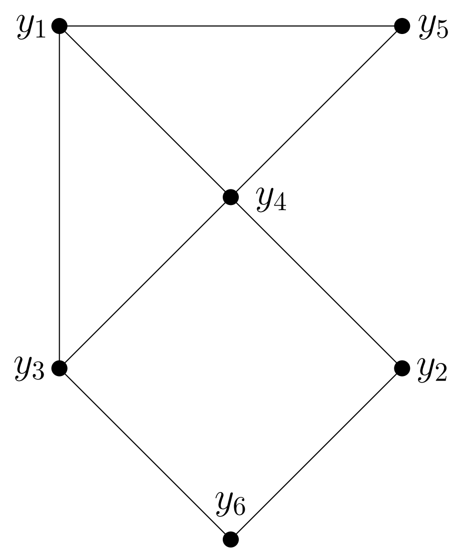
- Kirchhoff's Matrix Tree Theorem
- Random Walk on a Graph
- Wilson's Algorithm
- Proof of Wilson's Algorithm and Kirchhoff's Matrix Tree Theorem
- Application: Cayley's Formula

Set-up

Suppose that $\Gamma = (V, E)$ is a finite graph consisting of $n+1$ vertices labelled $y_1, y_2, \cdots, y_n, y_{n+1}$.

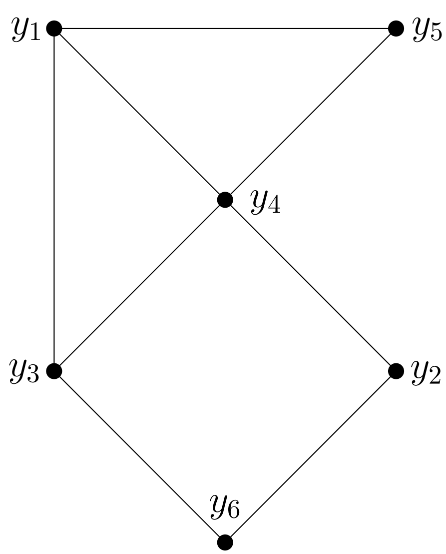
- undirected
- connected
- no multiple edges

Note that $y_i \sim y_j$ are nearest neighbours if $(y_i, y_j) \in E$.



The Graph Laplacian Matrix

Recall that the <u>graph Laplacian</u> $\mathcal{L}$ is the matrix $\mathcal{L} = \mathcal{D} - \mathcal{A}$, where $\mathcal{D}$ is the degree matrix and $\mathcal{A}$ is the adjacency matrix.



$$\mathcal{L} = \begin{array}{c} \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{array} \begin{array}{c} \begin{array}{cccccc} y_1 & y_2 & y_3 & y_4 & y_5 & y_6 \end{array} \\ \begin{bmatrix} 3 & 0 & -1 & -1 & -1 & 0 \\ 0 & 2 & 0 & -1 & 0 & -1 \\ -1 & 0 & 3 & -1 & 0 & -1 \\ -1 & -1 & -1 & 4 & -1 & 0 \\ -1 & 0 & 0 & -1 & 2 & 0 \\ 0 & -1 & -1 & 0 & 0 & 2 \end{bmatrix} \end{array}$$

Kirchhoff's Matrix Tree Theorem

Suppose that $\mathcal{L}^{\{k\}}$ denotes the submatrix of $\mathcal{L}$ obtained by deleting row k and column k corresponding to vertex y_k.

Suppose that $\lambda_1, \lambda_2, \cdots, \lambda_n$ are the nonzero eigenvalues of $\mathcal{L}$.

Theorem (Kirchhoff). If $\Omega = \{\text{spanning trees of } \Gamma\}$, then

$$\det[\mathcal{L}^{\{1\}}] = \det[\mathcal{L}^{\{2\}}] = \cdots = \det[\mathcal{L}^{\{n\}}] = \det[\mathcal{L}^{\{n+1\}}] = \frac{\lambda_1 \cdots \lambda_n}{n+1}$$

and that these are equal to $|\Omega|$, the number of spanning trees of Γ.

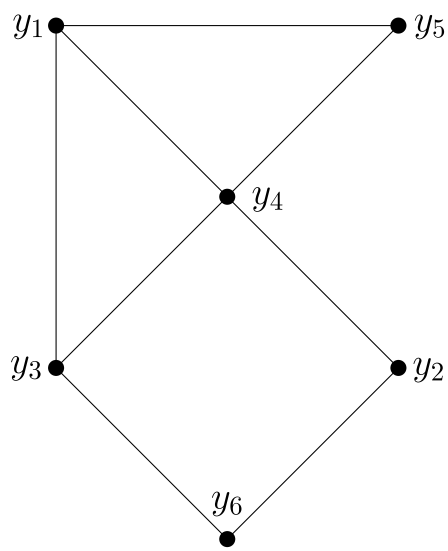
Practically, this is very hard to compute!

Usual modern way to prove MTT is purely algebraic and involves Cauchy-Binet formula.

Example

This graph has 29 spanning trees.

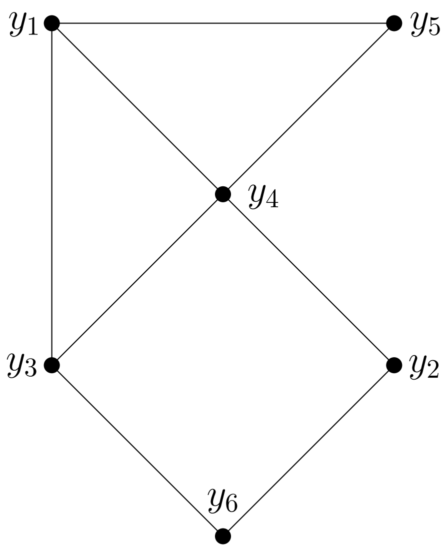
To see this, consider $\deg(y_4)$ in the spanning tree.



$\deg(y_4)$	No. Spanning Trees
4	2
3	10
2	13
1	4
	29

Example (cont.)

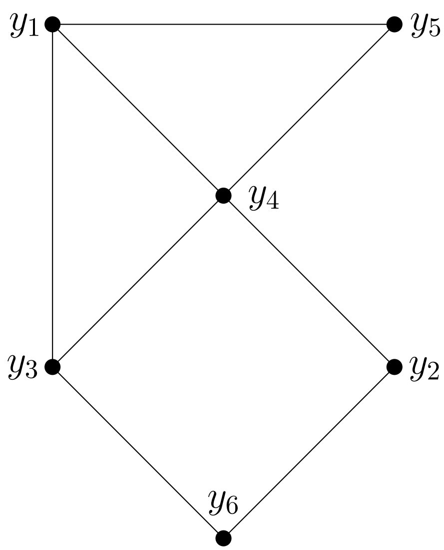
This graph has 29 spanning trees. For example, using MTT, $\det[\mathcal{L}^{\{5\}}] = 29$.



$$\mathcal{L}^{\{5\}} = \begin{array}{c} \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_6 \end{array} \begin{array}{c} \begin{array}{ccccc} y_1 & y_2 & y_3 & y_4 & y_6 \end{array} \\ \begin{bmatrix} 3 & -1 & -1 & -1 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ -1 & -1 & 4 & -1 & -1 \\ -1 & 0 & -1 & 3 & 0 \\ 0 & 0 & -1 & 0 & 2 \end{bmatrix} \end{array}$$

Example: Random Walk on a Graph

Choose the next step equally likely from among all possible nearest neighbours.



$$\mathbb{P} = \mathcal{D}^{-1}\mathcal{A} = \begin{array}{c} \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{array} \begin{array}{c} \begin{array}{cccccc} y_1 & y_2 & y_3 & y_4 & y_5 & y_6 \end{array} \\ \begin{bmatrix} 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{4} & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Random Walk on a Graph

Formally, a simple random walk $\{S_k,\ k = 0, 1, \cdots\}$ on graph Γ is a time-homogeneous Markov Chain with transition probabilities

$$
\begin{aligned}
P\{S_{k+1} = y_j \mid S_0 = y_{i_0}, \cdots, S_{k-1} = y_{i_{k-1}}, S_k = y_i\}& \\
= P\{S_{k+1} = y_j \mid S_k = y_i\}& \\
= P\{S_1 = y_j \mid S_0 = y_i\}& \\
= p(i, j)&
\end{aligned}
$$

where $p(i, j)$ is the (i, j)-entry of $\mathbb{P} = \mathcal{D}^{-1}\mathcal{A}$.

Note that

$$
p(i, j) = \begin{cases} \frac{1}{\deg(y_i)}, & \text{if } y_i \sim y_j \\ 0, & \text{else.} \end{cases}
$$

Random Walk on a Graph

Recall. The graph Laplacian matrix $\mathcal{L}$ is defined by $\mathcal{L} = \mathcal{D} - \mathcal{A}$.

We can rewrite it as

$$\mathcal{L} = \mathcal{D}(\mathbb{I} - \mathcal{D}^{-1}\mathcal{A}) = \mathcal{D}(\mathbb{I} - \mathbb{P}).$$

Let $\Delta \subset V$, $\Delta \neq \emptyset$. Then

$$\mathcal{L}^{\Delta} = \mathcal{D}^{\Delta}(\mathbb{I}^{\Delta} - \mathbb{P}^{\Delta})$$

for the matrices obtained by deleting the rows and columns associated to the entries in Δ.

Note that $\mathbb{P}^{\Delta}$ is strictly substochastic; that is, non-negative entries and rows sum to at most 1 with at least one row sum less than 1.

Thus $(\mathbb{I}^{\Delta} - \mathbb{P}^{\Delta})^{-1}$ exists.

The Key Random Walk Facts

Let $\zeta^{\Delta} = \inf\{j \geq 0 : \ S_j \in \Delta\}$ be the first time the random walk visits $\Delta \subset V$.

For $x, y \notin \Delta$, let

$$G_{\Delta}(x,y) = E^x \left[\sum_{k=0}^{\infty} 1\{S_k = y, \ k < \zeta^{\Delta}\} \right]$$

be the expected number of visits to y by simple random walk on Γ starting at x before entering Δ.

This is often called the random walk Green's function.

- If $\mathbb{G}^{\Delta} = [G_{\Delta}(x,y)]_{x,y \in V \setminus \Delta}$, then

$$\mathbb{G}^{\Delta} = (\mathbb{I}^{\Delta} - \mathbb{P}^{\Delta})^{-1}.$$

- If $r_{\Delta}(x)$ denotes the probability that simple random walk starting at x returns to x before entering Δ, then

$$G_{\Delta}(x,x) = \sum_{k=0}^{\infty} r_{\Delta}(x)^k = \frac{1}{1 - r_{\Delta}(x)}.$$

Wilson's Algorithm (1996)

Wilson's Algorithm generates a spanning tree uniformly at random without knowing the number of spanning trees.

- Pick any vertex. Call it v.
- Relabel remaining vertices $x_1, \cdots, x_n$.
- Start a simple random walk at x_1. Stop it the first time it reaches v.
- Erase loops.
- Find the first vertex not in the backbone.
- Start a simple random walk at it.
- Stop it when it hits the backbone.
- Erase loops.
- Repeat until all vertices are included in the backbone.

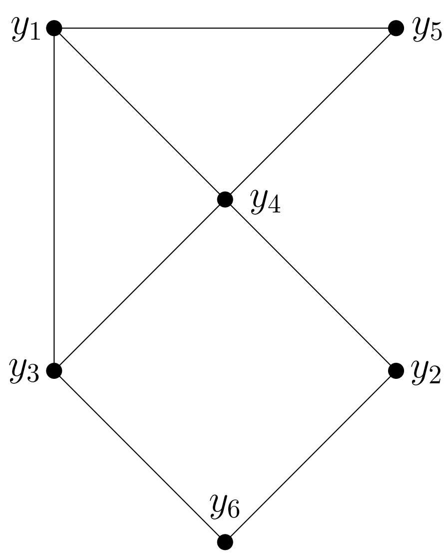
Clearly, this generates a spanning tree. We will prove that it is uniform among all possible spanning trees.

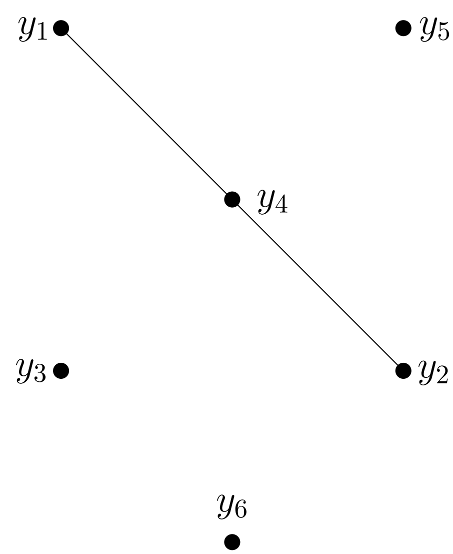
Example: Wilson's Algorithm on Γ

It is easier to explain this example illustrating Wilson's algorithm without relabelling the vertices.

Start a SRW at y_2. Stop it when it first reaches y_1.

Assume the loop-erasure is $[y_2, y_4, y_1]$. Add this branch to the spanning tree.

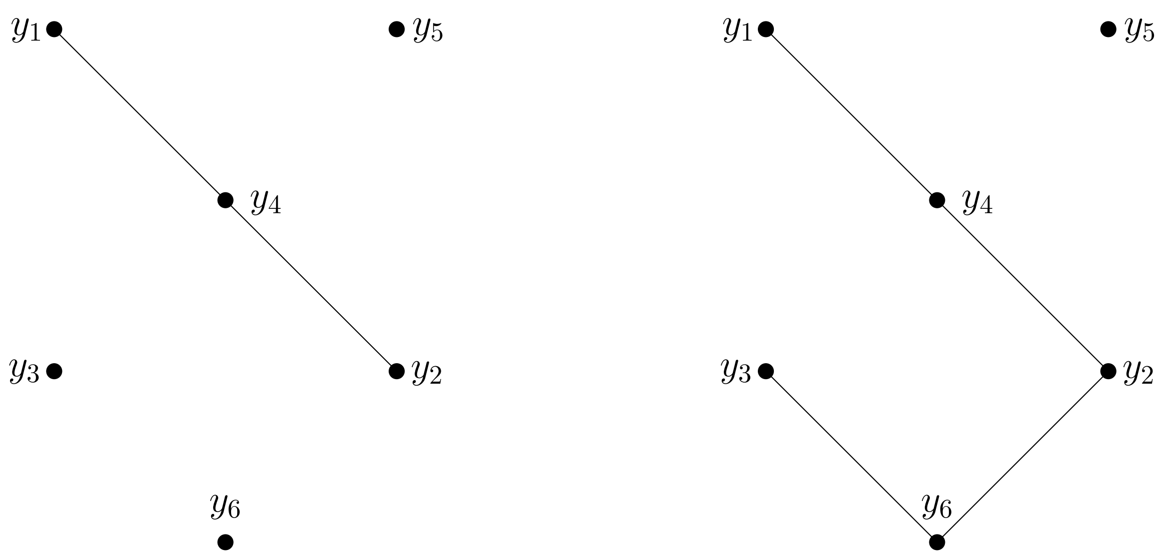




Example: Wilson's Algorithm on Γ

Start a SRW at y_3. Stop it when it reaches $\{y_2, y_4, y_1\}$.

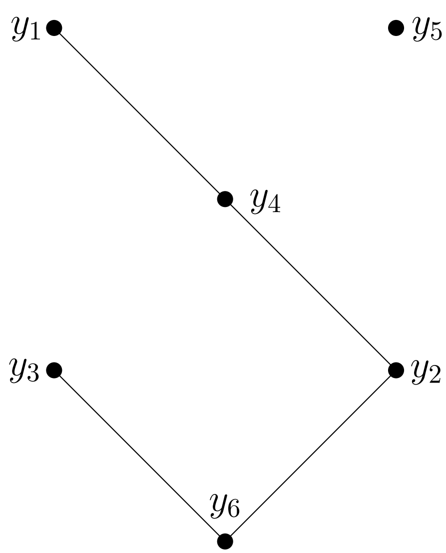
Assume the loop-erasure is $[y_3, y_6, y_2]$. Add this branch to the spanning tree.

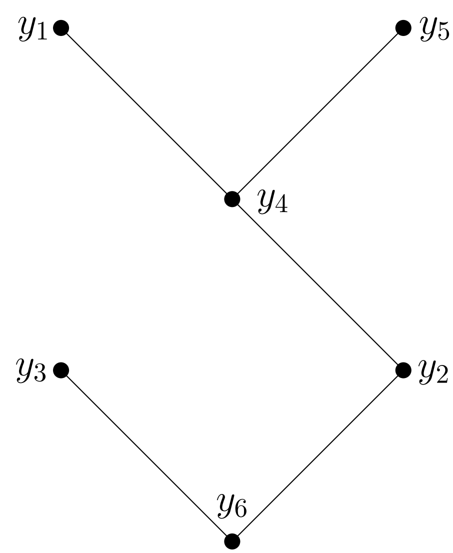


Example: Wilson's Algorithm on Γ

Finally, start a SRW at y_5 and stop it when it reaches $\{y_2, y_4, y_1\} \cup \{y_3, y_6\}$.

Assume the loop-erasure is $[y_5, y_4]$. Add this branch to the spanning tree.





We have generated a spanning tree of Γ with three branches
$\Delta_1 = [y_2, y_4, y_1]$, $\Delta_2 = [y_3, y_6, y_2]$, $\Delta_3 = [y_5, y_4]$.

Some History

- Kirchhoff - 1800s
  Gustav Kirchhoff was motivated to study spanning trees by problems arising from his work on electrical networks.
- Wilson's Algorithm - 1996
  David Wilson used "cycle-popping" to prove his algorithm generated a uniform spanning tree. His original proof is of a very different flavour. The Matrix Tree Theorem does not follow directly from the cycle-popping proof.
- Greg Lawler - 1999
  Lawler discovered a new, computational proof of Wilson's Algorithm via Green's functions. The Matrix Tree Theorem follows immediately as a corollary to his proof.

Our original goal was to give an expository account of Lawler's proof. However, in addition to simplifying his proof, we discovered that these ideas could be applied to deduce results for Markov processes.

Computing a Loop-Erased Walk Probability

Suppose $\Delta \subset V, \;\; \Delta \neq \emptyset$.

Let $x_1, \cdots, x_K$ be distinct elements of a connected subset of $V \backslash \Delta$ labelled in such a way that $x_j \sim x_{j+1}$ for $j = 1, \cdots, K$. Note that $x_{K+1} \in \Delta$.

Consider simple random walk on Γ starting at x_1. Set $\xi^{\Delta} = \inf\{j \geq 0 \; : \; S_j \in \Delta\}$.

Let

$$P^{\Delta}(x_1, \cdots, x_K, x_{K+1}) := P\{L(\{S_j, \; j = 0, \cdots, \xi^{\Delta}\}) = [x_1, \cdots, x_K, x_{K+1}]\}$$

denote the probability that loop-erasure of $\{S_j, \; j = 0, \cdots, \xi^{\Delta}\}$ is exactly $[x_1, \cdots, x_{K+1}]$.

Computing a Loop-Erased Walk Probability

Question: How can we compute

$$P\{L(\{S_j,\ j=0,\cdots,\xi^{\Delta}\}) = [x_1,\cdots,x_K,x_{K+1}]\}?$$

For the loop-erasure to be exactly $[x_1,\cdots,x_{K+1}]$, we need that:

- the SRW started at x_1, then
- made a number of loops back to x_1 without entering Δ, then
- took a step from x_1 to x_2, then
- made a number of loops back to x_2 without entering $\Delta \cup \{x_1\}$, then
- took a step from x_2 to x_3, then
- made a number of loops back to x_3 without entering $\Delta \cup \{x_1, x_2\}$, then
- $\cdots$
- made a number of loops back to x_K without entering $\Delta \cup \{x_1, x_2, \cdots, x_{K-1}\}$, then
- took a step from x_K to $x_{K+1} \in \Delta$.

Computing a Loop-Erased Walk Probability

So,

$$
\begin{aligned}
&P^{\Delta}(x_1, \cdots, x_{K+1}) \\
&\quad = \sum_{m_1, \cdots, m_K = 0}^{\infty} r_{\Delta}(x_1)^{m_1} p(x_1, x_2) r_{\Delta \cup \{x_1\}}(x_2)^{m_2} p(x_2, x_3) \cdots \\
&\qquad\qquad \cdots r_{\Delta \cup \{x_1, \cdots, x_{K-1}\}}(x_K)^{m_K} p(x_K, x_{K+1}) \\
&\quad = \prod_{j=1}^{K} \frac{1}{\deg(x_j)} \frac{1}{1 - r_{\Delta(j)}(x_j)} \\
&\quad = \prod_{j=1}^{K} \frac{1}{\deg(x_j)} G_{\Delta(j)}(x_j, x_j)
\end{aligned}
$$

where $\Delta(1) = \Delta$ and $\Delta(j) = \Delta \cup \{x_1, \cdots, x_{j-1}\}$ for $j = 2, \cdots, K$ and the last line follows from the key random walk fact.

Proof of Wilson's Algorithm

Suppose that $\mathcal{T} \in \Omega$ was produced by Wilson's algorithm with branches

$$\Delta_0 = \{v\},\ \Delta_1 = [x_{1,1}, \cdots, x_{1,k_1}], \cdots, \Delta_L = [x_{L,1}, \cdots, x_{L,k_L}].$$

We know that each branch in Wilson's algorithm is generated by a loop-erased random walk.

$$P(\mathcal{T} \text{ is generated by Wilson's algorithm}) = \prod_{l=1}^{L} P^{\Delta^l}(x_{l,1}, \cdots, x_{l,k_l})$$

where $\Delta^l = \Delta_0 \cup \cdots \cup \Delta_{l-1}$ for $l = 1, \cdots, L$.

Proof of Wilson's Algorithm

Recall: The Loop-Erased Walk Probability Calculation

$$P^{\Delta}(x_1, \cdots, x_K, x_{K+1}) = \prod_{j=1}^{K} \frac{G_{\Delta(j)}(x_j, x_j)}{\deg(x_j)}.$$

Hence, the probability that $\mathcal{T}$ is generated by Wilson's algorithm is

$$\prod_{l=1}^{L} P^{\Delta^l}(x_{l,1}, \cdots, x_{l,k_l}) = \prod_{l=1}^{L} \prod_{j=1}^{k_l-1} \frac{G_{\Delta^l(j)(x_{l,j}, x_{l,j})}}{\deg(x_{l,j})}$$

where $\Delta^l(1) = \Delta^l$ and $\Delta^l(j) = \Delta^l \cup \{x_{l,1}, \cdots, x_{l,j-1}\}$ for $j = 2, \cdots, k_l - 1$.

To finish the proof we need some facts from linear algebra.

Some Linear Algebra

M is a non-degenerate $N \times N$ matrix and $\Delta \subset \{1, 2, \cdots, N\}$.

M^{Δ}: matrix formed by deleting rows and columns corresponding to indices in Δ.

1. Cramer's Rule

$$(M^{-1})_{ii} = \frac{\det[M^{\{i\}}]}{\det[M]}$$

2. Suppose $(\sigma(1), \cdots, \sigma(N))$ is a permutation of $(1, \cdots, N)$. Set $\Delta_1 = \emptyset$ and for $j = 2, \cdots, N$, let $\Delta_j = \Delta_{j-1} \cup \{\sigma(j-1)\} = \{\sigma(1), \cdots, \sigma(j-1)\}$. If $M^{\Delta(j)}$ is non-degenerate for all $j = 1, \cdots, N$, then

$$\det[M]^{-1} = \prod_{j=1}^{N} (M^{\Delta_j})^{-1}_{\sigma(j),\sigma(j)}.$$

Proof of Wilson's Algorithm

Recall that we picked an arbitrary vertex v where we stopped our initial walk.

Also recall that $\mathbb{G}^{\Delta} = (\mathbb{I}^{\Delta} - \mathbb{P}^{\Delta})^{-1}$ and $\mathcal{L}^{\Delta} = \mathcal{D}^{\Delta}(\mathbb{I}^{\Delta} - \mathbb{P}^{\Delta})$.

By Linear Algebra fact 2,

$$\prod_{l=1}^{L} \prod_{j=1}^{k_l - 1} G_{\Delta^l(j)}(x_{l,j}, x_{l,j}) = \det[\mathbb{G}^{\{v\}}].$$

Thus

$$P(\mathcal{T} \text{ is generated by Wilson's algorithm}) = \frac{\det[\mathbb{G}^{\{v\}}]}{\det[\mathcal{D}^{\{v\}}]} = \frac{1}{\det[\mathcal{D}^{\{v\}}]\det[\mathbb{I}^{\{v\}} - \mathbb{P}^{\{v\}}]}$$
$$= \det[\mathcal{L}^{\{v\}}]^{-1}.$$

In addition, we can see that the right hand side of the equation is independent of the ordering of the remaining n vertices. Thus,

$$P(\mathcal{T} \text{ is generated by Wilson's algorithm}) = \det[\mathcal{L}^{\{v\}}]^{-1} = \frac{1}{|\Omega|}$$

Corollary: Proof of the Matrix Tree Theorem

Since

$$P(\mathcal{T} \text{ is generated by Wilson's algorithm}) = \det[\mathcal{L}^{\{v\}}]^{-1} = \frac{1}{|\Omega|}$$

we have

$$|\Omega| = \det[\mathcal{L}^{\{v\}}].$$

Since v was arbitrary, we conclude

$$|\Omega| = \det[\mathcal{L}^{\{1\}}] = \det[\mathcal{L}^{\{2\}}] = \cdots = \det[\mathcal{L}^{\{n\}}] = \det[\mathcal{L}^{\{n+1\}}].$$

<u>Note.</u> A separate argument is needed to show that

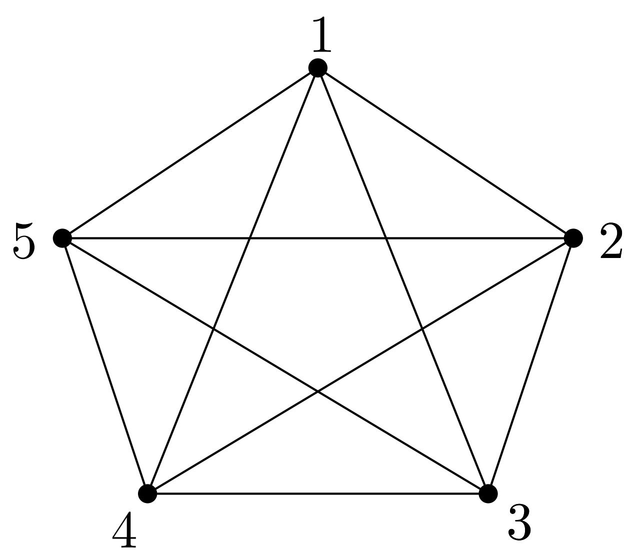
$$|\Omega| = \frac{\lambda_1 \cdots \lambda_n}{n+1}$$

where $\lambda_1, \ldots, \lambda_n$ are the non-zero eigenvalues of $\mathcal{L}$. (This is found in our paper, but not in this talk.)

Application: Cayley's Formula

If $\Gamma = (V, E)$ is a complete graph on $N + 1$ vertices; i.e., there is an edge connecting any two vertices in V. Then

$$\text{Number of spanning trees of } \Gamma = (N+1)^{N-1}.$$



The number of spanning trees of K_5 is $5^3 = 125$.

Application: Cayley's Formula

Start a simple random walk at x.

Suppose that $\Delta \subset V \backslash \{x\}$, where $\Delta \neq \emptyset,\ |\Delta| = m$.

<u>Recall</u>.
$r_\Delta(x)$ is the probability that simple random walk starting at x returns to x before entering Δ.

Let $r_\Delta(x; k)$ be the probability that simple random walk starting at x returns to x in exactly k steps without entering Δ so that

$$r_\Delta(x) = \sum_{k=2}^{\infty} r_\Delta(x; k).$$

Note that a SRW cannot return to its starting point in only 1 step.

Application: Cayley's Formula

Since Γ is the complete graph on $N+1$ vertices, we have partitioned the vertex set:

$$V_1 = \{x\},\ V_2 = \Delta \text{ with } |V| = m, \text{ and } V_3 \text{ with } |V_3| = N - m.$$

Thus,

$$\begin{aligned} r_\Delta(x;k) &= P\{S_0 = x,\ S_1 \in V_3, \cdots, S_{k-1} \in V_3, S_k = x\} \\ &= \frac{N-m}{N}\left(\frac{N-1-m}{N}\right)^{k-2}\frac{1}{N} \end{aligned}$$

and so

$$\begin{aligned} r_\Delta(x) &= \frac{N-m}{N^2}\sum_{k=2}^{\infty}\left(\frac{N-1-m}{N}\right)^{k-2} \\ &= \frac{N-m}{N(m+1)}. \end{aligned}$$

Application: Cayley's Formula

By the key random walk fact,

$$G_\Delta(x,x) = \frac{1}{1 - r_\Delta(x)} = \frac{N(m+1)}{m(N+1)}. \qquad (*)$$

Now, suppose that the vertices of Γ are $\{x_1, \ldots, x_{N+1}\}$. Start the SRW at x_1 and assume that $\Delta_j = \{x_1, \ldots, x_j\}$ for $j = 1, \ldots, N$.

Since $|\Delta_j| = j$, we have from our linear algebra fact and $(*)$ that

$$\det[\mathbb{G}^{\{x_1\}}] = \prod_{j=1}^{N} G_{\Delta_j}(x_j, x_j) = \prod_{j=1}^{N} \frac{N(j+1)}{j(N+1)} = \frac{N^N (N+1)!}{(N+1)^N N!} = \frac{N^N}{(N+1)^{N-1}}.$$

Since each of the $(N+1)$ vertices has degree N, we conclude

$$|\Omega| = \frac{\det[\mathcal{D}^{\{x_1\}}]}{\det[\mathbb{G}^{\{x_1\}}]} = \frac{N^N}{\frac{N^N}{(N+1)^{N-1}}} = (N+1)^{N-1}.$$

Thank you.