A random walk proof of Kirchhoff's matrix tree theorem

Michael J. Kozdron

University of Regina

http://stat.math.uregina.ca/∼kozdron/

Geometry of Random Walks and SLE

June 17, 2015

Based on joint work with Larissa Richards (Toronto) and Dan Stroock (MIT) with special thanks to Greg Lawler (Chicago) and Shlomo Sternberg (Harvard).

Preprint available from arXiv:1306.2059.

Some History

Kirchhoff's matrix tree theorem gives a formula for the number of spanning trees of a finite graph in terms of a matrix derived from that graph.

- 1880s: Gustav Kirchhoff was motivated to study spanning trees by problems arising from his work on electrical networks.
- 1979: Greg Lawler introduces the loop-erased walk in his Ph.D. thesis as a model of self-avoiding walk on the advice of Ed Nelson, his advisor: "Why don't you try erasing the loops of a random walk!"
- 1996: David Wilson used "cycle-popping" to prove an algorithm for generating 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.
- 1997-2010: In the book Probability on Trees and Networks by Russell Lyons with Yuval Peres, the cycle-popping proof of Wilson's algorithm is presented. They then use the cycle-popping proof to give a new proof of Cayley's formula for the number of spanning trees of a complete graph, but not MTT.
- 1999: Greg Lawler provides a new proof of Wilson's algorithm via Green's functions. (In an article in the Kesten Festschrift.) The article does not mention the MTT.

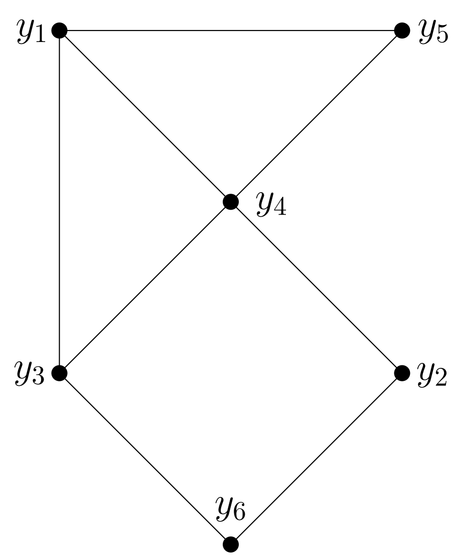
- 2010: Geoffrey Grimmett's book Probability on Graphs discusses Kirchhoff's work on electrical networks, but his proof of Wilson's algorithm uses cycle-popping.
- 2010: In Random Walk: A Modern Introduction by Greg Lawler and Vlada Limic, there is a chapter dedicated to loop-erased walk which includes his proof of Wilson's algorithm and discusses the fact that the MTT follows as an immediate corollary to his proof.
- 2011: Yves LeJan's Markov Paths, Loops, and Fields includes (essentially) this proof of Wilson's algorithm. Deducing Cayley's formula and the MTT left as exercises.
- These two books are for a specialized audience, and unfortunately, Lawler's proof of Wilson's algorithm is not widely known or immediately accessible, even among probabilists.
- As Wendelin Werner discussed on Monday in his report on news from the loop soup front, the idea of "adding loops" has proved to be very fruitful for studying SLE and more.
- 2013: Our original goal was to give an expository account of Lawler's proof. These ideas can be applied to deduce results for Markov processes. Lawler-Limic also deduce results for Markov processes, but not quite the same as ours.

Set-up

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

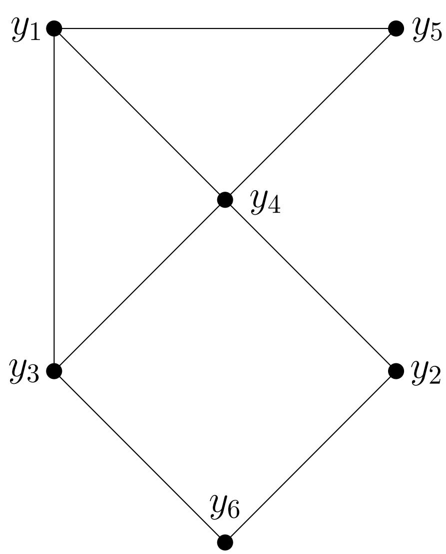
- undirected
- connected
- no multiple edges (easy to relax, but adds extra notation)

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



The Graph Laplacian Matrix

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



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

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.

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

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

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

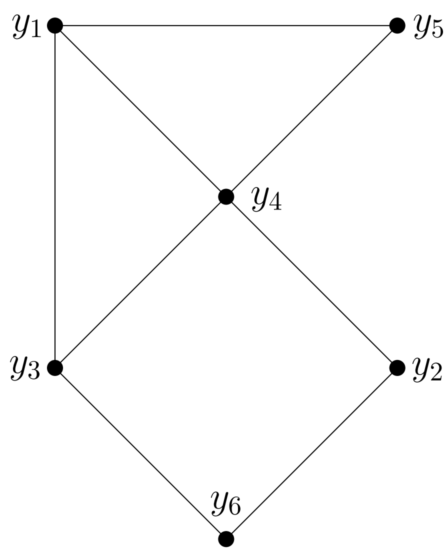
Practically, this is very hard to compute!

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

Example

This graph has 29 spanning trees.

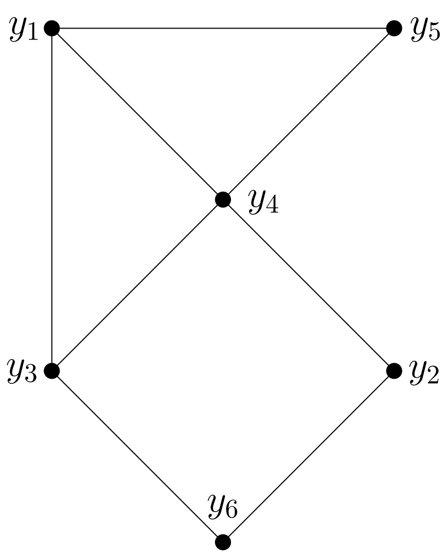
To see this, consider $\deg(y_4)$ in the spanning tree.



$\deg(y_4)$	# Spanning Trees
4	2
3	10
2	13
1	4
	29

Example (cont.)

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



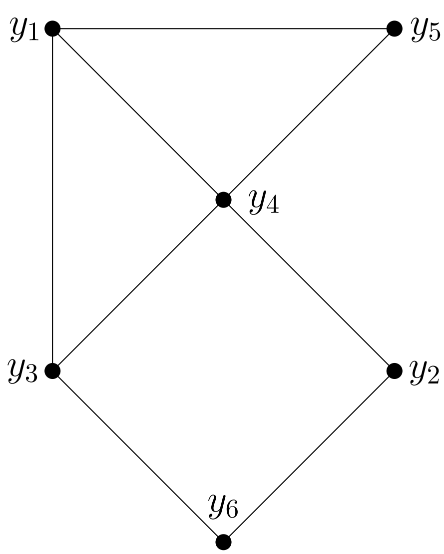
$$\mathcal{L}^{\{5\}} = \begin{array}{c c} & \begin{array}{ccccc} y_1 & y_2 & y_3 & y_4 & y_6 \end{array} \\ \begin{array}{c} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_6 \end{array} & \begin{bmatrix} 3 & -1 & -1 & -1 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ -1 & -1 & 4 & -1 & -1 \\ -1 & 0 & -1 & 3 & 0 \\ 0 & 0 & -1 & 0 & 2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

Random Walk on a Graph

Let $\{S_k,\ k = 0, 1, \cdots\}$ denote simple random walk on graph Γ which has transition probabilities

$$P\{S_1 = y_j \mid S_0 = y_i\} = p(i,j) = \begin{cases} \frac{1}{\deg(y_i)}, & \text{if } y_i \sim y_j \\ 0, & \text{else,} \end{cases}$$

i.e., each neighbour is equally likely to be chosen at the next step so that $p(i,j)$ is the (i,j)-entry of $\mathbb{P} = \mathcal{D}^{-1}\mathcal{A}$.



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

Random Walk on a Graph

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 Green's Function 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.

- 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.

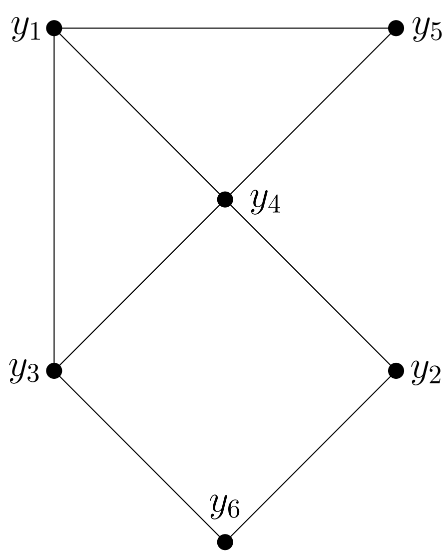
Clearly, this generates a spanning tree. We will prove that it is uniform among all possible spanning trees.

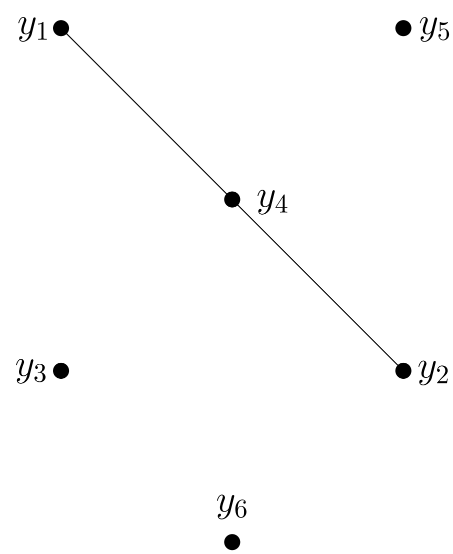
Example: Wilson's Algorithm on Γ

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.

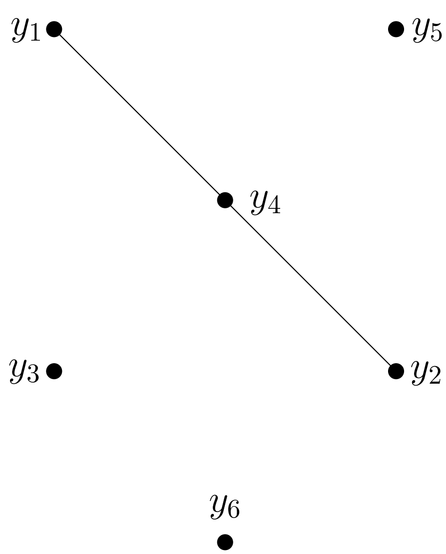


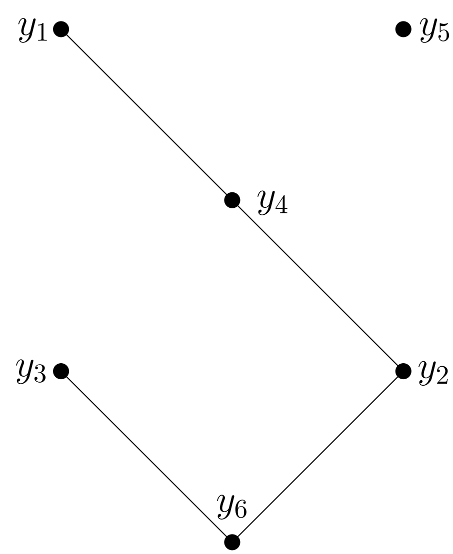


Example: Wilson's Algorithm on Γ

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.

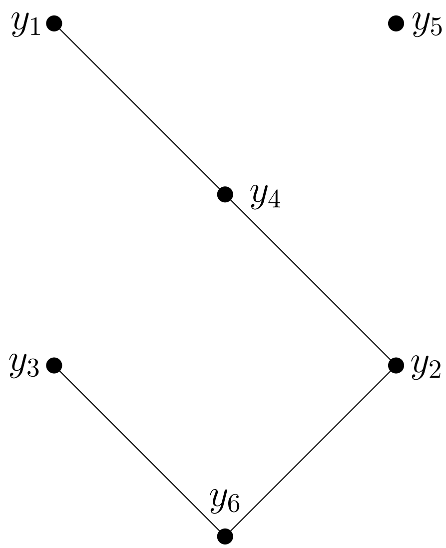




Example: Wilson's Algorithm on Γ

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
$\Delta_1 = [y_2, y_4, y_1]$, $\Delta_2 = [y_3, y_6, y_2]$, $\Delta_3 = [y_5, y_4]$.

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 $[x_1,\cdots,x_{K+1}]$, it must be the case 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 $j = 2, \cdots, K$.

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.

Understanding the Next Linear Algebra Slides

<u>Recall.</u> If $x, y \notin \Delta$, then $G_\Delta(x, y)$ is the expected number of visits to y by simple random walk on Γ starting at x before entering Δ.

<u>Exercise.</u>

If $x, y \notin \Delta$, then

$$G_\Delta(x,x)G_\Delta(y,y) - G_\Delta(x,y)G_\Delta(y,x) = G_\Delta(x,x)G_{\Delta\setminus\{x\}}(y,y),$$

i.e.,

$$\det \begin{bmatrix} G_\Delta(x,x) & G_\Delta(x,y) \\ G_\Delta(y,x) & G_\Delta(y,y) \end{bmatrix} = G_\Delta(x,x)G_{\Delta\setminus\{x\}}(y,y).$$

Split the number of visits to x by SRW starting at x into two pieces: those that occur before the first visit to y and those that occur after the first visit to y.

If $x, y, z \notin \Delta$, then

$$\det \begin{bmatrix} G_\Delta(x,x) & G_\Delta(x,y) & G_\Delta(x,z) \\ G_\Delta(y,x) & G_\Delta(y,y) & G_\Delta(y,z) \\ G_\Delta(z,x) & G_\Delta(z,y) & G_\Delta(z,z) \end{bmatrix} = G_\Delta(x,x)G_{\Delta\setminus\{x\}}(y,y)G_{\Delta\setminus\{x,y\}}(z,z).$$

A Linear Algebra Fact

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

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

Some Linear Algebra: Example

We illustrate how to use the notation of the linear algebra fact to do a computation.

Suppose that M is the non-degenerate 3×3 matrix

$$M = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 9/10 & 2/5 & -1/10 \\ 1/10 & -2/5 & -9/10 \\ -1/5 & 4/5 & -1/5 \end{bmatrix} \end{array}$$

so that $\det[M^{-1}] = \det[M]^{-1} = (4/5)^{-1} = 5/4$.

We will now calculate this determinant using the formula.

Let σ be any permutation of $\{1, 2, 3\}$, say $\{2, 3, 1\}$, so that

$\Delta_1 = \emptyset$, $\Delta_2 = \{\sigma(1)\} = \{2\}$, $\Delta_3 = \{\sigma(1), \sigma(2)\} = \{2, 3\}$. Hence,

$$(M^{\Delta_1})^{-1} = M^{-1} = \begin{array}{c c} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 1 & 0 & -1/2 \\ 1/4 & -1/4 & 1 \\ 0 & -1 & -1/2 \end{bmatrix} \end{array},$$

$$(M^{\Delta_2})^{-1} = \left(\begin{array}{c c} & \begin{array}{cc} 1 & 3 \end{array} \\ \begin{array}{c} 1 \\ 3 \end{array} & \begin{bmatrix} 9/10 & -1/10 \\ -1/5 & -1/5 \end{bmatrix} \end{array} \right)^{-1} = \begin{array}{c c} & \begin{array}{cc} 1 & 3 \end{array} \\ \begin{array}{c} 1 \\ 3 \end{array} & \begin{bmatrix} 1 & -1/2 \\ -1 & -9/2 \end{bmatrix} \end{array},$$

$$(M^{\Delta_3})^{-1} = \begin{array}{c c} & 1 \\ 1 & \begin{bmatrix} 10/9 \end{bmatrix} \end{array}$$

and so

$$\prod_{j=1}^{3} (M^{\Delta_j})^{-1}_{\sigma(j)\sigma(j)} = (M^{\Delta_1})^{-1}_{22} (M^{\Delta_2})^{-1}_{33} (M^{\Delta_3})^{-1}_{11} = -\frac{1}{4} \cdot -\frac{9}{2} \cdot \frac{10}{9} = \frac{5}{4}.$$

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 the linear algebra fact,

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

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

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

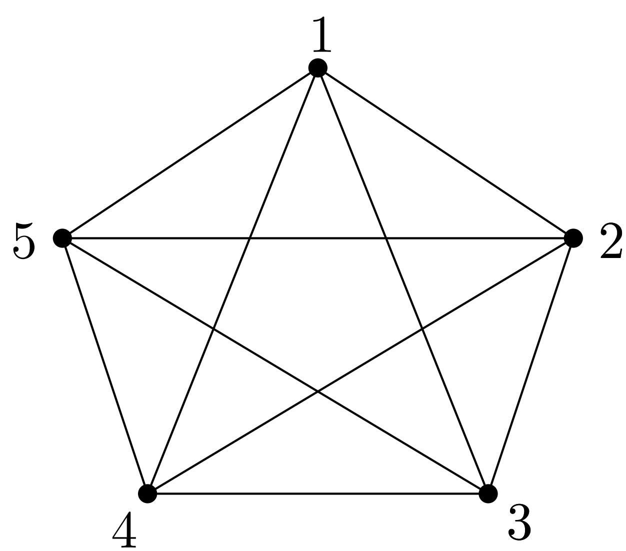
Since v was arbitrary, we conclude

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

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

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



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

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

Hence,

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

Application: Cayley's Formula

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

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

Application: Markov Chains

Note that this is a special case of a more general theorem for Markov processes.

Theorem (K-Richards-Stroock). If $\mathbb{P}$ is the transition matrix for an irreducible, aperiodic time-homogeneous Markov chain on $\{1, \ldots, N\}$, then its unique stationary probability distribution $\pi = (\pi_k,\ k = 1, \ldots, N)$ is given by

$$\pi_k = \frac{\det[(\mathbb{I} - \mathbb{P})^{\{k\}}]}{\sum_{j=1}^{N} \det[(\mathbb{I} - \mathbb{P})^{\{j\}}]} \qquad (*)$$

where $(\mathbb{I} - \mathbb{P})^{\{k\}}$ is obtained from $\mathbb{I} - \mathbb{P}$ by deleting row k and column k.

For a given state j, let ρ_j be the first time after 0 that the chain visits j; in other words, if the chain starts at state i, then ρ_j is the time of the first visit to j if $i \neq j$, whereas ρ_j is the time of the first return to j if $i = j$. If $\mathbf{P}_i$ is the distribution of the Markov chain assuming it starts at i, then

$$\mathbf{P}_i\{\rho_j \leq \rho_i\} = \frac{\det[(\mathbb{I} - \mathbb{P})^{\{j\}}]}{\det[(\mathbb{I} - \mathbb{P})^{\{i,j\}}]}. \qquad (**)$$

Example

Suppose that

$$\mathbb{P} = \begin{bmatrix} 3/4 & 0 & 1/4 \\ 1/8 & 1/8 & 3/4 \\ 1/12 & 1/4 & 2/3 \end{bmatrix}$$

which is the transition matrix for an irreducible, aperiodic Markov chain on $\{1, 2, 3\}$. Since

$$(\mathbb{I}-\mathbb{P})^{\{1\}} = \begin{bmatrix} 7/8 & -3/4 \\ -1/4 & 1/3 \end{bmatrix}, \ (\mathbb{I}-\mathbb{P})^{\{2\}} = \begin{bmatrix} 1/4 & -1/4 \\ -1/12 & 1/3 \end{bmatrix}, \ (\mathbb{I}-\mathbb{P})^{\{3\}} = \begin{bmatrix} 1/4 & 0 \\ -1/8 & 7/8 \end{bmatrix}$$

so that

$$\det[(\mathbb{I}-\mathbb{P})^{\{1\}}] = \frac{5}{48} = \frac{10}{96}, \ \det[(\mathbb{I}-\mathbb{P})^{\{2\}}] = \frac{1}{16} = \frac{6}{96}, \ \det[(\mathbb{I}-\mathbb{P})^{\{3\}}] = \frac{7}{32} = \frac{21}{96},$$

we see immediately from $(*)$ that

$$\pi_1 = \frac{10}{10+6+21} = \frac{10}{37}, \quad \pi_2 = \frac{6}{10+6+21} = \frac{6}{37}, \quad \pi_3 = \frac{21}{10+6+21} = \frac{21}{37}.$$

Example

Assuming that the chain starts in state 3, the probability that it visits state 2 before its first return to state 3 is

$$\mathbf{P}_3\{\rho_2 \le \rho_3\} = \frac{\det[(\mathbb{I}-\mathbb{P})^{\{2\}}]}{\det[(\mathbb{I}-\mathbb{P})^{\{2,3\}}]} = \frac{1/16}{1/4} = \frac{1}{4}.$$

Of course, we could have deduced this immediately. Observe from $\mathbb{P}$ that if the chain is in state 1, then it cannot move to state 2. Thus, starting in state 3, the only way for the chain to visit state 2 before its first return to state 3 is if $X_1 = 2$. This occurs with probability $p(3,2) = 1/4$.

However, it is more involved to directly compute the probability that the same chain visits state 1 before its first return to state 3. From $(**)$, however, the probability is easily found to be

$$\mathbf{P}_3\{\rho_1 \le \rho_3\} = \frac{\det[(\mathbb{I}-\mathbb{P})^{\{1\}}]}{\det[(\mathbb{I}-\mathbb{P})^{\{1,3\}}]} = \frac{5/48}{7/8} = \frac{5}{42}.$$

Thank you.