27 March 2023
MAT 2250

Coming back to the proof for Kirchhoff's theorem:

$EE^T = L$

F = E taken at the 1st row.

$FF^T = L_{1,1}$ ← the $(1,1)$ minor of L.

S = a subset with $n-1$ elements of $\{1, \dots, m\}$

F_S = the $(n-1)\times(n-1)$ matrix constructed by the $n-1$ columns in S.

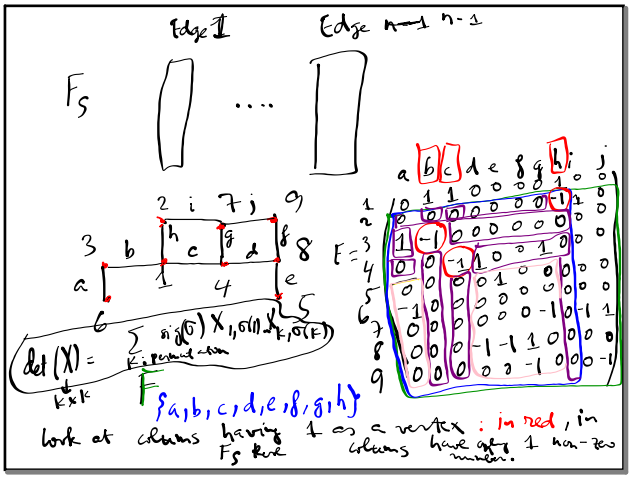
$$\det(L_{1,1}) = \sum_S \det(F_S)^2$$

(Cauchy-Binet's theorem).

(G connected) Each S gives us a spanning tree or $\det(F_S) = 0$ ↑ then $\det(F_S) = \pm 1$

Look first at edges in S which contains the vertex 1. Two possibilities:

* Case 1: there is no edge in S which contains the vertex 1.



Then entries in F_S on the same row as those non-zero numbers (in purple column) won't count toward the determinant of F_S, so we can take them out. (the same as taking out from F_S some of the columns & rows.

For example, for our example, we have taken out the rows 2, 3, 4, & the columns b, c, h from F_S → we are left with a 5×5 matrix.

Do again the same argument as for vertex 1, but now for the new vertices which we look at. (Q.E.D.) □

HW: at home, try to do practice with one explicit graph to understand this.

Come back to §7.3

Basis: Let $M = (S, \mathcal{U})$ be a matroid. (Remind $\mathcal{U} \subseteq 2^S$ = the set of all subsets of S, $|S| < \infty$)

A set $B \in \mathcal{U}$ satisfying some axioms.

is a basis for M if it is maximal w.r.t. the inclusion relation.

That is: $\nexists\, C \in \mathcal{U}$: $B \subseteq C$, $B \neq C$.

Lemma: If B_1 & B_2 are 2 bases of $M = (S, \mathcal{U})$, then

$$|B_1| = |B_2|.$$

Proof: If not, then assume for example that $|B_1| = m < |B_2|$.

Let $B' = \{b_1, \dots, b_{m+1}\} \subseteq B_2 \in \mathcal{U}$.

By axiom 2: $B' \in \mathcal{U}$.

By axiom 3, $\exists\, j$ so that

$B_1 \subseteq B_1 \cup \{b_j\} \in \mathcal{U}$

$\Rightarrow B_1$ is not maximal, a contradiction. (Q.E.D.)

So we can define

$$\text{rank}(M) = \#B,$$

where B is any basis for M.

$\text{rank}(M)$ is like the rank of a matrix (If M comes from a matrix).

In number theory, $\text{rank}(M)$ = is like the transcendental degree of an field extension

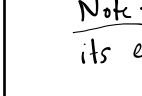
Matroid from a graph: Let $G = (V, E)$ be a graph.

Define $G = (V, E)$

$$W = \{ H = (V, A) : H \text{ is a forest} \}$$

(= disjoint union of trees)

Note: A can be identified with its edges.



→ can be identified with I, II, III, (when the vertices are fixed.)

Because the edges come with the vertices. Also, in a tree, between 2 vertices, there is at most 1 path connecting them.

Observation: A maximal element in $\mathcal{W}$ is a spanning tree (if G is connected) or more generally, a spanning forest (if G is not connected).

Proof:

Spanning tree = a tree which connects every vertex (if G is connected)

Spanning forest = a union of spanning trees, one for each connected component of G.

If let H_0 be a maximal element.

If it is not spanning, then $\exists$ a vertex v in G which is not covered by H_0.

v is in 1 connected component of G, say G_1, & let $H_1 = H_0|_{G_1}$ $\Rightarrow$ it is a tree in G_1.

G_1 is connected, $H_1 \subseteq G_1$, $v \in G_1$ $\Rightarrow$ we can make a bigger tree H_1' which connects v & H_1.

Replace H_1 by H_1' in H_0 $\Rightarrow$ we get a bigger forest H_0', so H_0 is not maximal. (a contradiction).

Thm 7.12: If $G = (V, E)$ is a graph, $M = (E, \mathcal{W})$ is a matroid.

Proof:

Check 3 axioms:

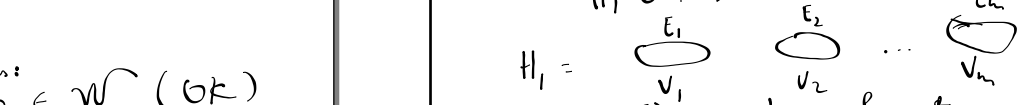
Axiom 1: $\emptyset \in \mathcal{W}$ (OK)

Axiom 2: If $H \in \mathcal{W}$, $H_1 \subseteq H$ $\Rightarrow$ H is a forest, hence H_1 is also a forest, $\Rightarrow H_1 \in \mathcal{W}$.

Axiom 3: If $H_1, H_2 \in \mathcal{W}$ &

$|H_2| = |H_1| + 1 \Rightarrow \exists$ an edge $e \in H_2$ such that

$H_1 \cup \{e\}$ is a forest.

$H_1 =$ 

each (V_j, E_j) is a tree, & they are disjoint. $|E_j| = |V_j| - 1$.

$$|H_1| = \sum_{j=1}^{m} (|V_j| - 1)$$

Since $|H_2| = |H_1| + 1 \Rightarrow \exists$ e in H_2 which connects 2 different V_i & V_j.

If not, $\exists\, e \in H_2$, $\forall\, V_e \in \{V_1, \dots, V_m\}$ such that

$e \notin V_j$,

e connect 2 vertices in V_j.

Cannot happen because H_2 also has two vertices in V_j.

$\Rightarrow |H_2| \leq \sum_j |V_j| - 1$

$= |H_1|$, contradiction.

Another case is that e does not have any vertex in $V_1 \cup \dots \cup V_m$.

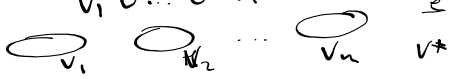
In both cases:

Case 1:



$\Rightarrow H_1 \cup \{e\}$ is still a forest.

Case 2: e does not have an edge in $V_1 \cup \dots \cup V_m$



$\Rightarrow H_1 \cup \{e\}$ is still a forest.

(Q. E. D.)

Summary: $G = (V, E)$

$M = (S, \mathcal{W})$ is a matroid.

$\mathcal{W}$: set of all forests of G

Corollary: If H is a basis for M

$\Rightarrow H$ is a spanning forest for G.

& if H is a basis with minimal weight for M $\Rightarrow$ we get a spanning forest with minimal weight for G. (G connected $\Rightarrow$ we find a root to visit all vertices with minimal cost.)

Greedy algorithm for to find a minimal basis for a matroid:

$M = (S, \mathcal{U})$ a matroid.

A weight on M is a way to assign to each element in S a real number, i.e. a function $w: S \to \mathbb{R}$.

If $A \in \mathcal{U} \Rightarrow$

$w(A) := \sum_{x \in A} w(x)$.

Algorithm:

$A_0 = \emptyset \in \mathcal{U}$

Step it1.: Assume:
$A_i = \{a_1, \dots, a_i\} \in \mathcal{U}$ is already constructed.
Let $X_i = \{x \in S \setminus A_i : A_i \cup \{x\} \in \mathcal{U}\}$ → eligible set
If $X_i = \emptyset \Rightarrow$ stop.
If $X_i \neq \emptyset \Rightarrow$ choose: $a_{i+1} = \underset{x \in X_i}{\operatorname{argmin}}\ w(x)$.
Update $A_{i+1} = A_i \cup \{a_{i+1}\}$. Repeat step it2.

Theorem 7.13: The greedy algorithm will find a minimal weight basis B for $M = (S, \mathcal{U})$. That is, if B' is another basis, then: $w(B') \geq w(B)$.

Proof: Let $B = \{a_1, \dots, a_n\}$ is the final set constructed by the greedy algorithm.
Then B is a basis (obvious). If not, then we can still add one more $x \in S$ to $A_n = \{a_1, \dots, a_n\}$ so that $A_n \cup \{x\} \in \mathcal{U}$.

Then by our algorithm we will add at least 1 more point to $A_n \Rightarrow B$ is not the final set.
$B' = \{b_1, \dots, b_n\}$ be another basis.
We will show that $w(B') \geq w(B)$.
Claim 1: $w(a_1) \leq w(a_2) \leq \dots \leq w(a_n)$.
Why? To show for $w(a_1) \leq w(a_2)$:
$\{a_1, a_2\} \subseteq \{a_1, \dots, a_n\} \in \mathcal{U}$
$\Rightarrow$ by axiom 2: $\{a_1, a_2\} \in \mathcal{U}$.
$\Rightarrow a_2 \in X_1 = \{x \in S \setminus A_1 : A_1 \cup \{x\} \in \mathcal{U}\}$.
$\{a_1\}, \{a_2\} \subseteq \{a_1, \dots, a_n\} \in \mathcal{U} \Rightarrow \{a_1\}, \{x\} \in \mathcal{U}$.

$a_1 = \underset{\substack{x \in S \\ \text{s.t.} \{x\} \in \mathcal{U}}}{\operatorname{argmin}}\ w(x)$
Now $\{a_1\}, \{a_2\} \in \mathcal{U} \Rightarrow$
$w(a_1) \leq w(a_2)$.
How to show that $w(a_2) \leq w(a_3)$?
Similarly: $\{a_1, a_2\}, \{a_1, a_3\} \in \mathcal{U} \Rightarrow a_2, a_3 \in X_1$ & we chose: $a_2 = \underset{x \in X_1}{\operatorname{argmin}}\ w(x)$
Similarly: for $w(a_3) \leq w(a_4) \Rightarrow$ look at $\{a_1, a_2, a_3\}$ & $\{a_1, a_2, a_4\}$...

We can order permute elements in $B' = \{b_1, \dots, b_n\}$ s.t.

$w(b_1) \le w(b_2) \le \dots \le w(b_n)$.

We will show that

$w(a_j) \le w(b_j) \quad \forall j = 1, \dots, n$.

(Then $w(B) = \sum w(a_j) \le \sum w(b_j) = w(B')$.)

$j = 1$: $\{a_1\}, \{b_1\} \in \mathcal{U}$

& $a_1 = \underset{x \in S}{\operatorname{argmin}}\, w(x)$

$\{x\} \in \mathcal{U} \Rightarrow w(a_1) \le w(b_1)$.

Induction:

Assume that $w(a_i) \le w(b_i)$

$\forall i = 1, \dots, j$.

We will show that $w(a_{j+1}) \le w(b_{j+1})$.

We have $\{b_1, \dots, b_j, b_{j+1}\} \in \mathcal{U}$

(by axiom 2)

So by axiom 3, $\exists\, k \in \{1, \dots, j+1\}$

such that $A_j = \{a_1, \dots, a_j\} \cup \{b_k\}$

$\in \mathcal{U}$. $b_k \in X_j$

$\Rightarrow$ Since $a_{j+1} = \underset{x \in X_j}{\operatorname{argmin}}\, w(x)$ $\Rightarrow w(a_{j+1}) \le w(b_k)$

We ordered that

$w(b_1) \le w(b_2) \le \dots \le w(b_j) \le \dots$

& $k \le j+1 \Rightarrow w(b_k) \le w(b_{j+1})$.

$\Rightarrow$ $w(a_{j+1}) \le w(b_k) \le w(b_{j+1})$

(Q.E.D.)

Complexity: We need to order the weights of the edges $\{e_1, \dots, e_q\} \in E$ so that

$w(e_1) \le \dots \le w(e_q)$.

One very naive way is like this:

Start with $w(e_1)$. Then compare $w(e_1)$ & $w(e_2)$, change the order if not. Then compare $w(e_3)$ with $w(e_1), w(e_2)$ ...

How many operations?

Step 1: $w(e_1)$ v.s $w(e_2)$: 1 comparison

Step 2: $(w(e_1), w(e_2))$ v.s $w(e_3)$

2 comparisons.

Step $q-1$: $(w(e_1), \dots w(e_{q-1}))$ v.s $w(e_q)$

$\Rightarrow$ $q-1$ comparisons

$\Rightarrow$ $1 + 2 + \dots + (q-1) = O(q^2)$.

(Q.E.D.)

There are better ways to sort. Look at Sort Algorithm on Wikipedia.