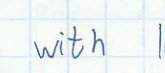
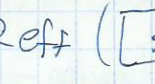
W3D3
11/11/14

Rand. Spanning Trees

From last time - μ prob. meas. on paths from $a \to z$

Then $\theta(\vec{e}) = \#\mathbb{E}[\text{net traversals of } e]$ is a flow with $\|\theta\| = 1$.



Example R_{eff} (). $\theta(\vec{e}) \le \mathbb{P}(e \text{ was traversed}) \le \frac{c}{k}$ if e is of euc. dist. k from ρ. $\mathcal{E}(\theta) \le c \sum_{k=1}^{n} k \frac{c}{k^2} = c \log n$

In $\mathbb{Z}^3$, $|\theta(e)| \le \frac{c}{k^2} \Rightarrow \mathcal{E}(\theta) < c \ (\forall n)$, so $\mathbb{Z}^3$ is transient while $\mathbb{Z}^2$ is recurrent.

Wilson's Algorithm - For drawing UST

Given a finite graph G with (positive) non-neg. weights $\{c_e\}_{e \in E}$.

The weighted UST of G is a prob. meas. on G's span. trees s.t. $\mathbb{P}(T=t) = \frac{\prod_{e \in t} c_e}{Z}$ where $Z = \sum_{t \in T} \prod_{e \in t} c_e$.

Recall, given a finite path $(x_0, \dots x_\ell)$ we denote by $LE(P)$ the loop erasure, the path obtained by P by erasing loops as they're created.

Wilson's Alg: choose init. vertex u, put $T(0) = \{u\}$.

given $T(i)$ which isn't spanning, choose $x \notin T(i)$, run a weighted rand. walk until it hits $T(i)$ and denote the result by P. Then $T(i+1) = T(i) \cup LE(P)$.

choose - we choose arbitrarily, not neccessarily at rand. Note that this works for a recurrent graph too.

Thm. Resulting tree is weighted UST.

Cor. given $(x,y) \in E$, $\mathbb{P}(xy \in UST) = R_{eff}(x \leftrightarrow y; \{c_e\}) \cdot c_{xy}$

Pf. start $T(0) = \{y\}$ and choose the 1st vertex to be x.

$(x,y) \in UST \leftrightarrow P_{-1} = (x,y)$. Look at the last part of P, going from x (without returning to it) to y.

$\mathbb{P}((x,y) \in UST) = \mathbb{P}((x,y) \text{ is in the path} \mid \tau_y < \tau_x) =$

$\mathbb{P}((x,y) \text{ is chosen first} \mid \tau_y < \tau_x) = \frac{\mathbb{P}((x,y) \text{ is chosen first and } \tau_y < \tau_x)}{\mathbb{P}(\tau_y < \tau_x)} =$

(last edge of P)

$$\frac{\mathbb{P}((x,y) \text{ is chosen first})}{\mathbb{P}(\tau_y < \tau_x)} = R_{\text{eff}}(x \leftrightarrow y) C_{xy}.$$

Another cor. The dis. of a loop-erased path from x to y on any recurrant network = The dist of the LEP from y to x.

Proof of Wilson's Alg. Put in every $x \in V$ an infinite stack $\{S_i^x\}_{i=1}^{\infty}$ of arrows drawn i.i.d. with the weighted transition probs., $\left\{\frac{C_{xy}}{\pi_x}\right\}_{y \in V}$. Now the walker's path is determined by the drawn stack - a starting position given S the walker is deterministic: pop a direction from the stack to the next dir. u the init. vertex gets the empty stack

At any time, the top items of the stacks determine a directed graph with edges (x, y) where y is at the top of S^x. It has $n-1$ edges so if there are no directed cycles it is a tree rooted at u. Once we get to this we'll stop. O/W arbitrarily we choose a cycle and pop it, (pop all vertices in it), then repeat.

The tree eminating to u grows larger with every pop with some positive prob. so it stops with prob. 1. We claim that the resulting (undirected) tree is the UST and that this proc. is Wilson's algorithm.

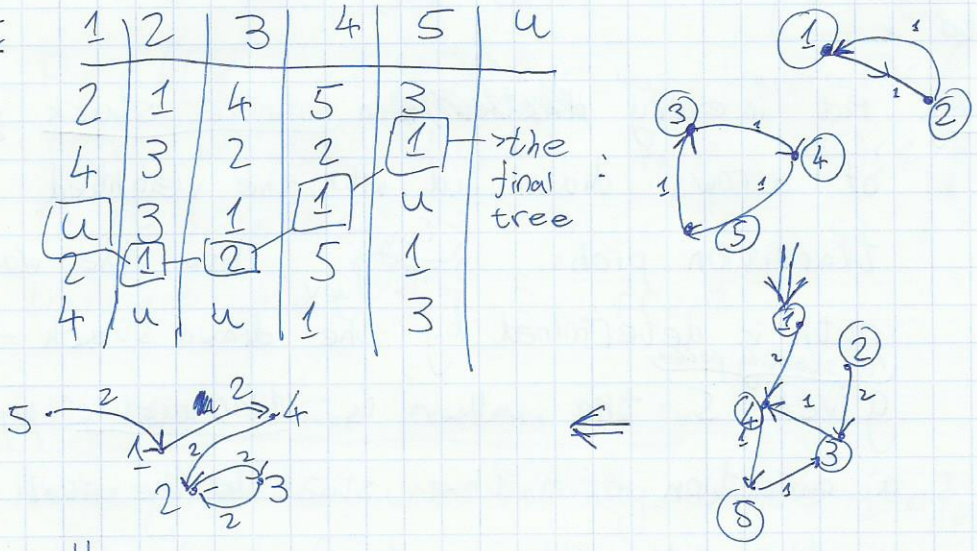
For the 2nd part it's enough to show that our popping choice won't matter. We keep track on the location in the stacks of popped edges - we say (x, S_i^x) has color i. A colorful cycle is a cycle with

(not necc. the same color.)

all edges are colored like this. The init. directed graph has all edges colored 1, and each colorful cycle can be popped once (because next time we'll have another color).

Example:

1	2	3	4	5	u
2	1	4	5	3	
4	3	2	2	1	→ the final tree
u	3	1	1	u	
2	1	2	5	1	
4	u	u	1	3	



We claim that the final tree is indep. of popping order - if there was a lower-color tree we would have stopped there.

claim the final ~~then~~ tree is indep. of popping order - either every order pops infinitely many cycles, or every order pops the same finite set of cycles (colorful), resulting in the same colorful directed spanning tree.

pf: we'll show that if C is any colorful cycle that can be popped (in some seq. of cycle poppings) $C_1 \dots C_n = C$ then ~~arrange~~ if $C' \neq C_i$ can be popped first, either $C' = C$ or C can still be popped after popping C' first.

So, if there are ∞ many it won't stop and if $<\infty$ they'll all be popped, resulting in the same tree.

If $C' \cap (C_1 \cup C_2 \cup \dots \cup C_n) = \emptyset$ then it's obvious - pop $C', C_1, C_2, \dots, C_n$. O/W - Let C_k be the 1st colourful cycle that has a vertex in common $x \in C' \cap C_k$. (C' as a vertex set)

C' edges have colour 1. From our choice of k, $x \notin \{C_1 \cup \dots \cup C_{k-1}\}$ so the edge from x in C_k as the same color in C_k - it points in the same direction. This is true for any such x, so $C' = C_k$.

So, we can just pop $C' = C_k, \underbrace{C_1, C_2, \dots C_{k-1}}_{\text{disjoint from } C_k}, C_{k+1}, \dots C_n = C_r$ or $n = k$ so $C_n = C = C'$.

Pf. of thm.

LE(P) just pops the cycles in some order, so our description matches Wilson's Algorithm. In particular, our choices won't affect the resulting tree. Each $T(i)$ is a directed colorful tree pointing at u, so they'll never be popped again. By prev. lemma given the stacks, all arbitrary choices will result in the same tree. In particular, for any choice of u, for all other choices will not affect the distribution. We'll prove it's the UST and therefore indep. of our choice of u as well.

Given stacks (with finite poppable cycles) define a finite set $\mathcal{O}$ of colorful-poppable cycles above a non-colored spanning tree T_N (T's colors are recovered from $\mathcal{O}$). $P(\text{the result is } t) = \sum_{\mathcal{O}} P(\mathcal{O}) \left(\prod_{e \in T} c_e\right) \alpha$

finite set of poppable colored cycles above T

origin of e

const.

$\prod_{e \in T} c_e$ - indep. of u, we get the UST.