CS224w:
Social and Information Network Analysis

Assignment Submission Fill in and include this cover sheet with each of your assignments. Assignments are due at 9:00 am. All students (SCPD and non-SCPD) must submit their homeworks via GradeScope (`http://www.gradescope.com`). Students can typeset or scan their homeworks. Make sure that you answer each question on a separate page. That is, one answer per page regardless of the answer length. Students also need to upload their code at `http://snap.stanford.edu/submit`. Put all the code for a single question into a single file and upload it. Please do not put any code in your GradeScope submissions.

Late Day Policy Each student will have a total of *two* free late periods. *One late period expires at the start of each class.* (Homeworks are usually due on Thursdays, which means the first late periods expires on the following Tuesday at 9:00am.) Once these late periods are exhausted, any assignments turned in late will be penalized 50% per late period. However, no assignment will be accepted more than *one* late period after its due date.

Honor Code We strongly encourage students to form study groups. Students may discuss and work on homework problems in groups. However, each student must write down their solutions independently i.e., each student must understand the solution well enough in order to reconstruct it by him/herself. Students should clearly mention the names of all the other students who were part of their discussion group. Using code or solutions obtained from the web (github/google/previous year solutions etc.) is considered an honor code violation. We check all the submissions for plagiarism. We take the honor code very seriously and expect students to do the same.

Your name: Devon Zuegel
Email: devonz@stanford.edu **SUID:** 005798092

Discussion Group: _____

I acknowledge and accept the Honor Code.

(Signed) Devon K Zuegel

Devon Zuegel // 3 Nov 2015 // CS 224W

Problem Set 3

Spoke with John Luttig and Ilan Goodman while working on this pset.

Problem 1 (10 points)

Parts A & B

Threshold Model
- Each individual i has a threshold t_i
- If there are $\geq t_i$ individuals that are rioting, then i will join
- implicitly assumes that each individual has knowledge of all others
- small threshold → innovators/early adopters
  large ——— → laggards/late adopters

Problem Description
- mob of n individuals
- histogram of thresholds $N = [N_0, \ldots, N_{n-1}]$ where N_i expresses the # of individuals that have threshold i

(a) Want: find conditions s/t individuals w/ threshold t become active

Solution: $(A_0, A_1, \ldots, A_l) \longrightarrow A_n := \sum_{i=0}^{n} N_i > n$

(b) Want: Expression for the final # of rioters for a given histogram N.

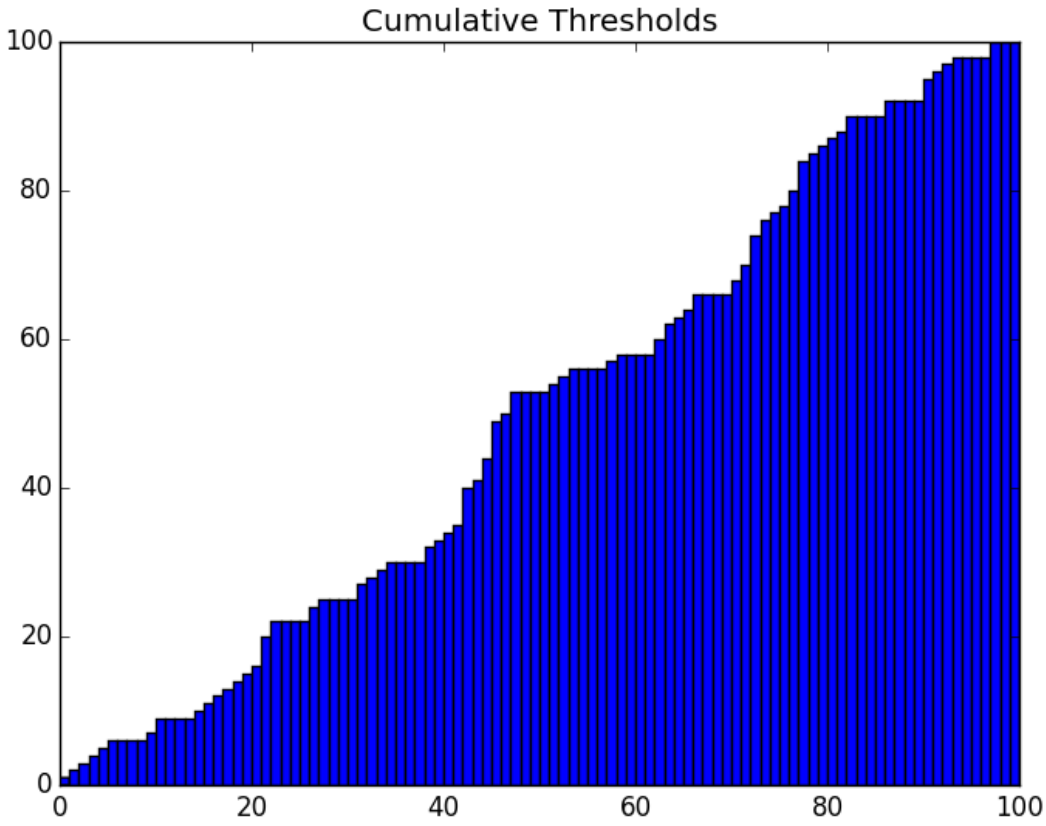
Setup:
- conditions = array of conditions $[A_0, A_1, \ldots, A_l]$ from a
- t = index of last condition in conditions that holds with all preceeding conditions also holding
- conditions[0:t] = longest possible run of conditions that hold, from 0 to t, the condition that precedes the first unmet condition

$$\text{numRioters}_{\text{final}} = \boxed{\sum_{i=0}^{t} N_i}$$

Part C

We would have 45 cumulative rioters. Output from `1c.py`:

```
Cumulative total # of rioters would be: 45
```



Problem 2 (40 points)

Parts A, B, and C

- Erdös-Renyi graph on n nodes, prob. p that there's a friendship edge between any two nodes
- Each node's threshold is a Bernoulli random var. w/ prob. $1-q_0$

Round 0: Each node riots or not with prob. q_0 (eg. threshold 0)
Rounds 1-on: Each inactive node in prev. round is activated if any of its neighbors is active
↳ A_r = set of nodes activated during round r

- all of these random variables are indep.

ⓐ a = "only specific subset of nodes S is active during round 0"

$P(a) = P(\underbrace{\text{each node} \in S \text{ is active}}_{a_1}) \cdot P(\underbrace{\text{each node} \notin S \text{ is inactive}}_{a_2})$

$P(a_1) = q_0^{|S|}$

$P(a_2) = (1-q_0)^{n-|S|}$

$$\boxed{P(a) = q_0^{|S|}(1-q_0)^{n-|S|}}$$

ⓑ b = "$|A_0| = k$ for some integer $0 \le k \le n$"

Consider our subset S from part a. Say our only constraint is that $|S| = k$. The number of unique subsets from the original set is: $\binom{n}{k} = {}_nC_k$

Thus there are $\binom{n}{k}$ possible ways to make statement b hold, and each individual way has $P(a)$ prob. of occurring. Thus:

$$P(b) = \binom{n}{k} P(a) = \boxed{\binom{n}{k} q_0^k (1-q_0)^{n-k}}$$

ⓒ $A_{[r]} = A_0 \cup A_1 \cup \dots \cup A_r \rightarrow$ all active nodes up thru round r

Want: 1) prob. of a node x belonging in $A_{[r]}$ in terms of $q_k = P(x \in A_k)$ for $k \le r$ ⟶ name this event C_1

2) prob. of $x \notin A_{[r]}$ in terms of $\{q_k\}_{k \le r}$ ⟶ C_2

$$P(C_1) = P(x \in A_{[r]}) = \sum_{k=0}^{r} P(x \in A_k) = \boxed{\sum_{k=0}^{r} a_k}$$

$$P(x \notin A_{[r]}) = 1 - P(x \in A_{[r]})$$

$$= \boxed{1 - \sum_{k=0}^{r} a_k}$$

Parts D, and E

(d) $E(x,S)$ = event that node x has ≥ 1 edge in set S

Want: $P(E(x,S))$ for some specific set S (non-random)

- $n(n-1)$ possible friendship edges
- p prob. that any given edge exists

Rephrased: What is the prob. that there's ≥ 1 edge btwn some node x (not in set S) and one of the nodes in S?

$S = \{s_0, s_1, \ldots, s_m\} \longrightarrow |S| = m$

$P(x \text{ is connected to } s_i \text{ where } 0 \leq i \leq m) = p$

$P(x \text{ ——//—— exactly one node in } S) = p \cdot (1-p)^{|S|-1}$

$$P(E(x,S)) = \sum_{i=1}^{m} p^i (1-p)^{m-i} = 1 - (1-p)^{|S|}$$

$= 1 - P(x \text{ not connected to } S)$

(e) Want: Expression for event $x \in A_r$ for $r \geq 1$ as intersection of 2 events

- Node x becomes activated in round r ($x \in A_r$) if:
  - x was previously not activated
  - one or more of x's friends was activated in round $r-1$

$P(x \text{ hasn't been activated yet}) = P(x \notin A_{[r-1]}) = 1 - \sum_{k=0}^{r-1} a_k$

$P(E(x, A_{[r-1]})) = \sum_{i=1}^{m} p^i (1-p)^{m-i}$ where $m = |A_{[r-1]}|$

$E_1(x,S)$ = event that $x \notin S$

$E_2(x,S)$ = event that node x has ≥ 1 edge connecting into S

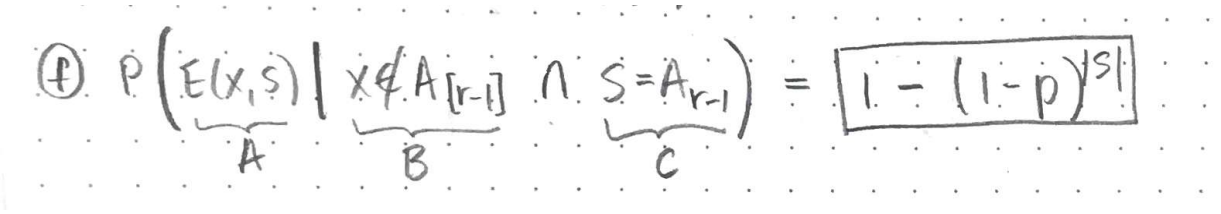
$E_3(x,S)$ = event that $x \in S$

$$E_3(x, A_r) = E_1(x, A_{[r-1]}) \cap E_2(x, A_{[r-1]}) \quad \text{for } r \geq 1$$

$$P(x \in A_r) = P(x \notin A_{[r-1]} \cap E(x, A_{[r-1]}))$$

Part F

Basically we're looking for `1 - P('x is not connected to S and was not activated in rounds 0-r')`:



Part G

Want:

- q_r, the probability that a give node riots in round `r`

Known:

- **NOTE:** Latex was doing something weird with the subscripts, so there are a few occurrences where I wrote $A[r-1]$ to mean $A_{[r-1]}$.
- Bayes' Rule:
  - $$P(X \mid Y) = \frac{P(Y \mid X) \cdot P(X)}{P(Y)}$$
  - $$P(X \mid YZ) = \frac{P(A \mid C) \cdot P(B \mid AC)}{P(B \mid C)}$$
- $$P\Big(x \in A_r\Big) = P\Bigg(x \in A_{[r-1]} \ \cap E(x, A[r-1])\Bigg)$$
- $$P\Big(E(x,S) \mid x \notin A_{[r]} \cap S = A_r\Big) = \boxed{1-(1-p)^{|S|}}$$
  $$= \sum_k^{|S|} \left(p^k (1-p)^{|S|-k} \binom{|S|}{k} \right)$$
- $$P\Big(X \notin A_{[r]}\Big) = 1 - \sum_k^r q_k$$
- $$P\Big(E(x,S)\Big) = \sum_k^{|S|} p^i (1-p)^{|S|-i}$$
  $$= \boxed{1-(1-p)^{|S|}}$$

- $$P\Big(S = A_{[r-1]} \mid x \notin A[r-1]\Big) = P(Z \mid Y)$$
- $$\begin{aligned} X &= E(x, S) \\ Y &= \text{event that } x \notin A[r-1] \\ Z &= \text{event that } S = A_{r-1} \end{aligned}$$

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A) * P(B|A)}{P(B)}$$

-
- `n` = # of nodes in the entire graph
- We only have to consider the event that `x` is connected to a node activated in the last round `r - 1`, because if it were connected to a node activated in any earlier round then it would be activated too (a.k.a. `x` $\in$ $A_{[r-1]}$).

Solution:

$$\begin{aligned} q_r &= P\Big(x \notin A[r-1] \cap E(x, A_{r-1})\Big) \\ &= P(A \cap B) = P(B \mid A) \cdot P(A) \qquad \texttt{\# Bayes Rule} \end{aligned}$$

$$= P\Big(E(x, A_{r-1}) \mid x \notin A[r-1]\Big) \cdot P\Big(x \notin A[r-1]\Big)$$

$$= P\Big(E(x, A_{r-1}) \mid x \notin A[r-1]\Big) \cdot \Big(1 - \sum_{k}^{r-1} q_k\Big)$$

@ $q_r = P(x \in A_r) = P(x \notin A_{[r-1]} \cap E(x, A_{r-1}))$

$= \sum_{S \subseteq [n] \setminus \{x\}} P(\underbrace{x \notin A_{[r-1]}}_{C} \cap \underbrace{E(x,S)}_{A} \cap \underbrace{S = A_{r-1}}_{B})$

we can sum these probs. b/c the events are independent.

BAYES' RULE

$P(ABC) = P(A|BC) \cdot P(BC) = P(A|BC) \cdot P(B|C) \cdot P(C)$

$q_r = \sum_{S \subseteq [n] \setminus \{x\}} \Big(P(E(x,S) \mid x \notin A_{[r-1]} \cap S = A_{r-1}) \cdot P(S = A_{r-1} \mid x \notin A_{r-1}) \cdot P(x \notin A_{[r-1]}) \Big)$

$\approx \sum_{S \subseteq [n] \setminus \{x\}} \Big(\big[1-(1-p)^{|S|}\big] \cdot \big[\prod_{\text{yes}} P(y \in A_r) \cdot \prod_{y \notin S \cup \{x\}} P(y \notin A_r)\big] \cdot \big[1 - \sum_{k=0}^{r-1} q_k\big] \Big)$

This last part of the expression doesn't depend on S, so we can pull it out of the summation.

$\approx \Big[\sum_{S \subseteq [n] \setminus \{x\}} (1-(1-p)^{|S|}) \Big(\prod_{\text{yes}} P(y \in A_r)\Big)\Big(\prod_{y \notin S \cup \{x\}} P(y \notin A_r)\Big)\Big] \cdot \big[1 - \sum_{k=0}^{r-1} q_k\big]$

This last part is equiv. to the last part of the expression we are trying to prove the whole expression here is equivalent to. Now let's drop that common last part to prove the first parts of each expression are equiv.

Want to prove: $1-(1-pq_{r-1})^{n-1} \approx \sum_{S \subseteq [n] \setminus \{x\}} \Big((1-(1-p)^{|S|}\, q_r^{|S|}\, (1-q_r)^{n-|S|-1} \Big)$ → n = total #

$n-1$ bc we're choosing subsets from $[n]$ that exclude node x

$\approx \sum_{k=0}^{n-1} \Big((1-(1-p)^k)\, q_r^k\, (1-q_r)^{n-k-1} \binom{n-1}{k} \Big)$

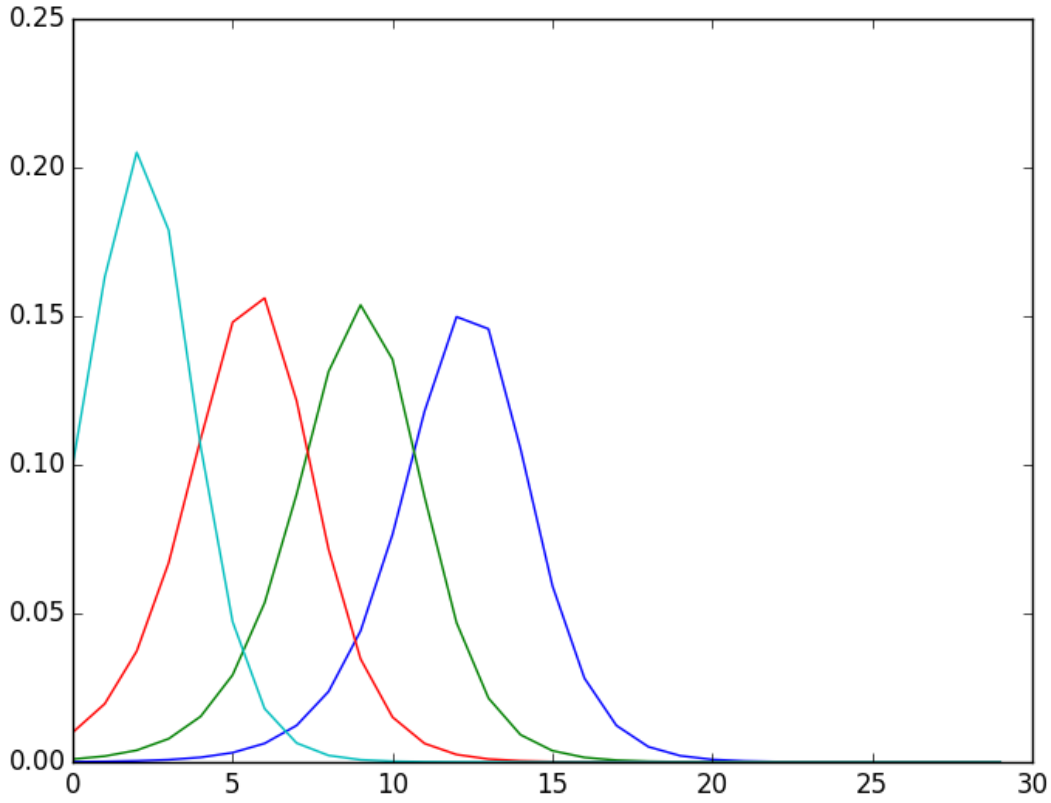
$\approx \sum_{k=0}^{n-1} \Big[\binom{n-1}{k} q_r^k (1-q_r)^{n-k-1} \Big] - \sum_{k=0}^{n-1} \Big[(1-p)^k \binom{n-1}{k} q_r^k (1-q_r)^{n-k-1} \Big]$

$(q_r + (1-q_r))^m = \sum [(1-p)q]^k (1-q_r)^{n-k} \binom{n-1}{k}$

1^m

$\Big(1 = (q_r - pq_r + 1 - q_r)^{n-1}\Big)$

Part i



Color	q_0 value
blue	0.0001
green	0.001
red	0.01
cyan	0.1

Each curve follows the same basic up-and-then-down shape. This makes sense, because at the beginning, there are few activated nodes. Then that number grows exponentially for several iterations because neighbor nodes activated in each subsequent round expand the frontier to exponentially more neighbor nodes. However, at some point this drops off as there are no other nodes in the connected component to activate. As the number of iterations goes to infinity, each marginal iteration adds no new nodes.

It makes sense that our curve with the highest q_0 starts at the highest point and then peaks and drops off the soonest. When the q_0 is high, the initial number of activated nodes is highest, and activation spread quickly through the graph. Conversely, it makes sense that the curve with the lowest q_0 starts at the lowest point, with very few initially activated nodes. Then, it rises steadily and peaks latest, since the others had a "head start" on it.

Part ii

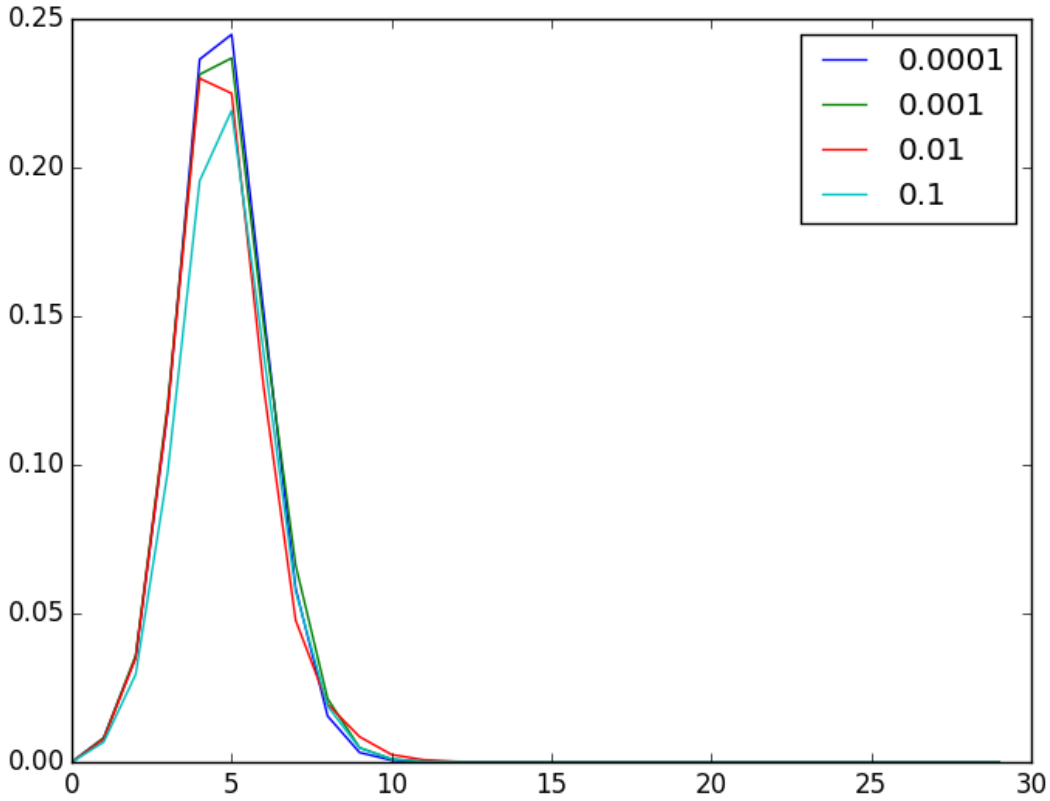
```
q0 = 0.0001    =>    Q29 = 0.796797231978
q0 = 0.001     =>    Q29 = 0.79710527594
q0 = 0.01      =>    Q29 = 0.800156416064
q0 = 0.1       =>    Q29 = 0.828251158185
```

The Q_{29} value for a particular starting q_0 is the sum of the probabilities that a given node `x` will be activated in round `0`, `1`, ..., `29`. As we can see from the plot in part i, the probability that `x` will be activated in round `i` goes to `0` as `i` $\rightarrow \infty$. This occurs well before `i = 30` for all four starting values of q_0.

For all intents and purposes, we can sav that the probability that `x` will be activated in round `i > 29` is zero. Thus, we can see that Q_{29} **represents the probability that `x` will be activated during the cascade**

Q_{29} is well under `1.00`, which means that a random node `x` has approximately a `20%` chance of never being activated. This can only happen if the graph is unconnected. Otherwise, `x` would eventually be reached by an expansion of the frontier.

Part iii

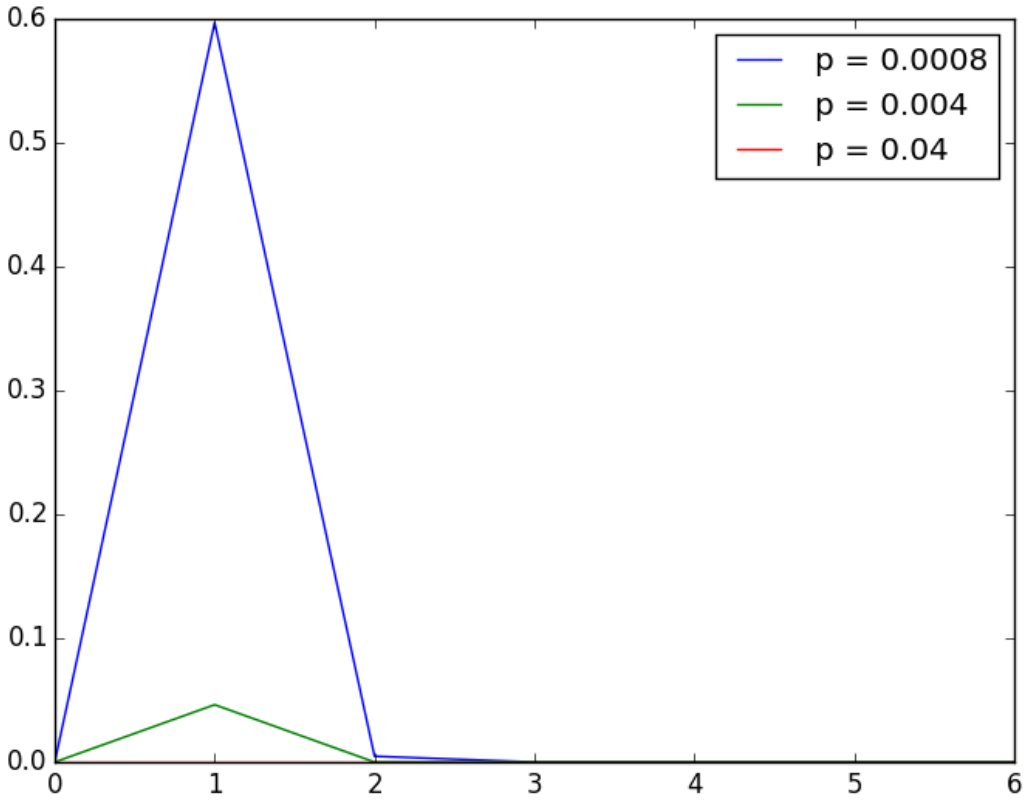


None of the graphs were connected. Output from `2iii.py`:

```
0 of the 40 graphs with q_0 = 0.0001 were connected
0 of the 40 graphs with q_0 = 0.001  were connected
0 of the 40 graphs with q_0 = 0.01   were connected
0 of the 40 graphs with q_0 = 0.1    were connected
```

I was surprised to discover that none of the graphs were connected. This likely contributed to the lower `q` values in our simulations as compared to those we computed theoretically. We don't account for this lack of connectivity in our theoretical equation, hence its dissimilarity with our simulations.

Part iv



Diameters:

```
q_0 = 0.0001 =>
  diameters = [32, 27, 28, 32, 30, 27, 28, 33, 30, 30, 28, 28, 31, 30, 31, 28, 31, 30, 34, 30
  max diameter = 34
  min diameter = 27

q_0 = 0.001 =>
  diameters = [29, 30, 28, 28, 30, 30, 31, 30, 31, 27, 27, 27, 31, 30, 29, 32, 30, 26, 33, 32
  max diameter = 33
  min diameter = 26

q_0 = 0.01 =>
  diameters = [34, 28, 30, 29, 29, 31, 32, 27, 31, 33, 31, 30, 32, 30, 27, 31, 31, 27, 29, 27
  max diameter = 34
  min diameter = 26

q_0 = 0.1 =>
  diameters = [27, 29, 28, 29, 32, 33, 30, 30, 30, 30, 29, 29, 28, 28, 35, 31, 30, 28, 29, 27
  max diameter = 35
  min diameter = 27
```

Problem 3 (25 points)

Description

- 2 possible versions of the social graph of voters. Each graph:
  - has `10,000` nodes, with ids `0 - 9999`
  - is undirected
- 40% for A, 40% for B, 20% undecided
  - if last digit of id is `0 - 3`, node supports B
  - if last digit of id is `4 - 7`, node supports A
  - if last digit of id is `8 - 9`, node is undecided
- 7-day decision period. Each day:
  i. Graphs initialized with every voter's initial state.
  ii. For each undecided voter:
  iii. If strict majority of their friends support B, they support B
  iv. If strict majority of their friends support A, they support A
  v. If equal number of their friends support A and B, we assign their side in alternating fashion (start with A)
    - To handle this, a single global variable should keep track of the current alternating vote type (initialize to A in the first round)
  vi. When updating the votes, use values from current iteration.
- NOTE: Decision process doesn't change loyalties of voters who have already made up their minds.

Part 1 :: Basic setup & forecasting (5 points)

Candidate B wins in both graphs (from files `g1.edgelist.txt` and `g2.edgelist.txt`). In `G1` and `G2`, (s)he wins with a margin of **126** and **174** votes respectively.

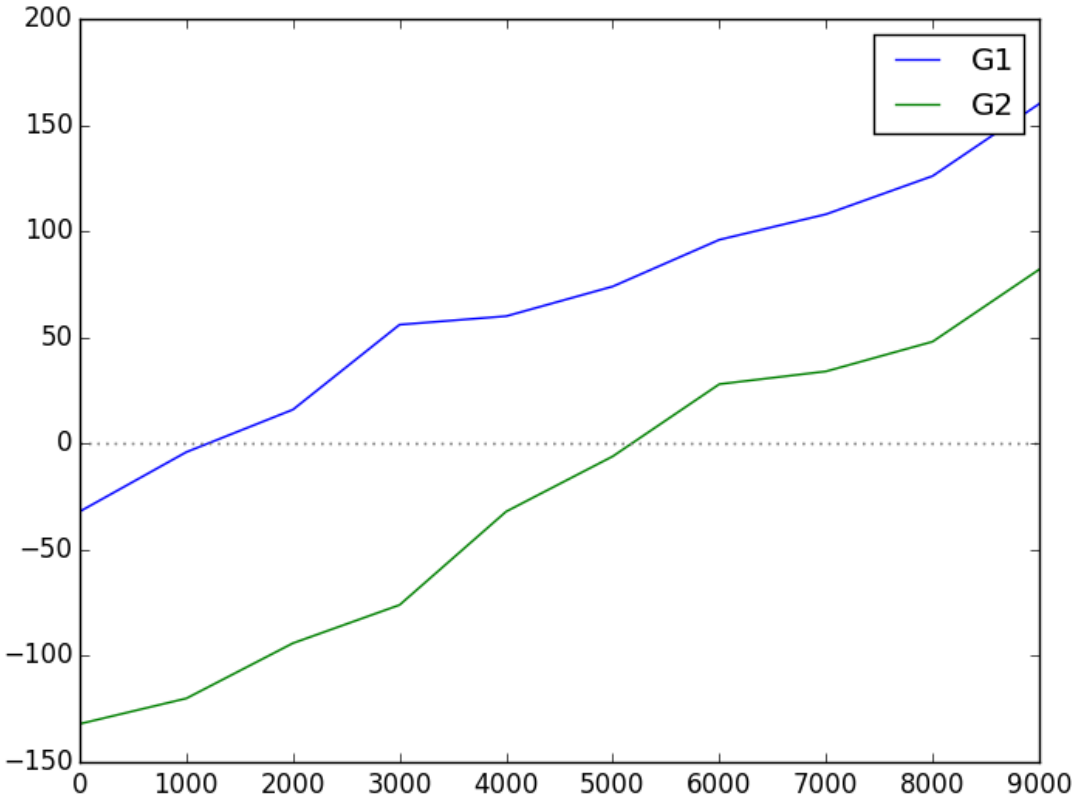
Output from `3-1.py`:

```
The election winner is B, wins by a margin of 126 votes
The election winner is B, wins by a margin of 174 votes
```

Part 2 :: TV Advertising (8 points)

Output from `3-2.py` :

```
Minimum expenditure required to win G1 = $2000
Minimum expenditure required to win G2 = $6000
```

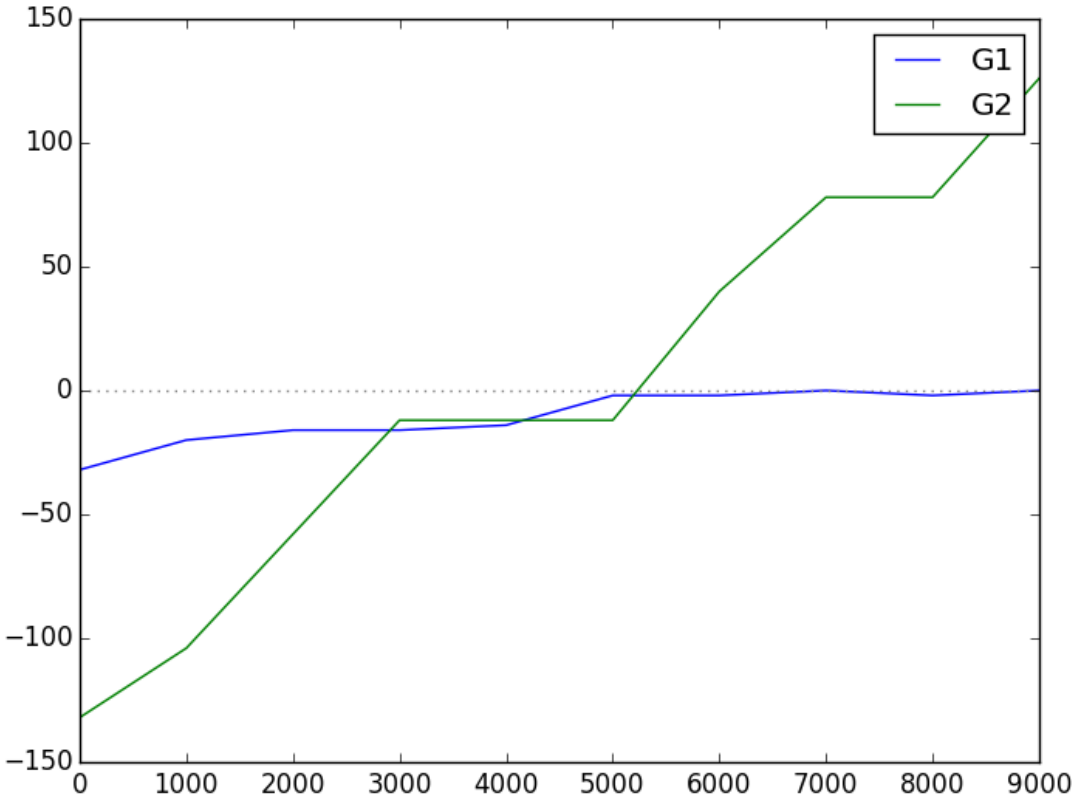


> NOTE: If we only count as undecided those who remain undecided after that round, our required expenditures become `$3,000` and `$5,000` for G1 and G2 respectively. It wasn't clear from the assignment handout how we should count these, so for my real answers I assume that we are counting as undecided those who were undecided before the persuasion round. This doesn't actually end up changing the basic results (i.e. for part c we still can't win in G1, even with this change).

Part 3 :: Wining & dining the high rollers (8 points)

Output from `3-3.py` :

```
No amount of spending will let you win G1 :(
Minimum expenditure required to win G2 = $6000
```



Part 4 :: Analysis (4 points)

Graphs 1 and 2 have totally different degree distributions. Running some Snap.py methods on both, it appears that G1 is basically random whereas G2 was created with some form of preferential attachment.

This analysis is consistent with the results from parts 2 and 3. In a random graph, the difference between the highest-degree nodes and lowest-degree ones aren't very different, so targeting the higher-degree ones doesn't buy you much. In a preferentially-attached graph, targeting those nodes that have the highest degree would be immensely helpful, because they have an outsized impact relative to other nodes.

The real world looks more like the preferentially-attached G2, particularly in the realm of political news. There are millions and millions of news sources out there (blogs, newspapers, etc.), but a very small number of them command the attention of the vast majority of eyes. Thus, if I were running a campaign, I would focus my efforts on those major players, following the "high-roller strategy".

Problem 4 (25 points)

Parts A & B

(a) • $i = 2$ → maximize influence of T by choosing the maximal set of two nodes
• greedy hill-climbing algorithm just looks at marginal benefit of adding the next node

Consider a graph with nodes A-K, where the following nodes have these influence sets:

A → {A, B, C, D, E, G, K} $|A| = 7$

F → {B, C, E, G, H, K} $|B| = 6$

J → {D, E, G, I, J} $|C| = 6$

The greedy solution fails because upon our first choice we grab node A, which has the largest marginal influence. It then chooses F, resulting in a total influenced set:

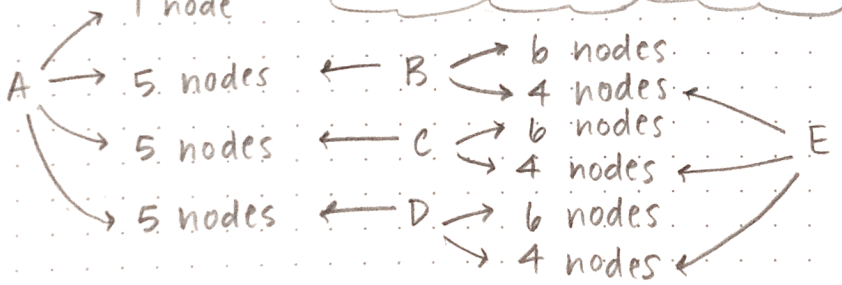
$S = \{A, B, C, D, E, F, G, H, K\} \Rightarrow |S| = 9$

However the optimal choice would've been nodes F & J with the following influenced set:

$S = \{B, C, D, E, F, G, H, I, J, K\} \rightarrow |S| = 10$

NOTE: Worked with John Luttig

(b)



Greedy solution: {A, E, D} (in that order, though the D could've also been B or C)
↳ yields $|S| = 16 + 12 + 6 = 34$

Optimal solution: {B, C, D}
↳ yields $|S| = (6+5+4) + (6+5+4) + (6+5+4) = 45$

Thus, we have $0.8 \cdot |S_{opt}| = 36 > |S_{greedy}| = 34$

Part C

One sufficient (but not necessary) property for greedy hill climbing to in fact be optimal is if **every node influences every other node** (i.e. each node's influence set contains all nodes).

Part D

> NOTE: Worked closely with a TA on this one, because I was really stuck.

Create an infinite number of unconnected, same-sized sub-graphs and say `f(S) = f(T)`. After any infinite number of steps `k`, there remain an infinite number of nodes that can still influence many further nodes.