Devon Zuegel // 21 Oct 2015 // CS 224W

Problem Set 2

Spoke with John Luttig while working on this pset.

Problem 1 :: Broadcasting the Network (25 points)

Part A

Undirected graph $G(V,E)$
Adjacency matrix $A \in \{0,1\}^{n \times n}$

For each round k:
- each vertex sends a msg to each of its neighbors at independent times within the interval $[k, k+1)$
  - For each edge (i,j), T_{ij} & T_{ji} are the times i & j send messages to j & i respectively
  - The times are uniformly distributed across the interval
- At any time $t \in [k, k+1)$ we can bug ≥ 1 node, intercepting any incoming msgs.
  - t_i = total time we bugged node $\boxed{i}$ during this round
  - t_i^β = prob. of us getting caught by node $\boxed{i}$

Objective: specify $t = [t_1, \ldots, t_n]$ during a single round to max. the expected # of msgs, while ensuring the prob. that we get caught is $< \gamma < 1$

(a) $m_i(t_i) = E(\text{\# msgs intercepted from node } \boxed{i} \text{ when monitoring for } \boxed{t_i})$

$A[i]$ = adjacency matrix for node $\boxed{i}$

For a given "friend" $\boxed{j}$, the prob. that we'll intercept a msg sent from $j \to i$ (lets call this statemt. A) is:

$Pr(A) = \underbrace{A[i][j]}_{\text{indicator var.}} \cdot t_i$ → % of round bugged

$m_i(t_i) = \sum_{j \in n} A[i][j] \cdot t_i$

Part B

(b) $A_i(t_i)$ = event that we get caught by node $\boxed{i}$

For any events A_1, A_2: $P(A_1 \cup A_2) \leq P(A_1) + P(A_2)$

Prove: The prob. of being caught by ≥ 1 node is $< \nu(t) = \sum_{i=1}^{n} t_i^\beta$

Thoughts:
- there are at most $n-1$ edges connecting node $\boxed{i}$ b/c it can't have an edge to itself
- $P(\text{caught by } \#i) = t_i^\beta = A_i(t_i)$

For any events A_1, A_2:

$$P(A_1 \cup A_2) \leq P(A_1) + P(A_2)$$

The probability of the event that we get caught by node `i`:

$$P(A_i(t_i)) = t_i^\beta$$

Let `P(A)` be the probability that we get caught by one or more nodes:

$$A = A_1(t_1) \cup A_2(t_2) \cup \ldots \cup A_2(t_2)$$
$$P(A) \leq P(A_1) + P(A_2) + \ldots + P(A_2)$$
$$\leq \sum_{i=1}^{n} P(A_i) = \sum t_i^{\beta} \ \blacksquare$$

Part C

Cauchy-Schwarz Inequality

$$\left(\sum_i a_i b_i\right)^2 \leq \left(\sum_i a_i^2\right)\left(\sum_i b_i^2\right) \rightarrow (a \bullet b)^2 \leq (a \bullet a)^2 (b \bullet b)^2$$

Goal: Find a strategy $t = [t_1, t_2, \ldots, t_n]$ that maximizes the total # of intercepted msgs.

Constraints: $\beta = 2$, $\gamma < 1$, $r(t) \leq \gamma$

$$M = \sum_i m_i(t_i) = \sum_i \sum_j t_i \cdot A(i,j) = \underbrace{\sum_i \left(t_i \sum_j A(i,j)\right)} = \sum_i a_i b_i$$

$$r(t) = \sum_i t_i^2 \leq \gamma$$

$$\left[\begin{array}{l} a_i = t_i \\ b_i = \sum_j A(i,j) = A_i \end{array}\right.$$

matrix 1×n

$$M = \left[\left(\sum_i a_i b_i\right)^2\right]^{1/2}$$
$$\leq \left[\left(\sum_i a_i^2\right)\left(\sum_i b_i^2\right)\right]^{1/2}$$
$$\leq \left[\underbrace{\left(\sum_i t_i^2\right)}_{r(t) \leq \gamma}\left(\sum_i A_i^2\right)\right]^{1/2} \leq \left[\gamma\left(\sum_i A_i^2\right)\right]^{1/2}$$

$$\boxed{M = \sqrt{\gamma \sum_i A_i^2}}$$

Part D

NOTE: It is not possible to have an undirected graph without self-loops with distinct degrees. However, for the sake of solving this problem we should maintain this assumption until the end.

Thoughts

- Constraints:
  - $0 \leq t_i \leq 1$ since the intervals are 1 unit of time long
  - $\sum_i t_i \leq \gamma < 1$
- Considering the case where all degrees are the same, we would want to bug each node for the same amount of time.
- If you have three companies that have a certain rates `(r1,r2,r3)` of return on your investment (all of them have zero risk), how would you invest your money?
  - You would invest all of your money into the company with the highest return rate.
  - If you had a constraint that said you could invest at most half of your money in any given company, you'd put half into the company with the best return and half into the one with the second-best return.

(d) $M = \sum_i \sum_j t_i A(i,j)$ ← Maximize this!

$r(t) = \sum_i t_i \le \lambda$ ← while keeping this true

$P(\text{getting caught by } \boxed{i}) = t_i^{\beta} = t_i$ (since $\beta = 1$)

Thoughts:

- $V = \{1,2,3\}$ $\quad t_1 + t_2 + t_3 \le .5$

  $E = \{(1,2),(2,3)\}$ $\quad \cancel{(\frac{2}{4} + \frac{2}{4} + \frac{1}{4})(.5) \le .5}$

- $t_i = \lambda \left(\frac{\# \text{ incoming edges}}{2 \cdot \text{total } \# \text{ edges}} \right)$

$$\begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & & 1 & \\ 2 & 1 & & 1 \\ 3 & & 1 & \end{array}$$

$$= \frac{\lambda \sum_j A(i,j)}{\sum_a \sum_j A(a,j)}$$

$$M = \sum_i \sum_j \left(\frac{\lambda \sum_b A(i,j)}{\sum_a \sum_b A(i,j)} \right)$$ ← Not maximal

Thoughts (again): $\quad t_1 \quad t_2 \quad t_3$

Better distribution: $\left(\frac{1}{16} + \frac{3}{8} + \frac{1}{16}\right) \le .5$

✗ $\left(\frac{1}{4} + \frac{2}{4} + \frac{1}{4}\right).5 \le .5$

✓ $\left(\frac{1}{8} + \frac{3}{4} + \frac{1}{8}\right).5 \le$

$t_1 \quad t_2 \quad t_3 \quad \lambda = .5$

1	2	3
1	2	1

$\frac{1}{8} \quad \frac{6}{8} \quad \frac{1}{8} \quad 2^3$

$$M = \left(\frac{1}{8}\right)(1) + \left(\frac{3}{4}\right)(2) + \left(\frac{1}{8}\right)(1) = 1.75$$

$$= \left(\frac{1}{8}\right)(0) + (1)(2) + \left(\frac{1}{8}\right)(0) = \boxed{2}$$

Answer

In the case where all node degrees are distinct, there exists a single node `n` whose degree `d` is greater than that of all other nodes in the network. Now let's consider some tiny amount of time `x`, which we can use to bug incoming messages of a single node in the network. We'd get most "bang for our buck" if we spent that `x` time at `n`, because it has `d` potential incoming messages at any given time. Thus, we want to spend every unit of time possible at that highest degree node. We can spend at most `1` unit at any given node since the interval is of length `1`, but since `\gamma < 1`, we don't have to worry about distributing it according to that original constraint since we don't have `> 1` unit of time to spend.

Part E (extra credit)

> TODO

Problem 2 :: Signed Triad Analysis (15 points)

Part A

> TODO

Part B

Calculated in `hw2p2a.py`:

```
number of positive edges = P = 592,551
number of negative edges = N = 119,232
total number of edges    = T = 711,783

p = P / T   # Fraction of positive
```

Part C

> TODO

Problem 3 :: Decentralized Search (40 points)

Part A

Setup:

- `T` = complete, perfectly-balanced `b` -nary tree
  - each node has `b` children
  - `b ≥ 2`
- `N` = # of network nodes
- `h(T)` = height of `T`

Thoughts:

- number of leaves grows by a factor of `b` each time the height increases
- the number of leaves on a tree `N` of height `h` is b^h

$$N = b^{h(T)}$$
$$h(T) = \log_b N$$

Part B

The maximum possible value of `h(v, w)` occurs when the ancestor common to `v` and `w` is the root of the original tree `T`. Therefore, this max value is the height of `T`, which we showed in part (a) to be $h(v, w) = \boxed{log_b N}$

Part C

> TODO fix me!!!

Say that `T_{v, d}` is our subtree that contains the desired height `d` and node `v`. It is easy to see that the leaves of $T*{v,d} – T*{v,d–1}$ satisfy `h(v, w) = d`. Since the number of leaves of a tree of height `h(T)` is $b^{h(T)}$, there are $b^d – b^{d–1}$ nodes satisfying h(v, w) = d.

Part D

Goal:
Prove the following:

$$Z = \sum_{w \neq v} b^{-h(h,w)} \ \leq log_b N$$

Thoughts:

- $$p_v(w) = \frac{1}{Z} b^{-h(h,w)} \ \rightarrow Z = \frac{1}{p_v(w)} \cdot b^{-h(h,w)}$$
- In part c we figured out that there are `b^i - b^(i - 1)` nodes `w` such that `h(v, w) = i`.

Answer:

$Z = \sum_{w \neq v} b^{-h(h,w)}$

$= \sum_{w \neq v} b^{-i}$

We can rearrange this and look at each level of the tree as our summation instead of looking at every `w ≠ v`. It is equivalent to consider each level `i` from `1` to $log_b N$ if we multiple each step by the number of “`w`s” exist at each level (which we figured out in part c):

$= \sum_{i=1}^{log_b N} (b^i - b^{i-1}) b^{-i}$

$= \sum_{i=1}^{log_b N} (1 - \frac{1}{b})$

$\leq log_b N$

Part E

> TODOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOOO

Let `d = h(v, t)`. For any of the b^{d-1} leaves `u` in `T'`, we have `h(v, u) = d`. The probability of `v` linking an edge to `u` is $\frac{b^{-d}}{Z}$. Therefore the probability of `v` linking an edge to `T'` is:

- `N(T)` = # of leaves in T'
- `prob` = prob of any 2 nodes connecting

$$\begin{aligned} N(T) \cdot \text{prob} &= \frac{b^{d-1} \cdot b^{-d}}{Z} \\ &= \frac{1}{bZ} \\ &= \frac{1}{b \sum_{w \neq v} b^{-h(h,w)}} \end{aligned}$$

$$\text{And we know that } Z \leq log_b N \text{, so we get the following:}$$

$$\geq \frac{1}{b \log_b N}$$

Part F

> TODO

Part G

> TODO

Part H

> TODO

Problem 4 :: Variations on a Theme of PageRank (25 points)

Part A

> TODO

Part B

> TODO

Part C

> TODO

Part D

> TODO

Part E

> TODO