COMS E6998-2: Analysis of Social Information Networks Mar 3, 2011

Lecture 7 – Effect of Network Topology

Instructor: *Augustin Chaintreau* Scribes: *Evangelia Skiani*

We study the effect of network topology on data dissemination. In particular, we are interested in the speed at which the information is spread throughout the network by a randomized gossip algorithm. An interesting question is to compare the optimal structured dissemination with the dissemination achieved by gossip, and compare the performance for different graph structures.

1 Algorithm

Before proceeding with the proof, let us first introduce some notation along with elementary definitions. We denote a graph $G = \{V, E\}$, where V is the set of nodes and E the set of edges. It may also be described by its adjacency matrix.

Definition 1. *Adjacency Matrix: A symmetric matrix with binary elements such that $A_{u,v} = 1$, provided there exists an edge from node u to node v, i.e. $(u, v) \in E$, and 0, otherwise.*

We will consider various algorithm to exchange information, characterized by a communication matrix which decides which nodes gets in contact with which at a given time.

Definition 2. *The communication Matrix P is any matrix that satifies:*

$$\sum_{u \in V, u \neq v} P_{uv} = 1 and a_{u,v} = 0 (i.e., \text{ no edge exists between } u \text{ and } v) \implies P_{u,v} = 0$$

It is therefore constrained by the topology of the graph G.

The gossip algorithm we analyze consists of the following steps:

- At each time step t, node u contacts a node v with probability P_{uv}, or nobody with probability P_{uu} (i.e. contacts itself). *Note here that, by definition, $\sum_{u \in V, u \neq v} P_{uv} = 1$, such that $P_{uu} = 1 - \sum_{u \in V, u \neq v} P_{uv}$.*
- *If either of the nodes u or v has received the information, then both of them possess the information at the next step, i.e. upon time $t + 1$.*

Observation 3. *In other words, there are two ways of communication, either a node already having the information, makes it flow to one of its neighbors, or a node receives the information from one of its incoming links. The process is random since at each step every node is randomly selected (according to P).*

We use the following notation:

- $S(t)$, the set of nodes $v \in V$ which possess the information at time t

- $S(0) = \{v\}$, the initial condition of the process (starting point)
- $|S(t)|$, a random variable (well defined) denoting the cardinality of the set V

The following theorem indicates the rate of convergence, eg. how fast the algorithm converges.

Theorem 4 (The Main Result)**.** *] If P is irreducible, symmetric, then by letting*

$$T^{one}_{spr}(\epsilon) = \sup_{u \in V} \inf\{t : Pr(S(t) \neq V | S(0) = \{u\}) \leq \epsilon\}$$

we have

$$T^{one}_{spr}(\epsilon) = O\left(\frac{\log n + \log \epsilon^{-1}}{\Phi(P)}\right)$$

where

$$\Phi(P) = \min_{S \subset V : |S| \leq \frac{n}{2}} \frac{\sum_{i \in S; j \in S^c} P_{ij}}{|S|}$$

Observation 5. *$\Phi(P)$ can determine the rate of convergence, e.g. it can be large for an expander graph, or relatively small for a line graph; in the latter case, it will take much more time for the algorithm to converge.*

1.1 Preliminaries

First, we prove two lemmata which are necessary for the proof.

A. **Markov Inequality**

Lemma 6. *Let X be a non-negative random variable such that $E[X] < \infty$ (its expectation is finite). Then, for $\alpha > 0$,*

$$P(X \geq \alpha) \leq \frac{E[X]}{a}$$

.

Proof. Let E be the set $\{X \geq \alpha\}$. We define the indicator function

$$g(x) = \mathbb{1}_E = \begin{cases} 1, & x \in E \\ 0, & x \in E^c \end{cases}$$

We also define the random variable $Y = \alpha g(x) = a \mathbb{1}_E$. It is trivial to prove that $Y \leq X$ holds a.e. Taking expectations on both sides, yields:

$$\begin{aligned} E[Y] \leq & \ E[X] \\ E[\alpha \mathbb{1}_E] \leq & \ E[X] \\ \alpha E[\mathbb{1}_E] \leq & \ E[X] \\ \alpha P(E) \leq & \ E(X) \\ \text{hence} P(X \geq \alpha) \leq & \ \tfrac{E[X]}{\alpha} \end{aligned} \tag{1}$$

□

B. **Conditional Expectation** On a probability space Ω, we define a random variable X, which is a mapping from the set Ω to another set in which X takes its values. let us assume that X takes value in the set of natural integers $\mathbb{N} = \{0, 1, 2, \ldots\}$. We then introduce a second random variable Y, with values also in $\mathbb{N}$. We which to define a new random variable denoted as $E[X|Y]$ and that will be defined by conditioning on the possible values taken by Y and consider the expected value of X.

First, let us consider how to condition w.r.t. an even $E \subseteq \Omega$. The conditional probability of X is defined as follows:

$$P(X = i|E) = \frac{P(\{X = i\} \cap E)}{P(E)} \tag{2}$$

Therefore, the conditional expectation, which is the average value taken by X if we assume that this event is verified, is obtained by integrating Eq.(2):

$$E[X|E] = \sum_i iP(X = i|E) \tag{3}$$

For simplicity, let us now consider a simple case, where Y takes value either 0 or 1. In other word, if we denote by E the event $\{Y = 0\}$ and by F the even $\{Y = 1\}$, then the whole knowledge of Y can be thought of knowing whether the event E or F is currently occuring. We consider the expected value of X conditioned on each of these two events, which are:

$$E[X|E] = \sum_i iP(X = i|Y = 0) \text{and} E[X|F] = \sum_i iP(X = i|Y = 1) \tag{4}$$

Let us denote in that case by $E[X|Y]$ a new random variable, which will take value $E[X|E]$ on the event E and the value $E[X|F]$ on the event F. It may be thought as the average value of X, conditioned on the value of Y, because this random variable connect the value of Y (and hence whether we are on the event E or F) to the value we can expect from X.

More generally, if Y takes any values in $\mathbb{N}$ we can consider the sequence of event $E_j = \{Y = j\}$ for any $j \in \mathbb{N}$ which forms a partition of the probability space. We can similarly defined $E[X|Y]$ as the only random variable that takes on each event E_j the constant value $E[X|E_j]$. Again this is a variable which represents, as a function of the value taken by Y what we can expect to see on average for X.

Note that it may also be formally defined as

$$E[X|E] = \sum_j \left(\sum_i iP(X = i|Y = j) \right) \cdot \mathbb{I}_{E_j} \tag{5}$$

where $\mathbb{I}_{E_j}$ denotes the indicator function, which takes value 1 for point in the event E_j and value 0 everywhere else.

It is possible to extend this definition to a general random variable Y. Note also that the expected value of $E[X|E]$ only depends on Y through the partition E_j. In particular any other variable Y with same partition (e.g. consider $-Y$ or Y^2) will lead to the same variable $E[X|Y]$.

Properties:

1. If X, Y are independent,

$$E[X|Y] = E[X]$$

i.e Y has no impact on the expected value of X and thus the conditional expectation is a deterministic constant rather than a general random variable taking different values.

2. Linearity of the conditional expectation.

$$E[X + \alpha Z|Y] = E[X|Y] + \alpha E[Z|Y]$$

3. If $\Phi(Y))$ is a function of Y, then

$$E[\Phi(Y)X] = E[\Phi(Y)E[X|Y]]$$

i.e. if $\Phi(Y)$ is known, we only consider the r.v. X on the pre-image set of Y.

1.2 Proof of the Main Theorem

We may prove Theorem 4 on a two-step approach assuming that we can divide the whole process into two distinct phases. We shall provide the proof only for the second phase; the same argument can be applied in phase #1, with an additional transformation involving martingales to apply Markov Inequality.

More specifically, for a graph containing $n = |V|$ nodes we look at the sequence $(S(t))_{t\geq 0}$ and consider different phases of the evolution:

- Phase #1: $S(t)$ grows from a singleton $S(0) = \{v\}$ to a set containinghalf of the node. This phase ends when $t = L - 1$, where $L = \inf\{t; |S(t)| \leq \frac{n}{2}\}$
- Phase #2: extending from time L, when $S(t)$ crosses the value $n/2$, until $|S(t)| = n$ and all nodes are reached.

The analysis of the second phase considers the shifted process, as if we start afresh e.g. at time $t = 0$, from a initial set $S(0)$ such that $|S(0)| \geq \frac{n}{2}$. In this case, equivalently, $|S(0)^c| \leq \frac{n}{2}$.

Lemma 7. *Let $E|S^c(t)|$ be the expected value of the size of the set S^c at time t. A lower bound is:*

$$E|S^c(t)| \leq \frac{n}{2}[1 - \Phi(P)]^t$$

Proof. We wish to show that for any value of $S(t)$ the relative progress on average after immediately t is large, hence that $S^c(t)$ shrank immediately after t.

We consider thus for any value of $S(t)$

$$E[|S^c(t)| - |S^c(t+1)|||S^c(t)|] \tag{6}$$

We introduce the variables defined for any j:

$$X_j = \begin{cases} 1, & j \text{ has the information at time } t\text{+1} \\ 0, & \text{otherwise} \end{cases}$$

Therefore, the nodes receiving the information exactly at time $t + 1$ is exactly the difference between the size of the set S^c at time t and its size at time $t + 1$. Therefore we have

$$|S^c(t)| - |S^c(t+1)| = \sum_{j \in S^c(t)} X_j \tag{7}$$

For any current set $S(t)$, every node j in $S^c(t)$ may decide to contact a node i in $S(t)$ with probability $P_{j,i}$, and in that case it will receive the information exactly at time $t+1$. Hence $EX_j|S(t)$ is at least $\sum_{i \in S(t)} P_{j,i}$, so that summing over all j, we have:

$E[\sum_{j \in S^c(t)} X_j|S(t)] \geq \sum_{j \in S^c(t)} \sum_{i \in S(t)} P_{i,j}$.

Let us define Δ as:

$$\Delta = \sum_{j \in S(t)^c} \sum_{i \notin S(t)^c} P_{ji}$$

so that we obtain a lower bound for the conditional expectation of Eq.(7) with regard to $S(t)$.

$$\begin{array}{rll} E[|S^c(t)| - |S^c(t+1)|||S^c(t)|] = & E\left[\sum_{j \in S^c(t)} X_j||S^c(t)|\right] & \\ & = \sum_{j \in S^c(t)} E[X_j|S^c(t)] & \text{(linearity of the expectation)} \\ & \geq \sum_{j \in S^c(t)} \sum_{i \notin S^c(t)} P_{ji} = \Delta & \end{array}$$

We proceed as:

$$\begin{array}{rll} E[|S^c(t)| - |S^c(t+1)|||S^c(t)|] & \geq \Delta & \\ E[|S^c(t)||S^c(t)|] - E[|S^c(t+1)|||S^c(t)|] & \geq \Delta & \text{linearity of the expectation} \\ |S^c(t)| - E[|S^c(t+1)|||S^c(t)|] & \geq \Delta & |S^c(t)| \text{ is } |S^c(t)|\text{-measurable} \\ \Rightarrow E[|S^c(t+1)|||S^c(t)|] & \leq |S(t)^c| - \Delta & \\ E[|S^c(t+1)|||S^c(t)|] & \leq |S^c(t)| \left(1 - \frac{\Delta}{|S^c(t)|}\right) & \end{array} \tag{8}$$

By the definition of the function $\Phi(P)$, $\frac{\Delta}{|S^c(t)|} \geq \Phi(P)$, i.e at least as large $\Phi(P)$.

Therefore,

$$E[|S^c(t+1)|||S^c(t)|] \leq |S^c(t)| \left(1 - \frac{\Delta}{|S^c(t)|}\right) \leq |S^c(t)|(1 - \Phi(P))$$

By taking expectations on both sides, we obtain:

$$E[|S^c(t+1)] \leq E|S^c(t)|](1 - \Phi(P)) \tag{9}$$

And by iterating Eq.(9) recursively:

$$\begin{array}{rll} E|S^c(t)| & \leq E|S^c(t-1)|(1-\Phi(P))^2 & \\ & \leq \ldots \leq E|S(0)][1 - \Phi(P)]^t & \\ & \leq \frac{n}{2}[1 - \Phi(P)]^t & \text{since } E|S(0)| \leq n/2 \text{ by assumption} \end{array} \tag{10}$$

□

Observation 8. *For all $j \in S^c(t)$, j has not yet received the information. Therefore, the size of $S^c(t)$ corresponds to the number of nodes which do not have the information.*

However, when t is large, we can prove that $|S^c(t)| \to 0$, i.e. $P(|S(t)^c| > 0]) \leq \epsilon$. This probability is equivalent to $P(|S(t)^c| \geq 1])$, since we assume that $t \in \mathbb{N}$.

We apply Markov's inequality since $|S^c(t)|$ is a non-negative random variable.

$$P(|S^c(t)| \geq 1]) \leq \frac{E[|S^c(t)|]}{1} \tag{11}$$

But, Eq.(10) yields:

$$E|S^c(t)| \leq \frac{n}{2}[1 - \Phi(P)]^t \leq \frac{n}{2}[\exp(-\Phi(P)]^t \leq \frac{n}{2}\exp(-t\Phi(P)) \tag{12}$$

To obtain the result, we refer to the convexity of $\exp(x)$, i.e. $1 + x \leq e^x$.

We choose

$$t = \frac{2\ln(n) + \ln(\frac{1}{\epsilon})}{\Phi(P)}$$

and Eq.(12) yields:

$$E|S^c(t)| \leq \frac{n}{2}\exp\left(-\Phi(P)\frac{2\ln(n) + \ln(\frac{1}{\epsilon})}{\Phi(P)}\right) = \frac{n}{2}\frac{\epsilon}{n^2} = \frac{\epsilon}{2n}$$

Hence, starting from a particular point $\{v\}$ in phase #1, after entering phase #2 the probability that $S(t) \neq V$ after t given above is less than $\epsilon/2n$. Summing up all the probabilities for any possible initial node, i.e starting point, and by Eq.(11), we have:

$$\sum_{v \in V} P(S(t) \neq V) \leq \sum_{v \in V} P(|S^c(t)| \geq 1]) \leq \sum_{v} \frac{\epsilon}{2n} \leq n\frac{\epsilon}{2n} = \frac{\epsilon}{2}$$

The complement of these events is the set $S(t) = V$ with probability greater than $1 - \frac{\epsilon}{2}$. This guarantees that the algorithm converges, i.e. it terminates.

2 Ranking

In this section, we discuss the problem of ranking the importance of a subset of nodes in a graph, based solely on topology, rather than content, or other features. We would like to deduce, for instance, whether an article could be useful in a literature review based on the its citations by different authors, something that can be estimated explicitly and immediately through any search engine, e.g. Google Scholar. Other applications where ranking, and more precisely, relative ranking, can be used include job search, webpage relevance in search engines etc.The metrics can be classified in three different categories. We note here that many different methods have been proposed which, nonetheless, can fail under certain conditions, i.e. for each of them there is always a counter-example for the ranking method to be inappropriate and unreliable. We divide these into two wide categories, Path Metrics(2.1) and Iterative Metric(2.2).

2.1 Path Metrics

Path metrics rely on network topology and, as indicated, they are mostly based on measuring the paths in the graph. Below, we describe four of them: node degree, closeness centrality, betweenness centrality and k-shell decomposition.

2.1.1 Node Degree

One prototypical and rather simple approach for evaluating the significance of the nodes involves looking at their degree. Therefore, nodes of higher degree than others imply more connections and therefore could be a good measure of their centrality in the network. This metric can be accurate when each link is costly enough to maintain or is extremely meaningful. However, the case where a node has many neighbors with few links would lead to a misinterpretation of its importance given that it might well be important but only locally, while there might exist nodes with fewer links which, however, provide 'bridges' among many different subsets, this implying that they might be more useful globally. (short-sightedness of the metric)

2.1.2 Closeness Centrality

As a second approach, we estimate the average distance of each node from every other node in the graph; lower values imply node centrality, since these nodes tend to increase connectivity in the network by exhibiting smallest paths. However, this metric is not discriminative enough and is sensitive to the size of the network, as larger graphs tend to generate a uniform distribution of the scores for almost every node and no conclusion is possible in that case.

2.1.3 Betweenness Centrality

In the third approach, we consider the number of shortest paths going through a node. In order to distinguish between central nodes and regular ones, we remove nodes from the graph, and re-estimate the average distance as in the previous approach. If the shortest path of the node increases significantly, this node is considered more important compared to other nodes whose removal had no effect on the graph as a whole.

2.1.4 k-shell Decomposition

In k-shell decomposition, we 'trim' the graph as follows: we assign a number to each node, indicating its importance. We start from the nodes with only one link and remove them. These nodes are assigned the number $k = 1$. We keep removing the nodes with a single edge until all nodes in the remaining network have at least 2 edges. This set of nodes is defined as the $\{k = 1\}$ set. We continue to the next step by removing nodes of degree 2, and keep trimming until all the remaining nodes have at least 3 edges. This is the set $k = 2$. And so on, until all nodes have been removed, so that termination is guaranteed. As easily understood, nodes in the set of the highest k are higher ranked. However, it is hard to apply this in large and complicated graphs, while, like all the other approaches, it can also cause inconsistencies.

In a nutshell, path metrics can be trustworthy but can fail under certain circumstances. Furthermore, their high computational cost is not negligible while, at the same time, they cannot distinguish whether some nodes acquire links strategically, in order to influence the results e.g. sending arbitrarily friend requests on Facebook, thus adding lots of meaningless links. As a consequence, more robust and flexible metrics need to be defined in an effort to eliminate the vulnerability of the existing ones and enhance ranking efficiency.

2.2 Iterative Metrics

In iterative metrics, we apply methods so as to remove nodes which are not 'relevant'. Suppose, for example, that we rank the importance of a group of websites with respect to their relationship to news updates. The ranking is based on the number of votes each website receives from a set of nodes. By counting the number of votes, we obtain a rough estimate of the importance of each website. However, this raw score can be misleading, especially when the votes are coming from isolated nodes, i.e. nodes who have voted only once as compared to others with more than one links. Therefore, we wish to identify the 'Authorities', i.e. the nodes that have received votes judiciously. This boils down to identifying the 'Hubs', the nodes who tend to provide more votes in the subset we examine.

In order to improve the initial estimation, we apply a recursive algorithm: we classify the nodes as 'hubs' and 'authorities', the ones providing and receiving votes respectively. We assign weights to the nodes according to their voting power, a_j the score of the j^{th} authority and h_i the score of the i^{th} hub. We update the scores iteratively, as described in the following section, until convergence of the process (after re-normalization of the weights) is attained (if it converges). We are interested to know whether this process converges, what is the rate of convergence and whether it depends on the initial vote counts.

2.3 Spectral Analysis

For the 'Authorities' and 'Hubs', previously discussed, we denote a_j as the score of the j^{th} authority and h_i as the score of the i^{th} hub.

Definition 9. *For a digraph $G(V, E)$ of m hubs and n authorities, let us define the matrix A as the adjacency matrix i.e. $A_{ij} = 1$ iff there is an edge between hub i and authority j and 0, otherwise.*

At each step, we update the weights of each hub according to the equation:

$$h_i \leftarrow \sum_{(i,j)\in E} a_j = \sum_{j=1}^{m} A_{ij} a_j$$

Therefore, the row matrix h is defined as:

$$h = A \times a$$

And the column matrix a is equal to

$$a = A^T h$$

We also update the weights of the authorities:

$$a_j \leftarrow \sum_{(i,j)\in E} h_i = \sum_{i=1}^{n} A_{ij} h_i = \sum_{i=1}^{n} A^T_{ji} h_i$$

Initially $h_i = 1\ \forall i = 1, \ldots m$. On the k^{th} step, we have

$$a(k+1) = A^T h(k+1) = A^T A a(k) = \cdots = (A^T A)^k a(0)$$

where the last equality follows by recursion. Under weak conditions for the initial weights $a(0)$ and the eigenvalues of the matrix, we can prove the fast convergence of the process to a global property.

Theorem 10. *If $\forall j$, $a_j(0) > 0$ and $A^T A$ satisfies $\lambda_1 > \lambda_2$(i.e. the largest eigenvalues of $A^T A$ satisfy $\lambda_1 > \lambda_2$), then*

$$\left(\frac{a(k)}{\lambda_1^k}\right) \text{ converges to } x_1 \text{ as } k \text{ becomes large}$$

where x_1 is the eigenvector corresponding to λ_1 (of unit norm)

To summarize, iterative metrics are insensitive to boundary effects and cheating. However, the dominant eigenvalue and, in turn, the eigenvector can impose restrictions on the convergence of these iterative processes. Added to this, convergence is also subject to the spectral gap. Path metrics, on the other hand, are much simpler to define but, as such requires sometimes to compute many paths which is not easy to make from a computational standpoint. The simpler ones (like the degree) are in general more sensitive to cheating also means that they should be manipulated with care.