AM 221: Advanced Optimization Spring 2016

Prof. Yaron Singer *Section 10 — Wednesday, Apr. 6th*

Today's section will be dedicated to the problem of Influence Maximization in Social Networks. As discussed in class, the motivation behind this problem is to exploit the ability of networks (*e.g.* social networks) to propagate content (*e.g.* ideas, a piece of information, etc.). In its most general form, the input to this problem is:

- a directed graph $G = (V, E)$, the network over which to perform Influence Maximization.
- a function $f : 2^V \to \mathbb{R}$ mapping a set S of nodes to their influence value $f(S)$. This represents how much influence on the network will be obtained if this set of nodes is used as a *seed* to the influence propagation process.
- a budget K.

The goal is to find a set S solution (possibly approximate) to:

$$\max_{|S|\leq k} f(S)$$

1 Models of Influence

The nature of the Influence Maximization problem depends on the lot on the choice of influence model, *i.e* the influence function f. It is not clear a priori how to quantify the influence of a set S of nodes on the entire network. In a famous paper [2], Kempe, Kleinberg and Tardos proposed a general class of influence models. The idea is to define a random process on the network originating from S and propagating like a contagion. The expected number of infected nodes at the end of the contagion is the influence $f(S)$ of the set S. Thus, we reduced the problem of defining influence to the one of defining a random contagion process on the network.

We will now present a specific, yet representative, influence model from [2], the *Independent Cascade Model*. In this model, the graph G is weighted: for each edge (u, v) there is a weight $p_{u,v} \in [0, 1]$ to be interpreted as the probability that this edge will propagate the contagion. The time is discretized, at each time step $t \in \mathbb{N}$ each node in the graph is in one of three states:

- *infected:* the node has just been infected and will remain active for one time step where it attempts to infect its neighbors.
- *susceptible:* the nodes hasn't been infected yet and will possibly become infected during this time step or later ones.
- *inactive:* this node was infected in the past but is now inactive. Infected nodes only stay infected during one time step, after which they become inactive.

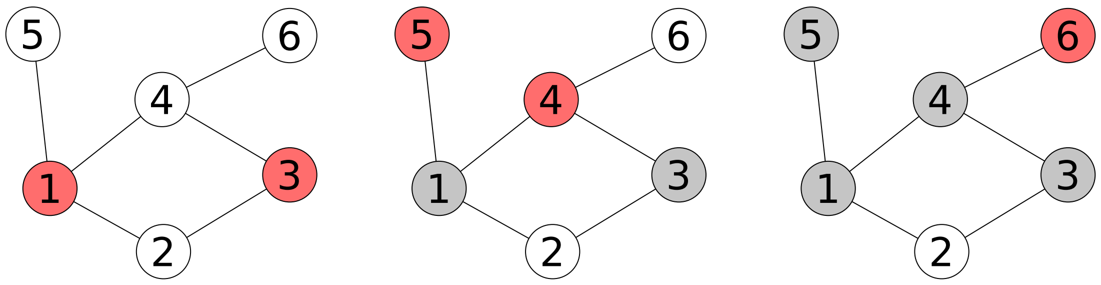

Figure 1: An illustration of the Independent Cascade Model on a graph with 6 nodes. At $t = 0$ (left graph), $I_0 = \{1, 3\}$, both infected nodes fail to infect node 2, but successfully infect nodes 4 and 5. At $t = 1$, $I_1 = \{4, 5\}$, nodes 1 and 3 are now inactive; 5 has no susceptible neighbor and stops propagating the contagion, while 4 successfully infects 6. At $t = 2$, $I_2 = \{6\}$, but since 6 has no susceptible neighbor the contagion process stops there.

As explained, each infected node stays active during one time step during each it attempts to infect each of its susceptible neighbors. For infected node u and susceptible node v, the attempt succeeds with probability $p_{u,v}$. The attempts are independent random events. From the point of view of a susceptible node, each of its infected parents attempts to infect him (in an arbitrary order). As long as one of the attempt succeeds, the susceptible nodes will be infected at the next time step and will in turn attempts to infect its neighbors.

Formally, let us write I_t and S_t the set of infected and susceptible nodes at time step t. For $v \in V$, let us also defined $P(v)$ the parents of v:

$$P(v) = \{u \in V \mid (u, v) \in E\}$$

Then, for a susceptible v in S_t, its probability of being infected at time step $t + 1$ is:

$$p(v, I_t) \overset{\text{def}}{=} 1 - \prod_{u \in P(v) \cap I_t} (1 - p_{u,v})$$

Indeed, for each of the infected parent of v (a node u in $P(v) \cap I_t$), the attempt fails independently with probability $(1 - p_{u,v})$ and v becomes infected exactly in the case where not all the attempts fail.

The sets I_t and S_t thus follow the following recursive definition:

$$I_0 = S \quad S_0 = V \setminus I_0$$

Given I_t and S_t, let I_{t+1} be initially empty. For each v in S_t, add v to I_{t+1} with probability $p(v, I_t)$. Finally define $S_{t+1} = S_t \setminus I_{t+1}$. An illustration of the Independent Cascade Model is given in Figure 1.

We define $I(S)$ the set of all infected nodes throughout a contagion process originating at S:

$$I(S) = \bigcup_{t \geq 0} I_t$$

Note that the union is only over a finite number of time steps: either the entire networks eventually becomes infected, or the infection dies from the infected nodes failing to infect their neighbors. Also note that the sets I_t are random sets, hence $I(S)$ is a random variable.

Finally, we define the influence function $f(S)$ in the independent cascade model as the expected number of infected nodes for a contagion originating at S, *i.e*:

$$f(S) = \mathbb{E}[|I(S)|]$$

where the expectation is taken over the random infection attempts from the infected nodes.

2 The Independent Cascade Model is submodular

A remarkable result from the paper [2] is that for the independent cascade model as well as for other influence models introduced in the same paper, the influence function is submodular. Intuitively, the influence of a node decreases when it is part of a larger seed because its role in propagating the contagion can be partially taken over by the other seed nodes.

An equivalent definition of the Independent Cascade Model. How to prove that $f(S)$ defined in the previous section is submodular? Tracing the number of infected nodes seems hard because it requires understanding the randomness at each time step. Fortunately, there is a way to give a reformulation of the Independent Cascade Model in which computing $I(S)$ will be easier.

The reformulation comes from the following two observations:

- if a node becomes infected, this means that it was connected to an infected node at the time of infection. The infected node was itself connected to an infected parent when it became infected. Repeating the argument, we see than we can trace a path from any infected node to an original node in the seed set I_0 of the nodes infected at time step 0. The edges along that path are edges which propagated the infection, *i.e* successful contagion attempts occurring with probability $p_{u,v}$ for each edge (u, v).
- each edge (u, v) is used at most one during a contagion process: if u becomes infected and v is susceptible when this happens, then the edge (u, v) will propagate the contagion from u to v with probability $p_{u,v}$. Furthermore all the edges succeed or fail independently of the others.

Using these two observations, let us define:

- D_G a distribution over subgraphs of G: these are subgraphs $G' = (V, E')$ over the same vertex set and where each edge $(u, v) \in E$ is included in E' independently with probability $p_{u,v}$. Formally:

$$\mathbb{P}_{D_G}[G'] = \prod_{e \in E'} p_e \prod_{e \notin E'} (1 - p_e)$$

- for a graph G' and a set S, let us define $R_{G'}(S)$ the set of nodes reachable from S in G', *i.e* for each node v in $R_{G'}(S)$, there exists a node u in S such that there exists a path from u to v in G'.

Then we can summarize the discussion above in the following theorem.

Theorem 1. *The two random variables $I(S)$ and $R_{G'}(S)$ (where $G' \sim D_G$) have the same distribution.*

Using this theorem, we are now ready to prove our main theorem.

Theorem 2. *The influence function $f(S) = \mathbb{E}[|I(S)|]$ in the Independent Cascade Model is monotone submodular.*

Proof. Using Theorem 1, we can equivalently prove that $\mathbb{E}[|R_{G'}(S)|]$ is monotone and submodular. Observe that:

$$\mathbb{E}[|R_{G'}(S)|] = \sum_{G'} \mathbb{P}_{D_G}[G']|R_{G'}(S)| \tag{1}$$

You will show in the problem set that a non-negative linear combination of monotone submodular functions is monotone submodular, hence it is sufficient to prove that for any G', $g(S) \overset{\text{def}}{=} |R_{G'}(S)|$ is monotone submodular

For a set S and $u \in V$, we have:

$$R_{G'}(S \cup \{v\}) = R_{G'}(S) \cup R_{G'}(v)$$

which shows that $g(S)$ is monotone. To prove that g is submodular, let us consider $S \subseteq T \subseteq V$ and $u \in V \setminus T$:

$$g(S \cup u) - g(S) = |R_{G'}(S \cup u)| - |R_{G'}(S)| = |R_{G'}(u) \setminus R_{G'}(S)|$$

Similarly:

$$g(T \cup u) - g(T) = |R_{G'}(T \cup u)| - |R_{G'}(T)| = |R_{G'}(u) \setminus R_{G'}(T)|$$

By monotonicity, we have $R_{G'}(S) \subseteq R_{G'}(T)$, hence:

$$g(S \cup u) - g(S) \geq g(T \cup u) - g(T)$$

which is exactly the definition of the submodularity of g. □

A consequence of Theorem 2 is that the Influence Maximization problem under the Independent Cascade Model inherits all the nice approximability guarantees of subdmodular optimization. In particular, we know that the simple greedy algorithm can be used to find a set S of nodes whose influence is at least 63% of the optimal influence achievable among all sets of size at most K.

Remark. A computational issue when applying the greedy algorithm to the function f defined above is that we need to evaluate f polynomially many times on different sets during the execution of the algorithm. Note however that the evaluation of Equation 1 requires summing over exponentially many ($2^{|E|}$) subgraphs of G). A way around this issue is to approximate the sum by Monte Carlo estimation: sample repeatedly a subgraph G' and average $R_{G'}(S)$ over the samples. Then we need to prove that the estimation error introduced by the Monte Carlo sampling does not affect the approximation ratio of the greedy algorithm too much. You will do this in the problem set.

3 Extension: Adaptive Seeding

We will now look at a variant of Influence Maximization called Adaptive Seeding. The motivation for this variant is that in many applications of Influence Maximization, the person doing the optimization does not have access to all the nodes in the network. For example, in applications to marketing campaigns, an online retailer can only communicate with the users who visited its website. To capture this constraint, let us assume that there is a core set X of nodes in V, such that the seed set S has to be a subset of the core set. The influence maximization problem then becomes:

$$\max_{S\subseteq X,\, |S|\leq K} f(S) \tag{2}$$

The curse of the Power Law. It has been well-observed that in social networks the degree distribution of nodes follow a power-law distribution. That is, denoting by $p(d)$ the probability that a random node in the network has degree d, we have:

$$p(d) \propto \frac{1}{d^{\alpha}}$$

where α is the exponent of the law. Power laws are heavily skewed: most nodes have a small degree. In particular, if X is small compared to V, then with probability most nodes in X have a small degree. And we should expect the optimal solution of Equation 2 to be much smaller than the optimal solution to the original Influence Maximization problem.

The friendship paradox. Another well-observed phenomenon in social networks is the friendship paradox: *"Your friends have more friends than you do."* Formally speaking, it is possible to prove that there is a gap between the average degree of a sample of nodes and the average degree of their friends.

The idea of Adaptive Seeding is to exploit the friendship paradox: if the nodes in X have a small degree, then instead of spending the entire budget K on them, it is beneficial to only spend a fraction on the budget, wait for their (possibly influential) friends to be infected and then spend the remaining part of the budget on the friends.

We can now formally describe the Adaptive Seeding problem:

- On day 1, select a subset $S \subseteq X$ of the core set X such that $|S| \leq K$.
- For each node $v \in N(X)$ (the neighbors of the nodes in X), there is a probability p_v that they will be available on day 2 if one of their neighbor in X is selected on day 1.
- On day 2, select an optimal set T for the influence function f among the nodes which are available on day 2 with the remaining part of the budget.

The probability that the set $R \subseteq N(S)$ becomes available on day 2 is:

$$p_S(R) = \prod_{v\in N(S)} p_v \prod_{v\notin N(s)} (1-p_v)$$

The goal of the Adaptive Seeding problem is to select optimally on day 1 so as to maximize the expected influence on day 2. In other words, the goal is to solve the following optimization program:

$$\max_{\substack{S\subseteq X\\ |S|\leq k}} \sum_{R\subseteq N(S)} \left(p_S(R) \max_{\substack{T\subseteq R\\ |T|\leq k-|S|}} f(T) \right)$$

The Adaptive Seeding problem was first introduced in [3]. Despite being a challenging optimization problem, it is possible to obtain a constant approximation algorithm for this problem when f is monotone submodular. In the specific case where f is an additive function, faster algorithms where obtained in [1]. An experimental evaluation on the Facebook social graph can also be found in [1].

References

[1] Thibaut Horel and Yaron Singer. Scalable methods for adaptively seeding a social network. In *Proceedings of the 24th International Conference on World Wide Web*, pages 441–451. International World Wide Web Conferences Steering Committee, 2015.

[2] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

[3] Lior Seeman and Yaron Singer. Adaptive seeding in social networks. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 459–468. IEEE, 2013.