COMS E6998-2: Analysis of Social Information Networks — Feb 17, 2011

Lecture 5 – Maximizing the Spread of Influence

Instructor: *Augustin Chaintreau* — Scribes: *Fang Qian*

1 Introduction

We assume a graph with some points distributed in it.The problem is to find an initial fixed size S_0 to maximize the total spread. Suppose node u is trying to affect node v. In this model the threshold Q_v is uniformly distributed in [0,1]. And v becomes active when $Q_v \leq g_v(X)$ and the probability is$p_{v(u,X)}$ where X is a set of nodes in $N(v)$ are initially active. $N(v)$ stands for all neighbors of v. The probability $p_{v(u,X)}$ is dependent on factor X because nodes in X must fail at affecting v to let node u successfully affect v. In other words, the influence of u on v depends only on u and v, not on the set of other nodes that have already tried and failed to influence v. Special cases are Granovetter, Morris and Independent. If p order independent, the two models are equivalent.
Assume S_0 is the initial set which is activated.

$S_0 = S_0$;
S_1=$S_0 \cup \{v \nsubseteq S_0 | Q_v \leq g_v(S_0) \cap N(v)\}$;
.
.
.
S_{t+1}=$S_0 \cup \{v \nsubseteq S_0 | Q_v \leq g_v(S_0) \cap N(v)\}$

After using the term of $p_v(u, X)$:

S_0=S_0;
S_1: $\forall u \in S_0, \forall v \notin S_0$ such that $(u, v) \in E$. For each of them with probability $P_v(u, \phi)$ because no one has tried before;
.
.
.
S_{t+1}: $\forall u \in S_t, \forall v \notin S_t$ such that $(u, v) \in E$. For each of them with probability $P_v(u, \text{“}S_{t-1}\text{”} \cap N_v)$ because no one has tried before;

2 Maximizing the spread of influence

We can formulate the question concretely as follows. There is a natural influence function $f(.)$ defined as follows: for a set S of nodes, $f(S)$ is the expected number of active nodes at the end of the process, assuming that S is the set of nodes that are initially active. Let $f(S)$ be the objective function. (We will assume in this section that all graphs are finite, so the processes we are considering terminate in a number of steps that is bounded by the total number of nodes n.) From other point of view, $f(S)$ is the expected number of total activated points if we get S to be the set of initial active nodes.

Problem 1. *Given a parameter k, find a k-node set S to maximize $f(S)$.*

Theorem 1. *Whenever p_v shows diminishing return: There exists a simple polynomial greedy algorithm computing S such that $f(S) \geq \left(1 - \frac{1}{e}\right) f(S^*)$.(Where S is a k-element set obtained by repeatedly including the element producing the largest marginal increase in f (for k iterations), and S^* is the set that maximizes the value of f over all k-element sets.)*

Proof. The proof that the greedy algorithm will give us the above guarantee follows in three steps:

1. Show that the result holds if f is submodular, i.e.

$$\forall S \subseteq T \subseteq N, \forall v \in N$$

$$f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T)$$

2. Show f is submodular under this condition on p_v.

3. Finally, prove that each step is polynomial (more involved).

Step1 We show that the result holds if f is submodular:

Lemma 2. *If f is non-negative, non-decreasing (e.g., $f(S+v) \geq f(S)$), and submodular; then greedy algorithm provides $1 - \frac{1}{e}$ approximation of maximizing $f(S)$ subject to $|S| = k$.*

Proof. First, let's explain submodularity. Submodularity (diminishing return): the benefit of adding elements decreases as the set to which they are being added grows. We also note that all the influence functions arising from our models here are monotone, in the sense that $f(\phi) = 0$ and $f(S) \leq f(T)$.

And $\forall S \subseteq T \subseteq N, \forall v \in N$

$f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T)$

Let $v_1, v_2, \ldots, v_k$ be the nodes selected by the greedy algorithm (in order in which they were selected), and denote $S_i = \{v_1, v_2, ...v_i\}$. Then, the marginal benefit derived from the addition of element v_i is $\delta_i = f(S_i) - f(S_{i-1})$. Let S^* be the optimal solution, and $W_i = S^* \cup S_i$. First, the monotonicity of f implies that $f(S^*) \leq f(W_i)$ for all i. Because the algorithm chooses to add the best available node in the (i+1)st iteration, and the benefit of any elements added later cannot be greater by submodularity, the total objective value for the set W_i is at most $f(W_i) \leq f(S_i) + k\delta_{i+1}$, and thus also $f(S^*) \leq f(S_i) + k\delta_{i+1}$.

Solving this for δ_{i+1}, and using that $f(S_{i+1}) = f(S_i) + \delta_{i+1}$ now show that $f(S_{i+1}) \geq f(S_i) + \frac{1}{k}.(f(S^*) - f(S_i))$. We will prove by induction that $f(S_i) \geq (1 - (1 - \frac{1}{k})^i).f(S^*)$. The base case $i = 0$ is trivial.

For the inductive step from i to $i+1$, we use the above inequality to write

$$f(S_{i+1}) \geq f(S_i) + \frac{1}{k}.(f(S^*) - f(S))$$

$$(1 - \frac{1}{k})f(S_i) + \frac{1}{k}.f(S^*)$$

$$\geq (1 - \frac{1}{k})(1 - (1 - \frac{1}{k})^i).f(S^*) + \frac{1}{k}.f(S^*)$$

$$= (1 - (1 - \frac{1}{k})^{i+1}).f(S^*)$$

completing the inductive proof. Using the fact that $(1-\frac{1}{k}^i) \geq \frac{1}{e}$ for all $i \leq k$, we obtain that the algorithm is a $(1-\frac{1}{e})$-approximation. □

Example 3. *Given $f(S) = |S|, S \subseteq V, v \in V \mapsto A_v \subseteq V$ and we'll get subset A_v if we take v in S_0, we'll get subset $A_{v'}$ if we take v' in S_0.*

So, $f(S) = \left|\bigcup_{v\in S} A_v\right|$, $f(S \cup \{x\}) = \left|\bigcup_{v\in S} A_v \cup A_x\right|$

For $S \subseteq T$, $f(T) = \left|\bigcup_{v\in S} A_v \cup \bigcup_{w\in T, w\notin S} A_w\right|$

$f(T \cup \{x\}) = \left|\bigcup_{v\in S} A_v \bigcup_{w\in T\S} A_w \cup \bigcup A_x\right|$

Then we get: if f and g are non-negative, monotone and submodular. Then (f+g)(S)=f(S)+g(S).

If $f_1, f_2...f_r$ are submodular functions, and $c_1, c_2...c_r$ are non-negative real numbers, then the function f defined by the weighted sum $f(X) = \sum_{i=1}^{r} c_i f_i(X)$

Step 2 Show f is a submodular function of S. Recall that a function on sets is monotone if $f(S') \geq f(S)$ whenever $S \subseteq S'$, and submodular if $f(S' \cup x) - f(S') \leq f(S \cup x) - f(S)$ whenever $S \subseteq S'$. Equivalently, submodularity is characterized by the condition that $f(S) + f(T) \geq f(S \cap T) + f(S \cup T)$ for all sets S, T.

Under the model of Independent Cascade Model:

Definition 4 (Independent Cascade Model)**.** *Given a complete directed graph G with edge probabilities p_e, we consider an infection model where once a node becomes active, it infects a neighboring node v with probability $p_(u, v)$. If the attempt succeeds, v becomes active; u, however, does not get to try infecting v again.*

Much of the challenge in proving submodularity here is that the computational intractability of our function f also translates into a kind of conceptual intractability: we don't know enough about its structure to simply plug it into the submodular inequality and hope to verify it directly. A much more powerful approach is instead to decompose f into simpler functions, and check submodularity for the parts of this decomposition. To lay the groundwork for this plan, we start by discussing two useful facts about submodularity. The first is this: if $f_1, f_2...f_r$ are submodular functions, and $c_1, c_2...c_r$ are non-negative real numbers, then the function f defined by the weighted sum $f(X) = \sum_{i=1}^{r} c_i f_i(X)$ is also submodular. The second fact is actually the identification of a simple, useful class of submodular functions, the following. Suppose we have a collection of sets $C_1, C_2...C_r$ and for a set $X \subseteq \{1, 2...r\}$ we define $f(X) = \left|\bigcup_{i\in X} C_i\right|$. For obvious reasons, we will call such a function f a size-of-union function. Then it is not hard to verify that any size-of-union function is submodular. As we will see below, such functions will arise naturally in our decomposition of the influence function f. Now, consider the following alternate way of viewing the Independent Cascade Process. In the standard view, each time a node u becomes active, it flips a coin of bias $p(u, v)$ to determine whether it succeeds in activating v.

Now, in the alternate, equivalent view of the process, suppose that for each edge (u, v), we flip a coin of bias $p(u, v)$ in advance, planning to only consult the outcome of the coin flip if we ever need to, when u becomes active.

If there are m edges in the graph, then there are $2m$ possible collective outcomes of the coin flips. Let α denote a particular one of these $2m$ outcomes, and let $f(S)$ denote the eventual number of activated nodes, given that S is the initial active set and is the outcome of the coin flips. Unlike f, the function f_α is easy to understand, as follows. For each edge (u, v), we say that it is live (with respect to α) if the advance coin flip came up heads. For a node s, we let $R_s^{\langle\alpha\rangle}$ denote the set of all nodes that are reachable

from s on paths consisting entirely of live edges. It is now easy to check that a node is eventually activated if and only if it is reachable from some node in S by some path consisting entirely o live edges, and hence $f_\alpha(S) = \left|\bigcup_{s \epsilon S} R_s^{\langle\alpha\rangle}\right|$

In other words, each f_α is a size of union function, and hence submodular.

But this is essentially all we need, since by definition

$f(S) = \sum_\alpha \text{Prob}[\alpha] \cdot f_\alpha(S)$.

That is, f is a non-negative weighted sum of submodular functions, and hence f is submodular, as desired.

□