Near Optimal Strategies under Knapsack Constraint for Targeted Marketing in Social Networks

Matthieu Cordier, Victor Catteau

05/09/2019

Outline

Outline

Introduction to the problem

Problem: **Which influencers should you hire to maximize advertising reach on a network ?**

One star influencer ? Several less famous and less costly influencers ? **What is the optimal selection strategy ?**

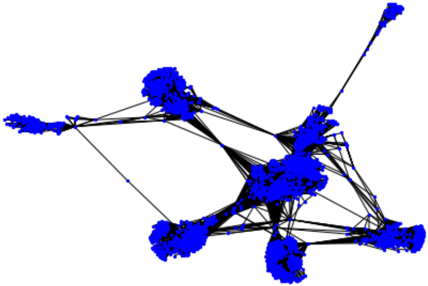


Figure: Visualization of a Facebook Graph

Outline

Diffusion model

We modelize word-of-mouth spread of information in the network as a **SIR infection model, with transition rates β_i and γ_i for each node i**

$I_0 \subset V$ **subset of initially infected nodes: influencers**

Simplifying assumption: β_i and γ_i are constants accross the network (not true in reality)

Outline

1. Introduction :Influence Maximization problem
   - Intuition
   - Modelization
   - Influence Maximization Formulation
2. First method : Topological heuristic
   - Predicting the Influence Function
   - Optimization heuristics
3. Second method : Submodular optimization
   - Desired property for f : submodular and monotone
   - Submodular functions
4. Experiments
   - Details on the experiment
   - Submodular functions and greedy algorithm : optimizing parameters
   - Experiment on a real social network
5. Summary

Influence Maximization Formulation

Influence Maximization Problem

$$S^* \in \underset{S \subset V}{\operatorname{argmax}}\, \sigma(S, T) \qquad \text{subject to} \quad c(S) \leq B \tag{1}$$

With the influence function:

$$\sigma(I_0, T) = \mathbb{E}[\sum_{v \in V} \mathbf{1}\{t_v < T\}] = \mathbb{E}[I_T + R_T] \tag{2}$$

$I_t \subset V$ and $R_t \subset V$ number of infected and removed nodes at time t.

Cost function linear on degree: $C(S) = \sum_{s \in S} c_s(s)$

The problem is NP-Hard

Main challenge (initially demonstrated by Kempe and al. who proposed a first heuristic, in a simplified model) [1]:

NP-Hardness

Maximizing the influence function $\sigma(I_0, T)$ is a NP-Hard problem

How do we avoid costly Monte-Carlo simulations to approximate the influence function ?

Outline

Predicting the Influence Function

Idea

Simplify the problem by creating a deterministic influence function based on network topology, to avoid having to do Monte-Carlo simulations for each node.

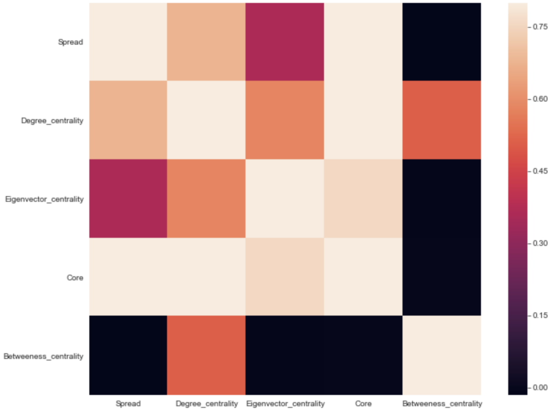
For a single initially infected node, this can be captured with centrality measures

Common centrality measures and calculation time:

- degree centrality: $O(1)$
- betweeness centrality: $O(mn)$
- eigenvector centrality: $\mathrm{O}(n^2)$
- k-core decomposition index: $O(m)$

Empirical prediction of influence function

We simulated starting an infection at 300 random nodes on the Facebook experiment network, and estimated the spreading influence function with Monte Carlo simulations



Empirical prediction of influence function

We performed a regression on spreading influence (estimated with Monte Carlo simulation) against those centrality measures.

$R^2 = \mathbf{0.85}$

OLS Regression Results

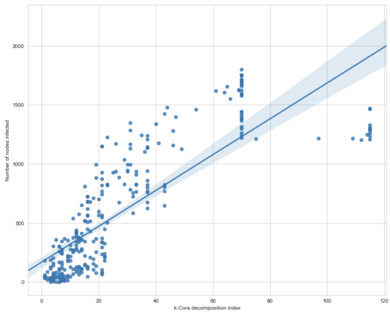
Dep. Variable:	Spread	**R-squared:**	0.848
Model:	OLS	**Adj. R-squared:**	0.846
Method:	Least Squares	**F-statistic:**	411.6
Date:	Tue, 07 May 2019	**Prob (F-statistic):**	2.55e-119
Time:	18:16:58	**Log-Likelihood:**	-142.56
No. Observations:	300	**AIC:**	295.1
Df Residuals:	295	**BIC:**	313.6
Df Model:	4		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	-1.2037	0.075	-16.138	0.000	-1.351	-1.057
Degree_centrality	0.0802	0.075	1.063	0.288	-0.068	0.229
Eigenvector_centrality	-0.6227	0.035	-17.759	0.000	-0.692	-0.554
Core	0.0419	0.002	16.938	0.000	0.037	0.047
Betweeneness_centrality	-0.0559	0.045	-1.246	0.214	-0.144	0.032

Empirical prediction of influence function

Result

Best predictor of spreading power of a node among common centrality measures: k-core decomposition index.



Empirical prediction of influence function

Definition

The k-core of a graph G is the largest induced subgraph of G in which every vertex has degree at least k.

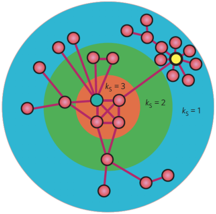


Figure: k-core decomposition of a small network

It is a good predictor of spreading power because it is both affected by local and global topology of a network at a node.

Outline

Heuristic 1: Naive Knapsack

We neglect interaction terms to approximate the influence function:

$$\sigma_{approx}(S, T) = \sum_{v \in S} \tilde{\sigma}(c_v, T) \quad (3)$$

Heuristic

$$\underset{S}{\text{argmax}} \sum_{v \in S} \tilde{\sigma}(v, T) \text{ s.t } \sum_{v \in S} c(v) < b \quad (4)$$

Advantage: Very quick to compute (linear time, dynamic programming)
Drawback: Tendency to produce naive solution (lots of close central nodes, with overlapping infection area)

Heuristic 2: Penalized Knapsack

We penalize choosing too close nodes, which might have overlapping infection zones:

Heuristic

$$\underset{S}{\text{argmax}} \sum_{v \in S} \tilde{\sigma}(v, T) + \lambda \sum_{v_1, v_2 \in S} dist(v_1, v_2) \text{ s.t } \sum_{v \in S} c(v) < b \tag{5}$$

i.e.

$$\text{argmax} \sum_{v \in V} \tilde{\sigma}(v, T) x_v + \lambda \sum_{v_1, v_2 \in V} dist(v_1, v_2) y_{v_1, v_2} \tag{6}$$

$$\text{s.t} \begin{cases} \sum_{v \in V} c_v x_v < b \\ \forall v_1, v_2 \in V, \quad y_{v_1, v_2} \leq x_{v_1} \\ \forall v_1, v_2 \in V, \quad y_{v_1, v_2} \leq x_{v_2} \\ \forall v, t \in V x_v, \quad y_{v,t} \in \{0, 1\} \end{cases} \tag{7}$$

Outline

Quick Recall : Submodularity

Definition

For any $R \subseteq S \subseteq V$ and $k \in V$, $k \notin S$, f is submodular if

$$f(S + \{k\}) - f(S) \leq f(R + \{k\}) - f(R) \qquad (8)$$

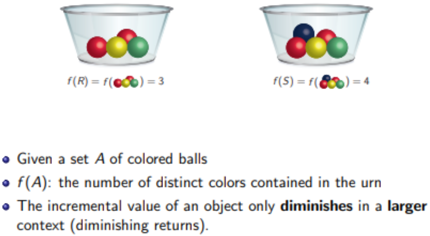


Figure: Submodularity example ([2])

Desired property for f : submodular and monotone

When f is both monotone and submodular, a greedy algorithm computes near-optimal with theoretical guarantee.

Modified Greedy Algorithm (1)

Algorithm 1 Modified greedy algorithm

$G \leftarrow \emptyset$
$U \leftarrow V$
while $U \neq \emptyset$ **do**
  $k \leftarrow \arg\max_{\ell \in U} \frac{f(G \cup \{\ell\}) - f(G)}{(c_\ell)^r}$
  $G \leftarrow G \cup \{k\}$ **if** $\sum_{i \in G} c_i + c_k \leq B$ **and** $f(G \cup \{k\}) - f(G) \geq 0$
  $U \leftarrow U \setminus \{k\}$
end while
$v^* \leftarrow \arg\max_{v \in V, c_v \leq B} f(\{v\})$
return $G_f = \arg\max_{S \in \{\{v^*\}, G\}} f(S)$

Figure: Modified Scaled Greedy Algorithm (Lin and Bilmes, 2010 [3])

Modified Greedy Algorithm (2)

Theorem

For normalized monotone submodular function f, Algorithm 1 with r=1 has a constant approximation factor as follows :

$$f(\hat{S}) \geq (1 - e^{-\frac{1}{2}}) f(S^*) \tag{9}$$

where S^* is an optimal solution (Lin and Bilmes, 2010 [3]).

Modified Greedy Algorithm (3) : "lazy" version

Another improvement in the algorithm : compute the argmax more efficiently using submodular property : **Accelerated Greedy Algorithm** [4]

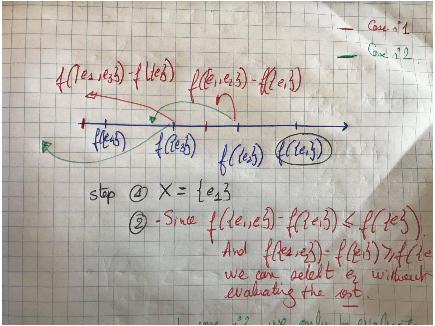


Figure: Simple example of the speed up in accelerated greedy

Outline

Submodular functions(1) : σ

σ is **submodular and monotone** [1], so we can optimize it with the modified greedy algorithm. But σ is hard to compute (simulated). Thus, we are looking for another monotone submodular function easier to compute for large networks.

Submodular function (2) : $\mathcal{F}_{net}$ (coverage/diversity function)

Inspired by the document summarization problem [2]...

Coverage/diversity function [2]

$$\forall S \subseteq V, \mathcal{F}_{net}(S) = \mathcal{L}(S) + \lambda \mathcal{R}(S) \tag{10}$$

$$= \sum_{i \in V} \min\{\sum_{j \in S} w_{i,j}, \alpha \sum_{j \in V} w_{i,j}\} + \lambda \sum_{i}^{K} \sqrt{\sum_{j \in P_i \cap S} \frac{1}{N} \sum_{i \in V} w_{i,j}} \tag{11}$$

$\mathcal{L}$ **represents the coverage of the network** by the subset S (relevance), and $\mathcal{R}$ **reward diversity**. The family $\{P_i\}_i$ is a partition of V (spectral clustering to create this partition). **The $w_{i,j}$ are cosinus similarity** between A_i and A_j where A is the adjacency matrix.

$\mathcal{F}_{net}$ is **submodular and monotone**, so we can optimize it with the modified greedy algorithm.

Outline

Details on the experiment

Now, the model we will consider is a simple continuous SIR model ($\forall i, \beta_i = \beta, \gamma_i = \gamma$). We first define a number of parameters before presenting the results.

- c (Cost function): The cost function $c : 2^V \to \mathbf{R}$ is define as the number of neighbors of each selected node : $\forall S \subseteq V, c(S) = \sum_{i \in S} deg(i)$
- B : We take the budget as the maximum of neighbors in the graph.
- N_0 (Number of iteration for simulation): We take 100 simulations to get a sufficiently good approximation of σ and to limit time computation.

Outline

Submodular functions and greedy algorithm : optimizing parameters

- We use a dataset from SNAP (facebook) and optimise λ (diversity importance) and α (saturation). We find $\lambda = 0.1$ and $\alpha = 0.7$.
- We also optimize with a grid-search the parameter r (scale of the return) for the greedy algorithm using $\mathcal{F}_{net}$ and find $r = 5$.
- We did not optimise r with the greedy algorithm using σ because it is too long (one optimisation over this graph take 11 hours approximately)

Computational time comparison

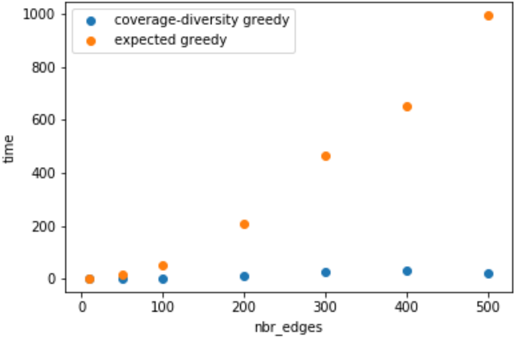


Figure: Time computation comparison between σ and $\mathcal{F}_{net}$ with accelerated greedy algorithm in function of the number of edges selected (SNAP youtube undirected graph)

Outline

Experiment on a real social network (1) : Dataset

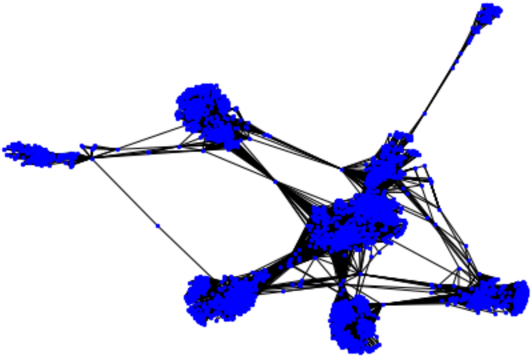


Figure: SNAP's facebook undirected graph

Experiment on a real social network (2) : results

	Random	Greedy σ	Greedy $\mathcal{F}_{net}$	Naive Knapsack	Penalized Knapsack
$\sigma(\hat{S}, T)$	88	264	455	237	331
I_0	24		372	174	253
Runtime	-	11 h	5 min	1s	4h

Table: Results on the SNAP facebook dataset (T=4)

- Why $\sigma(\hat{S}_\sigma) < \sigma(\hat{S}_{\mathcal{F}_{net}})$? The parameter r for the greedy algorithm had been only optimized for $\mathcal{F}_{net}$ (too long to compute for σ)
- Once optimized, the submodular optimization of $\mathcal{F}_{net}$ gives very good result, and is relatively cheap to compute (few min for this large network)
- $|\hat{S}_{\mathcal{F}_{net}}| = 372$: severalmicro-influencers is better than one macro-influencer.

Visualization for greedy $\mathcal{F}_{net}$

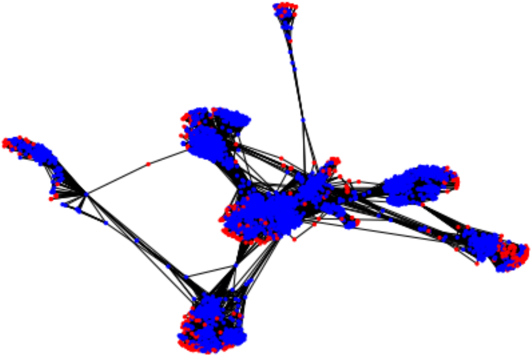


Figure: SNAP's facebook undirected graph. The red node are the nodes selected by greedy $\mathcal{F}_{net}$

Summary and discussion

- We design methods and function to maximize the contagion. $\mathcal{F}_{net}$ outperforms the rest (even for $\sigma(S,T) - S$)
- For the problem considered, it is better to diversify the budget in a lot of nodes.
- Continuation (1) : use a different model (β_i and γ_i functions of the degree)
- Continuation (2) : Simulations for a range of different parameter (T, budget B) and optimize r for greedy σ
- Continuation (3) : Reformulate the problem differently : $S^* \in \text{argmax}_{S \subset V} \, \sigma(S,T) - S \quad$ subject to $\quad c(S) \leq B$

For Further Reading I

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*. KDD '03. Washington, D.C.: ACM, 2003, pp. 137–146. ISBN: 1-58113-737-0. DOI: 10.1145/956750.956769. URL: http://doi.acm.org/10.1145/956750.956769.

Hui Lin and Jeff Bilmes. "A Class of Submodular Functions for Document Summarization". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*. HLT '11. Portland, Oregon: Association for Computational Linguistics, 2011, pp. 510–520. ISBN: 978-1-932432-87-9. URL: http://dl.acm.org/citation.cfm?id=2002472.2002537.

For Further Reading II

- Hui Lin and Jeff Bilmes. "Multi-document Summarization via Budgeted Maximization of Submodular Functions". In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. HLT '10. Los Angeles, California: Association for Computational Linguistics, 2010, pp. 912–920. ISBN: 1-932432-65-5. URL: `http://dl.acm.org/citation.cfm?id=1857999.1858133`.

- Michel Minoux. "Accelerated greedy algorithms for maximizing submodular set functions". In: *Optimization Techniques*. Ed. by J. Stoer. Berlin, Heidelberg: Springer Berlin Heidelberg, 1978, pp. 234–243. ISBN: 978-3-540-35890-9.