Online Influence Maximization

Siyu Lei, Silviu Maniu, Luyi Mo, Reynold Cheng, Pierre Senellart

KDD 2015, Sydney, Australia
August 13, 2015
Influence Maximization

Important problem in social networks, with applications in marketing, computational advertising

- **objective**: given a promotion budget, maximize the influence spread in the social network (word-of-mouth effect)

  select $k$ seeds (influencers) in the social graph, given an influence graph and a propagation model
Influence Maximization

Data model: influence graph \( G(V,E,p) \), where

- \( V \) and \( E \) and the vertices (users) and edges (follow relations, friendship, etc.) in the social network,
- \( p \) is a function mapping edges to influence probabilities.
Influence Maximization

Independent cascade model — a discrete time model of propagation:

• at time 0 — activate the seed s,

• node i activated at time t — influence is propagated at t+1 to neighbours j independently with probability p(i,j),

• once a node is activated, it cannot be deactivated or activated again.
Influence Maximization

The independent cascade model is a stochastic process.

Influence maximization in this model tries to optimize the expected influence spread, $\sigma(S)$, from a set of seeds $S$. 
Influence Maximization

Influence maximization is computationally hard — two sources of hardness:

• computing $\sigma(S)$ is hard = evaluating probability formulas

• even if we know $\sigma(S)$, computing the influence maximisation is NP-hard (submodular maximization subject to a constraint)

Solutions:

• for computing $\sigma(S)$: Monte Carlo simulations of influence spread

• for solving the influence maximization: greedy approximation algorithm

Multiple algorithms and estimators: CELF, TIM / TIM+
Online Influence Maximization (OIM)

What if we only know the social graph, but still want to maximize influence, with a budget?

• we need to keep an (uncertain) model of the influence graph

• classic trade-off between exploration (refine the model) and exploitation (use the model to maximize influence)

• lends itself to an iterative process over several rounds (online)
Online Influence Maximization Problem

Maximize the influence spread given a budget of $N$ rounds of choosing $k$ seeds in the network

- **Contribution**: an online framework — maximization and model refinement over multiple rounds
1: **Input**: # trials $N$, budget $k$, uncertain influence graph $G$
2: **Output**: seed nodes $S_n (n = 1 \ldots N)$, activation results $A$
3: $A \leftarrow \emptyset$
4: for $n = 1$ to $N$ do
5: $S_n \leftarrow \text{Choose}(G, k)$
6: $(A_n, F_n) \leftarrow \text{RealWorld}(S_n)$
7: $A \leftarrow A \cup A_n$
8: $\text{Update}(G, F_n)$
9: return $\{S_n | n = 1 \ldots N\}, A$
OIM Framework

Three ingredients:

• the model of the influence graph

• the explore-exploit strategy (Choose)

• after real-world feedback, update of the model (Update)
Uncertain Influence Graph

Probabilistic graph model:

- instead of a probability $p(i,j)$ on each edge $(i,j)$, we associate it with a distribution of probabilities
  \[
P(i, j) \sim \text{Beta}(\alpha_{ij}, \beta_{ij})
  \]

- by default, each edge is associated with a prior probability distribution $\text{Beta}(\alpha, \beta)$
Choose Strategies

The uncertain graph model allows us to explore different assumptions about the graph:

- **exploit** assumes that the influence probabilities are the expected value of $P(i,j)$

- **explore** uses either other assumptions about the graph, or uses heuristic strategies (random, max degree, degree discount)

For each branch, the IM algorithm is a black box (CELF, TIM, …) only the input influence graph is different.
Choose: Confidence Bound

A classic approach to use other assumptions about the influence graph is the Confidence Bound (CB) algorithm:

- each edge distribution is "moved" by $\theta$ standard deviations, and the IM algorithm is executed
- allows to "explore" other "possible influence graphs"
- exploit corresponds to the case where $\theta$ is 0

A probabilistic parameter $\epsilon$ allows the choice between different $\theta$ values (including 0 for exploit) — similar to $\epsilon$-greedy

1: **Input:** uncertain influence graph $G = (V, E, P)$, budget $k$
2: **Output:** seed nodes $S$ with $|S| = k$
3: for $e \in E$ do
4:   $\mu_{ij} \leftarrow \frac{\alpha_{ij}}{\alpha_{ij} + \beta_{ij}}$
5:   $\sigma_{ij} \leftarrow \frac{1}{(\alpha_{ij} + \beta_{ij})} \cdot \sqrt{\frac{\alpha_{ij} \beta_{ij}}{(\alpha_{ij} + \beta_{ij} + 1)}}$
6:   $p_{ij} \leftarrow \mu_{ij} + \theta \sigma_{ij}$
7: $G' \leftarrow G$, with edge probabilities $p_{ij}, \forall (i, j) \in E$
8: $S \leftarrow \text{IM}(G', k)$
9: return $S$
Choose: Confidence Bound

Advantages of CB:

- allows the update of $\varepsilon$ probabilities for a fixed choice of $\Theta$ values — Exponentiated Gradient (EG)

- using CB with EG allows a theoretical regret bound for a given choice of (constant) $\Theta$ values

```
1: Input: $\varphi$, probability distribution; $\delta$, accuracy parameter; $G_n$, the gain obtained; $j$, the index of latest used $\Theta_j$; $w$, a vector of weights; $N$, the number of trials.
2: Output: $\hat{\Theta}$
3: $\gamma \leftarrow \sqrt{\frac{\ln(q/\delta)}{qN}}$, $\tau \leftarrow \frac{4q\gamma}{3+\gamma}$, $\lambda \leftarrow \frac{\tau}{2q}$
4: for $i = 1$ to $q$ do
5:     $w_i \leftarrow w_i \times \exp\left(\lambda \times \frac{G_n \times \|i=j\|+\gamma}{\varphi_i}\right)$
6: for $i = 1$ to $q$ do
7:     $\varphi_i \leftarrow (1-\tau) \times \frac{\sum_{j=1}^{q} w_j}{\sum_{j=1}^{q} w_j} + \tau \times \frac{1}{q}$
8: return sample from $\hat{\Theta}$ according to $\varphi$ distribution
```
Real-World Feedback

Once a strategy has been chosen and a seed set identified:

• we test $S$ in the real-world (posting on Twitter, flyers in a city,...)

• in round $n$, we get activation feedback composed of activated nodes $A_n$, and feedback set $F_n$ — tuples $(i, j, a_{ij})$ for every affected edge
Update Step

Two approaches to Update:

• **local update**: each edge in the feedback is updated in a Bayesian manner

• **global update**: each edge in the graph is updated using methods such as maximum likelihood or least squares regression

• can also be combined
Local Update

Beta distribution is a conjugate prior of the Bernoulli distribution — the update is straightforward:

- **success** $a_{ij} = 1 \implies P_{ij} \sim \text{Beta}(\alpha_{ij} + 1, \beta_{ij})$
- **failure** $a_{ij} = 0 \implies P_{ij} \sim \text{Beta}(\alpha_{ij}, \beta_{ij} + 1)$
- same as **counting the number of successful and failed activations for each edge**
Global Update

Only using local update might be too sparse — especially for low influence probabilities, can lead to over reliance on the prior.

**Solution**: update also the prior for all edges, using all the feedback history
Global Update

**Ordinary Least Squares (LSE):** update via least squares estimation, from the formula of a spread of a node:

\[
\sigma_n(\{s\}) = 1 + \sum_{(s,i) \in E \atop i \notin A_n} p_{si} \times \sigma_n(\{i\}) + \sum_{(s,i) \in E \atop i \in A_n} p_{si} \times (\sigma_n(\{i\}) - 1)
\]

which leads to

\[
(|A_n| - 1)\beta = (1 - |A_n|)(t_s + 1) + (h_s + o_s)\hat{\sigma}_n - (h_{as} + a_s)
\]

\[
x_n\beta = y_n.
\]

\[
\hat{\beta} = (\bar{x} \cdot \bar{y}) / (\bar{x} \cdot \bar{x})
\]
Global Update

Maximum Likelihood (MLE): assume edges are independent:

\[ \mathcal{L}(F_n \mid \alpha, \beta) = \prod_{(i,j,a_{ij}) \in F_n} \frac{(\alpha + h_{ij})^{a_{ij}}(\beta + m_{ij})^{1-a_{ij}}}{\alpha + \beta + h_{ij} + m_{ij}}. \]

and the parameters can be estimated from

\[ \sum_{(i,j,a_{ij}) \in F_n, a_{ij}=1} \frac{1}{\alpha + h_{ij}} = \sum_{(i,j,a_{ij}) \in F_n, a_{ij}=0} \frac{1}{\beta + m_{ij}}. \]
Sampling Optimization

Even advanced algorithms rely on sampling for influence estimation — costly over multiple rounds

- incremental optimization approach — reuse of samples between rounds in little-affected parts of the graph

![Diagram](image-url)
Results: effectiveness of explore-exploit strategies

![Graph showing influence spread over trials for different strategies.](image)

- **Exploit**
- \( \epsilon \)-greedy
- **CB**

In Figure 4, we observe that CB+MLE is about 13% better than \( \epsilon \)-greedy by varying \( k \). This indicates that, with \( k = 1 \), the influence spread decreases. This is intuitive, as in the global check, for example, the running time for CB is about 38% faster than the one for \( \epsilon \)-greedy. When \( k \) increases, this framework, in all cases.

Another observation is that the improvement of CB+LSE is about 32% better than \( \epsilon \)-greedy by varying \( k \). This further strengthens the utility of using \( \epsilon \)-greedy for a general decrease in efficiency. We suggest to set a smaller value of \( k \) in practice. Figure 6b and Figure 3a together show a tradeoff that more trials are required to invest all budget, and so, TIM+ will depend on how much total time that the user can afford.

In Figure 5a, we compare three versions of the Influence Spread, \( k \), respectively. However, a smaller \( k \) is closer to the real graph, and thus, leads to a closer result to TIM+. When \( k \) increases, the non-incremental algorithm. First, for the case where \( k = 1 \), runs significantly slower than the one for \( k = 0 \). When \( k = 0 \), it achieves 20% improvement of \( \epsilon \)-greedy over \( k = 1 \). This indicates that more trials are required to invest all budget, and so, TIM+ will depend on how much total time that the user can afford.

In practice, the ratio is about 80% of the running time by re-using a sample. These two factors together make it impossible to re-use a sample. These two factors together make it impossible to re-use a sample. These two factors together make it impossible to re-use a sample. These two factors together make it impossible to re-use a sample.
Results: effectiveness of update methods

(a) Different updates

(b) Effect of priors

- CB+MLE
- CB+LSE
- CB+LOC
- CB+NO
Another direction is to increase the scalability of our methods; this can be achieved by utilizing IM methods (e.g., [28], [2]) that utilize community and topic information, and other influence propagation cues over time. We will extend our solution to handle other campaigns, especially when the influence probabilities are close to the oracle. To update the knowledge of the graph based on the feedback received from the real world, and showed experimentally that they are effective in longer campaigns. Even when the influence probabilities are significantly reduced, the running time of the algorithm for mining top-k influential nodes in mobile social networks, especially for a large dataset.

In the future, we will examine the scenario where budgets are total to get the result for DBLP.

Results: effectiveness versus heuristics

![Graph 1: Results for NetHEPT and DBLP](image1)

<table>
<thead>
<tr>
<th>Influence Spread</th>
<th>Trial (NetHEPT, k = 5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3,000</td>
<td>10</td>
</tr>
<tr>
<td>2,000</td>
<td>20</td>
</tr>
<tr>
<td>1,000</td>
<td>30</td>
</tr>
<tr>
<td>0</td>
<td>40</td>
</tr>
<tr>
<td>0</td>
<td>50</td>
</tr>
</tbody>
</table>

![Graph 2: Results for DBLP](image2)

<table>
<thead>
<tr>
<th>Influence Spread</th>
<th>Trial (DBLP, k = 5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 · 10^4</td>
<td>10</td>
</tr>
<tr>
<td>1.5</td>
<td>20</td>
</tr>
<tr>
<td>1</td>
<td>30</td>
</tr>
<tr>
<td>0.5</td>
<td>40</td>
</tr>
<tr>
<td>0</td>
<td>50</td>
</tr>
</tbody>
</table>

**Key:***
- **Real**
- **Random**
- **MaxDegree**
- **CB**
- **CB-INC**
We showed experimentally that explore–exploit based on the uncertainties over time), consider IM methods (e.g., [28], [27]) that utilize community and topic information, and other influence propagation models, such as linear threshold or credit distribution [12, 14, 26]. One direction is to increase the scalability of our methods; this may require distributed algorithms, such as distributed sampling. Another direction is to develop a new solution, where IM is performed in multiple trials, and we have proposed explore–exploit strategies for this problem.

9. CONCLUSIONS

In this paper, we examine how to perform influence maximization and we have proposed explore–exploit strategies for this problem.

Results: efficiency of sample reuse

Figure 7 and Figure 8 show representative results for CB-INC and we have proposed explore–exploit strategies for this problem.

Influence Spread

<table>
<thead>
<tr>
<th>Trial (NETHEPT, k = 1)</th>
<th>Random</th>
<th>MaxDegree</th>
<th>CB</th>
<th>CB-INC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 7: Effectiveness on other datasets

Running Time (in seconds)

<table>
<thead>
<tr>
<th>Trial (DBLP, k = 1)</th>
<th>Random</th>
<th>MaxDegree</th>
<th>CB</th>
<th>CB-INC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Running Time (in seconds)
Research Perspectives

• scalability is still a big issue in influence maximisation — even more so in the online setting

• adapting the framework to other influence models (threshold, credit distribution)

• learning also the influence model — do not rely on “synthetic” models such as independent cascade and threshold