

실시간으로 변화하는 동적인 소셜 네트워크 상에서

영향력 최대화 문제에 대한 연구

류준수[○] 김승욱

서강대학교

sok54833@sogang.ac.kr, swkim01@sogang.ac.kr

A Study on the Influential Maximization Problem
on Real-Time Dynamic Social NetworksJoonsu Ryu[○] Sungwook Kim

Sogang University

요 약

본 논문은 실시간으로 변화하는 소셜 네트워크 구조를 기반으로 수많은 노드 중 어떤 노드가 가장 영향력 있는 노드인지를 알아보는 영향력 최대화 문제를 살펴본다. 대부분의 영향력 최대화 문제에 대한 논문들은 노드들이 자신의 영향력이 변화하지 않는다고 가정하였으나, 실제로 현실 환경에서는 노드들의 영향력이 꾸준히 변동한다. 따라서 제안하는 방법은 매 시간 각 노드가 다른 노드에게 전파시킨 정보의 양을 기반으로 이전의 영향력에 비해 현재의 노드 영향력이 증가하는지, 또는 감소하는지를 학습을 통한 계산으로 영향력 최대화 문제에 대한 효율적인 해를 찾는다. 본 논문의 결과는 시뮬레이션을 통해 그 유효성을 입증하였다.

I. 서 론

사람은 살면서 수많은 사람을 만나게 된다. 인터넷의 발전으로 이제는 사람을 인터넷에서도 만나게 되고, 그들 간의 관계를 다양하게 구축하게 되었다. 소셜 네트워크란, 이런 사람들의 상호작용이나 관계를 나타낸 그래프를 일컫는다.[1]

소셜 네트워크의 그래프는 노드(Node)와 엣지(Edge)로 이루어져 있는데 노드는 사람이나 단체, 엣지는 노드 간의 관계를 의미한다. 서로 다른 두 노드 사이에 엣지가 있다는 것은, 그 두 노드는 서로 각각 영향을 받는다는 것을 의미한다. 이렇게 서로 영향을 주는 구조에서 소수의 몇몇은 다수의 사람에게 큰 영향을 줄 수 있다. 가령 유명 연예인들이나, 정치인들은 일반 사람들에 비해 다른 사람들에게 영향을 주기 쉽다. 이러한 사람들을 영향력(Influential)이 큰 사람이라 한다.

소셜 네트워크에서 과연 어떤 노드 혹은 노드의 집합의 영향력이 최대인지 찾는 것은 다양한 분야에서 관심을 보이고 있다. 가령 최근 신제품을 홍보하는 데 바이럴 마케팅(Viral Marketing)이 많이 쓰임에 따라 한정된 예산으로 최대한 많은 사람들에게 홍보하기 위해 유명 연예인들이나 블로거에게 신제품에 대한 홍보를 부탁하는 것이 대표적인 예이다.

이로 인해, 컴퓨터 공학 분야에서는 영향력을 찾는 방법을 지속적으로 연구하고 있다. 대표적으로 Independent Cascade Model [2], Linear Threshold Model [3] 등이 존재한다.

그러나 이러한 방법들의 문제점은, 소셜 네트워크에서 노드의

영향력이 변할 수 있다는 것을 가정하지 않은 것이다. 가령, 어떤 연예인이 엄청난 인기로 인해 사람들에게 큰 영향력을 행사하다가 불미스러운 사유로 인기가 떨어지게 되면 더 이상 일반사람들에게 큰 영향을 끼치지 못하게 되고, 또 혜성처럼 등장한 신인이 갑자기 유명해져서 사람들에게 큰 영향을 끼칠 수도 있다. 이러한 경우 시간에 따라 영향력 집합이 계속 변하게 되는데, 현재까지 연구된 기존의 방법들은 이러한 상황을 반영하고 있지 않다.

본 논문에서는 이러한 상황을 가정하여, 모든 노드의 영향력은 실시간으로 변동하여 새로이 계산하여 매 시간마다 일정 크기의 영향력 집합을 결과물로 만들어내는 방법을 제시하고, 시뮬레이션에서 이 방법이 얼마나 효과가 있는지 밝혀내도록 한다.

II. 본 론

소셜 네트워크를 나타낸 무방향 그래프를 $G=(N, e)$ 로 정의한다. 여기서 N 은 노드의 집합을 의미하고, e 는 엣지의 집합이다. 엣지가 의미하는 것은 영향력을 찾는 모델에 따라 다르지만, 여기서는 Independent Cascade Model의 의미를 차용하도록 한다. 즉, $e(u, v)$ 가 의미하는 것은 노드 u 가 노드 v 에게 자신의 정보를 전파시킬 확률이다.

기존의 논문들에서는 이 $e(u, v)$ 의 값을 무작위로 설정하거나, 통계를 이용한 값으로 넣고 영향력 노드를 찾는 방법을 사용하였다. 그러나 현실적으로, 어떤 노드가 다른 노드에게 정보

를 전파시킬 확률은 고정적이지 않다. 친해서 계속 정보를 주고받던 사람과 연락이 소원해져 영향력이 작아지거나, 또는 그렇게 친밀하지 않았는데 어떤 일을 계기로 관계가 친밀해져서 영향력이 늘어나는 경우가 잦다. 따라서 본 논문에서는 이러한 현실적인 배경을 반영하여, $e(u, v)$ 을 시간에 따라 변하는 값으로 정의할 것이다.

일반적으로, 특정한 노드가 어떤 노드로부터 영향을 받았는지는 알 수 있으나 (예를 들어 페이스북의 공유하기, 트위터의 리트윗 등) 정확한 영향력(즉, 정보가 전파될 확률 $e(u, v)$)을 알 수는 없다. 따라서 본 논문은 이를 이전 시간까지의 누적된 정보의 전파량으로 $e(u, v)$ 값을 근사적으로 계산했으며, 아래의 식 (1)과 같다.

$$e_{t+1}(u, v) = (1 - \alpha)e_t(u, v) + \alpha \frac{diff_u(t)}{\max_w diff_w(t)} \quad (1)$$

식 (1)에서 $e_t(u, v)$ 는 t 시간에서 노드 u 가 노드 v 를 전파시킬 확률이다. t 는 이산적인 시간으로, $0, 1, 2, \dots$ 등의 값을 가질 수 있다. $diff_u(t)$ 는 t 시간에서 노드 u 의 정보를 성공적으로 전파한 횟수이다. 따라서 $diff_u(t) / \max_w diff_w(t)$ 는 전체 노드가 정보를 성공적으로 전파한 횟수와 노드 u 가 정보를 성공적으로 전파한 횟수의 비율이다. $\alpha (0 \leq \alpha \leq 1)$ 는 이전 시간의 e 값과, 새로 반영되는 $diff_u(t) / \max_w diff_w(t)$ 값의 비율을 조절하기 위한 변수로, α 의 값이 1에 가까울수록 새로 반영되는 값을 많이 반영하게 되고, 0에 가까울수록 기존의 값을 많이 반영한다는 의미이다.

식 (1)은 시간이 증가할수록 반복적으로 계산하는 식이므로, 초기값인 $e_0(u, v)$ 의 값이 필요하다. 초기에는 그래프에 대한 정보밖에 없으므로, 아래 식 (2)와 같이 설정하였다.

$$e_0(u, v) = (1 - \beta) \frac{\deg_u}{\deg_{\max}} + \beta \prod_{w \in \text{neigh}(u)} \frac{\deg_w}{\deg_{\max}} \quad (2)$$

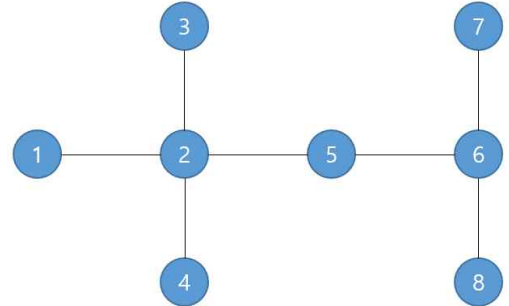
식 (2)에서 \deg_u 는 노드 u 의 degree를, $\text{neigh}(u)$ 는 u 의 이웃 노드를 의미한다. 그러므로 식 (2)는 전체 그래프에서 노드 u 의 degree 비율과 노드 u 의 이웃 노드들의 degree 비율을 $\beta (0 \leq \beta \leq 1)$ 만큼 나누어 합산한 것이다. β 값이 1에 가까울수록 해당 노드의 degree를 많이 반영하며, 0에 가까울수록 이웃 노드들의 degree를 많이 반영한다는 의미이다.

본 논문에서는 위와 같이 설계한 소셜 네트워크 정보 전파 모델을 시뮬레이션을 통해 실제로 각 노드들의 유동적인 영향력이 $e(u, v)$ 에 실시간으로 반영되는지 확인하였다.

III. 실험결과

시뮬레이션에 사용한 소셜 네트워크 그래프는 아래 그림 1과 같다. 이 그래프를 $G = (N, e)$ 라 하면 $N = \{1, 2, \dots, 8\}$ 이다. 그림에서 보이는 엣지는 총 7개이지만, 이 그래프는 무방향 그

래프이기 때문에 14개로 설정한다. 예를 들면, 노드 1과 노드 2 사이의 엣지는 $e(1, 2)$ 와 $e(2, 1)$ 로 구성되도록 한다. 소셜 네트워크에서는 서로간에 영향을 받을 수 있으므로 이렇게 무방향 그래프로 정의하였다.



(그림 1) 시뮬레이션에 사용한 소셜 네트워크

위에서 설명한대로 각 엣지의 초기값을 먼저 계산해야 한다. 시뮬레이션에서는 식 (1)의 $\alpha = 0.5$, 식 (2)의 $\beta = 0.5$ 로 대입해서 계산하도록 한다.

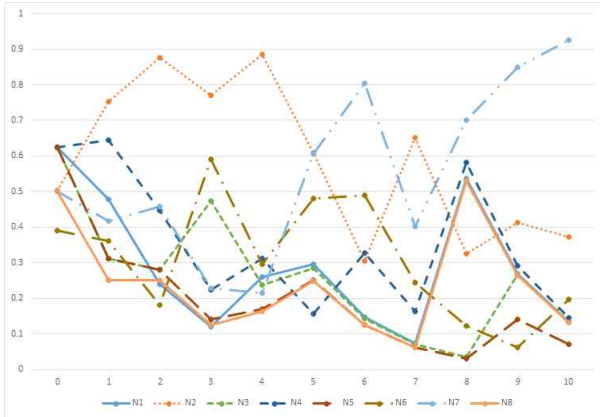
따라서 $e_0(1, 2) = 0.5 \times 0.25 + 0.5 \times 1 = 0.625$ 가 된다.

이 후, 실제로 각 노드가 매 시간 전파시킨 정보를 기반으로 매 시간 반복하여 $e_t(u, v)$ 를 계산해야 한다. 실험의 편의상 t 는 0부터 10까지 만으로 한정하고, 매 시간 각 노드가 전파시킨 정보의 양은 아래의 표 1처럼 정해 놓는다. 노드가 총 8개이므로, 각 항목의 최대값은 7이 된다.

t \ N	1	2	3	4	5	6	7	8	9	10
1	1	0	0	2	1	0	0	1	0	0
2	3	4	2	5	1	0	1	0	2	1
3	0	1	2	0	1	0	0	0	2	0
4	2	1	0	2	0	1	0	1	0	0
5	0	1	0	1	1	0	0	0	1	0
6	1	0	3	0	2	1	0	0	0	1
7	1	2	0	1	3	2	0	1	4	3
8	0	1	0	1	1	0	0	1	0	0

(표 1) 매 시간 각 노드가 전파시킨 정보의 양

위의 (표 1)에 해당하는 값을 매 시간 적용했을 때 변화하는 엣지의 값은 아래의 (그림 2)를 통해 확인할 수 있다.



Competitive Influence in Social Networks", International Workshop on Internet and Network Economics 2010. pp. 539-550.

(그림 2) 시뮬레이션 결과로 나온 시간대별 노드의 영향력

(그림 2)의 그래프를 보면 초기에는 degree만을 반영해 엣지의 값이 정해지지만, 매 시간 비교해서, 이전 시간에 전파한 자신의 정보의 양이 많을 수록 노드의 영향력이 점차적으로 증가하고, degree가 높아 초기 엣지의 값이 높더라도 전파력이 낮은 노드의 엣지값은 시간이 지날수록 낮아짐을 확인할 수 있다.

위의 시뮬레이션 결과를 기반으로 만약 해당 소셜 네트워크에서 가장 영향력이 높은 노드를 시간대별로 찾아낸다면 $t=5$ 에서는 노드 2(N2)이지만, $t=10$ 에서는 노드 7(N7)로 바뀌었음을 알 수 있다.

IV. 결 론

본 논문에서는 소셜 네트워크에서 정보를 전파하는 모델인 Independent Cascade Model을 보완하여 실시간으로 변동하는 노드의 영향력을 반영한 새로운 모델을 제시하였다. 또한 실험을 통해 이 방법이 노드가 퍼트린 정보가 많을수록 노드의 영향력이 엣지에 잘 반영되는 것을 확인하였다.

ACKNOWLEDGMENT

This work was supported by the Sogang University Research Grant (201719022.01)

참 고 문 헌

- [1] David K., Jon K., Eva T. "Maximizing the Spread of Influence through a Social Network", Proceeding of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 137-146
- [2] Dunia L. P. "Contagion and Coordination in Random Networks", International Journal of Game Theory. pp. 371-381.
- [3] Allan B., Yuval F., Joel O. "Threshold Models for