노드 근접도를 이용한 커뮤니티 통합 알고리즘 연구

전병현, 한치근
경희대학교 컴퓨터공학과

e-mail:(bhjun, cghan)@khu.ac.kr

A Study on a Community Integration Algorithm Using Vertex Betweenness

Byung-Hyun Jun, Chi-Geun Han
Dept of Computer Engineering, Kyung Hee University

요 약

SNS의 활용에 따라 대규모의 네트워크가 생성되고 있고, 네트워크에서 생성되는 커뮤니티의 구조를 파악하기 위한 많은 연구가 진행되고 있다. 파악된 커뮤니티를 통합하는 방안은 서로 다른 커뮤니티를 통합할 필요가 있을 때 활용된다. 본 연구에서는 파악된 2개의 커뮤니티를 하나의 커뮤니티로 만들기 위해 필요한 에지추가 방법을 연구한다. 파악된 커뮤니티를 통합하기 위해 각 커뮤니티를 연결하여야 하는데, 이 때 각 커뮤니티 내의 노드들에 대해 노드 근접도를 계산하여 커뮤니티의 중심을 찾는다. 중심에 가까운 노드들을 순차적으로 이용하여 커뮤니티가 통합될 때까지 에지를 생성하여 그래프에 추가한다.

1. 서론

사회학 연구나 생물학 연구에서 사회에 참여하는 객체간의 관계를 확인하여 존재하는 그룹을 파악하는 연구가 1960년대부터 진행되었다. 그리고 최근 SNS 활용에 따라 대규모의 네트워크가 생성되고 있는데, 동일한 취미, 관심사항을 갖고 있거나, 동일한 출신학교, 출신지역, 거주지역 등에 따라 유대관계를 갖게 되면, 자연스럽게 정보의 교류 등이 발생하게 된다. 이 관계(following 또는 이메일교환, facebook view 등)는 네트워크에서 두 노드(참여자) 사이의 에지(edge) 형태로 나타나게 된다. 네트워크 내에서 다른 집단보다 많은 연결(connection, edge)을 갖는 집단은 그렇지 않은 집단보다 높은 연결성을 갖게 되는데, 이러한 집단을 커뮤니티(community)(클러스터, 그룹, 모듈)라고 한다. 커뮤니티의 정의는 '그 집단 내(within)의 연결이 집단 간(inter)의 연결보다 많은 집단' 이라는 비정형적인 문구로 정의된다[1].

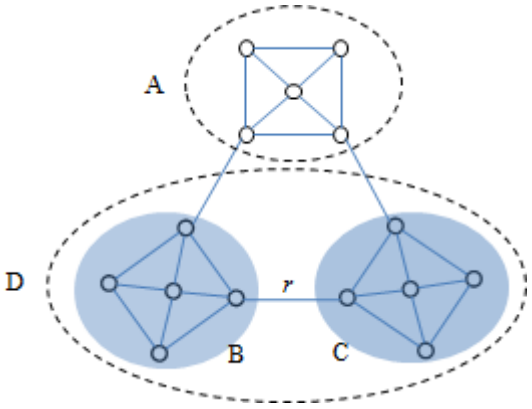
주어진 네트워크에서 존재하는 커뮤니티를 찾게 되면, 그 네트워크의 속성을 정확히 이해할 수 있게 된다. 최근에는 SNS, 웹상에서 생성되는 대규모 네트워크 내에 존재하는 커뮤니티를 찾는 연구가 많이 진행되었다. 그 방법으로는 분할방법, modularity에 기반을 둔 방법[2, 3], spectral 방법, 동적방법, 통계적인 방법 등으로 크게 나눌 수 있다[1].

본 연구에서는 커뮤니티들을 포함한 네트워크에서 2개의 특정 네트워크를 하나의 커뮤니티로 통합하기 위한 방법을 연구한다. 구체적으로, 노드 근접도를 이용하여 네트워크에 존재하지 않는 에지를 네크워크에 추가하여, 기존의 두 커뮤니티를 하나의 커뮤니티로 통합하는 방법을 제시하고, 알고리즘 설계에 고려하여야 할 문제점에 대해 설명한다.

2개의 커뮤니티를 통합하는 방법을 순차적으로 적용하여 2개 이상의 커뮤니티 통합에 활용할 수 있으므로, 본 연구에서는 2개의 커뮤니티 통합방법만 고려하기로 한다.

2. 연구 배경

하나의 객체는 그래프에서 하나의 노드(node)로 표시되고, 두 객체간의 연결은 그래프에서 하나의 에지(edge)로 표시된다. 본 연구에서는 비가중(unweighted) 그래프를 가정한다. [그림 1]은 일반적인 네트워크를 그래프로 나타내고, 그 그래프 내에 존재하는 커뮤니티를 나타낸 예이다.



[그림 1] 그래프 내에 존재하는 커뮤니티의 예

총 15개의 노드, 27개의 에지가 있는데, 이 그래프는 커뮤니티 A, B, C로 확연히 구분할 수 있다. 즉, 각 커뮤니티 내의 연결이 커뮤니티 간의 연결보다 상대적으로 많다는 것을 직관적으로 확인할 수 있다.

존재하는 다양한 커뮤니티 발견(detection) 알고리즘들을 이용하여, 주어진 그래프에서 커뮤니티들을 확인할 수 있다. 본 연구는 '확인된 커뮤니티들 중 일부를 하나의 커뮤니티로 만들기 위해서는 어떻게 해야 하나'에서 출발하고 있다. 즉, [그림 1]에서 커뮤니티 B와 C를 통합하여 하나의 커뮤니티 D를 만드는 방법을 설계하는 것이 본 연구의 목표이다.

본 연구는 조직 내에 분리되어 있는 두 그룹을 통합하여 하나의 그룹으로 만들 때 활용할 수 있다. 예를 들어 회사 내의 친목조직, 연구자 그룹 간의 정보교류가 원활하여야 할 경우나 웹페이지들의 연결이 이루어져야 할 경우 등, 커뮤니티가 통합되어야 하는 경우 효과적으로 응용될 수 있다.

커뮤니티 D를 만들기 위해서는 기존에 존재하지 않던, 새로운 에지들을 그래프에 추가하여야 한다. 여기에서 다음과 같은 이슈가 발생한다.

(1) 몇 개의 에지를 추가하여야 하나?

(2) 어느 노드 간에 에지를 추가하여야 하나?

(3) 새로운 에지의 추가로, 기존의 커뮤니티 A와 새로 통합되는 커뮤니티 D의 구성노드들은 달라지지 않는가? (A, D의 구성노드가 일부 변경되는 것을 허용할 수도, 허용하지 않을 수도 있다.)

물론, 커뮤니티 B와 C를 상호 모두 연결하면(5×5=25개의 에지 추가) 하나의 커뮤니티로 인식되는 데는 문제가 없다. 그러나 본 연구에서는 커뮤니티로 인식되기 위해 필요한 최소의 추가 에지수를 구하고자 한다.

3. 에지 근접도를 이용한 커뮤니티 발견 알고리즘

에지 근접도(edge betweenness: EB)는 해당 에지가 그래프의 모든 노드 쌍의 최단경로들에 사용되는 회수로 정의된다. 두 노드 간에 최단경로가 복수개가 있을 경우, 각 경로는 동일한 가중만큼 경로를 구성하는 에지의 에지 근접도를 증가시킨다[2]. 에지 근접도를 이용하여 커뮤니티를 발견하는 알고리즘은 '에지 근접도가 높은(노드 간의 최단 경로에 여러 번 사용되는 경우) 에지는 커뮤니티 사이에 존재할 가능성이 많다'라는 성질을 이용한다. 즉, 두 커뮤니티 사이에 존재하는 에지를 연쇄적으로 제거함에 따라 커뮤니티의 구조를 확인할 수 있다는 것이다. 다음은 에지 근접도를 이용한 커뮤니티 발견 알고리즘(CDEB로 칭함)의 단계이다[4].

[단계 1] 네트워크 내의 모든 에지에 대해 에지 근접도를 계산한다.

[단계 2] 가장 큰 에지 근접도를 갖는 에지를 제거한다.

[단계 3] 에지의 제거에 따라 달라지는 에지 근접도를 다시 계산한다.

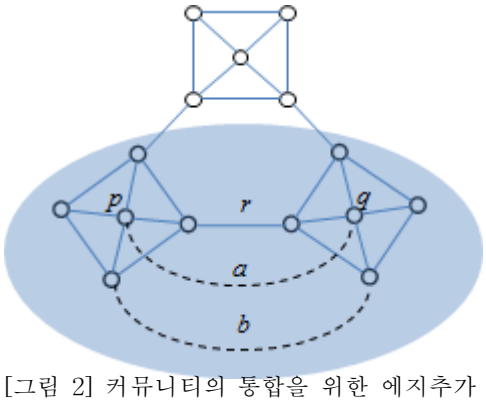
[단계 4] 모든 에지가 제거될 때 까지 [단계 2]를 수행한다.

그래프의 에지의 개수가 m, 노드의 개수가 n일 때 이 알고리즘은 $O(m^2n)$의 시간복잡도를 갖고 있다.

4. 커뮤니티 통합 알고리즘

커뮤니티 B와 C를 통합하여 커뮤니티 D를 만들기 위해서는 B와 C 사이에 존재하는 에지([그림 1]의 에지 r)의 EB값을 줄여줄 필요가 있다. 왜냐하면, B와 C간의 최단경로는 반드시 r을 거쳐야 하므로, r은 큰 EB값을 갖게 되고, CDEB 알고리즘의 초기단계에서 r이 삭제됨에 따라, 커뮤니티 B와 C가 구분되는 효과가 발생한다. 따라서 r의 EB값을 효과적으로 감소시켜 r이 가능한 오래 존재할 수 있도록 만드는 새로운 에지의 추가방법을 모색하여야 한다.

그러면, 커뮤니티의 통합을 위해서 어떠한 성질의 에지를 추가할 수 있는가? [그림 1]에서 커뮤니티 B와 C를 통합하기 위해 추가하는 에지의 유형은 [그림 2]에서 a, b로 나타내었다.



[그림 2] 커뮤니티의 통합을 위한 에지추가 유형

에지 a는 커뮤니티 B와 C의 각각 중심에 해당하는 노드 p, q를 연결하는 에지이고, 에지 b는 중심이 아닌 노드들 간을 연결하는 성질을 갖고 있는 에지이다.

커뮤니티의 중심을 설명하기 위해, 노드 근접성(vertex betweenness: VB)을 정의한다. VB는 해당 노드가 그래프의 모든 노드 쌍의 최단경로들에 사용되는 회수로 정의되며[5], 그래프의 중심으로 간주할 수 있다.

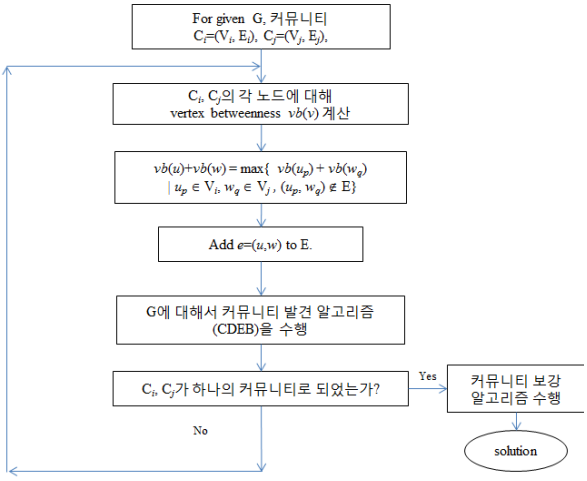
결론적으로 에지 a를 추가함에 따라 에지 r의 EB를 효과적으로 감소시킬 수 있다. 그 이유는 에지 a가 커뮤니티의 B와 C의 중심 p, q를 연결해 주고 있으므로, 에지 a

의 추가에 따라 B와 C에 있는 노드 쌍의 최단경로로 에지 r이 사용되는 경우를 급격하게 감소시킬 수 있기 때문이다. 에지 b보다 커뮤니티 B와 C간의 최단경로로 에지 a가 더 많이 사용될 수 있다는 것은 쉽게 관찰할 수 있다. 즉, 커뮤니티 B와 커뮤니티 C에 있는 각 노드의 VB를 계산하여, 커뮤니티 B에 속한 노드 하나와 커뮤니티 C에 속한 노드 하나를 선발하여 에지를 연결할 때, 두 노드의 VB 값 합이 큰 순서대로 고려하는 것이 효과적이라는 것을 알 수 있다.

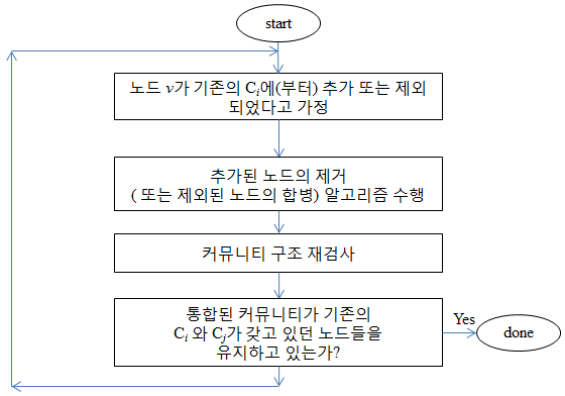
VB를 계산할 때, 각 커뮤니티를 구성하는 노드, 에지들로 이루어진 독립적인 그래프에서 값을 계산한다. 그렇게 해야만, 커뮤니티 내의 중심을 구할 수 있게 된다. 즉, 주어진 그래프가 G=(V, E)일 때, 커뮤니티 i는 $C_i=(V_i, E_i)$, $V=\cup_{i=1,k}V_i$, $E=\cup_{i=1,k}E_i \cup E_r$, k는 G가 갖고 있는 커뮤니티의 개수, E_r은 어느 커뮤니티에도 속하지 않는 에지들의 집합을 나타낸다. 커뮤니티 i와 j를 통합하고자 하면, C_i, C_j에 대해 이들에 속한 노드들의 VB값을 계산하게 된다.

에지 하나를 추가하고 난 후, CDEB를 수행하여 네트워크의 커뮤니티를 확인하여야 한다. 원하는 대로 커뮤니티 B와 C가 통합될 수도 있고(이 경우 종료), 아직 통합이 완료되지 않을 수 있다. 통합이 미완성일 경우, 추가적인 에지 선정 작업을 수행한다. 또한, 에지의 새로운 추가에 따라 커뮤니티 B와 C의 구성 노드들이 바뀔 수 있는 가능성이 있으므로, 통합이 이루질 경우, 커뮤니티의 구조 변경을 확인하는 절차가 필요하다.

다음 [그림 3]은 커뮤니티 통합 알고리즘(CI라 칭함)의 흐름도를 보여 주고 있다.



[그림 3] 커뮤니티 통합 알고리즘(CI)의 흐름도



[그림 4] 커뮤니티 보강 알고리즘 흐름도

[그림 4]에 포함되어 있는 추가되거나, 제외된 노드를 처리하는 알고리즘의 기본적인 개념은 해당 노드의 VB값을 이용하여 커뮤니티를 재구성하는 것인데, 이에 대한 구체적인 설명은 생략하도록 한다.

4. 결론

본 연구에서는 노드 근접성을 이용하여 추가되는 에지를 생성하여 커뮤니티를 통합하는 방안을 제시하였다. CI 알고리즘의 기본 개념을 설명하고, 이 알고리즘이 작동하는 기본원리를 설명하였다. 커뮤니티를 발견하는 알고리즘이 달라지게 되면, 그에 따라 커뮤니티를 통합하는 또 다른 방법을 설계할 수 있다. 예를 들어, modularity를 이용하여 커뮤니티를 발견할 때는, modularity 값을 이용하여 추가 에지를 선정할 수 있다.

향후 CI를 구현하여, 본 알고리즘의 정확성을 확인할 예정이다. 기존의 커뮤니티 발견 알고리즘에 사용되는 예의 네트워크를 이용하여, 커뮤니티 통합이 효과적으로 수행되는지를 확인하고자 한다.

참고문헌

[1] Fortunato, Santo, "Community Detection in Graphs", Physics Reports 486, pp. 75-174, Jan. 2010.

[2] Blondel1, V. D., Guillaume, J-L., Lambiotte, R. and Lefebvre1, E., Fast Unfolding of Communities in Large Networks, Journal of Statistical Mechanics: Theory and Experiment, P10008, 2008.

[3] Newman, M.E.J., Fast Algorithm for Detecting Community Structure in Networks, Phys. Rev. E 69, 066133, 2004.

[4] Girvan, M. and Newman,M.E.J., Proceedings of the National Academy of Sciences 99(12), pp. 7821-7826, June 2011.

[5] Freeman, L. C., A Set of Measures of Centrality Based on Betweenness, Sociometry 40, pp.35-41, (1977).