커뮤니티 검출 알고리즘의 비교

Girvan-Newman method, CNM(Clauset-Newman-Moore) method, Louvain method 비교

2015-11923 최한결

1. Introduction

현실 세계에서는 서로 다른 분야의 복잡한 시스템이 네트워크를 형성한다. 그 중에는 소셜 네트워크, 웹 페이지, 운송 네트워크, 인용 네트워크, 대사 네트워크, 단백질 상호작용 네트워크, 유전자 네트워크 등이 있다. 네트워크는 신경과학에서부터 사회학, 경제, 정보기술에 이르기까지 인간 존재의 거의 모든 영역에 존재한다. [1]
그래프 이론은 거의 300년이 지났지만 복잡한 네트워크에 대한 연구는 10년이 조금 넘은 젊은 학문이다. 소셜 네트워크의 커뮤니티는 유사한 선호나 관심사를 가진 사람들을 대표할 수 있다. 또한 질병의 확산을 막기 위해 지역사회와 사람들 사이의 연관성을 인식하는데 사용할 수 있다. [2] 이러한 예로부터, 복잡한 네트워크에서 커뮤니티의 탐지는 다양한 과학 분야에서의 광범위한 응용을 가지고 있는 것이 분명하다. 최근 몇 년 간 네트워크의 크기가 급속도로 커지고 있고 데이터가 큰 만큼 효과적인 커뮤니티 검출 알고리즘을 찾는게 중요하다. 따라서 우리는 각 다양한 커뮤니티 검출(community detection) 알고리즘을 수행하고 성능을 비교해보았다. 사용한 알고리즘은 Girvan-Newman method, CNM(Clauset-Newman-Moore), Louvain method 세 가지이다.

2. Description of methods, measures (metric)

2.1. Method

2.1.1. Modularity

각 커뮤니티 검출 알고리즘의 성능을 분석하기 위해 modularity를 사용하였다. Modularity는 clustering의 지표로서 커뮤니티가 없는 랜덤 그래프의 평균 edge의 수와 비교해 각 커뮤니티들에 얼마나 많은 Edge

가 존재하는지를 나타낸다. 이를 통해 그래프의 clustering이 잘 되었는가를 판단할 수 있는 근거가 된다.
즉 Graph가 S={$s_1, s_2, s_3, \dots, s_k$}로 partition되어있다면 Modularity는 각 노드의 degree를 유지한 채로 랜덤하게 연결된 네트워크에서 두 노드 간의 예상 edge 수를 계산할 수 있다. 이는 각 노드 i, j에 대해 $\frac{k_i k_j}{2m}$ 이다. (m = |E|, k_i = i의 degree) 이 때 Modularity Q는 다음과 같고 [-1,1]의 범위를 가진다.

$$Q = \sum_{s \in S} (Actual\ number\ of\ edges\ within\ group\ s - E[edges\ within\ group\ s])$$
$$= \frac{1}{2m} \sum_{s \in S} \sum_{i \in s} \sum_{j \in s} (A_{ij} - \frac{k_i k_j}{2m})$$

2.1.2. Girvan-Newman algorithm

Girvan-Newman method는 완성된 그래프로부터 시작해 각 Edge의 betweenness centrality를 기준으로 하여 edge를 제거해나가는 Divisive method이다. 기존 graph에서 각각 edge의 betweenness centrality를 모두 구한 후 betweenness centrality가 높은 edge부터 차례대로 제거해나가면 edge가 줄어듬에 따라 edge로 연결되지 않은 community가 형성된다.

betweenness centrality는 각 노드가 얼마나 이웃들과 연결되어 있는지 판단하는게 아닌 전체 네트워크에서 노드들과 얼마나 잘 연결되어 있는지를 나타낸다. 두 지점 I,j에 대해 i에서 j로의 경로에 어떤 edge k를 항상 지나야 한다면 edge k는 두 지점을 연결하는데 중심이 된다. 이를 수치화 시킨 것이 betweenness centrality이다. Betweenness는 네트워크 모든 노드쌍에 대하여 최단거리 루트를 구하고 그 안에 등장하는 edge의 빈도를 구해 수치화한다.

```
betweenness=[0,0,…,0](size=m)
For s in G.nodes:
    add BFS(G,s) for each edge in Betweenness
```

Textbox 1 Betweenness를 구하는 pseudo code

이를 통해 각 Step마다 betweenness를 구하고 가장 betweenness가 큰 edge부터 차례로 제외시키면 점점 작은 크기의 네트워크로 나누어진다.

```
While G.edges is not empty:
    highest_edge <- index of
max(Betweenness(G))
    G.remove(highest_edge)
```

Textbox 2 Girvan-Newman method pseudo code

betweenness를 구하는 수행시간이 O(n(m+n))이고 그래프의 Edge만큼 반복하므로 Girvan-Newman method의 총 수행시간은 O(mn(m+n))이다.

2.1.3. CNM(Clauset-Newman-Moore) algorithm

Girvan-newman method와는 반대로 독립 노드로부터 그래프의 edge들을 하나씩 연결해 나가며 Community Detection을 하는 Agglomerative method이다. 모든 edge를 하나씩 연결해보고 modularity를 비교한다. 가장 modularity가 높아지는 edge를 그래프에 포함시킨다. 이를 반복하면 node개수만큼 존재하던 community들이 존재 합쳐지면서 community의 개수가 점점 줄어들다 모든 edge가 연결되게 된다. 이 중 가장 높은 modularity를 가지는 그래프를 추출한다.

```
G = isolated nodes graph
remain_edges = edges of graph
while remain_edge is empty:
    For i in remain_edges:
        g = G.copy
        g.add_edge(i)
        if m < g.modularity():
            m = g.modularity()
            m_max_edge = i
    remain_edge.remove(m_max_edge)
    G.add_edge(m_max_edge)
```

Textbox 3 CNM(Clauset-Newman-Moore) method pseudo code

Modularity를 구할 때 각 노드쌍에 대해 연산한 번이므로 modularity의 시간복잡도는 $O(n^2)$이다. 이것이 $\frac{m(m+1)}{2}$ 번 반복되므로 CNM의 시간복잡도는 $O(m^2n^2)$이다.

2.1.4. Louvain algorithm [3]

Louvain 알고리즘은 Phase1 과 Phase2 로 이루어진다. 우선 Phase1 에서는 각 노드별로 인접한 community 에 노드를 재배치 하여 modularity 의 변화량을 구한다. 이후 modularity 의 변화량 ΔQ 가 가장 큰 community 로 그 노드를 포함시킨다. 이는 CNM 알고리즘과 비슷하지만 Louvain algorithm 에서는 총 그래프의 modularity 를 구하지 않고 modularity 의 변화량을 다음과 같이 정의한다.

$$\Delta Q = \left[\frac{\sum_{in}+k_{i,in}}{2m} - \left(\frac{\sum_{tot}+k_i}{2m}\right)^2\right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right]$$

Modularity가 변하지 않을 때까지 community에 노드를 포함시키고 더 이상 변하지 않을 때 Phase 2가 시작된다.
Phase2는 Phase1에서 생성된 community 각각을 하나의 노드로 취급한다. 이를 통해 크기가 작아진 그래프에 대해서 다시 Phase1 부터 반복시킨다.

```
For each node i:
    For each node j≠i:
        delta_m= delta_modularity(j.community.add(I))
        If max_delta_modularity > delta_m
            max_i = i
            max_j = j
            max_delta_modularity=delta_m
    G.community(j).add(i)
Until no improvement
```

Textbox 4 Louvain algorithm phase1 pseudo code

```
result = []
for community_p2 in communities_p2:
    tmp = []
    for comm in community_p2:
        tmp += communities_p1[comm]
        //커뮤니티를 하나의 노드로서 더한다.
    result += [tmp]
```

Textbox 5Louvain algorithm phase2 pseudo code

3. **Details of experiment**

실험 데이터

1. Zachary's karate club

미국 대학교의 가라데 클럽 멤버들의 네트워크를 나타낸 데이터이다. 클럽 외부에서 교류가 있는 사람끼리 연결시켰다. 가장 유명한 네트워크 커뮤니티 예시중 하나이다. 총 노드 수는 34개 엣지의 수는 78개이다.

2. Email-Eu-core network

이 데이터는 유럽의 대형 연구기관의 이메일 데이터를 사용하여 생성되었다. 연구기관의 구성원들 사이의 이메일의 송수신정보를

받아서 각각 edge로 둔 데이터이다.
총 노드 수는 1005가지 이고 총 edges는 25571가지이다.

3. Askubuntu-a2q

약 7년간 askubuntu.com의 질문과 답변을 주고받은 유저끼리 연결시켰다.
총 노드 수는 137517 가지이고 총 edges는 280102가지이다.

세 dataset 각각에 세 method 각각을 적용시켰다. 크기가 작은 Zachary's karate club 데이터의 경우 모든 method에서 도출된 결과를 시각화 시켰다. Timeout을 24시간으로 잡고 돌렸기에 부족한 데이터가 존재한다.

4. Performance analysis

4.1. Zachary's karate club

Zachary's karate club 데이터에 Girvan-Newman method, CNM(Clauset-Newman-Moore) method, Louvain method를 사용하여 그래프를 partition하였다. 이 때 각 step 중 가장 높은 modularity를 갖는 step에서의 community를 시각화 한 모습은 다음과 같다.

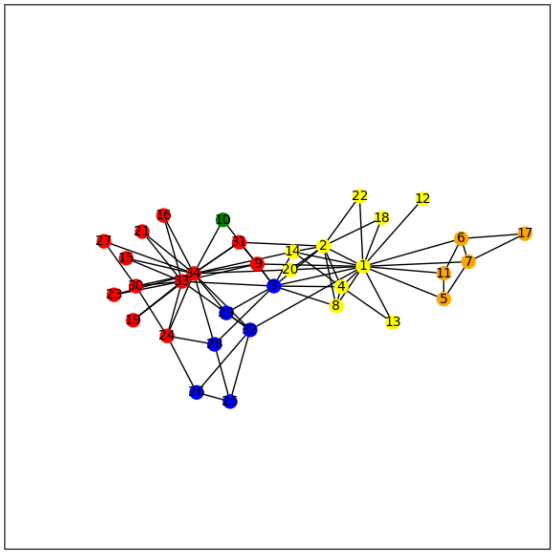

Figure 1 Girvan-Newman method 사용했을 때 Zachary karate club의 community

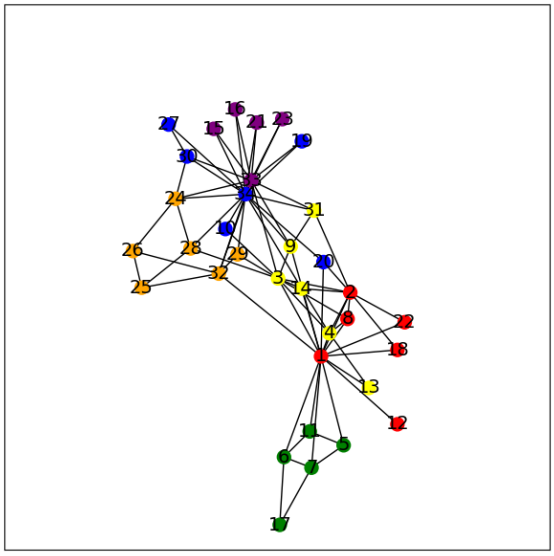

Figure 2 CNM method 사용했을 때 Zachary karate club의 community

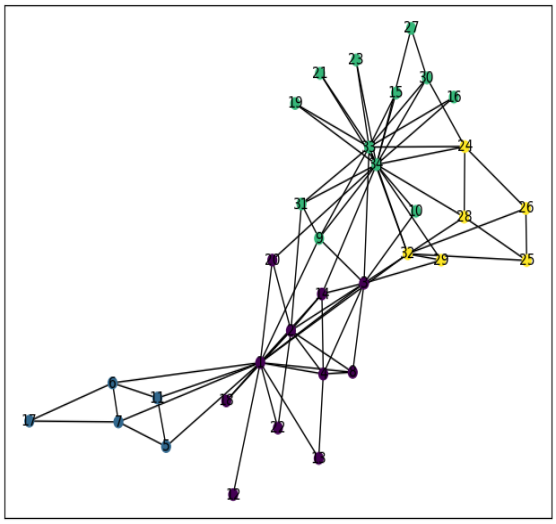

Figure 3 Louvain method 사용했을 때 Zachary karate club의 community

Zachary karate club의 네트워크 그래프를 세 방법을 사용해서 partition된 모습만을 봤을 때 세 method가 큰 차이를 보이지 않았지만 CNM method가 셋 중 가장 차이가 큰 모습을 볼 수 있었다.

각각의 알고리즘의 partition 성능을 더 정확히 알아보기 위해 Step마다 modularity를 측정한 결과는 다음과 같다. (Louvain은 phase 2가 끝나는 것을 한 step으로 취급했다.)

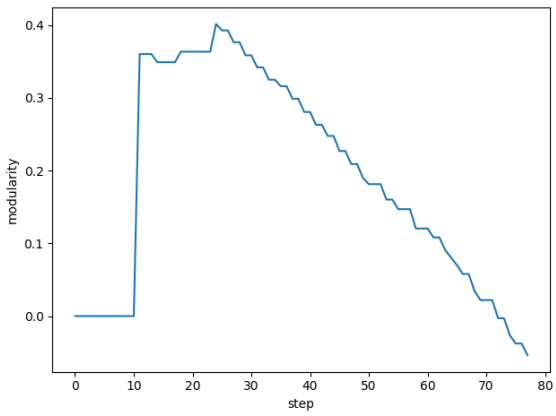

Figure 4 Zachary karate club 데이터에 Girvan-Newman method를 사용했을 때 modularity의 변화

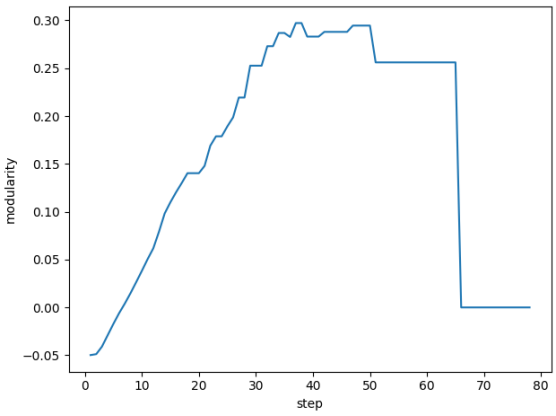

Figure 5 Zachary karate club 데이터에 CNM method를 사용했을 때 modularity의 변화

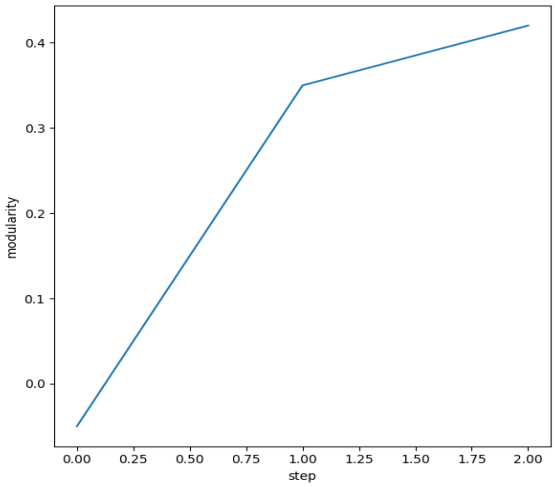

Figure 6 Zachary karate club 데이터에 Louvain method를 사용했을 때 modularity의 변화

각 method의 최대 modularity를 비교해봤을 때 결과는 다음과 같다.

	GN	CNM	Louvain
modularity	0.421	0.3	0.43

따라서 zachary karate club의 community detection은 modularity 기준으로 CNM method가 가장 성능이 낮았고 Girvan-Newman method와 Louvain method가 거의 비슷했다.

각 method의 소요시간은 다음과 같았다.

	GN	CNM	louvain
Time(sec)	0.1307	2.5602	0.009

4.2. Eu-email

Eu-email 데이터도 마찬가지로 각 method 각 step에서 modularity를 측정해보았다.

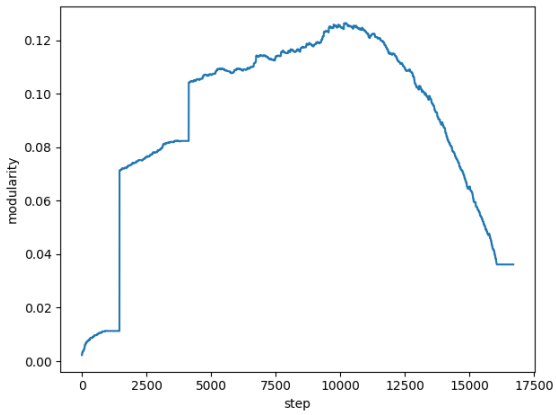

Figure 7 Eu-email 데이터에 Girvan-Newman method를 사용했을 때 modularity의 변화

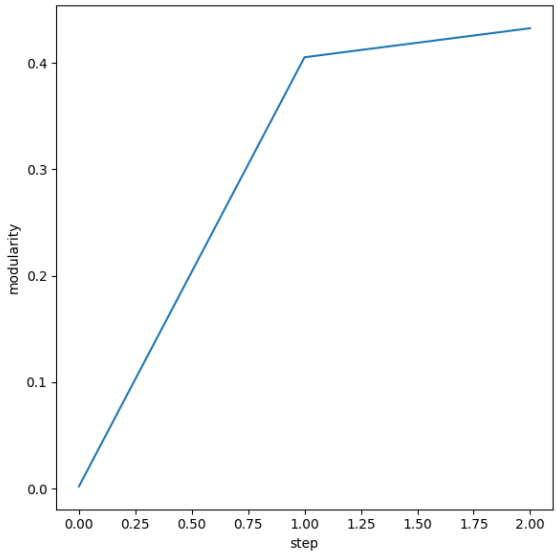

Figure 8 Eu-email 데이터에 Louvain method를 사용했을 때 modularity의 변화

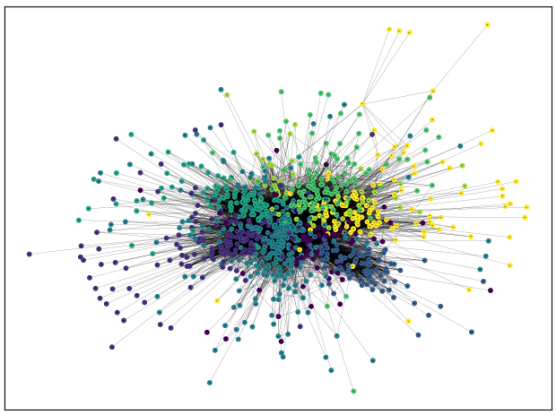

Figure 9 Eu-email 데이터에 Louvain method를 사용했을 때의 community 모습

Girvan-Newman method를 사용했을 때 modularity가 가장 높은 지점에서 그래프는 731개의 커뮤니티를 형성했고 Louvain method를 사용했을 때 그래프는 23개의 커뮤니티를 형성했다.

4.3. Askubuntu-a2q

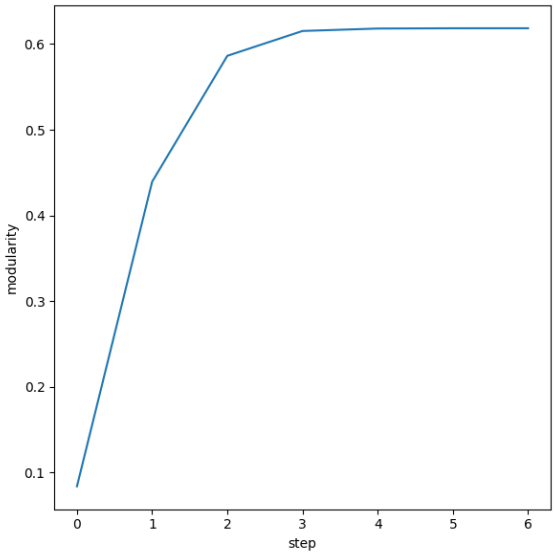

Figure 10 askubuntu 데이터에 Louvain method를 사용했을 때 modularity의 변화

10만개 이상의 노드, 28만개의 엣지를 가진 askubuntu 데이터에서 louvain method를 썼을 때 9299개의 커뮤니티를 형성했다.

4.4. 알고리즘 성능 시간 기준 비교

각 단계별로 전체 Modularity를 계산하는 데에 $O(n^2)$만큼의 시간이 들기 때문에 알고리즘에 걸리는 소요시간만을 평가했다. 이 때 소요시간이 다음과 같은 결과가 나왔다.

	GN	CNM	Louvain
Zachary Nodes 34 Edges-78	0.1307 (sec)	2.5602 (sec)	0.009 (sec)
Eu-email Nodes1005 Edges25571	35566.4 (sec)	More than 24hours	0.4389 (sec)
Askubuntu Nodes137517 Edges280102	More than 24hours	More than 24hours	37.438 (sec)

Timeout을 24시간으로 잡아서 몇몇 case에 대해서 몇몇 case에 대해서는 정확한 측정이 불가능했다. 전체적으로 소요시간이 CNM > GN > Louvain 이다.

4.5. 알고리즘의 partition성능 비교- modularity 기준

각각의 최대 Modularity는 다음과 같다.

	GN	CNM	Louvain
Zachary	0.421	0.3	0.43
Eu-email	0.128	X	0.44
askubuntu	X	X	0.63

CNM method는 작은 크기의 네트워크에서도 community partition 성능이 매우 낮았고 Girvan-Newman method와 Louvain method를 비교했을 때 작은 네트워크에서는 두 방법이 비슷한 성능이지만 네트워크의 크기가 커질수록 Girvan-Newman method의 성능이 떨어졌고 Louvain은 partition 성능이 유지되거나 높아졌다.

5. Conclusion

그래프 이론은 다양한 영역에서 다양한 응용에 널리 사용된다. 커뮤니티를 탐색해 어떻게 하면 질병의 전파를 막을 수 있을지, 항공사나 고속도로를 결정할 수 있을지 등등을 알 수 있다. 이 알고리즘은 우리를 둘러싼 세계를 경험하는데 도움을 준다. [2] 네트워크 연구가 오래되지 않았지만 Girvan-Newman method, CNM method, Louvain method 등 성능이 계속 좋은 알고리즘이 발견되는 중이다. 실험결과로 알 수 있듯이 Louvain method의 경우 나머지 두 알고리즘보다 시간 복잡도 기준, modularity 기준 모두 압도적으로 좋은 알고리즘이다. 2002년에 Girvan-Newman 알고리즘이 나온 후로 community detection 알고리즘의 성능이 이 정도로 발전했다는 것은 큰 의미가 있다. 비교 대상이 되는 두 알고리즘도 커뮤니티 분석에 큰 영향을 끼쳤다. Girvan- Newman도 최초의 커뮤니티 분석 알고리즘이라는 의의가 있고 CNM도 Louvain알고리즘의 기초가 되었다는 것에 의미가 있다.

6. References

[1] Newman, M. E. J. (2006). "Modularity and community structure in networks". Proceedings of the National Academy of Sciences of the United States of America. ***103*** *(23): 8577–8696.*

[2] Kitchovitch, S., and Liò, P. Community Structure in Social Networks: Applications for Epidemiological Modelling. PloS one 6, 7 (2011), e22220.

[3] Fast unfolding of communities in large networks, Vincent D et al.(2008)

[4] Newman, M. E. J. Fast algorithm for detecting community structure in networks. Physical Review E 69, 2 (2004), 1–5.