

공통 이웃 그래프 밀도를 사용한 소셜 네트워크 분석

(Social Network Analysis using Common Neighborhood Subgraph Density)

강 윤 섭 * 최 승 진 **
(Yoonseop Kang) (Seungjin Choi)

요 약 소셜 네트워크를 비롯한 네트워크로부터 커뮤니티를 발견하려면 네트워크의 노드를 그룹 내에서는 서로 조밀하게 연결되고 그룹 간에는 연결의 밀도가 낮은 그룹들로 군집화하는 과정이 꼭 필요하다. 군집화 알고리즘의 성능을 위해서는 군집화의 기준이 되는 유사도 기준이 잘 정의되어야 한다. 이 논문에서는 네트워크 내의 커뮤니티 발견을 위해 유사도 기준을 정의하고, 정의한 유사도를 유사도 전파(affinity propagation) 알고리즘과 결합하여 만든 방법을 기존의 방법들과 비교한다.

키워드 : 소셜 네트워크, 커뮤니티, 그래프 밀도, 유사도 전파

Abstract Finding communities from network data including social networks can be done by clustering the nodes of the network as densely interconnected groups, where keeping interconnection between groups sparse. To exploit a clustering algorithm for community detection task, we need a well-defined similarity measure between network nodes. In this paper, we propose a new similarity measure named "Common Neighborhood Sub-graph density" and combine the similarity with affinity propagation, which is a recently devised clustering algorithm.

- 본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원 사업의 연구결과로 수행되었음(NIPA-2010-C1090-1031-0009)
- 이 논문은 제36회 추계학술발표회에서 '공통 이웃 그래프 밀도를 사용한 소셜 네트워크 분석'의 제목으로 발표된 논문을 확장한 것임

* 학생회원 : POSTECH 컴퓨터공학과
e0en@postech.ac.kr
** 종신회원 : POSTECH 컴퓨터공학과 교수
seungjin@postech.ac.kr
논문접수 : 2009년 12월 24일
심사완료 : 2010년 2월 2일

Copyright©2010 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지 : 컴퓨팅의 실제 및 레터 제16권 제4호(2010.4)

Key words : Social network, community, graph density, affinity propagation

1. 서 론

복잡한 계는 대개 다수의 개체 간의 관계를 그래프 형태로 나타낼 수 있는 네트워크로 되어 있다. 생명체, 생태계, 경제, 월드 와이드 웹, 소셜 네트워크 등이 좋은 예이다. 네트워크 형태의 데이터를 잘 이해하는 방법 중 하나가 바로 네트워크 안의 커뮤니티 구조를 발견하는 것이다. 네트워크 안에서 커뮤니티를 찾는 것은 곧 네트워크를 나타낸 그래프의 노드를 서로 밀접하게 연결된 그룹들로 군집화하는 문제와 같다.

그래프 분할이나 군집화 등 커뮤니티 발견을 위한 다양한 방법들이 발표되었다[1-3]. 특히 군집화 문제는 커뮤니티 발견 문제와 매우 유사하기 때문에 기존에 존재하는 군집화 알고리즘을 이용하여 네트워크에서 커뮤니티를 찾고자 하는 시도가 자연스럽게 나타났다. 커뮤니티 발견 문제를 일반적인 군집화 문제로 변환하는데 있어서 가장 중요한 것은 노드 간의 유사도를 적절하게 정의하는 것이다. 그러나 기존의 알고리즘에서는 그래프 이론에서 정의된 노드 간 유사도를 주로 활용하였으며 커뮤니티 발견 문제를 위해 유사도를 새로 정의해 사용하고자 하는 시도는 많지 않았다.

이 논문에서는 커뮤니티 발견 문제를 위해 새로 정의한 유사도인 공통 이웃 그래프 밀도(Common neighborhood sub-graph density, CND)를 소개한다. 공통 이웃 그래프 밀도는 두 노드가 공통적으로 이웃하고 있는 노드로 구성되는 부분 그래프의 링크 밀도를 통해 직접적으로 두 노드가 같은 커뮤니티에 속하는지의 여부를 예측한다. 군집화 방법으로는 유사도 전파 알고리즘[4]을 사용하여 공통 이웃 그래프 밀도와 결합하여 커뮤니티 발견을 수행한다. 더 나아가 공통 이웃 그래프 밀도와 유사도 전파 알고리즘 사이의 연관관계를 파악하고 이를 통해 커뮤니티 발견 알고리즘의 성능 향상을 꾀하고, 기존의 알고리즘과 성능을 비교한다.

2. 공통 이웃 그래프 밀도

두 노드가 서로 같은 커뮤니티에 속하는지 여부를 나타내는 유사도가 정의되어 있으면 이를 통해 군집화 알고리즘을 적용시킴으로써 커뮤니티 발견 문제를 해결할 수 있다. 커뮤니티는 서로 조밀하게 연결되며 다른 그룹과는 희박한 연결만을 가지는 노드의 그룹으로 정의된다. 즉, 같은 커뮤니티에 속한 노드 간에는 조밀한 연결이 존재한다.

따라서 가장 간단하게는 두 노드 사이의 링크가 있는지 여부에 따라 유사도를 결정하는 방법이 있다. 그러나

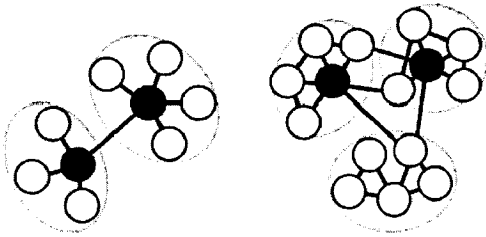


그림 1 링크 가중치 및 공통 이웃 노드 개수가 커뮤니티 발견을 위한 유사도로써 적절치 못한 사례

서로 다른 커뮤니티 사이에도 링크는 존재하고, 이런 커뮤니티 간 링크가 많은 네트워크에서는 노드 사이의 링크 존재 여부만으로는 두 노드가 같은 커뮤니티에 있는지 판별할 수 없다.

두 노드가 동시에 링크로 연결되어 있는 이웃 노드의 개수를 유사도로 사용하는 방법이 있다. 그러나 이 기준으로는 이웃 개수가 매우 적은 노드 혹은 이웃 노드가 많아서 여러 커뮤니티에 걸쳐 있는 노드를 포함하는 노드 쌍의 경우에는 두 노드가 같은 커뮤니티에 속하는지 정확히 판별할 수 없다. 두 노드의 이웃이 많은 경우에는 서로 다른 커뮤니티에 속한 노드 쌍이라도 많은 공통 이웃 노드를 가질 수 있기 때문이다. 또한 공통 이웃 노드를 많이 갖는다는 사실은 그룹 내의 링크 밀도로 정해지는 커뮤니티의 정의와는 상관이 없다. 따라서 공통 이웃 노드의 수만으로 두 노드가 같은 커뮤니티에 속한다고 판단하기는 어렵다(그림 1).

위 두 방법들의 단점을 해소하기 위해 두 개의 노드와 공통 이웃 노드로 이루어지는 부분 그래프의 링크 밀도를 두 노드 간의 유사도로 사용할 수 있다. 이 부분 그래프의 링크 밀도가 높다는 것은 두 노드와 두 노드의 공통 이웃 노드로 이루어진 그룹 내의 링크 밀도가 높다는 것이며, 이는 두 노드가 같은 커뮤니티에 있을 가능성이 높다는 것을 시사한다.

서로 다른 커뮤니티의 두 노드가 서로 연결되어 있거나 두 노드가 수많은 이웃 노드를 공유하는 경우에도 공통 이웃 노드 간의 연결까지 조밀할 가능성은 낮기 때문에 이 유사도는 이전에 언급된 두 유사도보다 두 노드가 같은 커뮤니티에 속하는지 여부를 판단하는데 있어서 더 효과적인 기준이라고 할 수 있다.

엄밀하게는 두 노드와 그 공통 이웃 노드로 이루어진 부분 그래프의 링크 밀도는 부분 그래프에 존재하는 링크 수와 부분 그래프에 존재할 수 있는 최대 링크 수의 비율로 정의된다:

$$K_{ij} = \frac{\sum_{k \in C_{ij}} \sum_{l \in C_{ij}} A_{kl}}{|C_{ij}|(|C_{ij}| - 1)} \quad (1)$$

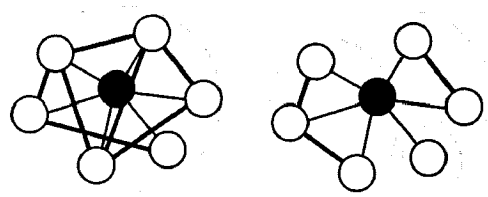


그림 2 유사도 전파 알고리즘으로써의 노드의 이웃 수와 자기 자신과의 CND 값 간의 비교. 왼쪽과 오른쪽 노드 모두 6개의 노드와 이웃하고 있지만 커뮤니티의 기본 노드로써는 왼쪽의 노드가 오른쪽보다 더 적합하다.

위 식에서 C_{ij} 는 노드 i, j 와 그 공통 이웃 노드로 이루어진 집합, A_{kl} 는 인접 행렬 (affinity matrix) A 의 (k, l) 번째 원소이며, 노드 k, l 사이에 링크가 있으면 1, 아니면 0의 값을 갖는다. 인접 행렬이 비대칭인 경우도 있으나, 이 논문에서는 대칭 행렬인 경우만을 고려한다 ($A_{kl} = A_{lk}$).

이렇게 새로 정의한 유사도 역시 두 노드 간의 공통 이웃 노드가 적은 경우에는 커뮤니티 간 링크의 존재에 민감한 특성을 보인다. 예를 들어 두 노드가 단 하나의 공통 이웃 노드를 가지는 경우에는 어떤 상황에서도 두 노드의 유사도는 2/3 이하로 떨어지지 않는다.

이 문제를 해결하려면 두 노드가 적은 공통 이웃 노드를 가질수록 낮은 유사도를 주도록 값을 조절하면 된다. 여러 가지 방법이 있을 수 있지만 여기에서는 간단하게 상수 $|C_{ij}|^\gamma$ 를 곱하였다. 이에 따라 최종적으로 얻어지는 유사도인 공통 이웃 그래프 밀도(CND)의 새로운 정의는 다음과 같게 된다:

$$K_{ij} = |C_{ij}|^\gamma \frac{\sum_{k \in C_{ij}} \sum_{l \in C_{ij}} A_{kl}}{|C_{ij}|(|C_{ij}| - 1)} \quad (2)$$

여기에서 γ 는 매개변수이다. γ 의 값이 클수록 공통 이웃 노드가 많은 노드에 높은 유사도를 부여한다.

3. 유사도 전파 알고리즘과의 관계

새로 정의한 유사도를 이용해서 네트워크로부터 커뮤니티를 찾기 위해서는 군집화 알고리즘이 필요하다. K 평균 군집화 알고리즘이나 계층적 군집화 알고리즘과 같은 기존에 널리 쓰이던 알고리즘 방법들은 그 클러스터의 개수를 사용자가 미리 설정해 주어야 한다는 약점을 갖고 있다. 이와 같은 알고리즘들은 자동으로 정확한 개수의 커뮤니티를 찾아야 하는 커뮤니티 발견 문제에 적용하기에 어려움이 있다. 유사도 전파 알고리즘은 이 클러스터 개수를 자동으로 결정하는 군집화 알고리즘이다[4].

이 논문에서는 유사도 전파 알고리즘에 CND 를 유사도로 주어 커뮤니티 발견을 수행한다. 기본적으로 유사도 전파 알고리즘은 데이터 점들 중에 클러스터를 대표하는 견본 데이터를 찾는 방식으로 군집화를 수행한다. 이 알고리즘은 견본 데이터를 정하고, 그 견본 데이터와 같은 클러스터에 속하는 데이터를 찾는 작업을 교대로 반복하며 군집화 결과를 도출한다.

견본 데이터를 결정하는 데에는 입력된 각각의 데이터 점 k 에 대한 선호도 $s(k, k)$ 가 중요한 역할을 한다. 큰 선호도를 갖는 점일수록 그 점이 클러스터를 대표하는 견본 데이터로 선정될 가능성이 높아진다.

따라서 유사도 전파 알고리즘의 결과로 얻어지는 클러스터의 개수 및 군집화 결과의 정확성은 입력 선호도에 크게 좌우된다. 전체적으로 입력 선호도의 값이 커지면 결과로 얻어지는 클러스터의 개수가 늘어나게 된다. 모든 노드에 $s(k, k)$ 의 값을 동일하게 주는 것도 실제 상황에서 잘 동작하는 방법이지만, 여기에는 개선의 여지가 더 있다.

이 경우에 새로 제안한 유사도인 CND 를 활용할 수 있다. CND 는 기본적으로 두 노드의 공통 이웃 노드 간의 연결의 밀도를 측정하는 유사도이다. 따라서 한 노드와 자기 자신의 CND 값은 자기 자신과 이웃 간의 연결의 밀도를 측정하게 된다. 따라서 높은 CND 값은 그 노드가 많은 노드와 밀접하게 연결되어 있다는 것을 의미하며, 이는 그 노드의 이웃들이 거의 한 개의 커뮤니티에 속한다는 것을 뜻한다. 이러한 노드는 한 개의 커뮤니티를 대표하는 값으로, 이러한 노드에 높은 입력 선호도 값을 부여하는 것은 타당하다고 볼 수 있다(그림 2).

4. 실험 및 분석

4.1 잡음이 있는 네트워크에서의 매개변수 γ 의 효과

CND 를 정의할 때, 잡음이 있는 네트워크, 즉 커뮤니티 간에 연결이 많은 네트워크에서 공통 이웃이 적은 서로 다른 커뮤니티에 속하는 노드 쌍이 큰 유사도를 갖는 것을 방지하기 위해 부분 그래프 밀도에 상수 $|C_{ij}|^\gamma$ 을 곱하였다. 이렇게 곱해진 상수가 커뮤니티 발견의 정확도에 끼치는 영향을 알아보기 위해 매개변수 γ 의 값을 바꾸어 가며 인공적으로 생성한 잡음이 있는 네트워크로부터 커뮤니티를 발견하는 작업을 수행했다.

잡음이 있는 네트워크를 생성하기 위해 128개의 노드를 각각 32개의 같은 크기의 커뮤니티로 나누어 무작위로 평균적으로 32개의 링크를 연결했다. 각 노드에 N 개의 링크가 연결된다고 할 때 평균적으로 pN 개의 노드는 같은 커뮤니티, $(1-p)N$ 개의 노드는 다른 커뮤니티에 연결되도록 했다. 따라서 p 값을 바꿈으로써 커뮤니티 간 링크의 수, 즉 잡음의 정도를 조절할 수 있게 된다.

커뮤니티 발견 알고리즘의 정확도 기준으로는 발견한 커뮤니티의 평균 순도(average purity)를 사용했다. 그 정의는 다음과 같다:

$$\frac{1}{K} \sum_k \sum_{i \neq j \text{ and } i, j \in C_k} \frac{\delta(i, j)}{\sum_k |C_k|^2} \quad (3)$$

K 는 발견된 커뮤니티 개수이며 $\delta(i, j)$ 는 노드 i 와 j 가 실제로도 같은 커뮤니티에 속하면 1, 아니면 0의 값을 갖는다. $|C_k|$ 는 발견된 커뮤니티 중 k 번째 것의 크기를 뜻한다[5].

테스트해 본 모든 γ 값에서 CND 와 유사도 전파 알고리즘을 사용한 커뮤니티 발견 방법은 정확히 4개의 커뮤니티를 발견했다. 그러나 발견된 커뮤니티의 평균 순도는 매개변수 γ 의 값과 p 값에 따라 변화했다. 네트워크의 잡음이 비교적 낮은 경우에는 γ 의 값이 증가함에 따라 평균 순도는 약간 감소하는 양상을 보였다. 그러나 네트워크의 잡음이 증가할수록 γ 의 값이 증가할수록 평균 순도도 증가하다가 γ 가 1.0 근처가 되는 시점부터 다시 평균 순도가 낮아지는 모습을 보였다(그림 3). 이로부터 우리는 네트워크의 잡음 정도에 따라 매개변수 γ 의 값이 중요해지며 낮은 γ 값보다는 높은 값에서 비교적 높은 정확도로 커뮤니티를 발견할 수 있다는 것을 알 수 있다.

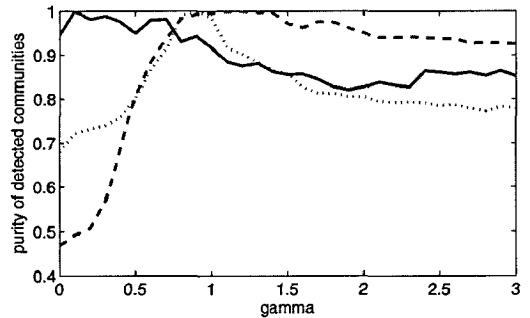


그림 3 잡음이 있는 네트워크에서 CND 의 매개변수 γ 의 영향(청색 실선: $p=0.8$, 녹색 점선: $p=0.5$, 적색 패선: $p=0.2$).

그러나 γ 값이 일정 값을 넘으면 정확도가 떨어지기 때문에 γ 의 최적 값을 찾는 것은 중요한 문제이다. 이 값을 찾기 위해 modularity[6]라는 기준을 이용했다. 어떤 그래프와 그 그래프에서 발견된 커뮤니티 정보가 주어졌을 때, modularity Q 는 발견된 커뮤니티 내의 링크 개수와 그 그래프와 각 노드의 degree만이 같은 무작위 그래프 내의 링크 개수의 차이로 정의된다. 엄밀하게는 다음과 같이 정의된다):

$$Q = \frac{1}{4m} \sum_i \sum_j s(i, j) \left(A_{ij} - \frac{k_i k_j}{2m} \right) \quad (4)$$

위 식에서 A 는 인접 행렬, m 은 그래프 내의 전체 링크 개수, k_i 는 i 번째 노드의 degree로 정의된다. 그리고 $s(i, j)$ 는 두 노드가 같은 커뮤니티에 소속되어 있으면 1, 아니면 -1의 값을 갖는다. Modularity Q 는 커뮤니티 발견 방법의 성능 평가 기준으로 널리 쓰이고 있으며, 이 값이 높으면 높을수록 커뮤니티 내의 링크가 무작위 그래프보다 조밀하다는 것을 의미하며 알고리즘의 성능이 높다는 척도로 쓰인다. 따라서 이후의 실험에서도 γ 는 이 modularity를 최대화하는 값으로 선택하여 사용하였다.

4.2 유사도 전파 알고리즘의 입력 선호도

3장에서도 언급했듯이 유사도 전파 알고리즘의 성능에는 각 노드의 입력 선호도 $s(i, i)$ 의 값이 매우 중요하다. CND를 입력 선호도로 사용했을 때의 커뮤니티 발견 정확도 증가를 확인하기 위해 모든 노드에 같은 입력 선호도를 부여한 경우(Const), 그리고 각 노드의 이웃 수를 입력 선호도로써 부여한 경우(Degree), 그리고 노드와 자기 자신의 CND 값을 입력 선호도로 부여한 경우(Autosim)을 비교하였다. Const의 경우에는 기존에 자주 사용되던 대로 유사도 행렬의 중앙값을 사용했다.

유사도 전파 알고리즘에 사용되는 유사도는 0보다 작아야 한다는 조건이 있기 때문에, 유사도 행렬의 값에서 유사도 행렬의 최대값의 2.5배를 뺀 값을 알고리즘의 입력으로 사용하였다.

실험에는 College football[7], Political books¹⁾, Political blogs[8] 데이터 세트를 사용했다. 이 데이터 세트들은 각 데이터 점이 속한 커뮤니티 정보까지 포함하고 있다. College football은 12개, Political books는 2개, Political blogs는 3개의 커뮤니티로 구성되어 있다.

유사도 전파 알고리즘과 CND를 결합하여 커뮤니티 발견을 수행한 결과, 모든 데이터 세트에서 Autosim이 가장 정확한 커뮤니티 개수를 얻어냈다. 또한 College football 데이터 세트에서 가장 높은 커뮤니티 순도를 보인 것도 Autosim이었다(표 1). Const는 두 데이터 세트에서 가장 높은 커뮤니티 순도를 기록했으나, 그 순도가 Autosim에 비해 크게 높지 않았고, 발견한 커뮤니티 개수의 정확성도 다른 방법들에 비해 크게 떨어졌다. 따라서 Autosim, 즉 유사도 전파 알고리즘의 입력 선호도로 데이터 자신의 CND 값을 이용한 경우에 가장 효과적으로 커뮤니티를 찾을 수 있다고 할 수 있다.

표 1 CND와 유사도 전파를 사용한 커뮤니티 발견 방법에 다양한 입력 선호도 값을 주어 커뮤니티 발견 작업을 수행한 실험 결과

데이터		Const	Degree	Autosim
College football	개수	15	11	11
	순도	0.8507	0.8368	0.8529
Political blogs	개수	12	2	2
	순도	0.8397	0.8205	0.8238
Political books	개수	7	4	3
	순도	0.7136	0.6768	0.7098

4.3 기존 알고리즘과 성능 비교

이 논문에서 제안한 유사도인 CND의 효과를 확인하기 위해, 자카드 지수와 링크 가중치를 각각 유사도 전파 알고리즘을 조합한 방법과 CND와 유사도 전파 알고리즘을 조합한 결과를 비교했다.

기존의 알고리즘으로써 가장 성공적인 것 중 하나인 Newman의 leading eigenvector 방법[9]과도 비교하였다. 또 최근에 개발된 방법으로써 R. Gosh 등이 고안한 영향-기반 modularity 방법도 비교 대상에 포함하였다. 이는 modularity를 일반화한 것으로, 한 노드가 다른 노드에 미치는 영향을 유사도로 활용한다.

이 실험에서도 College football, Political blogs, Political books 등 3개의 데이터 세트를 사용하였다. CND에 사용되는 매개변수 γ 와 Gosh의 영향-기반 modularity 방법에 사용되는 매개변수 α 는 모두 발견된 커뮤니티의 modularity를 최대화하는 값으로 설정되었다.

CND는 College football 데이터에서는 가장 높은 커뮤니티 순도를 기록했으나, 나머지 두 개의 데이터에서는 각각 두 번째, 세 번째로 높은 값을 기록하였다(표 2). 가장 높은 값을 기록하지 못한 데이터에서도 1위의 알고리즘과 크게 다르지 않은 결과를 보였으나, 커뮤니티 순도만 보아서는 CND가 다른 알고리즘보다 커뮤니티 발견 작업에 더 적합하다고 보기는 힘들다. 하지만 각 알고리즘이 발견한 커뮤니티 숫자를 데이터의 실제 커뮤니티 개수와 비교해 보면 CND의 장점이 확연히 드러난다. CND와 유사도 전파 알고리즘을 합친 방법은 모두 거의 정확한 개수의 커뮤니티를 찾은 반면, 나머지 알고리즘이 찾은 커뮤니티 개수는 실제 개수와 거리가 멀었다(표 2). 커뮤니티 순도와 개수를 모두 고려하여 비교해 볼 때, 실험에 사용된 알고리즘 중에서는 CND가 유일하게 정확한 개수의 커뮤니티를 높은 순도로 발견하는 방법이었다.

6. 결론

이 논문에서는 소셜 네트워크에서의 커뮤니티 발견을

1) <http://www.orgnet.com/>

표 2 CND와 유사도 전파를 사용한 커뮤니티 발견 방법과 기존의 방법으로 커뮤니티 발견 작업을 수행한 실험 결과

데이터		CND	자카드 지수	링크 가중치	Newman 알고리즘	Gosh의 알고리즘
College football	개수	11	16	14	12	6
	순도	0.8345	0.834	0.4739	0.5865	0.5808
Political blogs	개수	2	118	61	2	4
	순도	0.8357	0.8664	0.7776	0.892	0.6457
Political books	개수	2	17	12	6	3
	순도	0.7548	0.7892	0.7471	0.5867	0.6351

위해 새로운 유사도 기준인 CND를 제시하였다. CND는 한 쌍의 노드와 그들의 공통 이웃으로 이루어지는 부분 그래프의 링크 밀도로부터 두 노드가 같은 커뮤니티에 속하는지 여부를 추정하여 유사도 값으로 내놓는다. 또한 각 노드와 자기 자신의 CND 값은 각 노드의 이웃이 얼마나 한 커뮤니티에 집중되어 있는지를 나타내는 척도로 해석할 수 있기 때문에 유사도 전파 알고리즘을 수행할 때 변수로 받는 입력 선호도를 각 노드와 자신의 CND 값으로 사용하여 알고리즘의 성능을 개선할 수 있다. 실제계의 네트워크 데이터에서 CND와 다른 알고리즘을 비교한 결과 CND와 유사도 전파 알고리즘의 조합이 가장 정확한 개수의 커뮤니티를 높은 순도로 찾는다는 것이 확인되었다.

참 고 문 헌

- [1] S. Fortunato and C. Castellano, "Community structure in graphs," In *Encyclopedia of Complexity and System Science*. Springer Verlag, Heidelberg, Germany, 2009.
- [2] Karl Branting, "Overcoming Resolution Limits in MDL Community Detection," In *2008 KDD workshop on Social Network Analysis*, Las Vegas, Nevada, USA, 2008.
- [3] Nicolas Neubauer and Klaus Obermayer, "Towards Community Detection in k-Partite k-Uniform Hypergraphs," In *Workshop on Analyzing Networks and Learning with Graphs*, NIPS 2009 December 11, 2009 in Whistler, BC, Canada.
- [4] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, 315, February 2007.
- [5] Y. Wang, H. Song, W. Wang, and M. An, "A microscopic view on community detection in complex networks," In *Proceeding of the 2nd PhD Workshop on Information and Knowledge Management*, Napa Valley, California, 2008.
- [6] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, 69, 2004.
- [7] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, USA, 103(23):8577 - 8582, 2006.
- [8] L. A. Adamic and N. Glance, "The political blogosphere and the 2004 U.S. election: Divided they blog," In *Proceedings of the 3rd International Workshop on Link Discovery*, Chicago, Illinois, 2005.
- [9] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical Review E*, 74, 2006.