

# 상호간에 연결이 존재하지 않는 일부 노드를 포함한 네트워크를 시각화 할 때 노드-링크 다이어그램과 인접행렬의 비교

## Comparison of node-link diagram and adjacency matrix visualizing network with a group of nodes which have no interconnections

정석원  
Seokweon Jung

서울대학교  
Seoul National University  
swjung@hcil.snu.ac.kr

서진욱  
Jinwook Seo

서울대학교  
Seoul National University  
jseo@snu.ac.kr

### 요약문

노드-링크 다이어그램과 인접행렬은 네트워크 데이터를 시각화 할 때 가장 널리 사용되는 두 가지 기본적인 시각화들이다. 두 시각화를 비교하는 이전의 여러 연구들[1, 2, 3, 5]에 따르면 노드-링크 다이어그램과 인접행렬의 서로 다른 특성들로 인하여 두 시각화들은 각각 특정 작업을 수행하는데 있어서 강점을 가지고 있다. 그렇지만, 이러한 우위는 시각화 하려는 네트워크 데이터의 구조에 큰 영향을 받기 때문에 상황에 따라서 달라질 수 있다[2]. 본 연구에서는 네트워크에 속한 일부 개체들이 상호 간에 전혀 연결이 존재하지 않는 구조를 가진 네트워크 데이터를 노드-링크 다이어그램과 인접행렬로 표현해보고, 네트워크의 서로 다른 두 개체들 사이의 경로를 파악하는 작업을 수행할 때 각각의 시각화가 가지는 장점과 단점에 대해서 알아보았다.

### 주제어

네트워크 시각화, 노드-링크 다이어그램, 인접행렬

### 1. 서론

우리 주변에 존재하는 많은 데이터들은 네트워크의 속성을 가진다. 소셜 미디어, 통화, 금융 거래 등 타인과 행하는 행위들의 기록은 네트워크 형태의 데이터로 남게 된다. 이러한 네트워크 데이터들을 시각화 하는 방법에는 여러가지 방법들이 존재하지만, 그 중 가장 대표적인 것이 노드-링크 다이어그램과 인접행렬을 통한 시각화이다[2].

노드-링크 다이어그램과 인접행렬은 네트워크의 시각화에 있어서 가장 익숙하고 기초적인 시각화들으로써, 각자 장단점을 가지고 있다. 특정 작업을 수행할 때 유리한 시각화와 불리한 시각화가 나뉘게 된다. 일반적으로, 군집을 탐색하는 데는 인접행렬이, 경로를 탐색하는 데는 노드-링크 다이어그램이 유리하다고 한다[1, 3]. 시각화에 익숙하지 않은 초심자들을 대상으로 한 실험에서 이러한 사실은 증명이 되었다[5]. 두

시각화 방법의 장점을 모두 취하기 위해서 둘을 융합하려는 시도도 있어왔다[4].

노드-링크 다이어그램과 인접행렬이 각각 특정 분야에서 강점을 가지는 것은 맞지만, 모든 상황에 적용되는 것은 아니다. 연구에 따르면 노드-링크 다이어그램과 인접행렬을 통한 작업 수행 능력은 네트워크 데이터의 크기와 밀도 뿐만 아니라 네트워크 데이터에 숨어있는 구조에도 크게 영향을 받는다[2]. 따라서 둘 중 어느 시각화를 선택해야 원하는 목표를 이루는데 더 도움이 될지는, 분석하려는 데이터의 특성에 따라서 달라지게 된다.

이번 연구에서는 여러 종류의 네트워크 데이터들 중 크기가 크고 밀도가 높으면서 네트워크의 일부 개체(노드)들은 서로 간의 연결(링크)이 전혀 존재하지 않는 구조를 가진 네트워크 데이터를 대상으로 연구를 진행하였다. 위와 같은 구조를 가지는 네트워크 데이터를 노드-링크 다이어그램과 인접행렬을 사용하여 시각화를 하고, 각각의 경우에 대해서 데이터를 탐색하는 작업을 할 때의 효율을 비교해보았다. 이를 통해서 두 시각화 방법들이 본 연구에서 대상으로 한 네트워크 데이터를 시각화 할 때 각각 어떤 장점과 단점을 가지는지에 대해서 알아보고, 이를 활용하여 어떻게 하면 시각화를 개선할 수 있을지에 대해서 탐구해보도록 하였다.

### 2. 본문

#### 2.1 네트워크 데이터의 구조

우리가 다루려는 네트워크 데이터는 두 개의 서로 다른 노드 그룹으로 나누어져 있다. 두 그룹을 각각 Primary 노드들과 Secondary 노드들이라고 명명하였다. Primary 노드들은 네트워크에 속하는 다른 모든 노드들과 링크를 가질 수 있는 노드들이다. Secondary 노드들은 오직 Primary 노드와의 링크만을 가질 수 있고, 다른 Secondary 노드와는 링크를 가질 수가 없다.

현실세계에서 이러한 구조를 가지는 네트워크 데이터의 예시로는, 범죄 수사를 위해서 수집한 데이터가 존재한다. 수사를 위해서 데이터를

확보하면, 수사 대상이 된 용의자들의 통화, 금융거래 등의 네트워크 데이터를 확보할 수 있다. 이러한 네트워크 데이터들은 용의자들을 중심으로, 각 용의자들과 직접적으로 연결되어 있는 인물들만 파악이 가능하다. 따라서 용의자들이 이 네트워크의 Primary 노드, 용의자들과 연결로만 존재하는 인물들이 네트워크의 Secondary 노드가 된다.

링크의 방향성에 따라서 네트워크 데이터를 directed 와 undirected 로 구분할 수 있다. Directed 네트워크 데이터는 directed 네트워크 데이터에 비해서 연결의 방향성을 추가적으로 고려해줘야 하기 때문에 더 복잡한 시각화를 요구한다. 그러나, 본 연구에서는 연결의 방향성을 고려하지 않은 실험을 계획하였기 때문에 방향성을 고려하지 않아도 된다고 판단하였다. 따라서 시각화의 대상으로 undirected 네트워크로 한정하고 연구를 진행하였다.

## 2.2 노드-링크 다이어그램과 인접행렬

노드-링크 다이어그램[그림 1]은 네트워크의 개체들을 특정 형상으로 시각화하고, 개체들의 관계를 선으로 나타낸 표현법이다. 노드-링크 다이어그램은 직관적인 표현법이기 때문에, 데이터의 크기가 작고 밀도가 낮은 경우에는 인접행렬에 비해서 거의 모든 작업을 수행할 때 성능상에 우위를 갖고[5], 특히 두 개체 사이의 경로탐색과 관련된 작업을 할 때 유리하다. 그러나 데이터의 크기가 커지고 밀도가 높아질 경우, 화면에 표시해야 할 점과 선이 지나치게 많아져서 서로 겹치는 cluttering 문제가 발생하기 때문에 작업 수행능력이 크게 떨어지게 된다.

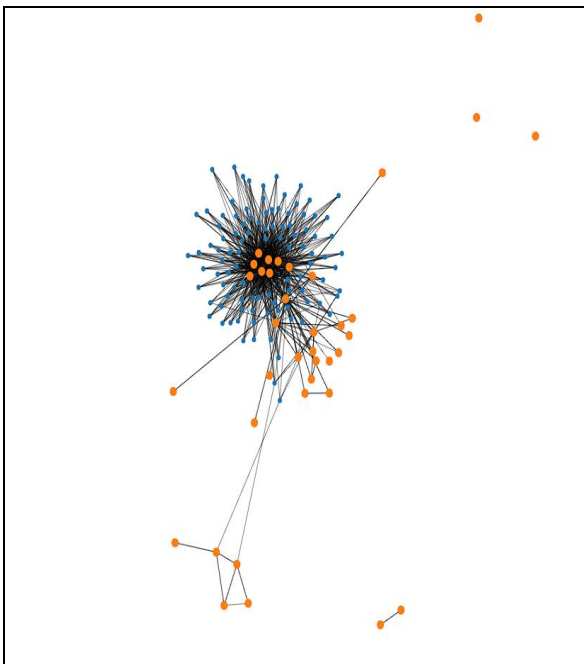


그림 1. 대상 네트워크 데이터의 노드-링크 다이어그램 시각화.

인접행렬[그림 2]은 네트워크의 개체들을 각각 x 축과 y 축에 나열하고, 개체들의 관계를 표의 형태로 나타낸 것이다. 일반적으로 인접행렬은 노드-링크 다이어그램에 비해서 군집을 찾는 작업을 할 때 유리하고[1], 노드-링크 다이어그램에 비해서 네트워크가 크고 밀도가 높더라도 작업 수행이 용이하다. 이는 화면에 표시해야 하는 개체수가 많아지면 cluttering 이 발생하는 노드-링크 다이어그램에 비해서, 인접행렬은 cluttering 문제에 대해서 자유롭기 때문이다. 그러나 네트워크가 지나치게 커지면 네트워크를 나타내는 표의 칸들이 작아져서 시인성이 나빠지기 때문에, 데이터의 크기가 커지면 작업 수행에 악영향을 끼치는 것은 마찬가지이다.

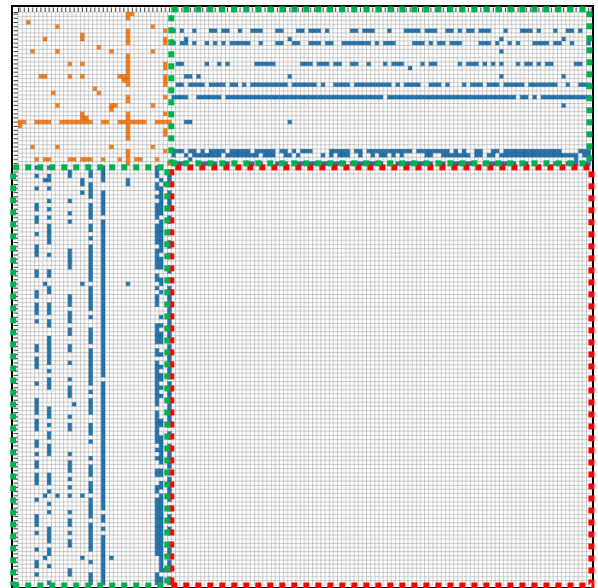


그림 2. 대상 네트워크 데이터의 인접행렬 시각화. 빨간색 사각형은 Secondary 노드들 사이의 링크가 존재하는 영역, 두 개의 초록색 사각형은 Primary - Secondary 링크가 존재하는 영역이다.

## 2.3 구현

React[6]와 D3.js[7]를 이용하여 간단한 노드-링크 다이어그램과 인접행렬을 구현하였다. 노드-링크 다이어그램의 경우 D3.js의 Force directed layout 을 이용하여 구현하였다. 추가적으로 Primary 노드들과 Secondary 노드들을 구분하기 위해서 color encoding 을 활용하였고, Primary 노드들을 강조하기 위해서 Secondary 노드들에 비해서 더 큰 점을 할당하였다. 인접행렬의 경우에는 Primary 노드들을 앞부분에 두어서 군집을 이루도록 하고, color encoding 을 통하여 Primary - Primary 링크와 Primary - Secondary 링크를 구분하였다.

사용한 데이터에 존재하는 총 노드의 개수는 138 개이고, 그 중 Primary 노드는 37 개, Secondary 노드는 101 개 존재한다. 총 링크의

개수는 659 개이고, Primary - Primary 노드들 사이의 링크는 111 개, Primary - Secondary 노드들 사이의 링크는 548 개 존재하였다.

### 3. 결과

네트워크 데이터를 노드-링크 다이어그램과 인접행렬로 시각화 한 결과 [그림 1]과 [그림 2]의 모습으로 시각화가 되었다. 그리고 각각의 시각화를 이용하여 임의의 두 Primary 노드를 연결하는 경로가 존재하는지 탐색하는 작업을 수행하였다.

### 4. 결론 및 논의

#### 4.1 결론

노드-링크 다이어그램을 이용한 시각화[그림 1]의 경우, Primary 노드들을 중심으로 Primary 노드들 사이의 징검다리가 되어주는 Secondary 노드들이 둘러싸는 형태로 나타나는 것을 확인할 수 있었다. 때문에 네트워크의 밀도가 낮은 주변부에 위치한 Primary 노드들에 대해서는 두 Primary 노드들이 여러 노드들을 거쳐서 연결이 되더라도 쉽게 파악이 가능했다. 그러나 밀도가 높은 중심부에 위치한 Primary 노드들을 대상으로 할 때는 Primary 노드들을 둘러싼 Secondary 노드들이 Primary 노드들 사이의 징검다리가 되어준다는 사실은 알 수 있었지만, 정확히 어떤 두 노드들을 연결해주는 것인지 파악하는 것은 cluttering 문제로 인하여 불가능에 가까웠다.

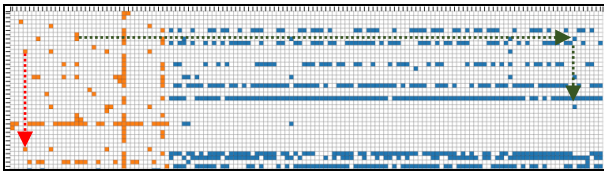


그림 3. 인접행렬의 상단 일부분. 빨간색 화살표는 두 Primary 노드 사이의 직접적인 연결관계가 존재하는 경우의 탐색 방법을, 초록색 화살표는 두 Primary 노드 사이에 Secondary 노드를 경유하는 연결관계가 이어

인접행렬을 이용한 시각화의 경우, [그림 2]의 붉은색 사각형 부분과 같이 Secondary 노드들 사이의 링크를 나타내는 부분은 전부 빈칸으로 나타나는 것을 확인할 수 있었다. 또한, [그림 2]의 초록색 사각형 부분과 같이 Primary-Secondary 링크를 나타내는 부분의 경우 현재 표현하고 있는 데이터가 undirected 네트워크 데이터이기 때문에, 두 초록색 사각형은 같은 정보를 담고 있다는 것을 확인할 수 있었다. 따라서 [그림 3]과 같이 인접행렬의 일부분만을 가져오는 것 만으로 [그림 2]의 인접행렬을 대신할 수 있다는 사실을 발견할 수 있었다.

인접행렬을 통해서 서로 다른 두 Primary 노드들 사이의 연결관계를 파악하려고 시도할 경우, 두 Primary 노드들끼리 직접 연결이 되어있는 경우

[그림 3]의 빨간색 화살표와 같이 한 번의 움직임을 통해서 탐색이 가능하다. 다른 노드를 한번 거쳐서 연결이 되어있는 경우 [그림 3]의 초록색 화살표와 같이 두 번의 움직임을 통해서 탐색이 가능하다. 그러나 두 Primary 노드들이 여러 개의 다른 노드들을 거쳐서 연결되어 있는 경우에는, 여러 번에 걸쳐서 움직임을 가져야 했기 때문에 도중에 처음 시작점의 위치를 놓치는 경우가 많았고, 또한 링크를 나타내는 칸의 크기가 작았기 때문에 정확한 위치를 확인하기가 힘들었다.

종합적으로 살펴보았을 때, 네트워크의 밀도가 높은 중심부에서의 연결관계를 확인할 때는 cluttering 문제의 영향을 받지 않고, 데이터의 특성으로 인하여 탐색 범위가 줄어든 인접행렬이 노드-링크 다이어그램보다 두 Primary 노드 사이의 경로를 탐색하는 작업을 수행하기에 유리했다. 비록 두 노드들이 다른 여러 노드들을 거쳐서 연결이 되어 있다면 탐색의 난이도가 올라가기는 했지만, 그럼에도 불구하고 cluttering 문제로 인한 어려움을 능가하지는 않았다. 그러나, 네트워크의 밀도가 낮은 주변부에 대해서는 노드-링크 다이어그램이 이전의 연구에서 밝혔듯이 인접행렬보다 우월함을 증명하였다. 이를 통하여, 추후에 본 연구에서 대상으로 삼는 네트워크 데이터를 시각화 할 때 어떤 시각화를 택하는 것이 유리할지에 대한 판단 기준을 확보 할 수 있었다.

#### 4.2 논의

결과를 관측한 후에, 추가적인 검증의 필요성을 느꼈다. 우리가 노드-링크 다이어그램과 인접행렬을 관찰하면서 얻은 결과들이, 실제로 타인들도 얻을 수 있는 공통된 결과인지에 대한 증거가 있어야만 이러한 데이터셋에 대해서 인접행렬을 통한 시각화가 노드-링크 다이어그램에 비해서 연결된 두 노드를 찾는 작업을 수행하는데 유리하다는 것을 확실히 증명할 수 있을 것이다.

비슷한 구조를 가지고 있는 네트워크 데이터라도, 데이터의 크기와 밀도에 따라서 결과는 달라질 수 있다. 이번 연구에는 한 가지 데이터만 활용했기 때문에, 후속 연구에서는 같은 구조를 가진 다양한 크기와 밀도의 네트워크 데이터를 대상으로 실험을 진행해보고, 밀도와 크기가 변함에 따라 노드-링크 다이어그램과 인접행렬 두 시각화의 우위가 어떻게 변화하는지를 실험해 볼 수 있을 것이다.

네트워크 데이터의 크기와 밀도 뿐만 아니라, 네트워크의 방향성 유무에 의해서도 결과가 달라질 수도 있다. 후속 연구에서는 네트워크가 데이터가 가지는 구조적 특성은 그대로 유지하되, 방향성이 존재하는 directed 네트워크 데이터를 노드-링크 다이어그램과 인접행렬을 이용하여

시각화를 해 보고, undirected 데이터를 시각화했을 시와 비교하여 어떤 차이점이 존재하는지, 그리고 두 시각화 방법들은 어떤 장점과 단점을 가지는지에 대해서 탐구해볼 수 있을 것이다.

결론 문단에서 언급했듯이, 인접행렬로 이러한 구조의 네트워크 데이터를 시각화 하는 경우에는 [그림 3]과 같이 인접행렬의 일부분을 생략하는 것이 가능하다. 데이터를 시각화 할 때는 항상 표현할 수 있는 공간의 제약이 존재하기 때문에, 이처럼 공간을 절약할 수 있다는 점은 분명히 큰 장점이라고 할 수 있을 것이다. 후속 연구에서는 단순히 공간을 절약하는 것에 그치지 않고, 절약한 공간에 다른 시각화를 결합하여 보다 효율적인 네트워크 데이터의 시각화 방법을 탐구해볼 수 있을 것이다.

## 참고 문헌

1. GHONIEM, Mohammad; FEKETE, J.-D.; CASTAGLIOLA, Philippe. A comparison of the readability of graphs using node-link and matrix-based representations. In: IEEE Symposium on Information Visualization. Ieee, 2004. p. 17-24.
2. KELLER, René; ECKERT, Claudia M.; CLARKSON, P. John. Matrices or node-link diagrams: which visual representation is better for visualising connectivity models?. Information Visualization, 2006, 5.1: 62-76.
3. OKOE, Mershack; JIANU, Radu; KOBOUROV, Stephen. Node-link or adjacency matrices: Old question, new insights. IEEE transactions on visualization and computer graphics, 2018, 25.10: 2940-2952
4. HENRY, Nathalie; FEKETE, Jean-Daniel; MCGUFFIN, Michael J. Nodetrix: a hybrid visualization of social networks. IEEE transactions on visualization and computer graphics, 2007, 13.6: 1302-1309.
5. NOBRE, Carolina, et al. Evaluating multivariate network visualization techniques using a validated design and crowdsourcing approach. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems. 2020. p. 1-12.
6. React, <https://reactjs.org/>
7. D3.js, <https://d3js.org/>