자료구조 중간 대체 프로젝트

그래프의 중요한 노드를 찾아가는 과정에 대한 탐구(Pagerank를 사용하여)

학과 : 정보통계보험수리학과

학번 : 20171421

이름 : 김성연

그래프 상에서 중요한 노드(정점)을 찾는 일은 매우 중요한 일 입니다. 예시로는 네트워크에서 가장 중요한 노드가 무엇인지, 한 사회 내 인간관계에 대해서 핵심 인물은 누구인지, 대한민국 도로망에서 가장 차량통행이 많은 도로(혹은 휴게소)가 어디인지, 서울 지하철 노선도에서 유동인구가 많은 역은 어디인지 등등 많이 있습니다. 예시에서 살펴보듯 그래프는 실 생활 곳곳에 존재하는 현상을 수치화 시킨 것이며, 앞서 말한데로 어느 노드(정점)가 중요한 가를 푸는 문제로 중요한 사회문제를 해결할 수 있습니다.

그렇다면 중요한 노드를 무슨 기준으로 판별할 것인가가 문제입니다. 우선 이는 문제에 따라 다릅니다. 가령 네트워크에서는 허브가 되는 노드가 중요한 노드가 될 것이고, 인간관계에서는 친구가 많은 사람이 될 것입니다. 하지만 다양한 예시들을 살펴보면 중요한 노드가 되는 기준을 크게 두 가지로 분류할 수 있습니다.

1. 다른 노드에서 하이퍼링크를 많이 받는 노드가 더 중요한 노드이다.

2. 중요한 노드(하이퍼링크를 많이 받는)가 주는 하이퍼링크는 상대적으로 덜 중요한 노드가 주는 하이퍼링크에 비해 더 가치가 있다.

쉽게 말해 인간관계에서 핵심인물이 되기 위해선 친구숫자가 절대적으로 많은 것도 중요하고, 유명한 인물(중요한 노드)와 친구인 것 또한 중요합니다.

중요한 노드를 판별하기 위해서 Random Walker 개념을 도입합니다. Random Walker 란 매 시간 가중치에 비례해 무작위로 이웃 노드로 이동하는 Walker 개념입니다. (여기서 가중치가 없거나 가중치가 클수록 관계가 작아지는 그래프는 다루지 않습니다. 가중치가 없는 그래프는 실생활에 많지 않으며 가중치가 있는 그래프를 단순화한 것으로 취급할 수 있습니다. 또 가중치가 클수록 관계가 작아지는 그래프는 다소 어려운 문제이기에 생략합니다.) 이 때 특정 시간동안 Random Walker 가 많이 방문한 노드가 방금 말한 중요한 노드라고 볼 수 있습니다. 왜나하면 중요한 노드라면 연결되어 있는 노드가 많기 때문에 Random Walker 가 방문하기 용이하고, 중요한 노드가 주는 하이퍼링크가 많은 노드여도 Random Walker 가 방문하기 용이합니다. 즉 위 두가지 기준을 만족한다면 Random Walker 가 방문한 횟수가 많을 수록 더 중요한 노드라고 할 수 있습니다. 다만 이런 식으로 Random Walker 를 구성한다면 시작 점 근처에 있는 노드들이 중요한 노드인 것과 별개로 더 많이 방문할 수 있기 때문에 매 시간 q 의 확률로 Random Walker 가 다른 노드로 랜덤하게 이동합니다.

Random Walker 의 구현을 간단히 설명하겠습니다. 우선 시작 정점를 랜덤함수를 통해 정합니다. 여기서 한가지 고려해야 할 것은 간선이 하나도 없는 동떨어진 정점인 경우 q 의 확률로 Random Walker 가 랜덤하게 이동하는 경우 이외에는 Random Walker 가 이동할 수 없게 됩니다. 실제로 간선이 없는 정점은 아까 정의한 두 가지 중요한 정점 기준과 완전히 반대이기 때문에 Random Walker 가 가지 않도록 해줍니다.

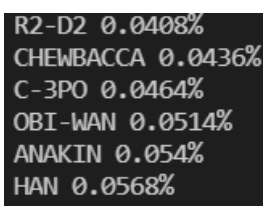
다음으로 q 의 확률로 랜덤하게 다른 정점으로 이동하는 부분을 만들고, 1-q 의 확률로 현재 Random Walker 가 위치한 정점에서 인접한 정점으로 이동하는 부분을 만듭니다. 이 때 간선의 가중치에 비례하여 Random Walker 가 이동할 확률을 올립니다. 이 부분은 연결된 간선의 가중치에 합을 구하는 함수와 함께 구현할 수 있습니다. 그리고 방문기록을 표시해주는 'countVecter' 정수 배열을 만들어서 Random Walker 가 방문할 때 마다 해당 값을 하나씩 올려줍니다.

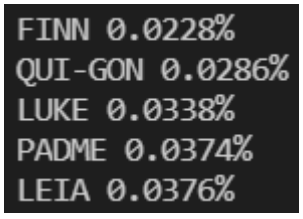
위와 같은 일을 정해진 횟수 N 번을 반복합니다. 그러면 정점 별로 Random Walker 가 방문한 횟수가 기록되는데요. 이 횟수를 각각 N 으로 나눠주면 Random Walker 가 해당 정점을 방문할 확률을 추정한 값이 됩니다. 이를 Pagerank 라고 합니다. 모든 정점의 Pagerank 를 더하면 당연히 1 이 되겠지요. Pagerank 값이 앞서 말한 중요한 노드를 말하는 척도가 됩니다.

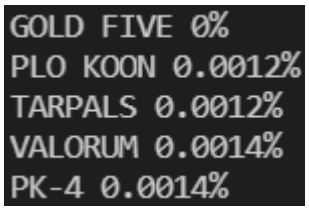
그래프 자료구조를 표현하는 방식은 두 가지가 있는데요. 인접 리스트와 인접 행렬이 있습니다. 이번 Random Walker 를 구하는 과정에서는 인접 리스트를 사용하겠습니다. 왜냐하면 Random Walker 를 구하는 과정에 핵심이 Random Walker 가 간선을 통해 이동한다는 것 입니다. 여기에 가중치가 있는 그래프는 Random Walker 가 이동할 확률을 구해야 하기 때문에 Random Walker 가 머물러 있는 노드에서 연결 된 간선의 모든 가중치의 합 또한 구해야 합니다. 이 두 부분은 인접 행렬보다 인접 리스트가 유리합니다. 인접 행렬은 간선이 없는 부분도 탐색을 해야 합니다. 즉 Random Walker 가 머무른 노드에서 연결 된 가중치의 합을 구할 때 정점의 총 개수 만큼 탐색을 해야하며 실제 Random Walker 가 이동할 노드를 고르는 과정에서도 최악의 경우 정점의 총 개수 만큼 탐색을 해야합니다. 이에 비해 인접 리스트는 간선의 개수만큼만 탐색이 이루어 지기 때문에 인접 리스트를 활용하여 분석하겠습니다.

스타워즈 데이터를 활용하여 방금 말한 Pagerank 이론을 적용하겠습니다. 스타워즈 데이터는 인물의 이름이 쓰인 파일과 인물 1, 인물 2, 같이 나온 횟수를 기록한 파일로 나뉩니다. 여기서 인물을 그래프의 정점으로 보고 두 인물이 같이 나온 장면이 있으면 간선이 있는 것이고 같이 나온 횟수는 가중치라고 생각할 수 있습니다. 같이 나온 횟수가 많으면 많을 수록 더 관계가 깊다는 것이기 때문에 가중치가 클수록 정점 사이의 관계가 더 커지는 그래프입니다. 인물의 이름이 쓰인 파일 형식이 '1 R2-D2'로 되어있는데 앞에 숫자는 큰 의미가 없으므로 엑셀을 통해 1 열을 지운 텍스트 파일을 사용하여 정점을 입력합니다. 인물 1, 인물 2, 같이 나온 횟수를 기록한 파일은 'insertEdge' 함수를 이용하여 간선을 입력하는데 사용됩니다. 이 때 방향성이 없는 데이터이므로 'insertEdge' 함수를 첫 번째, 두 번째 항목을 서로 바꿔서 두 번 적용해야 합니다.

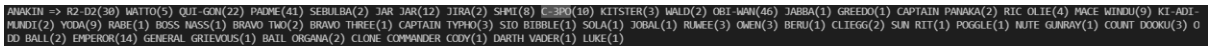
실제 스타워즈 데이터를 활용해 분석한 결과 Pagerank 값이 다음과 같이 나옵니다.(실험 1)

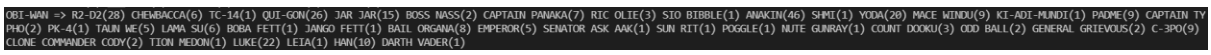






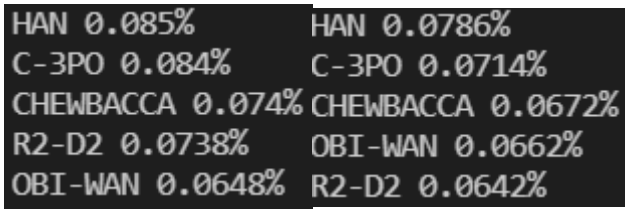
여기서 GOLE FIVE 는 앞서 언급한 대로 간선이 없는 노드이기 때문에 Pagerank 값이 0 프로가 나왔으며, Pagerank 값이 가장 큰 HAN, ANAKIN, OBI-WAN 등은 다른 노드에 비해 간선이 많고 간선들도 가중치가 큽니다.

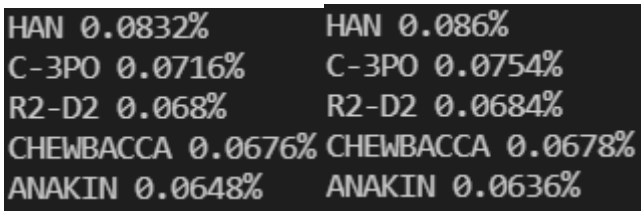




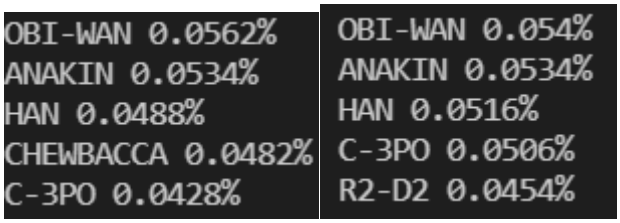
(Q 값은 0.2, N 값은 5000 을 넣었습니다. N 값은 정점 대비 적당히 큰 수를 넣었으며 Q 값은 임의에 값을 넣었습니다. Random Walker 이기 때문에 실행할 때 마다 Pagerank 값은 바뀝니다.) 다음과 같은 실제 데이터 분석을 통해 간선이 많고, 그 간선들의 가중치가 상대적으로 큰 정점을 중요한 정점라고 했을 때 Pagerank 값은 중요한 정점을 나타내는 좋은 지표임을 알 수 있습니다.

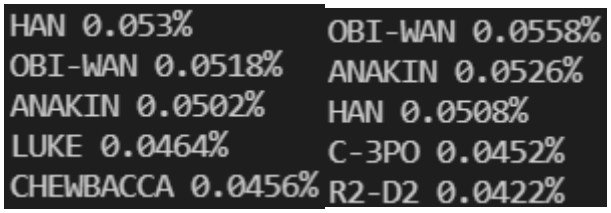
그렇다면 Q 와 N 값은 어떻게 결정해야 하는 것인가가 다음 문제가 됩니다. 우선 Q 값이 무엇인지부터 살펴봅시다. Q 값은 임의의 노드로 점프할 확률로 정의했습니다. 그렇다면 임의의 노드로 점프하는 것이 왜 필요할까요? 앞서 설명한 대로 점프하는 것이 일어나지 않는다면 랜덤으로 선택된 처음 시작한 노드 근처의 Pagerank 값이 상대적으로 크게 나타납니다. 물론 그래프에서 N 의 크기를 크게 하는 경우에 이 부분은 어느정도 상쇄가 됩니다. 이 부분을 확인하기 위해 스타워즈 데이터를 Q = 0, N = 5000 인 형태로 4 번 다시 돌립니다. 또한 동일 데이터를 Q = 20, N = 5000 으로 4 번 돌리고 결과를 비교합니다.(실험 2)





(Q = 0, N = 5000)





(Q = 0.2, N = 5000)

눈에 띄는 점은 Q 의 값을 0 으로 했을때와 비교해 Q 의 값을 0.2 로 하면 Pagerank 의 최대값에 크기가 떨어지는 것 입니다. 이는 앞서 예상한 대로 처음 시작한 정점에 Pagerank 값이 의존한다는 점을 나타냅니다. 또 Q 값이 1 에 가깝게 클 때 또한 부작용이 있음을 예상할 수 있습니다. Q 의 확률로 랜덤으로 노드를 점프하는 것이 Pagerank 값이 처음 시작한 정점에 의존하는 부작용을 방지하기 위해서 인데 Q 값이 크면 Random Walker 의 본질이 무너집니다. 다시 말해 Q 는 부작용을 방지하기 위해서 보정하는 것이고 핵심부분은 1-Q 확률에 있는 것이기 때문에 Q 의 값이 지나치게 큰 것은 주의해야 합니다.

본론으로 돌아와 그러면 Q 값을 0 이나 작은 값, 또 1 에 가까운 큰 값을 쓰면 안된다는 것을 알았는데 적절한 Q 의 값은 어떻게 구해야 할까요? 답은 그래프마다 다르다 입니다. Pagerank 값이 처음 시작한 정점에 많이 의존한다면 Q 값을 조금 크게 설정해야 할 것이고, 그렇지 않다면 Q 의 값을 작게 해야 합니다. 그렇다면 Pagerank 값이 처음 시작한 정점에 의존을 많이 하는 그래프는 어떤 그래프일까요? Random Walker 가 점프 없이 특정 부분에서 빠져 나가기 힘든 그래프 입니다. 특정 부분에 간선이 집중 되있고 그 부분에서 다른 부분으로 가는 간선의 수가 많이 적고 가중치 또한 낮다면 Random Walker 가 점프 없이 빠져나가기 힘들 것 입니다. 시각적으로 그래프가 어떠한 가를 파악하여 그래프마다 다른 Q 값을 적용해야 합니다. 하지만 그래프의 규모가 커질수록 그래프를 직관으로 판단하기 힘들기 때문에 Q 값을 계속 바꿔가면서 그래프의 특성에 맞는 Q 값을 적용할 필요가 있습니다.

다음은 N 값입니다. N 값은 Random Walker 이동 시행횟수 입니다. N 값이 커질 수록 더 일관된 결과를 가져오며 반대로 N 값이 작을 수록 매 실험마다 정점간 Pagerank 값의 크기 순서가 차이가 날 것입니다. 이 부분을 확인하기 위해 스타워즈 데이터를 Q = 0.2 로 고정하고 N = 500 인 형태로 4 번 돌리고, N = 5000000 형태로 4 번 돌리고 결과를 비교합니다. (실험 3)

실험 3 의 결과를 확인해보면 N 의 크기를 키웠을 때에 값이 더 일관된 결과를 가져옵니다. 사용하는 데이터에 정점의 수 등 데이터의 크기에 따라 달라지겠지만 일정 숫자 이상 N 값이 커지면 실제 중요한 노드를 정확히 찾아낼 확률이 높아집니다. 이는 Pagerank 값이 실험횟수가 늘어나면 더 모수에 가까운 값을 추정해준다는 좋은 추정량의 일치성에 성질을 띔을 알 수 있습니다. 다음으로 N 값이 데이터의 크기에 따라 일정 값보다 커져야 한다는 것은 당연한데 그렇다면 컴퓨터가 허용하는 한, N 값을 무수히 크게 하면 되지 않겠는가에 대한 질문이 생깁니다.

```
C-3PO 0.068%
HAN 0.066%
ANAKIN 0.06%
R2-D2 0.05%
OBI-WAN 0.046%

OBI-WAN 0.072%
ANAKIN 0.066%
QUI-GON 0.052%
HAN 0.048%
C-3PO 0.038%

ANAKIN 0.062%
PADME 0.06%
OBI-WAN 0.056%
C-3PO 0.054%
LUKE 0.05%

ANAKIN 0.07%
OBI-WAN 0.058%
C-3PO 0.05%
HAN 0.046%
LUKE 0.044%
```

```
ANAKIN 0.054108%
OBI-WAN 0.053266%
HAN 0.05261%
C-3PO 0.048112%
R2-D2 0.043548%

ANAKIN 0.054384%
OBI-WAN 0.053352%
HAN 0.053176%
C-3PO 0.047466%
R2-D2 0.04383%

ANAKIN 0.054376%
OBI-WAN 0.053182%
HAN 0.052596%
C-3PO 0.048114%
R2-D2 0.043788%

ANAKIN 0.054004%
OBI-WAN 0.05334%
HAN 0.052636%
C-3PO 0.047894%
R2-D2 0.043458%
```

(첫번째 N = 500, 두번째 N = 500000)

N 값이 크면 클수록 주어진 데이터를 더욱 더 극한으로 활용하였다는 의미입니다. N 값을 키우는 데 큰 리스크가 없다면 키우는 것은 좋은 일입니다. 하지만 너무 주어진 데이터에만 과적합하는 것이 아닌가 하는 생각도 해볼 필요가 있습니다. 주어진 데이터가 더 추가될 일 없고, 완벽한 데이터라면 상관 없습니다. 하지만 지금 사용한 스타워즈 데이터 또한 완벽한 데이터는 아닙니다. 중요한 등장인물이라는 것이 얼마나 많은 인물과 많은 횟수로 출연하는지에 대한 수치와 비례하는 것은 맞지만 중요한 등장인물이 꼭 이 부분에만 의존하는 것은 아닙니다. 나온 분량도 중요한 요소가 될 수 있으며, 다수가 출연한 장면이 많다고 해서 꼭 중요한 등장인물이라고 할 수도 없습니다. 또 새로운 시리즈가 나오면 데이터가 추가되기도 합니다. 정리해서 이야기하면 N 값을 키우면 주어진 데이터를 과 적합 할 수도 있으니 유의해야 하며 이 문제는 N 값의 크기 뿐만 아니라 나온 결과를 무조건적으로 받아들이는 것 또한 주의해야 한다는 것 입니다.