과제2 201921521_류한별

소셜 네트워크

소셜 네트워크의 등장

- 스마트 디바이스의 급격한 확산 디바이스의 확산으로 인해 온라인 접근성이 증대되었고 SNS 활용 역시 크게 확산되었다.

- SNS 즉 소셜미딩는 온라인 네트워크와 메시지가 결합된 구조이다.

소셜 미디어의 유형

- 관계.소통형(인맥관리 및 연결)
- 관계.협업형(리뷰,뉴스 등 정보 생산)
- 공유.소통형(동영상 및 사진 공유)
- 공유.협업형(컨텐츠 및 서비스 협업으로 생산 공유)
    - 한 유형의 속성이 아닌 다른 유형의 속성과 일부 공통적 특징을 지닌다.

네트워크란

- 각각의 객체들이 상호 연결된 구조
- 구성요소: 노드와 링크

네트워크 역사

- 오일러의 다리건너기 문제
- 쾨니히스베르크
    - 사람들간의 상호 관계를 과학적 분석의 대상으로 발전시킴
    - 소셜네트워크 분석의 초석이다.
    - 사회측정법인 소이소메트리로 발전
- 바베라스와 리빗의 실험

- 밀그램의 6단계 분리이론
- 그라노베터의 "약간 연결의 강점"
- 프리만의 "중심성 지표"개발
- 버트의 "구조적 공백"
- 왓츠,스트로가츠의 "좁은 세상 네트워크"
- 바라바시,알버트,정하웅의 "무척도 네트워크"

그래프의 기본

- 네트워크를 표현하는 과정에서 중요하게 사용됨
    - 객체들 간의 모델을 수학적으로 표현된 그래프로 나타내는 연구분야 그래프는 노드로 구성, 노드는 링크로 연결되어 있음
    - 무방향 네트워크

    - 방향 네트워크
- 네트워크 그래프에서 노드와 노드간 관계의 유무만이 아닌 관계의 정도를 함께 표시할 수 있음

- 관계의 유무만을 표현한 경우 :이진 네트워크 그래프
- 관계의 정도를 함께 표시한 경우: 가중 네트워크 그래프
- 관계의 정도를 가중치를 직접 표시하거나 선의 굵기를 다르게 표시하는 방법이 있다.

표현방법

- 인접행렬방식
    - 인접한 노드들 간의 연결관계를 행렬로 표현
- 노드리스트 방식
    - 인접한 링크가 있는 노드들만 표기하는 방법
- 엣지 리스트 방식
    - 노드를 연결한 링크들의 리스트 "링크의 목록"

분류방법

- 링크의 방향성 유무와 가중치 유무에 따라 4가지 유형으로 구분

- 관계의 위상에 따른 분류

    - Ring
    - Mesh
    - Star
    - Fully connected
    - Line
    - Tree
    - Bus
- 네트워크 크기에 따른 분류

    - 소규모 (최대 링크 수 <= 10000)
    - 중규모 (최대 링크 수 <=1000000)
    - 대규모 (최대 링크 수 <= 100000000)
    - 초대규모(최대 링크 수 = 무한)
- 분석 대상에 따른 분류

    - 전체 네트워크: 네트워크를 구성하고 있는 모든 노드를 포함한 네트워크 /(완전 네트워크)
    - 하위 네트워크: 전체 네트워크의 일부분으로 부속되어 있는 것 /(컴포넌트, 파당, 클러스터 등)

네트워크 수준의 속성 분석

- 네트워크 크기

    - 네트워크를 구성하는 노드 수
    - 크기가 증가할 수록 구조 복잡성도 증가
    - 소셜네트워크 분석에 있어 중요한 역할

- 네트워크 밀도
  - 네트워크를 구성하는 노드 간 연결된 정도
  - 연결이 많아질 수록 밀도는 높아짐

분석지표

- 중심성 지표
  - 가장 일반적으로 사용되는 지표
  - 한 행위자가 전체 네트워크에서 중심에 위치하는 정도로 표현
  - 전체 네트워크에서 중요한 역할을 하는 노드가 무엇인지 파악 가능
  - 중심적인 역할을 하는 구성원을 탐색, 효율적인 정보 전달과 의사전달을 가능케 함으로 문제해결
  - 연결정도 중심성; 한 노드가 얼마나 많은 다른 노드들과 연결관계를 맺고 있는지 측정
  - 근접 중심성: 한 노드가 얼마나 전체 네트워크의 중심에 근접해 있는지 측정
  - 매개 중심성: 한 노드가 다른 노드들 간의 네트워크 관계 형성에 있어 중개자 혹은 매개자 역할을 얼마나 수행하는 지 측정
  - 아이겐 벡터 중심성: 연결된 다른 노드들의 중요도를 함께 반영

소셜네트워크 분석 도구

- Node XL
  - 마이크로 소프트 엑셀에 애드인 형태로 결합하여 구동
  - 쉽고 간편하게 데이터 수집, 분석, 시각화 가능
  - 유튜브, 트위터 등 다양한 온라인 데이터 수집 가능
- UCINET
  - 다양한 소셜네트워크 분석기법을 적용할 수 있는 종합적인 분석 프로그램
  - 별도로 내장된 프로그램을 통해 시각화 기능 제공
- Pajek
  - 네트워크 분석과 시각화 기능을 제공하는 공개 소프트웨어
  - 대규모 네트워크 분석 가능
- NetMiner
  - UCINET과 KrackPlot장점을 통합하여 개발
  - 분석된 결과물을 프로젝트별로 관리
  - 각 네트워크 분석 지표별 구현된 시각화 기능 제공
- Gephi
  - 탁월한 시각화 기능
  - 모든 종류의 복잡계 네트워크 분석에 적합
  - 최대 백만 노드까지 분석가능

소셜네트워크 분석 사례

- 하이퍼 링크 네트워크 분석
- 트위터 네트워크 분석
- 트위터 에고 네트워크 분석

- 유튜브 동영상 네트워크 분석
- 페이스북 네트워크 분석
- 의미 네트워크 분석

In [105]:
```
import numpy as np
import pandas as pd
import re
import networkx as nx
import matplotlib.pyplot as plt
import seaborn as sns
```

In [106]:
```
data = pd.read_csv('C:/Users/gksquf/OneDrive/Documents/캡디/subway.csv', encoding="ut
data.head()
```

Out[106]:

	from	to	line
0	소요산	동두천	1
1	동두천	보산	1
2	보산	동두천중앙	1
3	동두천중앙	지행	1
4	지행	덕정	1

In [107]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 696 entries, 0 to 695
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   from    696 non-null    object
 1   to      696 non-null    object
 2   line    696 non-null    object
dtypes: object(3)
memory usage: 16.4+ KB
```

In [108]:
```
data.describe()
```

Out[108]:

	from	to	line
count	696	696	696
unique	587	594	22
top	왕십리	공덕	1
freq	4	4	97

In [109]:
```
print(len(data.isnull().any()))
data.isnull().any()
data['from'].replace(np.nan, '0', inplace = True) #null값 채워주기
data['from'].head(10)
```

```
3
```

Out[109]:
```
0      소요산
1      동두천
```

```
2       보산
3     동두천중앙
4       지행
5       덕정
6       덕계
7       양주
8       녹양
9       가능
Name: from, dtype: object
```

In [110]:

```
print( str(data['from'].nunique()))
name = pd.DataFrame(data['from'].unique().tolist(), columns = ['from']) #포켓몬 종류
```

```
587
```

In [119]:

```
g = nx.Graph()
g = nx.nx.from_pandas_edgelist(data, source = 'from', target = 'to')
print(nx.info(g))
```

```
Name:
Type: Graph
Number of nodes: 605
Number of edges: 689
Average degree:   2.2777
```

In [132]:

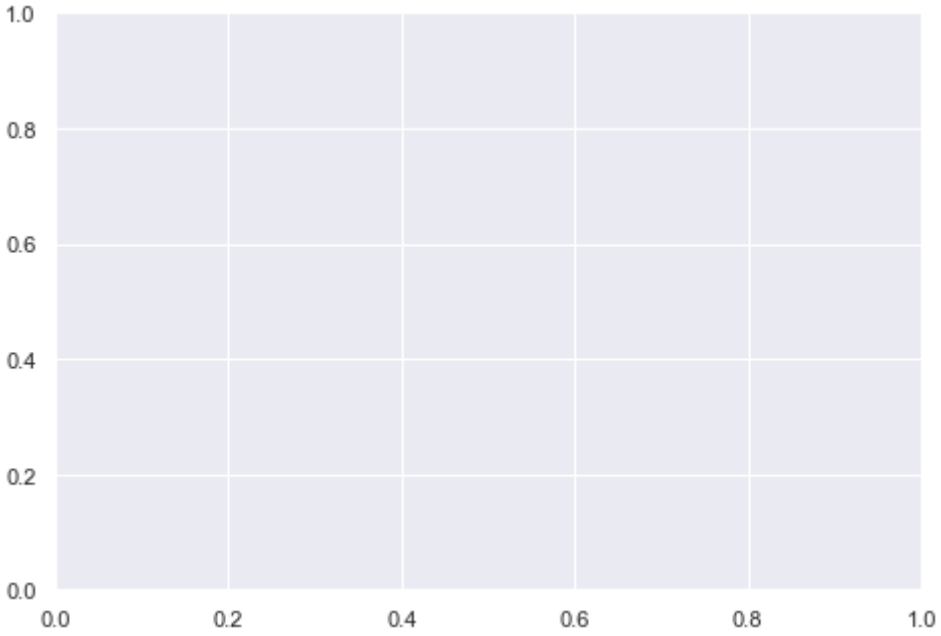
```
pos = nx.spring_layout(g)
nx.draw(g,pos=pos,with_labels=True)
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
~Wanaconda3WlibWsite-packagesWnetworkxWutilsWdecorators.py in _random_state(f
unc, *args, **kwargs)
    395         try:
--> 396             random_state_arg = args[random_state_index]
    397         except TypeError as e:

IndexError: tuple index out of range

The above exception was the direct cause of the following exception:

NetworkXError                             Traceback (most recent call last)
<ipython-input-132-8b05c8088936> in <module>
----> 1 pos = nx.spring_layout(g)
      2 nx.draw(g,pos=pos,with_labels=True)
      3 plt.show()

~Wanaconda3WlibWsite-packagesWdecorator.py in fun(*args, **kw)
    229             if not kwsyntax:
    230                 args, kw = fix(args, kw, sig)
--> 231             return caller(func, *(extras + args), **kw)
    232     fun.__name__ = func.__name__
    233     fun.__doc__ = func.__doc__

~Wanaconda3WlibWsite-packagesWnetworkxWutilsWdecorators.py in _random_state(f
unc, *args, **kwargs)
    398             raise nx.NetworkXError("random_state_index must be an intege
r") from e
    399         except IndexError as e:
--> 400             raise nx.NetworkXError("random_state_index is incorrect")
from e
    401
    402         # Create a numpy.random.RandomState instance

NetworkXError: random_state_index is incorrect
```

In [129]:

```
nx.draw(g,with_labels=True, node_color = 'blue')
plt.show(g)
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
~Wanaconda3WlibWsite-packagesWnetworkxWutilsWdecorators.py in _random_state(f
unc, *args, **kwargs)
    395         try:
--> 396             random_state_arg = args[random_state_index]
    397         except TypeError as e:

IndexError: tuple index out of range

The above exception was the direct cause of the following exception:

NetworkXError                             Traceback (most recent call last)
<ipython-input-129-62986ea9327c> in <module>
----> 1 nx.draw(g,with_labels=True, node_color = 'blue')
      2 plt.show(g)

~Wanaconda3WlibWsite-packagesWnetworkxWdrawingWnx_pylab.py in draw(G, pos, a
x, **kwds)
    121         kwds["with_labels"] = "labels" in kwds
    122
--> 123     draw_networkx(G, pos=pos, ax=ax, **kwds)
    124     ax.set_axis_off()
    125     plt.draw_if_interactive()

~Wanaconda3WlibWsite-packagesWnetworkxWdrawingWnx_pylab.py in draw_networkx
(G, pos, arrows, with_labels, **kwds)
    331
    332     if pos is None:
--> 333         pos = nx.drawing.spring_layout(G)  # default to spring layout
    334
    335     draw_networkx_nodes(G, pos, **node_kwds)

~Wanaconda3WlibWsite-packagesWdecorator.py in fun(*args, **kw)
    229             if not kwsyntax:
    230                 args, kw = fix(args, kw, sig)
--> 231             return caller(func, *(extras + args), **kw)
    232     fun.__name__ = func.__name__
    233     fun.__doc__ = func.__doc__

~Wanaconda3WlibWsite-packagesWnetworkxWutilsWdecorators.py in _random_state(f
unc, *args, **kwargs)
    398             raise nx.NetworkXError("random_state_index must be an intege
r") from e
    399         except IndexError as e:
--> 400             raise nx.NetworkXError("random_state_index is incorrect")
from e
    401
    402         # Create a numpy.random.RandomState instance

NetworkXError: random_state_index is incorrect
```



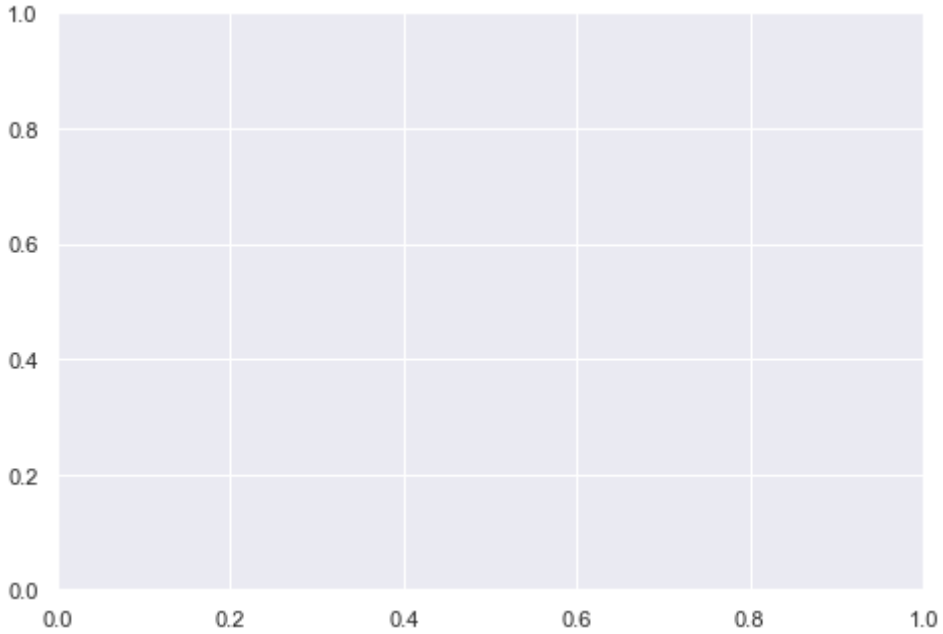
In [135]:
```
import networkx as nx
import matplotlib.pyplot as plt

# 그래프 생성
G = nx.DiGraph()
```

In [136]:
```
G.add_nodes_from([1, 2, 3, 4, 5])
```

In [141]:
```
G.add_edges_from([(1, 2), (2, 1), (2, 3), (2, 3), (2, 3), (2, 3), (2, 3), (2, 3), (2,
                  (2, 3), (2, 3), (2, 3), (2, 3), (2, 3), (2, 3), (2, 3), (2, 3), (2, 3
degree = nx.degree(G)
print(degree)
nx.draw(G, with_labels=True)
```

```
[(1, 2), (2, 5), (3, 1), (4, 2), (5, 0)]
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
~Wanaconda3WlibWsite-packagesWnetworkxWutilsWdecorators.py in _random_state(f
unc, *args, **kwargs)
    395         try:
--> 396             random_state_arg = args[random_state_index]
    397         except TypeError as e:

IndexError: tuple index out of range

The above exception was the direct cause of the following exception:

NetworkXError                             Traceback (most recent call last)
<ipython-input-141-28fd67fd0d1b> in <module>
      3 degree = nx.degree(G)
      4 print(degree)
----> 5 nx.draw(G, with_labels=True)

~Wanaconda3WlibWsite-packagesWnetworkxWdrawingWnx_pylab.py in draw(G, pos, a
x, **kwds)
    121         kwds["with_labels"] = "labels" in kwds
    122
--> 123     draw_networkx(G, pos=pos, ax=ax, **kwds)
    124     ax.set_axis_off()
    125     plt.draw_if_interactive()
```

```
~₩anaconda3₩lib₩site-packages₩networkx₩drawing₩nx_pylab.py in draw_networkx
(G, pos, arrows, with_labels, **kwds)
    331
    332     if pos is None:
--> 333         pos = nx.drawing.spring_layout(G)  # default to spring layout
    334
    335     draw_networkx_nodes(G, pos, **node_kwds)

~₩anaconda3₩lib₩site-packages₩decorator.py in fun(*args, **kw)
    229             if not kwsyntax:
    230                 args, kw = fix(args, kw, sig)
--> 231             return caller(func, *(extras + args), **kw)
    232     fun.__name__ = func.__name__
    233     fun.__doc__ = func.__doc__

~₩anaconda3₩lib₩site-packages₩networkx₩utils₩decorators.py in _random_state(f
unc, *args, **kwargs)
    398             raise nx.NetworkXError("random_state_index must be an intege
r") from e
    399         except IndexError as e:
--> 400             raise nx.NetworkXError("random_state_index is incorrect")
from e
    401
    402         # Create a numpy.random.RandomState instance

NetworkXError: random_state_index is incorrect
```



- 다양한 예제로 시도했는데 그래프가 나오지 않습니다..어떤 부분이 잘못되었는지 궁금합니다ㅠ

In [ ]: