

NetworkX를 이용한 네트워크 분석

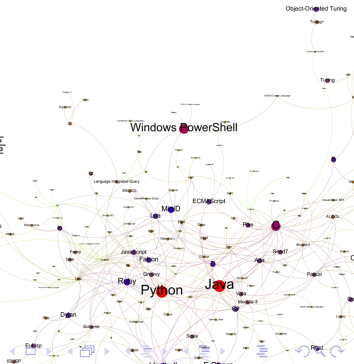
김경훈

유니스트 수리과학과

kyunghoon@unist.ac.kr

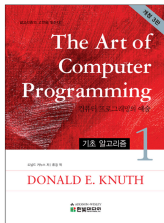
2014년 8월 30일

숙명여자대학교 창학관 켄마홀



이번 TALK의 목적

① LaTeX



```
\begin{frame}
\frametitle{이번 TALK의 목적}
\begin{enumerate}
\item LaTeX\\
\includegraphics[scale=0.2]{donald.png}
\includegraphics[scale=0.5]{latex.png}
\item 점과 선의 자유로움
\end{enumerate}
\end{frame}
```

② 점과 선의 자유로움

③ 파이썬이라서 자유로운 점

About me

Speaker

김경훈 (대학원생)

UNIST (Ulsan National Institute of Science and Technology)

자연과학부 수리과학과

Lab

Adviser : Bongsoo Jang

Homepage : <http://amath.unist.ac.kr>



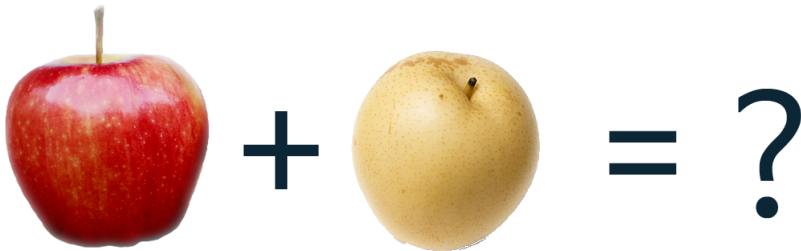
"Be the light that shines the world with science and technology."

목차

- ① 네트워크?
- ② 네트워크 그리기
- ③ 네트워크 계산
- ④ 데모
 - 드라마 네트워크
 - 프로그래밍 언어 네트워크
 - 페이스북 친구 네트워크
- ⑤ 기타 기능

네트워크?

데이터 추상화



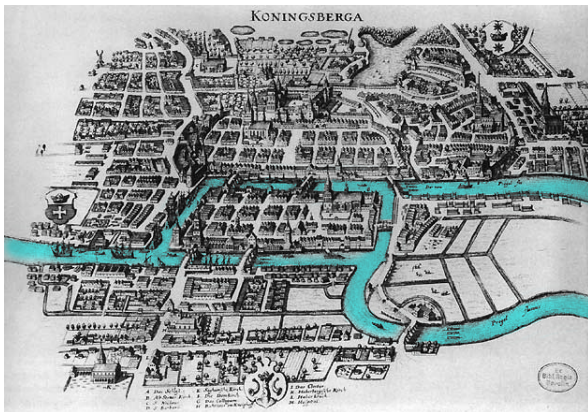
네트워크?

데이터 추상화



네트워크?

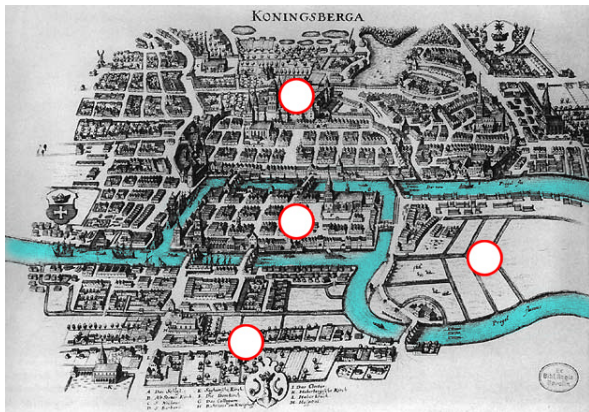
데이터 추상화



<http://en.wikipedia.org/wiki/K%C3%B6nigsberg>

네트워크?

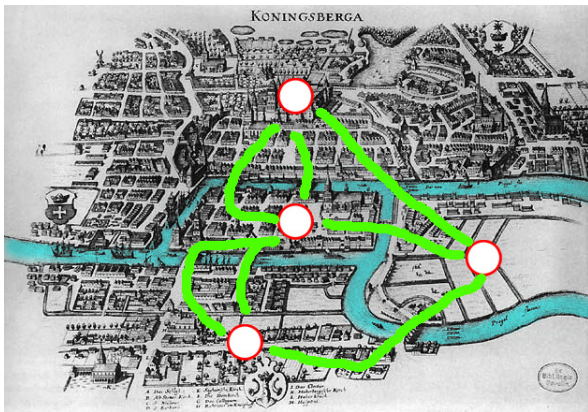
데이터 추상화



<http://en.wikipedia.org/wiki/K%C3%B6nigsberg>

네트워크?

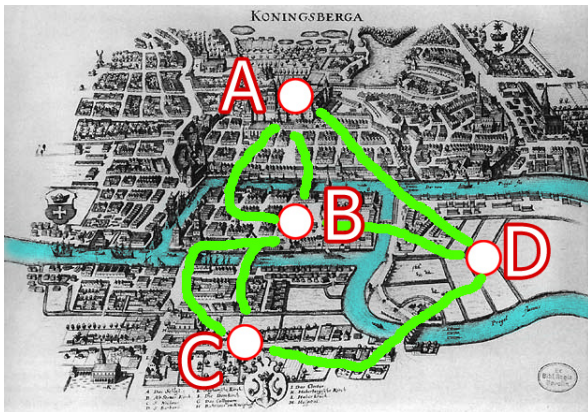
데이터 추상화



<http://en.wikipedia.org/wiki/K%C3%B6nigsberg>

네트워크?

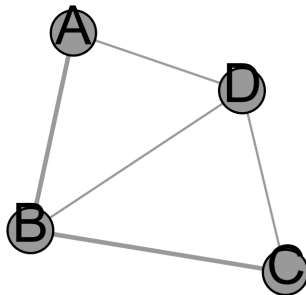
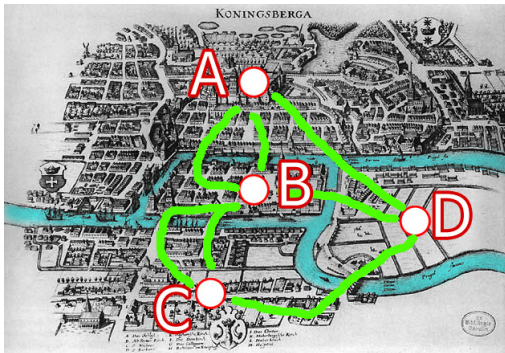
데이터 추상화



<http://en.wikipedia.org/wiki/K%C3%B6nigsberg>

네트워크?

데이터 추상화



네트워크?

```
>>> G = nx.MultiGraph()
>>> G.add_nodes_from(['A', 'B', 'C', 'D'])
>>> G.add_edges_from([('A', 'B'), ('A', 'B'), ('A', 'D'), ('B', 'D'),
, ('B', 'C'), ('B', 'C'), ('C', 'D')])
>>> nx.draw(G)
>>> plt.show()
```


네트워크?

네트워크	노드(Node)	엣지(Edge)
인터넷 네트워크	컴퓨터	케이블
월드와이드웹	웹 페이지	하이퍼링크
인용 네트워크	글	인용
전력망	발전기	전선
친구 네트워크	사람	친구관계
신진대사 네트워크	대사 산물	반응
신경망	뉴런	시냅스
먹이사슬 네트워크	종	포식

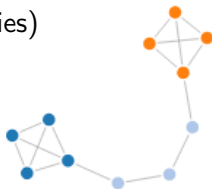
표: 네트워크 예제

NetworkX : Python Library

NetworkX is a Python language software package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.

Features

- ① 무방향성, 방향성, 다중그래프 등의 데이터 구조
- ② Nodes can be "anything" (e.g. text, images, XML records)
- ③ Edges can hold arbitrary data (e.g. weights, time-series)
- ④ 표준적인 그래프 알고리즘.
- ⑤ BSD 라이선스
- ⑥ Well tested: more than 1800 unit tests
- ⑦ PYTHON LIBRARY!



NetworkX 설치방법

- 수동 설치

<https://pypi.python.org/pypi/networkx/>

- 자동 설치

easy_install networkx

pip install networkx

sudo apt-get install python-networkx

- 통합 설치

Enthought Canopy

Continuum Analytics Anaconda

Python(x,y)

WinPython

네트워크의 간단한 예

```
>>> import networkx as nx % 라이브러리를 nx로 불러오기
```

네트워크의 간단한 예

```
>>> import networkx as nx % 라이브러리를 nx로 불러오기  
>>> import matplotlib.pyplot as plt % plt로 불러오기
```

네트워크의 간단한 예

```
>>> import networkx as nx % 라이브러리를 nx로 불러오기
>>> import matplotlib.pyplot as plt % plt로 불러오기
>>> G = nx.Graph() % 빈 그래프 구조 G 생성
```

네트워크의 간단한 예

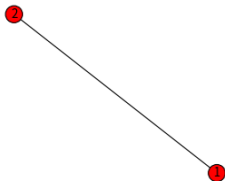
```
>>> import networkx as nx % 라이브러리를 nx로 불러오기
>>> import matplotlib.pyplot as plt % plt로 불러오기
>>> G = nx.Graph() % 빈 그래프 구조 G 생성
>>> G.add_edge(1,2) % 엣지 추가
```

네트워크의 간단한 예

```
>>> import networkx as nx % 라이브러리를 nx로 불러오기
>>> import matplotlib.pyplot as plt % plt로 불러오기
>>> G = nx.Graph() % 빈 그래프 구조 G 생성
>>> G.add_edge(1,2) % 엣지 추가
>>> nx.draw(G) % 그래프 G 그리기
```


네트워크의 간단한 예

```
>>> import networkx as nx % 라이브러리를 nx로 불러오기
>>> import matplotlib.pyplot as plt % plt로 불러오기
>>> G = nx.Graph() % 빈 그래프 구조 G 생성
>>> G.add_edge(1,2) % 엣지 추가
>>> nx.draw(G) % 그래프 G 그리기
>>> plt.show() % pyplot으로 보여주기
```



노드를 정의하는 방법

```
>>> G = nx.Graph()
>>> G.add_node('apple') % 노드 'apple' 추가
>>> G.add_nodes_from(['banana', 'kiwi', 'mango'])
% 리스트로 추가 1

or

>>> fruits = ['banana', 'kiwi', 'mango']
>>> G.add_nodes_from(fruits) % 리스트로 추가 2

>>> G.nodes() % 노드들 보기
['apple', 'kiwi', 'mango', 'banana']
```

엣지를 정의하는 방법

```
>>> G = nx.Graph()

>>> G.add_edge('apple', 'banana') % 'apple'과 'banana'의 관계(엣지) 추가

>>> G.add_edges_from([('apple', 'mango'), ('apple', 'kiwi')])

% 리스트로 추가 1

or

>>> relations = [('apple', 'mango'), ('apple', 'kiwi')]

>>> G.add_edges_from(relations) % 리스트로 추가 2

>>> G.edges() % 엣지들 보기

[('apple', 'banana'), ('kiwi', 'apple'), ('mango', 'apple')]
```

노드의 속성 부여하기 1

```
>>> G.nodes()
['kiwi', 'mango', 'apple', 'banana']
>>> G.node['kiwi']
{}
>>> G.node['kiwi']['kind'] = 'fruit'
>>> G.node['kiwi']
{'kind': 'fruit'}
>>> G.nodes(data=True)
[('kiwi', {'kind': 'fruit'}), ('mango', {}), ('apple', {}),
 ('banana', {})]
```

노드의 속성 부여하기 2

```
>>> G.add_node('kiwi', kind='fruit')
>>> G.add_nodes_from(['banana', 'apple'], kind='fruit')
>>> G.node['banana']
{'kind': 'fruit'}
```

엣지에 속성 부여하기 1

```
>>> G.edges()
[('kiwi', 'apple'), ('mango', 'apple'), ('apple', 'banana')]
>>> G.edges(data=True)
[('kiwi', 'apple', {}), ('mango', 'apple', {}), ('apple',
  'banana', {})]

>>> G.add_edge('apple', 'mango', weight=2.5)
>>> G.add_edges_from(relations, color='blue')
>>> G['apple']['mango']['weight']=5
>>> G.edge['apple']['kiwi']['weight']=2
```

엣지에 속성 부여하기 2

```
>>> G.edges(data=True)
[('kiwi', 'apple', {'color': 'blue'}),
 ('mango', 'apple', {'color': 'blue', 'weight': 5}),
 ('apple', 'banana', {})]
```

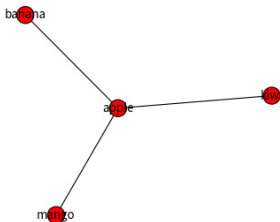
네트워크 그리기

네트워크가 정의되기 위해서는?

- 점(Objects, Vertices, Nodes, Sites, Actors, ...)
- 선(Relations, Edges, Links, Bonds, Ties, ...)
- +
- 점의 위치

네트워크 그리기

```
>>> G = nx.Graph()
>>> relations = [('apple', 'banana'), ('kiwi', 'apple'),
                 ('mango', 'apple')]
>>> G.add_edges_from(relations) % 점, 선 생성
>>> nx.draw(G) % 점의 위치를 spring layout으로 생성
>>> plt.show()
```



네트워크 그리기

```
>>> nx.draw(G) % 기본 그리기
>>> nx.draw_circular(G) % 원 위에 노드 놓기
>>> nx.draw_graphviz(G) % Graphviz 사용
>>> nx.draw_random(G) % 균등 분포를 이용한 랜덤
>>> nx.draw_shell(G) % 동심원 위에 노드 놓기
>>> nx.draw_spectral(G) % 그래프 라플라시안의 고유 벡터 기반
>>> nx.draw_spring(G) % Fruchterman-Reingold force-directed alg. 기반
```

점의 위치 정하기

```
>>> pos = nx.spectral_layout(G) % nx.spring_layout, ...
>>> print pos
{'kiwi': array([ 1.,  0.]), 'mango': array([ 0.,  1.]),
 'apple': array([ 0.66666667,  0.66666667]), 'banana':
  array([ 1.,  1.])}

>>> nx.draw(G, pos)
>>> plt.show()
```

지금까지 한 것

지금까지 한 것

- 점, 선 정의
- 점, 선 속성 정의
- 점 위치 정의



지금까지 한 것

지금까지 한 것

- 점, 선 정의
- 점, 선 속성 정의
- 점 위치 정의

뒤에서 할 것

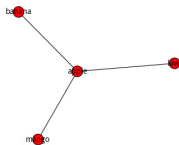
- 점의 **특징** 계산

네트워크 계산

각 점의 특징 계산

```
>>> G.nodes()
['kiwi', 'mango', 'apple', 'banana']

>>> nx.to_numpy_matrix(G)
matrix([[ 0.,  0.,  1.,  0.],
        [ 0.,  0.,  1.,  0.],
        [ 1.,  1.,  0.,  1.],
        [ 0.,  0.,  1.,  0.]])
```



네트워크 $\Rightarrow$ 행렬 $\Rightarrow ?_1$

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

네트워크 $\Rightarrow$ 행렬 $\Rightarrow ?_1$

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 3 \\ 1 \end{bmatrix}$$

Degree 계산하기

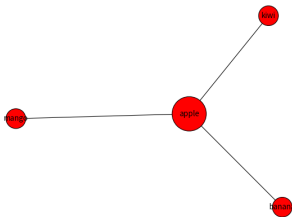
```
>>> print G.degree()
```

```
'kiwi': 1, 'mango': 1, 'apple': 3, 'banana': 1
```

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 3 \\ 1 \end{bmatrix}$$

Degree에 따른 노드의 크기 변화

```
>>> relations = [('kiwi', 'apple'), ('mango', 'apple'),  
('apple', 'banana')]  
>>> G.add_edges_from(relations)  
>>> degree = nx.degree(G)  
>>> nx.draw(G, node_size=[v*1000 for v in degree.values()])  
>>> plt.show()
```



nx.draw()

```
nx.draw(G, pos=None, ax=None, hold=None, **kwds)
```

kwds 문서 링크

```
: pos, with_labels, ax, nodelist, edgelist, node_size,  
   node_color, node_shape, alpha, cmap, vmin, vmax,  
   linewidths, width, edge_color, edge_cmap, edge_vmin,  
   edge_vmax, style, labels, font_size, font_color,  
   font_weight, font_family, label, ...
```

Degree Rank 그리기

```
__author__ = """Aric Hagberg <aric.hagberg@gmail.com>""" 문서 링크
```

```
import networkx as nx
```

```
import matplotlib.pyplot as plt
```

```
G = nx.gnp_random_graph(100,0.02) % binomial_graph 생성
```

```
degree_sequence=sorted(nx.degree(G).values(),reverse=True) # Degree 리스트 값 생성
```

```
dmax=max(degree_sequence)
```

```
plt.loglog(degree_sequence,'b-',marker='o')
```

```
plt.title("Degree rank plot")
```

```
plt.ylabel("degree")
```

```
plt.xlabel("rank")
```

```
# draw graph in inset
```

```
plt.axes([0.45,0.45,0.45,0.45])
```

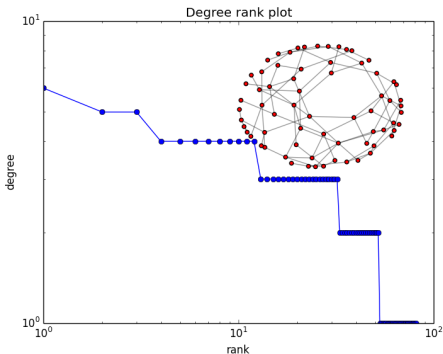
```
pos=nx.spring_layout(G)
```

```
plt.axis('off')
```

```
nx.draw_networkx_nodes(G,pos,node_size=20)
```

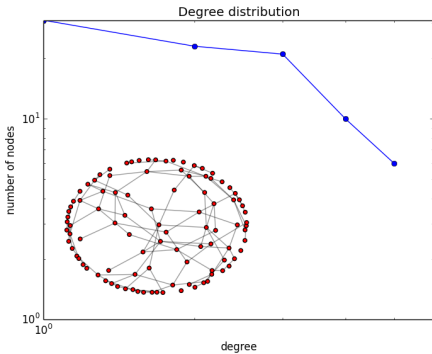
```
nx.draw_networkx_edges(G,pos,alpha=0.4)
```

```
plt.savefig("degree_rank.png")
```



Degree 분포 그리기

```
__author__ = """Aric Hagberg <aric.hagberg@gmail.com>"""  
import networkx as nx  
import matplotlib.pyplot as plt  
G = nx.gnp_random_graph(100,0.02) # binomial_graph 생성  
  
degrees = nx.degree(G).values()  
dmax=max(degrees)  
  
h, bins = np.histogram(degrees, bins=dmax)  
hmax = max(h)  
plt.axis([1, dmax, 1, hmax])  
x=bins.compress(h)  
y=h.compress(h)  
  
plt.loglog(x,y, 'bo-')  
plt.title("Degree distribution")  
plt.xlabel("degree")  
plt.ylabel("number of nodes")  
plt.savefig("degree_distribution.png")  
plt.show()  
# plt.axes([0.1,0.1,0.5,0.5])
```

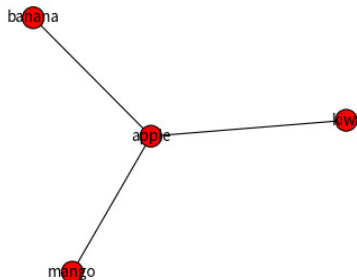


네트워크 $\Rightarrow$ 행렬 $\Rightarrow ?_2$

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}^2 = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 3 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

네트워크 $\Rightarrow$ 행렬 $\Rightarrow ?_2$

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}^2 = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 3 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$



```
>>> G.nodes()
```

```
['kiwi', 'mango', 'apple', 'banana']
```

네트워크 $\Rightarrow$ 행렬 $\Rightarrow ?_2$

```
>>> A = nx.to_numpy_matrix(G)
>>> print A
matrix([[ 0.,  0.,  1.,  0.],
        [ 0.,  0.,  1.,  0.],
        [ 1.,  1.,  0.,  1.],
        [ 0.,  0.,  1.,  0.]])
>>> print A**2
matrix([[ 1.,  1.,  0.,  1.],
        [ 1.,  1.,  0.,  1.],
        [ 0.,  0.,  3.,  0.],
        [ 1.,  1.,  0.,  1.]])
```


경로 길이

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}^2 = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 3 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

$$N_{ij}^{(2)} = \sum_{k=1}^n A_{ik} A_{kj} = [\mathbf{A}^2]_{ij}$$

$$N_{ij}^{(3)} = \sum_{k,l=1}^n A_{ik} A_{kl} A_{lj} = [\mathbf{A}^3]_{ij}$$

최단 경로 알고리즘 [문서 링크](#)

- `nx.shortest_path`
- `nx.all_shortest_paths`
- `nx.shortest_path_length`
- `nx.average_shortest_path_length`
- `nx.has_path`

다익스트라 알고리즘

- `dijkstra_path`
- `dijkstra_path_length`

네트워크의 중심성

```
>>> nx.betweenness centrality(G)
'kiwi': 0.0, 'mango': 0.0, 'apple': 1.0, 'banana': 0.0
>>> nx.closeness centrality(G)
'kiwi': 0.6, 'mango': 0.6, 'apple': 1.0, 'banana': 0.6
>>> nx.degree centrality(G)
'kiwi': 0.33, 'mango': 0.33, 'apple': 1.0, 'banana': 0.33
```

내용 설명 링크 [71페이지](#), [66페이지](#), [60페이지](#)

네트워크 $\Rightarrow$ 행렬 $\Rightarrow ?_3$

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

행렬의 고유값, 고유벡터

```
>>> A = nx.to_numpy_matrix(G)
>>> [w,v]=np.linalg.eig(A)
>>> w # eigenvalues
array([ -1.73205081e+00,   1.79977561e-17,   1.73205081e+00,
        0.00000000e+00])
>>> v # normalized eigenvectors
matrix([[ 4.08248290e-01, -8.16496581e-01,  4.08248290e-01,
          0.00000000e+00],
        [ 4.08248290e-01,  4.08248290e-01,  4.08248290e-01,
          -7.07106781e-01],
        [-7.07106781e-01,  9.45046688e-17,  7.07106781e-01,
          0.00000000e+00],
        [ 4.08248290e-01,  4.08248290e-01,  4.08248290e-01,
```

고유벡터 중심성

```
>>> nx.eigenvector centrality(G)
Traceback (most recent call last)
  File <stdin>, line 1, in <module>
    File eigenvector.py, line 103, in eigenvector centrality
      power iteration failed to converge in %d iterations.
networkx.exception.NetworkXError eigenvector centrality():
power iteration failed to converge in %d iterations.%(i+1)
```

내용 설명 링크 58페이지

NetworkX 내용 정리

- 데이터 추상화
- 객체(노드)와 관계(엣지)의 자유로운 정의
- `import networkx as nx`
- `G = nx.Graph()`
- 노드, 엣지 추가
- 노드, 엣지 속성 부여
- 네트워크 그리기
- 행렬 아이디어 $?_1$, $?_2$, $?_3$

- ① 드라마 네트워크
- ② 프로그래밍 언어 네트워크
- ③ 페이스북 네트워크

드라마 대본을 이용한 네트워크

부산과학고등학교 2014 R&E

연구자 : 권기영, 김채린, 배지환, 이가영, 한송미



드라마 대본을 이용한 네트워크

한국콘텐츠진흥원 KOCCA 방송대본데이터베이스

<http://db.kocca.kr/db/bbs/read/broadcastdb.do?nttNo=1931073>



-

III

+

◀

/ 55

GO

▶

1 회

▼

검색

한 상 의 커튼 제 1 회

S# 1, 바닷가 도로 (D)

달리는 철수의 낯은 지프차.
철수 급히 운전 중이고, 옆 좌석 덕구 함께 앉았는데, 둘 긴장한 표정
(분위기가 살짝하게 어딘가로 담그려 가는...)

철수 연장 행했지?
덕구 (고덕하지만 불안) 형 우리 이제 하지말자! 거긴 우리 구역도 아니잖아! 난 못가!
철수 너 지금 배신이라 하겠다는 거냐?
덕구 (미 악문는)
철수 형도 너 없으면 안 돼! (목 잘라) 막 악기 나뉘어야! 이참에 구역도 넓히고, (다짐)해 보자.

괴로워하는 덕구와 함께 철수차 급하게 달리고.

S# 2, 여탕 발의실 (D)

철수 덕구, 문 살짝 열고 들으려는 풍은, 비정한다.
보면 입구 문에 쓰여진 붉은색 '여탕'
표지판만 시골 목욕탕.
둘네 아줌마들 목욕하다 쉬는 적당히 가려 주고 입에 준 차림으로
등상 위에 둘러 앉아, 계란 까먹으려 고스를 중인데, 재갈인 양 느긋
모양 젊은 손님을 구석 수련실 커튼 뒤에 우르르 숨어 머리만 내놓고 보는.

탕주의 아줌마, 철수 일행 기다린 듯 맞이하고

철수 (뚝어행적 들어매고) 어딤니까? (뻔뻔당당)

보면 입구엔 카운터에 판매 위해 걸려둔 풀들이 여자숙도 있고,
덕구 보고는 얼른 쪽팔려 고개 푹 숙였는데,

철수 연장 준비했지? 들어가자.

철수 덕구 여자를 일행 지나치며 탕 쪽으로 들어서는데
아줌마들 등상은 주책스레 신나,
몸 앞은 여자들은 구석 등상은 짜증

아줌마1 아쭈 총각들이 여탕 구경 오고 좋겠네~.
일동 (까르르~.)

아줌마2 जि다 아줌마네 뭐 좋아~, 총각들이 봐주면 우리가 더 좋지~
일동 (더욱 주책스레 까르르~.)

철수 뻔뻔 당당인데,
덕구 괴롭고 쪽팔려 별게지고,
마침 입구 들어서려는데,
그저이 나오는 다음 둘네 아줌마와 덕구 마주칠 뻔,
덕구 암마!~! 화들짝.

아줌마3 (태연뻔뻔) 잘 놀러~ (덕구 등판 철석한 대 치머칠 총게 생겼네.

김경훈 (UNIST)

NetworkX with Network Analysis

2014년 8월 30일

54 / 94

드라마 대본을 이용한 네트워크

동시 출현성 (Co-Occurrence)

- 객체(노드) : 등장인물
- 관계(엣지) : 같은 문장 내에 등장하는 인물들

그래. 아간 정말 폭풍전야 같았어. 장철수는 이제 안나한테 죽~었어~!!

장철수 - 안나

드라마 대본을 이용한 네트워크

동시 출현성 (Co-Occurrence)

- 객체(노드) : 등장인물
- 관계(엣지) : 화자(말하는 사람)가 언급하는 인물

빌리 : 그래. 아깐 정말 폭풍전야 같았어. 장철수는 이제 안나한테 죽~었어~!!

빌리 - 장철수

빌리 - 안나

노드(Node)와 엣지(Edge)는
어떤 것도 될 수 있다!

드라마 대본을 이용한 네트워크

씬/45 철수집 거실 (D)

철수 화장실 나와서

‘전화가 왜 이렇게 돼 있냐,,’ 툭 올려 두고, ‘근데 내 핸드폰 어딴냐,,’ 찾는데
안나 핸드폰 꽃아 둔 쇼파위에 깔고 앉는다.

안나 : 꼬시다 꽃다발.. (씩소 하는데)

철수 : (나두고 가지뭐,,) 나 나간다. 애들 밥 챙겨라. (하고)

안나 : 예씨.. 왜 또 기어나가는 거야. 지금 가면 꽃다발 만날 텐데..

씬/46 철수마당 (D)

철수 나가는데 안나 나오며 ‘장철수~’

철수 보면

안나 : 꽃순이 집이 부서진 거 같던데, 봤어? 고쳐 주고 가.

철수 : 꽃순이가 일부러 부서논 거야. (가면)

드라마 대본을 이용한 네트워크

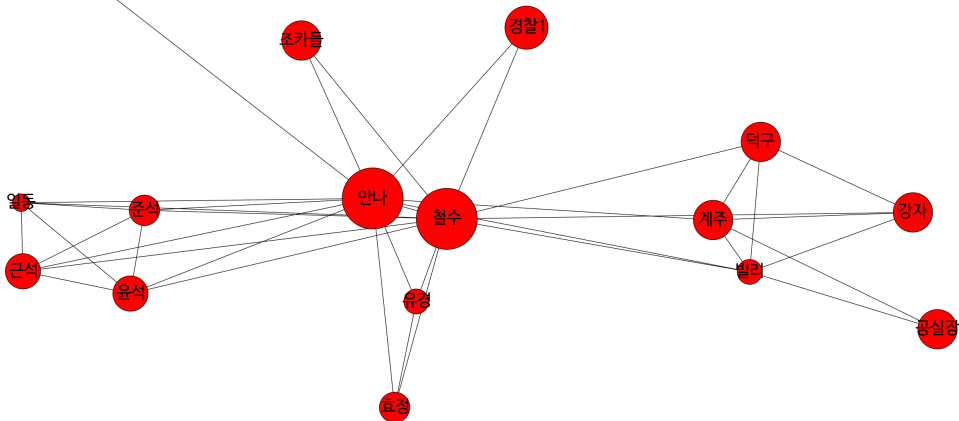
동시 출현성 (Co-Occurrence)

- 객체(노드) : 등장인물
- 관계(엣지) : 같은 씬에 등장하는 인물들

```
s = read(filename)
contents = s.split('씬/')
for content in contents:
    ...
    G.add_edge(등장인물1, 등장인물2)
    ...
```

드라마 대본을 이용한 네트워크

```
degree = nx.degree(G).values()  
nx.draw(G, node_size=[v*1000 for v in degree], font_size=30)  
plt.show()
```



드라마 대본을 이용한 네트워크

Network Information

```
print "Number of Nodes : ", nx.number_of_nodes(G)
print "Number of Edges : ", nx.number_of_edges(G)
degreelist = list(G.degree().values())
print "Avg. Node Degree : ", float(sum(degreelist))/nx.number_of_nodes(G)
print "Avg. Path Length : ", nx.average_shortest_path_length(G)
print "Avg. Clustering Coefficient : ", nx.average_clustering(G)
```

Number of Nodes : 16

Number of Edges : 38

Avg. Node Degree : 4.75

Avg. Path Length : 1.775

Avg. Clustering Coefficient : 0.765340909091

드라마 대본을 이용한 네트워크

Network Information

```
print "::: Betweenness Centrality"
x = nx.betweenness_centrality(G)
sorted_list = sorted(x.iteritems(), key=operator.itemgetter(1), reverse=True)
for s in sorted_list[:20]:
    print s[0], s[1]
```

::: Betweenness Centrality

안나 0.400793650794

철수 0.328571428571

빌리 0.09444444444444

계주 0.0571428571429

강자 0.00238095238095

덕구 0.00238095238095

드라마 대본을 이용한 네트워크

Network Information

```
print " ::: Closeness Centrality"
x = nx.closeness centrality(G)
sorted_list = sorted(x.iteritems(), key=operator.itemgetter(1), reverse=True)
for s in sorted_list[:20]:
    print s[0], s[1]
```

::: Closeness Centrality

안나 0.833333333333

철수 0.833333333333

빌리 0.625

계주 0.6

근석 0.576923076923

일동 0.576923076923

드라마 대본을 이용한 네트워크

참고자료

- 언어 네트워크 분석으로 신뢰-비신뢰 요인 한눈에 본다
- 공공앱의 사용자 리뷰에 대한 분석 : 언어네트워크분석을 중심으로

드라마 대본을 이용한 네트워크

참고자료

- 언어 네트워크 분석으로 신뢰-비신뢰 요인 한눈에 본다

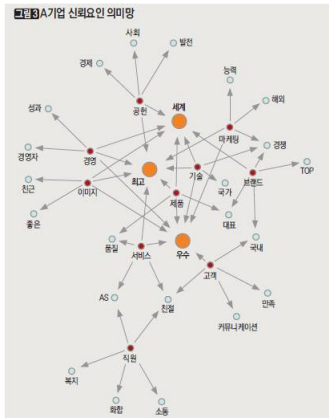
표1 A기업 신뢰인식 핵심 단어군

Centrality Rank	신뢰요인			비신뢰요인		
	Words	Centrality	Word frequency	Words	Centrality	Word frequency
1	제품	0.680444	135	이미지	0.589007	50
2	이미지	0.40982	72	경영	0.527949	50
3	기술	0.39925	58	윤리	0.389136	39
4	브랜드	0.308561	51	직원	0.332653	25
5	서비스	0.303627	46	제품	0.228202	18
6	경영	0.296589	37	공헌	0.225522	23
7	고객	0.206767	30	서비스	0.158099	20
8	직원	0.18703	16	고객	0.153327	16
9	마케팅	0.163231	18	문화	0.134534	12
10	공헌	0.156125	20	경영자	0.106607	6

드라마 대본을 이용한 네트워크

참고자료

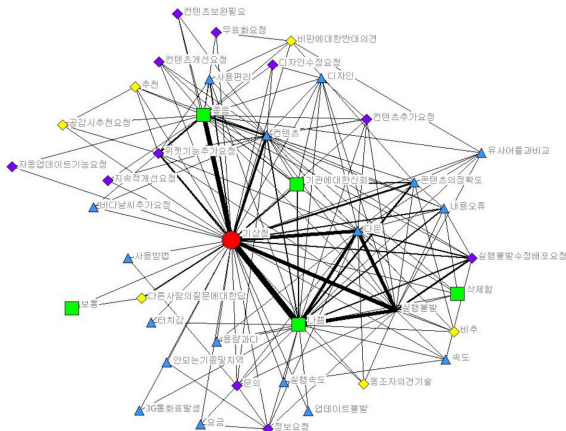
- 언어 네트워크 분석으로 신뢰-비신뢰 요인 한눈에 본다



드라마 대본을 이용한 네트워크

참고자료

- 공공앱의 사용자 리뷰에 대한 분석 : 언어네트워크분석을 중심으로



〈그림 2〉 ‘기상청날씨’ 고객리뷰 언어네트워크분석 결과: 네트워크 구조도

드라마 대본을 이용한 네트워크

한국어 대사 분석 방법

자바, 미안하다! 파이썬 한국어 NLP

 Tweet 0

 +1 0

 Like 0

 데이터 / 바이오 파이썬

 2014-08-30 (토요일) 14:30 - 15:15

 한국어

 계단강의실

 박은정 / lucypark

설명

NLTK 덕에 파이썬으로 자연어처리를 하는 것이 편리해졌다. 단, 한국어만 분석하려하지 않는다면, 파이썬으로 한국어를 분석할 수는 없을까? 국문, 영문, 중문 등 다양한 문자가 섞여 있는 문서는 어떻게 분석을 할 수 있을까?

이 발표에서는 자연어처리의 기초적인 개념을 다룬 후, NLTK 등의 자연어처리 라이브러리와 한국어 분석을

▶ 🔍 ↺

프로그래밍 언어 네트워크

Document :

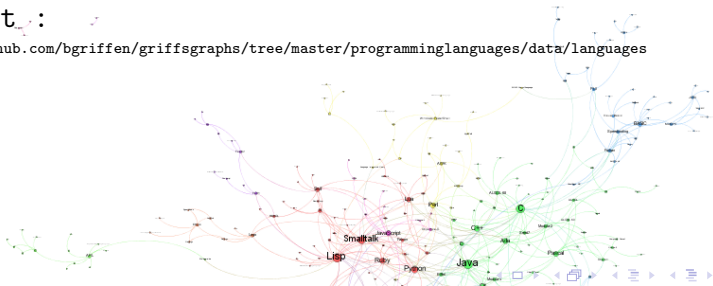
<http://brendangriffin.com/gow-programming-languages/>

Python :

<https://github.com/bgriffen/griffsgraphs/blob/master/programminglanguages/proglanguages.py>

Dataset :

<https://github.com/bgriffen/griffsgraphs/tree/master/programminglanguages/data/languages>



프로그래밍 언어 네트워크

Python (programming language)

From Wikipedia, the free encyclopedia

Python is a widely used [general-purpose, high-level programming language](#).^{[15][16][17]} Its design philosophy emphasizes code [readability](#), and its syntax allows programmers to express concepts in fewer [lines of code](#) than would be possible in languages such as [C](#).^{[18][19]} The language provides constructs intended to enable clear programs on both a small and large scale.^[20]

Python supports multiple [programming paradigms](#), including [object-oriented](#), [imperative](#) and [functional programming](#) or [procedural](#) styles. It features a [dynamic type system](#) and automatic [memory management](#) and has a large and comprehensive [standard library](#).^[21]

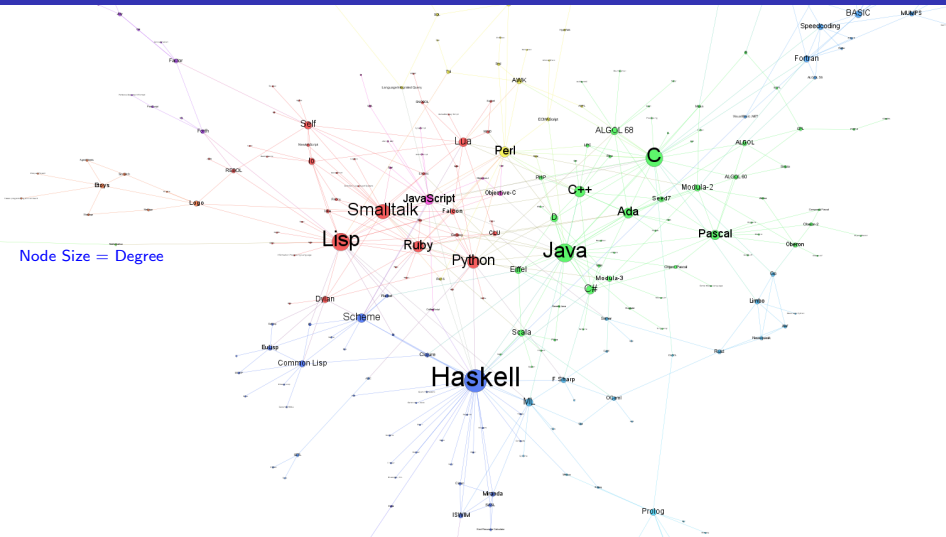
Using third-party tools, such as [Py2exe](#) or [Pyinstaller](#),^[22] Python code can be packaged into standalone executable programs. Python interpreters are available for many operating systems.

[CPython](#), the [reference implementation](#) of Python, is [free and open source software](#) and has a community-based development model, as do nearly all of its alternative implementations. CPython is managed by the non-profit [Python Software Foundation](#).

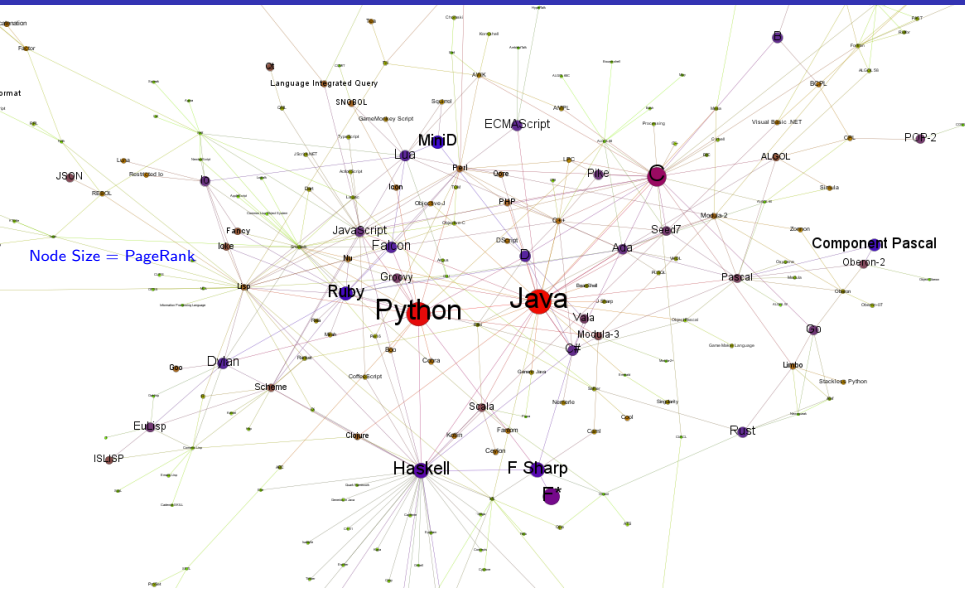
Contents [hide]	Influenced by
1 History	ABC , ^[3] ALGOL 68 , ^[4]
2 Features and philosophy	C , ^[5] C++ , ^[6] Dylan , ^[7]
3 Syntax and semantics	Haskell , ^[8] Icon , ^[9]
3.1 Indentation	Java , ^[10] Lisp , ^[11]
3.2 Statements and code blocks	Modula-3 , ^[6] Perl
3.3 Expressions	
3.4 Methods	Influenced
3.5 Typing	Boo , Cobra , D , F# ,
3.6 Mathematics	Falcon , Go , Groovy ,
4 Libraries	JavaScript ^[citation needed] ,
5 Development environment	Julia , ^[12] Ruby , ^[13]
6 Implementations	Swift ^[14]
7 Development	
8 Naming	

 python ™	
Paradigm(s)	Multi-paradigm: object-oriented, imperative, functional, procedural, reflective
Designed by	Guido van Rossum
Developer	Python Software Foundation
Appeared in	1991; 23 years ago
Stable release	3.4.1 / 18 May 2014 ^[1] 2.7.8 / 31 May 2014 ^[2]
Typing discipline	duck, dynamic, strong
Major implementations	CPython, PyPy, IronPython, Jython
Dialects	Cython, RPython, Stackless Python
Influenced by	ABC , ^[3] ALGOL 68 , ^[4] C , ^[5] C++ , ^[6] Dylan , ^[7] Haskell , ^[8] Icon , ^[9] Java , ^[10] Lisp , ^[11] Modula-3 , ^[6] Perl
Influenced	Boo , Cobra , D , F# , Falcon , Go , Groovy , JavaScript ^[citation needed] , Julia , ^[12] Ruby , ^[13] Swift ^[14]

프로그래밍 언어 네트워크



프로그래밍 언어 네트워크



프로그래밍 언어 네트워크

웹 크롤링 방법

30분만에 따라하는 동시성 웹 스크래퍼

 Tweet { 0 }  +1 { 0 }  Like { 0 }

 파이썬 응용

 2014-08-30 (토요일) 13:45 - 14:30

 한국어

 수정당2

 강철 / kang

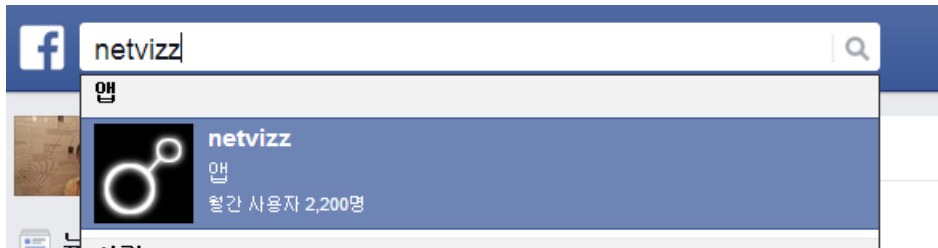
설명

빅데이터, 데이터마이닝, 공공데이터, 오픈데이터 - 그 어느때보다 데이터 분석 및 활용이 중요해진 이 시기에 웹 상의 수많은 공개된 자료를 직접 수집할 수 있는 웹 스크래핑/크롤링 기술은 데이터 수집 및 활용 능력에 큰 도움이 됩니다.

이 강의에서는 크롤링 프레임워크를 사용하지 않고 HTTP, DOM, concurrency를 담당하는 기본적인 라이브러리를 사용하며,   

페이스북 친구 네트워크

Netvizz



페이스북 친구 네트워크

Netvizz

netvizz v1.01

Netvizz is a tool that extracts data from different sections of the Facebook platform (personal profile, groups, pages) for research purposes. The data can be easily analyzed in standard software.

For questions, please consult the [FAQ](#) and [privacy](#) sections. Non-commercial use only.

New: there is now an [overview video](#) that introduces the different modules and other things to consider.

Big networks may take some time to process. **Be patient and try not to reload!**

Developing and hosting netvizz costs time and money. If the tool is useful for you, please consider to

[Donate](#)

The following modules are currently available:

[personal network](#) - extracts your friends and the friendship connections between them

[personal like network](#) - creates a network that combines your friends and the objects they liked in a bipartite graph

[group data](#) - creates networks and tabular files for both friendships and interactions in groups

[page like network](#) - creates a network of pages connected through the likes between them

[page data](#) - creates networks and tabular files for user activity around posts on pages

페이스북 친구 네트워크

Netvizz

netvizz v1.01

your personal friend network:

Creates a network file (gdf format) with all the friendship connections in your personal network, as well as a stat file (tsv format).

Select user data to include in the file (sex, interface language, and account age ranking are standard):

☐ friends' like and post count (public and visible to logged user, first 1000 only), includes counts for received likes and comments on posts, adds an additional ± 6 seconds of waiting time per friend

start

file fields (network file - gdf format - nodes are users):

sex: user specified sex

locale: user selected interface language

agerank: accounts ranked by creation date where 1 is youngest

like_count: number of user likes

post_count: number of user posts

post_like_count: number of likes on user's posts

post_comment_count: number of comments on user's posts

post_engagement_count: $\text{post_comment_count} + \text{post_like_count}$

file fields (stat file - tsv format - rows are users):

sex: user specified sex

locale: user selected interface language

agerank: accounts ranked by creation date where 1 is youngest

Netvizz

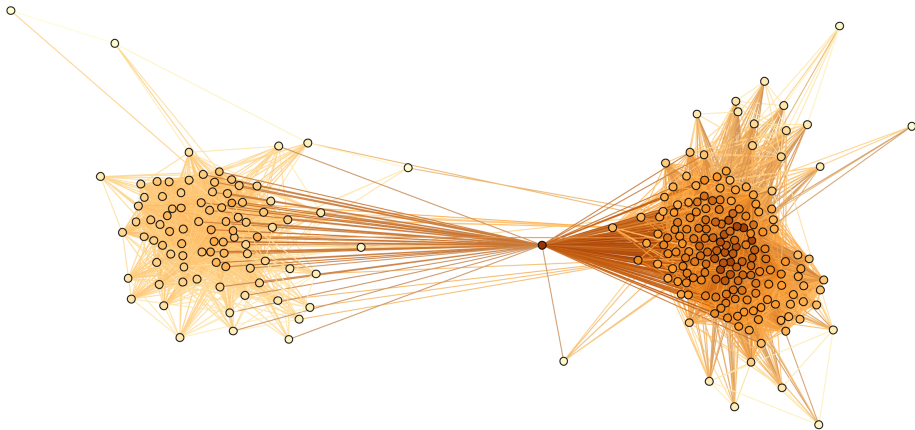
Your **gdf file** (right click, save as...).

Your **tab file** (right click, save as...).

Attention: some browsers add a .txt extension to the files, which must be removed after saving. When in doubt, use Firefox.

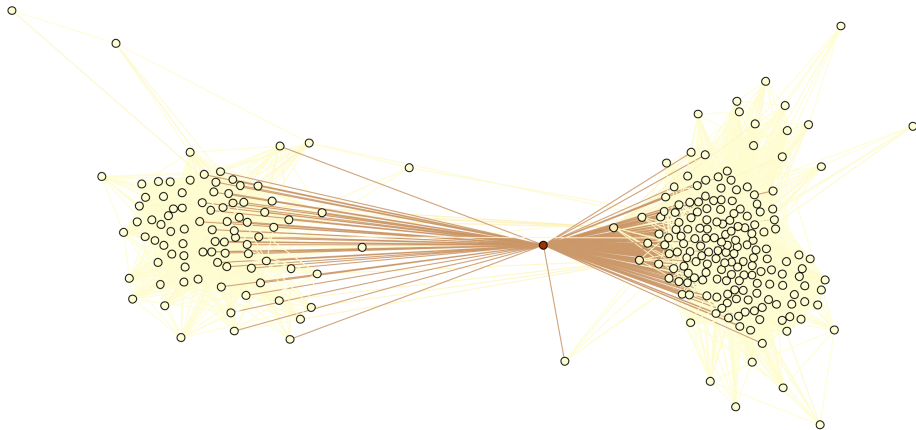
페이스북 친구 네트워크

페이스북 네트워크 with Degree Centrality

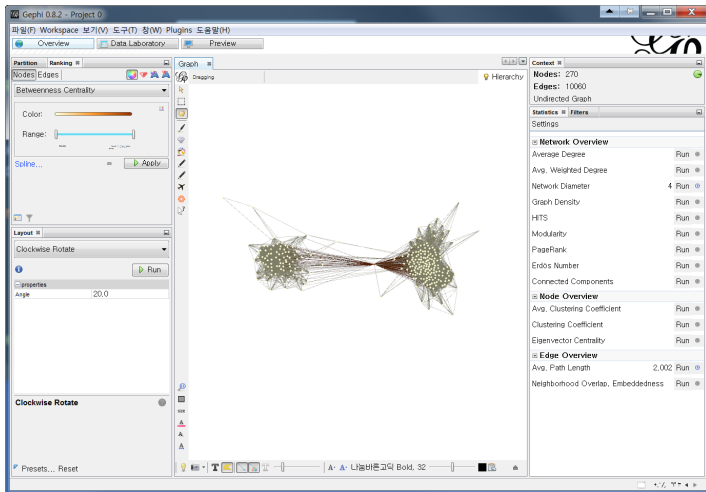


페이스북 친구 네트워크

페이스북 네트워크 with Betweenness Centrality



Gephi로 네트워크 그리기



- 파일 내보내기
- 파일 불러오기
- 노드의 한글 표현 방법
- 네트워크의 링크 예측

파일 내보내기

```
nx.write_graphml(G, './graphfile.graphml')  
nx.write_gexf(G, '파일 경로')  
nx.write_gml  
nx.write_yaml  
nx.write_pajek  
nx.write_shp  
nx.dump # for JSON
```

파일 불러오기

```
nx.read_graphml('./graphfile.graphml', unicode)
nx.read_gexf(G, '파일 경로')
nx.read_gml
nx.read_yaml
nx.read_pajek
nx.read_shp
nx.load # for JSON
```

노드의 한글 표현 방법

방법1. nx_pylab.py의 수정

/usr/lib/pymodules/python2.7/networkx/drawing/nx_pylab.py

C:\Anaconda32\Lib\site-packages\networkx\drawing\nx_pylab.py

수정 방법 : <https://gist.github.com/koorukuroo/3aae9a3e900e868840ea>

수정된 코드 : <https://gist.github.com/koorukuroo/5efb58a86c5396f13650>

사용방법

```
import matplotlib.font_manager as fm
```

```
fp1 = fm.FontProperties(fname="./NotoSansKR-Regular.otf")
```

```
# 무료 폰트 https://www.google.com/get/noto/pkgs/NotoSansKorean-windows.zip
```

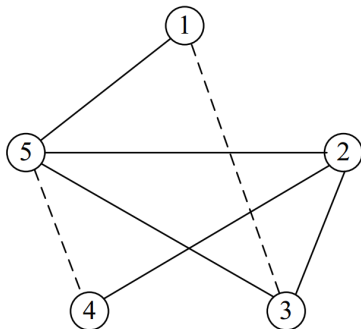
```
nx.set_fontproperties(fp1)
```

```
G = nx.Graph()
```


네트워크의 링크 예측

대구과학고등학교 2014 R&E

연구자 : 김요한, 김현, 유주형, 유준승, 한승원



Lü, Linyuan, and Tao Zhou. "Link prediction in complex networks: A survey." *Physica A*(2011): 1150-1170.

네트워크의 링크 예측

Link Prediction

Link prediction algorithms.

<code>resource_allocation_index</code> (G [, ebunch])	Compute the resource allocation index of all node pairs in ebunch.
<code>jaccard_coefficient</code> (G [, ebunch])	Compute the Jaccard coefficient of all node pairs in ebunch.
<code>adamic_adar_index</code> (G [, ebunch])	Compute the Adamic-Adar index of all node pairs in ebunch.
<code>preferential_attachment</code> (G [, ebunch])	Compute the preferential attachment score of all node pairs in ebunch.
<code>cn_soundarajan_hopcroft</code> (G [, ebunch, community])	Count the number of common neighbors of all node pairs in ebunch using community information.
<code>ra_index_soundarajan_hopcroft</code> (G [, ebunch, ...])	Compute the resource allocation index of all node pairs in ebunch using community information.
<code>within_inter_cluster</code> (G [, ebunch, delta, ...])	Compute the ratio of within- and inter-cluster common neighbors of all node pairs in ebunch.

네트워크의 링크 예측

NetworkX 버전 업그레이드

```
pip install --upgrade networkx==1.9
```

```
pip install --upgrade git+https://github.com/networkx/networkx.git#egg=networkx
```

<http://networkx.github.io/documentation/latest/install.html>

To be continue(?)

Appendix1. KLDP 개발자 네트워크

공개소프트웨어 커뮤니티의 네트워크 위상구조 연구:

KLDP 사이트를 중심으로

조원영¹⁾, 장대철²⁾, 안병훈³⁾

한국과학기술원 테크노경영대학원 경영공학과

초 록

소프트웨어의 소스코드가 공개되어 있어 누구나 이를 수정하거나 재배포할 수 있는 공개소프트웨어의 개발과 사용이 점차 증가하고 있는데, 인터넷 커뮤니티 활동을 통해 이루어지는 개발자간 협업 및 공개소프트웨어 개발 과정은 복잡계 시스템의 특성을 지니고 있다. 본 연구는 국내의 대표적인 공개소프트웨어 개발 커뮤니티 중 하나인 KLDP(Korea Linux Documentation Project) 사이트를 분석하여 네트워크 위상구조적 특성을 파악하고자 했다. 이를 통해 KLDP 커뮤니티 네트워크의 밀도는 낮지만 결속계수가 높음을 확인했으며, 결점들 간의 연결 수 분포가 멱함수 법칙을 따른다는 것을 밝혔다. 또한 네트워크의 구성에 중요한 역할을 하는 결점들 간에 유유상종 현상이 나타남을 관찰했다. 한편 네트워크 위상구조적 특성과 결점 자체가 가지고 있는 특성과의 관계를 분석함으로써 동일 군집에 속하는 프로젝트 간에 공통적인 속성이 있음을 밝혔다.

Key Words : 공개소프트웨어, 인터넷 커뮤니티, 네트워크 위상구조, 복잡계 시스템

Appendix1. KLDP 개발자 네트워크

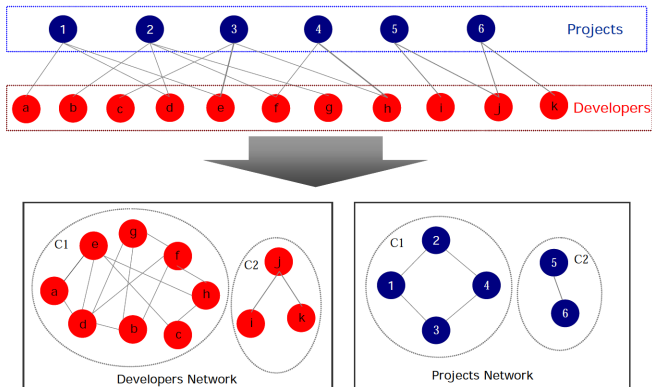


Fig. 1 Formations of Developers/Projects Network in Affiliation Network

Appendix1. KLDP 개발자 네트워크

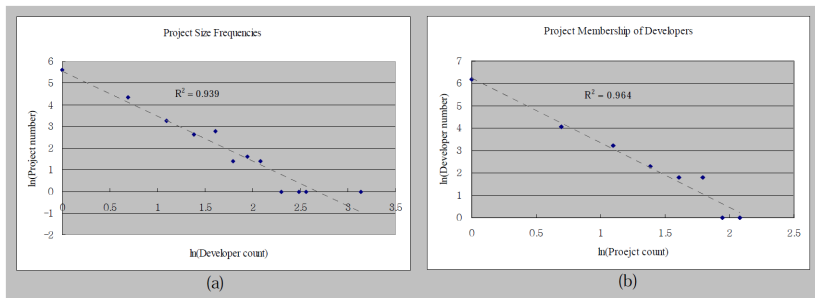


Fig. 3 Log-log Transformation of (a) Project Size Frequencies and (b) Project Membership Developers

<http://dspace.kaist.ac.kr/handle/10203/21356>

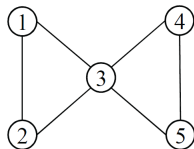
Appendix2. 점과 선의 체인지



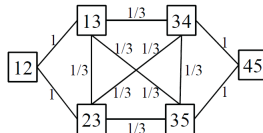
SBS 시크릿가든 공식 홈페이지



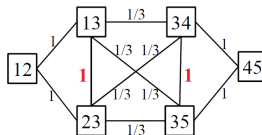
Appendix2. 점과 선의 체인지



(a) Original graph.



(b) Line graph.



(c) Link-space graph.

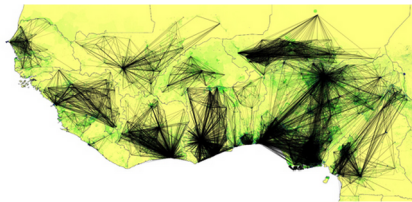
Lim, Sungsu, et al. "LinkSCAN*: Overlapping community detection using the link-space transformation." Data Engineering (ICDE), 2014 IEEE 30th International Conference on. IEEE, 2014.

Appendix3. Predict Ebola's Spread

Cell-Phone Data Might Help Predict Ebola's Spread

Mobility data from an African mobile-phone carrier could help researchers recommend where to focus health-care efforts.

By David Talbot on August 22, 2014



On the move: This model of West African regional transportation patterns was built using, among other sources, mobile-phone data for Senegal, released by the mobile carrier Orange.



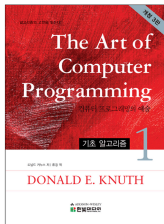
WHY IT MATTERS

<http://www.technologyreview.com/news/530296/cell-phone-data-might-help-predict-ebolas-spread/>



이번 TALK의 목적 (다시 확인)

1 LaTeX



```
\begin{frame}
\frametitle{이번 TALK의 목적}
\begin{enumerate}
\item LaTeX\\
\includegraphics[scale=0.2]{donald.png}
\includegraphics[scale=0.5]{latex.png}
\item 점과 선의 자유로움
\end{enumerate}
\end{frame}
```

2 점과 선의 자유로움

3 파이썬이라서 자유로운 점



The End