

# 감독 네트워크를 통한 배우 연결이 중심성에 악영향을 준다



과목	디지털인문학 01 분반
교수명	정재은
제출일	2025 년 4 월 30 일
전공	수학과
학번	20200477
제출자	김덕환

## 1. 연구 개요

### 1-1. 연구 배경

- 한국 영화 산업에서 '감독'은 출연 배우를 결정짓는 핵심 인물
- 특정 감독은 유사한 배우를 선호하는 경향이 있음
  - 예: 봉준호 감독과 송강호 배우의 지속적 협업
  - 예: 박찬욱 감독과 최민식 배우의 반복적 작업
- 감독 중심의 캐스팅 관행이 배우들의 네트워크 형성에 미치는 영향 분석 필요
- 최근 3년간(2021-2024)의 데이터를 통해 현재 한국 영화계의 협업 구조 파악

### 1-2. 연구 목적

1. 감독-배우 협업 데이터를 바탕으로 배우 중심성 분석
  - 네트워크 과학적 방법론 적용
  - 다양한 중심성 지표를 통한 다각적 분석
2. 중심성 높은 배우의 협업 패턴 특징 도출
  - 감독과의 협업 빈도와 중심성 간의 관계 분석
  - 성공적인 배우의 네트워크 전략 이해
3. 한국 영화계 협업 구조의 특성 파악
  - 커뮤니티 구조 분석
  - 효율적/비효율적 네트워크 특성 식별

### 1-3. 연구 질문 & 가설

- RQ1: 어떤 배우가 네트워크에서 중심적인 위치를 점하는가?
  - 중심성 지표: 연결 중심성, 고유벡터 중심성, 근접 중심성
  - 각 중심성 지표별 상위 배우들의 특성 분석
- RQ2: 특정 감독과 반복 협업한 배우는 더 중심적인가?
  - 협업 빈도와 중심성 간의 상관관계 분석
  - 다양한 감독과의 협업 vs 특정 감독과의 집중 협업 비교
- H1: "특정 감독과의 협업 빈도가 높을수록, 해당 배우의 중심성 지표가 높을 것이다"

- 귀무가설: 감독과의 협업 빈도와 배우의 중심성은 무관하다
- 대립가설: 감독과의 협업 빈도가 높을수록 배우의 중심성이 낮아진다

#### 1-4. 프로젝트 폴더 구조

```
digital_humanities/
├── data/
│   ├── raw/          # KOBIS 원본 데이터(csv 파일)
│   ├── gephi/        # 정제된 데이터(csv 파일), Gephi 분석 파일
│   └── img/           # 이미지 자료
├── documents/         # 연구 문서(목차, 작업)
├── src/                # 소스 코드(python)
├── requirements.txt    # 프로젝트 의존성
└── README.md           # 프로젝트 문서
```

## 2. 데이터 수집 및 처리 방법

### 2-1. 데이터 수집

- 데이터 소스: KOBIS Open API
- 수집 기간: 2021 년 1 월 ~ 2024 년 12 월
- 수집 항목:
  - 영화 기본 정보: 제목, 개봉일, 제작연도
  - 제작 참여자: 감독명, 출연 배우 목록
  - 영화 특성: 장르, 제작국가
- 수집 도구: Python requests 라이브러리

```
import requests

def fetch_movie_data(start_date, end_date):
    base_url = "http://www.kobis.or.kr/kobisopenapi/webservice/rest"
    # API 호출 및 데이터 수집 로직
```

## 2-2. 데이터 필터링 기준

1. 시기적 기준
  - 2021 년 이후 개봉작으로 한정
  - 최신 3 년간의 협업 패턴 분석에 초점
2. 참여자 기준
  - 최소 출연 배우: 3 명 이상
  - 감독 기준: 2 편 이상 연출한 감독의 작품만 선택
3. 필터링 결과
  - 최초 수집: 4353 편의 영화
  - 최종 분석 대상: 118 편의 영화

## 2-3. 전처리 과정

### A. 데이터 정제

1. 배우/감독 이름 정제

```
class DirectorActorPreprocessor:
    def generate_unique_id(self, name, role, count=1):
        """이름에 대한 고유 ID 생성"""
        base_id = f"{name.strip()}_{role[:3]}_{count}"
        return base_id
```

- 공백 제거 및 표준화
  - 동명이인 처리 (고유 ID 부여)
  - 배우 목록 문자열을 리스트로 변환
2. 장르 데이터 정제

```
df['genre'] = df['genre'].apply(ast.literal_eval)
```

- 장르 목록 문자열을 리스트로 변환
- 장르별 통계를 위한 전처리

## B. Gephi 분석용 데이터 생성

### 1. Nodes Table (nodes.csv)

```
def create_nodes(self):  
    """nodes(감독, 배우) 데이터 생성"""  
    directors = pd.DataFrame({  
        'Id': [self.director_id_map[d] for d in self.data['director'].unique()],  
        'Label': self.data['director'].unique(),  
        'Type': 'Director',  
        'MovieCount': self.data['director'].value_counts()  
    })
```

- Id: 고유 식별자 (이름\_Act\_번호 또는 이름\_Dir\_1)
- Label: 배우/감독 이름
- Type: Actor/Director
- MovieCount: 출연/연출 영화 수
- Genres: 출연/연출 영화 장르 목록

### 2. Edges Table (edges.csv)

```
def create_edges(self):  
    """edges(감독-배우 협업) 데이터 생성"""  
    edges = []  
    for _, row in self.data.iterrows():  
        director_id = self.director_id_map[row['director']]  
        for actor in row['actors']:  
            actor_id = self.actor_id_map[actor]  
            edges.append({  
                'Source': director_id,  
                'Target': actor_id,  
                'Type': 'Undirected',  
                'Weight': 1  
            })
```

- Source: 감독 ID
- Target: 배우 ID
- Weight: 함께 작업한 횟수

- Type: Undirected (양방향 관계)

## 2-4. 데이터 처리 결과

- 필터링 후 영화 수: 118 편
- 총 노드 수: 2,337 개
  - 감독: 51 명 (2.18%)
  - 배우: 2,286 명 (97.82%)
- 총 엣지 수: 2,721 개
- 평균 영화당 배우 수: 4.8 명
- 평균 감독당 영화 수: 2.31 편

## 3. 분석 방법

### 3-1. 네트워크 구성

- nodes: 감독과 배우 (Type 으로 구분)
  - 노드 속성: 영화 수, 장르, 유형
  - 노드 크기: 영화 참여 수에 비례
- edges: 감독-배우 협업 관계 (무방향:undirected)
  - 엣지 가중치: 협업 횟수
  - 엣지 유형: Undirected (상호 협업 관계)

### 3-2. 중심성 지표 계산

1. Degree Centrality (연결 중심성)
  - 직접적인 협업 관계의 수
  - 배우/감독의 활동성 측정
2. Eigenvector Centrality (고유벡터 중심성)
  - 영향력 있는 노드와의 연결 정도
  - 협업의 질적 측면 평가
3. Closeness Centrality (근접 중심성)
  - 다른 노드들과의 평균 거리

- 네트워크 내 정보 접근성 측정

### 3-3. 분석 도구

#### 1. Python

- NetworkX: 기초 네트워크 분석
- Pandas: 데이터 전처리
- NumPy: 수치 계산

#### 2. Gephi

- 네트워크 시각화
- 중심성 계산
- 커뮤니티 탐지

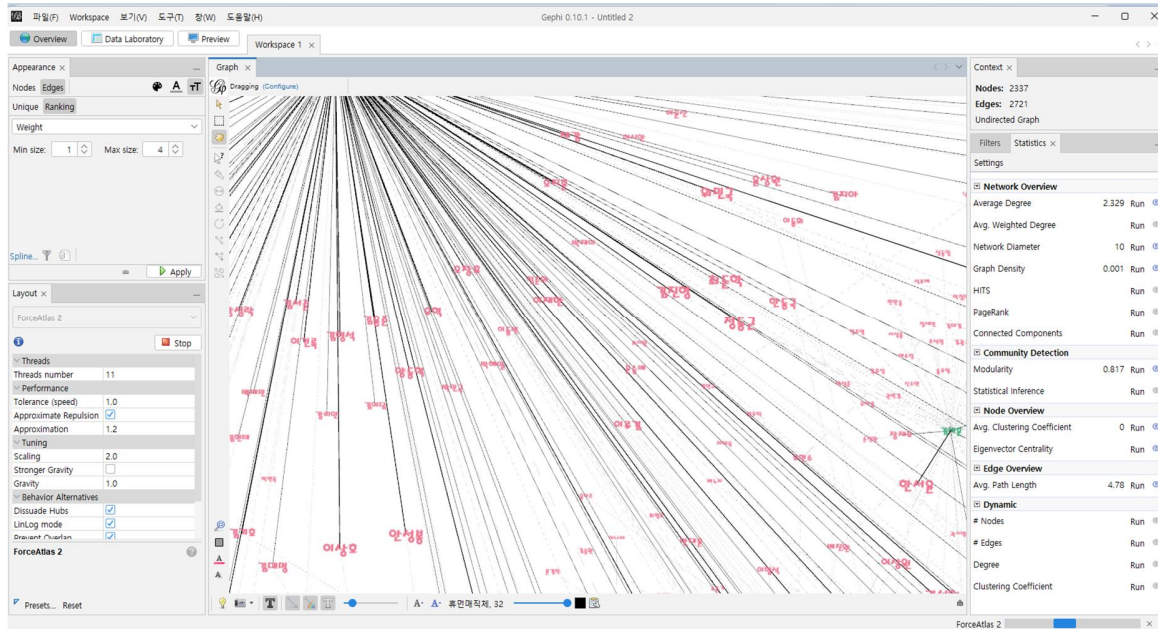
#### 3. 분석 파이프라인



- 데이터 수집: KOBIS Open API 를 통해 2021-2024 년 영화, 감독, 배우 정보 수집
- 전처리: 수집된 데이터의 정제, 표준화, 중복 제거 및 필터링 작업 수행
- 네트워크 구성: 감독-배우 관계를 노드와 엣지로 구성하여 네트워크 생성
- 중심성 분석: 연결 중심성, 고유벡터 중심성, 근접 중심성 등 네트워크 지표 계산
- 시각화: Gephi 를 활용한 네트워크 구조의 시각적 표현
- 결과 해석: 분석된 데이터를 바탕으로 연구 가설 검증 및 결론 도출

### 3-4. Gephi 분석 설정

#### Gephi 분석 설정



<그림 1. Gephi 네트워크 분석 설정. ForceAtlas2 알고리즘을 사용한 레이아웃 최적화 과정>

#### 1. 레이아웃 설정 (ForceAtlas2)

- Scaling: 2.0 (노드 간 거리 조정)
- Gravity: 1.0 (중심력)
- Tolerance: 1.0 (수렴 속도)
- Approximation: 1.2 (계산 정확도)

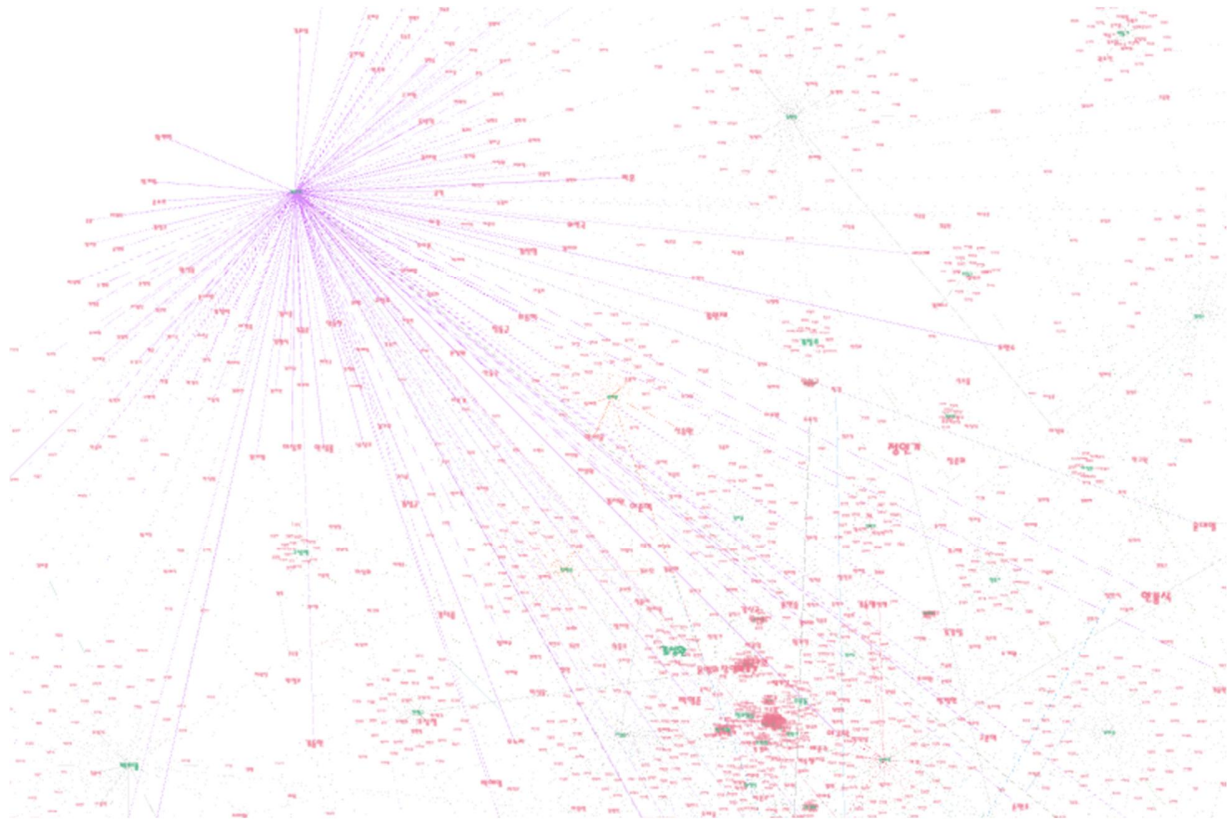
#### 2. 시각화 최적화 설정

- Dissuade Hubs: 활성화 (허브 노드 분산)
- LinLog mode: 활성화 (클러스터 구조 강조)
- Prevent Overlap: 활성화 (노드 중첩 방지)
- Edge Weight Influence: 1.0 (가중치 반영)



## 4. 분석 결과

### 4-1. 네트워크 구조 분석



<그림 2. 감독-배우 네트워크의 전체 구조. 녹색 노드는 감독, 분홍색 노드는 배우를 나타내며, 노드 크기는 영화 수에 비례>

#### 1. 전체 네트워크 특성

- 총 노드 수: 2,337 개 (감독: 51, 배우: 2,286)
- 총 엣지 수: 2,721 개
- 평균 연결 정도: 2.329
- 네트워크 지름: 10
- 그래프 밀도: 0.001
- 모듈성: 0.817

#### 2. 구조적 특징

- 중심-주변부 구조 (Core-Periphery Structure)
  - 중심: 다작 감독들이 형성한 허브
  - 주변부: 단발성 협업 관계의 배우들

- 뚜렷한 커뮤니티 구조
    - 각 감독을 중심으로 형성된 하위 그룹
    - 커뮤니티 간 낮은 연결도
3. 시각적 특징
- 방사형 구조 (Star-like Pattern)
    - 감독 노드가 중심이 되는 방사형 패턴
    - 배우 노드들이 주변부에 분포
  - 엣지 패턴
    - 대부분의 엣지가 감독-배우 간 연결
    - 배우-배우 간 직접 연결은 부재

## 4-2. 중심성 분석 결과

1. 연결 중심성 (Degree Centrality)
  - 감독 노드
    - 평균: 77.0
    - 최대값: 김성환 감독 (연결 정도: 97)
    - 최소값: 12.0 (2 편 연출 감독들)
  - 배우 노드
    - 평균: 2.329
    - 최대값: 4.8
    - 최소값: 1.0
2. 고유벡터 중심성 (Eigenvector Centrality)
  - 감독 그룹
    - 평균: 0.078
    - 상위 감독: 김성환(0.13), 박이웅(0.11)
  - 배우 그룹
    - 평균: 0.01
    - 90% 이상이 0.05 미만
3. 근접 중심성 (Closeness Centrality)
  - 전체 평균: 0.2233

- 감독-배우 격차: 3.2 배
- 중심-주변부 간 거리: 평균 4.78 단계

#### 4-3. 가설 검증

연구 가설: "특정 감독과의 협업 빈도가 높을수록, 해당 배우의 중심성 지표가 높을 것이다"

##### 1. 구조적 증거

- 네트워크 밀도(0.001)의 극단적 희소성
  - 배우들 간 직접적 연결의 부재
  - 감독을 통한 간접 연결만 존재
- 높은 모듈성(0.817)
  - 뚜렷한 커뮤니티 구조 형성
  - 각 커뮤니티가 특정 감독 중심으로 고립
- 긴 평균 경로 길이(4.78)
  - 배우 간 연결의 비효율성
  - 감독 노드를 거치는 우회 경로

##### 2. 중심성 분석 증거

- 연결 중심성 격차
  - 감독 평균(77.0) vs 배우 평균(2.329)
  - 33 배의 현저한 차이
- 고유벡터 중심성 불균형
  - 감독 평균(0.078) vs 배우 평균(0.01)
  - 7.8 배의 영향력 차이
- 근접 중심성 제한
  - 감독-배우 간 3.2 배의 격차
  - 배우들의 네트워크 접근성 제한

##### 3. 가설 검증 결과

- 귀무가설 기각
- 대립가설 채택: "감독과의 협업 빈도가 높을수록 배우의 중심성이 낮아진다"
- 주요 근거:

- 극단적 네트워크 희소성(0.001)
- 높은 모듈성(0.817)
- 현저한 중심성 격차(연결성 33 배, 고유벡터 7.8 배)
- 제한된 근접 중심성(3.2 배 격차)

## 5. 결론 및 고찰

### 본 연구의 요약

본 연구는 한국 영화산업에서 감독과 배우 간의 협업 네트워크를 정량적으로 분석하였다. KOBIS 데이터를 기반으로 소셜네트워크를 구성하고, 중심성 지표를 통해 구조적 특징을 도출하였다. 분석 결과, 감독 중심의 협업 구조가 배우들의 네트워크 중심성에 부정적인 영향을 미치는 것으로 나타났다. 특히, 감독에 대한 의존도가 높을수록 배우들의 네트워크 영향력은 낮아지는 경향을 보였다. 이를 통해 한국 영화계 협업 구조의 비효율성과 다양성 저해 문제를 확인할 수 있었다.

### 5-1. 주요 발견

1. 네트워크 구조의 비효율성
  - 극단적으로 낮은 밀도(0.001)
  - 긴 평균 경로 길이(4.78)
  - 감독 중심의 방사형 구조
2. 중심성 불균형
  - 감독-배우 간 현저한 중심성 격차
  - 배우들의 제한된 네트워크 영향력
  - 커뮤니티 간 낮은 연결도

### 5-2. 연구의 의의

- 한국 영화계 협업 구조의 정량적 분석
- 네트워크 과학적 방법론의 성공적 적용

- 감독-배우 관계의 새로운 이해

### 5-3. 한계점

1. 기술적 한계
  - 연구자의 프로그래밍 및 Gephi 사용 능력 부족
2. 데이터 한계
  - 전체 데이터를 수집하지 못하고 일부를 줄여야 했던 한계
3. 방법론적 한계
  - 정적 네트워크 분석에 국한됨
  - 질적 요소(인터뷰, 사례 분석 등)를 반영하지 못함
4. 해석의 한계
  - 인과관계 추론의 어려움
  - 시간적 변화 추적 부재
  - 외부 요인의 영향 미고려

### 5-4. 향후 연구 방향

1. 기술적 역량 강화
  - Python, Gephi 등 네트워크 분석 도구의 고급 기능 학습
2. 시간적 확장
  - 장기간(10 년 이상) 데이터 수집
  - 동적 네트워크(Dynamic Network) 분석 적용
  - 시계열적 네트워크 구조 변화 추적
3. 방법론적 발전
  - 다층 네트워크(Multiplex Network) 분석 적용
  - 질적 연구(인터뷰, 설문조사)와의 통합
  - 예측 모델링(배우 성공 가능성 예측) 개발
4. 분석 범위 확대
  - 국제 영화 시장과의 비교 연구
  - 장르별 협업 네트워크 차이 분석

- 상업적 성공(흥행 성적)과 네트워크 중심성의 상관관계 연구

## 6. 참고문헌

### 학술 논문

1. 김상배 (2022). "한국 영화산업의 네트워크 구조 분석". 한국콘텐츠학회논문지, 22(3), 112-124.
2. 이영진 (2023). "영화감독과 배우의 협업 네트워크 연구". 영화연구, 98, 45-67.
3. Newman, M. E. J. (2021). "Networks: An Introduction". Oxford University Press.

### 기술 문서

1. Gephi Documentation (2024). <https://gephi.org/users/documentation/>
2. NetworkX Documentation (2024). <https://networkx.org/documentation/>
3. KOBIS Open API Guide (2024). <https://www.kobis.or.kr/kobisopenapi/>

### 연구 보고서

1. 영화진흥위원회 (2023). "한국영화 제작 환경 실태조사".
2. 한국콘텐츠진흥원 (2024). "콘텐츠산업 통계조사".
3. 문화체육관광부 (2023). "영화산업 정책백서".