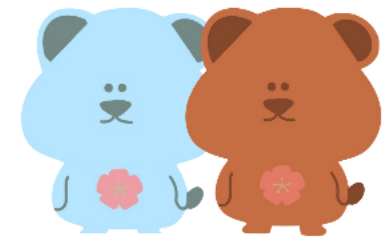




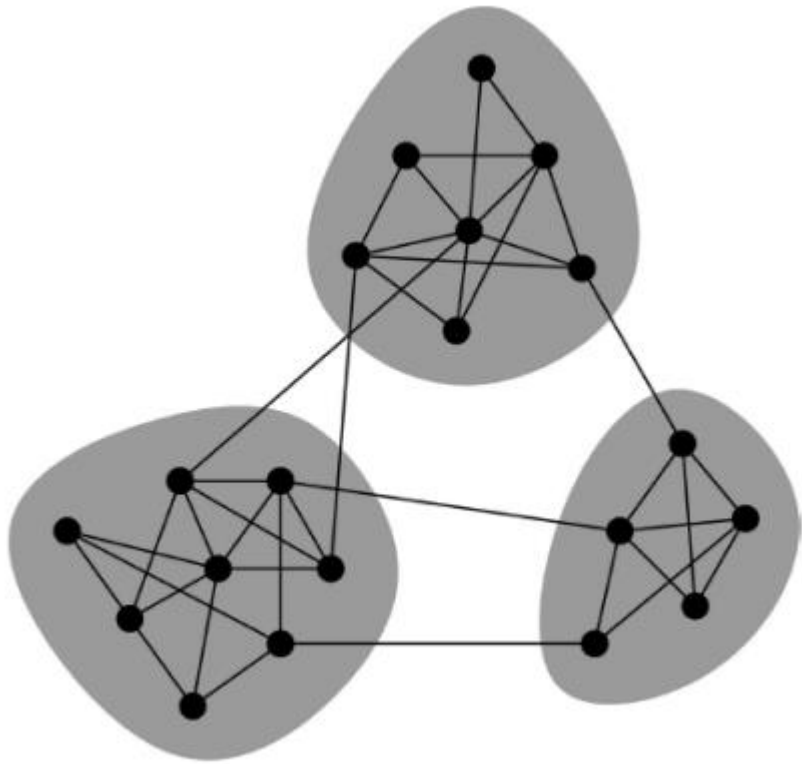
19주차 세션

DL팀 이다현

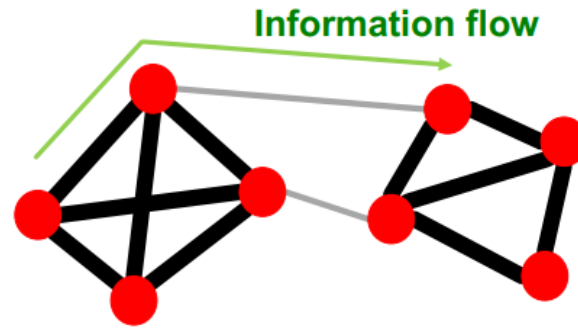
Community detection



Community Detection



Social network

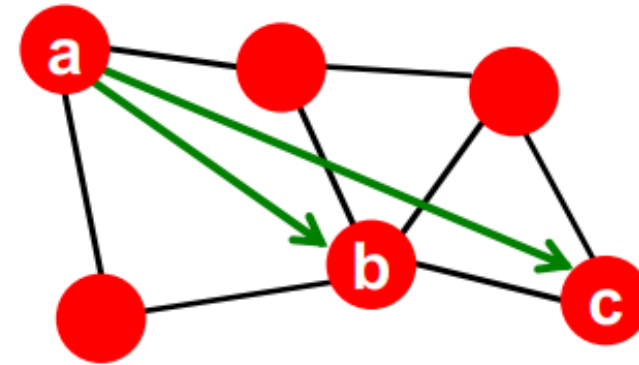
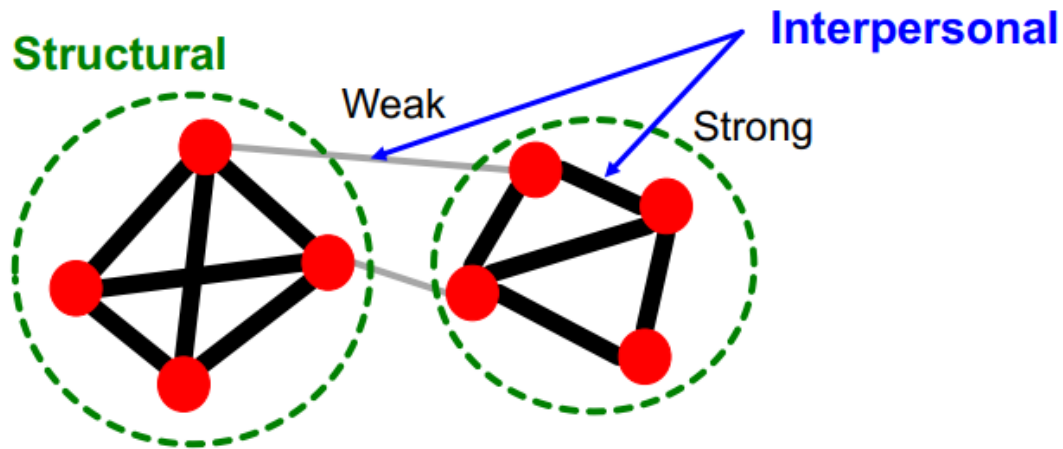
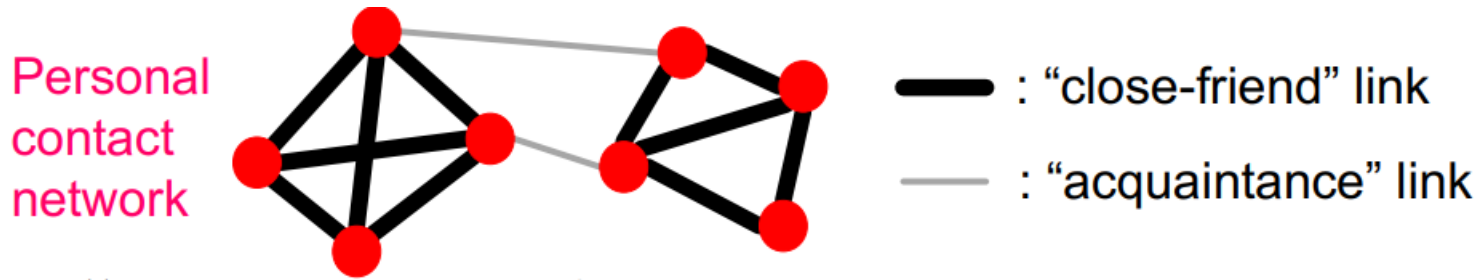


Information flow

- : person
- : “short” link
- : “long” link

- 네트워크를 “Community” 라는 관점에서 살펴볼 수 있다.
- Community : 서로 간 밀접하게 뭉쳐있는 노드의 집합

Granovetter 's theory



a-b 사이에 엣지가 발생할 확률이 더 높다.
공통된 친구 2명을 a-b 사이에 가지고 있기 때문이다.
공통된 친구를 가지고 있으면, 그들 또한 친구가 될 가능성이 높다.

- Edge 에 대한 2가지 관점 : Structure , Information
- Triadic Closure (high clustering coefficient) : 공통된 이웃노드를 많이 가지고 있을수록 두 노드 사이에 연결될 가능성이 높다 ➡ edge strength 에 대한 정의가 등장

Edge overlap vs strength

■ Edge overlap:

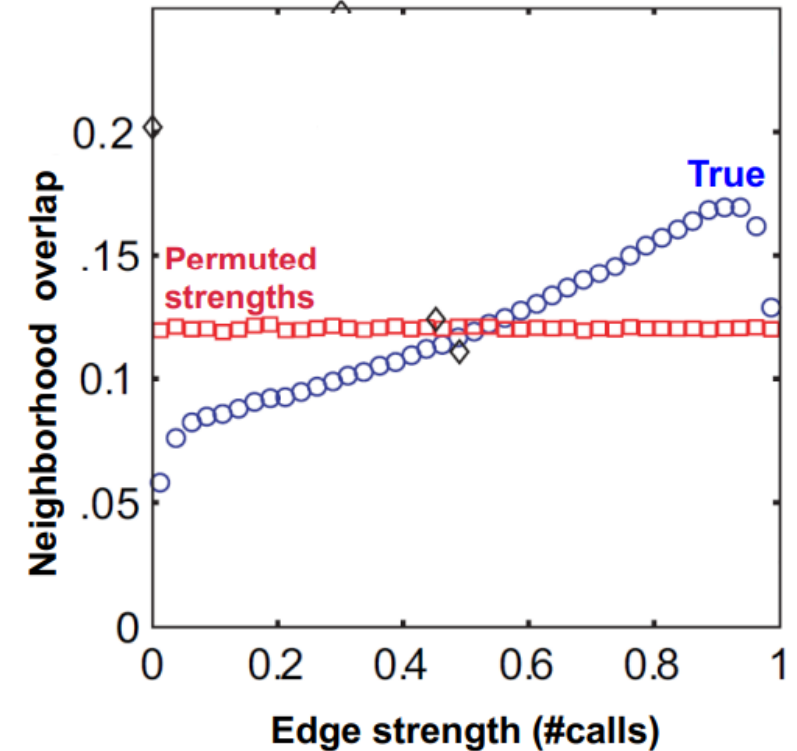
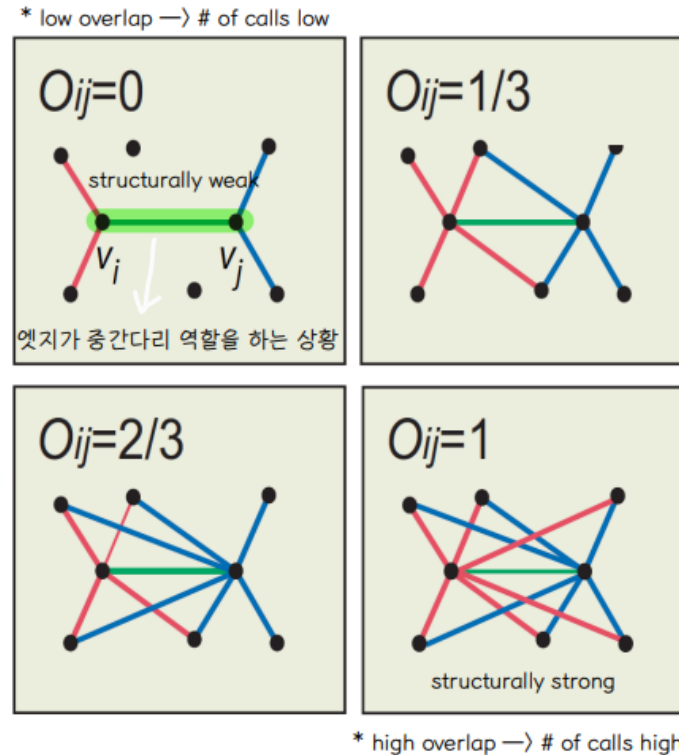
$$O_{ij} = \frac{|(N(i) \cap N(j)) - \{i, j\}|}{|(N(i) \cup N(j)) - \{i, j\}|}$$

겹쳐지는 이웃노드의 수

- $N(i)$... the set of neighbors of node i

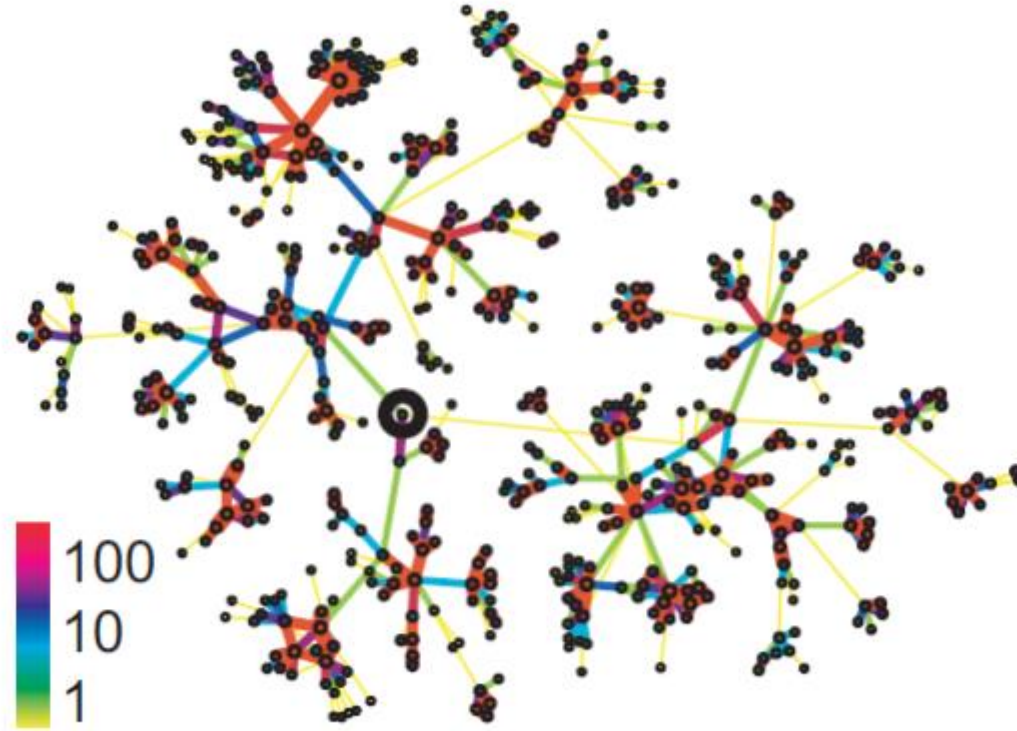
네트워크에서 얼마나 많은 지인을 공유하고 있는가에 대한 정보를 나타내는 수치

- **Note: Overlap = 0**
when an edge is a **local bridge**



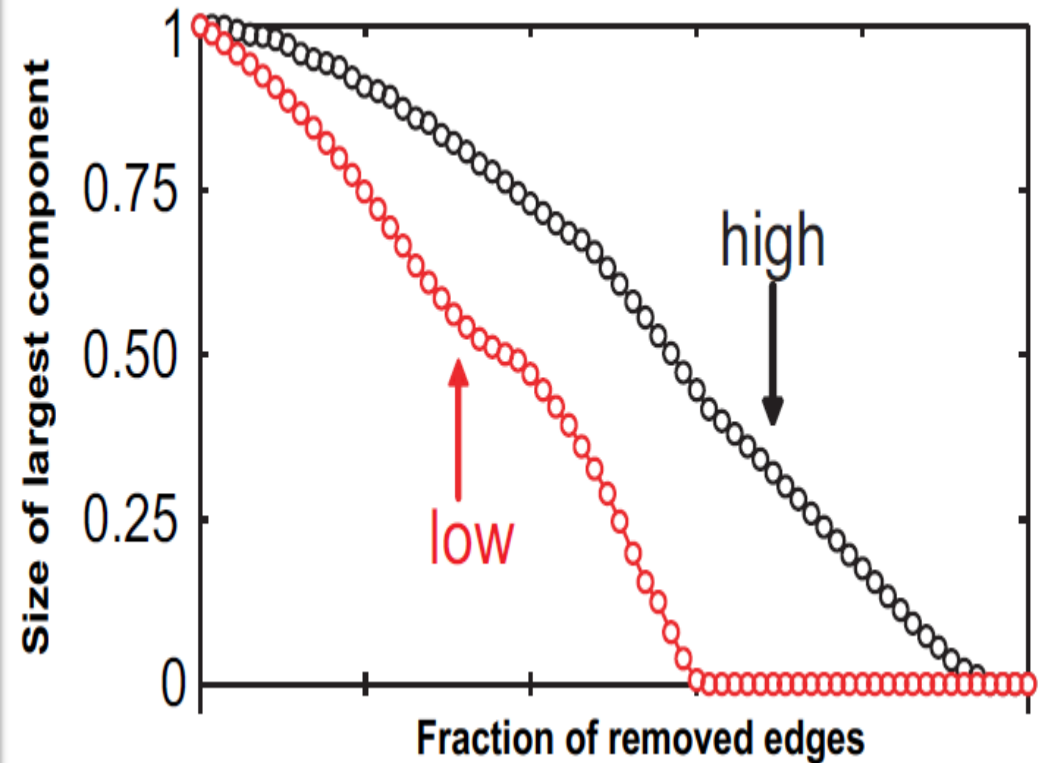
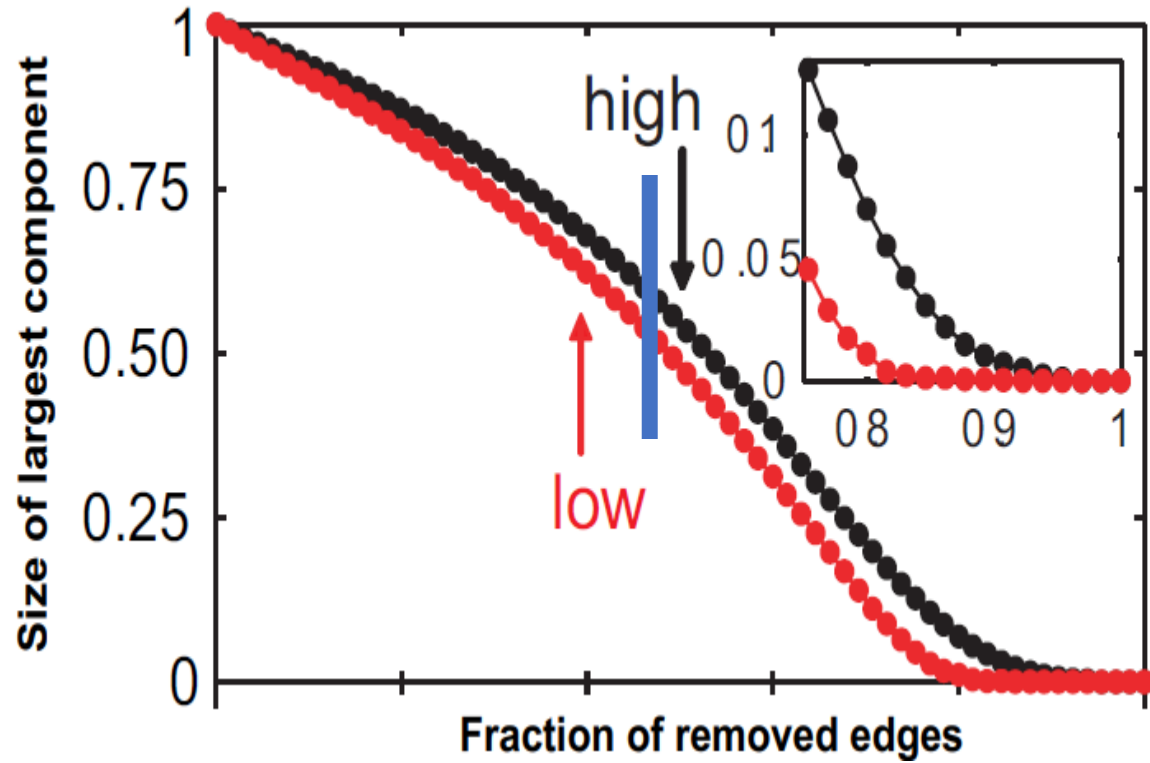
- Onnela 의 2007년도 실험 : EU 소속 국가 인구의 20%의 휴대폰 네트워크 데이터. 통화 횟수로 엣지 가중치를 정의
- 실제 데이터를 살펴본 결과, 통화하는 횟수가 많을수록 겹치는 지인 수도 높아짐을 발견 (permuted strength 는 엣지 가중치를 랜덤하게 설정한 basic model, 비교하기 위한 임계값이라 보면 됨) ➡ edge strength 가 존재하는구나!

Community Detection



- 실제 데이터를 시각화 했을 때, 유대관계가 높은 (strong edges) 엣지, 즉 통화 수가 빈번할수록 주변 지인이 겹쳐지는 경우가 많음 (= community 를 형성)

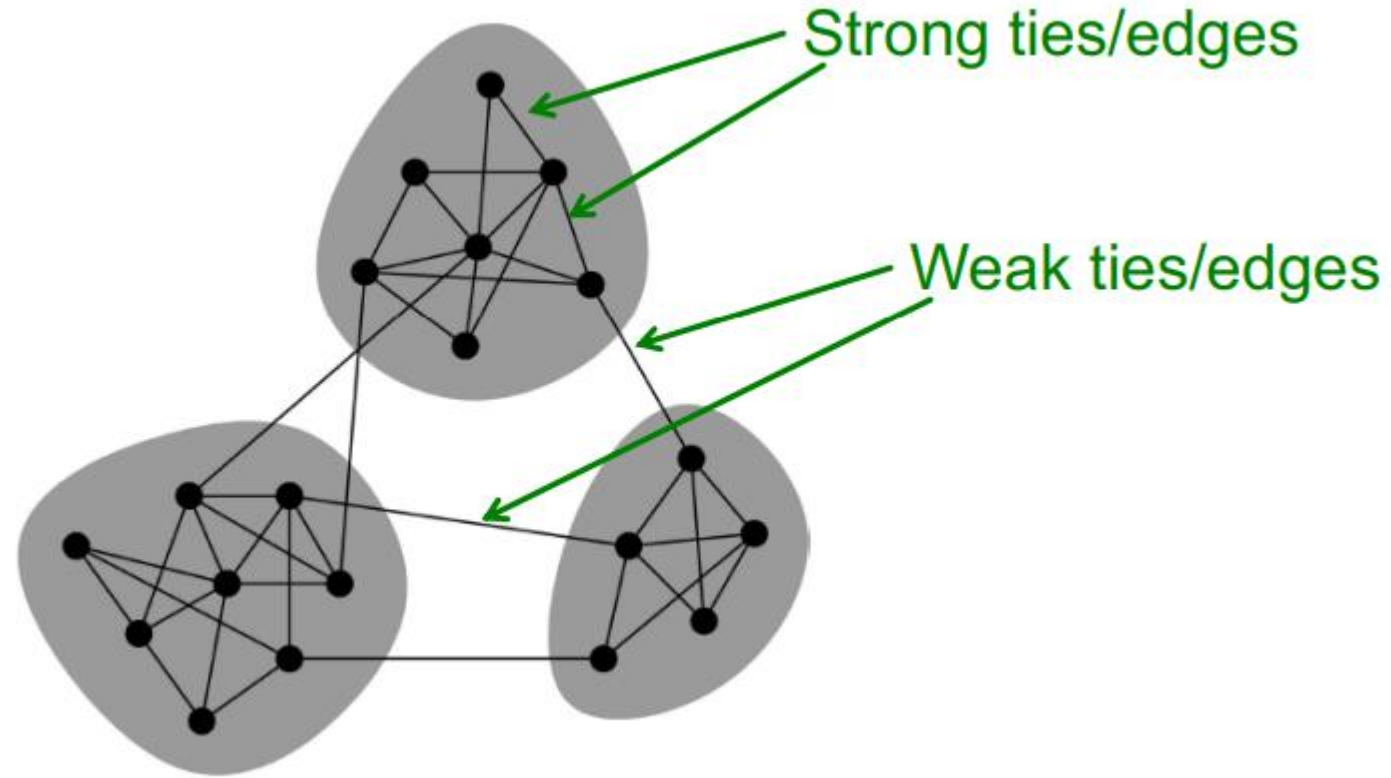
Edge removal by strength



- based on strength (# calls) : 통화 수에 기반해 엣지를 하나씩 지워가봄
- based on edge overlap : 겹치는 이웃노드 수에 기반해 엣지를 제거해나가 봄

☞ Low to high : 낮은 개수부터 지워나간게 더 빠른 속도로 네트워크가 disconnected 됨을 확인

Conclusion



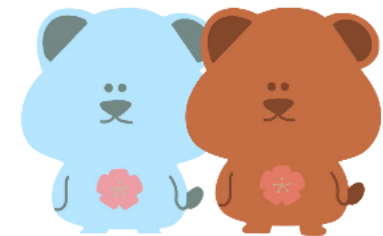
- 네트워크는

- > 내부적으로는 수많은 연결들 : strong edges

- > 외부적으로는 적은 연결들 : weak edges

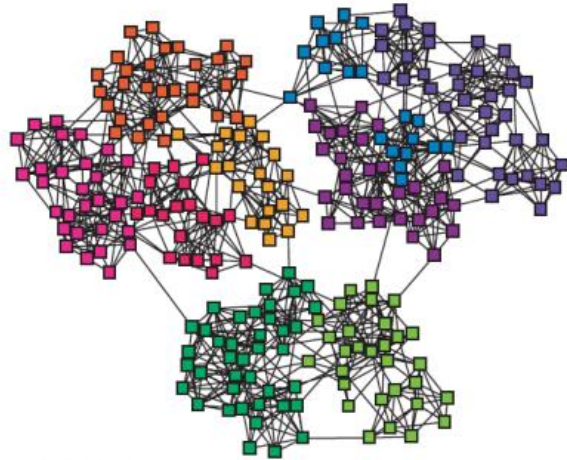
로 구성된 노드 집합으로 정의해볼 수 있다.

Network Communities



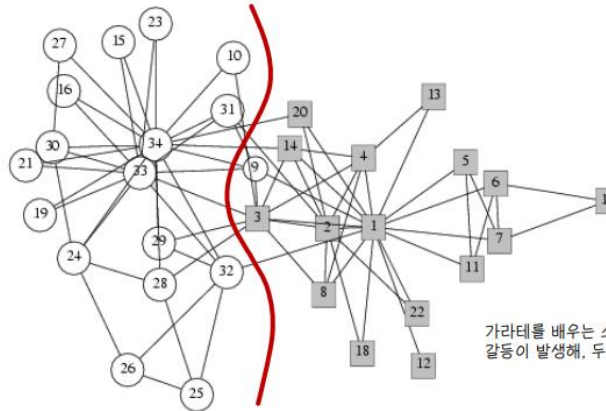
Network Communities

✓ community 를 찾을 수 있는 방법은 ...

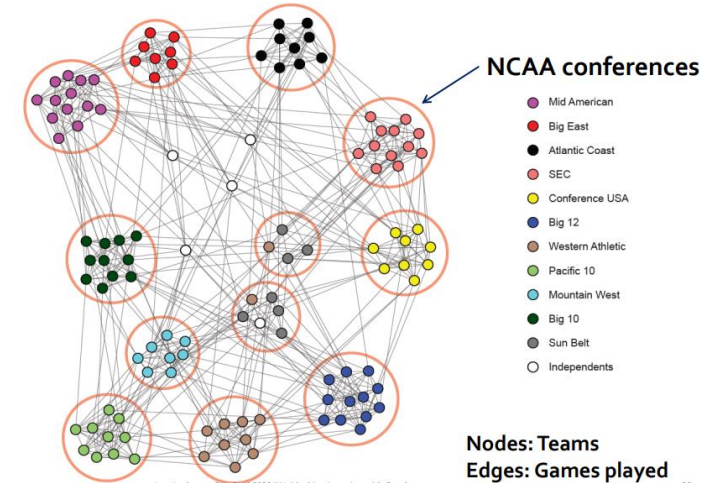
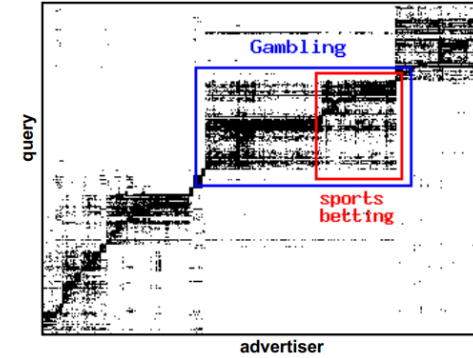


부르는 용어

Communities, clusters,
groups, modules



가라테를 배우는 스포츠 동아리에서
갈등이 발생해, 두 집단으로 파가 갈림



- Lots of internal connection = strong edges , few external ones = weak edges

Network Communities

✓ community 를 찾을 수 있는 방법은

📌 Modularity : 네트워크가 얼마나 잘 커뮤니티로 구분 (partition) 되어 있는지 측정하는 지표

$$Q \propto \sum_{s \in S} [(\# \text{ edges within group } s) - \underbrace{(\text{expected } \# \text{ edges within group } s)}_{\text{Need a null model}}]$$

노드 i와 j사이에 존재하는 엣지수의 기대값

$$k_i \cdot \frac{k_j}{2m} = \frac{k_i k_j}{2m}$$

$$Q(G, S) = \underbrace{\frac{1}{2m}}_{\text{Normalizing const.: } -1 \leq Q \leq 1} \sum_{s \in S} \sum_{i \in s} \sum_{j \in s} \left(A_{ij} - \frac{k_i k_j}{2m} \right)$$

- 음수 : 기대한 것보다 실제 연결된 개수가 작은 경우
→ 별 상관없는 커뮤니티를 정의한 것
- 양수 : 기대한 것보다 실제 연결된 개수가 많은 경우
→ 유의미한 커뮤니티를 정의한 것 (0.3~0.7 정도가 유의미한 커뮤니티라 볼 수 있음)

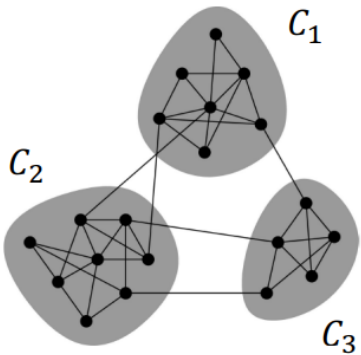
• Q 를 최대화 하는 커뮤니티를 찾으면 됨

Louvain Algorithm



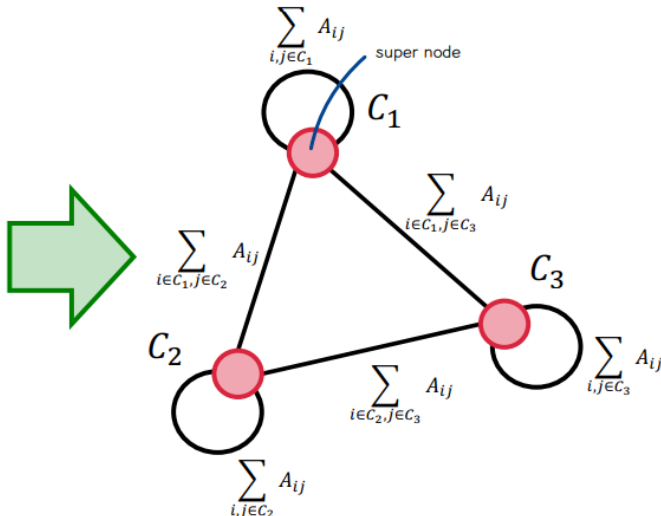
Louvain Algorithm

Community assignment
obtained after 1st phase



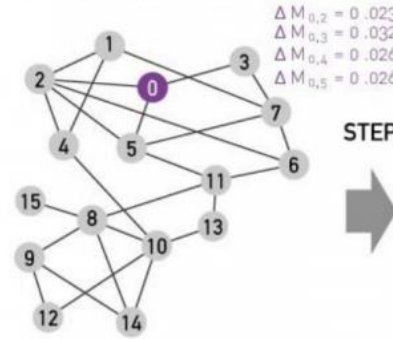
3/6/21

Jure Leskovec, Stanford CS224W: Machine Learning with Graphs



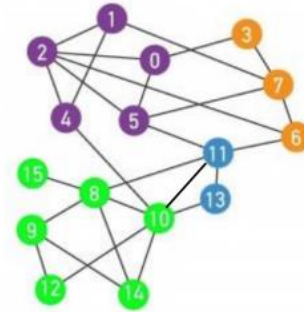
42

1ST PASS

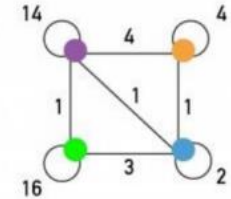


$\Delta M_{0,2} = 0.023$
 $\Delta M_{0,3} = 0.032$
 $\Delta M_{0,4} = 0.026$
 $\Delta M_{0,5} = 0.026$

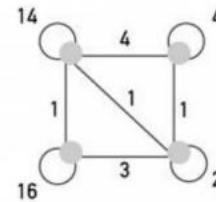
STEP I



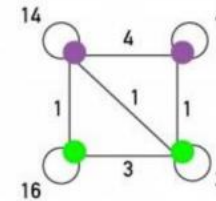
STEP II



2ND PASS



STEP I



STEP II



- 커뮤니티를 발견하기 위한 greedy 한 알고리즘 : 빠르게 수렴하고 좋은 커뮤니티를 결과로 가져옴

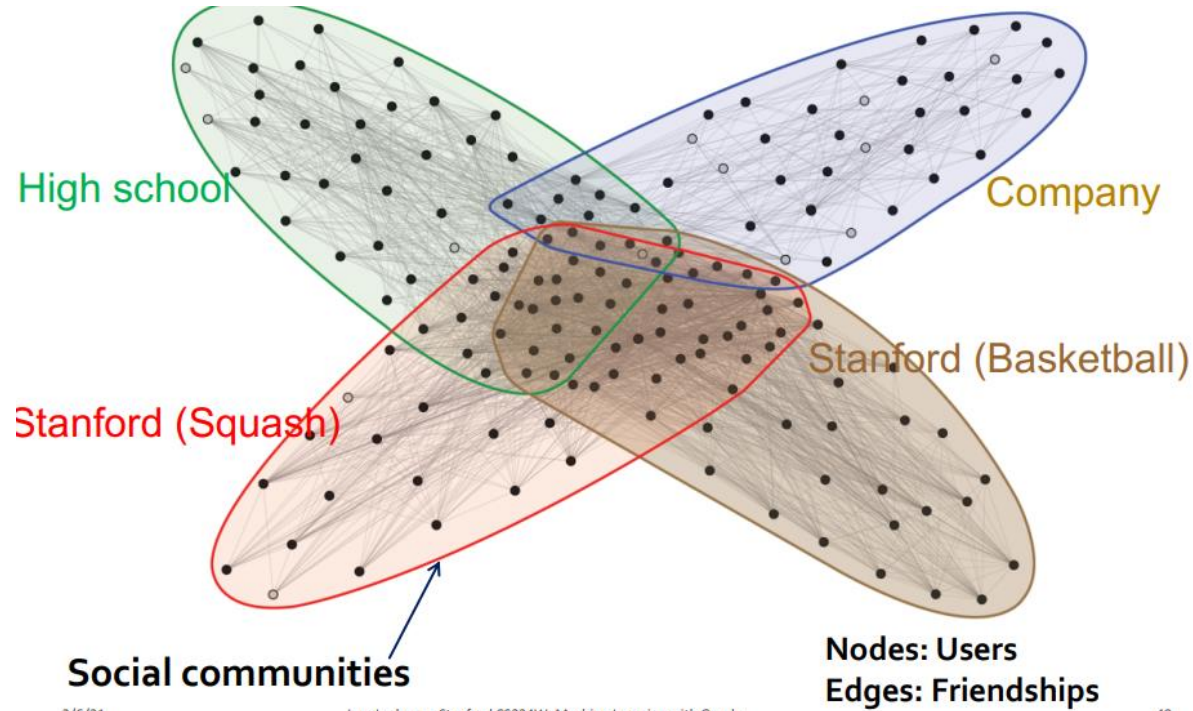
- (1) local 한 범위에서 변화를 가해 Q 를 최적화 시킴

- (2) Super node 를 만들어 최적의 community 를 찾음

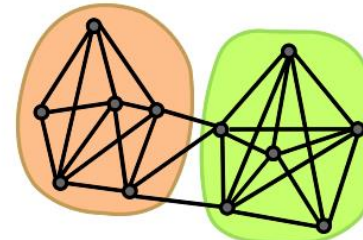
Overlapping Communities



Overlapping Communities

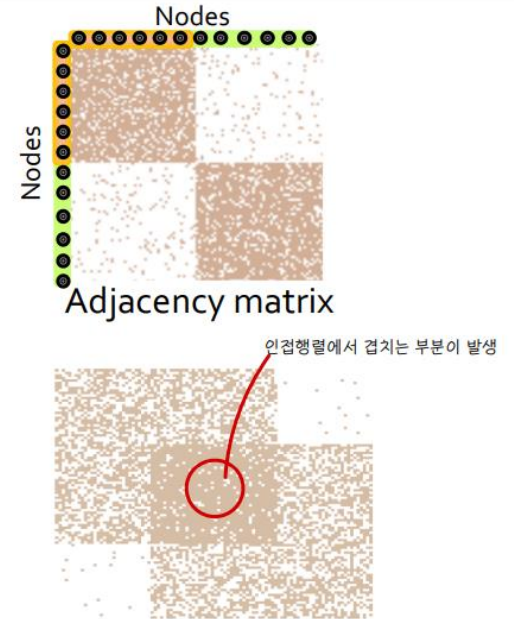
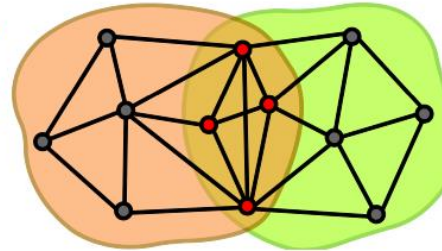


Non-overlapping



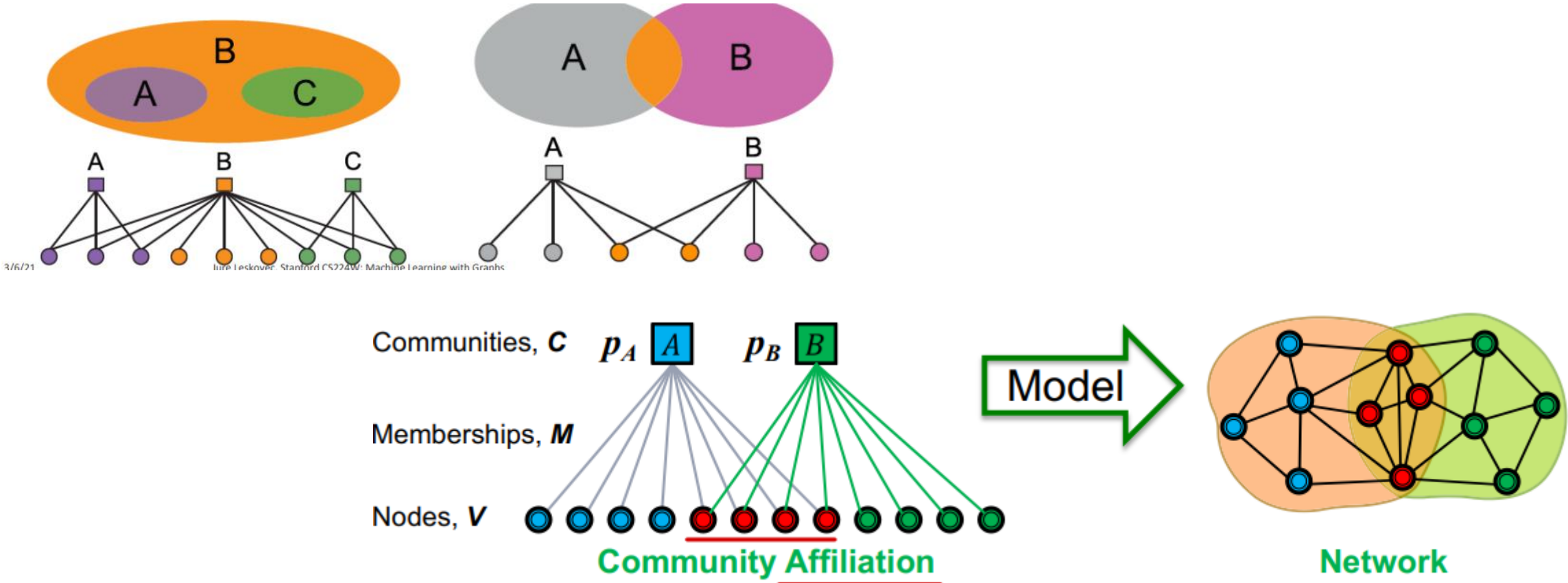
Network

Overlapping



- overlapping 된 커뮤니티는 어떻게 detection 할까?
- 실제 데이터에서는 overlapping 된 경우가 훨씬 많다. 인접행렬에서 겹치는 부분이 발생.

AGM

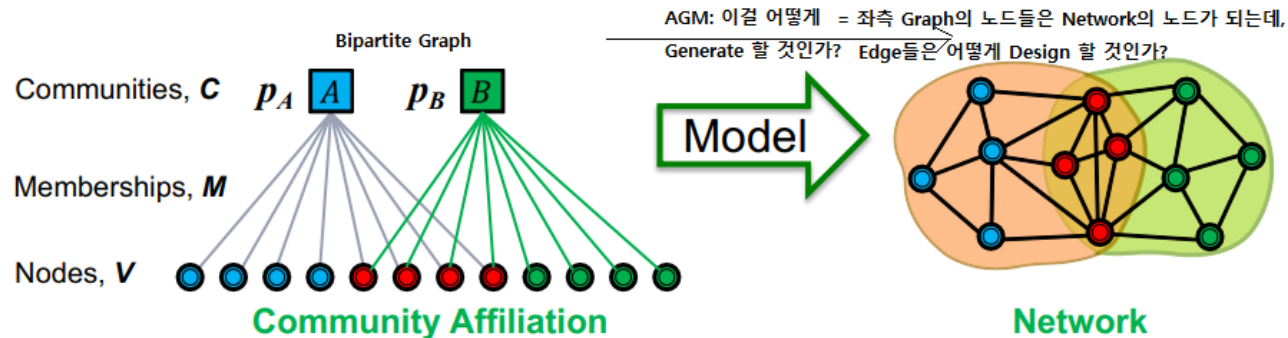


- overlapping community 를 detection 하는 방법
- (1) Generative model 을 정의 (ex. AGM) \hookrightarrow AGM 이 뭐지 ?
- (2) 주어진 그래프 G 를 생성할 수 있는 최적의 AGM 을 찾는다 \hookrightarrow 최적화를 어떻게 하지?

Overlapping Communities

① Generative model and model parameter

- 왼쪽 구조를 인풋으로 받아, 오른쪽의 네트워크 구조로 생성해내는 모델



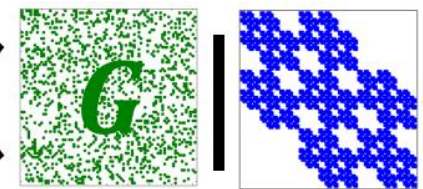
- 모델 파라미터

- V : node , C : community , M : membership, P_c : 커뮤니티 c 의 노드들이 서로 연결되어 있을 가능성도

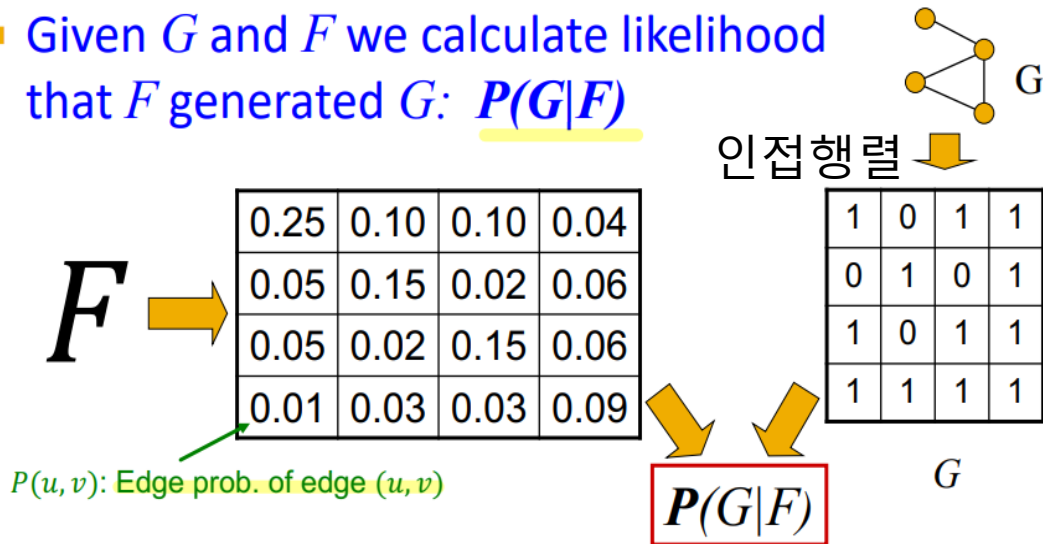
👉 모델 파라미터 최적화를 어떻게 할까 : membership strength와 MLE 를 사용

Overlapping Communities

② optimize => BigCLAM model

$$\arg \max_F P(G | F)$$


- Given G and F we calculate likelihood that F generated G : $P(G|F)$



F가 G 를 generate 할 likelihood

$$P(G|F) = \prod_{(u,v) \in G} P(u, v) \prod_{(u,v) \notin G} (1 - P(u, v))$$

Likelihood of edges in the graph Likelihood of edges not in the graph

$$P(u, v) = 1 - \exp(-F_u^T F_v)$$

- u와 v가 서로 연결되어 있을 확률
- shared membership 의 strength 에 비례함 (많은 걸 공유할 수록 연결될 확률이 올라감)

- F_{uA} : The membership strength of node u to community A ($F_{uA} = 0$: no membership)
- 노드 u 의 community A 에 대한 소속감 (membership strength)