

Graph Representation Learning

Chapter 1, 2

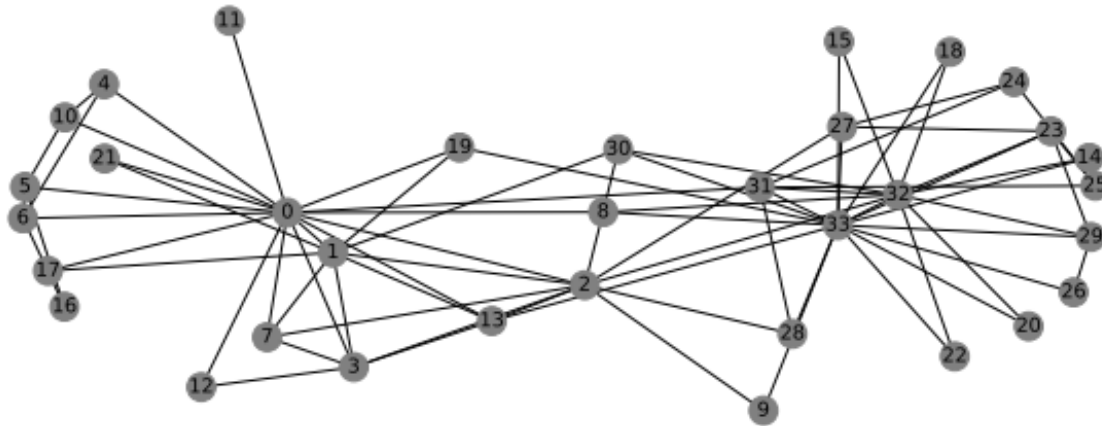
Yurim Lee

MLV Lab Intern

GNN Study

Chapter 1

1. Introduction



collection of objects (i.e., nodes)
a set of interactions (i.e., edges)

Figure 1.1: The famous *Zachary Karate Club Network* represents the friendship relationships between members of a karate club studied by Wayne W. Zachary from 1970 to 1972. An edge connects two individuals if they socialized outside of the club. During Zachary's study, the club split into two factions—centered around nodes 0 and 33—and Zachary was able to correctly predict which nodes would fall into each faction based on the graph structure [Zachary, 1977].

1.1 What is graph?

$G = (V, E)$, a set of nodes V , a set of edges E between these nodes
 $(u, v) \in E$, Edge going from node $u \in V$ to node $v \in V$

adjacency matrix $A \in \mathbb{R}^{|V| \times |V|}$

$A[u, v] = 1$ if $(u, v) \in E$ and $A[u, v] = 0$ otherwise

Undirected edge만 있는 경우 adjacency matrix가 symmetric이지만 directed인 경우 항상 symmetric은 아님
weighted edges는 adjacency matrix에서 0,1 대신 real value 사용

1.1.1 Multi-relational Graphs

Multi-relational Graph:

두 종류 이상의 관계에 대한 연결성을 나타낼 수 있는 그래프

$(u, \tau, v) \in E$, τ = relation edge, $u \&v$ = node, τ 마다 adjacency matrix A_τ 존재

$$A \in \mathbb{R}^{|V| \times |R| \times |V|}$$

Heterogeneous Graph: 노드 집합 V 가 disjoint subsets의 합집합으로 분리될 수 있는 그래프

$$V = V_1 \cup V_2 \cup \dots \cup V_k \text{ where } V_i \cap V_j = \emptyset, \forall i \neq j$$

한 종류의 관계 τ_i 는 특정 두 노드 부분 집합 사이의 연결을 나타냄

$$(u, \tau_i, v) \in E \rightarrow u \in V_j, v \in V_k$$

Multiparatite Graph는 Heterogeneous의 한 예로 edge는 서로 다른 종류의 node만을 연결 가능

$$(u, \tau_i, v) \in E \rightarrow u \in V_j, v \in V_k \wedge j \neq k$$

1.1.1 Multi-relational Graphs

Multiplex Graph:

K 개의 layer 존재

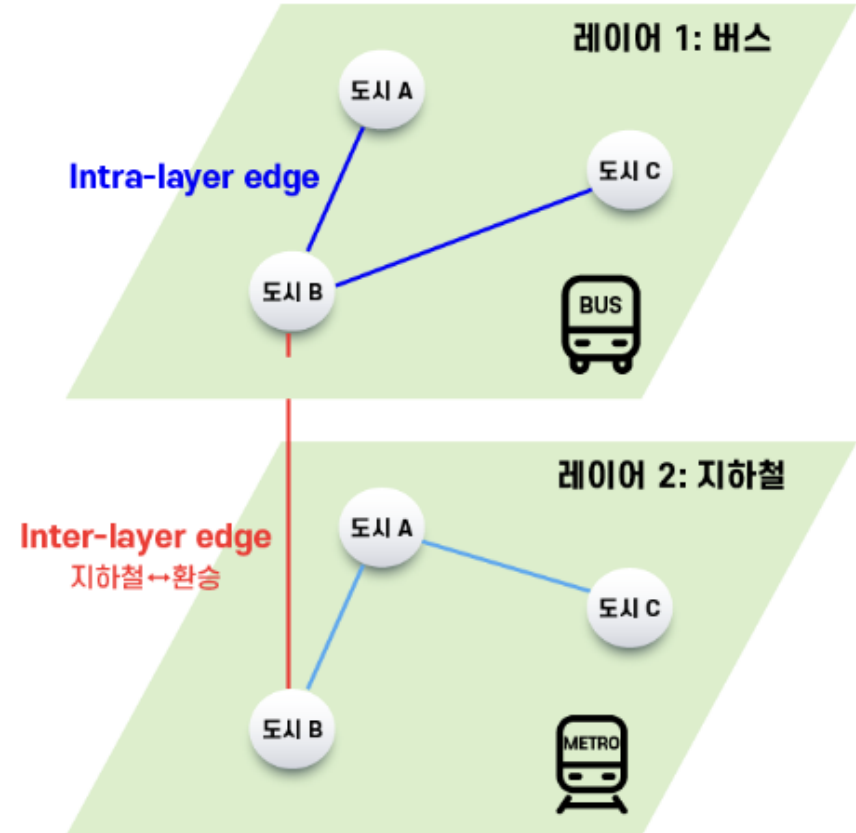
모든 node는 모든 layer에 존재

각 layer는 하나의 관계를 나타냄

Intra-layer edge: layer 내부에서 연결되는 edge

Inter-layer edge:

서로 다른 두 layer에서 같은 node끼리 연결되는 edge



1.1.2 Feature Information

Attribute 또는 Feature 라고 부르며 이는 노드에 부여된 속성값

Node의 order와 adjacency matrix order가 일치할 때 $\mathbf{X} \in \mathbb{R}^{|V| \times m}$

Heterogenous 그래프에서는 다른 노드가 다른 feature를 갖고 있다고 가정

Discrete edge type외에 real-valued edge 타입 고려하거나
real-valued feature를 전체 그래프에 부여하기도 함

1.2.1 Node classification

소량의 labeled node를 통해 unlabeled된 node들 클래스 분류

기존의 지도 학습 모델은 independent and identically distributed(i.i.d.)되어 있어야 하지만 그래프 내 node들은 i.i.d. 되어있지 않아 상호연결된 node를 모델링할 수 있어야 함

Homophily: 이웃된 node는 비슷한 feature를 공유할 것

Structural Equivalence: 이웃 연결 구조가 비슷한 node끼리는 비슷한 label을 가질 것

Heterophily: 자신과 label이 다른 node와 우선적으로 연결될 것

1.2.2 Relation prediction

Relation Prediction: 그래프 안에서 node 사이의 연결성 예측
(Link prediction, graph completion, relational inference..)

set of nodes V 와 $E_{\text{train}} \subset E$ 를 통해서 missing edge $E \setminus E_{\text{train}}$ 의 연결성 예측

Graph Domain에 specific한 inductive bias를 필요로 함

1.2.3 Clustering and community detection

Community Detection: 주어진 그래프에서 node community를 detect 하는 task

Node들은 같은 community에 속하는 node들과 edge를 형성

Latent community structure를 detect하는 것이 어려움

Chapter 2

2.1.1 Node-level statistics and features

Node degree

u의 차수는 노드 u에 인접한 이웃 node의 개수

$$d_u = \sum_{v \in V} A[u, v].$$

node u와 v가 이웃이라면 $A[u, v]=1$, otherwise 0
따라서 이의 sum은 이웃 node의 개수

But, 이웃 node 정보 반영 불가

2.1.1 Node-level statistics and features

Node centrality

Eigenvector Centrality

이웃 node의 중요도를 반영하여 나의 중요도 결정

그래프 안에서 infinite length의 random walk할 때, 해당 node에 방문할 확률

$$e_u = \frac{1}{\lambda} \sum_{v \in V} \mathbf{A}[u, v] e_v \quad \forall u \in \mathcal{V}, \quad \lambda \mathbf{e} = \mathbf{A} \mathbf{e}.$$

$$\mathbf{e}^{(t+1)} = \mathbf{A} \mathbf{e}^{(t)}.$$

$\mathbf{e}^{(0)} = (1, 1, \dots, 1)^\top$ 에서 시작하여 power iteration을 하고, 첫번째 iteration 후에 $\mathbf{e}^{(1)}$ 은 모든 node의 degree를 담을 것. $\mathbf{e}^{(t)}$ 은 해당 node에 t 회 워킹으로 도착하는 경우의 수

2.1.1 Node-level statistics and features

Betweenness Centrality

해당 node가 다른 두 node 사이의 최단 경로에 있는 경우의 수

Closeness Centrality

해당 노드와 다른 노드 사이의 최단 경로의 평균값

2.1.1 Node-level statistics and features

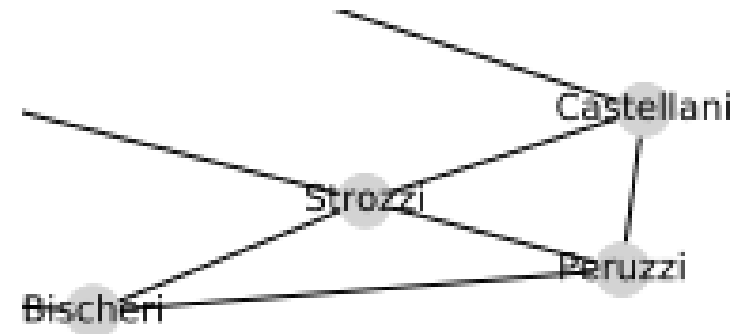
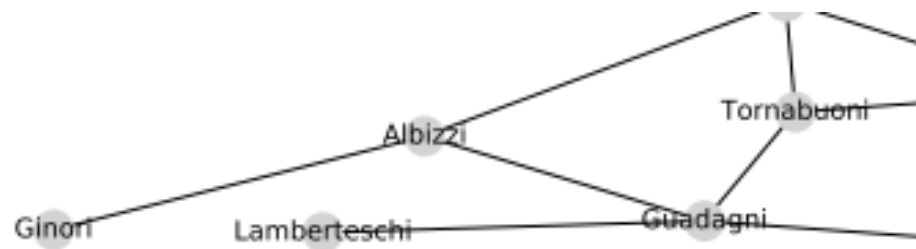
Clustering Coefficient

한 노드와 이웃 노드들이 얼마나 긴밀하게 연결되어 있는가

$$c_u = \frac{|(v_1, v_2) \in \mathcal{E} : v_1, v_2 \in \mathcal{N}(u)|}{\binom{d_u}{2}}.$$

$$\mathcal{N}(u) = \{v \in V : (u, v) \in E\}$$

U의 clustering Coefficient 계수가 1이면, u의 이웃은 서로 모두 연결되어 있음



2.1.2 Graph-level features and graph kernels

The Weisfeiler-Lehman kernel

Iterative neighborhood aggregation을 통해 두 그래프의 isomorphism 확인

1. 모든 node에 초기 label을 할당.
일반적으로 degree를 할당함
2. 현재 label의 mulsti-set을 해시 함수에
대입하여 새로운 고유 레이블 생성
3. K번 반복하여 k-hop neighborhood의
정보를 반영한 레이블에 대한
histogram을 feature vector 로 사용

Multi-set은 원소의 중복을 허용

HASH함수는 multi-set으로 표현되는 label이 겹치지 않도록 index 부여

1. First, we assign an initial label $l^{(0)}(v)$ to each node. In most graphs, this label is simply the degree, i.e., $l^{(0)}(v) = d_v \forall v \in V$.
2. Next, we iteratively assign a new label to each node by hashing the multi-set of the current labels within the node's neighborhood:

$$l^{(i)}(v) = \text{HASH}(\{\{l^{(i-1)}(u) \forall u \in \mathcal{N}(v)\}\}), \quad (2.6)$$

where the double-braces are used to denote a multi-set and the HASH function maps each unique multi-set to a unique new label.

3. After running K iterations of re-labeling (i.e., Step 2), we now have a label $l^{(K)}(v)$ for each node that summarizes the structure of its K -hop neighborhood. We can then compute histograms or other summary statistics over these labels as a feature representation for the graph. In other words, the WL kernel is computed by measuring the difference between the resultant label sets for two graphs.

2.1.2 Graph-level features and graph kernels

Bag of Nodes

Node centrality, clustering coefficient, degree와 같은 정보들을 aggregate하여 사용
But, local node level information에 기반하기에 중요한 global property를 놓칠 수 있음

Graphlets and path-based methods

Graphlet이란 그래프 내에서 작은 subgraph의 구조

Graphlet kernel은 이러한 graphlet이 전체 graph에서 얼마나 자주 등장하는지를 나타냄. 그러나 computation difficulty 존재

Path-based method는 graphlet대신에 발생 가능한 path의 종류를 조사하여 사용

2.2 Neighborhood overlap detection

Neighborhood overlap detection

공유하는 이웃 노드 개수를 통해 node 간 유사도 측정

$$\mathbf{S}[u, v] = |\mathcal{N}(u) \cap \mathcal{N}(v)|,$$

S는 모든 pairwise node statistics에 대한 similarity matrix

Relation Prediction

실제 edge 집합의 일부인 Etrain만 알고 있을 때 실제 E를 찾는 방법

Given $\mathbf{S}[u, v]$, $\mathbf{A}[u, v]$ 가 1일 확률은 $\mathbf{S}[u, v]$ 에 비례한다고 가정

언제 edge prediction을 할지에 대한 threshold만 필요

$$P(\mathbf{A}[u, v] = 1) \propto \mathbf{S}[u, v].$$

2.2.1 Local overlap measures

Function of the number of sharing neighbors

But, edge가 많은 node는 그 값이 클 것이기에 normalization 필요

$$\mathbf{S}_{\text{Sorenson}}[u, v] = \frac{2|\mathcal{N}(u) \cap \mathcal{N}(v)|}{d_u + d_v}, \quad \mathbf{S}_{\text{Jaccard}}[u, v] = \frac{|\mathcal{N}(u) \cap \mathcal{N}(v)|}{|\mathcal{N}(u) \cup \mathcal{N}(v)|}.$$

The lower, the more informative

$$\mathbf{S}_{\text{RA}}[v_1, v_2] = \sum_{u \in \mathcal{N}(v_1) \cap \mathcal{N}(v_2)} \frac{1}{d_u},$$

Resource allocation

$$\mathbf{S}_{\text{AA}}[v_1, v_2] = \sum_{u \in \mathcal{N}(v_1) \cap \mathcal{N}(v_2)} \frac{1}{\log(d_u)}.$$

Adamic Adar

But, only consider local node neighbors라는 한계

Ex. 공유하는 이웃 node가 없다고 둘이 같은 community에 속하지 않으라는 법 없음

2.2.2 Global overlap measures

Katz index

두 노드를 연결하는 모든 길이의 경로의 합

$$\mathbf{S}_{\text{Katz}}[u, v] = \sum_{i=1}^{\infty} \beta^i \mathbf{A}^i[u, v],$$

$$\mathbf{S}_{\text{Katz}} = (\mathbf{I} - \beta \mathbf{A})^{-1} - \mathbf{I},$$

식 (2.15)에 대한 직관적인 설명은 다음과 같다. 학창시절에 배운 무한 등비급 수를 잠시 떠올려보자. $0 \leq r < 1$ 일 때 $\sum_{n=1}^{\infty} r^n$ 의 값은 $\frac{r}{1-r} = \frac{1}{1-r} - 1$ 이었다. 식 (2.15)는 무한 등비급수의 행렬판이라고 생각하면 쉽다. $\beta \mathbf{A}$ 을 r 으로, \mathbf{I} 를 1으로, 역행렬을 역수로 생각하고 대입하면 정확히 식 (2.15)가 나오게 된다. 교재에 보다 더 자세한 증명이 나와있으니 필요한 사람들은 찾아보면 좋을 것 같다.

But, 차수가 높은 node일수록 그 값이 커진다는 한계

2.2.2 Global overlap measures

Geometric series of matrices The Katz index is one example of a geometric series of matrices, variants of which occur frequently in graph analysis and graph representation learning. The solution to a basic geometric series of matrices is given by the following theorem:

Theorem 1. Let \mathbf{X} be a real-valued square matrix and let λ_1 denote the largest eigenvalue of \mathbf{X} . Then

$$(\mathbf{I} - \mathbf{X})^{-1} = \sum_{i=0}^{\infty} \mathbf{X}^i$$

if and only if $\lambda_1 < 1$ and $(\mathbf{I} - \mathbf{X})$ is non-singular.

Proof. Let $s_n = \sum_{i=0}^n \mathbf{X}^i$ then we have that

$$\begin{aligned} \mathbf{X}s_n &= \mathbf{X} \sum_{i=0}^n \mathbf{X}^i \\ &= \sum_{i=1}^{n+1} \mathbf{X}^i \end{aligned}$$

and

$$\begin{aligned} s_n - \mathbf{X}s_n &= \sum_{i=0}^n \mathbf{X}^i - \sum_{i=1}^{n+1} \mathbf{X}^i \\ s_n(\mathbf{I} - \mathbf{X}) &= \mathbf{I} - \mathbf{X}^{n+1} \\ s_n &= (\mathbf{I} - \mathbf{X}^{n+1})(\mathbf{I} - \mathbf{X})^{-1} \end{aligned}$$

And if $\lambda_1 < 1$ we have that $\lim_{n \rightarrow \infty} \mathbf{X}^n = 0$ so

$$\begin{aligned} \lim_{n \rightarrow \infty} s_n &= \lim_{n \rightarrow \infty} (\mathbf{I} - \mathbf{X}^{n+1})(\mathbf{I} - \mathbf{X})^{-1} \\ &= \mathbf{I}(\mathbf{I} - \mathbf{X})^{-1} \\ &= (\mathbf{I} - \mathbf{X})^{-1} \end{aligned}$$

2.2.2 Global overlap measures

Leicht, Holme, and Newman (LHN) similarity

$$\frac{\mathbf{A}^i}{\mathbb{E}[\mathbf{A}^i]}, \quad \mathbb{E}[\mathbf{A}[u, v]] = \frac{d_u d_v}{2m}, \quad \mathbb{E}[\mathbf{A}^2[v_1, v_2]] = \frac{d_{v_1} d_{v_2}}{(2m)^2} \sum_{u \in \mathcal{V}} (d_u - 1) d_u.$$

$$\mathbf{A} \mathbf{p}_i = \lambda_1 \mathbf{p}_{i-1}, \quad \mathbb{E}[\mathbf{A}^i[u, v]] = \frac{d_u d_v \lambda_1^{i-1}}{2m}. \quad \mathbf{S}_{\text{LNH}}[u, v] = \mathbf{I}[u, v] + \frac{2m}{d_u d_v} \sum_{i=0}^{\infty} \beta^i \lambda_1^{1-i} \mathbf{A}^i[u, v],$$

$$\mathbf{S}_{\text{LNH}} = 2\alpha m \lambda_1 \mathbf{D}^{-1} \left(\mathbf{I} - \frac{\beta}{\lambda_1} \mathbf{A} \right)^{-1} \mathbf{D}^{-1},$$

node 간의 expected path를 계산하여 node간 실제 path가 expected path보다 높을 때만 높은 유사도 부여

2.2.2 Global overlap measures

Random walk methods

한 node u 에서 시작한 random walk가 node v 에 자주 방문했다면 둘은 유사할 것

$$\mathbf{P} = \mathbf{A}\mathbf{D}^{-1}$$

\mathbf{P} 는 Stochastic Matrix

$$\mathbf{q}_u = c\mathbf{P}\mathbf{q}_u + (1 - c)\mathbf{e}_u.$$

C : probability that random walk restarts at node u at each time step.

C 가 없으면 probability는 eigenvector centrality의 normalized 분산으로 수렴할 것

$$\mathbf{S}_{\text{RW}}[u, v] = \mathbf{q}_u[v] + \mathbf{q}_v[u],$$

2.3 Graph Laplacian and Spectral Methods

Unnormalized Laplacian

$$\mathbf{L} = \mathbf{D} - \mathbf{A},$$

1. It is symmetric ($\mathbf{L}^T = \mathbf{L}$) and positive semi-definite ($\mathbf{x}^T \mathbf{L} \mathbf{x} \geq 0, \forall \mathbf{x} \in \mathbb{R}^{|\mathcal{V}|}$).
2. The following vector identity holds $\forall \mathbf{x} \in \mathbb{R}^{|\mathcal{V}|}$

$$\mathbf{x}^T \mathbf{L} \mathbf{x} = \frac{1}{2} \sum_{u \in \mathcal{V}} \sum_{v \in \mathcal{V}} \mathbf{A}[u, v] (\mathbf{x}[u] - \mathbf{x}[v])^2 \quad (2.27)$$

$$= \sum_{(u,v) \in \mathcal{E}} (\mathbf{x}[u] - \mathbf{x}[v])^2 \quad (2.28)$$

3. \mathbf{L} has $|V|$ non-negative eigenvalues: $0 = \lambda_{|\mathcal{V}|} \leq \lambda_{|\mathcal{V}|-1} \leq \dots \leq \lambda_1$

2.3 Graph Laplacian and Spectral Methods

Normalized Laplacian

$$\mathbf{L}_{\text{sym}} = \mathbf{D}^{-\frac{1}{2}} \mathbf{L} \mathbf{D}^{-\frac{1}{2}},$$

$$\mathbf{L}_{\text{RW}} = \mathbf{D}^{-1} \mathbf{L}$$

Scale의 정도 차는 존재하지만, 유사함

2.3.2 Graph Cuts and Clustering

Graph Cuts

$\text{cut}\{A_1, A_2, \dots, A_K\}$ 는 그래프의 노드 집합 V 를 공통된 원소 없이 K 개의 노드 부분 집합으로 나눈 것

Cut은 partition of node를 가로지르는 edge의 개수

$$\text{cut}(\mathcal{A}_1, \dots, \mathcal{A}_K) = \frac{1}{2} \sum_{k=1}^K |(u, v) \in \mathcal{E} : u \in \mathcal{A}_k, v \in \bar{\mathcal{A}}_k|.$$

두 node 집합 사이에 연결된 edge 개수의 합

2.3.2 Graph Cuts and Clustering

Approximating the RatioCut with the Laplacian spectrum

$$\mathbf{a}[u] = \begin{cases} \sqrt{\frac{|\bar{\mathcal{A}}|}{|\mathcal{A}|}} & \text{if } u \in \mathcal{A} \\ -\sqrt{\frac{|\mathcal{A}|}{|\bar{\mathcal{A}}|}} & \text{if } u \in \bar{\mathcal{A}} \end{cases}.$$

$$\begin{aligned} \mathbf{a}^\top \mathbf{L} \mathbf{a} &= \sum_{(u,v) \in \mathcal{E}} (\mathbf{a}[u] - \mathbf{a}[v])^2 \\ &= \sum_{(u,v) \in \mathcal{E} : u \in \mathcal{A}, v \in \bar{\mathcal{A}}} (\mathbf{a}[u] - \mathbf{a}[v])^2 \\ &= \sum_{(u,v) \in \mathcal{E} : u \in \mathcal{A}, v \in \bar{\mathcal{A}}} \left(\sqrt{\frac{|\bar{\mathcal{A}}|}{|\mathcal{A}|}} - \left(-\sqrt{\frac{|\mathcal{A}|}{|\bar{\mathcal{A}}|}} \right) \right)^2 \\ &= \text{cut}(\mathcal{A}, \bar{\mathcal{A}}) \left(\frac{|\mathcal{A}|}{|\bar{\mathcal{A}}|} + \frac{|\bar{\mathcal{A}}|}{|\mathcal{A}|} + 2 \right) \\ &= \text{cut}(\mathcal{A}, \bar{\mathcal{A}}) \left(\frac{|\mathcal{A}| + |\bar{\mathcal{A}}|}{|\bar{\mathcal{A}}|} + \frac{|\mathcal{A}| + |\bar{\mathcal{A}}|}{|\mathcal{A}|} \right) \\ &= |\mathcal{V}| \text{RatioCut}(\mathcal{A}, \bar{\mathcal{A}}). \end{aligned}$$

2.3.2 Graph Cuts and Clustering

Approximating the RatioCut with the Laplacian spectrum

$$\sum_{u \in \mathcal{V}} \mathbf{a}[u] = 0, \text{ or equivalently, } \mathbf{a} \perp \mathbb{1} \quad (\text{Property 1})$$

$$\|\mathbf{a}\|^2 = |\mathcal{V}| \quad (\text{Property 2}),$$

$$\min_{\mathbf{a} \in \mathcal{V}} \mathbf{a}^\top \mathbf{L} \mathbf{a}$$

s.t.

$$\mathbf{a} \perp \mathbb{1}$$

$$\|\mathbf{a}\|^2 = |\mathcal{V}|$$

\mathbf{a} defined as in Equation 2.38.

2.3.2 Graph Cuts and Clustering

Approximating the RatioCut with the Laplacian spectrum

$$\min_{\mathbf{a} \in \mathbb{R}^{|\mathcal{V}|}} \mathbf{a}^\top \mathbf{L} \mathbf{a}$$

s.t.

$$\mathbf{a} \perp \mathbf{1}$$

$$\|\mathbf{a}\|^2 = |\mathcal{V}|.$$

$$\begin{cases} u \in \mathcal{A} & \text{if } \mathbf{a}[u] \geq 0 \\ u \in \bar{\mathcal{A}} & \text{if } \mathbf{a}[u] < 0. \end{cases}$$

2.3.2 Graph Cuts and Clustering

Optimal한 node cluster의 한 방법은 cluster의 개수 k 가 최소인 k 를 찾는 것

$$\text{RatioCut}(\mathcal{A}_1, \dots, \mathcal{A}_K) = \frac{1}{2} \sum_{k=1}^K \frac{|(u, v) \in \mathcal{E} : u \in \mathcal{A}_k, v \in \bar{\mathcal{A}}_k|}{|\mathcal{A}_k|},$$

군집이 작으면 penalty 부여

$$\text{NCut}(\mathcal{A}_1, \dots, \mathcal{A}_K) = \frac{1}{2} \sum_{k=1}^K \frac{|(u, v) \in \mathcal{E} : u \in \mathcal{A}_k, v \in \bar{\mathcal{A}}_k|}{\text{vol}(\mathcal{A}_k)},$$

Cluster에 속한 node degree 합으로 cut을 나눔

모든 cluster가 비슷한 개수의 edge를 갖도록 만드는 효과

2.3.3 Generalized spectral clustering

Node를 k 개의 cluster로 나누는 method

1. Find the K smallest eigenvectors of \mathbf{L} (excluding the smallest):
 $\mathbf{e}_{|\mathcal{V}|-1}, \mathbf{e}_{|\mathcal{V}|-2}, \dots, \mathbf{e}_{|\mathcal{V}|-K}.$
2. Form the matrix $\mathbf{U} \in \mathbb{R}^{|\mathcal{V}| \times (K-1)}$ with the eigenvectors from Step 1 as columns.
3. Represent each node by its corresponding row in the matrix \mathbf{U} , i.e.,

$$\mathbf{z}_u = \mathbf{U}[u] \quad \forall u \in \mathcal{V}.$$

4. Run K-means clustering on the *embeddings* $\mathbf{z}_u \quad \forall u \in \mathcal{V}.$

Graph node를 Graph Laplacian을 통해 표현 가능
node간 relation과 similarity를 활용하여 효과적으로 clustering 가능

Thank you!