DSBA
Data Science & Business Analytics

Probabilistic Graphical Model and Network Data [IME659]

Data Imputation Using Graph

Final Presentation

3조

2018010938 이정호
2020010551 김혜연
2020021319 이윤승
2020021326 윤훈상
2020011132 정의석

Index

Index

Problem Definition

Importance of Data Imputation

이름	성별	나이	키	…	Y
김혜연	F	26	161	…	1
정의석	M	27	174	…	0
윤훈상	M	29	183	…	0
이윤승	F	26	163	…	1

<완전 데이터>

이름	성별	나이	키	…	Y
김혜연	F	NaN	161	…	1
정의석	M	27	NaN	…	0
윤훈상	NaN	29	183	…	0
이윤승	F	26	NaN	…	1

<불완전 데이터>

❖ 데이터의 완결성 (Data Completeness)

- 데이터 분석 알고리즘은 데이터의 완결성을 가정으로 둠
- 데이터 집합에 속하는 모든 개체들의 속성 값이 빠짐 없이 존재하는 것을 의미함
- 하지만, 대부분의 설문조사 데이터, User-Item matrix 등, 현실에서 수집된 데이터들은 결측치들이 다수 존재함

Project Overview

Missing Data Imputation

Data Matrix with Missing Values

	F_1	F_2	F_3	F_4
O_1	0.3	0.5	NA	0.1
O_2	NA	NA	0.6	0.2
O_3	0.3	NA	NA	0.5

Labels

Y
y_1
y_2
?

❖ 결측 데이터의 처리 (Data Imputation)

- 분석 시 결측치를 제거하는 **"완전 제거법"**은 상관 관계, 회귀계수 등 통계적 분석결과에 영향을 끼치기 때문에, 일반적으로 통계적 기법을 이용하여 결측치를 채워 넣음

❖ 프로젝트 문제 정의 (Project Problem Definition)

- 임상데이터의 경우 표본을 수집하기 어렵고, 개인정보에 의한 결측이 빈번하게 발생하기 때문에 Missing Value Imputation은 상당히 중요한 문제임
- 본 프로젝트에서는 **"그래프 기반 결측치 처리 기법을 통한 임상데이터 결측치 처리"**를 주제로 연구를 진행함

Index

Data Imputation Methods

Statistical Method for Data Imputation

통계값 사용
(Single Imputation)

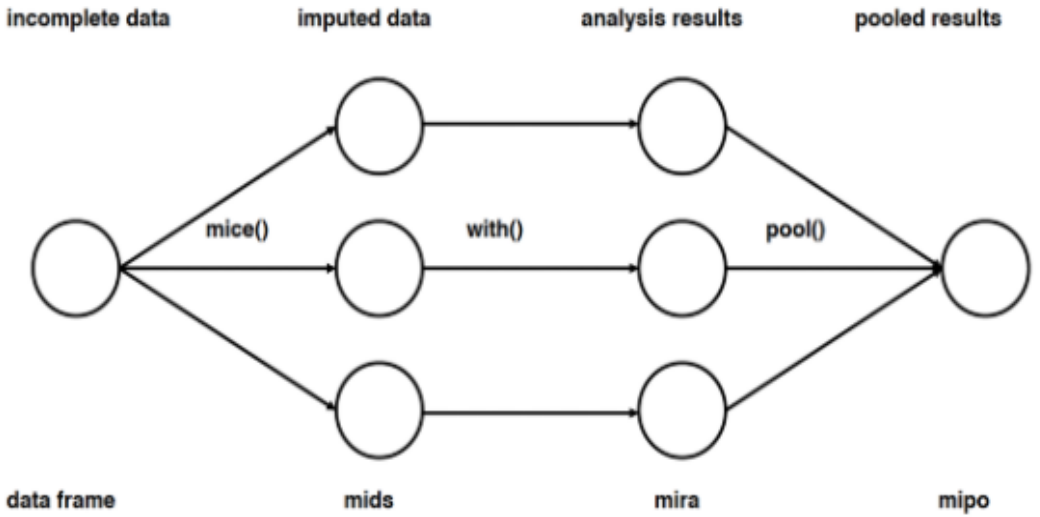
	col1	col2	col3	col4	col5
0	2	5.0	3.0	6	NaN
1	9	NaN	9.0	0	7.0
2	19	17.0	NaN	9	NaN

mean()

	col1	col2	col3	col4	col5
0	2.0	5.0	3.0	6.0	7.0
1	9.0	11.0	9.0	0.0	7.0
2	19.0	17.0	6.0	9.0	7.0

- 결측이 없는 데이터들의 통계값을 사용
- 연속형 변수 : 평균,중앙값
- 범주형 변수 : 최빈값
- 단점 : 채워지는 값들이 동일하게 설정되어, 독립된 객체들의 특징을 손실

MICE
(Multiple Imputation by Chained Equations)



- 일반적으로 통계적 방법론 중 가장 많이 사용됨
- Single imputation 을 거친 여러 개의 데이터셋을 통해 결측값을 평가

Data Imputation Methods

Graphical Method for Data Imputation

❖ GRAPE 모델 설명

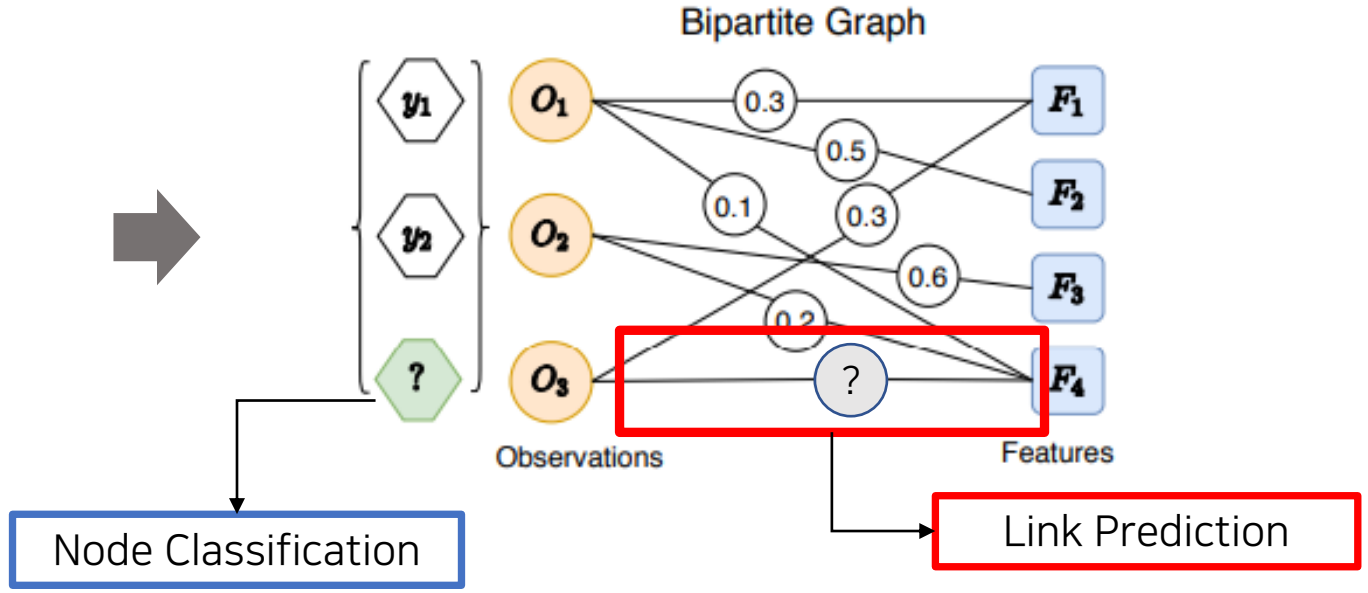
이분할 그래프로 Matrix 를 표현함

Data Matrix with Missing Values

	F_1	F_2	F_3	F_4
O_1	0.3	0.5	NA	0.1
O_2	NA	NA	0.6	0.2
O_3	0.3	NA	NA	0.5

Labels

Y
y_1
y_2
?



- Observations 와 Features 는 서로 weight 를 가지는 edge 로 연결 되어 있다고 가정

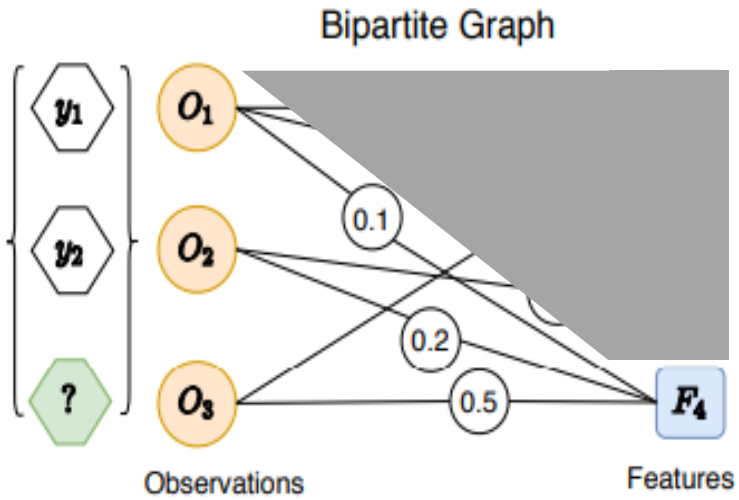
  → 두 Type 이 다른 O,F Node 관계에서 이루는 값을 edge weight 로 가정함

  → 이 때 Masking(NA) 되어있는 결측값은 연결이 되지 않은 edge 가 되며, 이 edge 의 weight 값을 Linked Prediction 으로 해결함

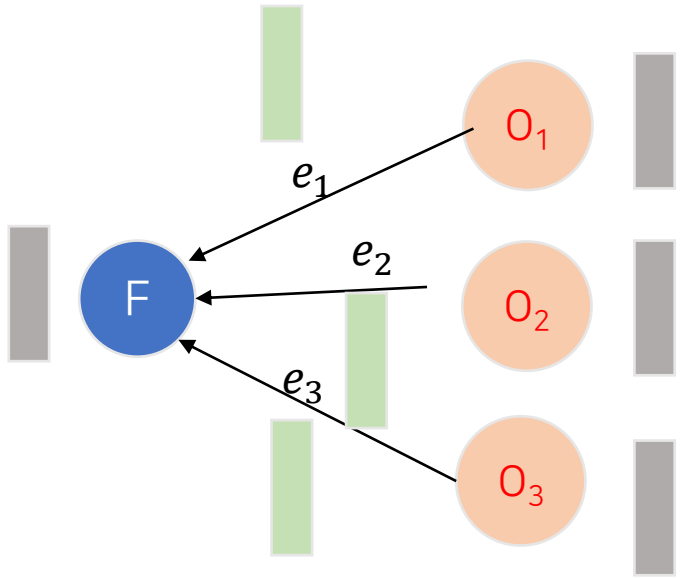
Data Imputation Methods

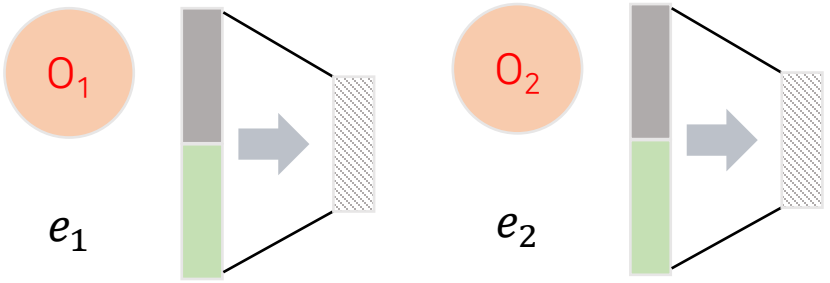
Graphical Method for Data Imputation

❖ GRAPE 모델 설명



O1, O2, O3 에서
F4 연결 부분 확대





Step 1. Object node 와 edge node 의 결합

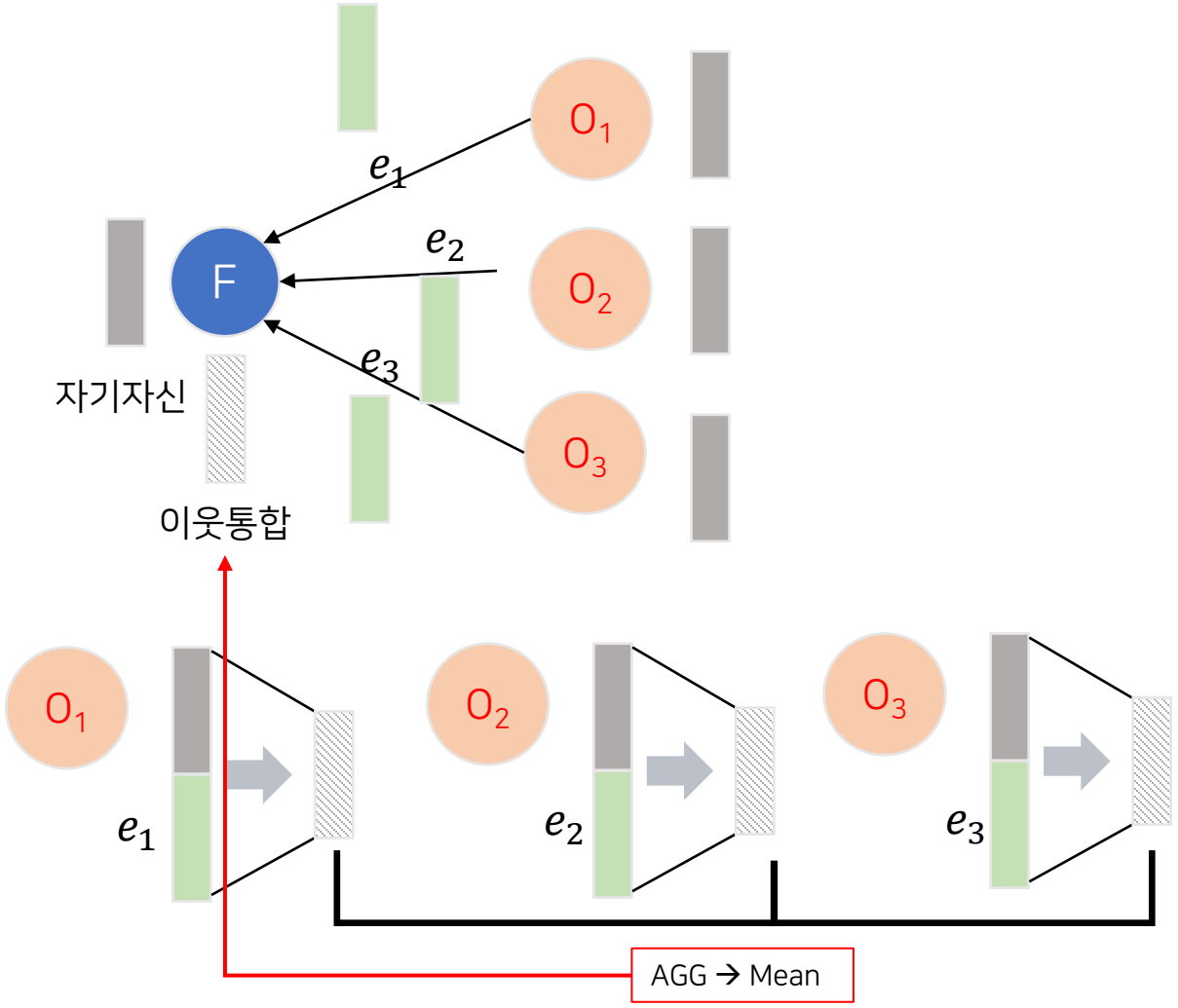
→ Object 와 edge 값을 통해 어떤 Feature 와 연결 될지 알아 낼 수 있음

✓ P : Learnable
✓ Relu
✓ Normalization

Data Imputation Methods

Graphical Method for Data Imputation

❖ GRAPE 모델 설명



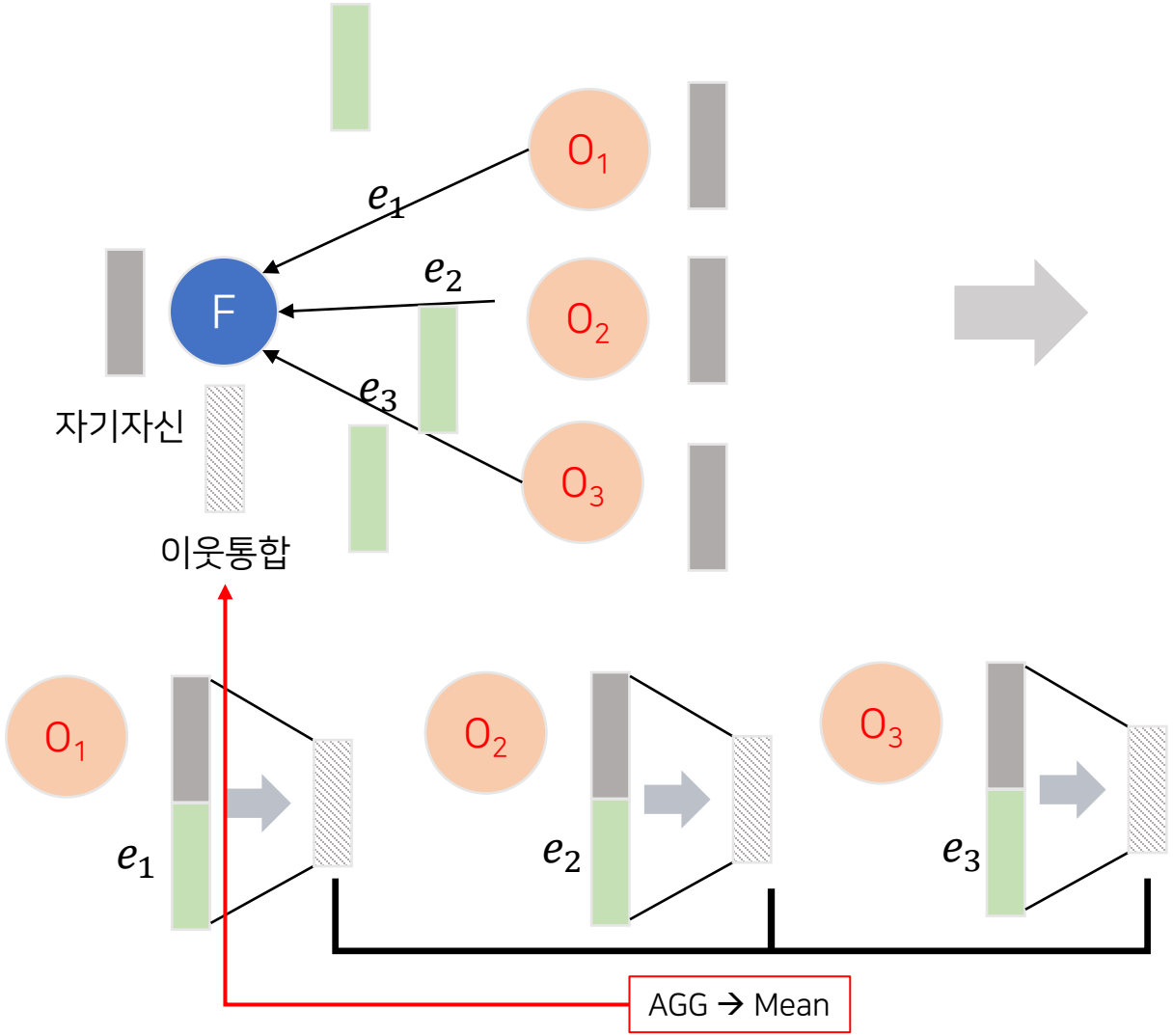
Step 2. 결합된 정보를 평균을 통해 Aggregation 함

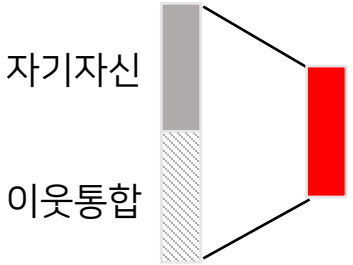
→ 이웃들의 정보(Edge 연결된) 를 대표하는 값을 표현함

Data Imputation Methods

Graphical Method for Data Imputation

❖ GRAPE 모델 설명





✓ Q : Learnable
✓ Relu
✓ Normalization

Step 3. 자신과 이웃대표값을 결합

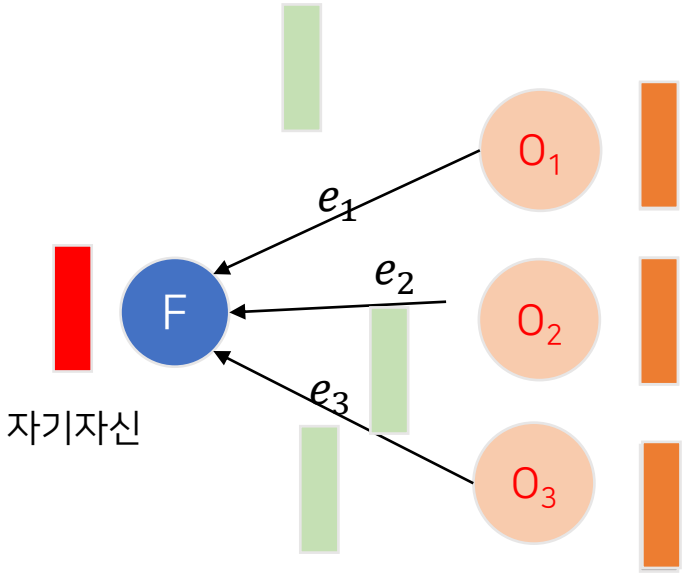
Step 2. 결합된 정보를 평균을 통해 Aggregation 함

→ 이웃들의 정보(Edge 연결된) 를 대표하는 값을 표현함

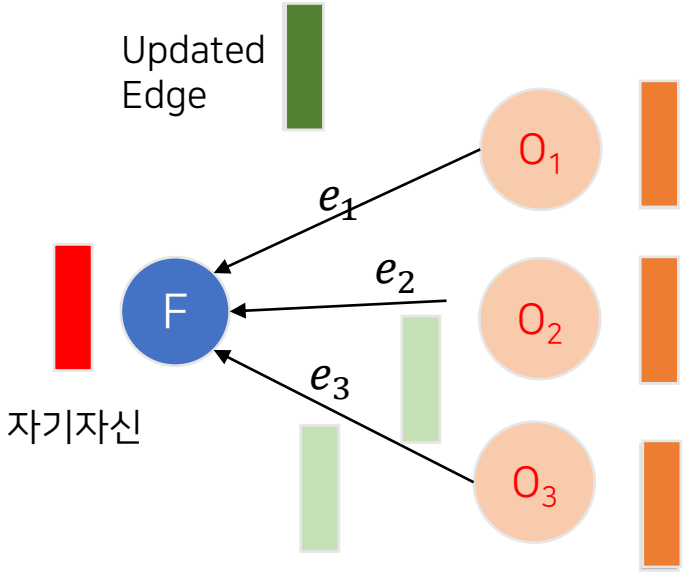
Data Imputation Methods

Graphical Method for Data Imputation

❖ GRAPE 모델 설명

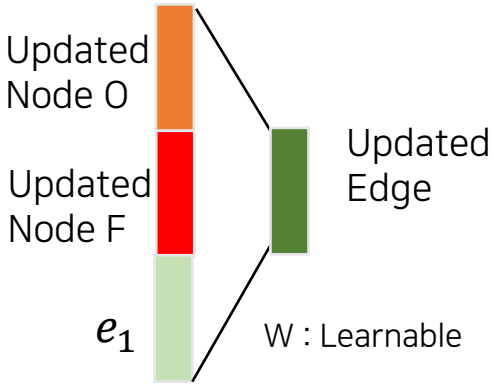


Edge Update



Step 4. Node Update 를

위와 같은 방식으로 모든 노드에 진행함



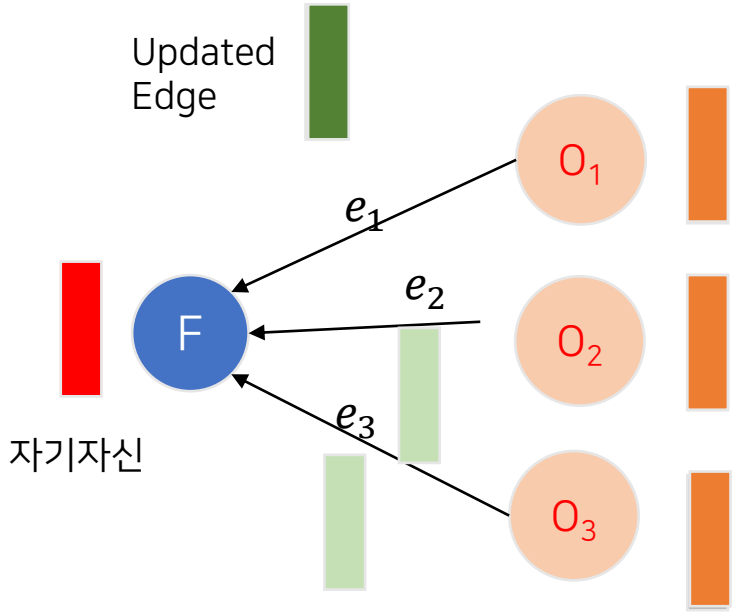
Step 5. Update 된 두 노드와 Edge 를

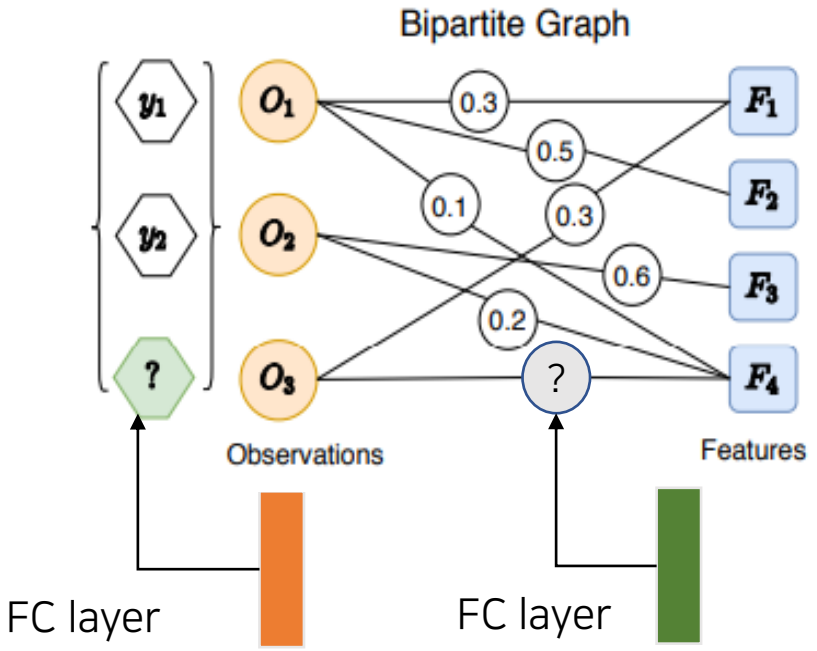
결합하여 Edge 정보를 변환함

Data Imputation Methods

Graphical Method for Data Imputation

❖ GRAPE 모델 설명





Step 6. 최종 Representation 을 통해

Node Classification, Link Prediction 을 계산

Node Classification

Link Prediction

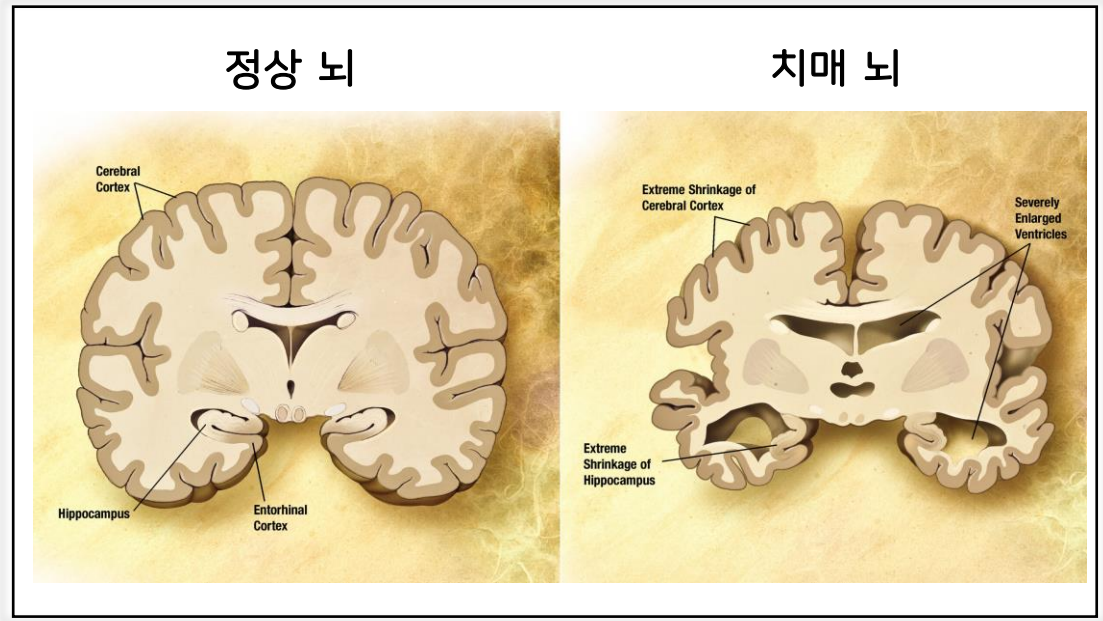
Index

Dataset

Alzheimer Dataset

❖ 알츠하이머 환자 정보 데이터셋

⇒ Alzheimer: 치매의 75%가 알츠하이머병으로 가장 흔한 형태



❖ 데이터셋 출처

⇒ Open Access Series of Imaging Studies

❖ 데이터셋 구성

- 60~96세의 150명의 환자 진료 정보 (373건)
- 각 환자에 대한 뇌 MRI Dataset
- MRI Data와 함께 개인정보 / 사회적 정보가 함께 포함

Experiments

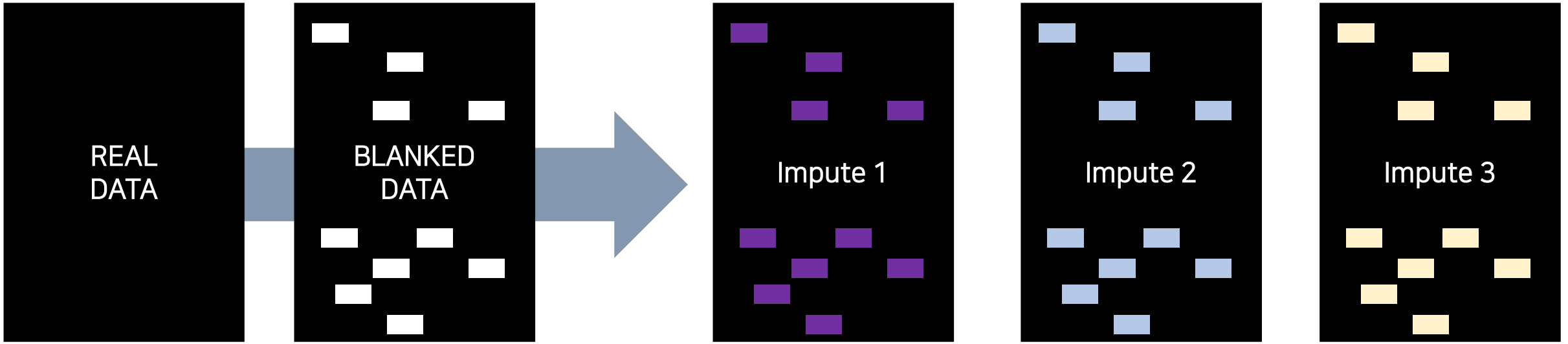
Pipeline

❖ 실험 목적

Graph Neural Network 기반의 Imputation 기법이 원본 데이터의 결측치를 잘 보완할 수 있는지 판단

❖ 실험 설계

앞에서 소개한 기법들을 사용해 Data Imputation 수행 : Mean, MICE와 GRAPE 간의 Imputation 성능 비교를 진행

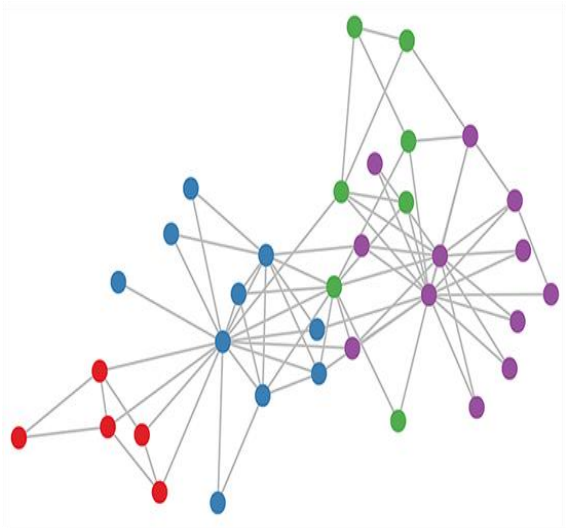


Experiments

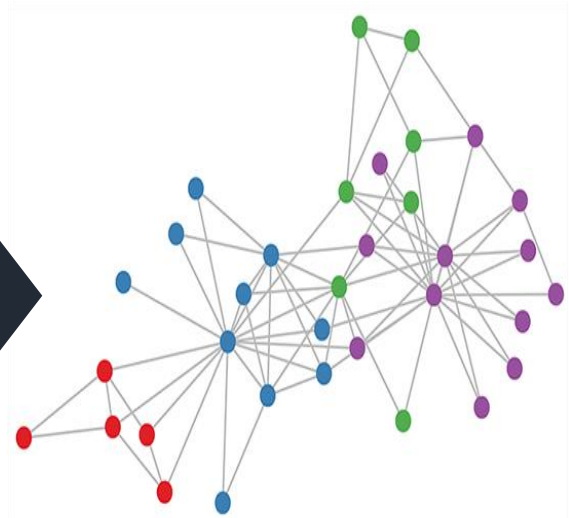
Evaluation

❖ 성능 평가

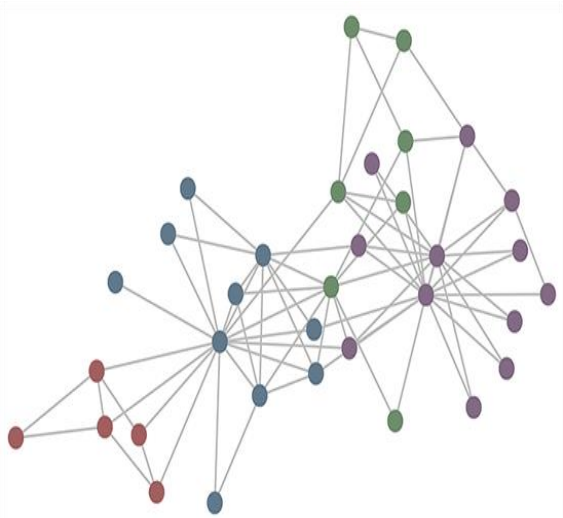
- 성능 비교는 Community Detection과 Classification 수행 결과가 실제 REAL DATA와 유사하면 좋은 성능으로 판단함
- Community Detection의 경우, 각각의 node는 instance를 의미하며, edge는 instance간의 유사도(≥ 0.95)를 가리킴
- Classification Model의 경우, Support Vector Machine, Random Forest, Logistic Regression 3가지를 수행함



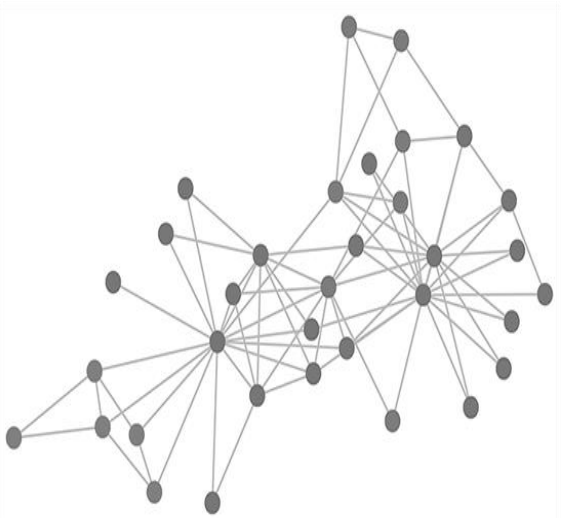
Original Network



GRAPE Imputation Network



MICE Imputation Network

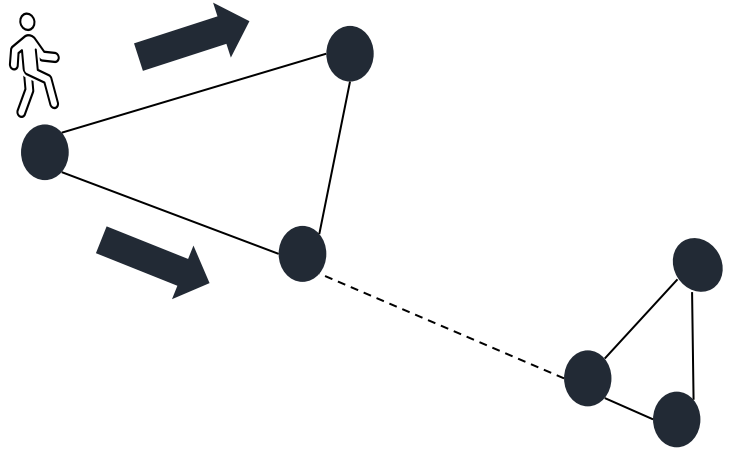


Mean Imputation Network

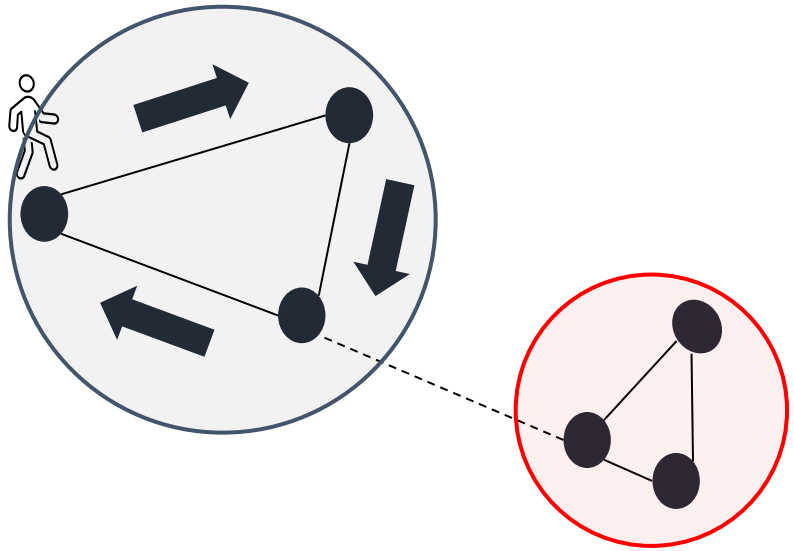
Community Detection

Walk Trap

- ✓ **Random Walk :** 각 단계에서 보행자(walker)가 꼭지점(vertex)에서 시작하여 이웃들 사이로 무작위적(randomly), 그리고 균일하게 (uniformly) 다음 꼭지점을 선택한다. Random Walk는 꼭지점 간의 유사도 계산에도 사용될 수 있다.
- ✓ **Walk Trap :** 무작위(random)하게 걷는 보행자는 통상적으로(Intuitively) 촘촘하게 연결되어 있는 부분, 커뮤니티 안에 갇히는 (trapped) 경향이 있다.



Random Walk



Walk Trap

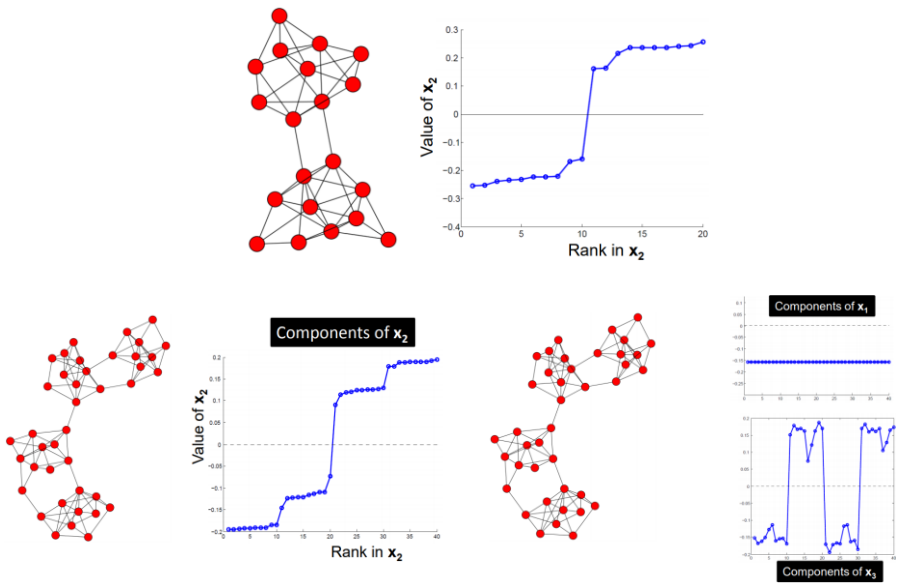
Community Detection

Modularity Based

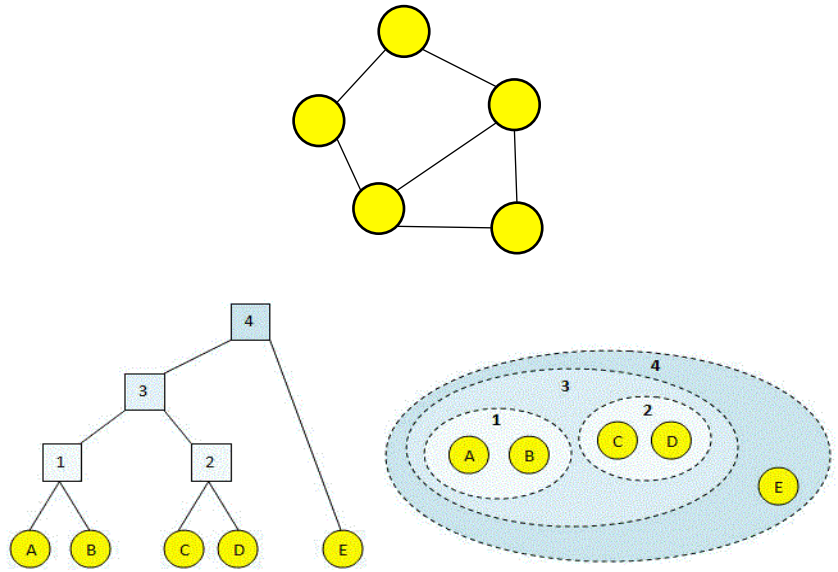
✓ **Eigen Vector (Partition)** : Graph / Network에 대한 Modularity Matrix의 Eigenvector 부호를 통해 Community 구성.

$Ax = \lambda x \rightarrow$ 가장 큰 Eigenvalue의 Eigenvector의 부호로 Partition

✓ **Clauset-Newman-Moore Greedy Modularity Maximization** : Modularity가 가장 크게 증가하는 Community Pair를 Greedy하게 합쳐가는 것으로 Community를 구성.



Eigen Vector (Partition)



Clauset-Newman-Moore Greedy Modularity Maximization

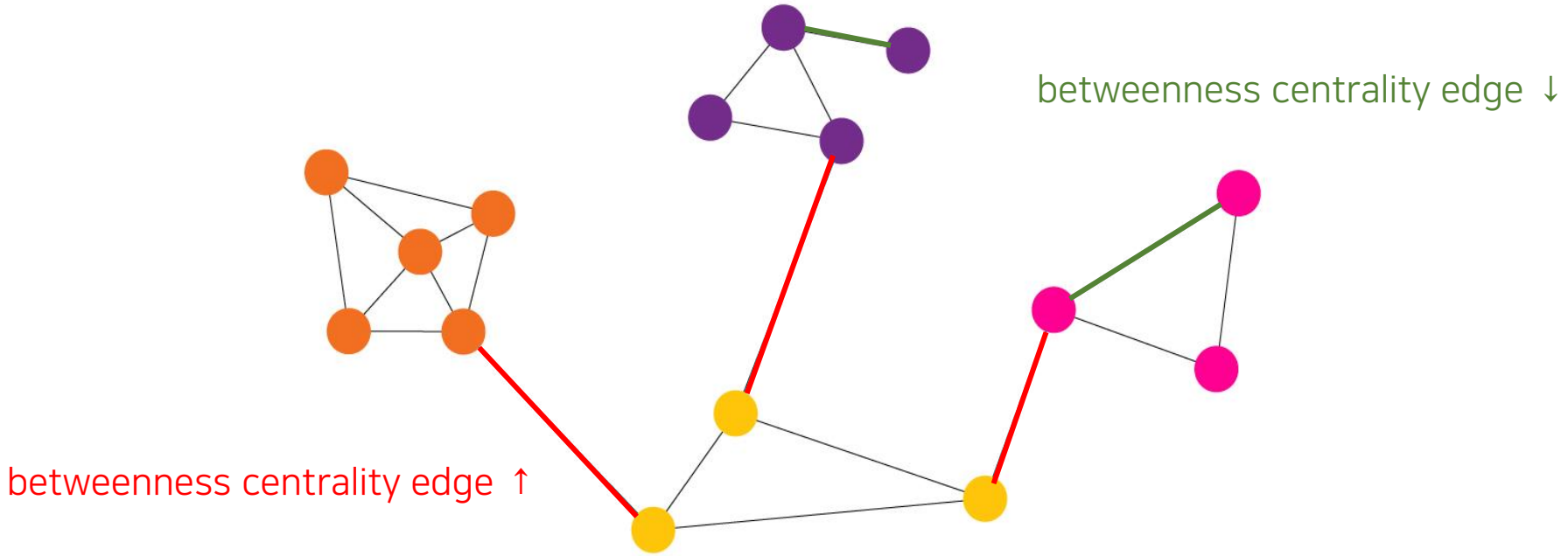
Community Detection

Betweenness

✓ Betweenness

- Single node가 고립될 우려가 있는 Hierarchical Clustering를 보완하고자 제안된 방법
- Betweenness Centrality를 edge별로 계산하고, 높은 강도의 edge 삭제 (community간 다리 역할이라 판단)

  * Betweenness Centrality: 최단 경로에서 특정 edge의 등장빈도



Index

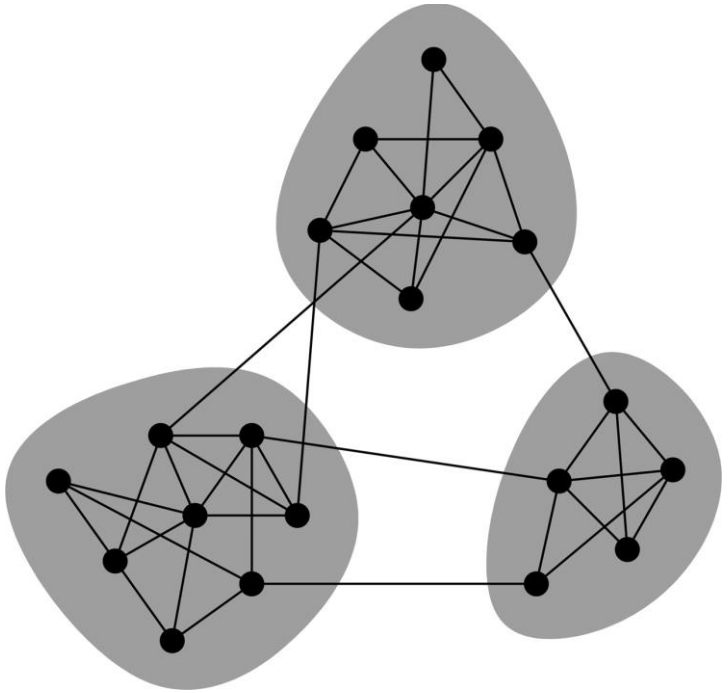
Modularity
Definition

❖ Modularity 정의

✓ 동일한 Community내에서, Random으로 구성된 Edge에 비해서 현재 연결되어 있는 Edge가 나타날 기대값

⇒ 높을수록 모듈성이 높으며, Community내의 결속력이 높다.

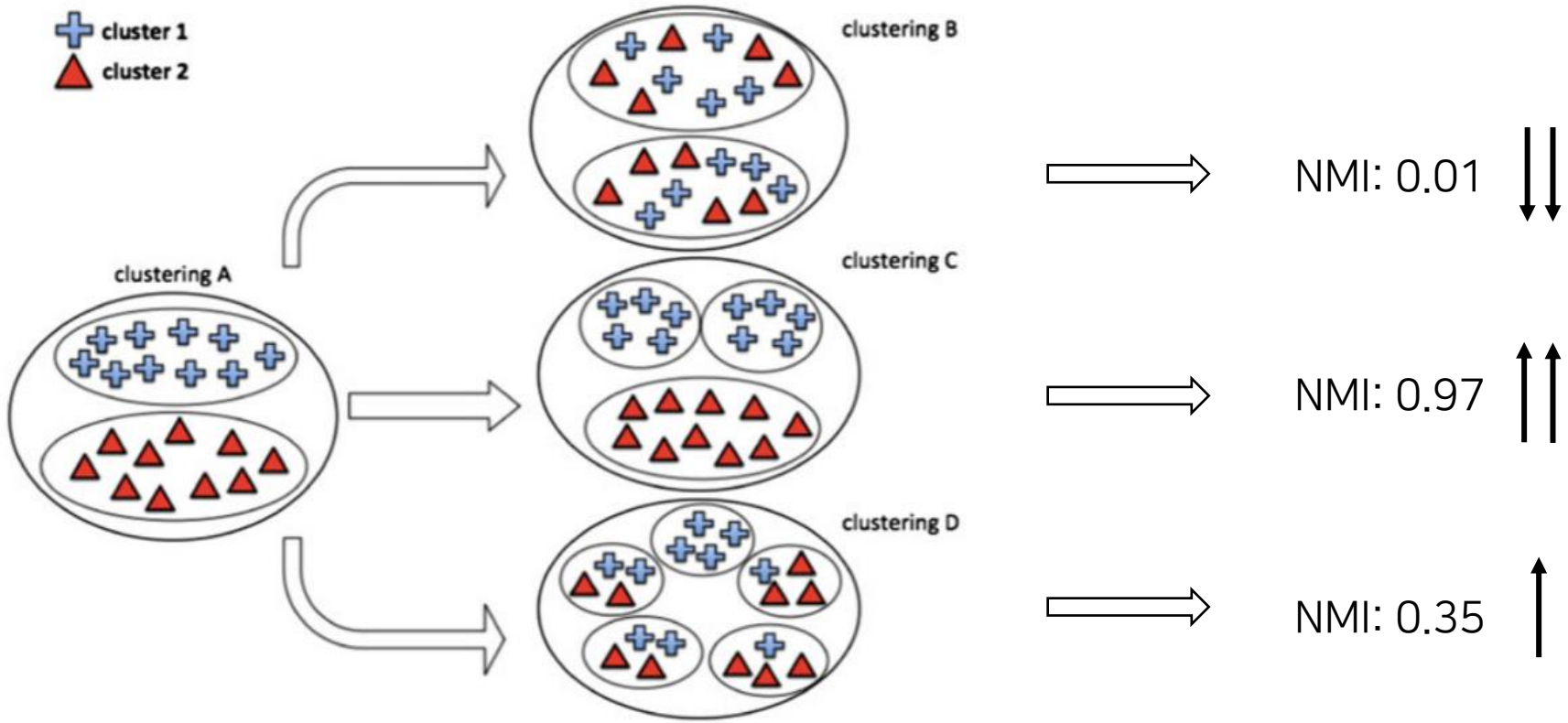
$$\Rightarrow \boldsymbol{Q} = \frac{\mathbf{1}}{\mathbf{2}\boldsymbol{M}} \sum_{\boldsymbol{i},\boldsymbol{j}}^{\boldsymbol{N}} \left(\boldsymbol{a_{ij}}, \boldsymbol{t_{ij}}\right) \boldsymbol{\delta}\left[\boldsymbol{C}[\boldsymbol{i}], \boldsymbol{C}[\boldsymbol{j}]\right]$$



Score

NMI Score

❖ NMI 정의



✓ Mutual Information 값으로 Clustering의 성능을 표현하는 점수

Score

Modularity

Community	Original	Mean Imputation	MICE Imputation	GRAPE Imputation
Eigen	**0.783**	0.692	0.785	0.924
Greedy	**0.783**	0.698	0.785	0.924
Betweenness	**0.720**	0.423	0.767	0.912
WalkTrap	**0.760**	0.670	0.785	0.924

- ✓ Original Dataset에서 모든 기법들에 대한 Modularity가 약 0.7로 나타남
- ✓ Mean Imputation, Mice Imputation은 Modularity가 Original과 작거나 유사하나 GRAPE는 Modularity가 더 강화되는 양상을 보임

Score

NMI Score

Community	Original	Mean Imputation	MICE Imputation	GRAPE Imputation
Eigen	**0.4104**	0.1105	0.1609	0.1879
Greedy	**0.4586**	0.1115	0.1680	0.1877
Betweenness	**0.1879**	0.1221	0.1609	0.1879
WalkTrap	**0.1879**	0.1098	0.1609	0.1879

* **빨간색** 글씨는 Original과 가장 유사한 성능

✓ Threshold 95일 때, Eigen, Betweenness, WalkTrap, Greedy 네가지 Community Detection에서 GRAPE가 Original Data와 가장 근접한 NMI 결과를 보임

Score

Classification Model Accuracy

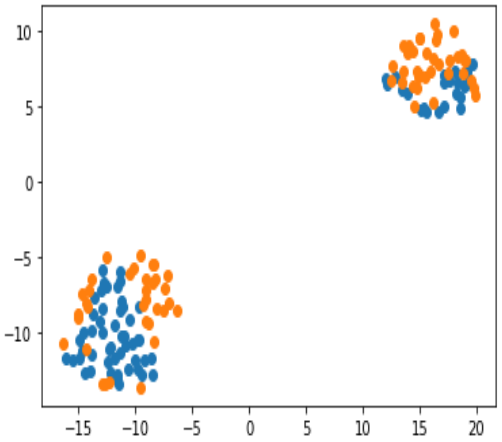
Model	Original	MICE	MEAN	GRAPE
SVM	98.27 %	82.39 %	87.32 %	89.03 %
Random Forest	98.69 %	85.21 %	83.80 %	88.58 %
Logistic Regression	88.73 %	83.09 %	88.89 %	88.62 %

* **빨간색** 글씨는 Original과 가장 유사한 성능

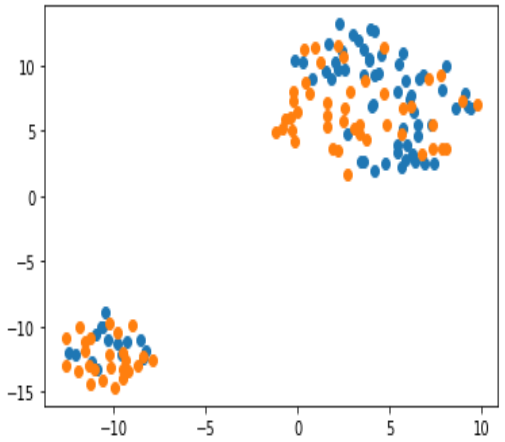
- ✓ SVM, Random Forest, Logistic Regression 3가지 Classification 모델을 통해 예측을 해본 결과 실제 Original 데이터를 사용한 것만큼 성능이 나오지는 못함
- ✓ 하지만, 다른 Imputation 방법론에 비해 가장 유사한 성능을 이끌어낸 것을 확인할 수 있음

Visualization

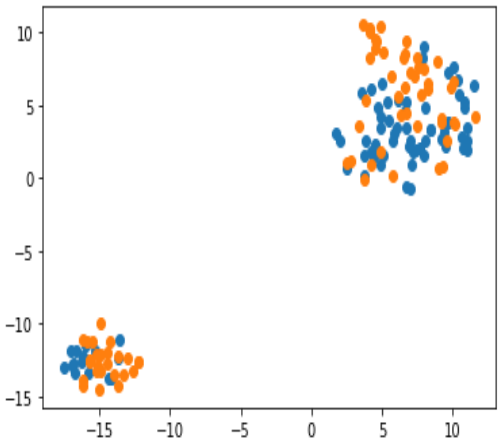
T-SNE



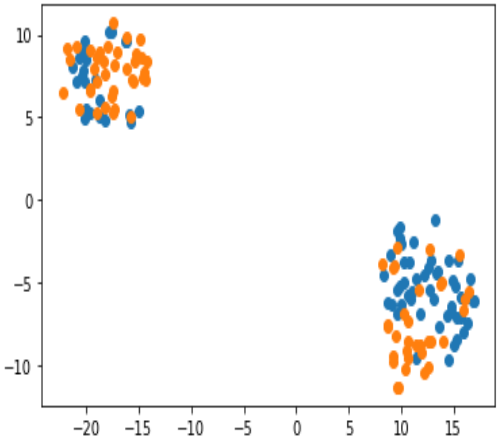
[Original]



[MICE]



[MEAN]



[GRAPE]

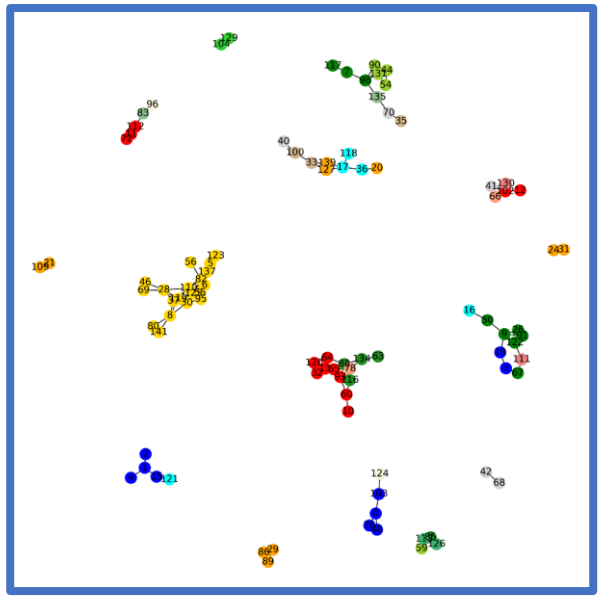
- ✓ 데이터 셋을 T-SNE 로 시각화 한 결과 원본 데이터에서도 암환자 여부는 혼재 되어있는 상황
- ✓ MICE, MEAN Imputation 은 밀집도가 원본과 조금 달라짐
- ✓ GRAPE Imputation 은 원본과 상당히 유사한 분포를 보임
- ✓ 위 데이터 셋을 Community detection 을 통해 군집화 결과를 보도록 함

  * coefficient threshold : 0.95

Visualization

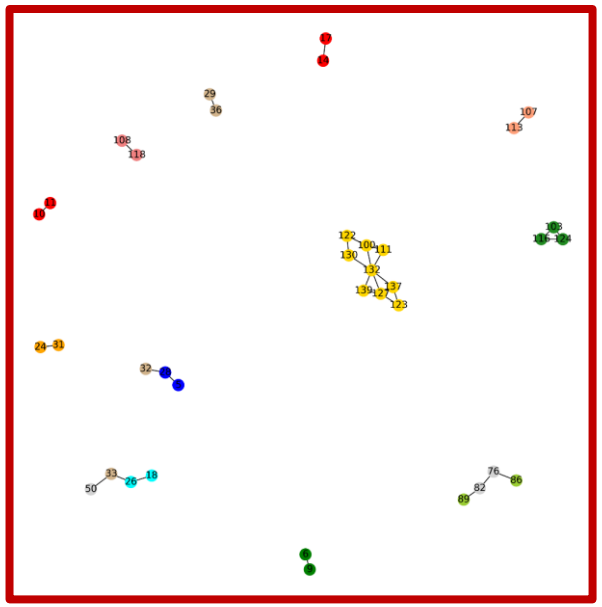
Eigen Value

[Num edges : 270]



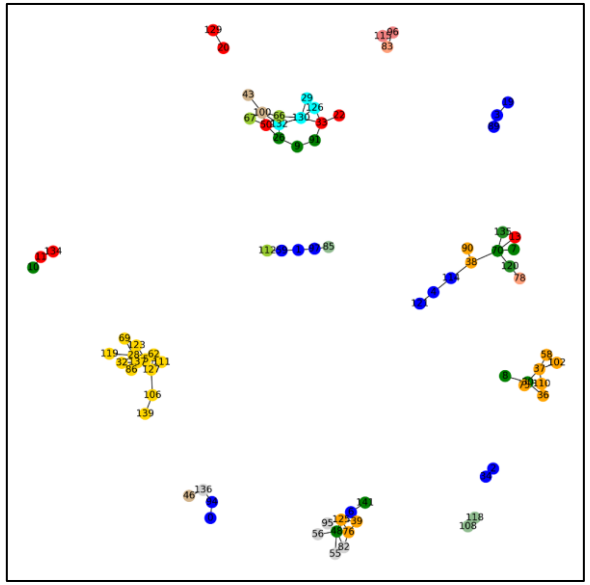
[Original]

[Num edges : 62]



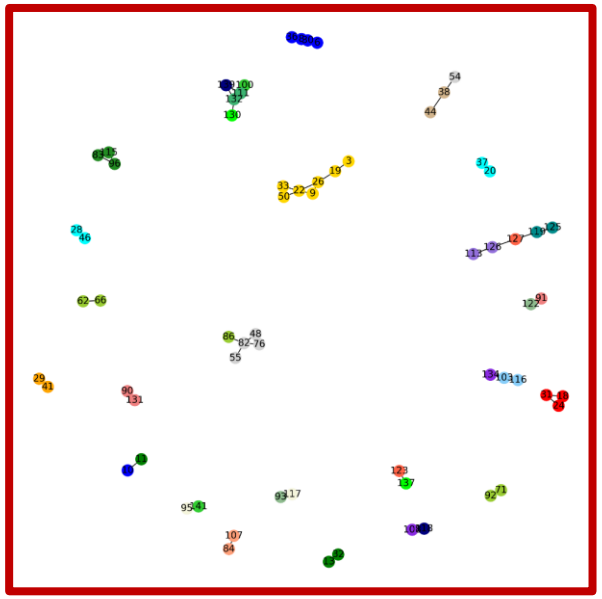
[MICE]

[Num edges : 224]



[MEAN]

[Num edges : 104]



[GRAPE]

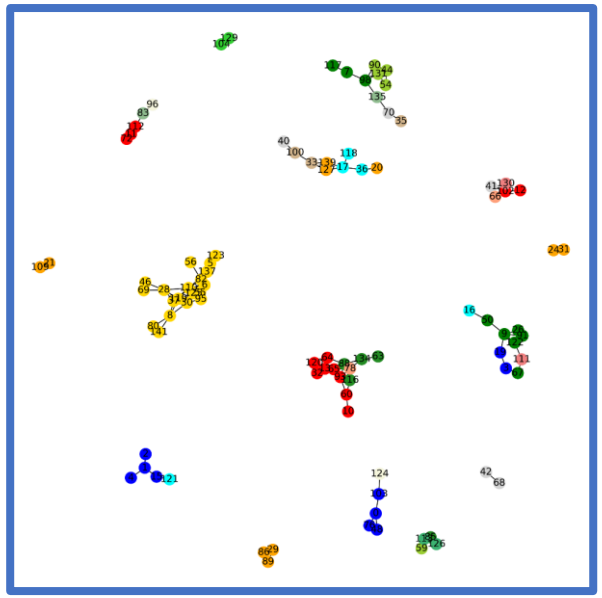
✓ GRAPE, MICE 의 경우 원본데이터에 비해 edge 수가 상당히 줄어듬

- 이웃의 속성이 상당히 유사한 노드들만 살아남음
- 특히, MICE 의 경우 데이터의 수가 많을 때, 일반적으로 효과가 좋다고 알려져 있으나, 본 데이터 셋의 총 Node 수는 142개로 비교적 적음
  그렇기 때문에, 편향된 데이터 분포에 맞게 Imputation 이 진행되고, 상관계수 계산시 값이 낮게 산출 된 것으로 해석함

Visualization

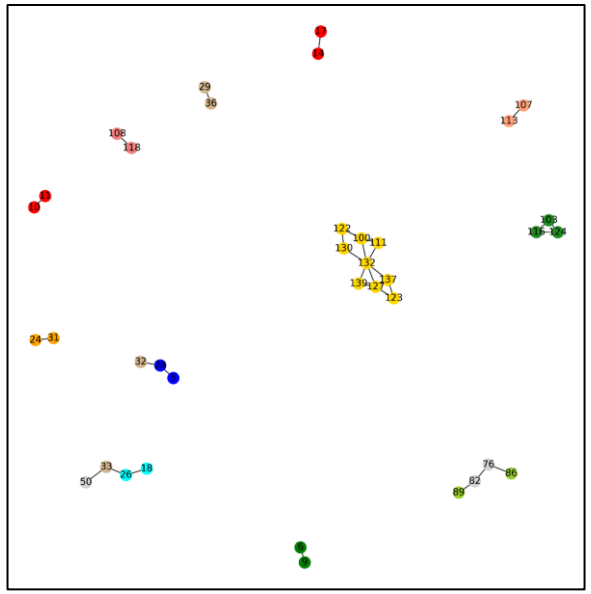
Eigen Value

[Num edges : 270]



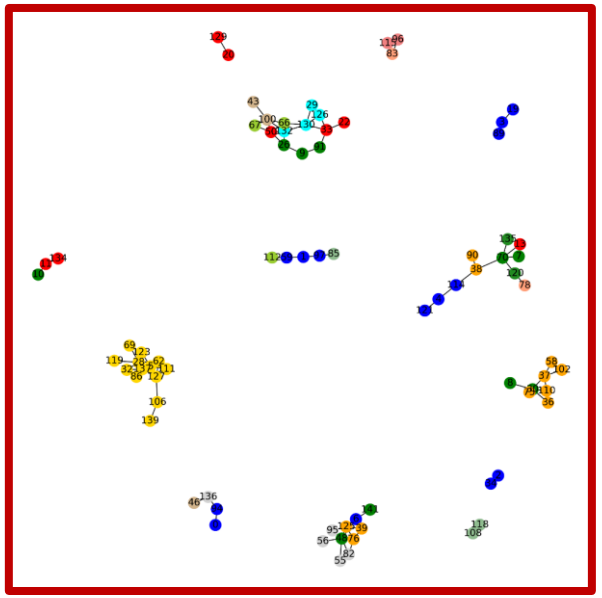
[Original]

[Num edges : 62]



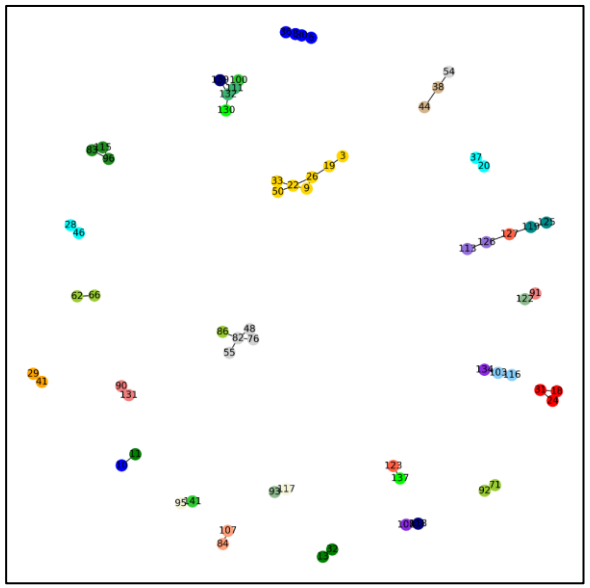
[MICE]

[Num edges : 224]



[MEAN]

[Num edges : 104]



[GRAPE]

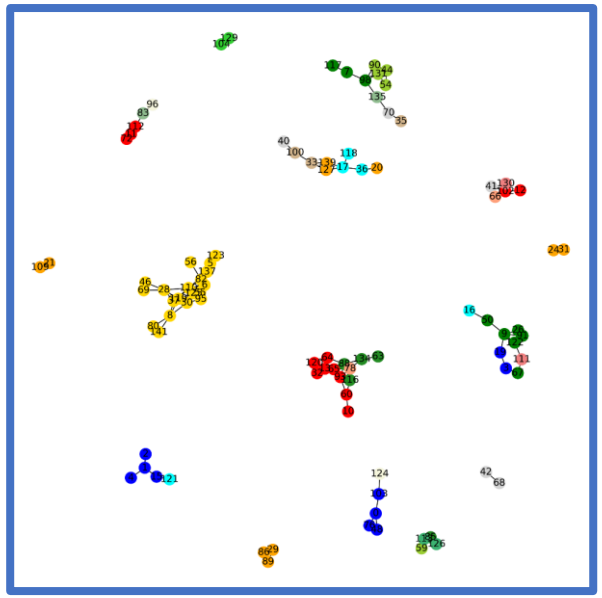
✓ MEAN Imputation

- Blanked Ratio 가 0.3 으로 적은 Missing value 를 예측 시, 평균값은 일반적으로 성능이 좋다고 함
- 위 경우, 원본 데이터와 추출된 edge 가 상당히 유사함을 알 수 있었고, 시각화 결과도 유사함
- 다만, 좀 더 많은 데이터(다양한 분포), Missing Ratio 가 높을 수록 동일한 값으로의 예측은 원본과 달라질 수 있을 것임

Visualization

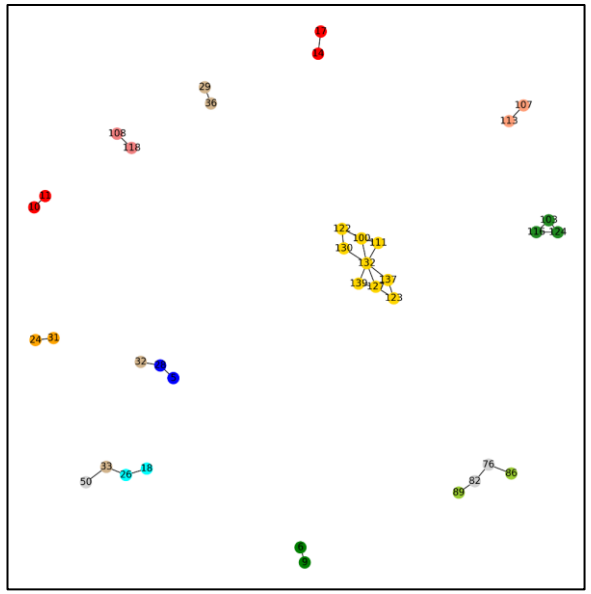
Eigen Value

[Num edges : 270]



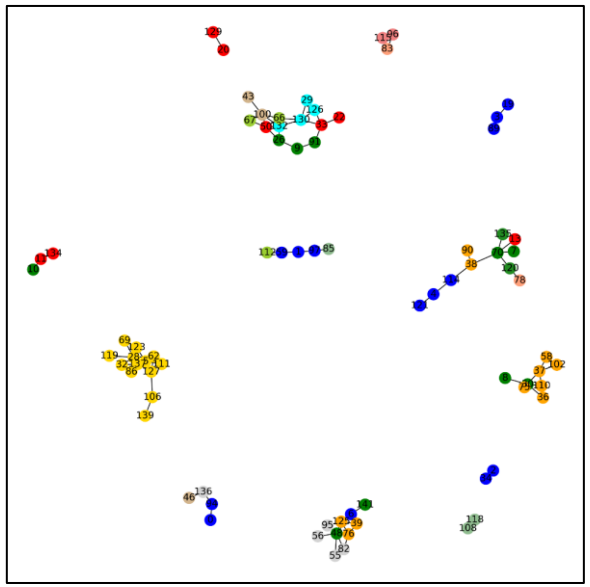
[Original]

[Num edges : 62]



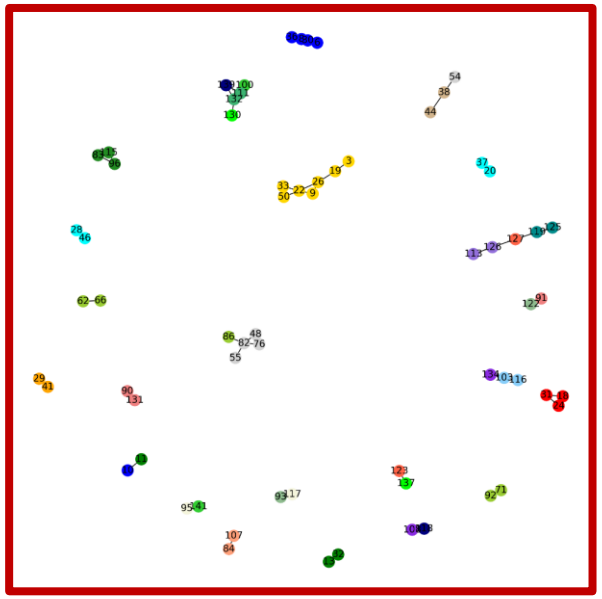
[MICE]

[Num edges : 224]



[MEAN]

[Num edges : 104]



[GRAPE]

✓ GRAPE Imputation

- 원본 데이터에 비해 edge 수가 많이 줄어들면서, 시각화 결과 또한 다른 결과가 보임
- GRAPE 는 상당히 다양한 값을 생성해 내며, 이 때 각 Node 의 특징을 최대한 반영하려 함
- 원본과 시각화 결과를 비교 했을 때, 각 군집이 edge 로 연결되어 있는 비율이 GRAPE 는 상당히 낮아지는 것을 확인 할 수 있음
- Graph 의 Homophily(동일 라벨,군집이 edge로 연결된 비율)가 낮아짐으로써, Classification 같은 downstream task 에서 좋은 결과를 보임

Thank You!