Contents

Course Structure & Current Focus

- Module 1: High-Dimensional Data (현재 단원: 데이터 압축과 시각화)
  - 1.1 The Curse of Dimensionality ($p \gg n$의 공포)
  - 1.2 PCA via SVD (효율적인 선형 압축)
  - 1.3 MDS (거리 보존 시각화)
  - 1.4 t-SNE (비선형 구조 시각화)
- Module 2: Clustering & Networks (군집과 연결)
- Module 3: Time Series & Spatial Data (시간과 공간)

Chapter 1

Module 1. 고차원 데이터 분석과 시각화

이전의 기초 통계학이나 머신러닝 기초 과정에서는 "변수 3개, 데이터 1000개" 같은 예쁜 데이터를 다뤘습니다. 하지만 현실(특히 유전체학, 텍스트 마이닝)은 "변수 2만 개, 데이터 50개" 같은 괴상한 형태입니다. 통계적 추론이 불가능해 보이는 이 상황에서, 우리는 데이터를 **압축**하여 눈으로 **확인(Visualization)**하는 것부터 시작합니다.

☐ 개요 (Overview)

이 단원에서는 고차원 데이터가 가지는 희소성 문제(차원의 저주)를 이해하고, 이를 해결하기 위한 3가지 핵심 도구를 배웁니다. 분산을 보존하는 **PCA(SVD 기반)**, 거리를 보존하는 **MDS**, 그리고 복잡하게 꼬인 비선형 구조를 풀어내는 **t-SNE**를 통해 데이터를 2차원 화면에 시각화하는 방법을 마스터합니다.

☐ 핵심 용어 사전

용어 (Term)	직관적 의미 (Meaning)
Curse of Dimensionality	차원이 늘어나면 공간이 너무 넓어져서, 데이터가 텅텅 비게 되는 현상.
SVD (Singular Value Decomposition)	행렬 분해의 끝판왕. PCA를 가장 빠르고 안정적으로 계산하는 도구.
MDS (Multi-Dimensional Scaling)	"서울-부산 거리 400km" 같은 거리표만 주고 지도를 그려내는 기술.
t-SNE	고차원에서 뭉쳐있는 데이터들을, 저차원에서도 겹치지 않게 잘 펼쳐주는 시각화 알고리즘.

—

1.0.1 1. 차원의 저주 (Curse of Dimensionality)

〔개념〕 개념 1: 사막에서 바늘 찾기

한 줄 요약: 변수(p)가 늘어날수록 공간의 부피는 기하급수적으로 늘어나고, 데이터 포인트들 사이의 거리는 의미가 없어질 정도로 멀어집니다.

직관적 비유

- **1차원(선):** 10미터 선 위에 동전 10개를 놓으면 꽉 찹니다.
- **2차원(면):** 10×10 방바닥에 동전 10개를 놓으면 듬성듬성합니다.
- **100차원:** 우주 공간 같은 허공에 동전 10개가 떠다닙니다. 서로 만날 일이 없습니다.
- **결과:** "가깝다/멀다"의 개념이 붕괴하여, 유클리드 거리 기반 알고리즘(KNN, K-Means)이 작동하지 않습니다.

—

1.0.2 2. PCA의 심화: SVD 관점

〔개념〕 개념 2: 공분산 행렬을 만들지 마라

한 줄 요약: 변수가 10만 개일 때 공분산 행렬(10만 $\times$ 10만)을 만들면 컴퓨터가 멈춥니다. SVD를 쓰면 이를 우회하여 바로 주성분을 구할 수 있습니다.

1) SVD의 정의

모든 행렬 X $(n \times p)$는 다음과 같이 분해됩니다.

$$X = U\Sigma V^T$$

- **U (Left Singular Vectors):** 데이터 포인트들의 새로운 좌표 (Principal Component Scores).
- **Σ (Singular Values):** 대각 행렬. 각 축의 중요도(분산의 크기 σ).
- **V (Right Singular Vectors):** 각 변수(유전자, 픽셀)가 주성분에 기여하는 가중치 (Loadings).

2) PCA와의 관계

공분산 행렬 X^TX의 고유값 분해를 할 필요 없이, SVD의 V가 바로 주성분 방향(Eigenvectors)이 되고, Σ^2이 고유값(Eigenvalues)에 비례합니다.

$$\text{Eigenvalue } \lambda_i = \frac{\sigma_i^2}{n-1}$$

—

1.0.3 3. 다차원 척도법 (MDS)

〔개념〕 개념 3: 거리표로 지도 복원하기

한 줄 요약: 데이터의 절대 좌표는 몰라도 됩니다. "누가 누구랑 얼마나 먼가(Distance Matrix)"만 알면 지도를 그릴 수 있습니다.

1) 목표: Stress 최소화

고차원(원래 공간)에서의 거리 d_{ij}와 저차원(압축된 공간)에서의 거리 $||z_i - z_j||$를 최대한 비슷하게 만듭니다.

$$\text{Stress} = \sqrt{\sum_{i,j}(d_{ij} - ||z_i - z_j||)^2}$$

2) 활용 예시

유전자 서열 데이터는 좌표가 없습니다. 하지만 "유전자 A와 B는 90% 일치한다(거리가 가깝다)"는 알 수 있습니다. 이 **유사도(Similarity) 행렬**을 MDS에 넣으면 유전자 지도를 그릴 수 있습니다.

—

1.0.4 4. t-SNE (t-Distributed Stochastic Neighbor Embedding)

〔개념〕 개념 4: 구겨진 종이 펴기 & 거리두기

한 줄 요약: 고차원에서 '이웃'이었던 관계는 유지하되, 저차원으로 올 때 너무 빽빽해지는 것을 막기 위해 **t-분포(꼬리가 두꺼운 분포)**를 사용하여 점들을 밀어냅니다.

1) PCA/MDS의 한계

이들은 **선형(Linear)** 변환입니다. 스위스 롤(Swiss Roll)처럼 돌돌 말려있는 데이터는 펴지 못하고 찌그러뜨립니다.

2) t-SNE의 마법: Crowding Problem 해결

- **고차원:** 공간이 넓어서 데이터들이 여유롭게 퍼져 있습니다. (정규분포로 이웃 확률 계산)
- **저차원:** 공간이 좁아서 데이터들이 서로 겹치려고 합니다. (Crowding)
- **해결:** 저차원에서는 **t-분포**를 사용합니다. t-분포는 정규분포보다 꼬리(Tail)가 두꺼워서, 적당히 멀리 있는 점들을 더 멀리 밀어내는 성질이 있습니다. 덕분에 군집(Cluster) 간의 경계가 아주 뚜렷하게 시각화됩니다.

—

1.1 실전 시나리오: 넥슨 게임 로그 분석

〔시나리오〕 Scenario: 유저 행동 패턴 시각화

당신은 '메이플스토리' 데이터 분석가입니다. 500명의 VIP 유저가 수행한 10,000종류의 행동 로그(사냥, 채팅, 거래, 강화 등)를 분석해 유저 유형을 파악하고 싶습니다.

1. **데이터 상황:** $n = 500, p = 10,000$. 전형적인 고차원 데이터($p \gg n$).
2. **PCA 적용:**

- 전체 분산의 50%만 설명하는 데도 주성분 100개가 필요합니다.
- 2차원으로 그렸더니 모든 유저가 한 덩어리로 뭉쳐서 구분이 안 됩니다. (선형 변환의 한계)

3. **t-SNE 적용:**
   - 고차원 공간에서의 행동 유사도(확률)를 보존하며 2차원으로 압축합니다.
   - 결과: 화면에 3개의 뚜렷한 섬(Cluster)이 나타납니다.
4. **인사이트 발견:**
   - 섬 A: 채팅과 거래만 하는 '장사꾼 유저'
   - 섬 B: 특정 보스만 반복해서 잡는 '쌀먹 유저'
   - 섬 C: 모든 콘텐츠를 즐기는 '진성 유저'

—

1.2 자주 묻는 질문 (FAQ)

Q1. PCA와 MDS는 뭐가 다른가요? A. 입력이 다릅니다.

- **PCA:** 데이터 좌표 자체(X)가 필요합니다. 분산(Variance)을 보존합니다.
- **MDS:** 데이터 간의 거리 행렬(D)만 있으면 됩니다. 거리(Distance)를 보존합니다. (유클리드 거리를 쓰면 PCA와 결과가 같습니다.)

Q2. t-SNE 결과에서 클러스터 간의 거리가 의미가 있나요? A. 주의하세요! t-SNE는 국소적인 이웃 관계(Local Structure)만 보존하려고 노력합니다. 멀리 떨어진 클러스터 간의 거리나, 클러스터의 크기는 실제 데이터와 다를 수 있습니다. 시각적으로 분리되었다는 사실에만 집중해야 합니다.

Next Step: 데이터를 시각화하여 눈으로 확인했으니, 이제 본격적으로 통계적 모델링을 할 차례입니다. 다음 시간에는 비슷한 데이터끼리 묶는 **군집화(Clustering)**와, 데이터 간의 복잡한 연결 고리를 파악하는 **네트워크 분석(Network Analysis)**을 다룹니다.

〔요약〕 Module 1 핵심 요약

- **Curse of Dimensionality:** $p \gg n$이면 공간이 희소해져서 기존 방법이 안 통한다.
- **SVD:** PCA를 수행하는 가장 효율적인 계산법 ($X = U\Sigma V^T$).
- **MDS:** 좌표 없이 '거리(Distance)' 정보만으로 지도를 그린다.
- **t-SNE:** t-분포를 이용해 고차원의 복잡한 구조를 겹치지 않게 펼쳐주는 최고의 시각화 도구.

〔a4paper, 11pt〕book fontspec amsmath, amssymb, amsthm geometry graphicx xcolor hyperref booktabs array bm
left=25mm, right=25mm, top=30mm, bottom=30mm

Contents

Course Structure & Current Focus

- Module 1 (Part A): Dimensionality Reduction (시각화)
- Module 1 (Part B): Advanced Clustering (현재 단원: 구조적 군집화)
  - 1.5 Limits of K-Means (구형 가정의 한계)
  - 1.6 Spectral Clustering (그래프 컷과 라플라시안)
  - 1.7 Hierarchical Clustering (덴드로그램과 연결법)
- Module 2: Analysis of Networks (네트워크 과학)

Chapter 2

Module 1 (Part B). 고급 군집화 (Advanced Clustering)

지난 시간 t-SNE를 통해 복잡하게 꼬인 데이터를 눈으로 확인했습니다. 그런데 K-Means 알고리즘을 돌려보니, 눈으로 보기엔 분명히 다른 그룹인데 하나로 묶어버리는 실수를 범합니다. 왜 그럴까요? K-Means는 단순한 '거리'만 보기 때문입니다. 이제 우리는 **"친구의 친구는 친구다"**라는 **연결성(Connectivity)**을 이용해 복잡한 구조를 풀어내는 방법을 배웁니다.

□ 개요 (Overview)

이 단원에서는 K-Means의 기하학적 한계(볼록성 가정)를 극복하기 위해, 데이터를 그래프 관점에서 해석하는 **스펙트럴 클러스터링(Spectral Clustering)**과 데이터의 계층적 구조를 파악하는 **계층적 군집화(Hierarchical Clustering)**를 다룹니다. 특히 스펙트럴 클러스터링의 핵심인 **라플라시안 행렬**과 **고유값 분해**의 연결고리를 이해하는 것이 핵심입니다.

□ 핵심 용어 사전

용어 (Term)	직관적 의미 (Meaning)
Convex Cluster	공처럼 둥글고 꽉 찬 형태의 군집. (K-Means가 좋아하는 것)
Adjacency Matrix (W)	누가 누구랑 연결되었는지(유사한지) 나타내는 친구 관계표.
Laplacian Matrix (L)	$D - W$. 그래프의 구조적 성질을 담고 있는 핵심 행렬.
Graph Cut	그래프를 가위로 잘라서 그룹을 나누는 행위.
Dendrogram	데이터가 합쳐지는 순서를 기록한 족보(Tree) 그림.

—

2.0.1 1. K-Means의 한계 (The Convexity Assumption)

〔개념〕 개념 1: 둥근 구멍에 네모난 못

한 줄 요약: K-Means는 "모든 클러스터는 둥근 공 모양이다"라고 가정합니다. 초승달 모양이나 도넛 모양 데이터는 절대 구분하지 못합니다.

한계 상황

- **Two Moons:** 두 개의 초승달이 서로 맞물려 있는 경우. K-Means는 가운데를 수직으로 뚝 잘라서 반반씩 섞어버립니다.
- **Concentric Circles:** 계란 노른자와 흰자처럼 안팎으로 감싸는 경우. K-Means는 파이 조각처럼 잘라버립니다.
- **이유:** 유클리드 거리상으로는 흰자 끝과 끝보다, 흰자와 노른자가 더 가깝기 때문입니다. **연결성**을 무시한 결과입니다.

—

2.0.2 2. 스펙트럴 클러스터링 (Spectral Clustering)

〔개념〕 개념 2: 다리 끊기 (Graph Cut)
한 줄 요약: 데이터를 공간상의 점이 아니라 **'노드와 엣지로 이루어진 그래프'**로 봅니다. 그룹 간의 연결 다리를 끊어서(Cut) 섬을 만드는 것이 목표입니다.

1) 알고리즘 단계 (The Recipe)

1. **유사도 그래프 생성 (W):** 거리(d)가 가까우면 1, 멀면 0인 인접 행렬을 만듭니다. (또는 가우시안 커널 사용)
2. **라플라시안 행렬 계산 (L):** 그래프의 성질을 요약합니다.

$$L = D - W$$

(D: 차수 행렬, 각 노드가 친구가 몇 명인지 대각선에 적음)

3. **고유값 분해 (Spectral Embedding):** L의 고유벡터를 구합니다.
4. **K-Means 적용:** 고유벡터로 변환된 공간(Spectral Domain)에서는 초승달 모양이 둥근 공 모양으로 펴집니다. 여기서 K-Means를 돌립니다.

2) 왜 라플라시안(L)인가?

우리의 목표는 **"그룹 간 연결은 끊고(Min Cut), 그룹 내부는 뭉치게 하는"** 것입니다. 수학적으로 이 최적화 문제(Normalized Cut)를 풀면, 그 해답이 놀랍게도 **L 행렬의 두 번째로 작은 고유벡터(Fiedler Vector)**와 같다는 것이 증명되어 있습니다.

- 즉, 복잡한 그래프 자르기 문제를 **선형대수(행렬 분해)** 문제로 바꿔서 푸는 것입니다.

—

2.0.3 3. 계층적 군집화 (Hierarchical Clustering)

〔개념〕 개념 3: 족보 만들기 (Family Tree)
한 줄 요약: 처음엔 모두가 남남이었다가, 가장 친한 사람끼리 짝을 짓고, 그 커플끼리 또 합치면서 거대한 가족(트리)을 만듭니다.

1) 병합형 (Agglomerative) 방식

- 시작: 데이터 N개가 각각 1개의 클러스터.
- 반복: 가장 가까운 두 클러스터를 합침 ($N \to N-1 \to \cdots \to 1$).
- 결과: **덴드로그램(Dendrogram)**이라는 트리 구조가 나옵니다. 원하는 높이에서 자르면(K) 클러스터가 나뉩니다.

2) 연결법 (Linkage Methods): 거리의 정의

"점과 점 사이의 거리는 아는데, **그룹과 그룹 사이의 거리**는 어떻게 재나요?"

방법	정의 (Distance between clusters)	특징
Single	가장 가까운 멤버끼리의 거리 (최단)	길게 늘어진 뱀 모양 잘 찾음. 노이즈에 약함.
Complete	가장 먼 멤버끼리의 거리 (최장)	둥글고 컴팩트한 군집 형성.
Average	모든 멤버 간 거리의 평균	무난하고 안정적임.
Ward	합쳤을 때 **분산의 증가량**	가장 널리 쓰임. 크기가 고른 군집 선호.

—

2.1 실전 시나리오: 넥슨 게임 길드(Guild) 분석

〔시나리오〕 Scenario: 적대 길드와 동맹 길드 파악

당신은 '리니지'와 같은 MMORPG의 데이터 분석가입니다. 수천 개의 길드가 서로 전쟁하거나 동맹을 맺고 있습니다. 이 복잡한 정치 지형도를 그려야 합니다.

1. **데이터:** 길드 간 PK 횟수, 파티 사냥 횟수, 채팅 빈도.
2. **문제:** 단순히 "전투력이 비슷한 길드"끼리 묶는 것(K-Means)은 의미가 없습니다. 전투력이 달라도 서로 친하면 같은 편입니다. **"관계(Relation)"**가 중요합니다.
3. **스펙트럴 클러스터링 적용:**
    - **노드:** 각 길드.
    - **엣지(W):** 파티 사냥이 많으면 가중치 높음(친함), PK가 많으면 가중치 낮음(적대).
    - **라플라시안(L):** 그래프 구조 계산.
    - **결과:** 2차원 평면에 투영하니, 거대한 두 세력(공성측 vs 수성측)이 양극단으로 뚜렷하게 나뉩니다.
4. **인사이트:** "전투력은 낮지만 인맥으로 연결된 '제3세력'이 존재함"을 발견합니다.

—

2.2 자주 묻는 질문 (FAQ)

Q1. 스펙트럴 클러스터링은 만능인가요? A. 성능은 강력하지만 계산 비용이 비쌉니다. $N \times N$ 크기의 행렬을 만들고 고유값 분해를 해야 하므로, 데이터가 수만 개를 넘어가면($N > 10,000$) 매우 느려집니다. 이때는 Nyström 근사 같은 기법을 써야 합니다.

Q2. 덴드로그램은 어떻게 읽나요? A. 아래(Leaf)는 개별 데이터, 위(Root)는 전체 데이터입니다. 세로축은 **"합쳐질 때의 거리(비용)"**를 의미합니다. 세로선이 길다는 것은, 두 그룹이 아주 멀리 떨어져 있는데 억지로 합쳤다는 뜻입니다. 보통 이 긴 선을 자르는(Cut) 위치가 최적의 K가 됩니다.

Next Step: 우리는 데이터를 점(Point)이 아닌 연결된 구조(Graph)로 보기 시작했습니다. 그렇다면 이 '네트워크' 자체를 본격적으로 분석할 수는 없을까요? 누가 중심 인물(Hub)이고, 정보는 어떻게 전파될까요? 다음 **Module 2: 네트워크 분석 (Analysis of Networks)**에서 그래프 이론의 심화를 다룹니다.

〔요약〕 Module 1 (Part B) 핵심 요약

- **한계:** K-Means는 볼록(Convex)한 형태만 찾을 수 있다.
- **Spectral Clustering:** 데이터를 그래프로 보고, 라플라시안 행렬의 고유벡터를 이용해 비볼록 구조를 분리한다. (Graph Cut)
- **Hierarchical Clustering:** 데이터를 계층적 트리(Dendrogram)로 시각화한다.
- **Linkage:** 클러스터 간 거리를 정의하는 방법(Ward, Single, Complete)에 따라 결과가 달라진다.

〔a4paper, 11pt〕book fontspec amsmath, amssymb, amsthm geometry graphicx xcolor hyperref booktabs array bm
left=25mm, right=25mm, top=30mm, bottom=30mm

Contents

Course Structure & Current Focus

- Module 1 (Part A): Dimensionality Reduction (시각화)
- Module 1 (Part B): Clustering (구조 파악)
- Module 1 (Part C): Multiple Hypothesis Testing (현재 단원: 가짜 발견 제어)
  - 1.8 The Multiple Testing Problem (p-hacking의 위험)
  - 1.9 FWER & Bonferroni Correction (보수적 접근)
  - 1.10 FDR & Benjamini-Hochberg (실용적 접근)
- Module 2: Analysis of Networks (네트워크 과학)

Chapter 3

Module 1 (Part C). 다중 가설 검정 (Multiple Hypothesis Testing)

> 앞서 우리는 수만 개의 유전자 데이터나 고객 행동 로그를 군집화하여 의미 있는 그룹을 찾았습니다. 그런데 누군가 묻습니다. "이 그룹들이 진짜 의미가 있는 거야, 아니면 그냥 우연히 뭉친 거야?" 이 질문에 답하기 위해 우리는 $P < 0.05$라는 전통적인 기준을 넘어, 빅데이터 시대에 맞는 새로운 검증 기준을 세워야 합니다.

☐ 개요 (Overview)

이 단원에서는 변수가 많은 고차원 데이터에서 통계적 검정을 수행할 때 발생하는 **다중 검정의 문제(False Positive 폭증)**를 다룹니다. 이를 해결하기 위한 두 가지 주요 전략인 **FWER(본페로니 교정)**과 **FDR(벤자미니-호크버그 절차)**의 원리를 배우고, 연구 목적에 따라 어떤 방법을 선택해야 하는지 학습합니다.

☐ 핵심 용어 사전

용어 (Term)	직관적 의미 (Meaning)
Type I Error (α)	늑대가 없는데 "늑대다!"라고 외치는 실수. (거짓 양성)
Multiple Testing Problem	주사위를 수만 번 던지면, 6이 연속 10번 나오는 '기적'도 우연히 발생한다는 문제.
FWER (Family-Wise Error Rate)	"전체 실험 중 단 하나의 실수도 용납하지 않겠다."
Bonferroni Correction	유의 수준을 $1/N$로 나눠버리는 아주 엄격한 기준.
FDR (False Discovery Rate)	"건져 올린 물고기 중 쓰레기가 5% 정도 섞여 있어도 괜찮다."

—

3.0.1 1. 문제의 배경: 왜 $P < 0.05$로는 부족한가?

〔개념〕 개념 1: 로또 당첨의 함정

한 줄 요약: 한 번 시도해서 성공할 확률이 희박하더라도, 수만 번 시도하면 누군가는 반드시 성공합니다. 그걸 '실력'이라고 착각하면 안 됩니다.

1) 직관적 비유: 동전 던지기 대회

- 1명이 동전을 10번 던져서 전부 앞면이 나올 확률은 약 0.001 (0.1%)입니다. 매우 희박하죠.
- 하지만 **10,000명이 동시에 동전을 던진다면?**
- 확률적으로 약 **10명**은 동전 10개 모두 앞면이 나옵니다.
- **오류:** 연구자는 이 10명을 보고 "초능력자다!"라고 결론 내립니다. 하지만 이들은 그냥 **운 좋은 일반인(False Positive)**입니다.

2) 실제 문제 (유전체 분석)

- 유전자 20,000개를 검사합니다 ($m = 20,000$).
- 유의 수준 $\alpha = 0.05$를 적용합니다.
- 아무런 효과가 없어도(귀무가설 참), $20,000 \times 0.05 = 1,000$개의 유전자가 '유의미하다'고 나옵니다.
- 결과: **1,000개의 가짜 암 유전자 후보**가 발견됩니다. 연구비 낭비의 시작입니다.

—

3.0.2 2. 해결책 1: FWER 제어 (본페로니 교정)

〔개념〕 개념 2: 철통 보안 검색대

한 줄 요약: 단 하나의 위험물도 비행기에 태우지 않겠다며, 모든 승객을 현미경으로 검사하는 방식입니다. 안전하지만, 비행기를 탈 수 있는 사람이 거의 없습니다.

1) 본페로니 교정 (Bonferroni Correction)

전체 가설 중 **하나라도** 틀릴 확률(FWER)을 0.05로 맞추려면, 개별 가설의 기준을 N배 엄격하게 해야 합니다.

$$\alpha_{new} = \frac{\alpha}{m}$$

2) 계산 예시

유전자 20,000개를 검사할 때 ($\alpha = 0.05$):

$$P_{threshold} = \frac{0.05}{20,000} = 0.0000025$$

- **장점:** 가짜(False Positive)가 나올 확률이 거의 0에 가깝습니다. (확실한 것만 잡음)
- **단점:** 기준이 너무 높아서, **진짜 중요한 유전자(True Positive)까지 다 탈락**시킵니다. (검정력 Power 급락)

—

3.0.3 3. 해결책 2: FDR 제어 (벤자미니-호크버그)

〔개념〕 개념 3: 섞여도 괜찮아, 비율만 맞춰

한 줄 요약: 금을 캘 때 돌멩이가 좀 섞여 있어도, 금이 훨씬 많으면 성공입니다. "내가 찾았다고 주장한 것 중 가짜의 비율(FDR)"을 5%로 유지합니다.

1) 벤자미니-호크버그 (BH) 절차

매우 우아하고 실용적인 알고리즘입니다. P값들을 줄 세워서 동적으로 커트라인을 정합니다.
알고리즘 순서:

1. 모든 가설의 P값을 오름차순으로 정렬합니다: $P_{(1)} \le P_{(2)} \le \cdots \le P_{(m)}$.
2. 각 k번째 P값에 대해, 다음 기준선(Line) 아래에 있는지 확인합니다.

$$P_{(k)} \le \frac{k}{m} \times \alpha$$

(순위 k가 커질수록, 즉 뒤로 갈수록 기준을 널널하게 봐줍니다.)

3. 위 조건을 만족하는 **가장 큰 k**를 찾습니다.
4. 1등부터 k등까지 모든 가설을 "유의미하다"고 선언합니다.

2) 숫자 대입 계산 예시

총 5개의 가설을 검정했고 ($m = 5$), 목표 FDR $\alpha = 0.05$입니다. P값 결과: $[0.001, 0.009, 0.04, 0.045, 0.6]$

순위 (k)	P값 ($P_{(k)}$)	기준선 ($\frac{k}{5} \times 0.05$)	통과 여부 ($P \le$ 기준)
1	0.001	0.01	Pass (✓)
2	0.009	0.02	Pass (✓)
3	0.040	0.03	Fail (×)
4	0.045	0.04	Fail (×) - 잠깐!

(수정: 예시 수치를 BH 절차의 극적인 효과를 위해 재조정합니다)
재조정된 예시: $P = [0.001, 0.015, 0.025, 0.05, 0.6]$, $\alpha = 0.05$.

- $k = 1$: $0.001 \le 0.01$ (Pass)
- $k = 2$: $0.015 \le 0.02$ (Pass)
- $k = 3$: $0.025 \le 0.03$ (Pass) $\rightarrow$ 여기까지가 최대!
- $k = 4$: $0.05 > 0.04$ (Fail)

결론: 3번째 가설까지 모두 유의미하다고 인정합니다. (본페로니였다면 0.01 기준이라 $k = 1$만 통과했을 것입니다. BH 덕분에 2, 3번도 구제받았습니다.)

—

3.1 실전 시나리오: 신약 후보 물질 발굴

〔시나리오〕 Scenario: 수만 개의 화합물 스크리닝
당신은 제약회사의 데이터 과학자입니다. 10,000개의 화합물 중 암세포를 죽이는 효과가 있는 후보를 추려야 합니다.

1. **상황:** 10,000번의 실험을 통해 각각 P값을 얻었습니다.
2. **본페로니 적용 시:**
   - 기준: $0.05/10,000 = 0.000005$.
   - 결과: 후보 물질이 **2개** 나옵니다.
   - 평가: 너무 적습니다. 놓친 후보 중에 진짜 대박 신약이 있을 수 있습니다.
3. **FDR (BH) 적용 시:**
   - 목표: "후보 중 5% 정도는 꽝이어도 된다."
   - 결과: 후보 물질이 **150개** 나옵니다.
   - 평가: 이 150개를 가지고 2차 정밀 실험을 진행합니다. 가짜가 7 8개 섞여 있겠지만, 나머지 142개의 진짜 후보를 건졌으니 훨씬 이득입니다.

—

3.2 자주 묻는 질문 (FAQ)

Q1. 언제 FWER을 쓰고 언제 FDR을 쓰나요? A. 목적에 따라 다릅니다.

- **FWER (본페로니):** 최종 임상 시험처럼 **"단 하나의 실수도 치명적일 때"** 씁니다. (보수적)
- **FDR (BH):** 초기 탐색 단계, 유전체 분석, 추천 시스템 A/B 테스트처럼 **"많은 후보를 발굴하고 싶을 때"** 씁니다. (현대 데이터 과학의 표준)

Q2. P값을 정렬하는 것만으로 어떻게 오류가 제어되나요? A. 수학적으로 증명되어 있습니다. $y = x$ 그래프를 그렸을 때, P값들이 대각선 아래에 위치한다는 것은 "우연히 발생할 확률보다 더 드물게 발생했다"는 증거가 됩니다. BH 절차는 이 경계선을 동적으로 조정하여 가짜의 비율(Area)을 제어합니다.

Next Step: 우리는 고차원 데이터의 '시각화(Part A)', '군집화(Part B)', 그리고 '가설 검정(Part C)'까지 마쳤습니다. 이제 데이터 포인트들이 서로 독립적이지 않고 복잡하게 얽혀 있다면 어떨까요? 다음 **Module 2: 네트워크 분석 (Analysis of Networks)**에서는 그래프 이론을 통해 데이터 간의 관계를 분석합니다.

〔요약〕 Module 1 (Part C) 핵심 요약

- **다중 검정 문제:** 가설 검정을 많이 할수록 우연에 의한 가짜 발견(False Positive)이 폭증한다.

- **FWER (본페로니):** α/m. 매우 엄격하다. 진짜도 놓칠 위험이 크다.
- **FDR (벤자미니-호크버그):** 발견된 것 중 가짜의 비율을 제어한다. $P_{(k)} \le \frac{k}{m}\alpha$.
- **트렌드:** 빅데이터 분석에서는 너무 보수적인 FWER보다 실용적인 **FDR**을 주로 사용한다.

〔a4paper, 11pt〕book fontspec amsmath, amssymb, amsthm geometry graphicx xcolor hyperref booktabs array bm
left=25mm, right=25mm, top=30mm, bottom=30mm

Contents

Course Structure & Current Focus

- Module 1: High-Dimensional Data (데이터 시각화 및 군집화)
- Module 2: Analysis of Networks (현재 단원: 그래프 이론)
  - 2.1 Centrality Measures (누가 중요한가?)
  - 2.2 Community Detection (끼리끼리 뭉치기)
  - 2.3 Network Models (Random Graphs, SBM)
- Module 3: Time Series & Spatial Data

Chapter 4

Module 2. 그래프 중심성 (Centrality Measures)

Module 1에서 스펙트럴 클러스터링을 통해 데이터를 '그래프'로 바라보는 법을 배웠습니다. 이제 우리는 이 그래프 안으로 줌인(Zoom-in)합니다. 거대한 네트워크 안에서 가장 영향력 있는 노드(Key Player)는 누구일까요? 친구가 가장 많은 사람? 아니면 정보를 독점하는 사람? 상황에 따라 달라지는 '중요함'의 정의를 수학적으로 파헤칩니다.

☐ 개요 (Overview)

이 단원에서는 복잡한 네트워크에서 노드의 중요도를 평가하는 4가지 핵심 척도를 배웁니다. 양적 인기를 측정하는 **연결 중심성**, 흐름을 통제하는 **매개 중심성**, 질적 영향력을 보는 **고유벡터 중심성**, 그리고 이를 웹 환경에 맞게 발전시킨 **페이지랭크**까지 다룹니다.

☐ 핵심 용어 사전

용어 (Term)	직관적 의미 (Meaning)
Degree (차수)	한 노드에 연결된 엣지(다리)의 개수.
Shortest Path	노드 A에서 B로 가는 가장 빠른 길.
Adjacency Matrix (A)	그래프의 연결 상태를 0과 1로 나타낸 행렬.
Principal Eigenvector	행렬 A를 대표하는 가장 힘센 고유벡터.
Damping Factor (d)	웹 서퍼가 링크를 클릭하지 않고 딴청 피울 확률을 고려한 계수.

—

4.0.1 1. 연결 중심성 (Degree Centrality)

〔개념〕 개념 1: 마당발 (The Popular Kid)
한 줄 요약: "내 전화번호부에 친구가 몇 명 저장되어 있는가?" (양적 인기)

1) 정의 및 특징

$$C_D(v) = \deg(v) = \text{ of edges connected to } v$$

- **특징:** 가장 계산하기 쉽습니다.
- **한계:** **국소적(Local)** 정보만 봅니다. 내 친구 100명이 전부 '아싸'일 수도 있고, 친구 1명이 '대통령'일 수도 있는데, 이를 구분하지 못하고 똑같이 1명으로 칩니다.
- **활용:** 연예인의 트위터 팔로워 수, 허브 공항 찾기.

—

4.0.2 2. 매개 중심성 (Betweenness Centrality)

〔개념〕 개념 2: 문지기 (The Gatekeeper / Broker)
한 줄 요약: "도시 A에서 도시 B로 가는 최단 경로상에 내가 위치하는가?" (흐름 통제권)

1) 정의

네트워크 내의 모든 노드 쌍 (s, t)에 대하여, 최단 경로가 나(v)를 거쳐가는 비율입니다.

$$C_B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

(σ_{st}: $s \to t$ 최단 경로 총개수, $\sigma_{st}(v)$: 그중 v를 지나는 개수)

2) 특징 및 활용

- **Bridge:** 친구가 딱 2명뿐이어도, 그 2명이 서로 다른 거대 그룹의 리더라면 매개 중심성은 폭발합니다.
- **활용:** 무역로의 길목(파나마 운하), 사내 부서 간 조율자, 스파이 네트워크의 연락책.
- **의미:** 이 노드를 제거하면 네트워크가 두 동강 날 확률이 높습니다.

—

4.0.3 3. 고유벡터 중심성 (Eigenvector Centrality)

〔개념〕 개념 3: 유유상종 (The Influencer)
한 줄 요약: "친구가 몇 명인지가 중요한 게 아니라, **누구랑** 친구인지가 중요하다." (질적 인기)

1) 재귀적 정의 (Recursive Definition)

나의 중요도(x_i)는 내 친구들의 중요도(x_j)의 합에 비례합니다.

$$x_i = \frac{1}{\lambda} \sum_{j \in \text{neighbors}(i)} x_j \quad \iff \quad Ax = \lambda x$$

이 식은 선형대수학의 **고유값 문제(Eigenvalue Problem)**와 정확히 일치합니다.

2) 특징

중요하지 않은 사람 100명과 아는 것보다, 대통령 1명과 아는 것이 더 높은 점수를 받습니다.

- **활용:** 학계에서의 평판(노벨상 수상자가 인용해 줌), 사교계의 명사.

—

4.0.4 4. 페이지랭크 (PageRank)

〔개념〕 개념 4: 구글을 만든 알고리즘 (The Authority)
한 줄 요약: 고유벡터 중심성을 **방향성 그래프(웹)**에 맞게 수정한 버전. "권위 있는 페이지가 링크를 걸어주면 나도 권위가 생긴다."

1) 문제 상황: 스파이더 트랩 (Spider Trap)

웹(Web)은 링크가 한쪽으로만 가는 방향성 그래프입니다. 링크를 받기만 하고 내보내지 않는 페이지(Sink)가 있으면, 고유벡터 중심성 점수가 그곳으로 다 빨려 들어가서 다른 노드들이 0점이 됩니다.

2) 해결책: 랜덤 서퍼 모델 (Random Surfer)

웹 서퍼가 링크를 따라가다가, 지루해지면 주소창에 새로운 주소를 쳐서 **순간이동(Teleport)**한다고 가정합니다.

$$PR(A) = (1-d) + d \sum_{i \in \text{InLink}(A)} \frac{PR(T_i)}{C(T_i)}$$

- **d (Damping Factor):** 보통 0.85. (85%는 링크 클릭, 15%는 딴 데로 점프)
- **$C(T_i)$:** 나를 추천해 준 페이지가 링크를 100개나 달아놨다면, 내가 받는 추천 효력은 $1/100$로 희석됩니다.
- **결과:** 막다른 골목에 갇히지 않고 전 세계 웹페이지의 중요도 순위를 매길 수 있습니다.

—

4.1 실전 시나리오: 넥슨 게임 길드 네트워크 분석

〔시나리오〕 Scenario: 길드장과 숨은 실세 찾기
당신은 MMORPG 게임의 커뮤니티 매니저입니다. 길드 간의 채팅 로그를 분석해 여론을 주도하는 유저를 찾고 싶습니다.

1. **Degree (연결 중심성):**
    - 채팅을 제일 많이 하고, 아는 사람이 많은 유저.
    - **판단:** "그냥 수다쟁이일 수도 있음. 정보의 가치는 낮을 수 있음."
2. **Eigenvector (고유벡터 중심성):**
    - 길드장들이나 랭커들과 친하게 지내는 유저.

- **판단:** "**진짜 실세(비선 실세)**일 확률 높음. 이 사람을 잡으면 상위권 여론 파악 가능."

3. **Betweenness (매개 중심성):**
   - A 길드와 적대 관계인 B 길드 양쪽 모두와 귓속말을 하는 유저.
   - **판단:** "**이중 간첩**이거나 평화 협상가. 전쟁 발발의 열쇠를 쥐고 있음."

—

4.2 자주 묻는 질문 (FAQ)

Q1. 고유벡터 중심성과 페이지랭크의 결정적 차이는 뭔가요? A. 방향성과 나눔(Sharing)입니다.

- 고유벡터: 친구가 힘이 세면 나도 무조건 힘이 셉니다.
- 페이지랭크: 친구가 힘이 세도, 그 친구가 여기저기 다 추천하고 다니면(링크 남발) 내가 받는 힘은 $1/N$로 줄어듭니다.

Q2. 중심성이 높으면 무조건 좋은가요? A. 목적에 따라 다릅니다. 바이러스를 막으려면 '매개 중심성'이 높은 허브를 격리해야 하고, 마케팅 소문을 내려면 '고유벡터 중심성'이 높은 인플루언서를 공략해야 합니다.

Next Step: 개별 노드의 중요도를 파악했습니다. 이제 시야를 넓혀서, 이 노드들이 뭉쳐서 형성하는 **거대한 덩어리(Community)**를 찾아볼까요? 다음 시간에는 네트워크 버전의 군집화인 **커뮤니티 탐지(Community Detection)**를 배웁니다.

〔요약〕 Module 2 핵심 요약

- **Degree:** 이웃 수. (Local Popularity)
- **Betweenness:** 최단 경로 통과 횟수. (Flow & Control)
- **Eigenvector:** 이웃의 중요도 반영. (Global Influence)
- **PageRank:** 방향성 그래프에서의 권위. 랜덤 서퍼 모델(Teleport)로 쏠림 현상 해결.

〔a4paper, 11pt〕book fontspec amsmath, amssymb, amsthm geometry graphicx xcolor hyperref booktabs array bm
left=25mm, right=25mm, top=30mm, bottom=30mm

Contents

Course Structure & Current Focus

- Module 2 (Part A): Centrality Measures (핵심 인물 찾기)
- Module 2 (Part B): Community Detection (현재 단원: 파벌 나누기)
  - 2.4 Definition of Community (내부는 빽빽, 외부는 듬성)
  - 2.5 Modularity Q (랜덤보다 얼마나 더 뭉쳤나?)
  - 2.6 Spectral Partitioning (피들러 벡터를 이용한 절단)
- Module 2 (Part C): Network Models (그래프 생성 모델)

Chapter 5

Module 2 (Part B). 커뮤니티 탐지 (Community Detection)

> 지난 시간(Part A)에는 "이 네트워크에서 가장 중요한 인싸는 누구인가?"를 찾았습니다. 이제 시야를 넓혀봅시다. 인싸 주변에는 그를 따르는 무리가 있고, 네트워크는 이런 무리(Community)들의 집합으로 이루어져 있습니다. 이번 시간에는 "네트워크를 어떻게 쪼개야 자연스러운 파벌(Clique)이 드러나는가?"를 수학적으로 정의합니다.

☐ 개요 (Overview)

커뮤니티 탐지는 그래프의 노드들을 "내부 연결은 강하고 외부 연결은 약한" 그룹으로 나누는 기술입니다. 이를 위해 **모듈성(Modularity)**이라는 객관적인 점수표를 도입하고, 계산 난이도가 높은 그래프 분할 문제를 해결하기 위해 **스펙트럴 파티셔닝(피들러 벡터)**이라는 선형대수학적 해법을 배웁니다.

☐ 핵심 용어 사전

용어 (Term)	직관적 의미 (Meaning)
Community	끼리끼리 뭉친 그룹. (학교의 '반'이나 '동아리')
Null Model	"만약 아무런 규칙 없이 무작위로 연결했다면?" (비교 기준)
Modularity (Q)	(실제 연결) - (무작위 연결). 이 값이 클수록 끈끈한 그룹임.
Graph Cut	그래프를 가위로 잘라 두 덩어리로 만드는 것.
Fiedler Vector	그래프를 가장 예쁘게 자르는 방법을 알려주는 마법의 벡터 (λ_2).

—

5.0.1 1. 커뮤니티의 정의와 목표

〔개념〕 개념 1: 학교 식당 풍경

한 줄 요약: 우리끼리는 시끌벅적하게 떠들고(내부 연결 ↑), 옆 테이블이랑은 가끔 눈인사만 하는(외부 연결 ↓) 상태입니다.

1) 직관적 정의

좋은 커뮤니티 구조란 다음 두 조건을 만족해야 합니다.

- **Intra-cluster density:** 같은 그룹 내의 노드끼리는 엣지가 많아야 함.
- **Inter-cluster density:** 다른 그룹 간의 노드끼리는 엣지가 적어야 함.

2) 왜 어려운가?

단순히 "자른 엣지 개수(Cut Size)"를 최소화하려 하면, 그래프 전체에서 노드 딱 하나만 떼어내는 것이 수학적으로 최적이 되어버립니다. (이를 방지하기 위해 정규화된 컷(Normalized Cut)이나 모듈성을 사용합니다.)

—

5.0.2 2. 모듈성 (Modularity, Q)

〔개념〕 개념 2: 우연을 가장한 필연 찾기
한 줄 요약: "너네 진짜 친해서 뭉친 거야, 아니면 좁은 방에 있다 보니 우연히 옆에 선 거야?"를 판별하는 점수입니다.

1) 핵심 아이디어

우리가 찾은 그룹이 진짜 의미가 있으려면, **"랜덤하게 섞었을 때(Null Model) 기대되는 연결"**보다 훨씬 더 많이 연결되어 있어야 합니다.

2) 수식과 해석

$$Q = \frac{1}{2m} \sum_{i,j} \left(\underbrace{A_{ij}}_{\text{실제 연결}} - \underbrace{\frac{k_i k_j}{2m}}_{\text{기대 연결}} \right) \delta(c_i, c_j)$$

- A_{ij}: 실제 노드 i, j가 연결되었으면 1, 아니면 0.
- $\frac{k_i k_j}{2m}$: 노드 i와 j가 우연히 연결될 확률. (인기 많은 애들(k가 큼)끼리는 우연히 만날 확률도 높음).
- $\delta(c_i, c_j)$: i와 j가 같은 팀으로 배정되었을 때만 계산함.

3) 계산 예시 (간단 버전)

총 엣지 수 $m = 10$인 네트워크에서, 노드 A(친구 4명)와 노드 B(친구 5명)가 같은 그룹에 있다고 합시다.

- **실제(A_{AB}):** 둘이 연결됨 (값 = 1).
- **기대(E):** $\frac{4 \times 5}{2 \times 10} = \frac{20}{20} = 1$.
- **기여도:** $1 - 1 = 0$. (인기인끼리 연결된 건 당연한 거라 점수 안 줌)
- 만약 노드 C(친구 1명), D(친구 1명)가 연결되었다면?

- **기대(E):** $\frac{1\times1}{20} = 0.05$.
- **기여도:** $1 - 0.05 = 0.95$. (아싸끼리 연결된 건 운명이므로 높은 점수!)

—

5.0.3 3. 스펙트럴 파티셔닝 (Spectral Partitioning)

〔개념〕 개념 3: 그래프를 수직선 위에 줄 세우기
한 줄 요약: 복잡한 얽타래(그래프)를 1차원 직선 위에 쫙 펴서 나열한 뒤, 사이가 가장 먼 곳을 가위로 싹둑 자릅니다.

1) 피들러 벡터 (Fiedler Vector)

그래프 라플라시안 행렬 ($L = D - A$)의 **두 번째로 작은 고유값(λ_2)에 해당하는 고유벡터(v_2)**입니다.

- **역할:** 각 노드에 실수 값 하나씩을 부여합니다 ($v_2 = [0.1, 0.2, -0.5, \dots]$).
- **성질:** 서로 강하게 연결된 노드들은 피들러 벡터 값이 서로 비슷합니다.

2) 작동 방식

1. L의 고유벡터 v_2를 구합니다.
2. 노드들을 v_2 값의 크기순으로 정렬합니다.
3. 값의 부호가 바뀌는 지점(0)이나, 차이가 가장 큰 지점(Gap)을 기준으로 두 그룹으로 나눕니다.

3) 치거 부등식 (Cheeger's Inequality)

$$\text{자르기 힘든 정도} \approx \lambda_2$$

λ_2가 0에 가까우면 그래프가 이미 두 덩어리로 거의 나뉘어 있다는 뜻이고(자르기 쉬움), 크면 아주 끈끈하게 뭉쳐있다는 뜻입니다.

—

5.1 실전 시나리오: 불법 작업장 탐지

〔시나리오〕 Scenario: 넥슨 게임 내 '작업장' 길드 색출
당신은 보안팀 데이터 분석가입니다. 일반 유저 길드와 아이템 파밍용 봇(Bot) 길드(작업장)를 구분해야 합니다.

1. **네트워크 구성:** 유저=노드, 아이템/골드 거래=엣지.
2. **가설:**
    - **일반 길드:** 서로서로 복잡하게 거래함 (그물망 구조). 모듈성 Q가 높음.
    - **작업장:** 수백 개의 봇이 하나의 '창고 계정'으로만 골드를 보냄 (성게 모양, Star Structure).

3. **커뮤니티 탐지 수행 (Louvain or Spectral):** 전체 유저 네트워크를 커뮤니티 단위로 쪼갭니다.
4. **모듈성 분석:** 탐지된 커뮤니티 중, 내부 연결 패턴이 '성게 모양'이면서 모듈성 기여도가 비정상적인 그룹을 찾아냅니다.
5. **결과:** 500개의 작업장 계정을 일망타진합니다.

—

5.2 자주 묻는 질문 (FAQ)

Q1. Q값이 몇 이상이어야 좋은 건가요? A. 절대적인 기준은 없지만, 보통 $0.3 \sim 0.7$ 사이면 뚜렷한 커뮤니티 구조가 있다고 봅니다. Q가 너무 작으면 랜덤 그래프와 다를 바 없고, 이론적 최댓값은 1입니다.

Q2. 왜 두 번째 고유벡터(v_2)를 쓰나요? 첫 번째는요? A. 라플라시안 행렬 L의 가장 작은 고유값 λ_1은 항상 0이고, 그 벡터 v_1은 $[1, 1, \ldots, 1]$입니다. 이는 "모든 노드는 하나의 그래프에 있다"는 뻔한 사실을 말해주므로 정보가가 없습니다. 그래서 그 다음으로 작은 λ_2 (피들러 값)가 실질적인 분할 정보를 줍니다.

Next Step: 지금까지는 "주어진 그래프"를 분석했습니다. 그런데 이 그래프(소셜 네트워크, 인터넷 등)는 도대체 어떤 원리로 생성된 걸까요? 다음 시간에는 랜덤 그래프, 작은 세상 네트워크 등 **네트워크 생성 모델(Network Models)**을 통해 세상의 연결 법칙을 배웁니다.

〔요약〕 Module 2 (Part B) 핵심 요약

- **Community:** 내부 밀도가 높고 외부 밀도가 낮은 노드 집합.
- **Modularity (Q):** 실제 연결과 랜덤 연결(Null Model)의 차이. 군집화의 채점표.
- **Spectral Partitioning:** 라플라시안 행렬의 피들러 벡터(v_2)를 이용해 그래프를 수학적으로 최적 절단(Cut)하는 기법.

〔a4paper, 11pt〕book fontspec amsmath, amssymb, amsthm geometry graphicx xcolor hyperref booktabs array bm
left=25mm, right=25mm, top=30mm, bottom=30mm

Contents

Course Structure & Current Focus

- Module 2 (Part A): Centrality Measures (중심 인물 찾기)
- Module 2 (Part B): Community Detection (그룹 찾기)
- Module 2 (Part C): Network Models (현재 단원: 네트워크의 기원)
  - 2.7 Erd□s-R□nyi (무작위 그래프)
  - 2.8 Stochastic Block Models (SBM - 끼리끼리)
  - 2.9 Small-World & Power Law (현실의 네트워크)
- Module 3: Time Series & Spatial Data (시공간 데이터)

Chapter 6

Module 2 (Part C). 네트워크 생성 모델 (Network Models)

우리는 지난 시간에 '모듈성(Q)'을 계산할 때, "무작위로 연결된 네트워크(Null Model)"와 비교했습니다. 그 무작위 모델이 바로 이번에 배울 에르되시-레니 모델입니다. 하지만 현실은 무작위가 아닙니다. 현실은 "끼리끼리 뭉치고(SBM)", "친구의 친구를 알고(Small-world)", "유명한 사람이 다 독식하는(Power Law)" 세상입니다. 이 현상들을 수학적으로 모델링해 봅니다.

□ 개요 (Overview)

이 단원에서는 네트워크가 생성되는 3가지 메커니즘을 배웁니다. 1. 모든 것이 **우연(Random)**이라고 가정하는 기준점(Erd□s-R□nyi). 2. **그룹(Block)**이 존재한다고 가정하는 SBM. 3. 현실 세계의 **부익부 빈익빈(Hub)** 현상을 설명하는 척도 없는 네트워크(Scale-free)와 멱법칙(Power Law).

□ 핵심 용어 사전

용어 (Term)	직관적 의미 (Meaning)
Erd□s-R□nyi ($G(n,p)$)	주사위를 던져서 연결 여부를 정하는 완전 랜덤 그래프.
SBM	"같은 반끼리는 친하고 다른 반이랑은 덜 친해"를 확률로 정의한 모델.
Degree Distribution	"친구가 k명인 사람은 전체의 몇 %인가?"를 그린 그래프.
Power Law	"상위 1%가 전체 연결의 80%를 차지한다." (부익부 빈익빈)
Small-World	"여섯 다리만 건너면 대통령과도 아는 사이." (좁은 세상)

—

6.0.1 1. 에르되시-레니 모델 (Erd□s-R□nyi Model)

〔개념〕 개념 1: 눈 감고 선 긋기

한 줄 요약: 모든 사람은 평등하고, 누구나 서로 친구가 될 확률(p)이 똑같습니다. 인싸도 아싸도 없는 평평한 세상입니다.

1) 정의 $G(n, p)$

n개의 노드에 대해, 모든 가능한 쌍이 확률 p로 연결됩니다.

- **특징:** 허브(Hub)가 없습니다. 모든 노드의 친구 수(Degree)가 평균($\lambda = np$) 근처에 몰려 있습니다.
- **분포:** 친구 수(k)의 분포는 **푸아송 분포(Poisson Distribution)**를 따릅니다. 종 모양의 정규분포와 비슷합니다.

2) 한계와 의의

현실 세계(예: 인스타 팔로워)는 수백만 명을 거느린 스타가 존재하므로 이 모델과 맞지 않습니다. 하지만, **"우연히 발생한 패턴인지 아닌지"**를 검증할 때(가설 검정의 Null Hypothesis) 아주 중요한 기준점이 됩니다.

—

6.0.2 2. 스토캐스틱 블록 모델 (SBM)

〔개념〕 개념 2: 유유상종의 수학화
한 줄 요약: 사람들을 몇 개의 그룹(Block)으로 나누고, "같은 그룹끼리 연결될 확률(p_{in})"을 "다른 그룹과 연결될 확률(p_{out})"보다 높게 설정합니다.

1) 구조

에르되시-레니 모델에 **'커뮤니티 구조'**를 심은 것입니다.

- **생성 원리:** 먼저 노드가 속할 그룹을 정하고, 그 그룹 정보에 따라 확률적으로 엣지를 연결합니다.
- **활용:** 지난 시간에 배운 **커뮤니티 탐지(Community Detection)** 알고리즘들은 "이 네트워크가 SBM으로 만들어졌다면, 원래 그룹은 무엇이었을까?"를 역추적하는 과정입니다.

—

6.0.3 3. 작은 세상과 멱법칙 (Small-World & Power Law)

〔개념〕 개념 3: 좁은 세상 속 거인들
한 줄 요약: 현실 네트워크는 끼리끼리 뭉쳐있으면서도(Small-World), 소수의 거인(Hub)이 전체를 연결하는 불평등한 구조(Power Law)입니다.

A. 작은 세상 효과 (Small-World Effect)

- **Watts-Strogatz 모델:** 규칙적인 격자(내 옆 사람하고만 친구)에서 몇 개의 선을 랜덤하게 반대편으로 연결(Rewiring)합니다.
- **특징:**
  1. **높은 클러스터링:** 내 친구끼리는 서로 친구일 확률이 높음.
  2. **짧은 경로:** 몇 단계만 거치면 지구 반대편 사람과도 연결됨. (케빈 베이컨의 법칙)

B. 멱법칙 분포 (Power Law Distribution)

〔개념〕 핵심: 척도 없는 네트워크 (Scale-free)
친구 수가 k일 확률 $P(k)$가 $k^{-\gamma}$에 비례합니다. 그래프를 그리면 L자 모양(Long Tail)이 나옵니다.

- **부익부 빈익빈 (Preferential Attachment):** 새로운 노드가 들어올 때, 이미 친구가 많은 '인싸'에게 연결될 확률이 더 높습니다. (Barab□si-Albert 모델)
- **Scale-free의 의미:** 평균(Mean)이 의미가 없습니다. 평균 팔로워가 500명이라도, 1억 명 팔로워를 가진 사람이 평균을 왜곡합니다. 어떤 척도(Scale)로 봐도 모양이 비슷합니다(Fractal).

—

6.1 실전 시나리오: 넥슨 미디어 파트너십 전략

〔시나리오〕 Scenario: 신작 게임 마케팅 예산 분배
당신은 본부장으로서 한정된 마케팅 예산을 인플루언서들에게 분배해야 합니다. 네트워크 모델을 이용해 전략을 짭니다.

1. **랜덤 전략 (Erd□s-R□nyi):** 무작위로 100명의 유저에게 쿠폰을 뿌립니다. → **실패.** 전파력이 약합니다. 다들 고만고만한 친구들뿐이라 확산이 안 됩니다.
2. **허브 타겟팅 (Power Law):** 팔로워 수 상위 1%인 '대형 스트리머' 3명에게 몰빵합니다. → **성공 (인지도 확보).** 멱법칙에 의해 이들이 전체 트래픽의 80%를 장악하고 있으므로, 단기간에 엄청난 노출 효과를 얻습니다.
3. **브릿지 타겟팅 (Small-World):** RPG 커뮤니티와 FPS 커뮤니티 사이를 연결하는 '장르 파괴자' 스트리머를 공략합니다. → **성공 (유저 확장).** 끼리끼리 뭉친(High Clustering) 그룹 사이를 넘어가는 지름길(Short Path) 역할을 하여, RPG 유저를 FPS 게임으로 데려옵니다.

—

6.2 자주 묻는 질문 (FAQ)

Q1. 내 데이터가 멱법칙을 따르는지 어떻게 아나요? A. **Log-Log Plot**을 그려보세요. 가로축(Degree)과 세로축(Frequency)을 모두 로그 스케일로 그렸을 때, 데이터가 **우하향하는 직선** 형태를 띠면 멱법칙을 따르는 것입니다. (직선의 기울기가 $-\gamma$가 됩니다.)

Q2. SBM과 커뮤니티 탐지는 같은 건가요? A. 동전의 양면입니다.

- **SBM:** "확률 P를 줄 테니 그래프를 만들어봐." (생성 모델)
- **커뮤니티 탐지:** "그래프를 줄 테니 확률 P와 그룹을 찾아내봐." (추론)

Next Step: 지금까지는 데이터의 '구조'와 '연결'을 봤습니다. 하지만 데이터는 멈춰있지 않습니다. 주식 가격은 변하고, 위성은 움직입니다. 다음 **Module 3: 시계열 및 공간 데이터 (Time Series Spatial Data)**에서는 시간(t)과 공간(x, y) 축을 따라 변하는 데이터를 다루는 법을 배웁니다.

〔요약〕 Module 2 (Part C) 핵심 요약

- **Erd□s-R□nyi:** 완전 랜덤 그래프. 허브가 없다. (Poisson 분포)
- **SBM:** 그룹 내 연결 확률이 더 높은 모델. 커뮤니티 구조 설명.
- **Small-World:** 끼리끼리 뭉치지만(Clustering), 몇 단계면 다 통한다(Short Path).
- **Power Law (Scale-free):** 소수의 허브가 지배하는 불평등한 구조. 현실 세계의 가장 강력한 모델.

〔a4paper, 11pt〕book fontspec amsmath, amssymb, amsthm geometry graphicx xcolor hyperref booktabs array bm
left=25mm, right=25mm, top=30mm, bottom=30mm

Contents

Course Structure & Current Focus

- Module 1 & 2: Static Data (정적 데이터 분석)
- Module 3: Time Series & Spatial Data (현재 단원: 시공간 데이터)
  - 3.1 Trend & Seasonality (정상성 만들기)
  - 3.2 ARMA / ARIMA (통계적 예측 모델)
  - 3.3 Kalman Filter (상태 추정 및 노이즈 제거)
  - 3.4 Spatial Statistics (Kriging)

Chapter 7

Module 3 (Part A). 시계열 분석 (Time Series Analysis)

> 지금까지 우리는 데이터의 순서를 신경 쓰지 않았습니다(i.i.d 가정). 하지만 주가, 게임 접속자 수, 서버 로그는 순서가 핵심입니다. 어제의 데이터가 오늘의 데이터에 영향을 미치는 자기상관(Autocorrelation)의 세계로 들어갑니다.

☐ 개요 (Overview)

이 단원에서는 시계열 데이터를 분석 가능한 형태로 가공하는 **분해(Decomposition)** 기법, 과거 데이터를 기반으로 미래를 예측하는 **ARIMA** 모델, 그리고 노이즈가 섞인 실시간 데이터에서 진짜 상태를 추정하는 **칼만 필터(Kalman Filter)**를 다룹니다.

☐ 핵심 용어 사전

용어 (Term)	직관적 의미 (Meaning)
Stationarity (정상성)	평균과 분산이 시간에 따라 변하지 않고 일정한 상태. (예측의 전제 조건)
Seasonality (계절성)	주말마다 접속자가 늘어나는 것처럼 주기적인 패턴.
Lag (시차)	현재 시점(t)과 과거 시점($t-k$) 사이의 간격.
White Noise	패턴이 전혀 없는 완전 무작위 잡음. (예측 불가능한 영역)
State-Space Model	관측된 데이터 뒤에 숨겨진 '진짜 상태'가 있다고 가정하는 모델.

—

7.0.1 1. 추세(Trend)와 계절성(Seasonality) 제거

〔개념〕 개념 1: 양파 껍질 벗기기

한 줄 요약: 데이터에서 '뻔한 패턴(상승세, 주말 효과)'을 걷어내야, 진짜 중요한 '변화(신호)'가 보입니다. 이를 위해 데이터를 분해합니다.

1) 시계열 분해 (Decomposition)

$$Y_t = \underbrace{T_t}_{\text{추세}} + \underbrace{S_t}_{\text{계절성}} + \underbrace{R_t}_{\text{잔차}}$$

우리는 Y_t에서 T_t와 S_t를 제거하여, **정상성(Stationarity)**을 가진 잔차 R_t만 남기고 싶어합니다. 왜냐하면 통계적 모델은 "평균이 일정한 데이터"에서 가장 잘 작동하기 때문입니다.

2) 제거 방법

- **차분 (Differencing, I):** $Y_t - Y_{t-1}$. 오늘의 값에서 어제의 값을 뺍니다. 추세(기울기)를 제거하여 수평으로 만듭니다.
- **로그 변환:** 분산이 점점 커지는 경우(깔때기 모양), 로그를 씌워 변동 폭을 일정하게 맞춥니다.
- **계절 차분:** $Y_t - Y_{t-7}$ (일주일 주기인 경우). 이번 주 토요일 데이터에서 지난주 토요일 데이터를 뺍니다.

—

7.0.2 2. 자기회귀 이동평균 모델 (ARMA / ARIMA)

〔개념〕 개념 2: 관성(Inertia)과 충격(Shock)

한 줄 요약: "관성대로 가려는 성질(AR)"과 "외부 충격에 반응하는 성질(MA)"을 합쳐서 미래를 그립니다.

1) 구성 요소: ARIMA(p, d, q)

- **AR (p, AutoRegressive):** 자기회귀.

$$Y_t = \phi Y_{t-1} + \epsilon_t$$

  "어제 접속자가 많았으니, 관성에 의해 오늘도 많을 거야."

- **I (d, Integrated):** 차분. 추세를 제거하기 위해 데이터를 몇 번 미분(뺄셈)했는가? (정상성 확보용)
- **MA (q, Moving Average):** 이동평균.

$$Y_t = \epsilon_t + \theta \epsilon_{t-1}$$

  "어제 예측보다 사람이 갑자기 더 몰렸어(Shock ϵ_{t-1}). 그 여파가 오늘까지 이어질 거야."

2) 모델 선택

보통 **ACF(자기상관함수)**와 **PACF(편자기상관함수)** 그래프를 그려서 p와 q 값을 결정합니다.

—

7.0.3 3. 칼만 필터 (Kalman Filter)

〔개념〕 개념 3: 네비게이션의 원리
한 줄 요약: "내 예측(계산)"과 "센서값(측정)"은 둘 다 부정확합니다. 이 둘을 적절히 섞어서(가중평균) 최적의 "진짜 위치"를 찾아냅니다.

1) 상태 공간 모델 (State-Space Model)

- **Hidden State (x_t):** 진짜 값 (예: 유저의 실제 게임 실력). 눈에 안 보임.
- **Measurement (z_t):** 관측 값 (예: 승패 결과). 노이즈가 섞여 있음.

2) 작동 원리: 무한 루프

1. **Predict (예측):** 어제의 상태를 바탕으로 오늘의 상태를 추측합니다. (불확실성 증가)
2. **Update (보정):** 실제 데이터(z_t)가 들어옵니다. 내 예측과 실제 데이터 사이에서, **더 신뢰할 수 있는 쪽으로** 값을 수정합니다. (불확실성 감소)

3) 강점

과거 데이터를 다 들고 있을 필요 없이, **직전 상태($t-1$)**만 있으면 됩니다. 그래서 실시간(Real-time) 시스템에 매우 유리합니다.

—

7.1 실전 시나리오: 넥슨 게임 지표 분석

〔시나리오〕 Scenario 1: 주말 이벤트 효과 분석 (Decomposition)
토요일에 경험치 2배 이벤트를 했습니다. 접속자가 평일 대비 30% 늘었습니다. 이게 이벤트 덕분일까요, 아니면 원래 토요일이라서 그런 걸까요?

- **해결:** 시계열 분해를 통해 '계절성(토요일 효과)' 성분을 제거합니다.
- **결과:** 계절성을 뺐더니 상승분이 5%밖에 안 남았습니다. 이벤트 효과는 생각보다 미미했습니다.

〔시나리오〕 Scenario 2: 매치메이킹 시스템 (Kalman Filter)
유저의 실력(MMR)을 측정해야 합니다. 고수도 운 나쁘면 지고, 초보도 버스 타면 이깁니다. 승패(Measurement)는 노이즈 투성이입니다.

- **해결:** 칼만 필터를 적용합니다.
- **과정:** 유저가 이겼을 때, MMR을 확 올리지 않고 "이 유저의 실력 불확실성(분산)"을 고려해 조금만 올립니다. 판수가 쌓일수록 불확실성이 줄어들며 '진짜 실력'에 수렴합니다. (TrueSkill 알고리즘의 기초)

—

7.2 자주 묻는 질문 (FAQ)

Q1. 딥러닝(RNN/LSTM)이 ARIMA보다 무조건 좋은가요? A. 아닙니다. 데이터가 적거나(수백 개), 패턴이 단순하고 주기적이라면 **ARIMA**가 훨씬 정확하고 설명력(Why)도 좋습니다. 반면, 비선형적이고 복잡한 패턴(언어, 불규칙한 로그)은 딥러닝이 강합니다.

Q2. 칼만 필터는 예측 모델인가요? A. 엄밀히 말하면 **'추정(Estimation)'** 모델입니다. 현재의 '진짜 상태'를 가장 잘 맞추는 것이 목표입니다. 하지만 추정된 상태를 바탕으로 다음 단계를 Predict 하므로 예측에도 쓰입니다.

Next Step: 시간(t)에 대한 분석은 마쳤습니다. 그런데 데이터에는 시간뿐만 아니라 **위치(x, y)** 정보도 중요합니다. "강남구의 집값이 오르면 서초구도 오를까?" 다음 시간에는 공간적 상관관계를 다루는 **공간 통계(Spatial Statistics)와 크리깅(Kriging)**을 배웁니다.

〔요약〕 Module 3 (Part A) 핵심 요약

- **Decomposition:** $Y = T + S + R$. 추세와 계절성을 제거해야 예측이 가능하다.
- **ARIMA:** 과거의 나(AR)와 과거의 오차(MA)를 합쳐 미래를 예측하는 통계적 도구.
- **Kalman Filter:** 예측(Predict)과 보정(Update)을 반복하며 노이즈 속에서 진짜 상태를 찾아내는 알고리즘. (실시간 처리에 강함)

〔a4paper, 11pt〕book fontspec amsmath, amssymb, amsthm geometry graphicx xcolor hyperref booktabs array bm
left=25mm, right=25mm, top=30mm, bottom=30mm

Contents

Course Structure & Current Focus

- Module 3 (Part A): Time Series Analysis (시간의 흐름)
- Module 3 (Part B): Spatial Statistics (현재 단원: 공간의 분포)
  - 3.5 Gaussian Processes (함수의 확률 분포)
  - 3.6 Kriging & Variograms (지구통계학적 보간)
  - 3.7 Kernel Methods (공간적 유사도 정의)

Chapter 8

Module 3 (Part B). 공간 통계 (Spatial Statistics)

> 지난 시간(Part A)에는 "오늘의 나는 어제의 나"라는 시간적 자기상관(Autocorrelation)을 다뤘습니다. 이번에는 "모든 것은 다른 모든 것과 관련이 있지만, 가까운 것이 더 관련이 있다(Tobler's First Law)"는 지리학의 제1법칙을 바탕으로, 공간적 자기상관을 다룹니다.

☐ 개요 (Overview)

이 단원에서는 듬성듬성한 관측 데이터(예: 기상청 관측소)를 바탕으로 전체 공간(예: 대한민국 전체 미세먼지 지도)을 추정하는 방법을 배웁니다. 특히 **가우시안 프로세스(GP)**를 통해 예측의 불확실성을 정량화하고, 이를 지구통계학에 적용한 **크리깅(Kriging)** 기법을 마스터합니다.

☐ 핵심 용어 사전

용어 (Term)	직관적 의미 (Meaning)
Interpolation (보간)	점들 사이의 빈 공간을 부드럽게 채워 넣는 기술.
Uncertainty	"이 지역 값은 내가 확실히 아는데, 저기는 데이터가 없어서 잘 몰라(분산이 커)."
Variogram	거리가 멀어질수록 데이터 간의 차이(분산)가 어떻게 커지는지 나타내는 그래프.
Kernel (k)	두 지점이 '가깝다'는 것을 수학적으로 정의하는 함수.

—

8.0.1 1. 가우시안 프로세스 (Gaussian Processes, GP)

〔개념〕 개념 1: 함수의 확률 분포

한 줄 요약: 점 하나하나를 예측하는 게 아니라, "데이터를 설명할 수 있는 모든 가능한 곡선(함수)들의 모임"을 그립니다.

1) 핵심 아이디어

일반적인 회귀분석($y = wx + b$)은 파라미터 w를 찾지만, GP는 **함수 $f(x)$ 자체를 확률 변수**로 봅니다.

- **관측된 곳:** 오차가 거의 없이 점을 지나가야 하므로 불확실성(분산)이 0에 가깝습니다.
- **관측 안 된 곳:** 어떤 값이든 될 수 있으므로 불확실성(분산)이 풍선처럼 부풀어 오릅니다.

2) 구성 요소

$$f(x) \sim GP(m(x), k(x, x'))$$

- **평균 함수 (m):** 데이터의 전반적인 추세.
- **커널 함수 (k):** x와 x'가 가까우면 $f(x)$와 $f(x')$도 비슷한 값을 가질 확률이 높다는 '상관관계'를 정의합니다.

—

8.0.2 2. 크리깅 (Kriging)

〔개념〕 개념 2: 지능적인 가중 평균

한 줄 요약: 단순히 가까운 점을 평균 내는 게 아니라, 공간의 특성(배리오그램)을 분석해서 가장 오차가 적은 황금 비율(가중치)로 섞습니다.

1) 작동 원리 (BLUE)

미지의 지점 x_0의 값은 주변 데이터 $Z(x_i)$들의 선형 결합입니다.

$$\hat{Z}(x_0) = \sum_{i=1}^{n} \lambda_i Z(x_i)$$

여기서 가중치 λ_i를 구하는 것이 핵심인데, **배리오그램(Variogram)**을 이용해 오차 분산을 최소화하는 최적의 λ를 찾아냅니다.

2) 배리오그램 (Variogram)

"거리가 h만큼 떨어져 있으면, 값은 얼마나 차이가 날까?"를 나타내는 함수입니다.

- 가까운 거리: 차이가 적음 (상관관계 높음).
- 먼 거리: 차이가 큼 (상관관계 낮음).
- 크리깅은 이 구조를 파악해서, 멀리 있어도 상관관계가 유지되는 방향이라면 가중치를 더 줍니다.

—

8.0.3 3. 커널 방법론 (The Kernel Connection)

〔개념〕 개념 3: 공간의 질감(Texture) 정의하기
한 줄 요약: "이 지역은 부드러운 언덕인가(RBF), 아니면 거친 자갈밭인가(Matern)?"를 수학식(커널)으로 결정합니다.

주요 커널 종류

1. **RBF (Squared Exponential):**
    - 아주 매끄러운(Smooth) 곡선을 만듭니다.
    - 현실 세계의 거친 지형이나 급격한 변화를 설명하기엔 너무 이상적일 수 있습니다.
2. **Matern Kernel:**
    - 거칠기(Roughness)를 조절하는 파라미터가 있습니다.
    - 공간 통계에서 가장 널리 쓰이며, 현실적인 물리 현상을 잘 반영합니다.
3. **Periodic Kernel:**
    - 일정한 패턴이 반복될 때 사용합니다 (예: 계절에 따른 온도 변화).

—

8.1 실전 시나리오: 넥슨 글로벌 서버 랙(Lag) 지도

〔시나리오〕 Scenario: 전 세계 유저들의 네트워크 품질 추정
당신은 넥슨 글로벌 서비스 본부장입니다. 전 세계 곳곳에 흩어진 유저들의 핑(Ping) 데이터를 이용해, 어느 지역에 서버를 증설해야 할지 결정해야 합니다.

1. **데이터:** 일부 유저($n = 10,000$)에게서 수집된 〔위도, 경도, Ping〕 데이터. 대부분의 지역은 데이터가 없는 공백지입니다.
2. **Kriging 적용:**
    - 단순히 근처 유저 값을 평균 내는 게 아니라, 대륙별 인터넷망의 특성(Variogram)을 고려해 보간합니다.
    - 결과: 데이터가 없던 '동남아 시골 마을'의 핑도 주변 도시 데이터를 기반으로 정밀하게 추정해냅니다.
3. **Gaussian Process 활용 (핵심):**
    - GP는 **불확실성(Variance) 지도**를 함께 줍니다.
    - "아프리카 지역은 핑이 200ms로 예상되지만, 불확실성이 매우 높다(± 100)."
    - **의사결정:** 불확실성이 높은 지역은 섣불리 서버를 짓기보다, 먼저 테스트 서버를 열어 데이터를 더 수집하기로 결정합니다.

—

8.2 자주 묻는 질문 (FAQ)

Q1. 단순 평균(Nearest Neighbor)과 크리깅의 결정적 차이는? A. **불확실성(Uncertainty)**과 **구조(Structure)**입니다. 단순 평균은 "값"만 채우지만, 크리깅은 "이 값이 얼마나 믿을만한지(Variance)"를 알려줍니다. 또한, 크리깅은 데이터가 한쪽에 쏠려 있으면(Clustered), 그쪽 데이터들의 가중치를 낮춰서(De-clustering) 전체적인 균형을 맞춥니다.

Q2. 커널은 어떻게 고르나요? A. 데이터에 대한 **도메인 지식**이 필요합니다. "현상은 부드럽게 변하는가, 급격히 변하는가?", "주기적인가?"를 고민해야 합니다. 보통은 Matern 커널로 시작해서 Cross-Validation으로 최적의 커널을 찾습니다.

Next Step: 지금까지 정형 데이터(행렬, 그래프, 시계열, 공간)를 다뤘습니다. 이제 마지막 모듈에서는 비정형 데이터의 끝판왕인 **이미지와 텍스트**를 다루기 위해 다시 딥러닝(CNN, NLP)의 심화 주제로 돌아가거나, 강화학습의 응용을 다루게 됩니다. (MIT 커리큘럼에 따라 유동적)

〔요약〕 Module 3 (Part B) 핵심 요약

- **Gaussian Processes:** 함수를 확률적으로 추정하여 예측값과 불확실성(범위)을 동시에 제공한다.
- **Kriging:** 공간적 상관관계(Variogram)를 분석하여 미관측 지점을 정밀하게 보간하는 지구통계학 기법.
- **Kernel:** 공간적 유사도를 정의하는 핵심 엔진. (RBF = Smooth, Matern = Realistic).

〔a4paper, 11pt〕book fontspec amsmath, amssymb, amsthm geometry graphicx xcolor hyperref booktabs array bm
left=25mm, right=25mm, top=30mm, bottom=30mm

Contents

Course Structure & Current Focus

Chapter 9

Module 4. 인과 추론 (Causal Inference)

> 지금까지 우리는 "A가 변할 때 B도 변하더라(상관관계)"는 패턴을 기가 막히게 찾아냈습니다. 하지만 비즈니스 현장에서는 이것만으로는 부족합니다. 우리가 알고 싶은 건 "그래서 내가 A를 바꾸면 B가 바뀔까?(인과관계)"입니다. 이번 단원에서는 데이터 과학의 가장 높은 벽인 '인과의 세계'를 넘습니다.

☐ 개요 (Overview)

인과 추론은 "내가 이 행동(Treatment)을 하지 않았더라면 결과가 어땠을까?"라는 **반사실(Counterfactual)* 질문을 수학적으로 정의하는 과정입니다. 상관관계와 인과관계의 명확한 차이, 루빈의 잠재적 결과 프레임워크, 그리고 실제 비즈니스 효과(ATE)를 측정할 때 발생하는 선택 편향(Selection Bias)의 위험성을 다룹니다.

☐ 핵심 용어 사전

용어 (Term)	직관적 의미 (Meaning)
Treatment (W)	원인이 되는 행위나 개입 (약물 투여, 쿠폰 발송).
Outcome (Y)	처치로 인해 나타난 결과 (혈압, 구매액).
Confounder (교란변수)	원인과 결과 모두에게 영향을 미쳐 착시를 일으키는 제3의 변수.
Counterfactual (반사실)	"만약 그 약을 안 먹었더라면..." (현실에선 일어나지 않은 가정).
ATE	"이 정책을 도입하면 전체적으로 평균 얼마의 이득이 있는가?"

—

9.0.1 1. 상관관계 vs 인과관계 (Correlation vs Causation)

〔개념〕 개념 1: 보기만 하는 것 vs 행동하는 것
한 줄 요약: 비가 올 때 우산이 펴지는 것을 본다고 해서(상관), 내가 우산을 편다고 비가 오게 만들 수는 없습니다(인과 X).

1) 상관관계 (Correlation)

- **관찰(Observation)의 영역.** $P(Y|X)$.

- 예시: "아이스크림 판매량(X)이 늘면 상어 습격(Y)도 는다."
- 통계적으로 두 변수는 같이 움직입니다. 하지만 아이스크림을 금지한다고 상어가 사라지진 않습니다.
- **이유:** **기온(Summer)**이라는 **교란 변수(Confounder)**가 둘 다를 증가시켰기 때문입니다.

2) 인과관계 (Causation)

- **개입(Intervention)의 영역.** $P(Y|\text{do}(X))$. (Judea Pearl의 do-calculus)
- 의미: "내가 강제로 X를 변화시켰을 때, Y가 변하는가?"
- 데이터 과학의 목표: 우리는 단순히 예측(Prediction)을 넘어, 세상을 바꾸는 개입(Intervention)의 효과를 알고 싶습니다.

—

9.0.2 2. 잠재적 결과 프레임워크 (Potential Outcomes Framework)

〔개념〕 개념 2: 가지 않은 길 (The Road Not Taken)
한 줄 요약: 인과 효과를 안다는 것은 '평행우주'의 나를 훔쳐보는 것과 같습니다. "약을 먹은 나"와 "안 먹은 나"를 동시에 비교해야 합니다.

1) 루빈 인과 모델 (Rubin Causal Model)

어떤 대상 i에 대해 두 가지 잠재적 결과가 존재합니다.

- $Y_i(1)$: 처치($W = 1$)를 받았을 때의 결과.
- $Y_i(0)$: 처치($W = 0$)를 받지 않았을 때의 결과.
- **인과 효과 (Causal Effect):** $\tau_i = Y_i(1) - Y_i(0)$

2) 인과 추론의 근본 문제 (Fundamental Problem)

현실 세계에서 우리는 한 사람에 대해 **둘 중 하나만 관측**할 수 있습니다.

$$Y_i^{obs} = W_i Y_i(1) + (1 - W_i) Y_i(0)$$

약을 먹었다면 $Y_i(1)$은 보이지만, $Y_i(0)$은 영원히 알 수 없는 **반사실(Counterfactual)**이 됩니다. 이것은 본질적으로 **결측치(Missing Data) 문제**입니다.

—

9.0.3 3. 처치 효과 (ATE)와 선택 편향

〔개념〕 개념 3: 비교할 수 없는 것을 비교하지 마라
한 줄 요약: 병원 간 사람은 아프고, 안 간 사람은 건강합니다. 단순히 병원 간 사람의 사망률이 높다고 해서 "병원이 사람을 죽인다"고 할 수 없습니다. 이것이 선택 편향입니다.

1) 평균 처치 효과 (ATE, Average Treatment Effect)

개개인의 인과 효과는 알 수 없으니, 집단 전체의 평균을 봅니다.

$$\text{ATE} = \mathbb{E}[Y(1) - Y(0)] = \mathbb{E}[Y(1)] - \mathbb{E}[Y(0)]$$

2) 단순 비교의 함정 (Naive Comparison)

우리가 데이터에서 계산하는 단순 차이는 다음과 같이 분해됩니다.

$$\underbrace{\mathbb{E}[Y|W=1] - \mathbb{E}[Y|W=0]}_{\text{관측된 차이}} = \underbrace{\text{ATE}}_{\text{진짜 효과}} + \underbrace{(\mathbb{E}[Y(0)|W=1] - \mathbb{E}[Y(0)|W=0])}_{\text{선택 편향 (Selection Bias)}}$$

- **선택 편향:** 처치를 받은 그룹($W = 1$)과 안 받은 그룹($W = 0$)은 애초에 기본 상태($Y(0)$)가 다릅니다.
- **예:** 멤버십 가입자($W = 1$)는 가입 안 해도($Y(0)$) 원래 구매력이 높은 사람들($W = 0$보다 큼)일 가능성이 높습니다. 따라서 관측된 매출 차이를 온전히 멤버십 효과라고 보면 과대평가하게 됩니다.

—

9.1 실전 시나리오: 넥슨 멤버십 효과 분석

〔시나리오〕 Scenario: 월정액 상품이 진짜 돈을 벌어줄까?

당신은 '던전앤파이터'의 PM입니다. 월 1만 원짜리 '아라드 패스(멤버십)'를 출시했습니다. 패스 구매자의 월평균 결제액은 10만 원, 비구매자는 3만 원입니다. 패스의 효과가 +7만 원이라고 보고하면 될까요?

1. **단순 비교 (Correlation):** 10만 $- 3$만 $= +7$만.
2. **의심 (Selection Bias):** "패스를 산 사람들은 원래 게임을 열심히 하는 '핵과금러'들이잖아. 패스가 없었어도 9만 원은 썼을걸?"
3. **잠재적 결과 추정:**
   - $Y(1)$: 패스 구매 시 결제액 (관측됨: 10만)
   - $Y(0)$: 패스 미구매 시 결제액 (**반사실**: 9만 원으로 추정)
4. **인과 효과 (ATE):** 10만 $- 9$만 $= +1$만.
5. **결론:** 실제 증분 매출(Incrementality)은 7만 원이 아니라 1만 원입니다. 단순 비교로 보고했다면 성과를 7배 뻥튀기하는 치명적인 실수를 범했을 것입니다.

—

9.2 자주 묻는 질문 (FAQ)

Q1. 그럼 선택 편향을 어떻게 없애나요? A. 가장 확실한 방법은 **A/B 테스트(RCT, 무작위 대조 실험)**입니다. 동전을 던져서 강제로 처치 그룹과 통제 그룹을 나누면, 두 그룹의 성향이 통계적으로 같아져서 선택 편향이 0이 됩니다.

Q2. A/B 테스트를 못 하는 상황이면요? A. 관측 데이터만으로 인과를 추론해야 합니다. 이때는 **성향 점수 매칭(Propensity Score Matching)**이나 **도구 변수(Instrumental Variables)** 같은 고급 통계 기법을 사용하여, 최대한 비슷한 사람끼리 비교하도록 데이터를 보정합니다. (다음 시간에 배울 내용입니다.)

Next Step: 인과 추론의 개념을 잡았습니다. 하지만 현실은 A가 B를, B가 C를, C가 다시 A를 건드리는 복잡한 **인과 사슬**로 얽혀 있습니다. 다음 시간에는 이런 복잡한 관계를 그림으로 풀어내는 **인과 그래프(Causal Diagrams)**와, 실험 없이도 인과를 찾아내는 관찰 연구 기법들을 배웁니다.

〔요약〕 Module 4 핵심 요약

- **상관 $\neq$ 인과:** 같이 움직인다고 해서 원인과 결과는 아니다. (교란변수 주의)
- **잠재적 결과:** 인과 효과 = (약 먹었을 때의 나) - (약 안 먹었을 때의 나). 하나는 볼 수 없으므로 추론해야 한다.
- **ATE:** 집단 전체의 평균적인 인과 효과. 비즈니스 임팩트의 핵심 지표.
- **선택 편향:** "원래 그런 애들이 선택했다." 단순 비교는 효과를 왜곡한다.

〔a4paper, 11pt〕book fontspec amsmath, amssymb, amsthm geometry graphicx xcolor hyperref booktabs array bm
left=25mm, right=25mm, top=30mm, bottom=30mm

Contents

Course Structure & Current Focus

- Module 4 (Part A): Causal Basics (ATE, Selection Bias)
- Module 4 (Part B): Observational Studies (현재 단원: 관찰 연구 기법)
  - 10.1 Confounders & Simpson's Paradox (함정 피하기)
  - 10.2 Matching & Propensity Score (비슷한 것끼리 묶기)
  - 10.3 Instrumental Variables (도구 변수 활용)
  - 10.4 Regression Discontinuity (경계선 활용)
- Module 5: Deep Learning & Applications (심화 응용)

Chapter 10

Unit 10. 관찰 연구에서의 인과 추론

> Part A에서 우리는 "A/B 테스트(RCT)가 최고다"라고 배웠습니다. 하지만 현실에서는 담배의 해로움을 알기 위해 강제로 담배를 피우게 할 수 없고, 넥슨의 모든 유저에게 강제로 캐시를 지급할 수도 없습니다. 실험이 불가능한 상황에서, 이미 쌓여있는 로그 데이터(Observational Data)만으로 인과관계를 밝혀내는 고급 기술들을 배웁니다.

☐ 개요 (Overview)

이 단원에서는 관찰 데이터에 숨어있는 **교란 변수(Confounder)**의 위험성을 심슨의 역설을 통해 이해하고, 이를 통제하기 위한 3대장 기법을 배웁니다. 1. **성향 점수 매칭 (PSM):** 비슷한 사람끼리 비교하기. 2. **도구 변수 (IV):** 외부 충격을 이용하기. 3. **회귀 불연속 (RD):** 컷오프 경계선의 마법 활용하기.

☐ 핵심 용어 사전

용어 (Term)	직관적 의미 (Meaning)
Observational Study	연구자가 개입하지 않고, 자연 발생한 데이터를 관찰하는 연구.
Confounder (교란요인)	원인과 결과 뒤에서 둘 다 조종하는 흑막. ($Z \to X, Z \to Y$)
Simpson's Paradox	쪼개서 볼 때와 합쳐서 볼 때 결과가 정반대가 되는 기이한 현상.
Propensity Score	"이 사람이 처치(약/쿠폰)를 받을 확률". 복잡한 특성을 점수 하나로 요약.
Instrumental Variable	원인(X)에만 영향을 주고 결과(Y)에는 직접 영향을 안 주는 제3의 변수.

—

10.0.1 1. 교란 변수와 심슨의 역설

〔개념〕 개념 1: 통계의 착시 현상

한 줄 요약: 데이터를 뭉뚱그려(Aggregation) 보면 진실이 가려집니다. 쪼개서(Stratification) 봐야 합니다.

심슨의 역설 (Simpson's Paradox)

- **상황:** 신약 A의 완치율이 B보다 높습니다. (전체 통계)
- **진실:** 남성 그룹에서도 B가 높고, 여성 그룹에서도 B가 높습니다. (???)
- **원인:** A약은 주로 병이 가벼운 사람들에게 투여되었고, B약은 중환자들에게 투여되었기 때문입니다. **'환자의 중증도'**라는 교란 변수를 무시하고 합쳤기 때문에 발생한 착시입니다.
- **교훈:** 인과 추론의 제1원칙은 "비교 가능한 것끼리 비교하라(Ceteris Paribus)"입니다.

—

10.0.2 2. 성향 점수 매칭 (Propensity Score Matching, PSM)

〔개념〕 개념 2: 도플갱어 찾기
한 줄 요약: 약을 먹은 사람과, 안 먹었지만 먹을 뻔했던(성향이 비슷한) 사람을 짝지어서 비교합니다.

1) 원리

변수가 많으면(나이, 성별, 소득, 지역...) 완벽하게 똑같은 짝을 찾기 힘듭니다. 그래서 모든 특성을 **'처치 받을 확률(Propensity Score)'**이라는 하나의 숫자로 압축합니다.

$$e(x) = P(T = 1|X = x)$$

2) 매칭 (Matching)

1. 각 유저가 쿠폰을 받을 확률(성향 점수)을 계산합니다.
2. 실제 쿠폰 받은 사람(T=1, 점수 0.7)과 안 받은 사람(T=0, 점수 0.7)을 매칭합니다.
3. 이 둘의 구매액 차이를 계산하면 순수한 쿠폰 효과를 알 수 있습니다.

—

10.0.3 3. 도구 변수 (Instrumental Variables, IV)

〔개념〕 개념 3: 자연 실험(Natural Experiment) 이용하기
한 줄 요약: 우리가 직접 실험할 수 없을 때, 우연히 발생한 외부 충격(도구)을 이용해 인과관계를 발라냅니다.

1) 도구 변수의 조건

도구 변수 Z는 X(원인)는 건드리지만, Y(결과)는 직접 건드리지 않아야 합니다. ($Z \to X \to Y$)

- **예시:** "군 복무(X)가 평생 소득(Y)에 미치는 영향"
- **문제:** 애국심이나 건강 상태 같은 교란 변수가 섞여 있습니다.
- **도구(Z):** **생일 추첨(징병제 복권)**. 생일은 랜덤이므로 개인 특성과 무관하며, 오직 군대 가는 여부(X)에만 영향을 줍니다.

2) 2단계 최소자승법 (2SLS)

- **Stage 1:** Z로 X를 예측합니다. ($\hat{X}$ 생성) $\rightarrow$ X에서 교란 변수의 때를 벗겨냄.
- **Stage 2:** 깨끗해진 $\hat{X}$로 Y를 설명합니다.

—

10.0.4 4. 회귀 불연속 (Regression Discontinuity, RD)

〔개념〕 개념 4: 깻잎 한 장 차이
한 줄 요약: 커트라인(90점) 바로 위(90점)와 바로 아래(89점) 사람은 실력 차이가 거의 없습니다. 운명만 갈렸을 뿐이죠. 이들을 비교합니다.

1) 원리

임의의 컷오프(Threshold) 주변에서는 마치 **무작위 할당(Random Assignment)**이 일어난 것과 같다고 가정합니다.

- **예:** 장학금 기준이 토익 800점.
- 799점(탈락)과 800점(수혜) 학생을 비교하면 장학금의 인과 효과를 측정할 수 있습니다.

2) 시각적 확인

X축(점수)에 따른 Y축(결과) 그래프를 그렸을 때, 컷오프 지점에서 그래프가 **'점프(Jump)'** 한다면 그것이 바로 인과 효과입니다.

—

10.1 실전 시나리오: 넥슨 PC방 혜택 분석

〔시나리오〕 Scenario: PC방 혜택이 진짜 접속 시간을 늘릴까?
당신은 넥슨의 사업 PM입니다. "PC방에서 접속하면 경험치 +20%" 혜택을 줍니다. PC방 유저의 플레이 시간이 집 유저보다 2배 깁니다. 이게 혜택 덕분일까요?

1. **문제 (Selection Bias):** 원래 게임을 오래 하는 '하드코어 유저'들이 PC방을 더 많이 갑니다. (단순 비교 불가)
2. **해결책 1 (PSM):** 집 유저 중에서도 하드코어한 성향(레벨, 장비 수준 등)이 PC방 유저와 비슷한 사람들을 찾아 매칭하여 비교합니다.
3. **해결책 2 (IV - 도구 변수):** **"비 오는 날(Z)"**을 도구로 씁니다.
    - 비가 오면 귀찮아서 PC방을 덜 갑니다 ($Z \rightarrow X$).
    - 비 자체가 게임 실력(Y)에 영향을 주진 않습니다.
    - 비 때문에 '강제로' 집에서 하게 된 유저들의 플레이 시간 변화를 분석하여 순수한 PC방 효과를 추정합니다.

—

10.2 자주 묻는 질문 (FAQ)

Q1. 셋 중 뭐가 제일 좋나요? A. 상황에 따라 다릅니다.

- **RD (회귀 불연속):** 가장 신뢰도가 높지만(거의 RCT급), 컷오프 규칙이 있는 경우에만 쓸 수 있습니다.
- **IV (도구 변수):** 강력하지만, 완벽한 도구 변수(Z)를 찾는 것이 하늘의 별 따기입니다. (대부분 논문감)
- **PSM (매칭):** 가장 범용적이고 쉽지만, "우리가 관찰하지 못한 변수(Unobserved Confounder)"는 통제 못 한다는 약점이 있습니다.

Q2. 머신러닝이랑 인과 추론을 섞어 쓸 수 있나요? A. 네, 최신 트렌드인 **Causal ML**입니다. 'Double Machine Learning' 같은 기법은 성향 점수나 결과 예측 모델에 딥러닝(XGBoost 등)을 사용하여 추정의 정확도를 높입니다. (주로 Moloco 같은 애드테크에서 많이 씁니다.)

Next Step: 드디어 MIT 6.419x의 긴 여정이 마무리되었습니다. 고차원 데이터 시각화부터, 네트워크 분석, 시공간 통계, 그리고 인과 추론까지. 이제 이 모든 지식을 종합하여 넥슨의 데이터를 바라보면, 전에는 보이지 않던 **'구조'와 '원인'**이 보이기 시작할 것입니다. 고생 많으셨습니다!

〔요약〕 Module 4 (Part B) 핵심 요약

- **심슨의 역설:** 데이터를 뭉뚱그려 보면 인과관계가 뒤집힐 수 있다. 쪼개 봐야 한다.
- **PSM:** 성향이 비슷한 사람끼리 짝지어(Matching) 비교한다.
- **IV:** 외부 충격(도구)을 이용해 자연 실험 상황을 만든다.
- **RD:** 컷오프 경계선에서의 불연속 점프를 인과 효과로 간주한다.

〔a4paper, 11pt〕book fontspec amsmath, amssymb, amsthm geometry graphicx xcolor hyperref booktabs array bm
left=25mm, right=25mm, top=30mm, bottom=30mm

Contents

Course Structure & Current Focus

- Module 3: Spatial Statistics (위치 데이터)
- Module 4: Causal Inference (원인 분석)
- Module 5: Deep Learning Applications (현재 단원: 이미지와 이상 탐지)
  - 5.1 CNN Architecture (시각 정보 처리)
  - 5.2 Autoencoders (비지도 특징 추출)
  - 5.3 Anomaly Detection (이상 징후 포착)
- Module 5 (Part B): Generative Models (VAE, GAN)

Chapter 11

Module 5 (Part A). CNN과 오토인코더

우리는 1차원 데이터(시계열, 테이블)를 넘어 2차원 이상의 고차원 데이터인 **이미지**를 다루게 되었습니다. 단순히 픽셀을 나열하는 것(Flatten)으로는 부족합니다. 이미지의 공간적 구조를 이해하는 **CNN**과, 정답 없이 데이터의 핵심만 요약하는 **오토인코더**를 통해 AI의 활용 범위를 획기적으로 넓혀봅니다.

☐ 개요 (Overview)

이 단원에서는 이미지 데이터의 특성에 맞는 **전처리(Preprocessing)** 기법과 **CNN(합성곱 신경망)**의 핵심 블록을 설계하는 법을 배웁니다. 또한, 데이터를 압축했다 복원하는 **오토인코더(Autoenco 활용해, 라벨링 없이도 **'정상'과 '비정상(이상치)'을 구분하는 방법**을 학습합니다.

☐ 핵심 용어 사전

용어 (Term)	직관적 의미 (Meaning)
Data Augmentation	데이터를 회전, 반전시켜 문제집의 변형 문제를 잔뜩 만드는 것. (과적합 방지)
Pooling	이미지 사이즈를 줄여서 핵심만 남기는 요약 과정.
Latent Vector (Z)	오토인코더가 데이터를 압축해 놓은 '핵심 요약본'.
Reconstruction Error	원본과 복원본의 차이. 이 값이 크면 '이상치(Anomaly)'로 간주함.
Skip Connection	깊은 망에서 신호가 약해지지 않도록 지름길을 뚫어주는 기법. (ResNet)

—

11.0.1 1. 이미지 전처리 및 CNN 구조 설계

〔개념〕 개념 1: AI가 보기 좋게 차려주기

한 줄 요약: 제각각인 원본 이미지를 규격화(Resizing, Normalization)하고, 데이터를 뻥튀기(Augmentation)하 모델을 강하게 키웁니다.

1) 필수 전처리 3단계

1. **Resizing:** 1024×768도, 500×500도 모두 224×224로 통일합니다.

2. **Normalization:** 픽셀값 $0 \sim 255$를 $0 \sim 1$ 또는 평균 0, 분산 1로 바꿉니다. (학습 속도 향상)
3. **Data Augmentation:** 사진을 살짝 돌리거나($\pm 15°$), 좌우 반전하거나, 밝기를 조절합니다.
    - **효과:** 모델이 "고양이는 왼쪽을 봐도, 어두운 곳에 있어도 고양이다"라는 **불변성(Invariance)** 학습하게 됩니다.

2) CNN 핵심 블록

- **Conv Layer:** 특징 탐지. (선, 면, 눈, 코, 입...)
- **Pooling Layer:** 압축. (해상도를 줄여 연산량 감소 + 위치 불변성 확보)
- **Skip Connection (Tip):** 층이 깊어지면 학습이 안 되는 문제(기울기 소실)를 해결하기 위해, 입력값을 출력값에 바로 더해주는($x + F(x)$) 지름길을 만듭니다. (ResNet의 핵심)

—

11.0.2 2. 오토인코더 (Autoencoder): 압축과 복원

〔개념〕 개념 2: 기억해서 그려보기

한 줄 요약: 사진을 보고(X), 핵심만 기억했다가(Z), 다시 그려냅니다($\hat{X}$). 잘 그렸는지 비교해서($Loss$) 학습합니다.

1) 구조 (모래시계 형태)

- **Encoder:** 입력 X를 저차원 벡터 Z로 압축합니다.
- **Bottleneck (Z):** 데이터의 **잠재 특징(Latent Feature)**이 응축된 곳입니다.
- **Decoder:** Z를 이용해 다시 원본 크기의 $\hat{X}$로 복원합니다.

2) 학습 목표

입력을 그대로 뱉어내는 것이 목표입니다. (Identity Function 학습)

$$\text{Minimize } ||X - \hat{X}||^2$$

"이게 왜 쓸모 있나요?" $\rightarrow$ 병목 구간(Z) 때문에, 모델은 데이터의 **가장 중요한 패턴(압축 정보)**을 강제로 배울 수밖에 없습니다.

—

11.0.3 3. 활용: 이상 탐지 (Anomaly Detection)

〔개념〕 개념 3: 모르는 건 못 그린다

한 줄 요약: 정상 데이터만 보고 자란 오토인코더에게 '비정상 데이터'를 보여주면, 복원을 못 해서 에러가 폭발합니다. 이 에러가 바로 '이상 신호'입니다.

1) 작동 원리

1. **학습:** 오직 **정상 데이터(Normal)**만으로 오토인코더를 학습시킵니다. (정상 패턴 마스터)
2. **테스트:** 새로운 데이터가 들어오면 복원해보고, **재구성 오차(Reconstruction Error)**를 잽니다.
3. **판단:**
    - 오차가 작음 $\rightarrow$ "내가 아는 패턴이네. **정상**."
    - 오차가 큼 $\rightarrow$ "처음 보는 패턴이라 복원이 안 돼. **이상(Anomaly)**!"

2) 장점

비정상 데이터(해킹, 불량품)는 구하기가 매우 힘듭니다. 오토인코더는 **비정상 데이터가 하나도 없어도** 이상 탐지 모델을 만들 수 있습니다. (Unsupervised Anomaly Detection)

—

11.1 실전 시나리오: 넥슨 게임 보안 및 운영

〔시나리오〕 Scenario 1: 불건전 프로필 이미지 필터링 (CNN)

유저들이 올리는 프로필 사진 중 성인물이나 혐오 이미지를 걸러내고 싶습니다.

- **해결:** CNN 기반 분류기(Classifier)를 학습시킵니다.
- **전처리:** 다양한 해상도의 이미지를 224×224로 리사이징하고, 회전/자르기 증강(Augmentation)을 통해 모델을 강건하게 만듭니다.
- **효과:** 운영자가 일일이 보던 건수의 90%를 AI가 1차로 필터링하여 운영 비용을 절감합니다.

〔시나리오〕 Scenario 2: 작업장(매크로) 탐지 (Autoencoder)

'메이플스토리'에서 사람이 아닌 봇(Bot)이 24시간 사냥하는 것을 잡고 싶습니다. 봇의 패턴은 계속 진화해서 규칙(Rule)으로 잡기 힘듭니다.

- **해결:** 일반 유저(정상)의 이동 경로와 스킬 사용 시퀀스만 모아서 오토인코더를 학습시킵니다.
- **탐지:** 매크로 유저의 행동 데이터가 들어오면, 모델은 이를 '정상 패턴'으로 복원하지 못해 **재구성 오차가 치솟습니다.**
- **조치:** 오차가 임계값을 넘는 계정만 추출하여 집중 모니터링하거나 제재합니다.

—

11.2 자주 묻는 질문 (FAQ)

Q1. 오토인코더랑 PCA랑 비슷한가요? A. 네, 형제 지간입니다! 오토인코더에서 활성화 함수를 빼고 선형(Linear)으로만 만들면 **PCA와 수학적으로 거의 동일**합니다. 오토인코더는 비선형(ReLU 등) 함수를 써서 PCA보다 훨씬 복잡한 데이터 구조를 압축할 수 있는 **'Deep PCA'**라고 보시면 됩니다.

Q2. 오토인코더로 새로운 이미지를 만들 수 있나요? A. 기본 오토인코더는 복원(압축 해제)만 잘하지, 새로운 창조는 잘 못합니다. (잠재 공간 Z가 불연속적이라서요). 새로운 이미지를 생성하려면 **VAE(Variational Autoencoder)**나 **GAN**을 써야 합니다. (다음 시간에 배웁니다!)

Next Step: 오토인코더는 입력을 '복원'하는 데 그쳤습니다. 이제 AI에게 상상력을 불어넣어 볼까요? 넥슨의 차기작 아트 리소스를 무한대로 생성해낼 수 있는 생성형 AI의 기초, **VAE와 GAN**을 다음 **Unit 5 (Part B)**에서 다룹니다.

〔요약〕 Module 5 (Part A) 핵심 요약

- **Preprocessing:** Resizing, Normalization, Augmentation은 CNN 성능의 8할이다.
- **CNN:** 이미지의 공간적 특징을 추출(Conv)하고 압축(Pool)하여 분류한다.
- **Autoencoder:** 데이터의 압축(Encoder)과 복원(Decoder)을 통해 특징을 학습한다.
- **Anomaly Detection:** 오토인코더의 복원 오차(Reconstruction Error)를 이용해 학습하지 않은 이상 패턴을 잡아낸다.

〔a4paper, 11pt〕book fontspec amsmath, amssymb, amsthm geometry graphicx xcolor hyperref booktabs array bm
left=25mm, right=25mm, top=30mm, bottom=30mm

Contents

Course Structure & Current Focus

- Module 5 (Part A): CNN & Autoencoders (이미지와 이상 탐지)
- Module 5 (Part B): Graph Neural Networks (현재 단원: 관계형 데이터 학습)
  - 12.1 Why GNN? (유클리드 vs 비유클리드 데이터)
  - 12.2 Message Passing (이웃 정보 집계)
  - 12.3 Algorithms: GCN, GraphSAGE, GAT
  - 12.4 Tasks: Node, Link, Graph Classification
- Module 5 (Part C): Generative Models (VAE, GAN)

Chapter 12

Module 5 (Part B). 그래프 신경망 (GNN)

CNN은 픽셀이 격자처럼 예쁘게 정렬된 이미지에서만 작동합니다. 하지만 "철수와 영희는 친구다", "A유저는 B아이템을 샀다"와 같은 데이터는 상하좌우 개념이 없습니다. 이런 불규칙한 네트워크(Graph) 구조를 딥러닝에 넣으려면 어떻게 해야 할까요? GNN은 "친구를 보면 그 사람을 알 수 있다"는 철학으로 이 문제를 해결합니다.

☐ 개요 (Overview)

이 단원에서는 비유클리드 데이터(Graph)를 처리하는 GNN의 핵심 원리인 **메시지 패싱(Message Passing)**을 배웁니다. 또한 실시간 추천 시스템에 특화된 **GraphSAGE**, 중요도에 따라 정보를 선별하는 **GAT** 등 주요 알고리즘을 살펴보고, 이를 통해 노드 분류(어뷰징 탐지)와 링크 예측(추천)을 수행하는 방법을 익힙니다.

☐ 핵심 용어 사전

용어 (Term)	직관적 의미 (Meaning)
Node (Vertex)	네트워크의 점. (유저, 아이템, 게임 등)
Edge (Link)	점들을 잇는 선. (친구 관계, 구매 이력)
Message Passing	이웃 노드들의 정보를 모아서 내 정보를 업데이트하는 과정.
Embedding	노드의 특성과 관계 정보를 압축한 숫자 벡터.
Inductive Learning	학습 때 없었던 새로운 노드가 들어와도 바로 처리가 가능한 방식. (GraphSAGE)

—

12.0.1 1. 왜 GNN인가? (데이터의 구조)

〔개념〕 개념 1: 줄을 설 수 없는 데이터

한 줄 요약: 이미지는 고정된 그리드(3×3)가 있지만, 소셜 네트워크에서 내 친구는 1명일 수도, 100명일 수도 있어서 기존 CNN 필터를 쓸 수 없습니다.

1) 유클리드 vs 비유클리드

- **Euclidean Data:** 이미지, 텍스트, 오디오. (규칙적, 격자 구조) → CNN, RNN 사용.
- **Non-Euclidean Data:** 소셜 네트워크, 분자 구조, 지식 그래프. (불규칙) → **GNN 사용.**

—

12.0.2 2. 핵심 원리: 메시지 패싱 (Message Passing)

〔개념〕 개념 2: 소문 퍼뜨리기 (Gossip)
한 줄 요약: "내 정보"는 "내 이웃들의 정보의 합"으로 정의됩니다. 층(Layer)을 거듭할수록 내 친구의 친구, 그 친구의 친구 정보까지 나에게 도달합니다.

1) 2단계 프로세스

1. **Aggregate (집계):** 연결된 이웃 노드들의 정보를 수집합니다. (예: 평균, 합계)
2. **Update (갱신):** 수집된 이웃 정보와 나의 현재 정보를 결합해, 나의 새로운 상태(Embedding)를 만듭니다.

2) 수식 (GCN 예시)

$$H^{(l+1)} = \sigma(\underbrace{\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}}_{\text{정규화된 인접 행렬}} H^{(l)}W^{(l)})$$

- **의미:** 나의 다음 상태(H^{l+1})는 나와 연결된(A) 이웃들의 현재 상태(H^l)를 가중 평균(W) 낸 것입니다.

—

12.0.3 3. 주요 GNN 알고리즘

〔개념〕 개념 3: GNN의 진화
한 줄 요약: 기본형(GCN)에서 시작해, 대용량 처리를 위한 실전형(GraphSAGE), 중요한 친구를 더 챙기는 고급형(GAT)으로 발전했습니다.

1) GCN (Graph Convolutional Networks)

이미지의 컨볼루션을 그래프로 가져왔습니다. 모든 이웃의 정보를 **똑같은 비중**으로 평균 내어 가져옵니다.

2) GraphSAGE (Sample and Aggregate)

- **문제:** 넷플릭스나 넥슨처럼 노드(유저)가 수억 명이면, 모든 이웃을 다 계산할 수 없습니다.
- **해결:** 이웃 중 몇 명만 **샘플링(Random Sampling)**해서 정보를 가져옵니다.
- **강점 (Inductive):** 학습하지 않은 **신규 유저(New Node)**가 들어와도, 그 유저의 이웃 정보만 있으면 즉시 임베딩을 만들 수 있습니다. (추천 시스템의 핵심)

3) GAT (Graph Attention Networks)

- **아이디어:** 모든 친구가 다 똑같이 중요하진 않습니다.
- **해결:** **Attention 메커니즘**을 사용하여, 나에게 더 중요한 영향을 미치는 이웃의 정보에 **가중치(α)**를 더 줍니다.

—

12.0.4 4. GNN의 주요 태스크 (Tasks)

1. **Node Classification (점 분류):**
   - "이 유저 노드는 악성 봇(Bot)인가, 일반 유저인가?"
   - 활용: 어뷰징 탐지, 신용 사기 탐지.
2. **Link Prediction (선 예측):**
   - "유저 A와 아이템 B 사이에 연결선(구매)이 생길까?"
   - 활용: **추천 시스템 (Recommendation)**.
3. **Graph Classification (그래프 분류):**
   - "이 분자 구조 그래프는 독성이 있는가?"
   - 활용: 신약 개발 (Drug Discovery).

—

12.1 실전 시나리오: 넥슨 추천 시스템 및 보안

〔시나리오〕 Scenario 1: 차세대 게임 추천 시스템 (Link Prediction)
기존 협업 필터링(MF)은 "비슷한 구매 이력"만 봤습니다. GNN은 "유저의 사회적 관계"까지 봅니다.

- **그래프 구성:** 유저, 게임, 길드, 스트리머를 모두 노드로 만듭니다.
- **GNN 적용:**
  - 유저 A는 게임 B를 한 번도 안 봤지만,
  - 유저 A가 속한 '길드'의 길드장이 게임 B를 플레이했습니다.
  - **GraphSAGE**를 통해 길드장의 정보가 유저 A에게 전달(Message Passing)됩니다.
- **결과:** "길드장님이 하는 게임"이라며 추천이 뜹니다. 관계 기반의 정교한 추천이 가능해집니다.

〔시나리오〕 Scenario 2: 작업장 탐지 (Node Classification)
작업장 계정들은 서로 아이템을 주고받는 패턴이 일반 유저와 다릅니다.

- **특징:** 수백 개의 채굴 계정이 하나의 창고 계정으로 재화를 보냅니다 (Star 구조).

- **GNN 적용:** GNN은 각 노드의 '연결 구조적 특징'을 임베딩에 포함합니다.
- **결과:** 행동 로그를 조작해 일반인 흉내를 내더라도, **네트워크 구조상 창고 계정과 연결된 패턴**이 드러나므로 GNN이 이를 '이상 노드'로 즉시 분류해냅니다.

—

12.2 자주 묻는 질문 (FAQ)

Q1. 협업 필터링(MF)과 GNN 추천의 차이는? A. MF는 유저-아이템 상호작용 행렬만 씁니다. GNN은 유저의 프로필(성별, 나이), 아이템의 속성(장르, 가격), 그리고 친구 관계 등 **모든 부가 정보(Feature)**를 그래프에 녹여서 학습할 수 있어 훨씬 성능이 좋습니다. (Cold Start에 강함)

Q2. GNN은 학습이 오래 걸리지 않나요? A. 맞습니다. 그래프 전체를 메모리에 올리는 것은 불가능합니다. 그래서 현업(Pinterest, Uber 등)에서는 **GraphSAGE**처럼 이웃을 샘플링하거나, 그래프를 쪼개서(Sub-graph) 학습하는 기법을 필수로 사용합니다.

> Next Step: 관계를 학습하는 GNN까지 마쳤습니다. 이제 AI의 마지막 경지, **"없던 것을 만들어내는" 생성 모델**로 넘어갑니다. 넥슨의 게임 아트 리소스를 무한히 생성하거나, 가상 유저 데이터를 만들어낼 수 있는 **VAE와 GAN**을 다음 시간에 배웁니다.

〔요약〕 Module 5 (Part B) 핵심 요약

- **GNN:** 불규칙한 그래프 데이터를 처리하는 딥러닝. 관계(Edge)를 통해 정보를 학습한다.
- **Message Passing:** 이웃의 정보를 집계(Aggregate)하고 갱신(Update)하는 GNN의 엔진.
- **GraphSAGE:** 대용량 그래프를 위한 샘플링 기반 알고리즘. 신규 유저 처리에 탁월하다.
- **활용:** 소셜 네트워크 분석, 정교한 추천 시스템, 지능형 어뷰징 탐지.