

December 10, 2025

- 강의명: CS109A: 데이터 과학 입문
- 주차: Lecture 12
- 교수명: Pavlos Protopapas, Kevin Rader, Chris Gumb
- 목적: Lecture 12의 핵심 개념 학습

Contents

1	강의 개요	2
2	주요 공지: 중간고사 및 과제	3
2.1	중간고사 (Midterm)	3
2.2	과제 (Homework)	3
3	핵심 용어 정리	4
4	베이즈 추론: 복잡한 모델의 해법, 시뮬레이션	5
4.1	왜 시뮬레이션이 필요한가?	5
4.2	시뮬레이션 샘플의 활용	5
4.3	MCMC 기법 소개: 샘플은 어떻게 뽑는가?	5
5	빅데이터와 고차원성의 문제	7
5.1	빅데이터 (Big Data)란 무엇인가?	7
5.2	N 이 큰 경우: 많은 관측치	7
5.3	P 가 큰 경우: 많은 예측 변수 (고차원성)	7
6	주성분 분석 (Principal Components Analysis, PCA)	9
6.1	PCA의 핵심 아이디어: 정보의 요약	9
6.2	PCA의 수학적 직관: 고유벡터와 고유값	9
7	PCA의 활용: 시각화와 회귀 분석	10
7.1	활용 1: 고차원 데이터의 시각화 (Visualization)	10
7.2	활용 2: 주성분 회귀 (PCA for Regression, PCR)	10
7.3	PCA의 장단점 요약	11
8	중간고사 핵심 개념 복습	12
8.1	가설 검정 (Hypothesis Testing)	12

8.2 순열 검정 (Permutation Test)	12
8.3 부트스트랩 vs. 순열 검정	13
8.4 신뢰 구간 vs. 예측 구간	13
9 중간고사 대비 체크리스트	14
10 초심자를 위한 FAQ	15
11 빠르게 훑어보기 (1-Page Summary)	16

1 강의 개요

□ 핵심 요약

본 강의는 데이터 과학의 두 가지 주요 주제인 **베이지 추론**과 **차원 축소**를 다룹니다. 먼저, 복잡한 모델의 결과를 해석하기 위해 시뮬레이션(MCMC 등)을 사용하는 베이지 계산 방법을 배웁니다. 이후, 예측 변수(P)가 매우 많은 '고차원 데이터'의 문제점을 알아보고, 이를 해결하기 위한 강력한 기법인 **주성분 분석(PCA)**을 집중적으로 학습합니다. 마지막으로 중간고사를 대비하여 **가설 검정**, **p-value**, **순열 검정** 등 핵심 통계 개념을 복습합니다.

이번 주 학습 목표

- 복잡한 후험 분포(Posterior)를 시뮬레이션하는 MCMC 기법의 원리를 이해합니다.
- '고차원성의 저주'가 무엇인지, 왜 'P가 큰' 데이터가 문제가 되는지 설명할 수 있습니다.
- 주성분 분석(PCA)의 핵심 아이디어를 "최대 분산 방향 찾기"로 설명할 수 있습니다.
- PCA를 활용한 두 가지 주요 사례(시각화, 회귀 분석)를 구분하고 적용할 수 있습니다.
- 고전적 t-검정과 컴퓨터 기반의 순열 검정(Permutation Test)의 차이점을 설명할 수 있습니다.

2 주요 공지: 중간고사 및 과제

중간고사에 대한 주요 공지사항입니다. 시험 준비에 참고하세요.

2.1 중간고사 (Midterm)

- 시기: 다음 주 섹션 시간.
- 형식 (In-Class):
 - 섹션 시간 전체 (75분) 동안 진행됩니다.
 - 퀴즈보다 약 2.2배 긴 분량입니다.
 - 객관식 (multiple-choice) 문제와 단답형/빈칸 채우기 (fill-in-the-blank) 문제로 구성됩니다.
 - 오픈북이 아닙니다. (Closed book)
- 치트 시트 (Cheat Sheets):
 - 총 2장의 치트 시트 (양면 사용 가능)를 허용합니다. (퀴즈는 1장)
- 별도 코딩 시험 (Take-home Coding Portion):
 - 수업(섹션)에서 보는 필기시험과 별개로 진행됩니다.
 - 필기시험 이후에 공개되며, 24시간의 창(window)이 주어집니다.
 - 코딩 시험을 시작하면 2시간 (또는 3시간, 추후 확정 공지)의 제한 시간 내에 완료해야 합니다.
 - AI (LLM) 사용은 금지되지만, 강의 노트 등은 참고 가능합니다.
 - 예상 소요 시간은 2시간 미만이나, 문제 발생 시를 대비해 2시간을 부여합니다.
- 시험 범위: 오늘 강의 (Lecture 12)까지 다룬 모든 주제. (분류 모델링 등 다음 주 내용은 포함되지 않음)
- 연습 문제:
 - 지난 퀴즈의 정답 키 (answer key)가 게시될 예정입니다.
 - 추가 연습 문제 및 복습 자료가 금요일에 공개될 예정입니다.
 - 코딩 연습 문제 제공은 미정입니다. (No promises)

2.2 과제 (Homework)

- Homework 3 마감일: 11월 4일 경으로, 중간고사 이후 약 2주 뒤입니다.
- 중요: HW 3는 중간고사 범위를 많이 다루고 있습니다.
- 마감일이 멀더라도, 중간고사 준비를 위해 반드시 HW 3를 미리 시작하고 풀어보아야 합니다.
- 시험 전까지 모든 코드를 완벽하게 정리할 필요는 없지만, 내용을 읽고 시도해보는 것이 시험에 유리합니다.

3 핵심 용어 정리

이번 강의에서 다루는 주요 용어들을 정리했습니다.

용어	쉬운 설명	원어	비고
베이즈 추론	데이터(증거)를 바탕으로 기존의 믿음(사전 확률)을 업데이트하는 통계적 방식	Bayesian Inference	믿음 \rightarrow 증거 \rightarrow 새로운 믿음
후험 분포	데이터를 관찰한 후 업데이트된 파라미터의 확률 분포	Posterior Distribution	베이즈 추론의 '결과물'
MCMC	복잡한 후험 분포에서 샘플을 추출하는 시뮬레이션 기법의 총칭	Markov Chain Monte Carlo	무작위로 걸어 다니며 샘플 수집
고차원성	데이터의 특성(Feature) 또는 예측 변수(p)의 수가 매우 많은 상태	High Dimensionality	열(Column)이 매우 많은 데이터
차원의 저주	차원이 증가할수록 데이터가 희소(sparse)해지고 분석이 어려워지는 현상	Curse of Dimensionality	"데이터가 외로워진다"
주성분 분석 (PCA)	고차원 데이터의 정보를 최대한 보존하며 저차원으로 축소하는 기법	Principal Components Analysis (PCA)	데이터의 '정보 요약' 기법
주성분	PCA를 통해 새로 생성된, 데이터의 분산을 최대로 설명하는 축 (변수)	Principal Component (PC)	원본 변수들의 '선형 조합'
고유값	해당 축(고유벡터)이 설명하는 '분산의 크기' 또는 '중요도'	Eigenvalue	PCA에서는 PC의 중요도를 의미
고유벡터	데이터 공분산 행렬의 '방향'을 나타내는 벡터	Eigenvector	PCA에서는 새 축의 '방향'을 의미
순열 검정	데이터의 라벨을 무작위로 섞어(H_0 가정) 검정 통계량의 분포를 만드는 기법	Permutation Test	t-검정의 비모수적 대안

Table 1: PCA 및 베이즈, 통계 복습 관련 핵심 용어

4 베イズ 추론: 복잡한 모델의 해법, 시뮬레이션

4.1 왜 시뮬레이션이 필요한가?

베イズ 추론의 핵심은 사전 확률(Prior)과 가능도(Likelihood)를 결합하여 **후험 분포(Posterior Distribution)**를 얻는 것입니다.

- **단순한 경우:** 만약 우리가 사용한 사전 분포가 '켈레 사전 분포(Conjugate Prior)'처럼 공식이 잘 맞는 짝이라면, 후험 분포는 "정규 분포"나 "감마 분포"처럼 우리가 잘 아는 깔끔한 형태로 나옵니다. 이 경우 평균, 중위값, 신뢰 구간 등을 수학 공식으로 쉽게 계산할 수 있습니다.
- **복잡한 경우:** 하지만 현실의 모델(예: 다중 선형 회귀)에서는 파라미터가 매우 많습니다(예: $\beta_0, \beta_1, \dots, \beta_p$ 그리고 σ^2). 이 모든 파라미터들의 결합(joint) 후험 분포는 매우 복잡하고 다차원적인 형태를 띠게 됩니다. 이런 분포는 수학 공식 하나로 깔끔하게 표현하거나 적분하기가 거의 불가능합니다.

복잡한 분포를 이해하는 방법: 시뮬레이션 수학 공식으로 풀 수 없다면, 컴퓨터의 힘을 빌려 그 분포에서 수천, 수만 개의 샘플을 직접 뽑아보면 됩니다. 이렇게 뽑힌 샘플들의 분포(히스토그램)를 관찰하면, 원래의 복잡한 후험 분포가 어떻게 생겼는지 근사적으로 파악할 수 있습니다.

이처럼 복잡한 분포에서 샘플을 추출하는 계산(computational) 기법들을 **MCMC(Markov Chain Monte Carlo)**라고 부릅니다.

4.2 시뮬레이션 샘플의 활용

MCMC 등을 통해 후험 분포에서 N_{sims} 개의 샘플(예: 10,000 개의 β_1 값)을 얻었다고 가정합시다. 이 샘플들을 어떻게 사용할까요?

- **후험 평균(Posterior Mean):** 매우 쉽습니다. 10,000개 샘플의 **표본 평균**을 계산하면 됩니다.
- **신뢰 구간(Credible Interval):** 매우 쉽습니다. 10,000개 샘플을 정렬한 뒤, **백분위수(Percentile)**를 사용하면 됩니다. (예: 95% 신뢰 구간 = 2.5% 지점 값과 97.5% 지점 값). 이는 부트스트랩(Bootstrap)에서 신뢰 구간을 구하는 방식과 동일합니다.
- **후험 최빈값(Posterior Mode):** 어렵습니다. 샘플 데이터만으로는 분포의 가장 높은 '봉우리(peak)'를 정확히 찾기 어렵습니다. 히스토그램을 그려볼 순 있지만, 구간(bin) 설정에 따라 모양이 바뀝니다. 따라서 '커널 밀도 추정(Kernel Density Estimate, KDE)' 같은 기법으로 부드러운 곡선을 피팅한 후, 그 곡선의 최댓값을 찾는 '범프 헌팅(bump-hunting)' 과정이 필요합니다.

4.3 MCMC 기법 소개: 샘플은 어떻게 뽑는가?

MCMC는 복잡한 분포의 정확한 모양(수식)을 몰라도, 특정 지점의 '상대적 높이(확률 밀도)'만 알면 샘플을 뽑을 수 있게 해주는 기법들입니다.

- **적응적 기각 샘플링 (Adaptive Rejection Sampling):** 분포에 '다트'를 던지는 것과 비슷합니다. 분포를 감싸는 단순한 제안 분포(proposal distribution)에서 샘플(x, y 좌표)을 뽑습니다. 만약 뽑힌 샘플이 실제 분포 곡선 '아래'에 떨어지면 '수락(Accept)'하고, 곡선 '위'에 떨어지면 '기각(Reject)'합니다.
- **깁스 샘플링 (Gibbs Sampling):** 다변수 결합 분포를 다룰 때 유용합니다. $f(\theta_1, \theta_2, \theta_3)$ 를 직접 샘플링하는 대신, $\theta_1|\theta_2, \theta_3$ / $\theta_2|\theta_1, \theta_3$ / $\theta_3|\theta_1, \theta_2$ 처럼 각 변수의 **조건부 분포(conditional distribution)**를 번갈아 가며 샘플링합니다.

- **메트로폴리스-헤이스팅스 (Metropolis-Hastings):** 가장 유명한 MCMC 알고리즘 중 하나입니다.

메트로폴리스-헤이스팅스: 안대 쓴 등산가 복잡한 확률 분포를 '산맥'이라고 상상해봅시다. 우리는 이 산맥의 지형도(분포의 전체 형태)는 모르지만, 현재 위치의 '고도(확률 밀도)'는 측정할 수 있습니다.

1. **시작:** 산 중턱 임의의 지점(초기값 x)에서 시작합니다.
2. **제안:** 다음 발걸음을 내디딜 방향과 거리(새로운 위치 x^*)를 무작위로 제안합니다.
3. **결정:**
 - **오르막길 (이동):** 만약 x^* 의 고도가 현재 x 의 고도보다 **높다면** (즉, 확률이 더 높다면), **무조건 그쪽으로 이동합니다.** ($R = f(x^*)/f(x) > 1$)
 - **내리막길 (확률적 이동):** 만약 x^* 의 고도가 더 **낮다면**, **확률적으로 이동합니다.**
 - 완만한 내리막: (예: $R = 0.8$) 80% 확률로 이동합니다.
 - 가파른 내리막(절벽): (예: $R = 0.01$) 1%의 매우 낮은 확률로만 이동합니다.
 - **거절:** 만약 이동하지 않기로 결정되면(예: 80% 확률 이동에 실패), **제자리에 머무릅니다.** (그리고 현재 위치 x 를 샘플에 한 번 더 추가합니다.)
4. **반복:** 2 3번 과정을 수만 번 반복합니다.

결과: 이 '등산가'는 높은 고도(확률이 높은 지역)에서 더 많은 시간을 보내고, 낮은 고도(확률이 낮은 지역)에서는 적은 시간을 보내게 됩니다. 이 등산가의 발자취(방문한 위치 목록)를 모으면, 그것이 바로 우리가 원하던 후험 분포의 샘플이 됩니다.

5 빅데이터와 고차원성의 문제

5.1 빅데이터(Big Data)란 무엇인가?

'빅데이터'는 단순히 데이터가 많은 것을 의미하지만, 데이터 과학에서는 두 가지 다른 차원의 '큼'을 구분해야 합니다. 데이터셋을 행렬(Rows \times Columns)로 볼 때:

- **N이 큰 경우 (Large N):** 행(Row)의 수가 매우 많은 경우 (예: 수백만, 수십억 개의 관측치)
- **P가 큰 경우 (Large P):** 열(Column)의 수가 매우 많은 경우 (예: 수천, 수만 개의 예측 변수)

이 두 상황은 서로 다른 문제를 야기합니다.

5.2 N이 큰 경우: 많은 관측치

- **문제점:**
 - **계산 비용:** 알고리즘이 매우 느려집니다. 단순 평균 계산조차 오래 걸리며, 특히 교차 검증(CV)이나 부트스트랩처럼 반복 계산이 필요하면 시간이 기하급수적으로 늘어납니다.
 - **편향(Bias) 문제:** 데이터가 '많이' 편향된 방식(non-representative)으로 수집되었다면, N이 커질수록 오히려 편향이 심화되어 결과가 나빠질 수 있습니다. ("Garbage in, garbage out")
- **해결책:**
 - **서브샘플링 (Sub-sampling):** 전체 데이터의 10% 또는 1%만 무작위로 추출하여 모델을 학습해도 충분히 좋은 성능을 낼 수 있습니다.
- **특이점:** N이 극도로 커지면 (예: 수억 개), 통계적 추론(p-value, 신뢰 구간)의 중요성이 낮아집니다. 왜냐하면 표준 오차가 0에 수렴하여 모든 변수가 '통계적으로 유의미(p<0.05)'하게 나오기 때문입니다.

5.3 P가 큰 경우: 많은 예측 변수 (고차원성)

N(행)은 적당히 있지만 P(열, 예측 변수)가 N에 가깝거나 N보다 훨씬 많은 경우, 심각한 문제들이 발생합니다.

- **과적합 (Overfitting):** 모델의 자유도가 너무 높아져, 실제 신호(signal)가 아닌 학습 데이터의 노이즈(noise)까지 완벽하게 외워버립니다. 결과적으로 새로운 데이터에 대한 예측 성능이 급격히 떨어집니다.
- **다중공선성 (Multicollinearity):** 예측 변수 간에 높은 상관관계가 존재할 확률이 높습니다.
- **수학적 문제:** OLS(최소제곱법)에서 $X^T X$ 행렬의 역행렬을 계산할 수 없게 됩니다 (Unidentifiability).
- **차원의 저주 (Curse of Dimensionality):** P가 커질 때 발생하는 근본적인 문제입니다.

주의사항

차원의 저주(Curse of Dimensionality)란? 차원이 증가할수록(P가 커질수록) 데이터가 존재하는 공간의 부피(Volume)가 기하급수적으로 커집니다. 이로 인해 동일한 N개의 데이터라도, 차원이 높아지면 데이터 포인트들은 서로에게서 엄청나게 멀리 떨어져 희소(sparse)하게 흩어지게 됩니다.

직관적 비유 1: 큐브 속의 구

- **2차원 (정사각형 안의 원):** 한 변이 2인 정사각형(넓이 4) 안에 반지름 1인 원(넓이 $\pi \approx 3.14$)이 차지하는 비율은 $\pi/4 \approx 78.5\%$ 입니다.

- **3차원 (정육면체 안의 구):** 한 변이 2인 정육면체(부피 8) 안에 반지름 1인 구(부피 $\frac{4}{3}\pi \approx 4.19$)가 차지하는 비율은 $\approx 52.3\%$ 입니다.
- **10차원 (10D-큐브 안의 10D-구):** 이 비율은 약 0.25%로 급격히 떨어집니다.

결론: 차원이 높아질수록, 데이터는 대부분 구(중심부)가 아닌 큐브의 '모서리'에 존재하게 됩니다.

직관적 비유 2: 외로운 데이터 포인트

- 1차원에서는 10개의 점이 꽤 촘촘히 모여있습니다.
- 2차원, 3차원으로 갈수록 같은 10개의 점이 서로 멀리 떨어집니다.
- 1000차원에서는 모든 데이터 포인트가 서로 엄청나게 멀리 떨어져 있습니다. "가까운 이웃(neighbor)"이라는 개념 자체가 무의미해집니다. (k-NN 같은 알고리즘이 작동하기 힘든 이유)

6 주성분 분석 (Principal Components Analysis, PCA)

PCA는 'P가 큰' 고차원성 문제를 해결하기 위한 강력한 차원 축소 (Dimensionality Reduction) 기법입니다.

6.1 PCA의 핵심 아이디어: 정보의 요약

- **문제:** 1000개의 예측 변수 ($P=1000$)가 있지만, 이 중 상당수는 서로 상관관계가 높아 중복된 정보 (redundant)를 담고 있습니다.
- **목표:** 1000개의 변수에 흩어져 있는 '진짜 정보(분산)'를 최대한 보존하면서, 이들을 대표할 수 있는 새로운 축(변수) 몇 개(예: 10개)로 압축하고 싶다.
- **해결책 (PCA):** PCA는 원본 변수들의 선형 조합(linear combination)을 통해, 데이터의 분산(Variance)을 가장 크게 설명하는 새로운 축을 순서대로 찾아냅니다.

PCA: 데이터 구름의 '최적의 축' 찾기 2개의 변수(X_1, X_2)가 있고, 이들의 산점도(scatter plot)가 마치 '길고 얇게 기울어진 타원형 구름'처럼 보인다고 상상해봅시다.

- **기존 축 (X_1, X_2):** X_1 축이나 X_2 축만으로는 이 구름의 흩어짐(분산)을 잘 설명하지 못합니다.
- **PCA의 새 축 (Z):**
 - **제1 주성분 ($Z_1, PC1$):** PCA는 이 구름이 가장 길게 뻗어 있는 방향(기울어진 축)을 찾아냅니다. 이 축이 바로 Z_1 입니다. Z_1 은 이 데이터의 분산을 '최대'로 설명합니다 (예: 전체 분산의 88% 설명).
 - **제2 주성분 ($Z_2, PC2$):** Z_2 는 Z_1 에 수직(orthogonal)이면서 남은 분산을 최대로 설명하는 축입니다 (예: 나머지 12% 설명).

결론: X_1, X_2 대신 Z_1 하나만 사용해도 원본 정보의 88%를 보존할 수 있습니다. 즉, 2차원(X_1, X_2) 데이터를 1차원(Z_1) 데이터로 성공적으로 '차원 축소'한 것입니다.

6.2 PCA의 수학적 직관: 고유벡터와 고유값

(선형대수학을 모른다면 이 부분은 넘어가도 괜찮습니다.)

PCA가 이 '최적의 축'을 찾는 수학적 도구가 바로 **고유벡터(Eigenvector)**와 **고유값(Eigenvalue)**입니다. PCA는 원본 예측 변수 X 의 공분산 행렬($X^T X$)에 대해 '고유값 분해(Eigen-decomposition)'를 수행합니다.

- **고유벡터(Eigenvector):** 공분산 행렬의 '방향'을 나타냅니다.
 - → 주성분(PC)의 방향 (예: Z_1 축의 방향)이 됩니다.
- **고유값(Eigenvalue):** 해당 고유벡터 방향으로 데이터가 얼마나 '퍼져있는지(분산)'를 나타내는 '값'입니다.
 - → 해당 주성분이 설명하는 분산의 크기(PC의 '중요도')가 됩니다.

PCA는 가장 큰 고유값을 가진 고유벡터를 **제1 주성분(PC1)**으로, 두 번째로 큰 것을 제2 주성분(PC2)으로 순서대로 선택합니다.

PCA는 '회전'이다 PCA는 기존의 X_1, X_2 축을 데이터 분산이 최대가 되는 Z_1, Z_2 축으로 **회전(rotation)**시키는 선형 변환(linear transformation)입니다.

7 PCA의 활용: 시각화와 회귀 분석

7.1 활용 1: 고차원 데이터의 시각화 (Visualization)

- **문제:** 784개의 변수($P=784$)를 가진 Fashion MNIST 이미지 데이터를 어떻게 2D 평면에 시각화할 수 있을까요?
- **PCA 해결책:** 1. 784개 변수를 사용해 PCA를 실행합니다. (총 784개의 주성분 Z_1, \dots, Z_{784} 가 나옵니다.) 2. 이 중 가장 중요한 (즉, 분산을 가장 많이 설명하는) **단 2개** (Z_1, Z_2)만 선택합니다. 3. 모든 데이터 포인트를 Z_1 축과 Z_2 축으로 구성된 2D 평면에 뿌립니다.
- **결과:** 이 2D 산점도는 784차원 공간에 존재하는 데이터 그룹의 '가장 특징이 잘 드러나는 2차원 그림자'라고 할 수 있습니다. 우리는 이 2D 그림을 보고 "아, 데이터가 대략 3개의 덩어리(cluster)로 나뉘는구나" 하고 파악할 수 있습니다.
- **펭귄 데이터 예시:** 4개의 측정치(bill length, bill depth 등)를 PCA로 2차원(PC1, PC2)으로 축소하여 시각화했더니, 3종류의 펭귄(Adelie, Chinstrap, Gentoo)이 잘 분리되어 보이는 것을 확인할 수 있었습니다.

주의사항

PCA는 Y(라벨)를 모른다(Unsupervised) PCA는 시각화 시 데이터의 '종류'(예: 펭귄 종류, 옷 종류)를 전혀 고려하지 않습니다. PCA는 오직 X 변수들의 '퍼짐(분산)'만 보고 축을 정합니다. 그럼에도 불구하고 PCA 2D 플롯에서 라벨별로 군집이 잘 분리되었다면, 이는 X 변수들이 Y 를 예측하는 데 유용한 정보를 담고 있다는 강력한 신호입니다.

7.2 활용 2: 주성분 회귀 (PCA for Regression, PCR)

PCA를 회귀 분석의 전처리 단계로 사용하여 과적합을 방지할 수 있습니다.

1. **1단계: PCA 수행** P 개의 원본 변수(X_1, \dots, X_P)로 PCA를 수행하여 P 개의 주성분(Z_1, \dots, Z_P)을 만듭니다.
2. **2단계: 주성분 선택 (m 개)** P 개의 주성분 중 상위 m 개 (단, $m < P$)만 선택합니다. m 을 결정하는 방법은 3가지가 있습니다.
 - **A. 스크리 플롯 (Scree Plot) / 엘보우 방법 (Elbow Method):** 각 주성분(PC1, PC2, ...)이 설명하는 분산의 크기를 막대그래프로 그립니다. 그래프가 급격히 꺾이는 '팔꿈치(elbow)' 지점에서 m 을 결정합니다. (예: 20개 이후로는 설명력이 급감하니 20개만 쓰자)
 - **B. 누적 설명 분산 (Cumulative Variance Explained):** "전체 분산의 90%를 설명하는 지점까지" m 을 선택합니다. (예: PC 53개를 더하니 누적 분산이 90%가 되었다면 $m = 53$ 으로 설정)
 - **C. 교차 검증 (Cross-Validation):** m 을 모델의 하이퍼파라미터로 취급합니다. $m = 1, 2, 3, \dots$ 일 때의 검증 MSE를 각각 계산하여, MSE가 가장 낮은 최적의 m 을 선택합니다. (가장 성능 지향적인 방법)
3. **3단계: 회귀 모델 학습** 선택된 m 개의 주성분을 새로운 예측 변수로 사용하여 선형 회귀 모델을 학습합니다.

$$Y = \beta_0 + \beta_1 Z_1 + \beta_2 Z_2 + \dots + \beta_m Z_m$$

7.3 PCA의 장단점 요약

PCA는 유용하지만 만능은 아닙니다.

장점 (Pros)	단점 (Cons)
<ol style="list-style-type: none"> 1. 차원의 저주 해결: 과적합 위험을 크게 줄여줍니다. 2. 다중공선성 제거: 주성분들은 정의상 서로 수직 (직교) 하므로, 변수 간 상관관계가 0이 됩니다. 3. 시각화 가능: 고차원 데이터를 2D/3D로 시각화할 수 있습니다. 4. 계산 효율성 향상: 변수의 수가 줄어 모델 학습이 빨라집니다. 	<ol style="list-style-type: none"> 1. 해석력 상실: 주성분은 원본 변수들의 복잡한 조합 (예: $Z_1 = 0.5X_1 - 0.2X_2 + \dots$) 이므로, β_1 이 무엇을 의미하는지 직관적으로 해석하기 불가능해집니다. 2. Y 정보 무시 (Unsupervised): PCA는 Y 를 전혀 보지 않습니다. X 의 분산을 90% 설명하는 PC1이 Y 를 예측하는 데는 전혀 중요하지 않을 수도 있습니다. (최악의 경우 Y 예측에 중요한 정보가 PC100에 있을 수도 있음) 3. 예측 성능 향상 보장 없음: 성능이 항상 좋아지지는 않습니다.

Table 2: PCA의 장점과 단점

8 중간고사 핵심 개념 복습

8.1 가설 검정 (Hypothesis Testing)

가설 검정은 우리가 데이터에서 관찰한 효과(예: β_1 의 기울기)가 '실제 효과'인지, 아니면 '단순한 우연(random chance)'에 의한 것인지 판단하는 통계적 절차입니다.

가설 검정의 5단계:

1. 가설 설정:
 - 귀무가설 (H_0): "효과가 없다." (예: $\beta_1 = 0$. 즉, X 와 Y 는 관련이 없다.)
 - 대립가설 (H_A): "효과가 있다." (예: $\beta_1 \neq 0$)
2. 검정 통계량 선택: 가설을 검증할 측도(measure)를 정합니다. (예: t-statistic)
3. 검정 통계량 계산: 수집한 데이터로 해당 통계량을 계산합니다. (예: $\hat{\beta}_1/SE(\hat{\beta}_1)$)
4. p-value 계산 및 결정: 이 통계량이 H_0 하에서 얼마나 극단적인 값인지 확률(p-value)로 계산합니다. (보통 $\alpha = 0.05$ 와 비교)
5. 결론 도출:
 - $p < 0.05$: H_0 을 기각(Reject)합니다. (즉, $\beta_1 \neq 0$ 일 가능성이 높다.)
 - $p \geq 0.05$: H_0 을 기각하는 데 실패(Fail to Reject)합니다.

p-value란 무엇인가? p-value란, "만약 귀무가설(H_0)이 사실이라면, 우리가 관찰한 검정 통계량(예: $t = 2.5$)보다 같거나 더 극단적인 값이 나올 확률"을 의미합니다.

- p-value가 낮다 (예: 0.01): H_0 이 사실이라는 가정 하에서는 거의 일어나지 않을(1%) 일이 벌어졌다 \rightarrow "아무 효과가 없다"는 H_0 가정이 틀린 것 같다 $\rightarrow H_0$ 을 기각한다.
- "If the p-value is low, H_0 must go!" (p값이 낮으면, H_0 은 꺼져라!)

8.2 순열 검정 (Permutation Test)

왜 필요한가? 고전적인 t-검정은 데이터가 정규성, 등분산성 등의 가정을 만족해야 한다는 '수학적 짐(baggage)'을 가지고 있습니다. 만약 우리 데이터가 이 가정을 명백히 위반한다면 (예: 분산이 일정하지 않음), t-검정의 p-value를 신뢰할 수 없습니다.

순열 검정의 아이디어 (컴퓨터를 이용한 대안): 순열 검정은 H_0 이 사실이라는 가정(즉, X 와 Y 는 아무 관련이 없다)을 컴퓨터 시뮬레이션으로 구현합니다.

1. 관찰: X 와 Y 사이의 실제 기울기(예: $\hat{\beta}_1 = 0.58$)를 계산하여 저장합니다.
2. 가정 (H_0): X 와 Y 가 관련 없다면, Y 값들(price)을 무작위로 뒤섞어서(shuffle) X (sqft)와 다시 짝지어도 상관없을 것입니다.
3. 시뮬레이션:
 - Y 값을 무작위로 섞은 $Y_{permute}$ 를 만듭니다.
 - 이 $Y_{permute}$ 와 X 사이의 기울기 $\hat{\beta}_{permute}$ 를 계산합니다. (이 값은 H_0 이 사실일 때의 기울기 샘플입니다.)
 - 이 과정을 1000번 (또는 10000번) 반복합니다.
4. p-value 계산: 1000개의 $\hat{\beta}_{permute}$ 값들 (H_0 분포) 중에서, 우리가 처음에 관찰한 실제 기울기(0.58)보다 더 극단적인(절대값이 큰) 값의 비율을 계산합니다.

5. 결론: 만약 이 비율(p-value)이 0.05보다 작으면, H_0 을 기각합니다.

8.3 부트스트랩 vs. 순열 검정

두 기법 모두 데이터를 재추출(resampling)하지만, 목적과 방식이 완전히 다릅니다.

특징	부트스트랩 (Bootstrap)	순열 검정 (Permutation Test)
목표	추정 (Estimation)	가설 검정 (Hypothesis Testing)
질문	"내 통계량($\hat{\beta}_1$)이 얼마나 불확실한가?"	" H_0 이 사실이라는 가정 하에 내 $\hat{\beta}_1$ 이 흔한 값인가?"
결과물	신뢰 구간 (Confidence Interval)	p-value
가정	H_A (대립가설)을 가정. (관찰된 데이터가 모집단을 대표한다고 믿음)	H_0 (귀무가설)을 가정. (X, Y 는 관련 없음)
방법	복원 추출 (Sampling with replacement) (데이터셋에서 (x_i, y_i) 쌍을 그대로 뽑음)	비복원 서플링 (Sampling without replacement) (Y 라벨만 뒤섞음)

Table 3: 부트스트랩과 순열 검정의 비교

8.4 신뢰 구간 vs. 예측 구간

모델의 불확실성을 표현하는 두 가지 다른 '구간'입니다.

특징	신뢰 구간 (Confidence Interval, CI)	예측 구간 (Prediction Interval, PI)
질문	"특정 x_0 에서 평균 반응값 $E[Y x_0]$ 이 어디쯤 있을까?"	"특정 x_0 에서 새로운 데이터 1개 y_{new} 가 어디쯤 있을까?"
의미	모델(회귀선) 자체의 불확실성 (데이터를 다시 뽑으면 선이 얼마나 바뀔까?)	모델 불확실성 + 데이터 고유의 노이즈(ϵ) (같은 x_0 라도 Y 는 원래 흩어져 있음)
폭	좁다 (N이 커지면 0에 수렴)	항상 더 넓다 (N이 커져도 ϵ 의 불확실성은 남음)

Table 4: 신뢰 구간(CI)과 예측 구간(PI)의 비교

9 중간고사 대비 체크리스트

중간고사 준비: 자가 점검표

중간고사 시험 범위(오늘 강의까지)를 정확히 알고 있는가?

중간고사가 '필기 시험(In-Class)'과 '코딩 시험(Take-home)'으로 나뉘는 것을 이해했는가?

치트 시트 2장(양면)을 준비하기 시작했는가?

Homework 3를 (마감일과 상관없이) 시험 공부 목적으로 미리 풀어보고 있는가?

베이즈 MCMC의 '목적' (왜 공식을 안 쓰고 시뮬레이션 하는지)을 설명할 수 있는가?

'N이 큰' 문제(계산 속도)와 'P가 큰' 문제(과적합, 차원의 저주)를 구분할 수 있는가?

'차원의 저주'를 "데이터가 희소해지고(sparse) 이웃이 멀어진다"고 설명할 수 있는가?

PCA의 핵심 아이디어가 "데이터 분산이 최대가 되는 새 축을 찾는 것"임을 아는가?

PCA가 수학적으로 '공분산 행렬의 고유벡터'를 찾는 것과 같음을 이해하는가?

PCA를 언제 사용하는가? (1. 시각화, 2. 회귀 분석(PCR))

PCR에서 사용할 주성분의 개수(m)를 정하는 3가지 방법(Elbow, Variance, CV)을 아는가?

PCA의 가장 큰 단점이 '해석력 상실'과 'Y를 무시'하는 것임을 아는가?

가설 검정의 5단계를 말할 수 있는가? (H_0 설정, 통계량, 계산, p-value, 결론)

p-value를 " H_0 이 사실일 때, 관찰값보다 극단적인 값이 나올 확률"이라고 정의할 수 있는가?

t-검정의 가정이 깨졌을 때 '순열 검정'을 사용할 수 있음을 아는가?

부트스트랩(복원추출, 추정)과 순열 검정(서플링, 검정)의 차이를 설명할 수 있는가?

신뢰 구간(평균의 불확실성)과 예측 구간(새 데이터 1개의 불확실성)을 구분할 수 있는가?

10 초심자를 위한 FAQ

Q: PCA는 지도학습인가요, 비지도학습인가요? **A: 완벽한 비지도 학습(Unsupervised Learning)입니다.** PCA는 차원을 축소하기 위해 오직 예측 변수(X)의 정보(분산, 공분산)만을 사용합니다. 반응 변수(Y , 라벨)는 PCA 계산 과정에서 전혀 고려되지 않습니다.

Q: 주성분(PC)은 원본 변수와 다른가요? **A: 완전히 다릅니다.** PC1(제1 주성분)은 $Z_1 = w_{11}X_1 + w_{12}X_2 + \dots + w_{1p}X_p$ 처럼 모든 원본 변수(X_1 부터 X_p 까지)가 조금씩 섞인 새로운 변수입니다. 원본 변수 중 하나(예: X_1)를 선택하는 것(Feature Selection)과는 근본적으로 다릅니다.

Q: PC1이 항상 Y 를 가장 잘 예측하나요? **A: 절대 아닙니다.** PC1은 X 의 '분산'을 가장 잘 설명할 뿐입니다. Y 를 예측하는 데 가장 중요한 정보가 X 분산의 1%만 설명하는 PC10에 들어있을 수도 있습니다. 이것이 PCA의 한계입니다. (이와 달리 Y 정보까지 고려하여 축을 찾는 것을 '부분 최소 제곱, PLS'라고 부릅니다.)

Q: t-검정과 순열 검정 중 무엇을 써야 하나요? **A: 데이터의 가정을 먼저 확인해야 합니다.**

- **t-검정:** 데이터가 (특히 잔차가) 정규성을 따르고 등분산성을 만족하는 등, 고전적 통계 가정을 잘 만족할 때 사용합니다. 더 적은 계산으로 강력한 결과를 줍니다.
- **순열 검정:** 데이터가 정규성/등분산성 가정을 만족하지 않을 때 사용하는 '비모수적(non-parametric)' 대안입니다. 수학적 가정 대신 컴퓨터의 계산 능력에 의존합니다.

11 빠르게 훑어보기 (1-Page Summary)

베이지스 시뮬레이션 (MCMC)

- **Why?** 모델이 복잡해지면 '후험 분포'를 수학 공식으로 풀 수 없다.
- **What?** 공식 대신, 분포에서 수천 개의 '샘플'을 뽑아 분포의 모양을 근사한다.
- **How?** 메트로폴리스-헤이스팅스 (안대 쓴 등산가 비유: 높은 곳(확률)에서 더 많은 시간을 보냄), 깃스 샘플링 등
- **Use?** 샘플의 평균 (\rightarrow 후험 평균), 샘플의 백분위수 (\rightarrow 신뢰 구간)를 계산한다.

고차원성의 문제 (P is Big)

- **Problem?** 예측 변수(P)가 관측치(N)만큼 많거나 더 많은 상황.
- **Curse of Dimensionality:** 차원이 높아질수록 공간의 부피가 커져 데이터가 '희소(sparse)'해지고 모든 점이 서로 '멀어지는' 현상.
- **Result:** 과적합(Overfitting), 다중공선성, 모델 불안정.

주성분 분석 (PCA)

- **Goal:** 고차원(P가 큼) 데이터의 '정보(분산)'를 최대한 보존하며 저차원으로 '압축'.
- **Idea:** 데이터의 분산이 '최대'가 되는 새 축(방향)을 찾는다 (= Z_1 , PC1).
- **Math:** 공분산 행렬의 '고유벡터(Eigenvector)'가 새 축의 방향, '고유값(Eigenvalue)'이 그 축의 중요도(설명 분산)가 된다.
- **Usage 1 (Viz):** 고차원 데이터를 PC1, PC2의 2D 평면에 시각화 (Unsupervised).
- **Usage 2 (PCR):** 상위 m 개의 PC (Z_1, \dots, Z_m)를 회귀 모델의 예측 변수로 사용.
- **Trade-off:** 과적합은 막지만, 모델의 '해석력'을 잃는다.

가설 검정 복습

- **p-value:** H_0 (효과 없음)이 사실일 때, 내 관찰값보다 더 극단적인 값이 나올 확률. (낮으면 H_0 기각)
- **Permutation Test:** H_0 을 시뮬레이션(Y 셔플링)하여 p-value를 계산. (t-검정 가정이 깨졌을 때 사용)
- **Bootstrap vs. Permutation:** 부트스트랩(복원추출)은 '추정'(CI)용, 순열검정(셔플링)은 '검정'(p-value)용.
- **CI vs. PI:** CI(신뢰 구간)는 '평균'의 불확실성 (좁음). PI(예측 구간)는 '새 데이터 1개'의 불확실성 (넓음).