December 10, 2025

▣ **강의명:** CS109A: 데이터 과학 입문

▣ **주차:** Lecture 15

▣ **교수명:** Pavlos Protopapas, Kevin Rader, Chris Gumb

▣ **목적:** Lecture 15의 핵심 개념 학습

Contents

Abstract

개요 (Overview)

본 문서는 로지스틱 회귀를 3개 이상의 클래스를 분류하는 **다중 클래스(Multiclass)** 문제로 확장하는 방법을 다룹니다. 기준이 되는 하나의 클래스와 나머지를 비교하는 **다항(Multinomial) 로지스틱 회귀**와, 각 클래스와 그 외 모든 클래스를 비교하는 **OvR(One-vs-Rest)** 접근법을 배웁니다. 분류 모델의 성능을 평가하기 위한 **혼동 행렬(Confusion Matrix)**, **ROC 곡선**, **AUC** 개념을 학습합니다. 마지막으로, 파라미터를 확률 분포로 간주하는 **베이즈(Bayesian) 추론**의 기본 개념과, 이항 분포의 켤레 사전 확률인 **베타 분포(Beta Distribution)**를 활용한 **베타-이항 모델**을 살펴보고, **계층 모델(Hierarchical Model)**의 필요성을 소개합니다.

1 핵심 용어 정리

본 강의에서 다루는 주요 용어들을 미리 살펴봅니다.

[title=주요 용어]

- **다중 클래스 분류 (Multiclass Classification):** 결과 변수(Y)가 3개 이상의 범주를 가지는 분류 문제입니다. (예: 학생의 전공을 'CS', '통계', '기타'로 예측)
- **다항 로지스틱 회귀 (Multinomial Logistic Regression):** 다중 클래스 분류 기법 중 하나. 하나의 클래스를 '기준(reference)' (예: K번째 클래스)로 설정하고, 다른 모든 클래스(1, 2, ..., K-1)를 이 기준 클래스와 비교하는 $K-1$개의 이진 로지스틱 모델을 적합시킵니다.
- **One-vs-Rest (OvR) 로지스틱 회귀:** 다중 클래스 분류 기법 중 하나. 총 K개의 클래스가 있다면, K개의 이진 로지스틱 모델을 각각 적합시킵니다. 각 모델은 '특정 클래스 k' vs 'k를 제외한 나머지 모든 클래스'를 분류합니다.
- **소프트맥스 함수 (Softmax Function):** 여러 개의 점수(score)를 입력받아, 총합이 1이 되는 확률 값들의 집합으로 변환하는 함수입니다. OvR 모델 등에서 각 클래스에 속할 최종 확률을 계산하는 데 사용됩니다.
- **혼동 행렬 (Confusion Matrix):** 분류 모델의 예측 결과를 실제 값과 비교하여 표로 나타낸 것입니다. TP, FP, TN, FN 값을 포함합니다.
- **ROC 곡선 (Receiver Operating Characteristic Curve):** 분류 모델의 임계값(threshold)이 변함에 따라 **True Positive Rate (TPR, Y축)**와 **False Positive Rate (FPR, X축)**가 어떻게 변하는지를 그린 그래프입니다.
- **AUC (Area Under the Curve):** ROC 곡선의 아래쪽 면적. 1에 가까울수록 모델의 성능이 좋다고 평가하며, 0.5는 무작위 추측과 같은 수준임을 의미합니다.
- **베이즈 추론 (Bayesian Inference):** 모델의 파라미터를 고정된 값이 아닌 확률 분포로 간주하는 통계적 접근 방식입니다. **사전 확률(Prior)**에 **가능도(Likelihood)**를 곱하여(데이터를 반영하여) **사후 확률(Posterior)**을 계산합니다.
- **베타 분포 (Beta Distribution):** [0, 1] 사이의 값을 가지는 연속 확률 분포. 확률값(p) 자체의 불확실성을 모델링하는 데 사용되며, 이항 분포의 **켤레 사전 확률(Conjugate Prior)**입니다.

2 다중 클래스 로지스틱 회귀 (Multiclass Logistic Regression)

2.1 왜 다중 클래스 분류가 필요한가?

기존의 로지스틱 회귀는 반응 변수 Y가 0 또는 1 (예: 실패/성공, 스팸/아님)인 **이진 분류(Binary Classification)** 문제에 사용되었습니다.

하지만 현실의 많은 문제는 3개 이상의 범주를 가집니다.

- 학생의 전공 예측: {컴퓨터 과학, 통계학, 기타}
- 미식축구 플레이 예측: {패스, 런, 스페셜 팀}
- 상품 카테고리 분류: {의류, 가전, 식품, 도서}

이러한 문제를 **다중 클래스 분류(Multiclass Classification)**라고 부릅니다. 다중 클래스 문제는 범주의 순서 유무에 따라 두 가지로 나뉩니다.

- **명목형 (Nominal):** 범주 간에 순서가 없습니다. (예: 눈동자 색 - 파랑, 갈색, 초록)
- **순서형 (Ordinal):** 범주 간에 명확한 순서가 있습니다. (예: 평점 - 1점, 2점, 3점, 4점, 5점)

이번 강의에서는 **명목형** 다중 클래스 문제를 다루는 두 가지 주요 방법을 배웁니다.

2.2 방법 1: 다항 로지스틱 회귀 (Multinomial Logistic Regression)

이 방법은 '기준 그룹'을 하나 정하고, 다른 모든 그룹을 이 기준 그룹과 비교하는 방식입니다.

□ 예제: title

$K = 3$개의 클래스 {CS(1), Stat(2), Other(3)}가 있다고 가정합니다. 만약 **Other(3)를 기준(reference) 그룹**으로 삼는다면, 우리는 $K - 1 = 2$개의 이진 로지스틱 모델을 만듭니다.

- **모델 1:** CS(1) vs Other(3) 분류

$$\ln\left(\frac{P(Y=1)}{P(Y=3)}\right) = \beta_{0,1} + \beta_{1,1}X_1 + \cdots + \beta_{p,1}X_p$$

- **모델 2:** Stat(2) vs Other(3) 분류

$$\ln\left(\frac{P(Y=2)}{P(Y=3)}\right) = \beta_{0,2} + \beta_{1,2}X_1 + \cdots + \beta_{p,2}X_p$$

2.2.1 Q: 2개의 모델로 어떻게 3개의 확률을 얻나요?

좋은 질문입니다. 우리는 $P(Y=1), P(Y=2), P(Y=3)$ 세 가지를 알고 싶습니다. 위의 두 모델은 두 개의 방정식을 제공합니다. 하지만 미지수는 3개입니다. 이때, 확률의 기본 속성인 **"모든 확률의 합은 1이다"**라는 세 번째 방정식을 사용합니다.

1. $\frac{P(Y=1)}{P(Y=3)} = e^{\beta_1 X}$ ($\beta_1 X$는 모델 1의 선형 결합)
2. $\frac{P(Y=2)}{P(Y=3)} = e^{\beta_2 X}$ ($\beta_2 X$는 모델 2의 선형 결합)
3. $P(Y=1) + P(Y=2) + P(Y=3) = 1$

이 3개의 방정식을 연립하여 $P(Y=1), P(Y=2), P(Y=3)$을 모두 구할 수 있습니다. (예: 1번과 2번

식을 $P(Y=1)$과 $P(Y=2)$에 대해 정리하여 3번 식에 대입하면 $P(Y=3)$를 구할 수 있습니다.)

sklearn 라이브러리 사용 시 참고

이론적으로는 $K-1$개의 모델을 적합하지만, `sklearn`의 `LogisticRegression(multi_class='multinomial')`은 K개의 계수 세트(β)를 반환합니다.
이는 `sklearn`이 내부적으로 계산을 정규화(renormalize)하여, 각 클래스 k에 대해 $P(Y=k)$ vs $P(Y \neq k)$ (k vs k가 아닌 것)에 대한 해석이 가능하도록 변환해주기 때문입니다. 처음에는 혼동될 수 있지만, K개의 확률을 직접 다루는 것이 더 직관적일 수 있습니다.

2.3 방법 2: One-vs-Rest (OvR) 로지스틱 회귀

이 방법은 '기준 그룹' 없이, 각 클래스가 돌아가면서 주인공이 되는 방식입니다. K개의 클래스가 있다면 K개의 모델을 만듭니다.

□ 예제: title

$K=3$개의 클래스 {CS, Stat, Other}가 있다면, 3개의 이진 로지스틱 모델을 만듭니다.

- **모델 1:** CS vs (Stat + Other) 분류

$$\ln\left(\frac{P(Y=\text{CS})}{P(Y\neq\text{CS})}\right)=\beta_{\text{CS}}X$$

- **모델 2:** Stat vs (CS + Other) 분류

$$\ln\left(\frac{P(Y=\text{Stat})}{P(Y\neq\text{Stat})}\right)=\beta_{\text{Stat}}X$$

- **모델 3:** Other vs (CS + Stat) 분류

$$\ln\left(\frac{P(Y=\text{Other})}{P(Y\neq\text{Other})}\right)=\beta_{\text{Other}}X$$

2.3.1 Q: 이 3개의 확률은 합이 1이 되나요?

아니요, 보장되지 않습니다. 이 3개의 모델은 **독립적으로** 학습됩니다. 모델 1은 "이 학생이 CS일 확률" (p_{CS})을, 모델 2는 "Stat일 확률" (p_{Stat})을, 모델 3은 "Other일 확률" (p_{Other})을 계산합니다. 이 3개의 확률($p_{CS}, p_{Stat}, p_{Other}$)을 단순히 더하면 1이 되지 않을 수 있습니다.

2.3.2 해결책: 소프트맥스 (Softmax) 함수

이 문제를 해결하기 위해, 각 모델에서 나온 "점수"(score, βX)를 총합이 1이 되는 확률로 변환하는 **소프트맥스 함수**를 사용합니다.

[title=소프트맥스 함수 (Softmax Function)] K개의 클래스에 대한 점수(logits) $\vec{s}=(s_1,s_2,\ldots,s_K)$가 있을 때, k번째 클래스에 속할 확률 P_k는 다음과 같이 계산됩니다.

$$P_k=\frac{e^{s_k}}{\sum_{j=1}^{K}e^{s_j}}$$

직관적 해석: 1. e^{s_k} **(지수 함수):** 모든 점수를 양수로 만들고, 큰 점수와 작은 점수의 차이를 더욱 증폭시킵니다. (Winner-takes-most) 2. $\sum e^{s_j}$ **(총합):** 모든 클래스의 증폭된 점수 총합입니다. 3. **나누기:** 각 클래스의 증폭된 점수를 총합으로 나누어, 전체에서 차지하는 "비율"을 계산합니다. 이렇게 하면 모든 확률(P_k)의 합은 항상 1이 됩니다.

2.4 Multinomial vs. OvR: 무엇을 써야 할까?

두 방법은 종종 매우 유사한 예측 결과를 제공합니다. 미식축구(NFL) 플레이 타입을 예측하는 예제(패스, 런, 기타)에서도 두 모델의 예측 확률 그래프는 거의 동일한 경향을 보였습니다.

특징	다항 (Multinomial)	OvR (One-vs-Rest)
모델 개수	$K-1$ 개	K 개
개념	기준 클래스(K) vs. 나머지(k)	클래스(k) vs. 나머지($\neq k$)
효율성	약간 더 효율적 (모델 적음)	개념이 단순함
적합	추론/계수 비교에 유리	순수 분류(prediction)에 선호됨
결과	대부분의 경우 매우 유사한 성능을 보임	

어떤 모델이 더 나은지는 **교차 검증(Cross-validation)**을 통해 '테스트 데이터'에 대한 성능(예: 손실 함수 값)을 비교하여 결정할 수 있습니다.

2.5 다중 클래스에서의 예측과 손실 함수

예측 방법: 이진 분류에서는 $P(Y=1) > 0.5$이면 1로 예측했습니다. 다중 클래스에서는 어떤 클래스의 확률도 0.5를 넘지 않을 수 있습니다. (예: $P(A) = 0.4, P(B) = 0.3, P(C) = 0.3$) 따라서 **가장 큰 예측 확률을 가진 클래스**를 최종 예측값으로 선택합니다.

데이터 불균형 문제: 만약 특정 클래스가 데이터의 대부분을 차지한다면(예: NFL 플레이의 66%가 '패스'), 모델은 예측 정확도를 높이기 위해 거의 모든 예측을 '패스'로 할 수 있습니다. (예: "코카인 사용자 예측" 예제) 이 경우, 모델이 단순히 다수 클래스만 예측하더라도 '분류 정확도'는 높게 나옵니다. 하지만 이 모델이 소수 클래스에 대한 유의미한 관계를 포착했을 수 있습니다. 따라서 단순 '분류' 결과뿐만 아니라 '확률' 자체를 보는 것이 중요합니다.

손실 함수: 이진 분류의 손실 함수를 **Binary Cross-Entropy**라고 불렀습니다. 다중 클래스 분류의 손실 함수는 이를 일반화한 **Cross-Entropy** (또는 Multinomial Logistic Loss)라고 부릅니다. 이 손실 함수에 Ridge(L2)나 Lasso(L1) 페널티 항을 추가하여 **정규화(Regularization)**를 수행할 수 있습니다.

3 분류 모델 평가 (Evaluating Classifiers)

모델을 만들었다면, 이 모델이 얼마나 좋은지 평가해야 합니다. 숫자 예측(회귀)에서 MSE를 쓴 것처럼, 분류 문제에도 전용 평가 지표가 필요합니다.

3.1 혼동 행렬 (The Confusion Matrix)

모든 분류 평가는 **혼동 행렬**에서 시작합니다. 이는 모델의 **예측 값**과 **실제 값**을 비교한 2x2 표입니다. (이진 분류 기준)

		예측된 값 (Predicted)	
		Negative (0)	**Positive (1)**
실제 값 (Actual)	**Negative (0)**	**True Negative (TN)**	**False Positive (FP)**
	Positive (1)	**False Negative (FN)**	**True Positive (TP)**

[title=혼동 행렬의 4가지 요소]

- **True Positive (TP):** 실제 1(Positive)인 것을 1로 올바르게 예측. (예: 스팸 메일을 스팸으로 분류)
- **True Negative (TN):** 실제 0(Negative)인 것을 0으로 올바르게 예측. (예: 일반 메일을 일반 메일로 분류)
- **False Positive (FP) / 1종 오류:** 실제 0(Negative)인 것을 1로 잘못 예측. (예: 일반 메일을 스팸으로 분류)
- **False Negative (FN) / 2종 오류:** 실제 1(Positive)인 것을 0으로 잘못 예측. (예: 스팸 메일을 일반 메일로 분류)

3.2 임계값(Threshold)과 성능의 트레이드오프

로지스틱 회귀 모델은 '분류'(0 또는 1)를 직접 출력하는 것이 아니라 '확률'(예: 0.7)을 출력합니다. 우리는 이 확률을 **임계값(Threshold)**과 비교하여 최종 분류를 결정합니다. (보통 0.5 사용)

$$\hat{P}(Y = 1) > \text{threshold} \implies \text{Predict } 1$$

이 임계값을 조절하면 모델의 특성이 바뀝니다.

임계값(Threshold) 조절의 효과

- **임계값을 낮추면 (예: 0.4):** 모델이 '1'로 예측하기 쉬워집니다.
  - **장점:** 실제 1인 것을 놓치지 않습니다. (TP 증가, **FN 감소**)
  - **단점:** 실제 0인 것을 1로 오인합니다. (**FP 증가**)
  - **예시:** 암 진단 모델. 환자를 놓치는 것(FN)이 치명적이므로 임계값을 낮춰 민감하게 반응하도록 합니다. (재검사하더라도 일단 잡아냄)
- **임계값을 높이면 (예: 0.6):** 모델이 '1'로 예측하기 어려워집니다. (매우 확신할 때만 1로 예측)
  - **장점:** 실제 0인 것을 1로 오인하지 않습니다. (**FP 감소**)
  - **단점:** 실제 1인 것을 놓치게 됩니다. (TP 감소, **FN 증가**)

  - **예시:** 스팸 메일 필터. 일반 메일을 스팸으로 보내는 것(FP)이 매우 불편하므로 임계값을 높여 확실한 스팸만 걸러내도록 합니다.

결론: FN을 줄이면 FP가 늘어나고, FP를 줄이면 FN이 늘어나는 **트레이드오프(Trade-off)** 관계가 존재합니다.

3.3 ROC 곡선 (Receiver Operating Characteristic Curve)

"그렇다면, 수많은 임계값 중 어떤 것을 선택해야 할까요? 혹시 임계값에 상관없이 모델 자체의 성능을 평가할 수는 없을까요?"

이 질문에 답하는 것이 **ROC 곡선**입니다. ROC 곡선은 **모든 가능한 임계값**에 대해 모델의 성능을 그래프로 그린 것입니다.

- **Y축: True Positive Rate (TPR) / 민감도 (Sensitivity) / 재현율 (Recall)**

$$TPR = \frac{TP}{TP + FN}$$

  (실제 Positive 중에서 모델이 Positive라고 예측한 비율. 1에 가까울수록 좋음)

- **X축: False Positive Rate (FPR)**

$$FPR = \frac{FP}{FP + TN}$$

  (실제 Negative 중에서 모델이 Positive라고 잘못 예측한 비율. 0에 가까울수록 좋음)

ROC 곡선 해석하기

- **완벽한 분류기 (Perfect Classifier):** (0, 1) 지점을 지나는 곡선. (FPR=0이면서 TPR=1, 즉 모든 것을 완벽하게 분류함)
- **무작위 분류기 (Random Classifier):** (0, 0)에서 (1, 1)로 이어지는 대각선 (y=x). FPR 50%를 감수해야 TPR 50%를 얻는다는 의미로, 동전 던지기(무작위 추측)와 같습니다.
- **좋은 분류기 (Good Classifier):** 대각선보다 위쪽, 즉 **왼쪽 상단**에 최대한 가깝게(Hugging) 그려지는 곡선입니다. 이는 낮은 FPR(적은 오인)으로도 높은 TPR(많은 정답)을 달성한다는 의미입니다.

3.4 AUC (Area Under the Curve)

ROC 곡선은 모델의 전체적인 성능을 보여주지만, 두 모델의 곡선이 서로 교차하는 등 비교가 어려울 수 있습니다. **AUC(Area Under the Curve)**는 ROC 곡선 아래의 면적을 계산하여 모델의 성능을 **하나의 숫자**로 요약합니다.

- **AUC = 1.0:** 완벽한 분류기 (면적이 1x1 정사각형)
- **AUC = 0.5:** 무작위 분류기 (면적이 y=x 대각선 아래 삼각형)
- **AUC > 0.5:** 무작위보다 좋은 모델.

AUC는 임계값에 관계없이 모델이 얼마나 Positive와 Negative 샘플을 잘 구별하는지 나타내는 지표입니다. AUC가 높을수록 좋은 모델입니다.

4 베이즈 추론 (Bayesian Inference)

지금까지 우리는 **빈도주의(Frequentist)** 관점에서 통계를 다뤘습니다. 빈도주의에서는 모델 파라미터(β)가 '고정되어 있지만 알지 못하는 값'이라고 가정하고, 데이터를 사용해 이 값을 '추정'했습니다.

베이즈주의(Bayesian) 관점은 파라미터를 다르게 봅니다.

[title=베이즈 추론의 핵심] 베이즈 관점에서 파라미터(θ)는 고정된 값이 아니라, **불확실성을 가진 확률변수**입니다. 우리는 파라미터에 대한 **믿음의 분포(Distribution of Belief)**를 가지고 있으며, 데이터를 관찰함으로써 이 믿음을 **업데이트**합니다.

4.1 베이즈 정리 (Bayes' Theorem)

이 '믿음의 업데이트' 과정은 베이즈 정리를 통해 수학적으로 수행됩니다.

$$\underbrace{f(\theta|X)}_{\text{Posterior}} \propto \underbrace{f(X|\theta)}_{\text{Likelihood}} \cdot \underbrace{f(\theta)}_{\text{Prior}}$$

- **Prior (사전 확률)** $f(\theta)$**:** 데이터(X)를 보기 전, 파라미터 θ에 대해 우리가 가진 **초기 믿음**의 분포입니다. (예: "이 동전은 아마 공정할 거야" $\rightarrow p = 0.5$ 근처에 확률을 높게 부여)
- **Likelihood (가능도)** $f(X|\theta)$**:** '만약 파라미터가 θ라면, 우리가 가진 데이터 X가 관찰될 확률'입니다. (이는 빈도주의의 '가능도 함수'와 동일합니다.)
- **Posterior (사후 확률)** $f(\theta|X)$**:** 데이터(X)를 관찰한 후, 파라미터 θ에 대해 **업데이트된 믿음**의 분포입니다. 이 사후 확률은 우리의 '최종 결과물'입니다.

베이즈 추론의 과정

초기 믿음 (Prior) $\times$ 데이터의 증거 (Likelihood) $\implies$ 업데이트된 믿음 (Posterior)

5 베타-이항 모델 (The Beta-Binomial Model)

베이즈 추론의 가장 고전적인 예시인 '동전 뒤집기' 문제를 통해 베이즈 추론을 이해해 봅니다. 우리의 목표는 동전의 앞면이 나올 확률 p를 추정하는 것입니다. (단, p는 0.5가 아닐 수도 있습니다.)

5.1 1단계: 가능도 (Likelihood) - 데이터가 말하는 것

동전을 n번 던져 앞면(Success)이 Σx_i번, 뒷면(Failure)이 $n - \Sigma x_i$번 나왔다고 합시다. 파라미터 p가 주어졌을 때 이 데이터가 관찰될 확률(가능도)은 **이항 분포(Binomial Distribution)**를 따릅니다.

$$f(X|p) \propto p^{\Sigma x_i}(1-p)^{n-\Sigma x_i}$$

(앞면이 나온 횟수만큼 p가, 뒷면이 나온 횟수만큼 $(1-p)$가 곱해집니다.)

5.2 2단계: 사전 확률 (Prior) - 우리의 초기 믿음

이제 p에 대한 우리의 초기 믿음을 설정해야 합니다. p는 확률값이므로 [0, 1] 사이의 분포여야 합니다. 이때 **베타 분포(Beta Distribution)**가 사용됩니다.

[title=베타 분포 $Beta(\alpha, \beta)$] 베타 분포는 [0, 1] 사이의 값을 가지며, 두 개의 **하이퍼파라미터(hyperparameter)** α와 β에 의해 모양이 결정됩니다.

$$f(p|\alpha, \beta) \propto p^{\alpha-1}(1-p)^{\beta-1}$$

직관적 해석: 베타 분포는 "과거에 $\alpha-1$번의 성공과 $\beta-1$번의 실패를 본 것과 같은 믿음"을 나타냅니다.

- $E[p] = \frac{\alpha}{\alpha+\beta}$ (분포의 평균)
- $Beta(1,1) : f(p) \propto p^0(1-p)^0 = 1$. **균등 분포(Uniform Distribution)**와 같습니다. "나는 p에 대해 아무것도 모르며, 모든 p 값이 똑같이 가능하다"는 의미의 **무정보 사전 확률(non-informative prior)**입니다.
- $Beta(10,10) : E[p] = \frac{10}{20} = 0.5$. "$p$는 0.5일 것이라고 강하게 믿는다" (공정한 동전)
- $Beta(2,5) : E[p] = \frac{2}{7} \approx 0.28$. "뒷면이 더 잘 나오는 동전 같다"

5.3 3단계: 사후 확률 (Posterior) - 업데이트된 믿음

베이즈 정리에 따라 사전 확률과 가능도를 곱합니다.

$$\text{Posterior} \propto \text{Likelihood} \times \text{Prior}$$

$$f(p|X) \propto [p^{\Sigma x_i}(1-p)^{n-\Sigma x_i}] \times [p^{\alpha-1}(1-p)^{\beta-1}]$$

지수 법칙에 따라 같은 밑을 가진 항들을 합칩니다.

$$f(p|X) \propto p^{(\alpha+\Sigma x_i)-1} \cdot (1-p)^{(\beta+n-\Sigma x_i)-1}$$

놀라운 결과: 이 사후 확률의 형태는 또 다른 **베타 분포**입니다! 우리는 $Beta(\alpha, \beta)$ 사전 확률로 시작했는데, 데이터를 반영하니 $Beta(\alpha + \text{성공 횟수}, \beta + \text{실패 횟수})$라는 새로운 베타 분포가 되었습니다.

베타-이항 모델 업데이트 규칙

- **Prior:** $p \sim Beta(\alpha, \beta)$
- **Data:** n번 시도, 성공 Σx_i번, 실패 $(n - \Sigma x_i)$번
- **Posterior:** $p|X \sim Beta(\alpha + \Sigma x_i, \beta + n - \Sigma x_i)$

이처럼 사전 확률과 사후 확률이 동일한 분포족(distribution family)에 속할 때, 이 사전 확률을 **켤레 사전 확률(Conjugate Prior)**이라고 부릅니다. 베타 분포는 이항/베르누이 분포의 켤레 사전 확률입니다.

□ 예제: title

- **1. 사전 믿음 (Prior):** 동전에 대한 정보가 전혀 없어 $Beta(1,1)$ (균등 분포)을 사용합니다. (사전 성공 횟수=0, 사전 실패 횟수=0으로 해석 가능)
- **2. 데이터 (Data):** 동전을 10번 던져 앞면(성공)이 7번, 뒷면(실패)이 3번 나왔습니다. ($n = 10, \Sigma x_i = 7$)
- **3. 사후 믿음 (Posterior):** 우리의 믿음은 $Beta(1+7, 1+3) = Beta(8,4)$로 업데이트됩니다.
  - 데이터 반영 전, p의 기댓값: $E[p] = \frac{1}{1+1} = 0.5$
  - 데이터 반영 후, p의 기댓값: $E[p] = \frac{8}{8+4} = \frac{8}{12} \approx 0.67$

  데이터를 통해 우리의 믿음이 0.5에서 0.67로 이동했습니다.

6 베이즈 로지스틱 회귀와 계층 모델

6.1 베이즈 로지스틱 회귀 (Bayesian Logistic Regression)

"그렇다면 로지스틱 회귀에도 베타 분포를 사전 확률로 쓸 수 있을까요?"

NOPE!

아니요, 쓸 수 없습니다.

- 베타 분포는 [0, 1] 사이의 확률 p 자체에 대한 사전 확률입니다.
- 로지스틱 회귀의 파라미터는 p가 아니라, $\beta_0, \beta_1, \ldots$ 계수들입니다.
- β 계수들은 $(-\infty, \infty)$ 범위의 실수 값을 가질 수 있습니다.

따라서 로지스틱 회귀의 β 계수들에 대한 사전 확률로는 [0, 1] 범위의 베타 분포가 아니라, $(-\infty, \infty)$ 범위의 **정규 분포(Normal Distribution)** (또는 라플라스 분포 등)를 사용합니다.

$$\beta_j \sim N(\mu_0, \sigma^2)$$

만약 우리가 $\mu_0 = 0$으로 설정한다면, 이는 "β_j 계수는 아마 0에 가까울 것이다(즉, X_j는 Y에 영향이 없을 것이다)"라는 사전 믿음을 주는 것입니다. 이는 파라미터를 0으로 축소시키는 **Ridge (L2) 정규화**와 매우 유사한 베이즈적 접근 방식입니다.

6.2 계층 모델 (Hierarchical Modeling) 미리보기

베이즈 추론은 데이터의 구조가 복잡할 때 더욱 강력한 힘을 발휘합니다. 우리는 파라미터 θ를 모델링하기 위해 하이퍼파라미터 α, β를 사용했습니다.

$$Y \leftarrow p \leftarrow Beta(\alpha, \beta)$$

만약 α, β 값 자체를 정하는 것이 불확실하다면? α, β에도 사전 확률을 부여할 수 있습니다. (예: $\alpha \sim Gamma(\ldots)$) 이를 **하이퍼-사전확률(Hyperprior)**이라고 부르며, 이렇게 모델이 여러 층(level)을 가지는 것을 **계층 모델(Hierarchical Model)**이라고 합니다.

"왜 이렇게 복잡하게 모델링하나요?" 가장 큰 이유는 **데이터에 중첩된(nested) 구조**가 있기 때문입니다.

□ **예제: title**

데이터: Y_{ij} (선수 j의 i번째 슛), X_{ij} (슛 거리)
목표: 슛 거리에 따른 성공 확률(p_{ij})을 모델링

$$\log\left(\frac{p_{ij}}{1-p_{ij}}\right) = \alpha_j + \beta_1 X_{ij}$$

여기서 α_j는 선수 j의 고유한 '기본 슛 성공 능력'을 나타내는 절편입니다.
접근 1 (모델 없음): 모든 선수가 같다고 가정. ($\alpha_j = \alpha_0$) → 나쁨. **접근 2 (독립 모델):** 선수마다 α_j를 따로 추정. → 슛을 적게 쏜 선수의 데이터는 불안정함.
접근 3 (계층 모델):

- **Level 1 (데이터):** 각 선수의 슛은 그 선수의 능력(α_j)에 따라 결정됨.

$$\log\left(\frac{p_{ij}}{1-p_{ij}}\right) = \alpha_j + \beta_1 X_{ij}$$

- **Level 2 (선수):** 개별 선수의 능력(α_j)은 완전히 제멋대로가 아니라, 'NBA 선수 전체의 능력 분포'에서 샘플링된 값이라고 가정합니다.

$$\alpha_j \sim N(\alpha_{\text{league}}, \sigma_\alpha^2)$$

(모든 α_j는 리그 평균 α_{league}을 중심으로 σ_α^2 만큼 흩어져 있다)

장점: 이 모델은 '정보를 공유(Share information)'합니다. 슛을 많이 쏜 선수(예: 르브론 제임스)는 α_j가 자신의 데이터에 의해 결정됩니다. 하지만 슛을 10번만 쏜 신인 선수는, 그 10개의 데이터와 '리그 평균'(α_{league}) 사이의 가중 평균으로 α_j가 추정됩니다. 즉, 데이터가 부족한 관측치(신인 선수)의 추정값을 리그 평균 쪽으로 당겨와(shrink) 더 안정적인 추론을 가능하게 합니다.