

MIT 6.86x: 예측의 미학

Unit 1: Introduction to Machine Learning

Contents

1	Unit 1. 학습의 기초 (Fundamentals of Learning)	2
1.1	1. 학습 문제의 정의 (Formalizing the Problem)	2
1.2	2. 가설 공간 (Hypothesis Space, \mathcal{H})	3
1.3	3. 지도 학습 vs 비지도 학습	3
1.4	4. 일반화 (Generalization)	4
2	실전 시나리오: 넥슨 게임 이탈자 예측	5
3	자주 묻는 질문 (FAQ)	5

Course Structure Current Focus

- Unit 1: Introduction to ML (현재 단원: 운동장 세팅)
 - 1.1 Formalizing the Problem (함수 근사 문제)
 - 1.2 Hypothesis Space (\mathcal{H})
 - 1.3 Learning Paradigms (지도 vs 비지도)
 - 1.4 Generalization (E_{in} vs E_{out})
- Unit 2: Linear Classifiers (Perceptron)
- Unit 3: Neural Networks

1 Unit 1. 학습의 기초 (Fundamentals of Learning)

우리는 18.6501(통계학)에서 ”데이터가 어떤 분포에서 나왔는가(*Inference*)?”를 고민했습니다. 6.86x(머신러닝)에서는 질문을 바꿉니다. ”그래서, 내일 주가가 오를까 내릴까(*Prediction*)?” 분포의 모양보다는, 정답을 맞히는 성능에 목숨을 거는 새로운 여정이 시작됩니다.

□ 개요 (Overview)

이 단원은 코드를 짜기 전, 머신러닝 문제를 수학적으로 정의하는 단계입니다. 머신러닝을 ”미지의 목표 함수 f^* 를 찾기 위해 가설 공간 \mathcal{H} 를 탐색하는 과정”으로 정의하고, 학습의 궁극적 목표인 일반화(Generalization)**의 개념을 확립합니다.

□ 핵심 용어 사전

용어 (Term)	직관적 의미 (Meaning)
Target Function (f^*)	신(God)만이 아는 진짜 정답 규칙. 우리는 이걸 모른다.
Hypothesis (h)	우리가 f^* 라고 추측한 모델(가설).
Hypothesis Space (\mathcal{H})	h 를 찾을 수 있는 범위(예: 직선들, 신경망들).
Inductive Bias	데이터를 보기도 전에 정해둔 편견(예: ”답은 직선일 거야”).
Generalization	안 배운 문제(Test Data)도 잘 맞히는 능력.

1.1 1. 학습 문제의 정의 (Formalizing the Problem)

□ 개념 1: 함수 근사 (Function Approximation)

한 줄 요약: 머신러닝은 입력(x)과 출력(y) 사이의 숨겨진 관계식(f^*)을 데이터(S_n)를 통해 역추적하는 과정입니다.

1) 직관적 비유: 장인의 레시피 복원

유명 맛집의 비법 소스(f^*)가 있습니다. 사장님은 레시피를 절대 안 알려줍니다.

- **Data (S_n):** 우리가 맛볼 수 있는 건 결과물(음식) 뿐입니다.
- **Goal:** 맛을 보고 재료와 비율을 역추적해서, 사장님의 맛과 거의 똑같은 맛을 내는 나만의 레시피(h)를 만드는 것입니다.

2) 수학적 구성 요소

- **Input (\mathcal{X}):** 재료 (벡터 $x \in \mathbb{R}^d$)
 - **Output (\mathcal{Y}):** 맛 (값 y)
 - **Unknown Target (f^*):** $f^* : \mathcal{X} \rightarrow \mathcal{Y}$. (이상적인 정답 규칙)
 - **Data (S_n):** $\{(x^{(i)}, y^{(i)})\}_{i=1}^n$. (f^* 에 노이즈가 섞인 관측치)
 - **Final Hypothesis (h):** 우리가 찾은 최적의 함수. $h \approx f^*$ 이기를 희망함.
-

1.2 2. 가설 공간 (Hypothesis Space, \mathcal{H})

□ 개념 2: 수색 범위 설정

한 줄 요약: ”세상의 모든 함수” 중에서 정답을 찾을 수는 없습니다. ”직선 중에서 찾자” 혹은 ”곡선 중에서 찾자”처럼 탐색 범위를 제한해야 합니다.

1) 귀납적 편향 (Inductive Bias)

가설 공간 \mathcal{H} 를 정하는 순간, 우리는 데이터에 대한 편견(Bias)을 갖게 됩니다.

- 선형 회귀를 쓴다면? → ”세상은 선형적일 거야”라는 편견.
- 편견이 나쁜가요? 아니요, 필수입니다. 편견(제약 조건)이 없으면, 데이터 점들을 잇는 방법이 무한히 많아서 학습 자체가 불가능합니다.

2) Trade-off

- \mathcal{H} 가 너무 작음 (예: 상수 함수): 정답을 표현 못 함 (Underfitting).
 - \mathcal{H} 가 너무 큼 (예: 100차 다항식): 노이즈까지 다 외워버림 (Overfitting).
-

1.3 3. 지도 학습 vs 비지도 학습

□ 개념 3: 정답지의 유무

한 줄 요약: 문제집 뒤에 해설지가 있으면 지도 학습, 해설지 없이 문제들의 유형을 스스로 정리해야 하면 비지도 학습입니다.

1) 지도 학습 (Supervised)

- 데이터: (x, y) 쌍.
- 목표: $h(x) \approx y$ 인 h 를 찾음. (회귀, 분류)

2) 비지도 학습 (Unsupervised)

- 데이터: x 만 있음.
- 목표: 데이터의 구조, 패턴, 군집을 찾음. (클러스터링, 차원 축소)
- 예시: 네슨 유저들의 행동 로그(x)만 보고 ”이들은 하드코어 유저군이다”라고 그룹핑하기.

1.4 4. 일반화 (Generalization)

□ 개념 4: 머신러닝의 존재 이유

한 줄 요약: 연습문제(Training Data)를 100점 맞는 건 의미 없습니다. 실전 수능(Unseen Data)을 잘 보는 것이 진짜 목표입니다.

1) 두 가지 위험 (Risk)

- 경험적 위험 (E_{in} , Empirical Risk): 지금 가진 학습 데이터(S_n)에서의 오차. (모의고사 성적)
- 실제 위험 (E_{out} , True Risk): 전체 데이터 분포(\mathcal{D})에서의 기대 오차. (수능 성적)

2) 근본적인 긴장 관계 (Fundamental Tension)

우리는 알고리즘을 통해 E_{in} 을 줄입니다. 하지만 E_{in} 을 0으로 만든다고 E_{out} 이 0이 될까요?

- Overfitting: $E_{in} \approx 0$ 이지만 E_{out} 은 매우 큰 상태. (답을 달달 외워서 응용을 못함)
- Goal: E_{in} 도 작게 하면서, 동시에 $E_{out} \approx E_{in}$ 이 되도록 보장하는 것. 이것이 6.86x 과정의 핵심 질문입니다.

2 실전 시나리오: 넥슨 게임 이탈자 예측

□ Scenario: 완벽한 예측 모델의 함정

당신은 유저 이탈 예측 모델을 만들었습니다. 과거 1년 치 데이터(S_n)로 학습시켰더니, 정확도 99.9%($E_{in} \approx 0$)가 나왔습니다. ”완벽해!”라고 외치며 실 서버에 배포했습니다.

1. 결과: 다음 달, 모델은 이탈자를 하나도 못 맞췄습니다. (E_{out} 폭망)
2. 원인 분석: 모델을 뜯어보니 이런 규칙이 있었습니다. ”아이디가 'User1234'이고 3월 5일에 접속한 사람은 이탈한다.”
3. 해석: 가설 공간 \mathcal{H} 를 너무 복잡하게 잡아서, 유저의 행동 패턴(f^*)을 배운 게 아니라 특정 유저의 ID와 날짜(노이즈)를 외워버린 것입니다.
4. 해결: 모델을 단순화(Regularization)하거나 데이터를 더 모아서 일반화 성능(E_{out})을 챙겨야 합니다.

3 자주 묻는 질문 (FAQ)

- Q1. f^* 과 h 의 차이가 정확히 뭔가요?** A. f^* 은 ’진리(Truth)’이고 h 는 ’추측(Guess)’입니다. 예를 들어, 물리 법칙($F = ma$)은 f^* 입니다. 우리가 실험 데이터를 통해 $F = 0.98ma + 0.02$ 라는 식을 얻었다면 이것이 h 입니다. 우리는 영원히 f^* 를 완벽하게 알 수는 없고, h 를 f^* 에 가깝게 만들 뿐입니다.
- Q2. Inductive Bias가 왜 필요한가요?** A. 편견 없이는 학습도 없습니다(No Free Lunch Theorem). ”모든 것이 가능하다”는 말은 ”아무것도 알 수 없다”는 말과 같습니다. ”답은 연속적일 거야”, ”답은 간단할 거야” 같은 가정이 있어야만 유한한 데이터로부터 무한한 미래를 예측할 수 있습니다.

Next Step: ”일반화”가 중요하다는 것은 알았습니다. 그렇다면 가장 단순하면서도 강력한 가설 공간인 ’선형 모델(Linear Model)’부터 시작해볼까요? 다음 **Unit 2**에서는 직선 하나로 데이터를 분류하는 퍼셉트론(Perceptron) 알고리즘을 배웁니다.

Unit 1 핵심 요약

- **목표:** 데이터(S_n)를 이용해 미지의 함수 f^* 에 근사하는 h 를 찾는다.
- **가설 공간(\mathcal{H}):** 탐색 범위를 제한하는 것. 편향(Bias)을 동반한다.
- **일반화:** 학습 데이터 오차(E_{in})가 아니라, 보지 못한 데이터의 오차(E_{out})를 줄이는 것이 진짜 목표다.
- **Overfitting:** E_{in} 은 낮지만 E_{out} 이 높은 상태. (암기왕)