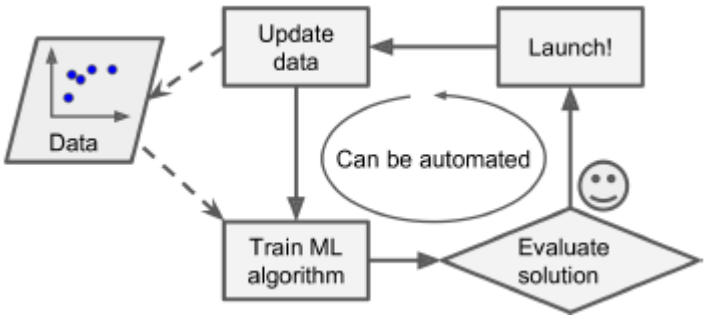
chapter. 1
01 한눈에 보는 머신러닝

머신러닝이란?

머신러닝
어떤 작업 **T**에 대한 컴퓨터 프로그램의 성능을 **P**로 측정했을 때 경험 **E**로 인해 성능이 향상됐다면, 이 컴퓨터 프로그램은 작업 **T**와 측정 **P**에 대해 경험 **E**로 학습한 것이다.

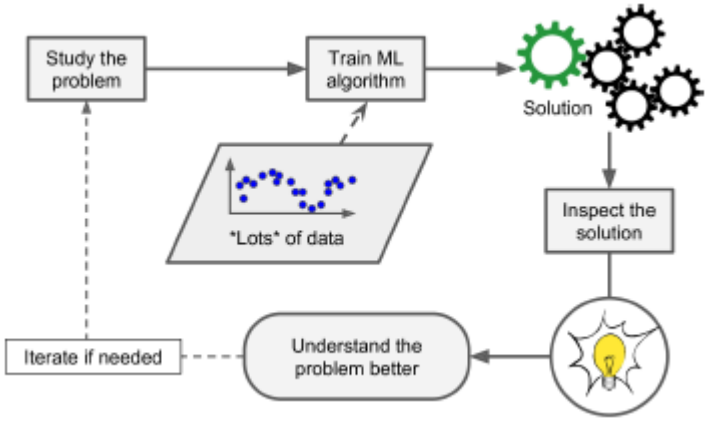
-톰 미첼(Tom Mitchell, 1997)-

훈련 세트(training set)
시스템이 학습하는 데 사용하는 샘플
훈련 사례(training instance) 또는 샘플
각 훈련 데이터
훈련 데이터(training data)
위 정의의 경험 E



→ 자동화된 머신러닝 접근

데이터 마이닝
머신러닝을 통해 배운다, 즉 머신러닝 알고리즘이 학습한 것을 조사.
머신러닝 기술을 적용해서 대용량의 데이터를 분석하면 겉으로는 보이지 않던 패턴을 발견 할 수 있다.



머신러닝이 뛰어난 분야

- 기존 솔루션으로 많은 수동 조정과 규칙이 필요한 문제
- 전통적인 방식으로는 해결 방법이 없는 복잡한 문제
- 유동적인 환경 (새로운 데이터에 적응) / 복잡한 문제와 대량의 데이터에서 통찰 얻기

머신러닝 시스템의 종류

지도 학습과 비지도 학습

'학습하는 동안의 감독 형태나 정보량'에 따라 분류할 수 있다.

지도 학습, 비지도 학습, 준지도 학습, 강화 학습 등 네 가지 주요 범주가 있다.

지도학습(supervised learning)

알고리즘에 주입하는 훈련 데이터에 **레이블(label)**이라는 원하는 답이 포함

→ 레이블의 범주를 **클래스(class)**라 한다.

- **분류(classification)**

전형적인 지도 학습 작업 ex) 스팸 필터

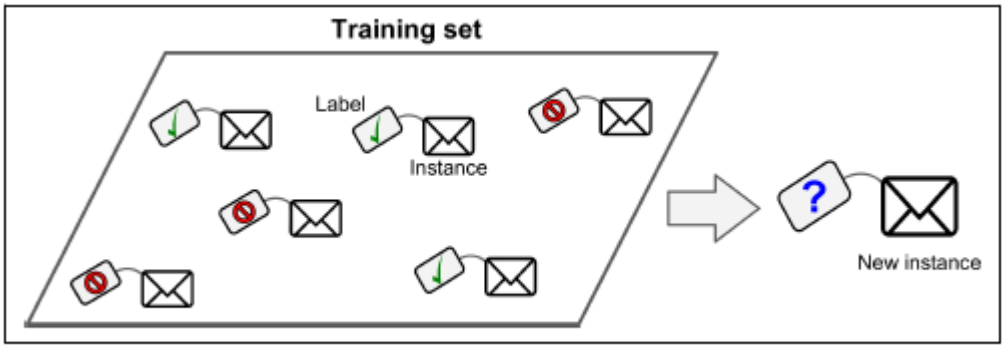


Figure 1-5. A labeled training set for supervised learning (e.g., spam classification)

- **회귀(regression)**

예측 변수(predictor variable)라 불르는 **특성(feature)**을 사용해 **타깃(target)** 수치를 예측

ex) ㄴ주행거리, 연식, 브랜드 등ㄴ ㄴ중고차 가격ㄴ

회귀 알고리즘을 분류에 사용하기도 하고, 분류 알고리즘을 회귀에 사용하기도 한다.

지도 학습 알고리즘

- k-최근접 이웃 (k-nearest neighbors)
- 선형 회귀 (linear regression)
- 로지스틱 회귀 (logistic regression)
- 서포트 벡터 머신 (support vector machine, SVM)
- 결정 트리 (decision tree)와 랜덤 포레스트(random forest)
- *신경망 (neural networks)

비지도 학습(unsupervised learning)

훈련 데이터에 레이블이 없음. 시스템이 아무런 도움 없이 학습해야 함.

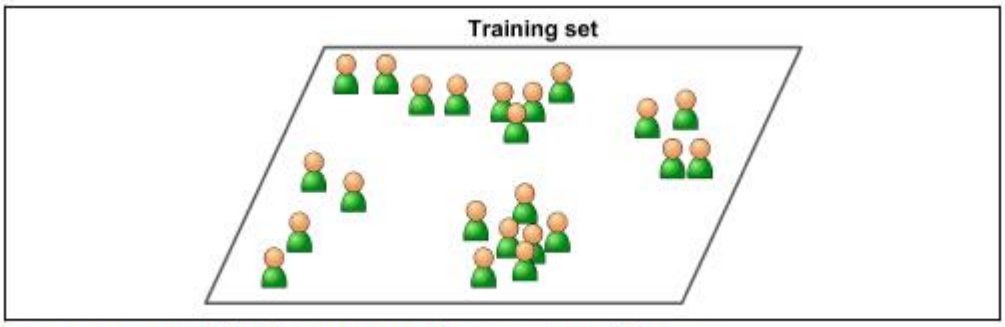


Figure 1-7. An unlabeled training set for unsupervised learning

비지도 학습 알고리즘

- 군집 (clustering)
  - k-평균 (k-means)
  - DBSCAN
  - 계층 군집 분석 (hieararchical cluster analysis, HCA)
  - 이상치 탐지 (outiler detection)와 특이치 탐지 (novelty detection)
  - 원-클래스 (one-class SVM)
  - 아이솔레이션 포레스트 (isolation forest)
- 시각화 (visualization)와 차원 축소 (dimensionality reduction)
  - 주성분 분석(principal component analysis, PCA)
  - 커널 (kernel) PCA
  - 지역적 선형 임베딩 (locally-linear embedding, LLE)
  - t-SNE (t-distributed stochastic neighbor embedding)
- 연관 규칙 학습 (association rule learning)
  - 어프라이어리 (Apriori)
  - 이클렛 (Eclat)

계층 군집(hierarchical clustering)

각 그룹을 더 작은 그룹으로 세분화할 수 있다.

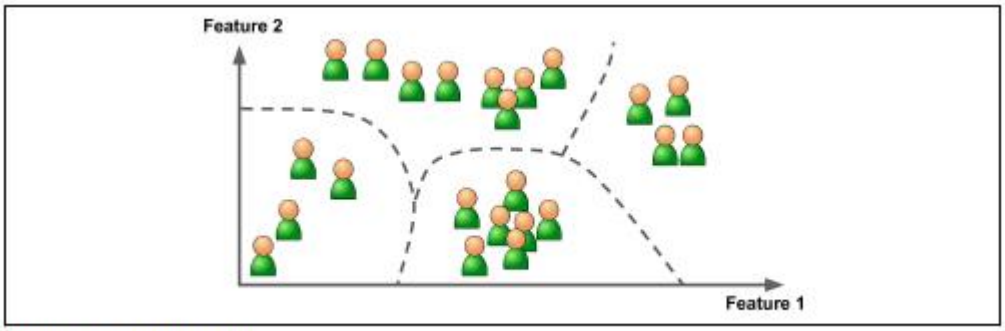


Figure 1-8. Clustering

시각화(visualization)

레이블이 없는 대규모의 데이터를 넣어 도식화가 가능한 2D나 3D 표현을 만들어준다.

이로써 데이터가 어떻게 조직되어 있는지 이해할 수 있고 예상하지 못한 패턴을 발견할 수도 있습니다.

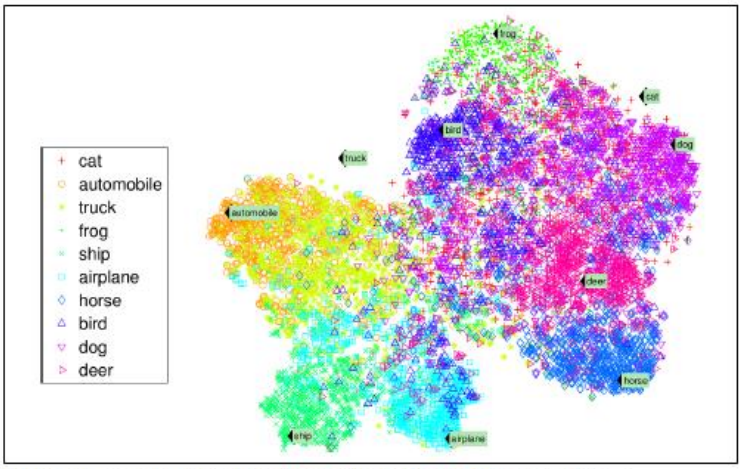


Figure 1-9. Example of a t-SNE visualization highlighting semantic clusters[3]

차원축소(dimensionality reduction)

너무 많은 정보를 잃지 않으면서 데이터를 간소화

→ 실행속도를 빠르게 하고, 디스크, 메모리 자원을 절약할 수 있다. → 경우에 따라 성능 향상

차원축소의 한가지 방법 : 특성 추출(feature extraction)

상관관계가 있는 여러 특성을 하나로 합치는 것.

ex) 차의 주행거리, 연식은 강하게 연관되므로 하나의 특성으로 합침

이상치 탐지(outlier detection) (원서: anomaly detection)

시스템은 훈련하는 동안 대부분 정상 샘플을 만나 이를 인식하도록 훈련됨.

그 다음 새로운 샘플을 보고 정상 데이터인지 혹은 이상치인지 판단.

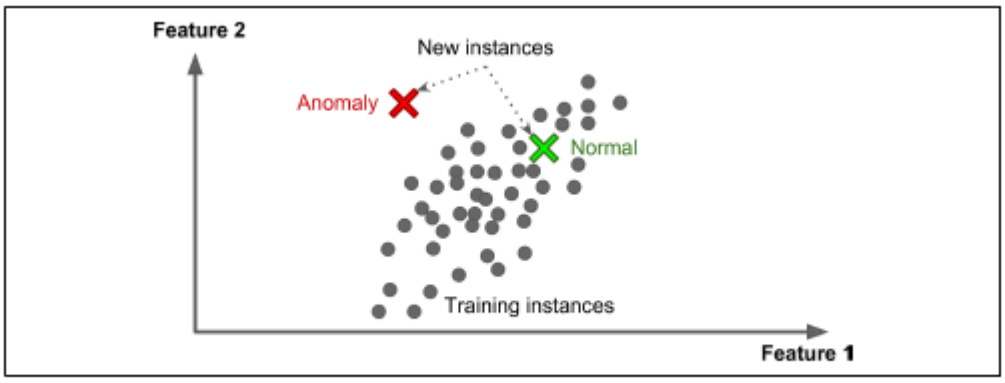
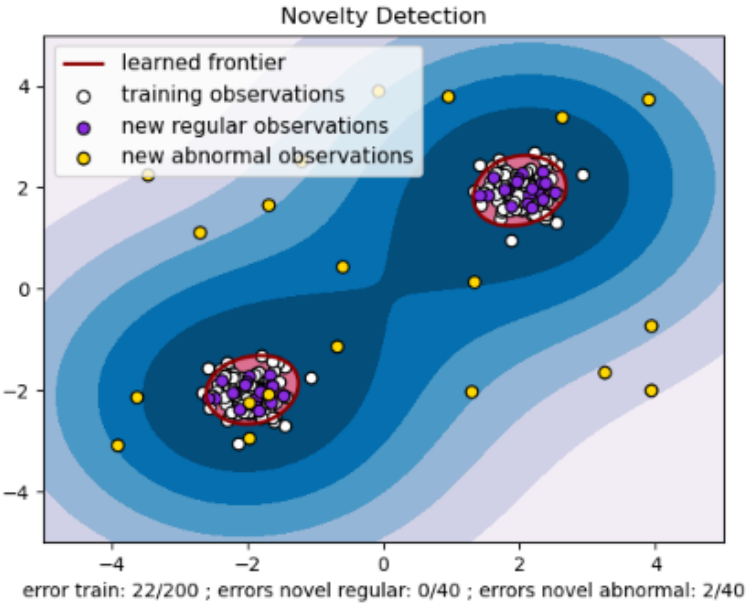


Figure 1-10. Anomaly detection

특이치 탐지(novelty detection)
훈련 세트에 있는 모든 샘플과 달라 보이는 새로운 샘플을 탐지하는 것이 목적.
알고리즘으로 감지하고 싶은 모든 샘플을 제거한 매우 '깨끗한' 훈련 세트가 필요



ex) 강아지 사진 수천장 중 1%가 치와와 사진
특이치 탐지: 치와와 사진을 새로운 특이한 것으로 처리하지 못함
이상치 탐지: 이 강아지 사진을 매우 드물고 다른 강아지와 다르다고 인식하여 이상치로 분류

이상치 탐지와 특이치 탐지의 차이 (원서)
the difference is that novelty detection algorithms expect to see only normal data during training, while anomaly detection algorithms are usually more tolerant, they can often perform well even with a small percentage of outliers in the training set.

연관 규칙 학습(association rule learning)
대량의 데이터에서 특성 간의 흥미로운 관계를 찾는 것

준지도 학습(semisupervised learning)
일부만 레이블이 있는 데이터를 다룬다.

대부분의 준지도 학습 알고리즘은 지도 학습과 비지도 학습의 조합으로 이루어져 있다.

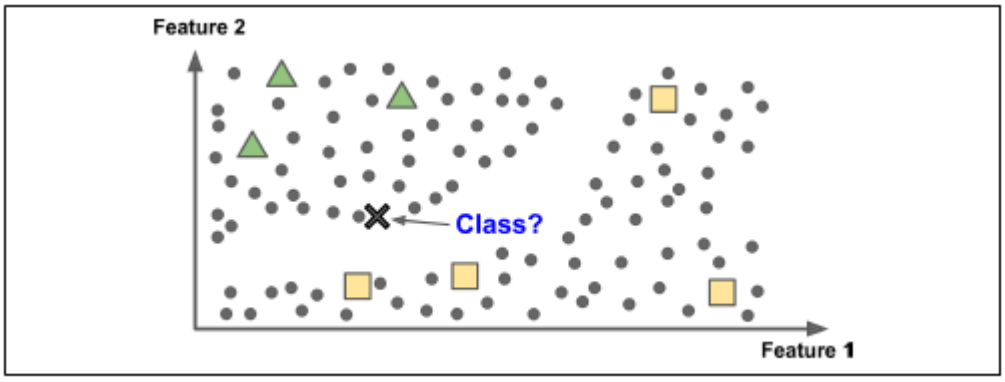


Figure 1-11. Semisupervised learning

→ 사각형 클래스에 더 가깝지만 레이블된 자료들이 해당 X를 삼각형으로 분류하는데 도움을 준다.

강화 학습(reinforcement learning)

매우 다른 종류의 알고리즘

에이전트: 학습하는 시스템

환경(environment)을 관찰해서 **행동(action)**을 실행하고 그 결과로 **보상(reward)** 또는 **벌점(penalty)**을 받습니다. 시간이 지나며 가장 큰 보상을 얻기 위해 **정책(policy)**이라고 부르는 최상의 전략을 스스로 학습합니다.

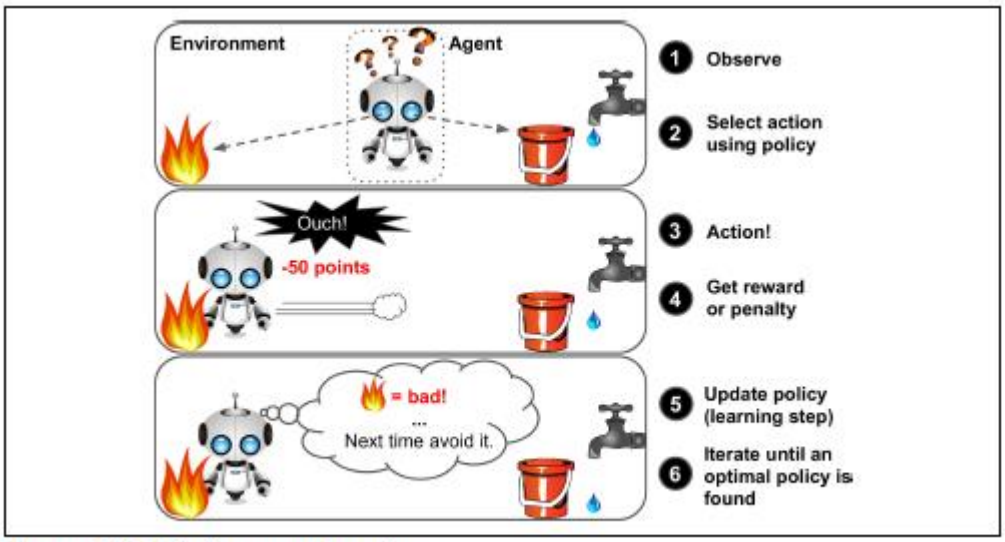


Figure 1-12. Reinforcement Learning

배치 학습과 온라인 학습

입력 데이터의 스트림(stream)으로부터 점진적으로 학습할 수 있는지 여부

배치 학습(batch learning) / 오프라인 학습(offline learning)

시스템이 점진적으로 학습할 수 없다.
가용한 데이터를 모두 사용해 훈련시킴 → 일반적으로 자원을 많이 소모하므로 오프라인에서 수행
먼저 시스템을 훈련시키고 그런 다음 제품 시스템에 적용하면 더 이상의 학습없이 실행된다.
즉, 학습한 것을 단지 적용만 한다. 이를 **오프라인 학습(offline learning)**이라고 한다.

온라인 학습(online learning)

데이터를 순차적으로 한 개씩 또는 **미니배치(mini-batch)**라 부르는 작은 묶음 단위로 주입하여 시스템을 훈련
학습 단계가 빠르고 비용이 적게 들어 데이터 도착 즉시 학습 가능.
새로운 데이터 샘플을 학습하면 학습이 끝난 데이터는 더는 필요하지 않으므로 버림. → 공간절약

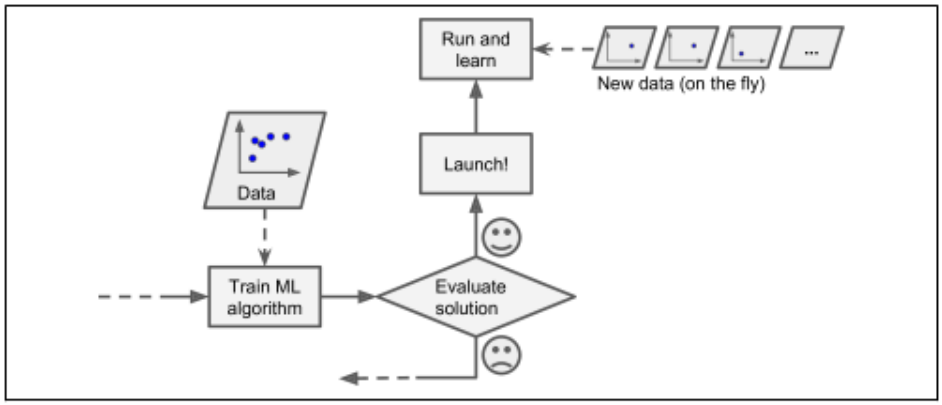


Figure 1-13. Online learning

%%%

외부 메모리(out-of-core) 학습 → 온라인 학습 알고리즘 사용하지만 보통 오프라인으로 실행
아주 큰 데이터셋을 학습시 알고리즘이 데이터 일부를 읽어 들이고 훈련 단계를 수행한다.
→ 전체 데이터가 모두 적용될 때까지 이 과정을 반복
실시간 시스템에서 수행되는 것이 아니므로 **점진적 학습(incremental learning)**이라고 생각

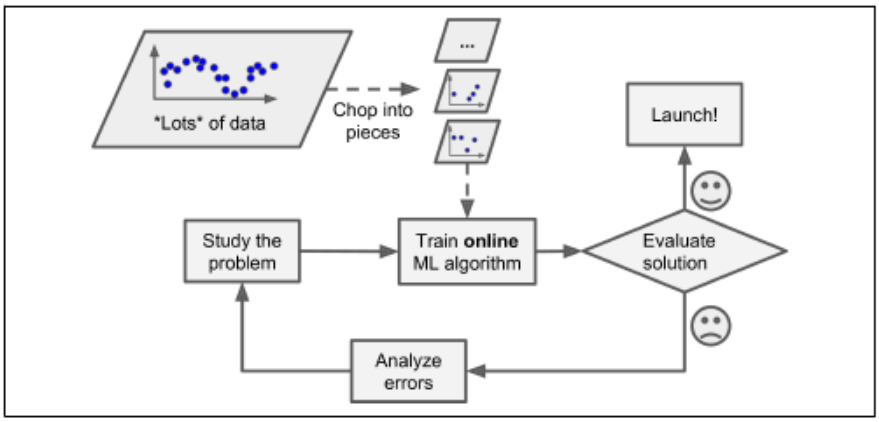


Figure 1-14. Using online learning to handle huge datasets

%%%

학습률(learning rate)
변화하는 데이터에 얼마나 빠르게 적응할 것인가 (온라인 학습 시스템에서 중요한 파라미터)

학습률을 높게 함 → 시스템이 데이터에 빠르게 적응 & 에전 데이터를 금방 잊음
학습률을 낮게 함 → 시스템 관성이 증가하여 더 느리게 학습 & 새로운 데이터의 잡음이나 대표성 없는 데이터 포인트에 덜 민감해짐

온라인 학습의 문제점
시스템에 나쁜 데이터가 주입되었을 때 시스템 성능이 점진적으로 감소

사례 기반 학습과 모델 기반 학습

어떻게 **일반화(generalize)**되는가에 따라 분류

사례 기반 학습(instance-based learning)

시스템이 훈련 샘플을 기억함으로써 학습합니다. 그리고 **유사도(similarity) 측정(measure)**을 사용해 새로운 데이터와 학습한 샘플을 (또는 학습한 샘플 중 일부를) 비교하는 식으로 일반화합니다.

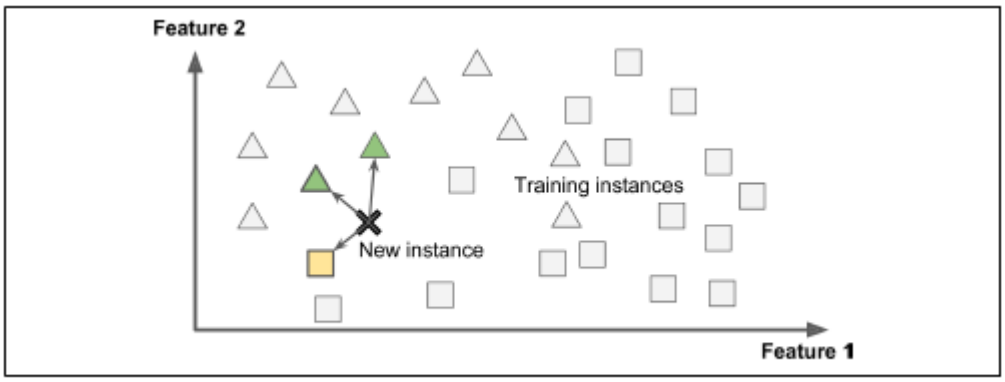


Figure 1-15. Instance-based learning

→ New instance의 가장 비슷한 샘플 중 다수가 삼각형이므로 삼각형으로 분류

모델 기반 학습(model-based learning)

샘플들의 모델을 만들어 **예측(prediction)**에 사용하는 것

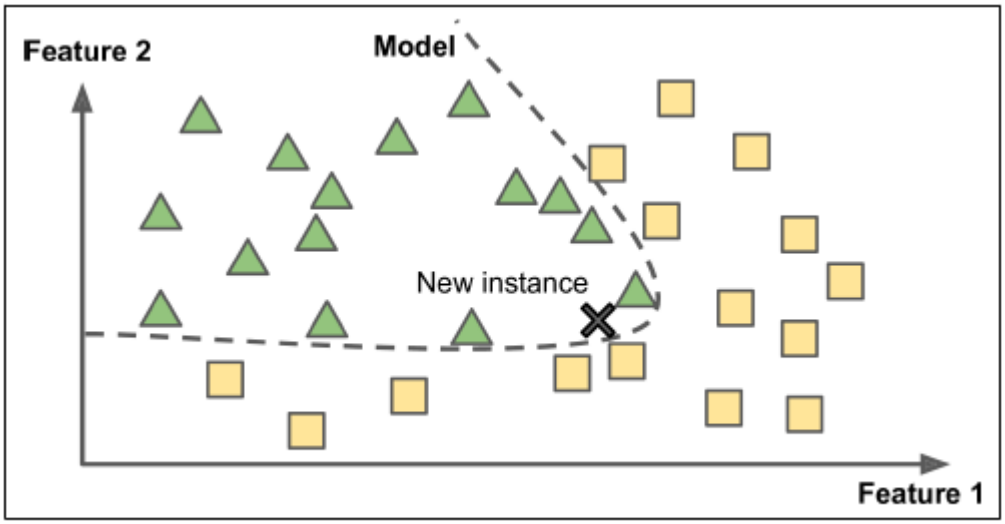


Figure 1-16. Model-based learning

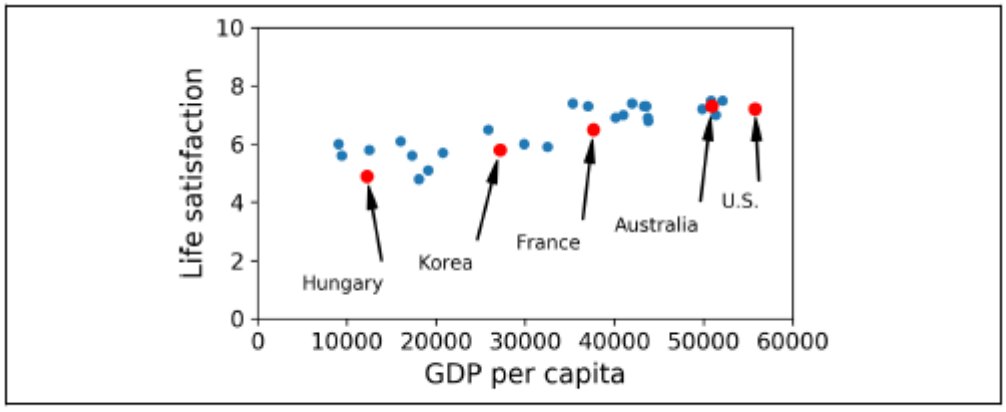


Figure 1-17. Do you see a trend here?

ex)

데이터가 흩어져 있음 → y가 x가 증가할수록 선형으로 같이 올라감 → 선형 함수로 삼의 만족도를 모델링 → 이 단계를 **모델 선택(model selection)**이라고 한다.

모델 파라미터(model parameter)를 정의하는데 모델이 최상의 성능을 내도록 하는 값을 알기 위해 측정 지표를 정한다.
모델이 얼마나 좋은지 측정하는 **효용 함수(utility function)** (또는 **적합도 함수(fitness function)**)를 정의하거나 얼마나 나쁜지 측정하는 **비용 함수(cost function)**를 정의할 수 있다.

ex) 선형회귀에서는 보통 선형 모델의 예측과 훈련 데이터 사이의 거리를 재는 비용 함수(최소화 목표)를 사용

선형 회귀(linear regression) 알고리즘 → 알고리즘에 훈련 데이터를 공급하면 데이터에 가장 잘 맞는 선형 모델의 파라미터를 찾는다. → 이를 모델을 **훈련(training)**시킨다고 말한다.

모델 기반 학습 작업 과정

1. 데이터 분석
2. 모델 선택
3. 훈련 데이터로 모델을 훈련(학습 알고리즘이 비용 함수를 최소화하는 모델 파라미터를 찾음)
4. 새로운 데이터에 모델을 적용해 예측을 하고 **(추론, inference**), 이 모델이 일반화되길 기대.

머신러닝의 주요 도전 과제

1. 나쁜 데이터
2. 나쁜 알고리즘

1. 나쁜 데이터

충분하지 않은 양의 훈련 데이터

대부분의 머신러닝 알고리즘이 잘 작동하려면 데이터가 많아야 한다.

대표성 없는 훈련 데이터

일반화가 잘되려면 우리가 일반화하기 원하는 새로운 사례를 훈련 데이터가 잘 대표하는 것이 중요.
샘플이 작으면 **샘플링 잡음(sampling noise,** 우연에 의한 대표성 없는 데이터)가 생기고
매우 큰 샘플도 표본 추출 방법이 잘못되면 대표성을 띠지 못할 수 있습니다. → **샘플링 편향(sampling bias)**

낮은 품질의 데이터

훈련 데이터가 에러, 이상치(outlier), 잡음이 가득하다면 내재된 패턴을 찾기 어려워 잘 작동하지 않음
→ 훈련 데이터 정제에 시간을 투자할 만한 가치는 충분

훈련 데이터 정제 예)

- 이상치로 판단된 샘플을 고치거나 무시
- 몇 개의 특성이 빠져있는 샘플이 있을 경우, 이 특성에 대한 판단하기(무시, 채우기, 따로 훈련)

관련 없는 특성

훈련 데이터에 관련없는 특성이 적고 관련 있는 특성이 충분해야 시스템이 학습할 수 있을 것이다.

머신러닝 프로젝트의 핵심 요소 → 훈련에 사용할 좋은 특성들을 찾는 것
이 과정을 **특성 공학(feature engineering)**이라 하며 다음과 같은 작업이다.

- **특성 선택(feature selection):** 가지고 있는 특성 중에서 훈련에 가장 유용한 특성을 선택
- **특성 추출(feature extraction):**특성을 결합하여 더 유용한 특성을 만듬. ex) 차원 축소
- 새로운 데이터를 수집해 새 특성을 만든다.

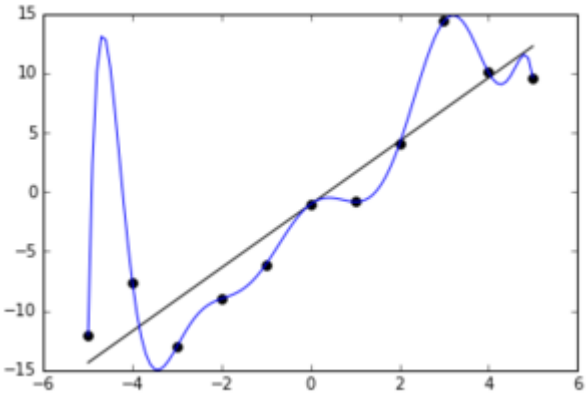
2. 나쁜 알고리즘

훈련 데이터 과대적합

과대적합(overfitting)

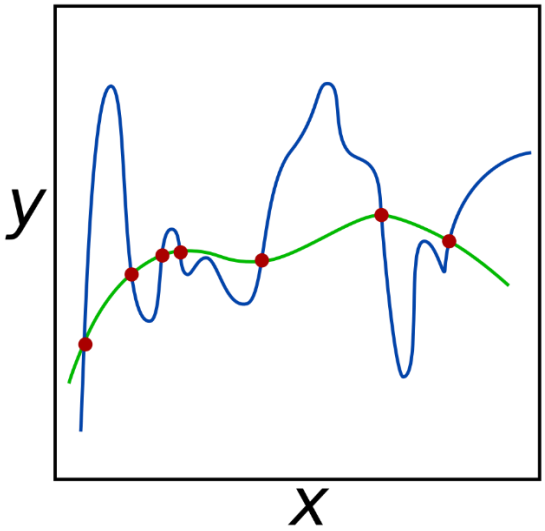
모델이 훈련 데이터에 너무 잘 맞지만 일반성이 떨어진다.
해결 방법

- 파라미터 수가 적은 모델을 선택(ex: 고차원 다항 모델 → 선형 모델), 훈련 데이터에 있는 특성 수를 줄이거나, 모델에 제약을 가하여 단순화
- 훈련 데이터를 더 많이 모은다.
- 훈련 데이터의 잡음을 줄인다(ex: 오류 데이터 수정, 이상치 제거)



규제(regularization)

모델을 단순하게 하고 과대적합의 위험을 감소시키기 위해 모델에 제약을 가하는 것



학습하는 동안 적용할 규제의 양은 **하이퍼파라미터(hyperparameter)**가 결정한다.

하이퍼파라미터는 모델이 아니라 학습 알고리즘의 파라미터이다.

학습알고리즘으로부터 영향을 받지 않으며, 훈련 전에 미리 지정되고, 훈련하는 동안에는 상수로 남아있다.

규제 하이퍼파라미터를 매우 큰 값으로 지정(기울기가 0에 가까운) → 거의 평편한 모델

→ 과대적합가능성은 낮아지나 좋은 모델을 찾지 못함 → 하이퍼 파라미터 튜닝은 매우 중요한 과정

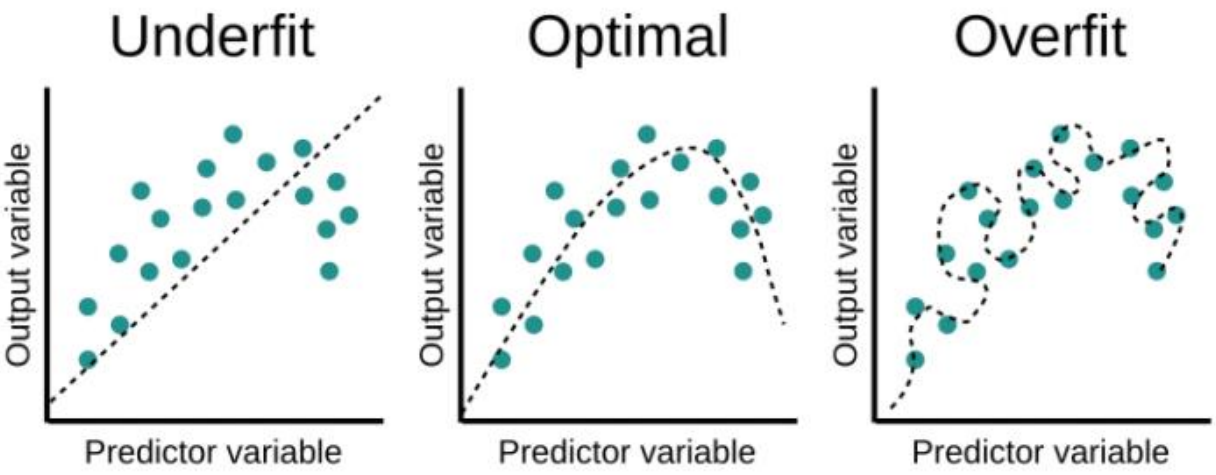
훈련 데이터 과소적합

과소적합(underfitting)

모델이 너무 단순해서 데이터의 내재된 구조를 학습하지 못함

해결하는 주요 기법

- 모델 파라미터가 더 많은 강력한 모델을 선택
- 학습 알고리즘에 더 좋은 특성을 제공(특성 공학)
- 모델의 제약을 줄임(ex) 규제 하이퍼파라미터를 감소시킴)


Underfit
Optimal
Overfit
Output variable
Predictor variable
Output variable
Predictor variable
Output variable
Predictor variable

테스트와 검증

훈련 데이터를 **훈련 세트**와 **테스트 세트** 두 개로 나눈다.
→**훈련 세트**를 사용해 모델을 **훈련**하고 **테스트 세트**를 사용해 모델을 **테스트**
일반화 오차(generalization error) or 외부 샘플 오차(out-of-sample error) : 새로운 샘플에 대한 오류 비율
→테스트 세트에서 모델을 평가함으로써 이 오차에 대한 **추정값(estimation)**을 얻는다.

훈련 오차가 낮지만(훈련 세트에서 모델의 오차가 적음) 일반화 오차가 높다 → 모델이 훈련 데이터에 과대적합

(주의)
모델 선택에 같은 테스트 세트를 반복해서 재사용하면 훈련 세트의 일부가 되는 셈이고 결국 모델은 과대적합될 것입니다. 이는 좋은 머신 러닝 작업 방식이 아닙니다. 테스트 세트는 별도로 보관하고 최종 모델을 평가하는 맨 마지막에 사용한다.

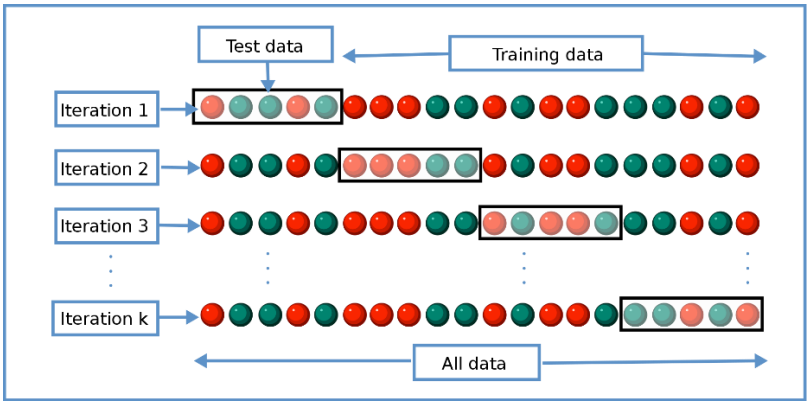
홀드아웃 검증(holdout validation)

훈련 세트의 일부를 떼어내어 후보 모델을 평가하고 가장 좋은 하나를 선택
→ 이 새로운 홀드 아웃 세트를 **검증 세트(validation set)**라고 부른다.
(**개발 세트(development set**, **데브 세트 (dev set)** 또는 **holdout cross validation set이라고도 함**)
→ 줄어든 훈련 세트(전체 훈련 세트에서 검증 세트를 뺀 데이터)에서 다양한 하이퍼파라미터 값을 가진 여러 모델을 훈련
→ 검증 세트에서 가장 높은 성능을 내는 모델을 선택
→ 최선의 모델을 (검증 세트를 포함한) 전체 훈련 세트에서 다시 훈련하여 최종 모델을 만듬
→ 최종 모델을 테스트 세트에서 평가하여 일반화 오차를 추정

<table>
<tr><td colspan="3">원본 데이터셋</td></tr>
<tr><td colspan="2">훈련세트(train set)</td><td>테스트 세트(test set)</td></tr>
<tr><td>훈련 세트(train set)</td><td>검증 세트(dev set)</td><td>테스트 세트(test set)</td></tr>
</table>

검증 세트가 너무 작을 경우
모델이 정확하게 평가되지 않아 최적이 아닌 모델을 잘못 선택할 수 있다.
검증 세트가 너무 클 경우
남은 훈련 세트가 전체 훈련세트보다 너무 작아진다.

해결: 작은 검증 세트를 여러 개를 사용해 반복적인 **교차 검증(cross-validation)**을 수행한다. 검증 세트마다 나머지 데이터에서 훈련한 모델을 해당 검증 세트에서 평가한다. 모든 모델의 평가를 평균하여 정확한 성능을 측정한다.
→ 훈련 시간이 검증 세트의 개수에 비례해 증가



Coss-Validation

데이터 불일치

	수집, 구매 데이터 ex) 일반적 말하기 인식 데이터 General speech recognition	실제 앱에서 받은 데이터 ex) 자동차 백미러에서 인식된 말하기 Rear view mirror speech
사람 수준 Human Level	Human Level Error	
훈련된 데이터에서 생긴 오차 Error on example trained on (Train Set)	Training error	
훈련되지 않은 데이터에서 생긴 오차 Error on example not trained on (Train-Dev Set)	Training-Dev Error	Dev/Test Error

회피가능 편향

분산 문제

데이터 불일치

1. 머신러닝을 어떻게 정의할 수 있나요?
머신러닝은 데이터로부터 학습할 수 있는 시스템을 만드는 것입니다. 이란 어떤 작업에서 주어진 성능 지표가 더 나아지는 것을 의미합니다.

2. 머신러닝이 도움을 줄 수 있는 문제 유형 네 가지
명확한 해결책이 없는 복잡한 문제, 수작업으로 만든 긴 규칙 리스트를 대체하는 경우, 변화하는 환경에 적응하는 시스템을 만드는 경우, 사람에게 통찰을 제공해야 하는 경우(예를 들면 데이터 마이닝)에 머신러닝이 도움을 줄 수 있다.

3. 레이블된 훈련 세트란 무엇인가요?
레이블된 훈련 세트는 각 샘플에 대해 원하는 정답(레이블)을 담고 있는 훈련 세트입니다.

4. 가장 널리 사용되는 지도 학습 작업 두 가지
가장 일반적인 두 가지 지도 학습 문제는 회귀와 분류입니다.

5. 보편적인 비지도 학습 작업 네 가지
보편적인 빚도 학습 문제는 군집, 시각화, 차원 축소, 연관 규칙 학습입니다.

6. 사전 정보가 없는 여러 지형에서 로봇을 걸어가게 하려면 어떤 종류의 머신러닝 알고리즘을 사용할 수 있나요?
알려지지 않은 지형을 탐험하는 로봇을 학습시키는 가장 좋은 방법은 강화 학습입니다. 이는 전형적으로 강화 학습이 다루는 유형의 문제입니다. 이 문제를 지도 학습이나 비지도 학습으로 표현하는 것도 가능하지만 일반적이지 않습니다.

7. 고객을 여러 그룹으로 분할하려면 어떤 알고리즘을 사용해야 하나요?
만약 그룹을 어떻게 정의할지 모른다면 비슷한 고객끼리 군집으로 나누기 위해 군집 알고 리즘 (비지도 학습)을 사용할수 있습니다. 그러나 어떤 그룹이 있어야 할지 안다면 분류 알고리즘(지도 학습)에 각 그룹에 대한 샘플을 주입합니다. 그러면 알고리즘이 전체 고객을 이런 그룹으로 분류하게 될 것입니다.

8. 스팸 감지의 문제는 지도 학습과 비지도 학습 중 어떤 문제로 볼 수 있나요?
스팸 감지는 전형적인 지도 학습 문제입니다. 알고리즘에 많은 이메일과 이에 상응하는 레이블(스팸 혹은 스팸 아님)이 제공됩니다.

9. 온라인 학습 시스템이 무엇인가요?
온라인 학습 시스템은 배치 학습 시스템과 달리 점진적으로 학습할 수 있습니다. 이 방식은 변화하는 데이터와 자율 시스템에 빠르게 적응하고 매우 많은 양의 데이터를 훈련시킬 수 있습니다.

10. 외부 메모리 학습이 무엇인가요?
외부 메모리 알고리즘은 컴퓨터의 주메모리에 들어갈 수 없는 대용량의 데이터를 다룰 수 있습니다. 외부 메모리 학습 알고리즘은 데이터를 미니배치로 나누고 온라인 학습 기법을 사용해 학습합니다.

11. 예측을 하기 위해 유사도 측정에 의존하는 학습 알고리즘은 무엇인가요?
사례 기반 학습 시스템은 훈련 데이터를 기억하는 학습입니다. 새로운 샘플이 주어지면 유사도 측정을 사용해 학습된 샘플 중에서 가장 비슷한 것을 찾아 예측으로 사용합니다.

12 모델 파라미터와 학습 알고리즘의 하이퍼파라미터 사이에는 어떤 차이가 있나요?
모델은 하나 이상의 파라미터 (예를 들면 선형 모델의 기울기)를 사용해 새로운 샘플이 주어지면 무엇을 예측할지 결정합니다. 학습 알고리즘은 모델이 새로운 샘플에 잘 일반화되도록 이런 파라미터들의 최적값을 찾습니다. 하이퍼파라미터는 모델이 아니라 이런 학습 알고리즘 자체의 파라미터입니다(예를 들면 적용할 규제의 정도).

13. 모델 기반 알고리즘이 찾는 것은 무엇인가요? 성공을 위해 이 알고리즘이 사용하는 가장 일반적인 전략은 무엇인가요? 예측은 어떻게 만드나요?
모델 기반 학습 알고리즘은 새로운 샘플에 잘 일반화되기 위한 모델 파라미터의 최적값을 찾습니다. 일반적으로 훈련 데이터에서 시스템의 예측이 얼마나 나쁜지 측정하고 모델에 규제가 있다면 모델 복잡도에 대한 페널티를 더한 비용 함수를 최소화함으로써 시스템을 훈련시킵니다. 예측을 만들려면 학습 알고리즘이 찾은 파라미터를 사용하는 모델의 예측 함수에 새로운 샘플의 특성을 주입합니다.

14. 머신러닝의 주요 도전 과제는 무엇인가요?
머신러닝의 주요 도전 과제는 부족한 데이터, 낮은 데이터 품질, 대표성 없는 데이터, 무의미한 특성, 훈련 데이터에 과소적합된 과도하게 간단한 모델, 훈련 데이터에 과대적합된 과도하게 복잡한 모델 등입니다.

15. 모델이 훈련 데이터에서 성능은 좋지만 새로운 샘플에서의 일반화 성능이 나쁘다면 어떤 문제가 있는 건가요? 가능한 해결책 세 가지는 무엇인가요?
모델이 훈련 데이터에서는 잘 작동하지만 새로운 샘플에서는 형편없다면 이 모델은 훈런 데이터에 과대적합되었을 가능성이 높습니다(또는 매우 운이 좋은 경우만 훈련 데이터에 있는 것입니다). 과대적합에 대한 해결책은 더 많은 데이터를 모으거나, 모델을 단순화하거나 (간단한 알고리즘을 선택하거나, 특성이나 파라미터의 수를 줄이거나, 모델에 규제를 가합니다), 훈련 데이터에 있는 잡음을 감소시키는 것입니다.

16. 테스트 세트가 무엇인고 왜 사용해야 하나요?
테스트 세트는 실전에 배치되기 전에 모델이 새로운 샘플에 대해 만들 일반화 오차를 추정
하기 위해 사용합니다.

17. 검증 세트의 목적은 무엇인가요?
검증 세트는 모델을 비교하는 데 사용됩니다. 이를 사용해 가장 좋은 모델을 고르고 하이퍼파라미터를 튜닝합니다.

18. 훈련-개발 세트가 무엇인가요? 언제 필요하고 어떻게 사용해야 하나요?
훈련-개발 세트는 (모델을 실전에 투입했을 때 사용될 데이터와 가능한 최대로 가까워야하는) 검증, 테스트 세트에 사용되는 데이터와 훈련 세트 사이에 데이터 불일치 위험이 있을 때 사용합니다. 훈련 세트의 일부에서 모델을 훈련하고 훈련-개발 세트와 검증 세트에서 평가합니다. 모델이 훈련 세트에서 잘 동작하지만 훈련-개발 세트에서 나쁜 성능을 낸다면 아마도 훈련 세트에 과대적합되었을 가능성이 높습니다. 훈련 세트와 훈련-개발 세트 양쪽에서 모두 잘 동작하지만 개발 세트에서 성능이 나쁘다면 훈련 데이터와 검증+테스트 데이터 사이에 데이터 불일치가 있을 가능성이 높습니다. 검증+테스트 데이터에 더 가깝게 되도록 훈련 데이터를 개선해야 합니다.

19 테스트 세트를 사용해 하이퍼파라미터를 튜닝하면 어떤 문제가 생기나요?
테스트 세트를 사용해 하이퍼파라미터를 튜닝하면 테스트 세트에 과대적합될 위험이 있고 일반화 오차를 낙관적으로 측정하게 됩니다(모델을 출시하면 기대한 것보다 나쁜 성능을 낼 것입니다).