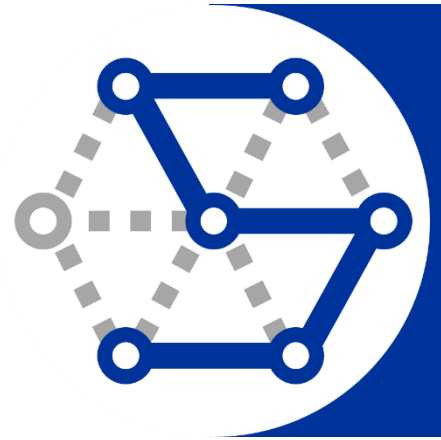

인공지능
Artificial Intelligence


04
머신러닝



1 머신러닝 개념과 종류

머신러닝(Machine Learning)

아서 새뮤얼 (Arthur Samuel), 1959

- 명시적인 프로그래밍 없이 컴퓨터가 학습하는 능력을 갖추게 하는 연구 분야

톰 미첼 (Tom Mitchell), 1997

- 어떤 작업 T에 대한 컴퓨터 프로그램의 성능을 P로 측정했을 때, 경험 E로 인해 성능이 향상됐다면, 이 컴퓨터 프로그램은 작업 T와 성능 측정 P에 대해 경험 E로 학습한 것

머신러닝의 장점

빅데이터의 폭발적 증가로 다양한 데이터 활용 가능

새로운 비즈니스와 수익 흐름에 대한 요구 증가

머신러닝 알고리즘의 발전

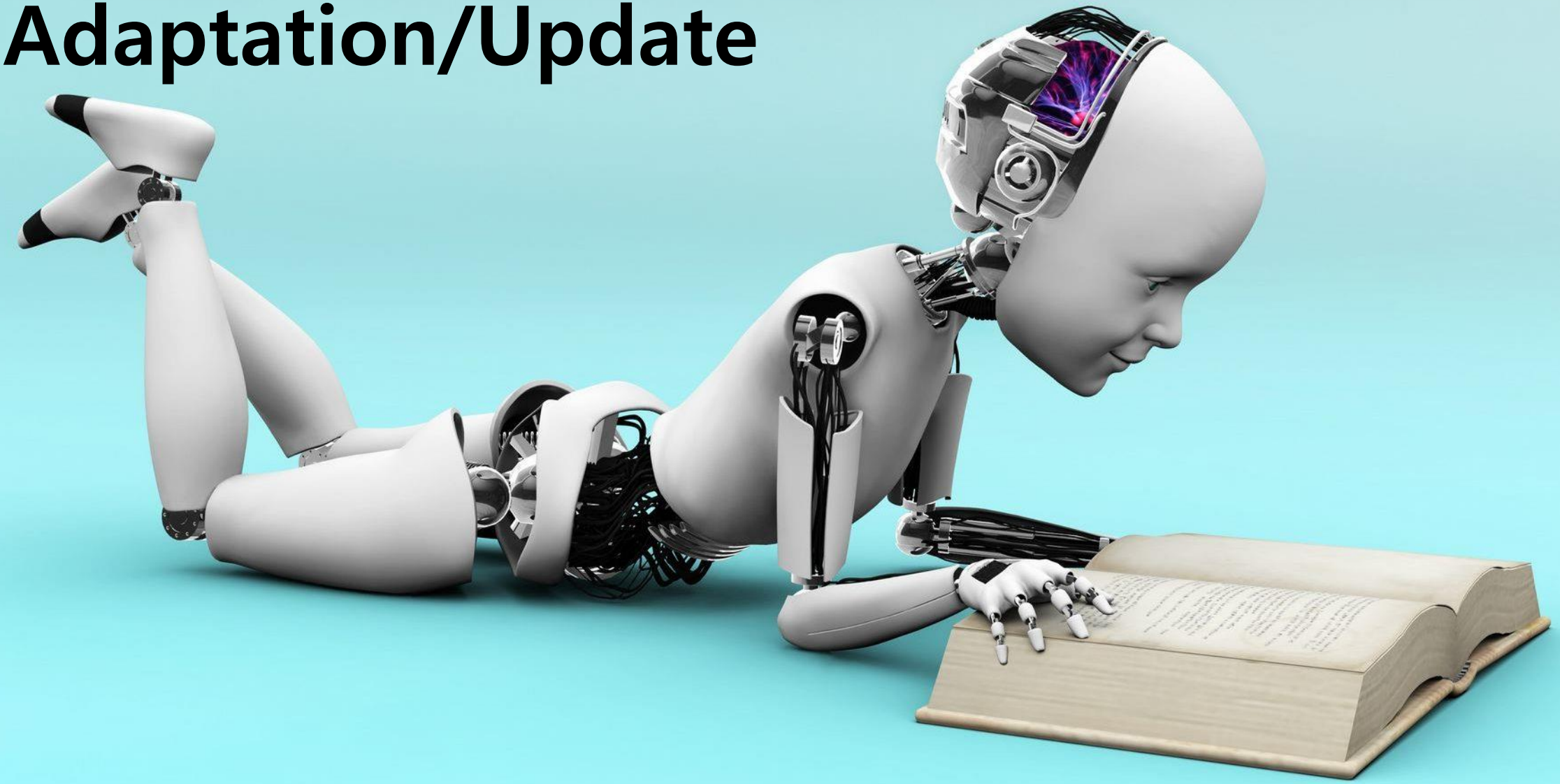
대규모의 빠른 컴퓨팅 능력을 갖춘 머신의 개발

데이터 처리 및 저장 용량 증가

지적재산권 등에 대한 소유권이 없어 누구나 이용 가능

러닝(Learning)=학습

Adaptation/Update



Machine Learning 학습 분류

지도학습 Supervised Learning	비지도학습 Unsupervised Learning	준지도학습 Semi-supervised Learning	강화학습 Reinforcement Learning
• 입력과 결과가 레이블로 표시 • 입력과 출력에 매핑되는 일반적인 규칙을 학습	• 원하는 출력 없이 입력 데이터 사용 • 입력 데이터의 구조나 패턴을 찾는 것이 목표	• 레이블이 있는 것과 없는 것이 혼합된 경우 사용 • 일반적으로 일부 데이터에만 레이블이 있음	• 동적 환경과 함께 상호작용 • 어떤 지도가 없이 일정한 목표를 수행

온라인 학습 vs. 배치 학습

온라인 학습 (Online Learning)

- 적은 데이터를 사용해 미니배치(mini-batch) 단위로 점진적으로 학습
- 실시간 시스템이나 메모리 부족의 경우 사용

배치 학습 (Batch Learning)

- 전체 데이터를 모두 사용해 오프라인에서 학습
- 컴퓨팅 자원이 풍부한 경우 사용

사례 기반 학습 vs. 모델 기반 학습

사례 기반 학습 (Instance-based Learning)

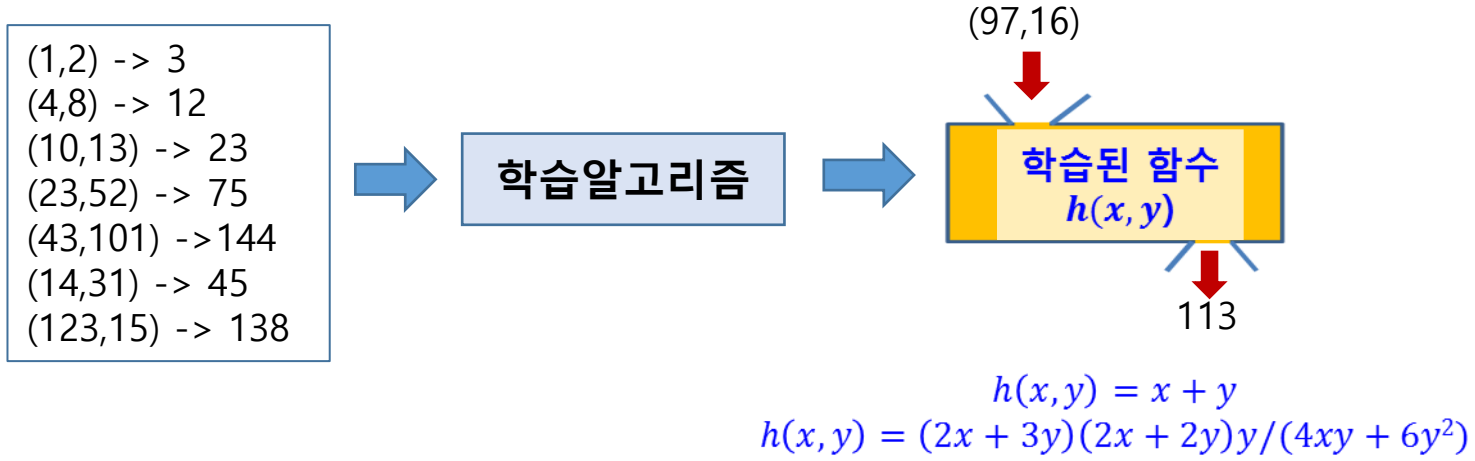
- 샘플을 기억하는 것이 훈련
- 예측을 위해 샘플 사이의 유사도 측정

모델 기반 학습 (Model-based Learning)

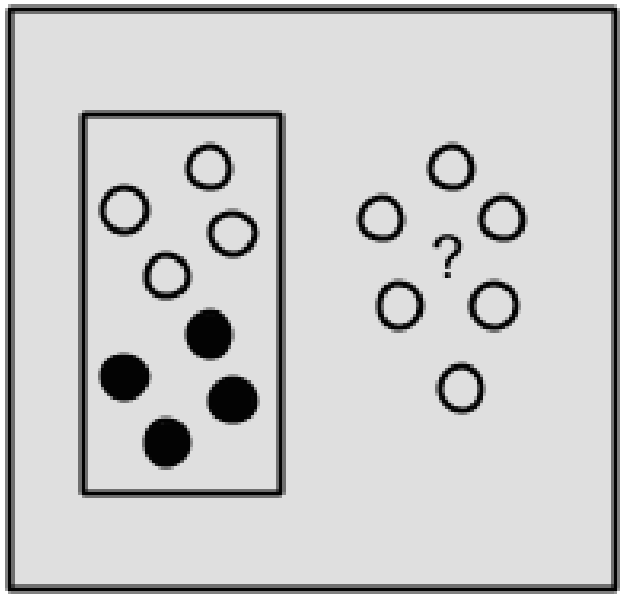
- 샘플을 사용해 모델을 훈련
- 훈련된 모델을 사용해 예측

연역적 학습 vs. 귀납적 학습

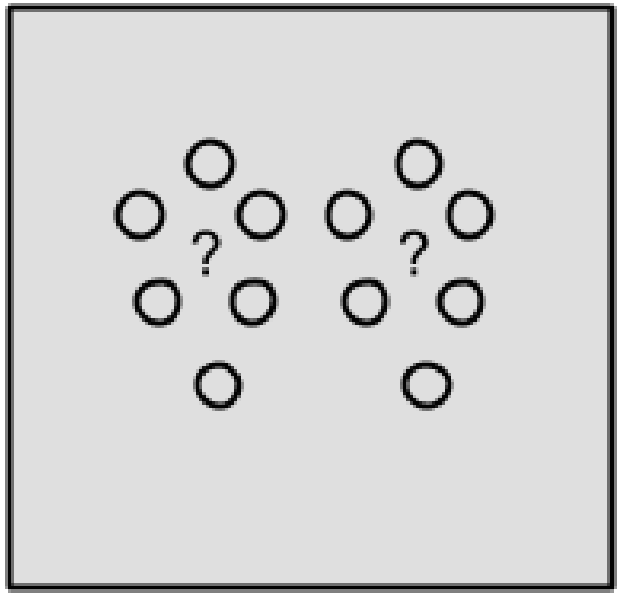
- 연역적 학습 (deductive learning): 연역적 추론(deductive inference)을 통한 학습
- 귀납적 학습 (inductive learning): 사례들(examples)을 일반화(generalization)하여 패턴(pattern) 또는 모델(model)을 추출하는 것
  - 일반적인 기계학습의 대상
  - 학습 데이터를 잘 설명할 수 있는 패턴을 찾는 것
  - 오컴의 면도날(Occam's razor): 가능하면 학습 결과를 간단한 형태로 표현하는 것이 좋음



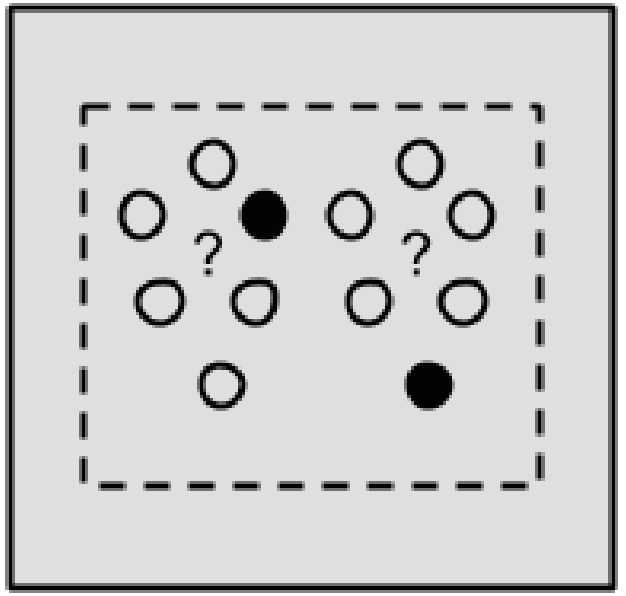
머신러닝의 학습 종류



Supervised Learning Algorithms



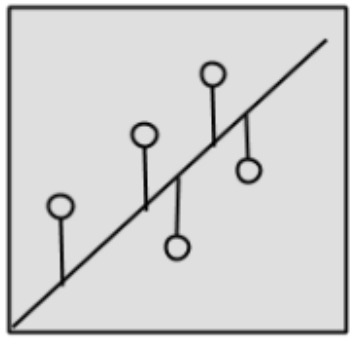
Unsupervised Learning Algorithms



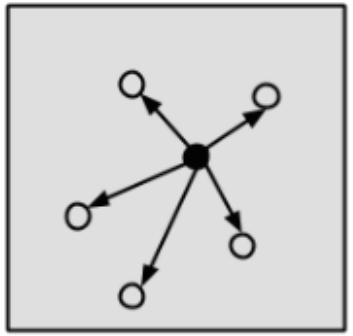
Semi-supervised Learning Algorithms

https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

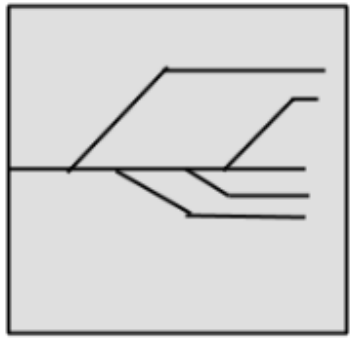
머신러닝 종류



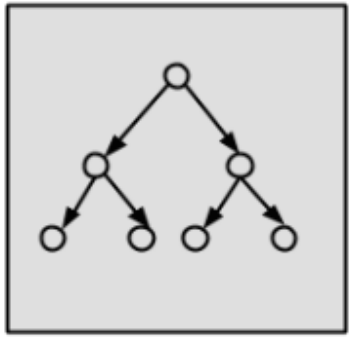
Regression Algorithms



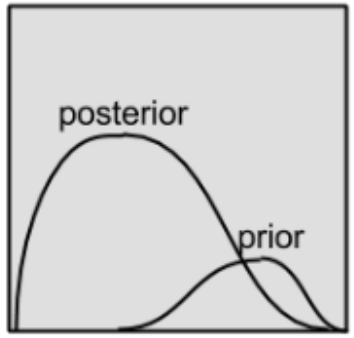
Instance-based Algorithms



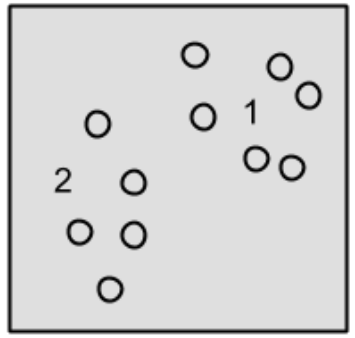
Regularization Algorithms



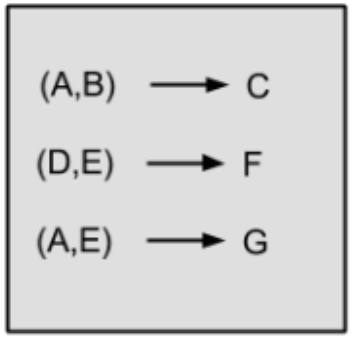
Decision Tree Algorithms



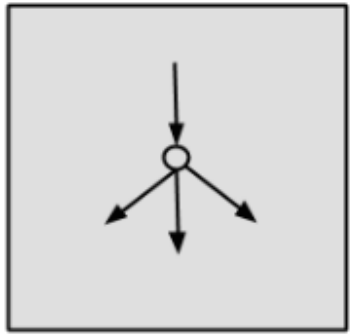
Bayesian Algorithms



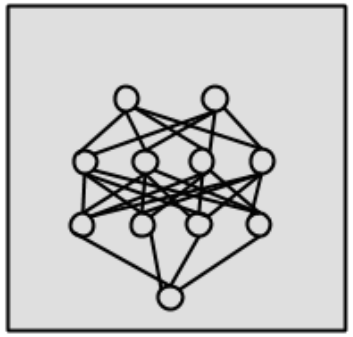
Clustering Algorithms



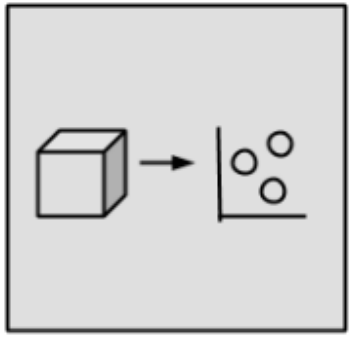
Association Rule Learning Algorithms



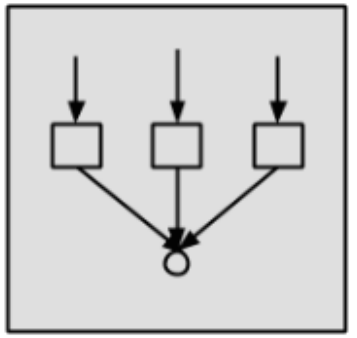
Artificial Neural Network Algorithms



Deep Learning Algorithms



Dimensional Reduction Algorithms



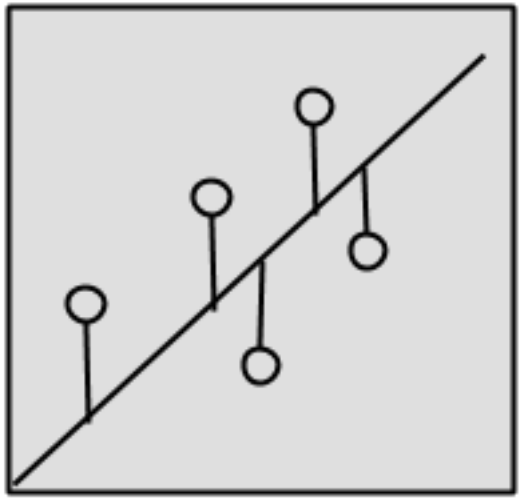
Ensemble Algorithms

Other Algorithms

https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

회귀 알고리즘(Regression Algorithms)

- 모델에 의한 예측을 위해 오차 측정을 사용하여 반복적으로 구체화된 변수 간의 관계를 모델링
- 통계의 핵심이며 통계 기반 기계학습에서 채택
- 알고리즘
  - Ordinary Least Squares Regression (OLSR)
  - Linear Regression
  - Logistic Regression
  - Stepwise Regression
  - Multivariate Adaptive Regression Splines (MARS)
  - Locally Estimated Scatterplot Smoothing (LOESS)

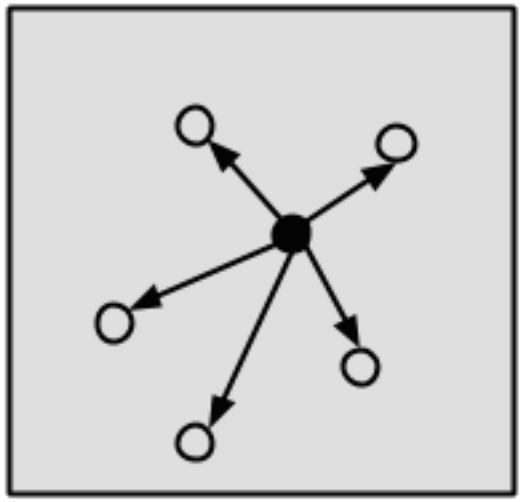


Regression Algorithms

https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

인스턴스 기반 알고리즘(Instance-based Algorithms)

- 모델에 중요하거나 필요하다고 여기는 학습 데이터의 인스턴스 또는 예제에 대한 의사 결정
- 예측을 위해 유사성 측정 등을 사용해 비교
- 알고리즘
  - k-Nearest Neighbor (Kenn)
  - Learning Vector Quantization (LVQ)
  - Self-Organizing Map (SOM)
  - Locally Weighted Learning (LWL)
  - Support Vector Machines (SVM)

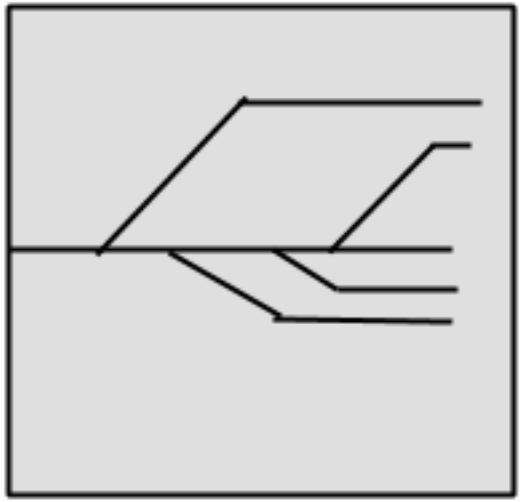


Instance-based Algorithms

https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

정규화 알고리즘(Regularization Algorithms)

- 복잡성에 따라 모델에 불이익을 주는 방법으로 일반화
- 일반적으로 회귀 방법 등에 확장되어 사용
- 알고리즘
  - Ridge Regression
  - Least Absolute Shrinkage and Selection Operator (LASSO)
  - Elastic Net
  - Least-Angle Regression (LARS)

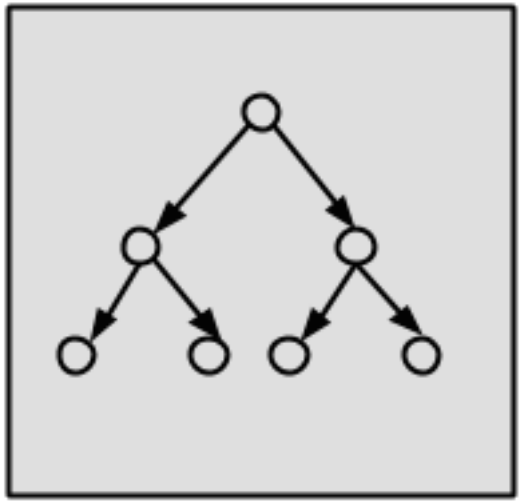


Regularization Algorithms

https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

의사 결정 트리 알고리즘(Decision Tree Algorithms)

- 데이터 속성의 실제 값을 기반으로 의사 결정 모델 구성
- 주어진 레코드에 대한 예측을 트리 구조에서 결정
- 분류 및 회귀 문제에 대한 데이터를 학습
- 알고리즘
  - Classification and Regression Tree (CART)
  - Iterative Dichotomies 3 (ID3)
  - C4.5 and C5.0 (different versions of a powerful approach)
  - Chi-squared Automatic Interaction Detection (CHAID)
  - Decision Stump
  - M5
  - Conditional Decision Trees

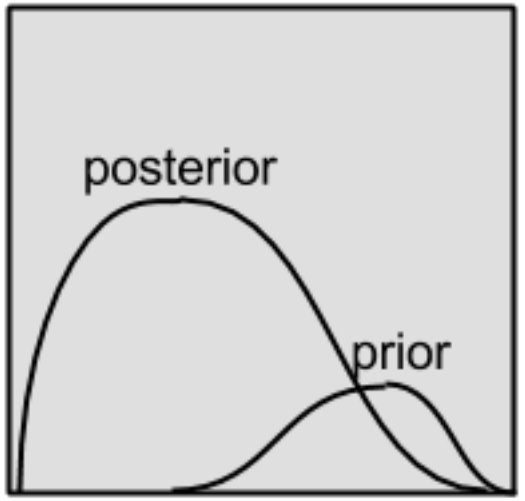


Decision Tree Algorithms

https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

베이지안 알고리즘(Bayesian Algorithms)

- 베이지안 확률 기반으로 불확실성에 대해 새로운 사건을 추정
- 분류 및 회귀와 같은 문제에 자주 사용
- 알고리즘
  - Naive Bayes
  - Gaussian Naive Bayes
  - Multinomial Naive Bayes
  - Averaged One-Dependence Estimators (AODE)
  - Bayesian Belief Network (BBN)
  - Bayesian Network (BN)

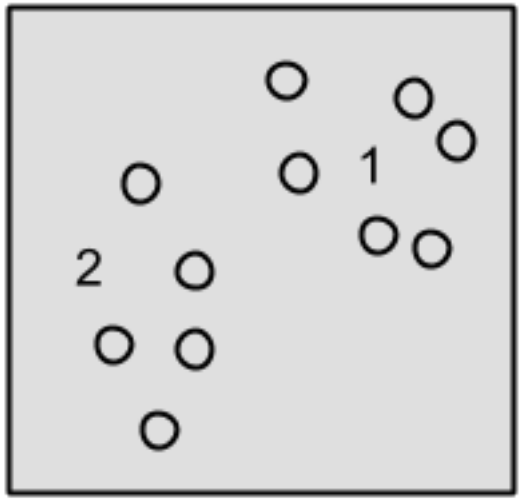


Bayesian Algorithms

https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

클러스터링 알고리즘(Clustering Algorithms)

- Centroid 기반 및 계층 기반의 모델 방법으로 구성
- 데이터를 최대 공통 그룹으로 잘 구성되도록 동작
- 알고리즘
  - k-Means
  - k-Medians
  - Expectation Maximization (EM)
  - Hierarchical Clustering

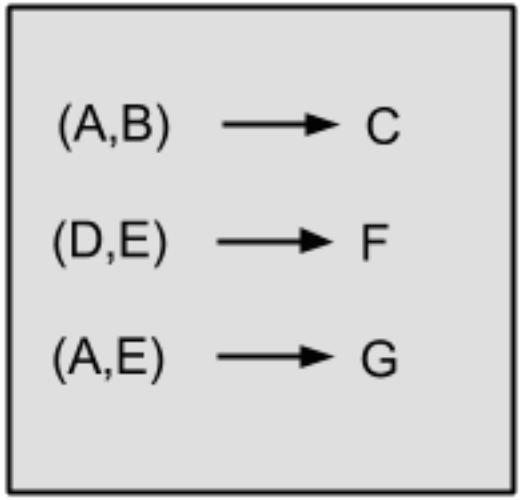


Clustering Algorithms

https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

연관 규칙 알고리즘(Association Rule Learning Algorithms)

- 변수 간 관찰된 관계를 가장 잘 설명하는 규칙을 추출
- 어떤 항목이 어떤 항목을 동반하여 등장하는 지를 파악
- 중요하고 유용한 연관성을 발견
- 알고리즘
  - Apriori algorithm
  - Eclat algorithm

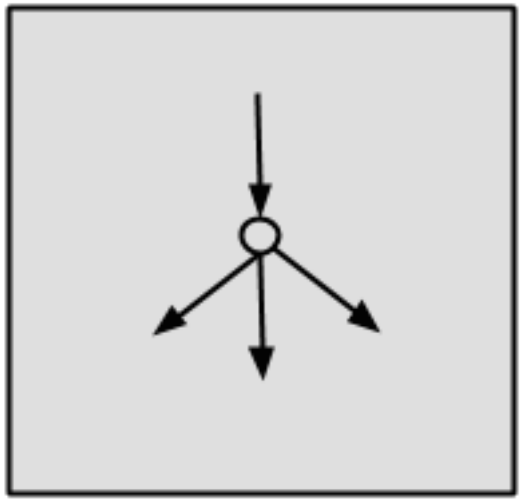


Association Rule
Learning Algorithms

https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

신경망 알고리즘(Artificial Neural Network Algorithms)

- 생물학적 신경망의 구조 또는 기능에서 영감을 받은 모델
- 회귀와 분류 문제에 일반적으로 사용되는 패턴 매칭의 한 종류
- 실제로는 모든 유형의 문제에 대한 다양한 알고리즘과 변화로 구성된 영역
- 알고리즘
  - Perceptron
  - Multilayer Perceptrons (MLP)
  - Back-Propagation
  - Stochastic Gradient Descent
  - Hopfield Network
  - Radial Basis Function Network (RBFN)

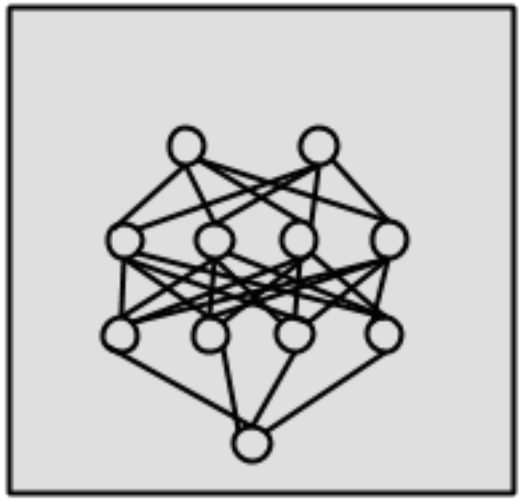


Artificial Neural Network Algorithms

https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

딥러닝 알고리즘(Deep Learning Algorithms)

- 인공신경망의 발전된 형태로 값싼 대규모 연산을 이용하는 방법
- 훨씬 크고 더 복잡한 신경망 구조를 가지고, 이미지, 텍스트, 오디오, 비디오와 같은 매우 큰 데이터와 관련됨
- 알고리즘
  - Convolutional Neural Network (CNN)
  - Recurrent Neural Networks (RNNs)
  - Long Short-Term Memory Networks (LSTMs)
  - Stacked Auto-Encoders
  - Deep Boltzmann Machine (DBM)
  - Deep Belief Networks (DBN)

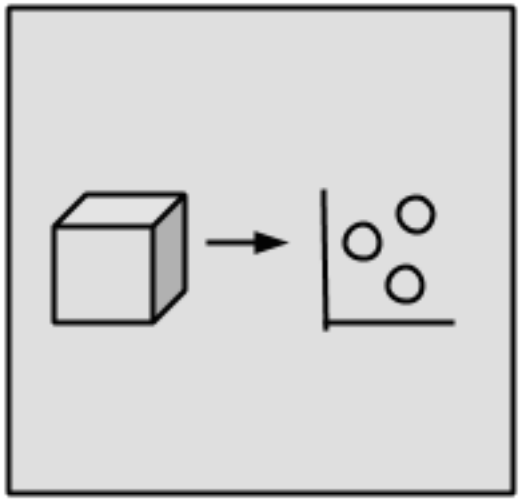


Deep Learning Algorithms

https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

차원 감소 알고리즘(Dimensional Reduction Algorithms)

- 데이터의 고유한 구조를 이용하여 데이터를 요약하거나 기술하는 방법
- 고차원의 데이터를 중요한 요소를 포함한 저차원으로 변환
- 알고리즘
  - Principal Component Analysis (PCA)
  - Principal Component Regression (PCR)
  - Partial Least Squares Regression (PLSR)
  - Multidimensional Scaling (MDS)
  - Linear Discriminant Analysis (LDA)
  - Mixture Discriminant Analysis (MDA)
  - Quadratic Discriminant Analysis (QDA)
  - Flexible Discriminant Analysis (FDA)

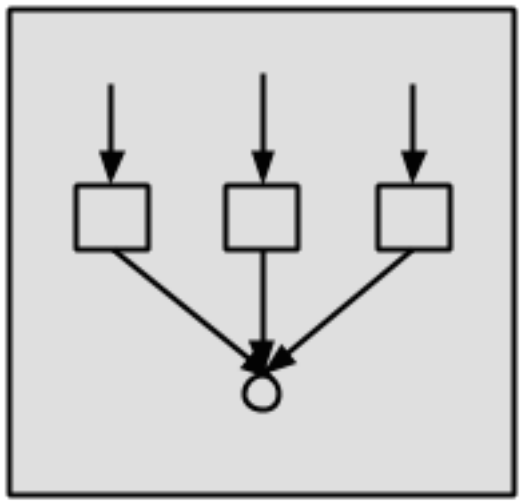


Dimensional Reduction Algorithms

https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

앙상블 알고리즘(Ensemble Algorithms)

- 독립적으로 훈련되고 전체적인 예측을 위해 어떤 식으로든 예측이 결합되는 복수의 약한 모델로 구성된 모델
- 어떤 유형의 약한 학습 모델을 결합시키고, 어떤 방식으로 결합할지가 중요
- 매우 강력하고 인기있는 기술
- 알고리즘
  - Bootstrapped Aggregation (Bagging)
  - AdaBoost
  - Weighted Average (Blending)
  - Stacked Generalization (Stacking)
  - Gradient Boosting Machines (GBM)
  - Gradient Boosted Regression Trees (GBRT)
  - Random Forest



Ensemble Algorithms

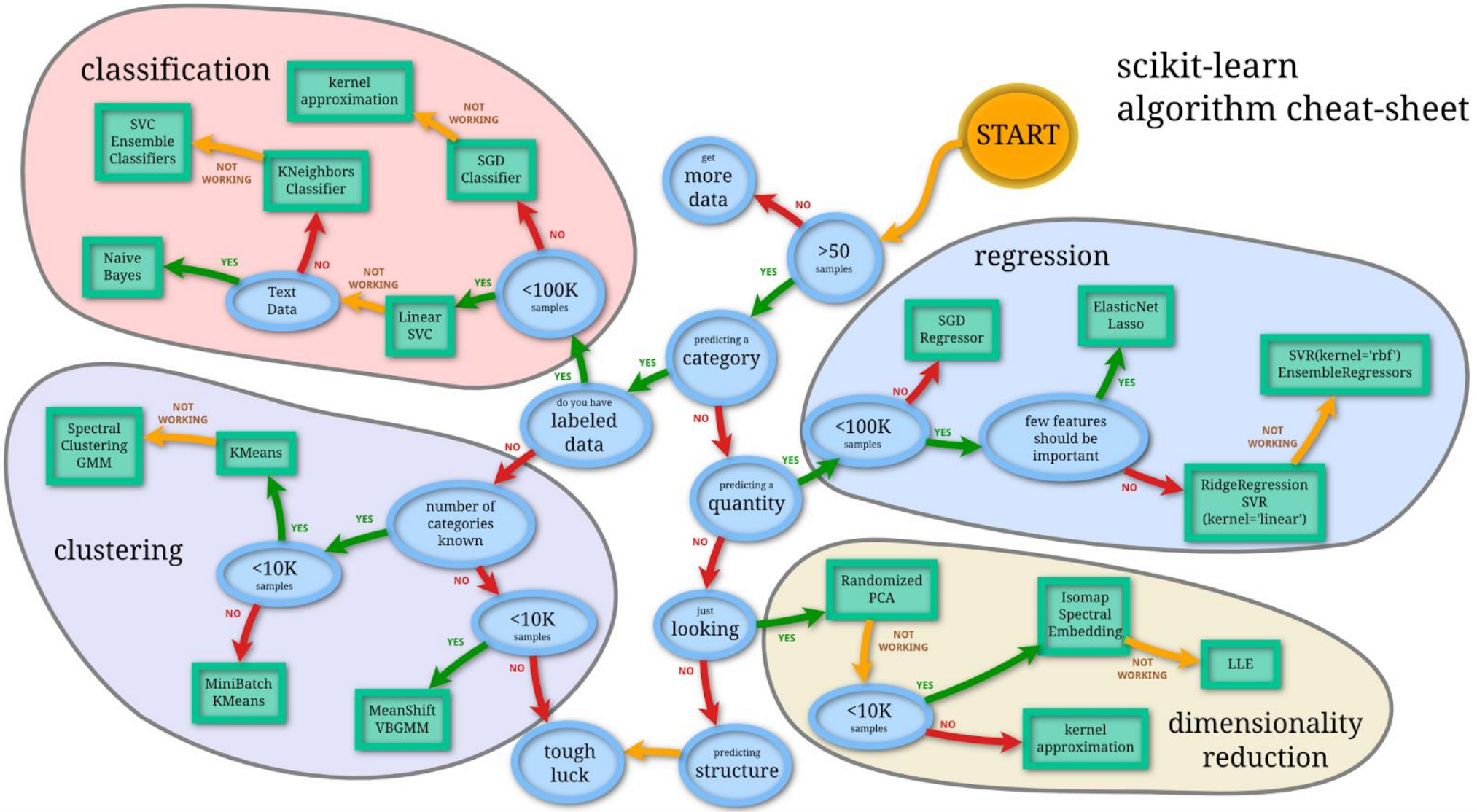
https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

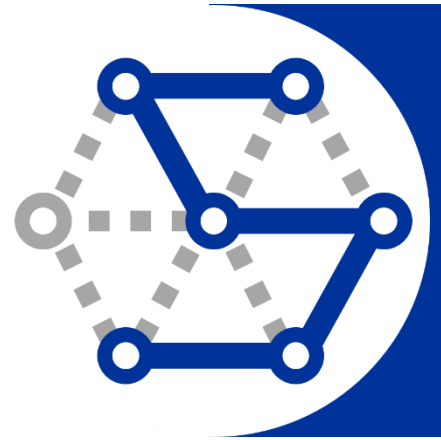
기타 머신러닝 알고리즘(Other Machine Learning Algorithms)

- 머신러닝 처리 단계의 작업
  - Feature selection algorithms
  - Algorithm accuracy evaluation
  - Performance measures
  - Optimization algorithms
- 전문적인 하위분야
  - Computational intelligence (evolutionary algorithms, etc.)
  - Computer Vision (CV)
  - Natural Language Processing (NLP)
  - Voice/Audio Processing
  - Recommender Systems
  - Reinforcement Learning
  - Graphical Models
  - …

https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

Machine Learning Cheat Sheet (for scikit-learn)

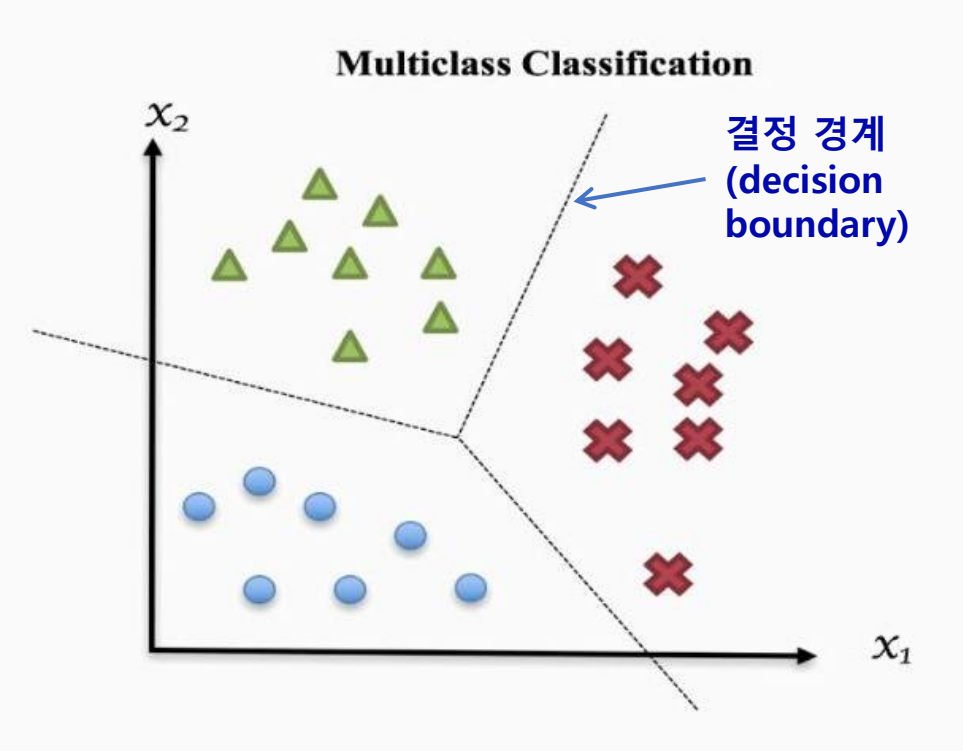




2 분류(Classification)

분류(classification)

- 데이터들을 정해진 몇 개의 부류(class, category)로 대응시키는 문제
- 분류 문제의 학습
  - 학습 데이터를 잘 분류할 수 있는 함수를 찾는 것
  - 함수의 형태는 수학적 함수일 수도 있고, 규칙일 수도 있음
- 분류기(classifier)
  - 학습된 함수를 이용하여 데이터를 분류하는 프로그램
- 이상적인 분류기
  - 학습에 사용되지 않은 데이터에 대해서 분류를 잘 하는 것
  - 일반화(generalization) 능력이 좋은 것



분류기 학습 알고리즘

- 결정트리(decision tree) 알고리즘
- K-근접이웃(K-nearest neighbor, KNN) 알고리즘
- 다층 퍼셉트론 신경망
- 딥러닝(deep learning) 알고리즘
- 서포트 벡터 머신(Support Vector Machine, SVM)
- 에이다부스트(AdaBoost)
- 랜덤 포리스트(Random Forest)
- 확률 그래프 모델 (Probabilistic Graphical Model)

데이터의 구분

학습 데이터 (training data)

- 모델을 학습하는데 사용하는 데이터 집합
- 학습 데이터가 많을수록 유리

테스트 데이터 (test data)

- 학습된 모델의 성능을 평가하는데 사용하는 데이터 집합
- 학습에 사용되지 않은 데이터이어야 함

검증 데이터 (validation data)

- 학습 과정에서 학습을 중단할 시점을 결정하기 위해 사용하는 데이터 집합

과대적합 vs. 과소적합

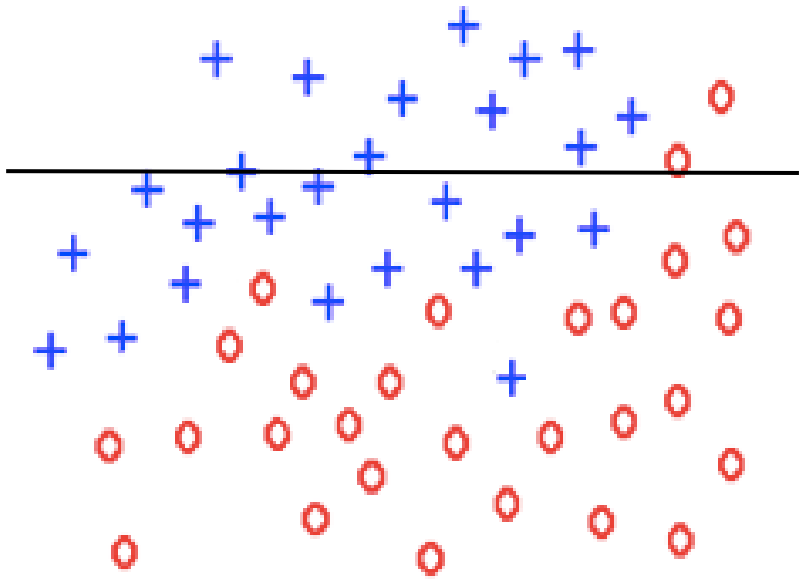
과대적합(Overfitting)

- 학습 데이터에 대해서 지나치게 잘 학습된 상태
- 데이터는 오류나 잡음을 포함할 개연성이 크기 때문에, 학습 데이터에 대해 매우 높은 성능을 보이더라도 학습되지 않은 데이터에 대해 좋지 않은 성능을 보일 수 있음
- 규제를 통해 과대적합 감소 가능

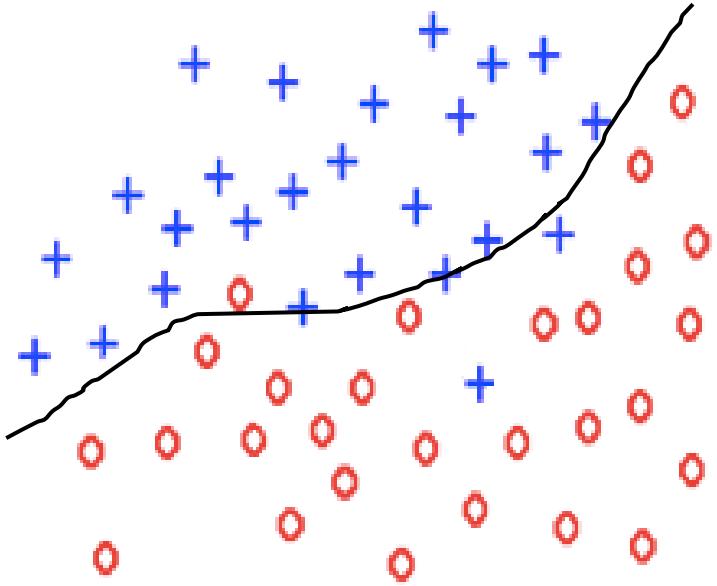
과소적합(Underfitting)

- 모델이 너무 단순하여 학습 데이터를 충분히 학습하지 못함
- 모델 파라미터가 많은 모델을 사용하거나 좋은 특성을 찾음

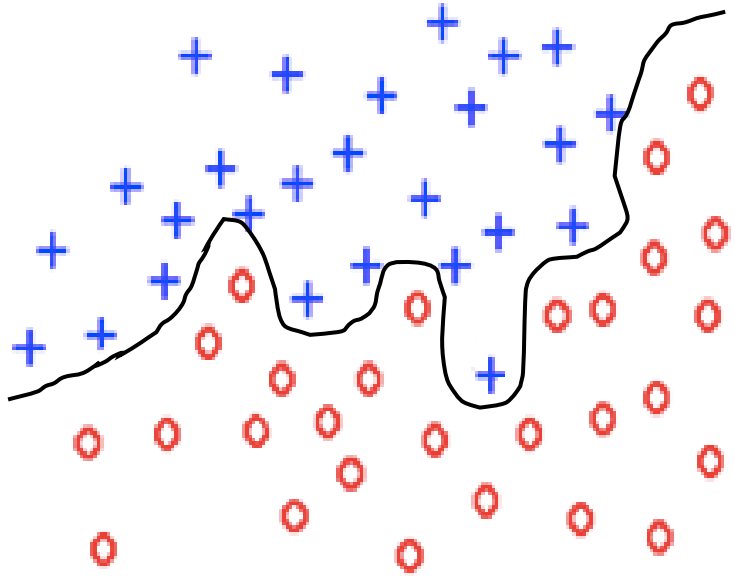
과대적합 vs. 과소적합



과소적합(underfitting)



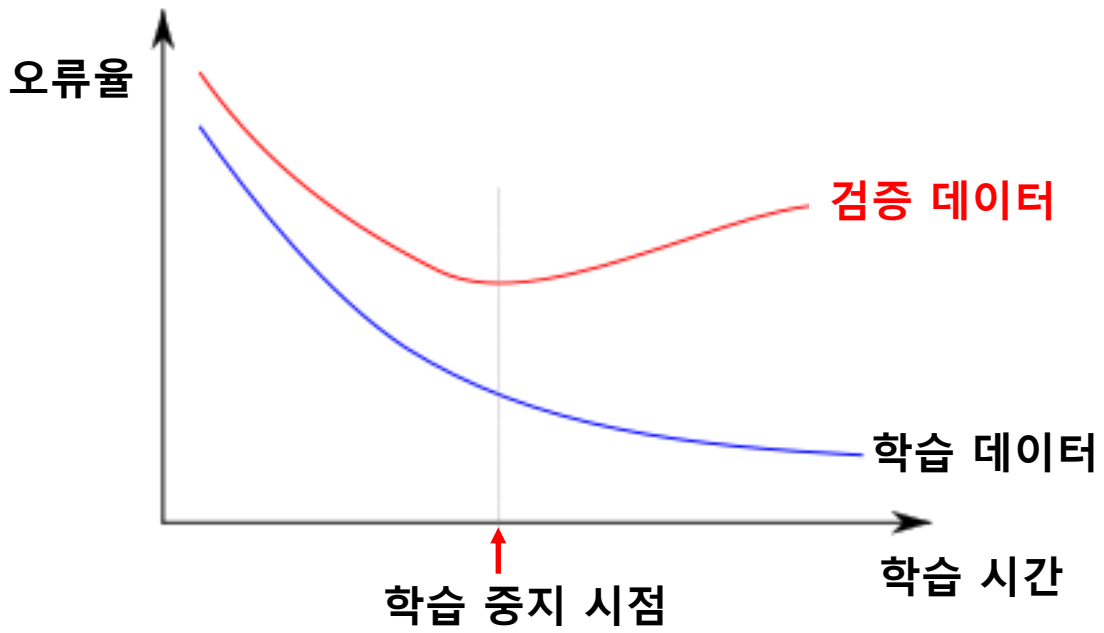
정적합(good fitting)



과대적합(overfitting)

과대적합 회피 방법

- 학습데이터에 대한 성능
  - 학습을 진행할수록 오류 개선 경향
  - 지나치게 학습이 진행되면 과대적합 발생
- 학습과정에서 별도의 검증 데이터 (validation data)에 대한 성능 평가
- 검증 데이터에 대한 오류가 감소하다가 증가하는 시점에 학습 중단



성능 평가

- 정확도(accuracy)
  - 얼마나 정확하게 분류하는가
  - 정확도 = (옳게 분류한 데이터 개수)/(전체 데이터 개수)
  - 테스트 데이터에 대한 정확도를 모델의 정확도로 사용
- 정확도가 높은 모델을 학습하기 위해서는 많은 학습데이터를 사용하는 것이 유리
- 학습데이터와 테스트 데이터는 겹치게 않도록 해야 함

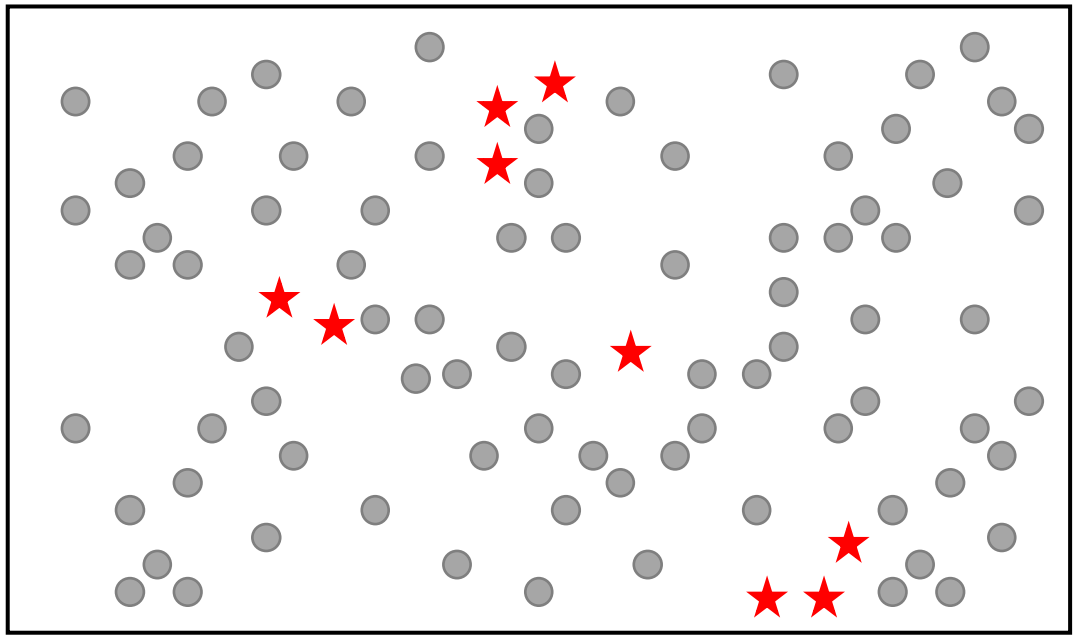
데이터 부족한 경우 성능평가

- 별도로 테스트 데이터를 확보하면 비효율적
- 가능하면 많은 데이터를 학습에 사용하면서, 성능 평가하는 방법 필요

- K-겹 교차검증(k-fold cross-validation) 사용
  - 전체 데이터를 k 등분
  - 각 등분을 한번씩 테스트 데이터로 사용하여, 성능 평가를 하고 평균값 선택

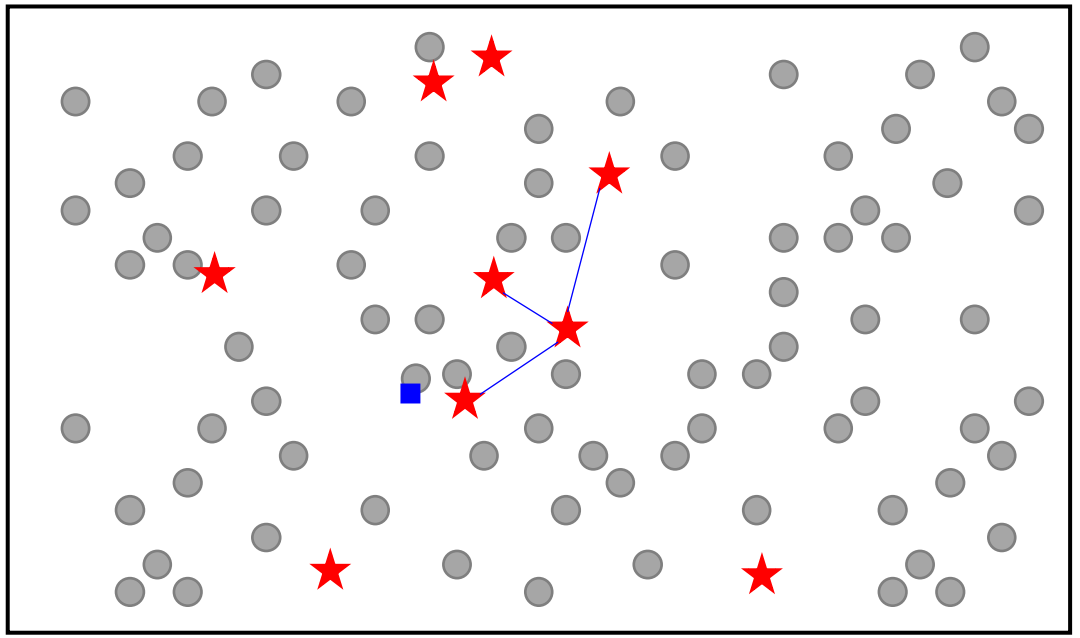
불균형 데이터(imbalanced data) 문제

- 특정 부류에 속하는 학습 데이터의 개수가 다른 부류에 비하여 지나치게 많은 경우
- 정확도에 의한 성능 평가는 무의미할 수 있음
- 예. A 부류의 데이터가 전체의 99%인 경우, 분류기의 출력을 항상 A 부류로 하더라도 정확도는 99%가 됨
- 대응방안
  - 가중치를 고려한 정확도 척도 사용
  - 많은 학습데이터를 갖는 부류에서 재표본추출(re-sampling)
  - 적은 학습데이터를 갖는 부류에 대해서 인공적인 데이터 생성



SMOTE(Synthetic Minority Over-sampling Technique) 알고리즘

- 빈도가 낮은 부류의 학습 데이터를 인공적으로 만들어 내는 방법
- 임의로 낮은 빈도 부류의 학습 데이터 $\boldsymbol{x}$ 선택
- $\boldsymbol{x}$의 k-근접이웃(k-nearest neighbor, KNN)인 같은 부류의 데이터 선택
- k-근접이웃 중에 무작위로 하나 y를 선택
- x와 y를 연결하는 직선 상의 무작위 위치에 새로운 데이터 생성



이진 분류기의 성능 평가

- 이진 분류기(binary classifier): 두 개의 분류만을 갖는 데이터에 대한 분류기

		실제 정답	
		True	False
분류 결과	True	True Positive	False Positive
	False	False Negative	True Negative

- True Positive(TP) : 실제 True인 정답을 True라고 예측 (정답)
- False Positive(FP) : 실제 False인 정답을 True라고 예측 (오답)
- False Negative(FN) : 실제 True인 정답을 False라고 예측 (오답)
- True Negative(TN) : 실제 False인 정답을 False라고 예측 (정답)

이진 분류기의 성능 평가

- 민감도(Sensitivity), 재현율(Recall)

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

- 특이도(Specificity)

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

- 양성예측도(Positive Predictive Value), 정밀도(Precision)

$$Precision = \frac{TP}{TP + FP}$$

- 음성예측도(Negative Predictive Value)

$$NPV = \frac{TN}{TN + FN}$$

		실제 정답	
		True	False
분류 결과	True	True Positive	False Positive
	False	False Negative	True Negative

이진 분류기의 성능 평가

- 위양성율

$$\frac{FP}{FP+TN} = 1 - Specificity$$

- 위발견율

$$\frac{FP}{TP+FP} = 1 - Precision$$

- 정확도(Accuracy)

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$$

- F1 측도(F1 score)

$$F1 = 2\frac{Precision \cdot Recall}{Precision + Recall}$$

		실제 정답	
		True	False
분류 결과	True	True Positive	False Positive
	False	False Negative	True Negative

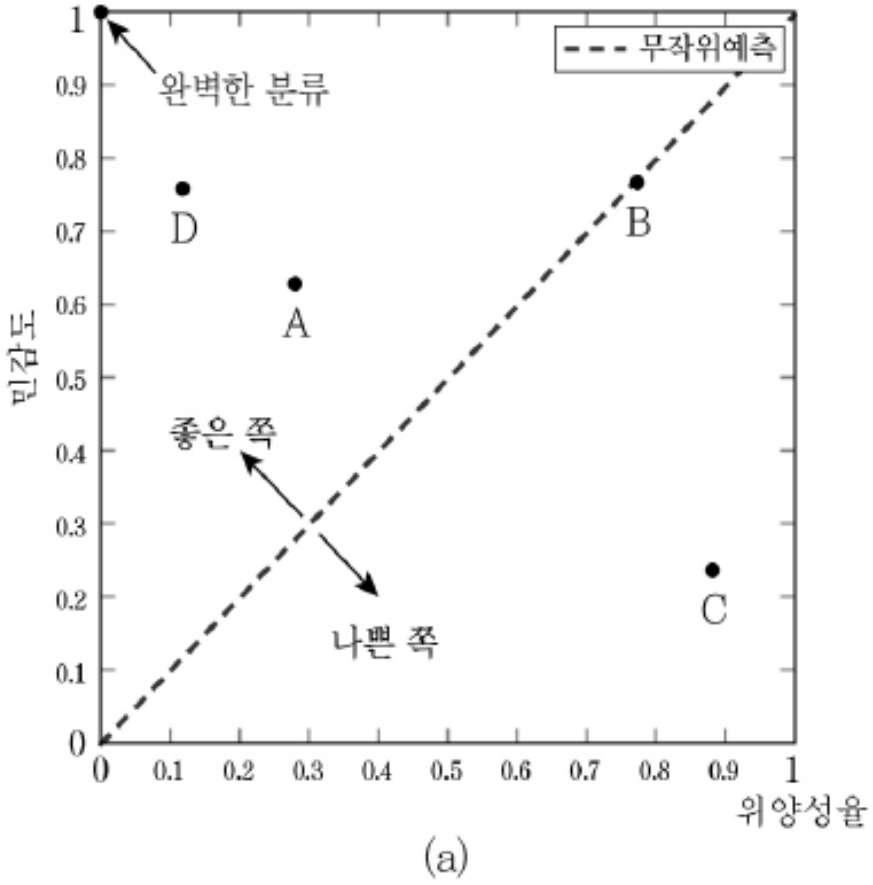
이진 분류기의 성능 평가

		True condition			
	Total population	Condition positive	Condition negative	Prevalence $= \frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	Accuracy (ACC) $= \frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$
Predicted condition	Predicted condition positive	**True positive**	**False positive**, Type I error	Positive predictive value (PPV), Precision $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$	False discovery rate (FDR) $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$
	Predicted condition negative	**False negative**, Type II error	**True negative**	False omission rate (FOR) $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$	Negative predictive value (NPV) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$
		True positive rate (TPR), Recall, Sensitivity, probability of detection, Power $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Positive likelihood ratio (LR+) $= \frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) $= \frac{\text{LR+}}{\text{LR−}}$
		False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	Specificity (SPC), Selectivity, True negative rate (TNR) $= \frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Negative likelihood ratio (LR−) $= \frac{\text{FNR}}{\text{TNR}}$	F_1 score $= 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$

https://en.wikipedia.org/wiki/Confusion_matrix

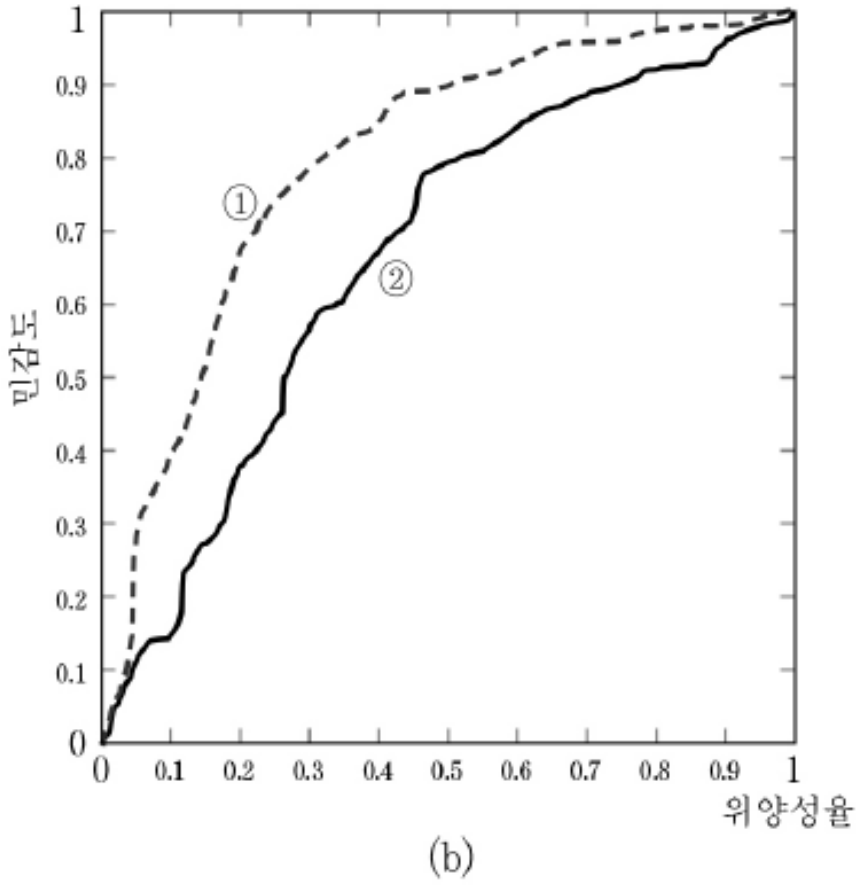
이진 분류기의 성능 평가

- ROC 곡선
  - 부류 판정 임계값에 따른 (위양성율, 민감도) 그래프

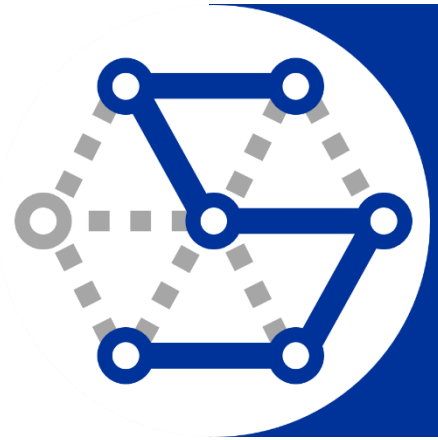


(a)

- AUC(Area Under the Curve)
  - ROC 곡선에서 곡선 아래 부분의 면적
  - 클수록 바람직



(b)

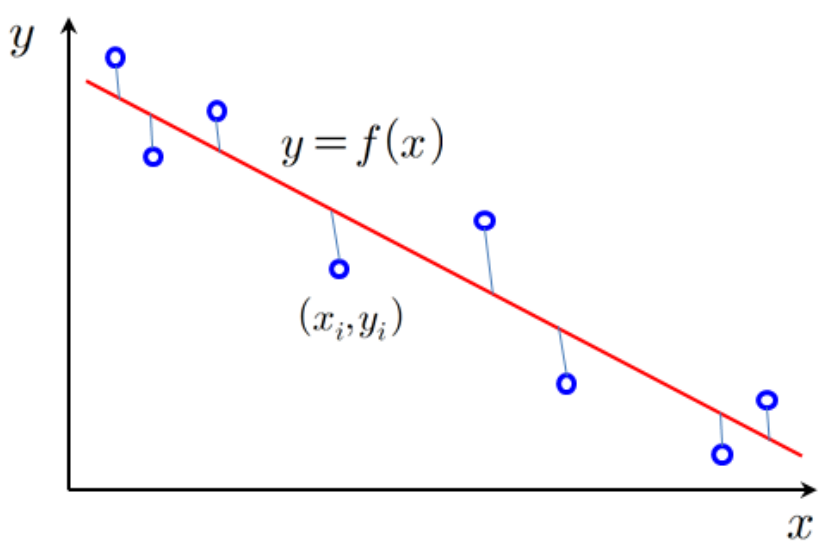


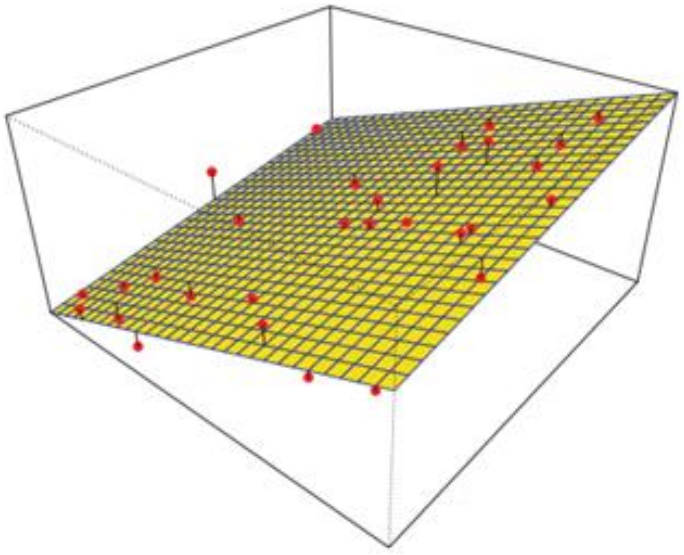
3 회귀(Regression)

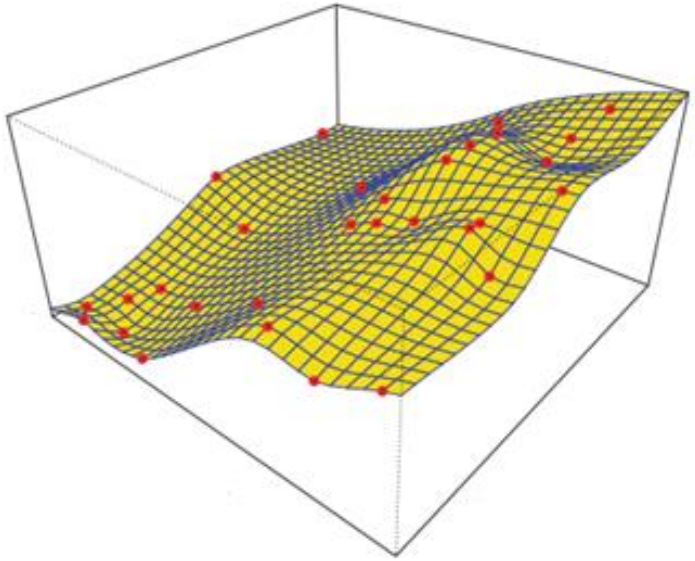
회귀(Regression)

- **학습 데이터에 부합**되는 **출력**값이 **실수**인 함수를 찾는 문제
- 성능
  - 오차 : 예측값과 실제값의 차이
  - 테스트 데이터들에 대한 (예측값 – 실제값)^{2}의 평균 또는 평균의 제곱근
  - 모델의 종류(함수의 종류)에 영향을 받음

$$f^*(x) = \arg\min_f \sum_{i=1}^{n} (y_i - f(x_i))^2$$

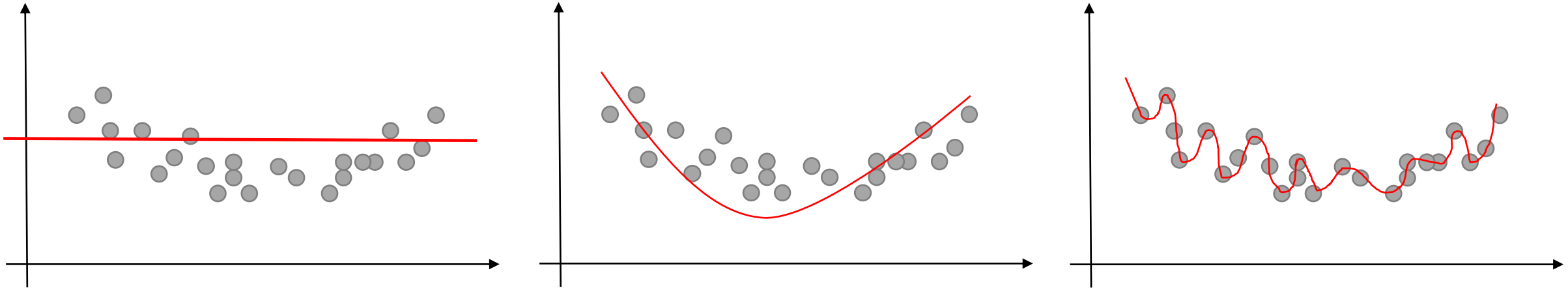






회귀의 과소적합(overfitting)과 과대적합(underfitting)

- 과대적합: 지나치게 복잡한 모델(함수) 사용
- 과소적합: 지나치게 단순한 모델(함수) 사용



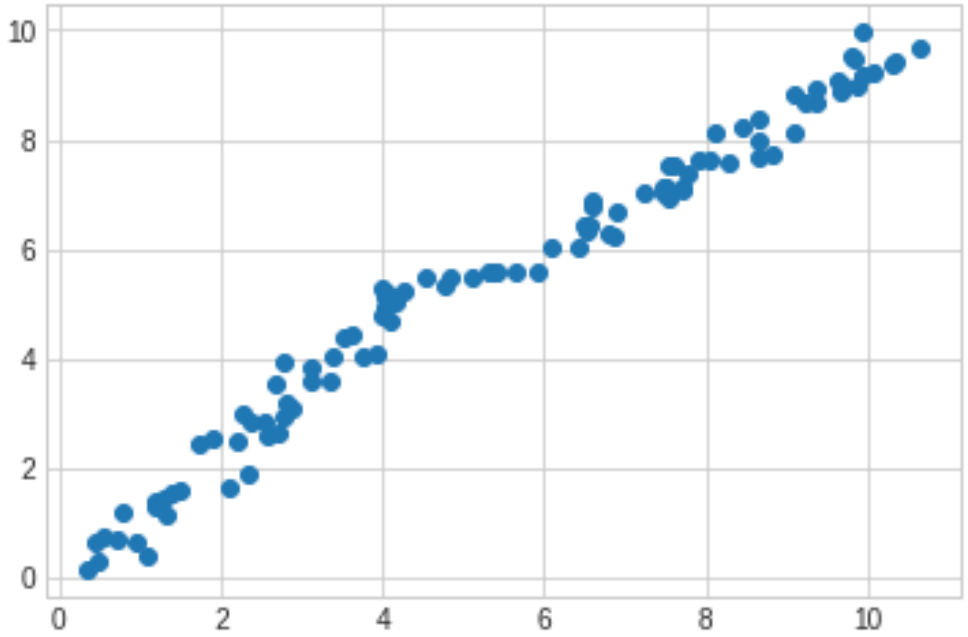
과소적합(underfitting)　　정적합(good fitting)　　과대적합(overfitting)

- 회귀의 과적합(overfitting) 대응 방법
  - 모델의 복잡도(model complexity)를 성능 평가에 반영
  - 목적함수 = 오차의 합 + (가중치) * (모델 복잡도)

선형 회귀(Linear Regression)

- 선형 모델은 과거 부터 지금 까지 널리 사용되고 연구 되고 있는 기계학습 방법
- 선형 모델은 입력 데이터에 대한 선형 함수를 만들어 예측 수행
- 회귀 분석을 위한 선형 모델은 다음과 같이 정의

- $\hat{y}(w, x) = w_0 + w_1 x_1 + \cdots + w_p x_p$
  - x : 입력 데이터
  - w : 모델이 학습할 파라미터
  - w_0 : 편향
  - $w_1 \sim w_p$: 가중치

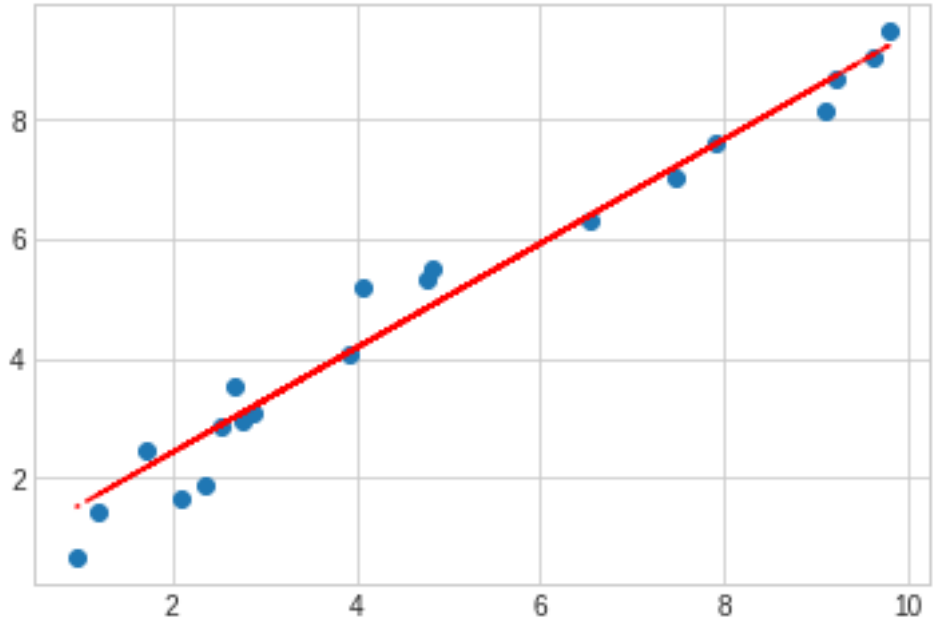


선형 회귀(Linear Regression)

- 선형 회귀(Linear Regression)또는 최소제곱법(Ordinary Least Squares)은 가장 간단한 회귀 분석을 위한 선형 모델
- 선형 회귀는 모델의 예측과 정답 사이의 평균제곱오차(Mean Squared Error)를 최소화하는 학습 파라미터 w 를 찾음
- 평균제곱오차

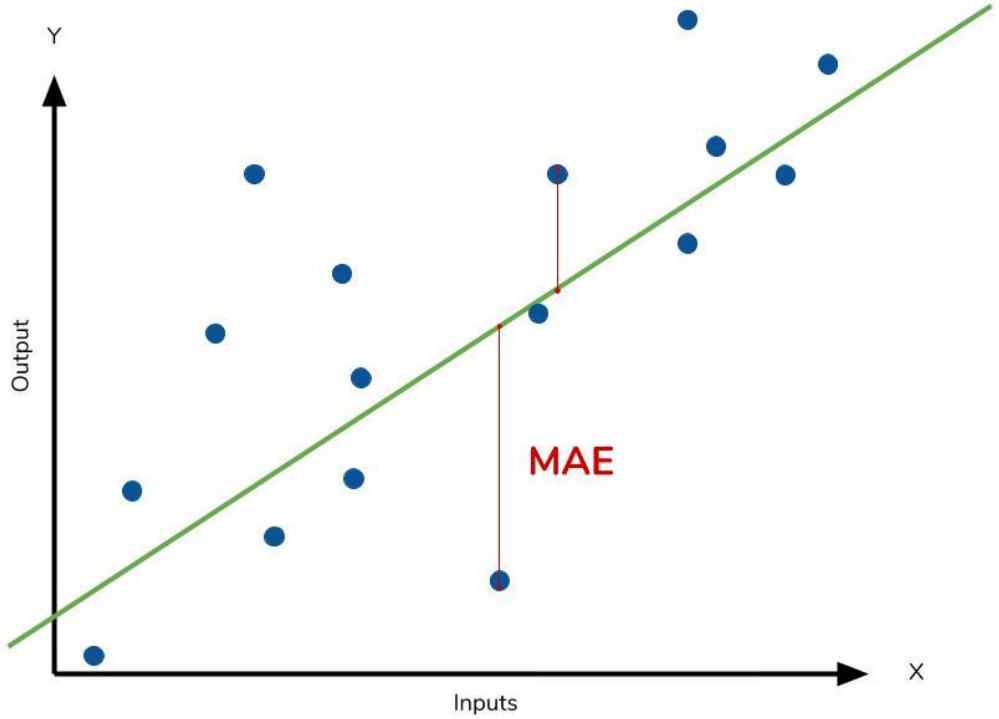
$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

  - y: 정답
  - $\hat{y}$: 예측 값을 의미



Mean Absolute Error (MAE)

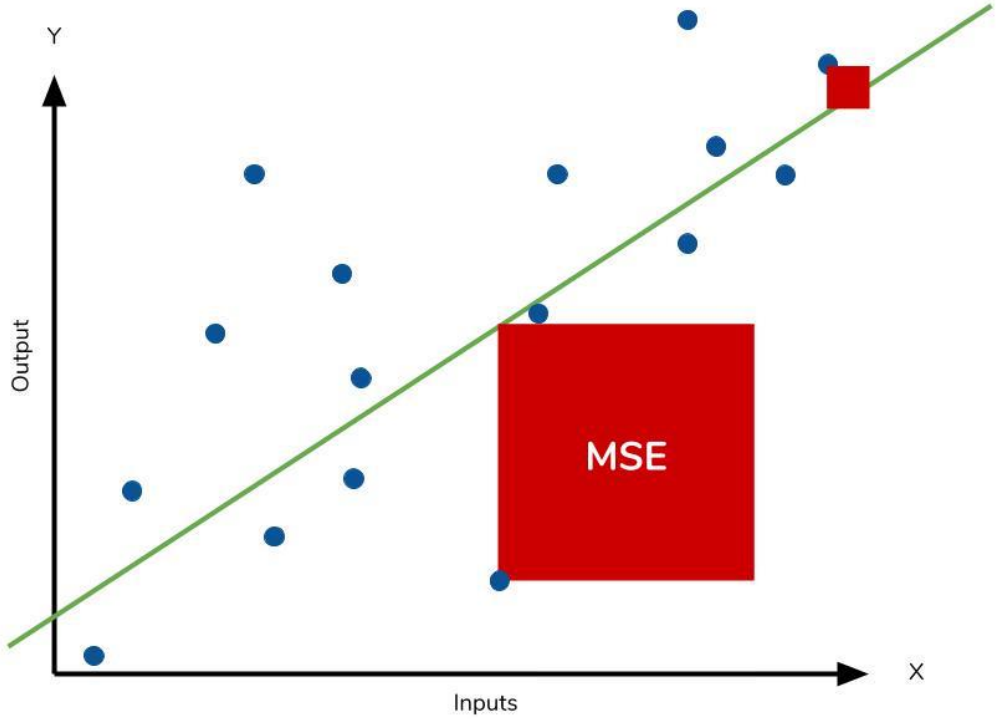
- $MAE = \frac{\sum|y-\hat{y}|}{N}$
- 모델의 예측값과 실제값의 차이를 모두 더한다는 개념
- 절대값을 취하기 때문에 가장 직관적으로 알 수 있는 지표
- MSE 보다 특이치에 robust
- 절대값을 취하기 때문에 모델이 underperformance 인지 overperformance 인지 알 수 없음
  - underperformance: 모델이 실제보다 낮은 값으로 예측
  - overperformance: 모델이 실제보다 높은 값으로 예측



https://partrita.github.io/posts/regression-error/

Mean Squared Error(MSE)

- $MSE = \frac{\sum(y-\hat{y})^2}{N}$
- 제곱을 하기 때문에 MAE와는 다르게 모델의 예측값과 실제값 차이의 면적의 합
- 특이값이 존재하면 수치가 많이 늘어나므로 특이치에 민감

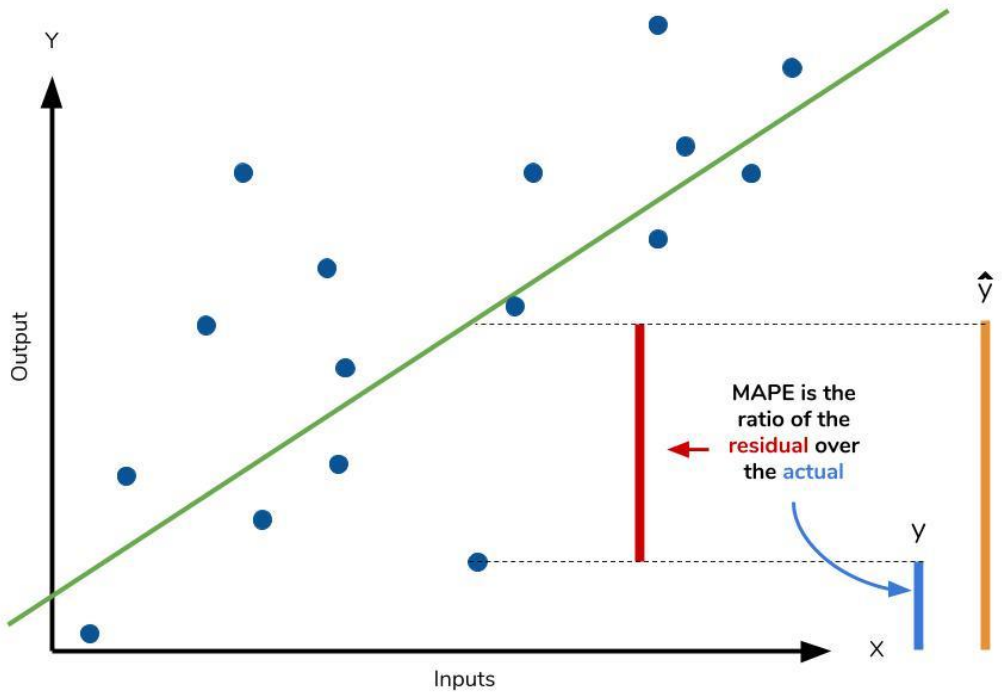


- $RMSE = \sqrt{\frac{\sum(y-\hat{y})^2}{N}}$
- RMSE는 MSE에 루트를 씌워 정의
- RMSE는 오류 지표를 실제 값과 유사한 단위로 다시 변환하여 해석을 쉽게 함

https://partrita.github.io/posts/regression-error/

Mean Absolute Percentage Error(MAPE)

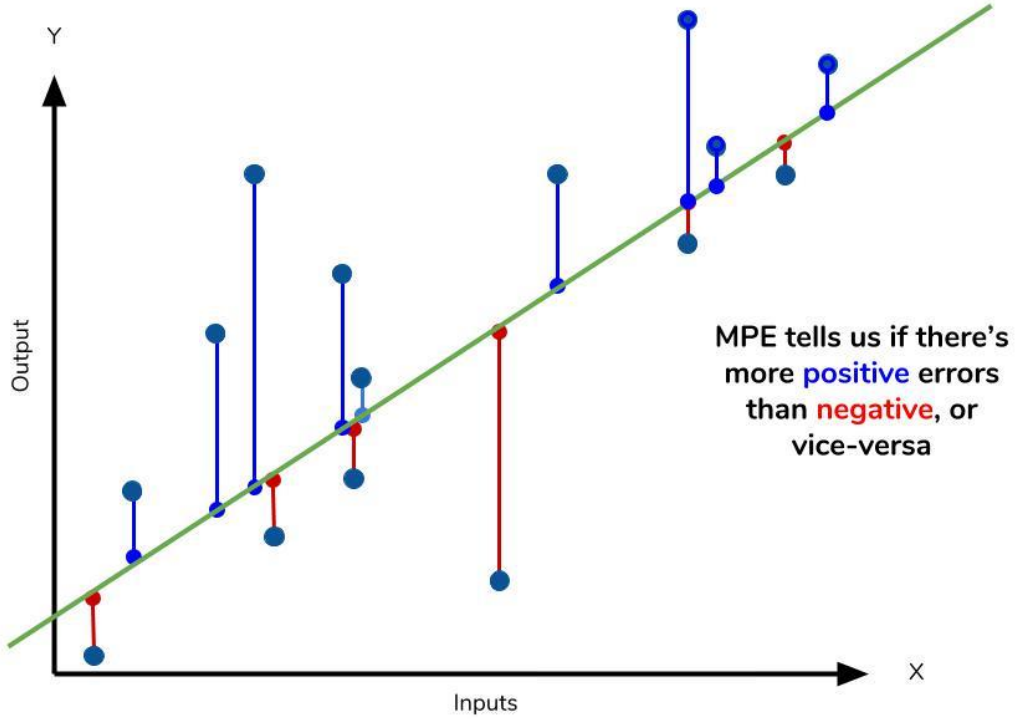
- $MAPE = \frac{\sum\left|\frac{y-\hat{y}}{y}\right|}{N} * 100\%$
- MAE와 마찬가지로 MSE보다 특이치에 robust하지만, MAE와 같은 단점을 가짐
- 추가적으로 모델에 대한 편향이 존재
- 이 단점으로 인해 MPE도 추가로 확인하는게 좋음
- 0 근처의 값에서는 사용하기 어려움



https://partrita.github.io/posts/regression-error/

Mean Percentage Error(MPE)

- $MPE = \frac{\sum\left(\frac{y-\hat{y}}{y}\right)}{N} * 100\%$
- MAPE에서 절대값을 제외한 지표
- MPE의 가장 큰 장점은 모델이 underperformance 인지 overperformance 인지 판단 할 수 있다는 것

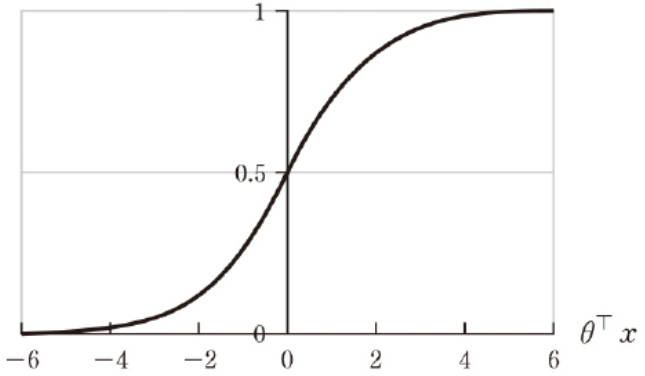


https://partrita.github.io/posts/regression-error/

로지스틱 회귀(Logistic Regression)

- 로지스틱 회귀는 이름에 회귀라는 단어가 들어가지만, 가능한 클래스가 2개인 이진 분류를 위한 모델
- 학습 데이터 : $\{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}, y_1 \in \{0, 1\}$
- 로지스틱 함수를 이용하여 함수 근사

$$f(\boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^\top \boldsymbol{x}}}$$

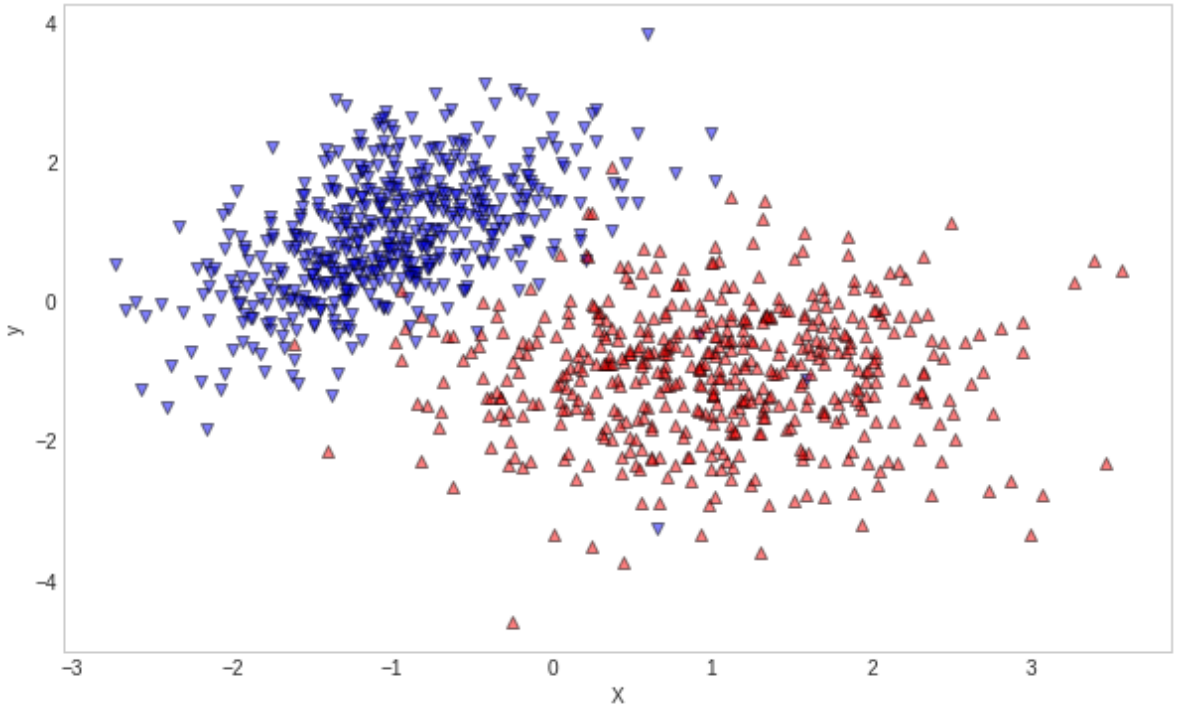


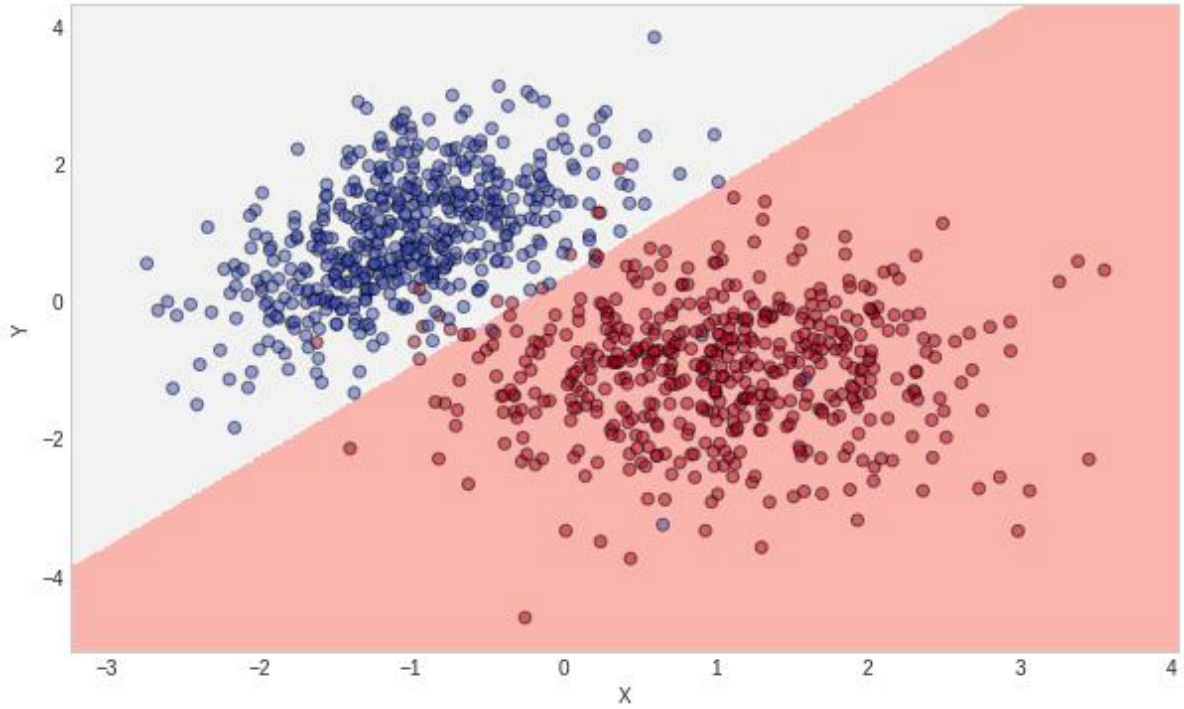
- 학습시 목적 함수

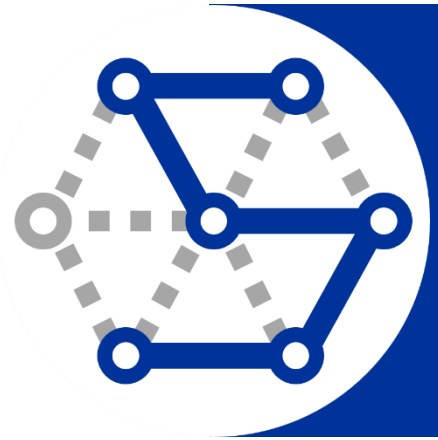
$$J(\boldsymbol{\theta}) = -\frac{1}{N}\sum_{i=1}(y_i \log f(\boldsymbol{x}_i) + (1 - y_i)\log(1 - f(\boldsymbol{x}_i)))$$

  - 경사 하강법 사용 학습

로지스틱 회귀(Logistic Regression)



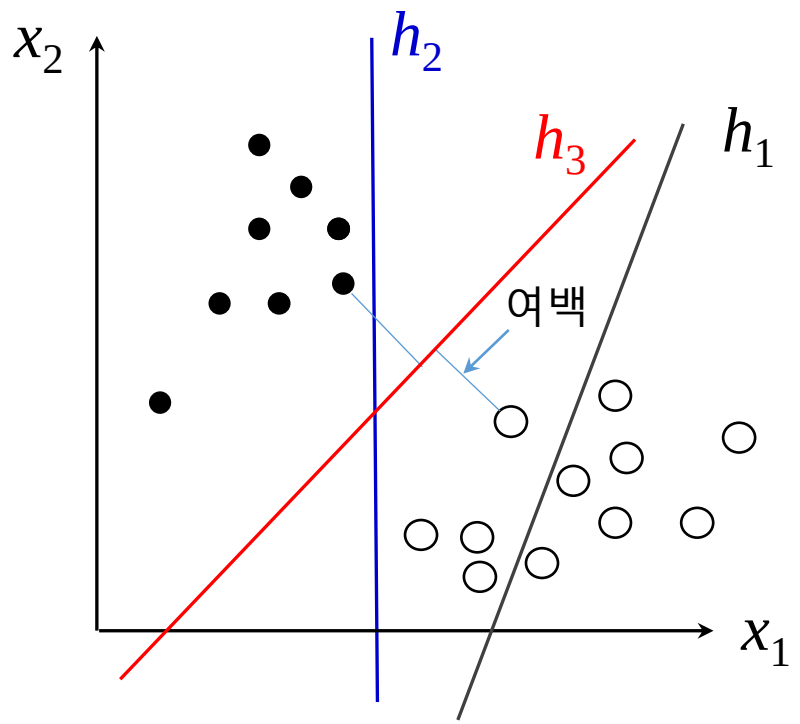




4 서포트 벡터 머신

Support Vector Machine (SVM)

- Vladimir Vapnik이 제안
- 분류 오차를 줄이면서 동시에 여백을 최대로 하는 결정 경계(decision boundary)를 찾는 이진 분류기(binary classifier)
- 회귀, 분류, 이상치 탐지 등에 사용되는 지도학습 방법
- 여백(margin): 결정 경계와 가장 가까이에 있는 학습 데이터까지의 거리
- 서포트 벡터(support vector): 결정 경계로부터 가장 가까이에 있는 학습 데이터들



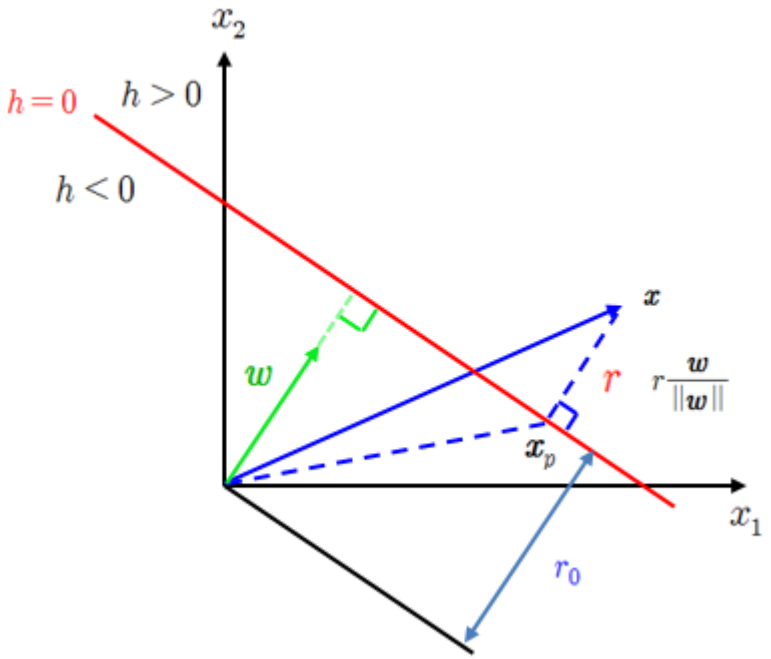
h_3가 h_2보다 우수

초평면(hyperplane)

- 4차원 이상의 공간에서 선형 방정식으로 표현되는 결정 경계

입력 데이터 $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_d \end{bmatrix}^\top$

$$h(\boldsymbol{x}) = w_1 x_1 + w_2 x_2 + \cdots + w_d x_d + b = \boldsymbol{w}^\top \boldsymbol{x} + b = 0$$



$$\boldsymbol{x} = \boldsymbol{x}_p + r\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}$$

$$\boldsymbol{w}^\top \boldsymbol{x} = \boldsymbol{w}^\top \boldsymbol{x}_p + r\frac{\boldsymbol{w}^\top \boldsymbol{w}}{\|\boldsymbol{w}\|}$$

$$\boldsymbol{w}^\top \boldsymbol{x} = \boldsymbol{w}^\top \boldsymbol{x}_p + r\|\boldsymbol{w}\|$$

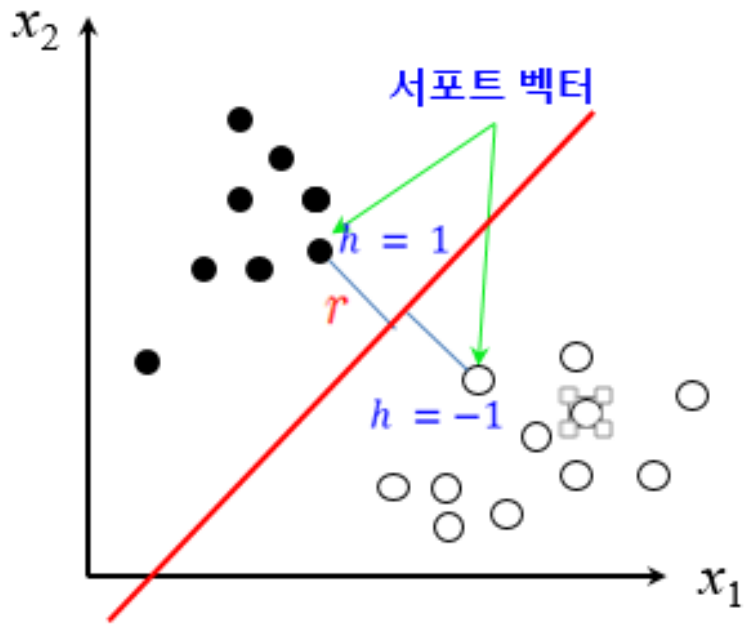
$$\boldsymbol{w}^\top \boldsymbol{x} + b = \boldsymbol{w}^\top \boldsymbol{x}_p + b + r\|\boldsymbol{w}\|$$

$$h(\boldsymbol{x}) = r\|\boldsymbol{w}\| \qquad r = \frac{h(\boldsymbol{x})}{\|\boldsymbol{w}\|}$$

$$r_0 = \frac{h(\boldsymbol{0})}{\|\boldsymbol{w}\|} = \frac{b}{\|\boldsymbol{w}\|}$$

SVM의 학습

- 학습데이터 $\boldsymbol{X} = \{(\boldsymbol{x}_1, t_1), (\boldsymbol{x}_2, t_2), \cdots, (\boldsymbol{x}_N, t_N)\}$ $t_i \in \{1, -1\}, \quad i = 1, \cdots, N$
- 분류를 위한 초평면의 만족조건



❶ $t_i h(\boldsymbol{x}_i) \geq 1, \quad i = 1, \cdots, N$

❷ 서포트 벡터와의 거리, 즉 여백을 최대로 한다.

조건 ❶ 서포트 벡터 $\boldsymbol{x}'$에서의 $|h(\boldsymbol{x}')| = 1$

$h(\boldsymbol{x}) > 0$인 공간에 $t_i = 1$

$h(\boldsymbol{x}) < 0$인 공간에 $t_i = -1$

조건 ❷ $r = \frac{h(\boldsymbol{x})}{\|\boldsymbol{w}\|}$ 서포트 벡터에 대해서는 $h(\boldsymbol{x}) = 1$

$r = \frac{1}{\|\boldsymbol{w}\|}$

Find $\boldsymbol{w}, b$ which minimizes $J(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2$

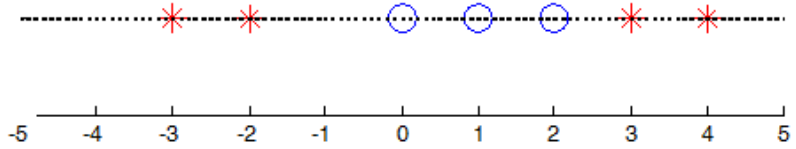
subject to $t_i h(\boldsymbol{x}_i) \geq 1, \quad i = 1, \cdots, N$

$h(\boldsymbol{x}) = \boldsymbol{w}^{\mathsf{T}}\boldsymbol{x} + b = 0$

비선형 SVM

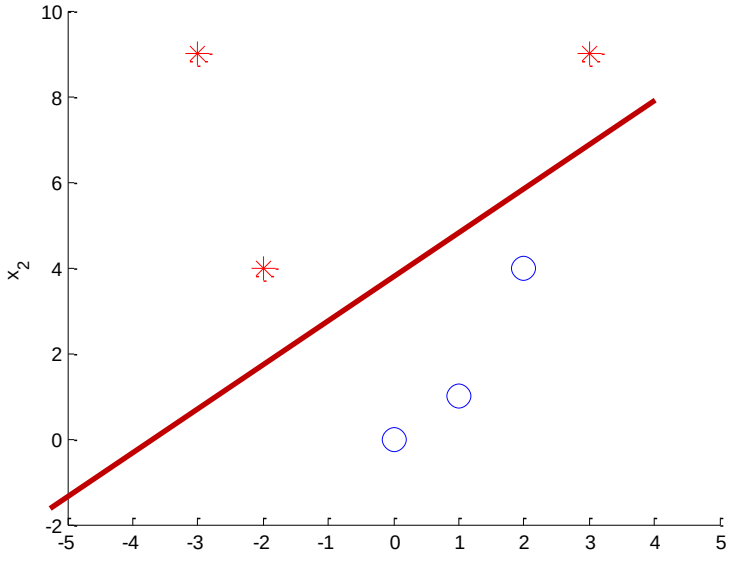
- 선형 SVM(linear SVM)
  - 선형인 초평면으로 공간 분할
- 데이터의 고차원 사상(high-dimensional mapping)
  - 데이터를 고차원 사상하면 선형 분리 가능

$$\Phi(x) = (x, x^2) = (x_1, x_2)$$



$$h(x_1, x_2) = w_1 x + w_2 x^2 + b$$

비선형 함수

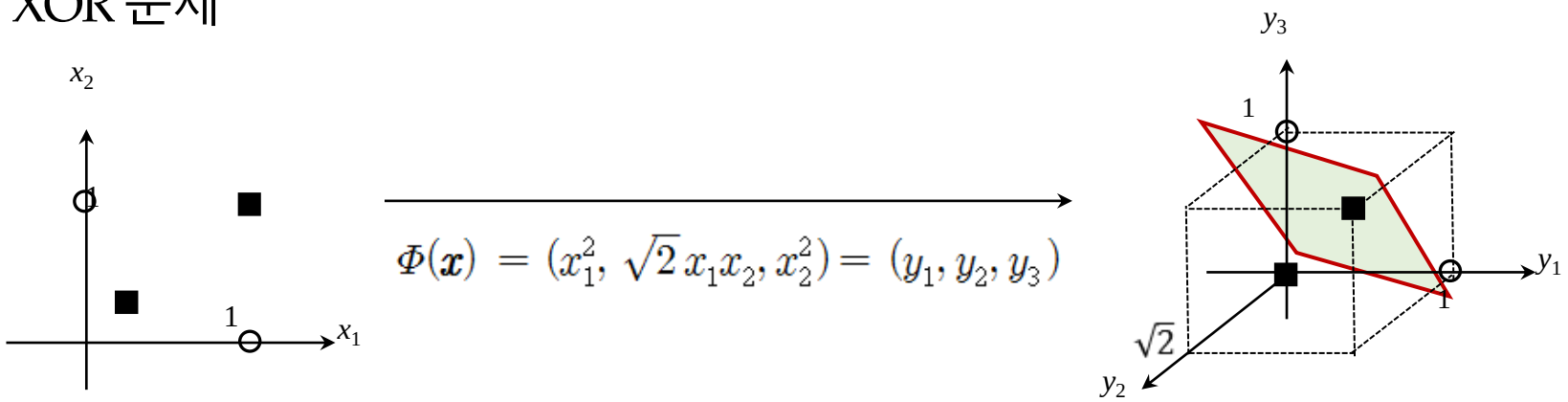


$$h(x_1, x_2) = w_1 x_1 + w_2 x_2 + b$$

선형 함수

비선형 SVM

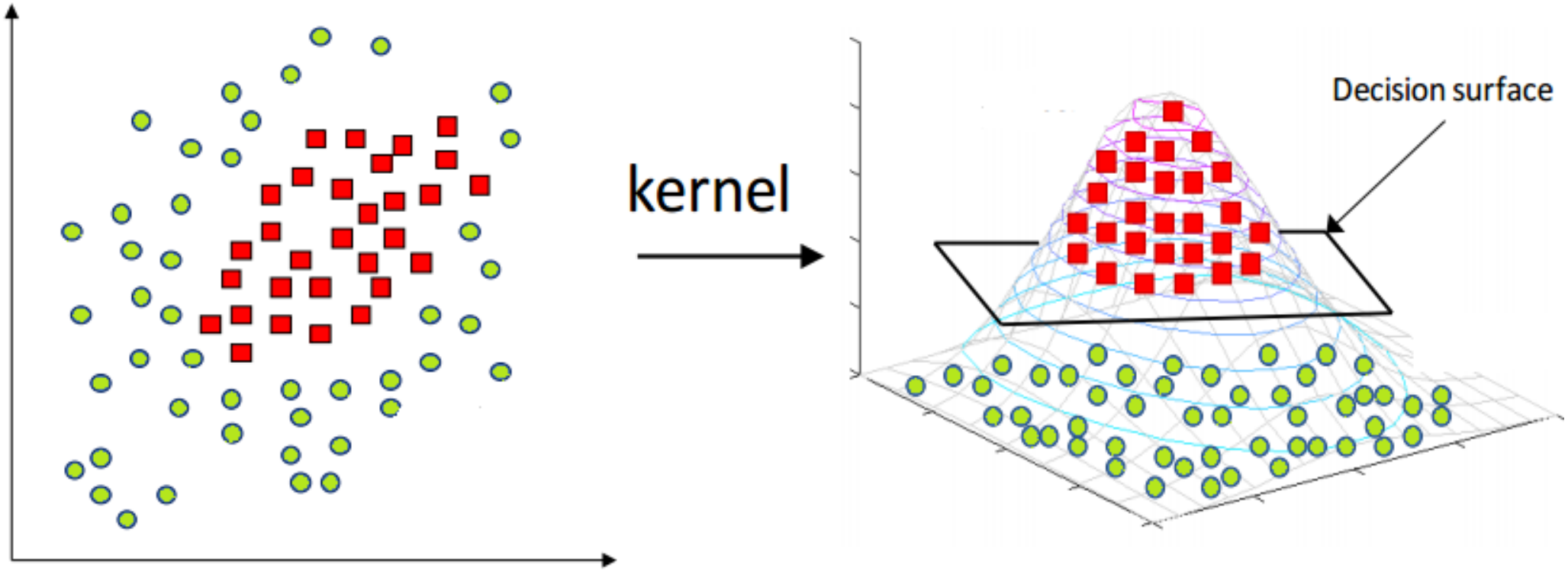
- 데이터의 고차원 사상
  - XOR 문제



- 고차원 변환의 문제점
  - 차원의 저주(curse of dimensionality) 문제 발생
  - 테스트 데이터에 대한 일반화(generalization) 능력 저하 가능
    - 여백(margin) 최대화를 통해 일반화 능력 유지
  - 계산 비용 증가
    - 커널 트릭(kernel trick) 사용으로 해결

비선형 SVM

- 커널 트릭(kernel trick)
  - 입력 데이터를 고차원 공간에 사상해서 비선형 특징을 학습할 수 있도록 확장하는 방법
  - 고차원 변환 없이 계산할 수 있는 커널 함수(kernel function) 사용



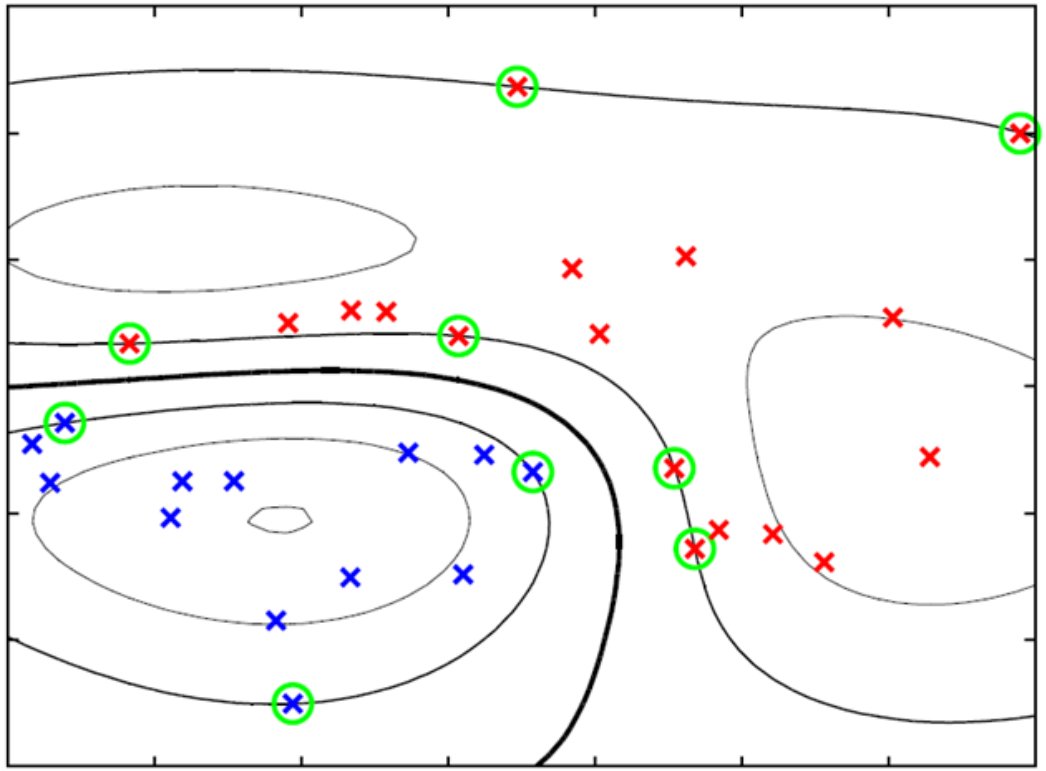
비선형 SVM

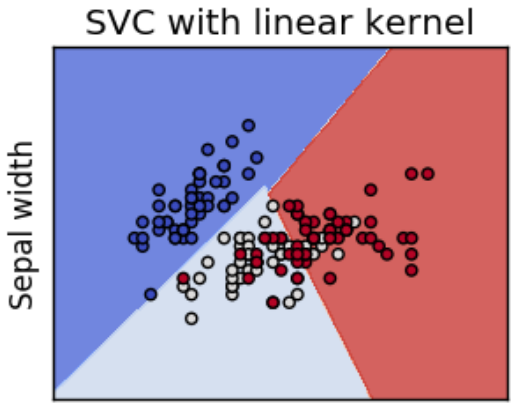
- 대표적인 커널 함수(kernel function)

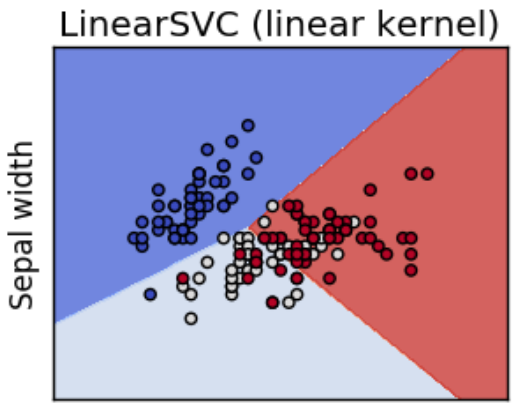
Polynomial	$k = (x_i, x_j) = (x_i, x_j + 1)^d$
Gaussian	$k = (x, y) = exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$
Gaussian RBF	$k(x_i, x_j) = exp\left(-\gamma \|x_i - x_j\|^2\right)$
Laplace RBF	$k(x, y) = exp\left(-\gamma - \frac{\|x - y\|}{\sigma}\right)$

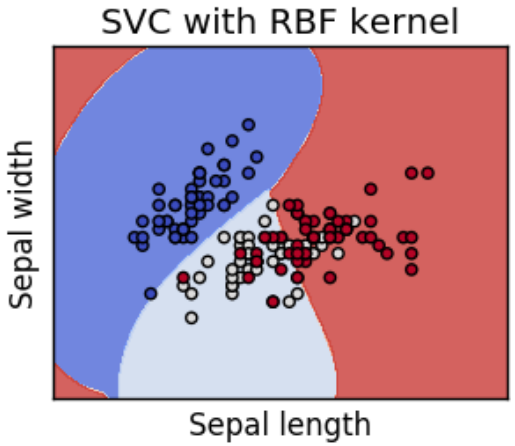
Sigmoid	$k(x, y) = tanh\left(\alpha x^T y + c\right)$
Hyperbolic Tangent	$k(x_i, x_j) = tanh\left(kx_i, x_j + c\right)$
Bessel Function of First Kind	$k(x, y) = \frac{J_v + 1(\sigma\|x - y\|)}{\|x - y\|^{-n(v+1)}}$
Anova Radial Basis	$k(x, y) = \sum_{k=1}^{n} \exp\left(-\sigma(x^k - y^k)^2\right)^d$

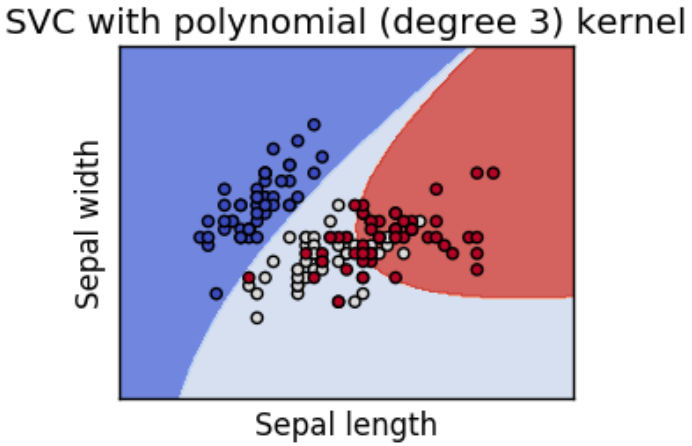
비선형 SVM




SVC with linear kernel
Sepal width
Sepal length


LinearSVC (linear kernel)
Sepal width
Sepal length

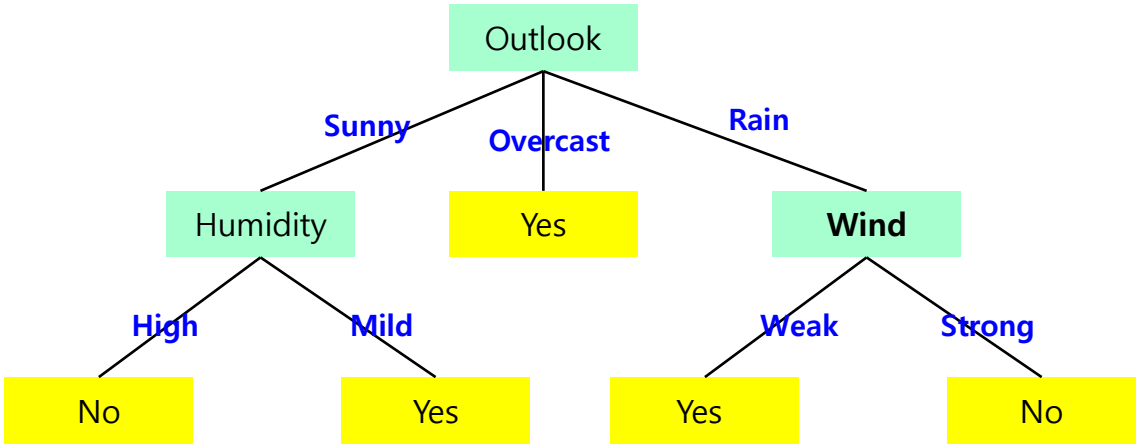

SVC with RBF kernel
Sepal width
Sepal length


SVC with polynomial (degree 3) kernel
Sepal width
Sepal length

5 결정 트리

결정 트리(decision tree)

- 트리 형태로 의사결정 지식을 표현한 것
- 내부 노드(internal node) : 비교 속성
- 간선(edge) : 속성 값
- 단말 노드(terminal node) : 부류(class), 대표값



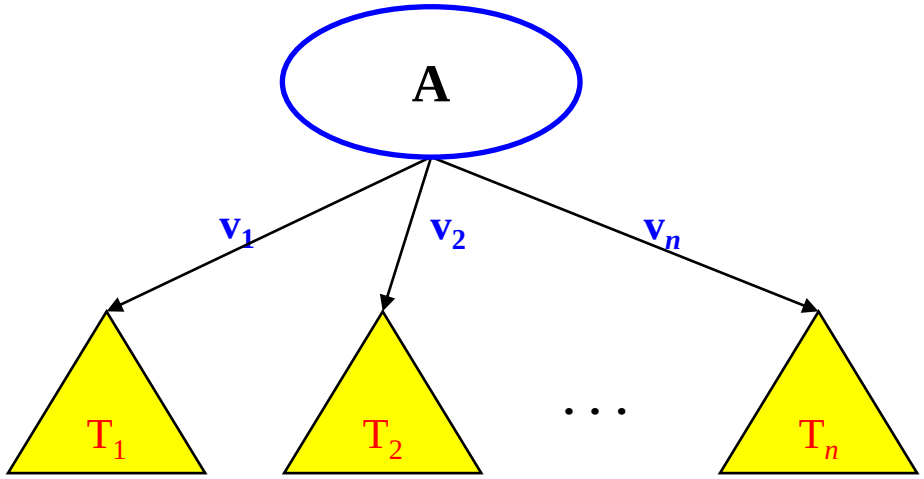
Day 날짜	Outlook 조망	Temperature 기온	Humidity 습도	Wind 바람	PlayTennis 테니스 여부
Day1	Sunny	Hot	High	Weak	No
Day2	Sunny	Hot	High	Strong	No
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	No
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	No
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	No

Outlook 조망	**Temperature** 기온	**Humidity** 습도	**Wind** 바람	**PlayTennis** 테니스 여부
Sunny	Hot	Mild	Weak	?
Rain	Hot	High	Weak	?

IF Outlook = Sunny **AND** Humidity = High **THEN** Answer = No

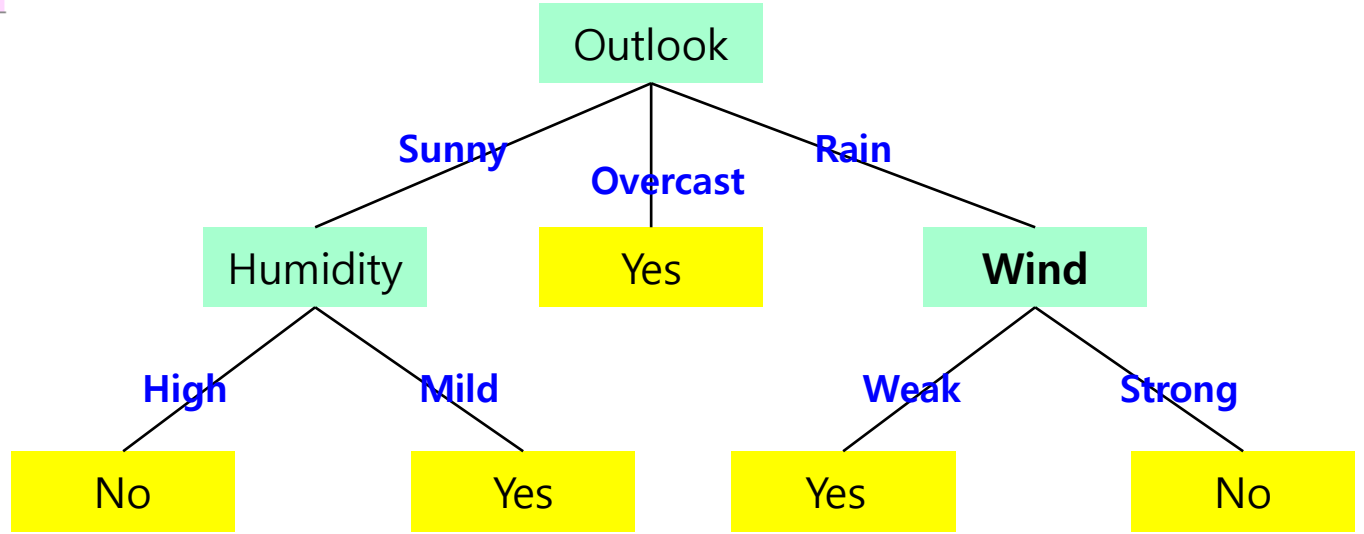
결정 트리 (decision tree) 알고리즘

- 모든 데이터를 포함한 하나의 노드로 구성된 트리에서 시작
- 반복적인 노드 분할 과정
  - 분할 속성(splitting attribute)을 선택
  - 속성값에 따라 서브트리(subtree)를 생성
  - 데이터를 속성값에 따라 분배

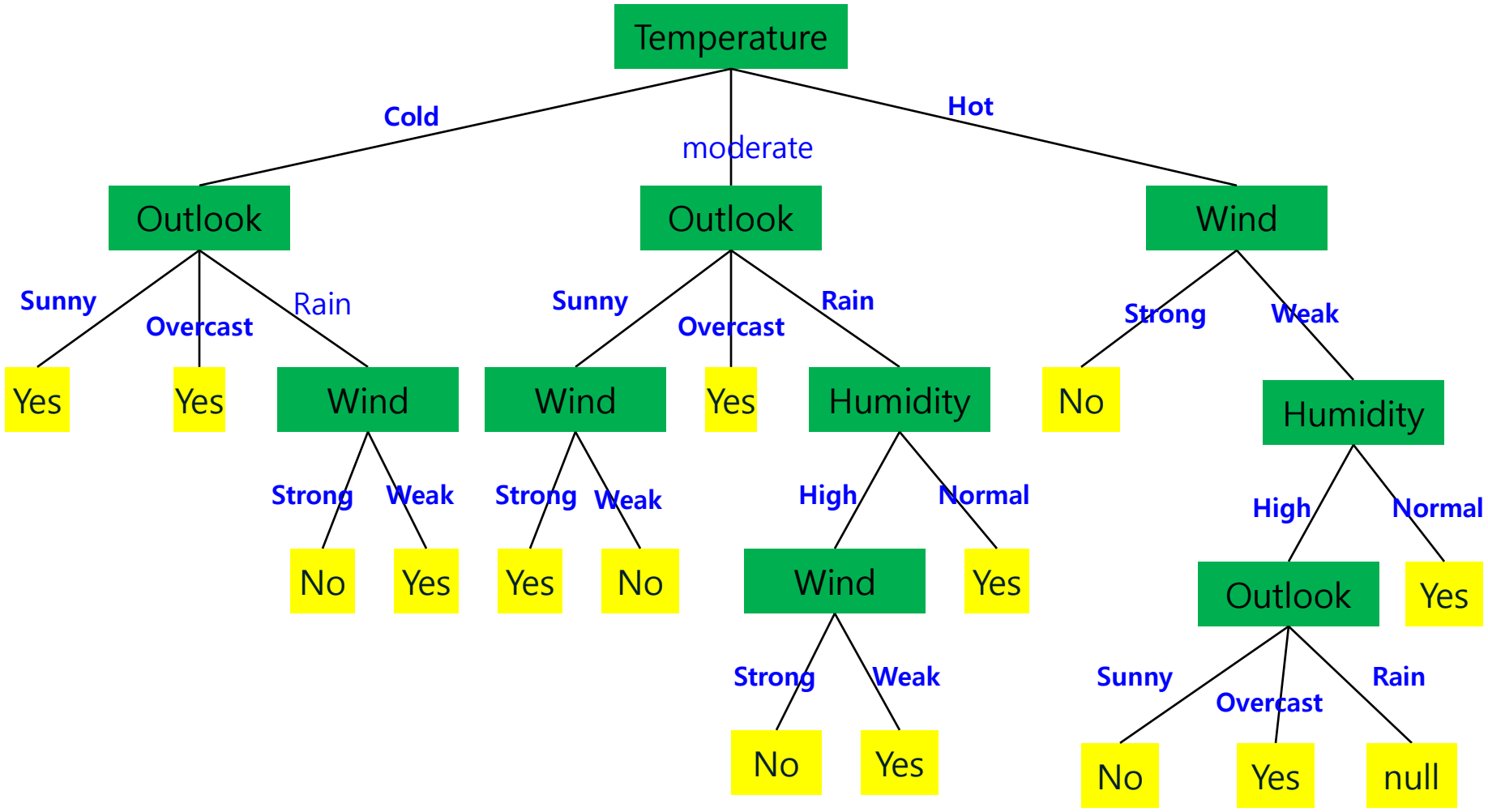


간단한 결정 트리

Day 날짜	Outlook 조망	Temperature 기온	Humidity 습도	Wind 바람	PlayTennis 테니스 여부
Day1	Sunny	Hot	High	Weak	No
Day2	Sunny	Hot	High	Strong	No
Day3	Overcast	Hot	High	Weak	Yes
Day4	Rain	Mild	High	Weak	Yes
Day5	Rain	Cool	Normal	Weak	Yes
Day6	Rain	Cool	Normal	Strong	No
Day7	Overcast	Cool	Normal	Strong	Yes
Day8	Sunny	Mild	High	Weak	No
Day9	Sunny	Cool	Normal	Weak	Yes
Day10	Rain	Mild	Normal	Weak	Yes
Day11	Sunny	Mild	Normal	Strong	Yes
Day12	Overcast	Mild	High	Strong	Yes
Day13	Overcast	Hot	Normal	Weak	Yes
Day14	Rain	Mild	High	Strong	No



복잡한 결정 트리

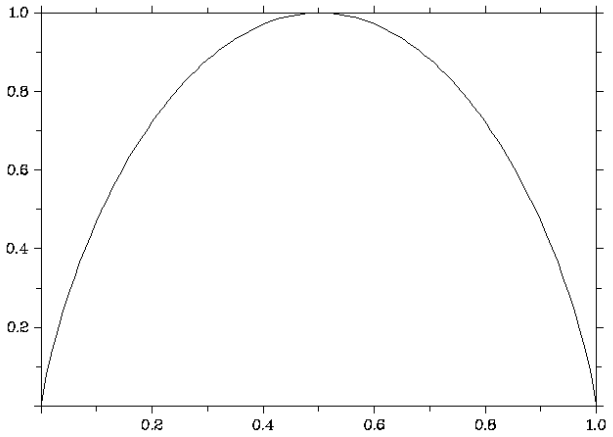

Temperature
Cold
moderate
Hot
Outlook
Outlook
Wind
Sunny
Overcast
Rain
Sunny
Overcast
Rain
Strong
Weak
Yes
Yes
Wind
Wind
Yes
Humidity
No
Humidity
Strong
Weak
Strong
Weak
High
Normal
High
Normal
No
Yes
Yes
No
Wind
Yes
Outlook
Yes
Strong
Weak
Sunny
Overcast
Rain
No
Yes
No
Yes
null

분할 속성(splitting attribute) 결정

- 어떤 속성을 선택하는 것이 효율적인가
  - 분할한 결과가 가능하면 동질적인(pure) 것으로 만드는 속성 선택
- 엔트로피(Entropy)
  - 동질적인 정도 측정 가능 척도
  - 원래 정보량(amount of information) 측정 목적의 척도

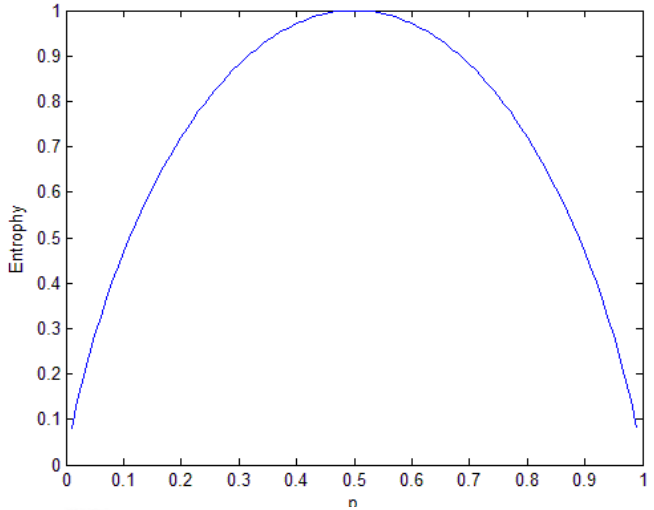
$$I = -\sum_{c} p(c) \log_2 p(c)$$

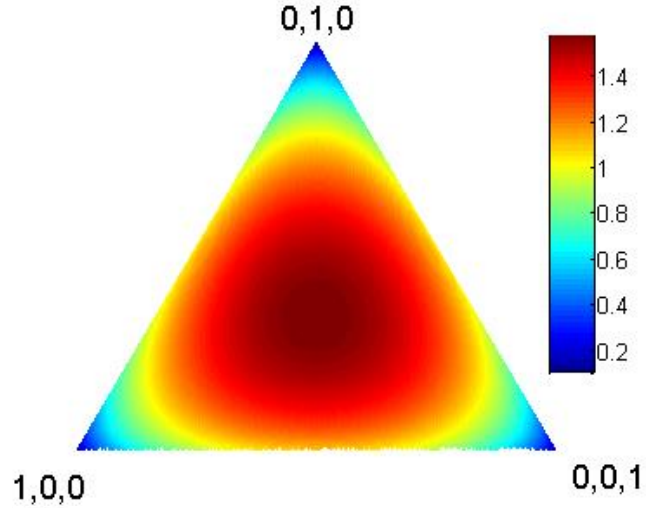
  - $-p(c)$: 부류 c 에 속하는 것의 비율
  - 2개 부류가 있는 경우 엔트로피

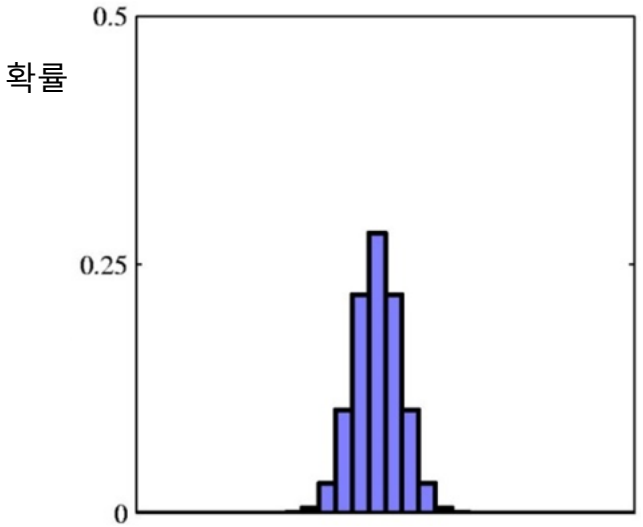


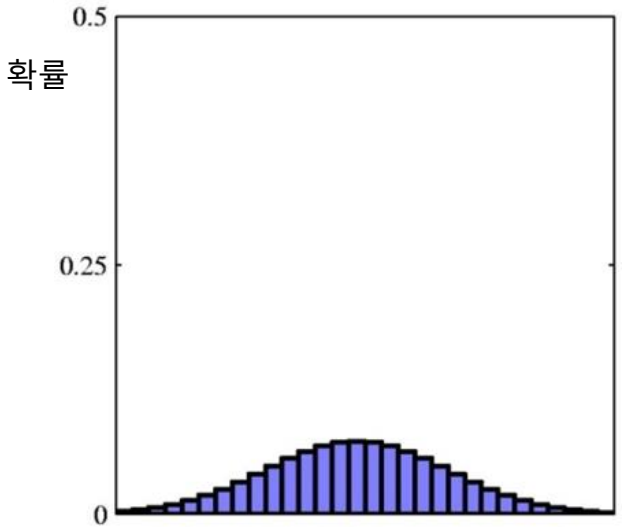
엔트로피의 특성

- 섞인 정도가 클 수록 큰 값







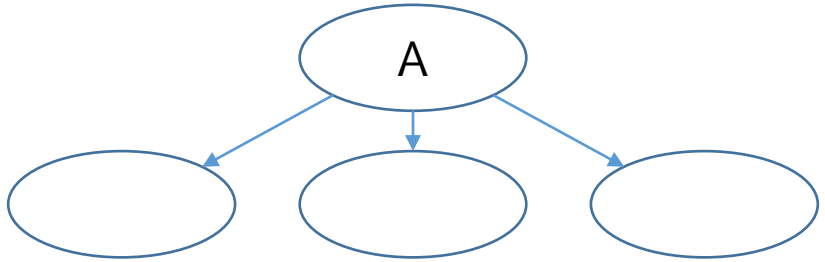


정보 이득(information gain)

- $IG = I - I_{res}$
- I_{res} : 특정 속성으로 분할한 후의 각 부분집합의 정보량의 가중평균

$$I_{res} = -\sum_{v} p(v) \sum_{c} p(c|v) \log_2 p(c|v)$$

$$IG = I - I_{res}(A) = -\sum_{c} p(c) \log_2 p(c) + \sum_{v} p(v) \sum_{c} p(c|v) \log_2 p(c|v)$$



- 정보이득이 클수록 우수한 분할 속성

학습 데이터의 예

- 부류(class) 정보가 있는 데이터

	속성			부류
	Pattern	*Outline*	*Dot*	*Shape*
1	수직	점선	무	삼각형
2	수직	점선	유	삼각형
3	대각선	점선	무	사각형
4	수평	점선	무	사각형
5	수평	실선	무	사각형
6	수평	실선	유	삼각형
7	수직	실선	무	사각형
8	수직	점선	무	삼각형
9	대각선	실선	유	사각형
10	수평	실선	무	사각형
11	수직	실선	유	사각형
12	대각선	점선	유	사각형
13	대각선	실선	무	사각형
14	수평	점선	유	삼각형

엔트로피 계산

- 9 □ (사각형)
- 5 Δ (삼각형)

- 부류별 확률(class probability)

$$p(\square) = \frac{9}{14}$$

$$p(\Delta) = \frac{5}{14}$$

- 엔트로피(entropy)

$$I = -\sum_{c} p(c) \log_2 p(c)$$

$$I = -\frac{9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log_2 \frac{5}{14} = 0.940 \text{ bits}$$

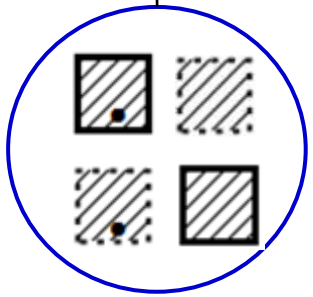
데이터 집합 분할과 정보이득

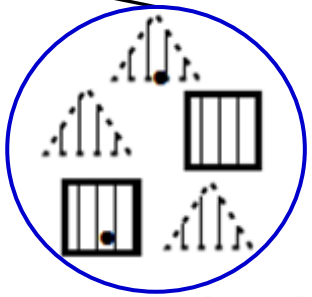
- Pattern 기준 분할

$$I = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.940$$

Pattern

수평 | 대각선 | 수직





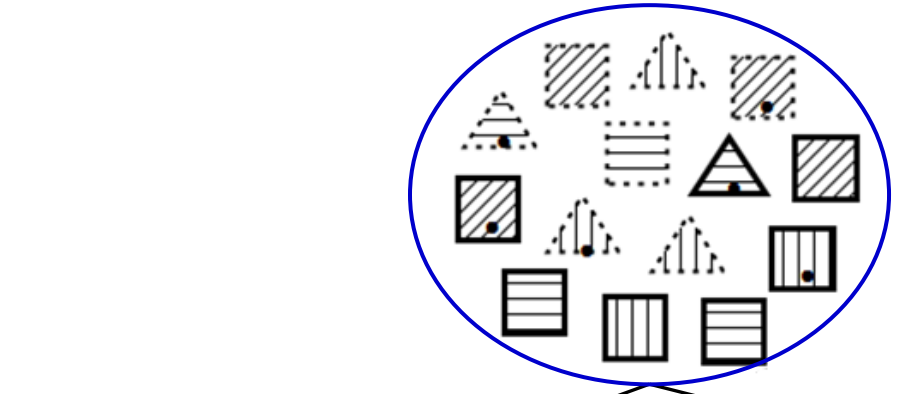
$$I_{horizontal} = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.971 \quad I_{diagonal} = -\frac{0}{4}\log_2\frac{0}{4} - \frac{4}{4}\log_2\frac{4}{4} = 0.0 \quad I_{vertical} = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.971$$

$$I_{res}(Pattern) = \sum p(v)I(v) = \frac{5}{14}\cdot 0.971 + \frac{4}{14}\cdot 0 + \frac{5}{14}\cdot 0.971 = 0.694$$

$$IG(Pattern) = I - I_{res}(Pattern) = 0.940 - 0.694 = 0.246$$

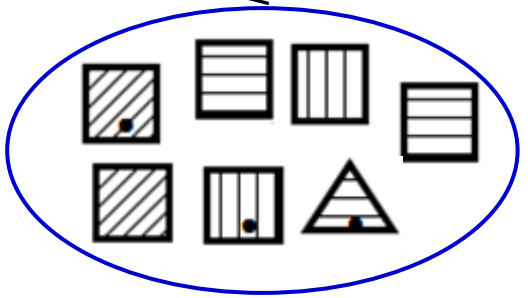
데이터 집합 분할과 정보이득

- Outline 기준 분할



$$I = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.940$$

점선 Outline 실선



$$I_{dashed} = -\frac{3}{7}\log_2\frac{3}{7} - \frac{4}{7}\log_2\frac{4}{7} = 0.985$$

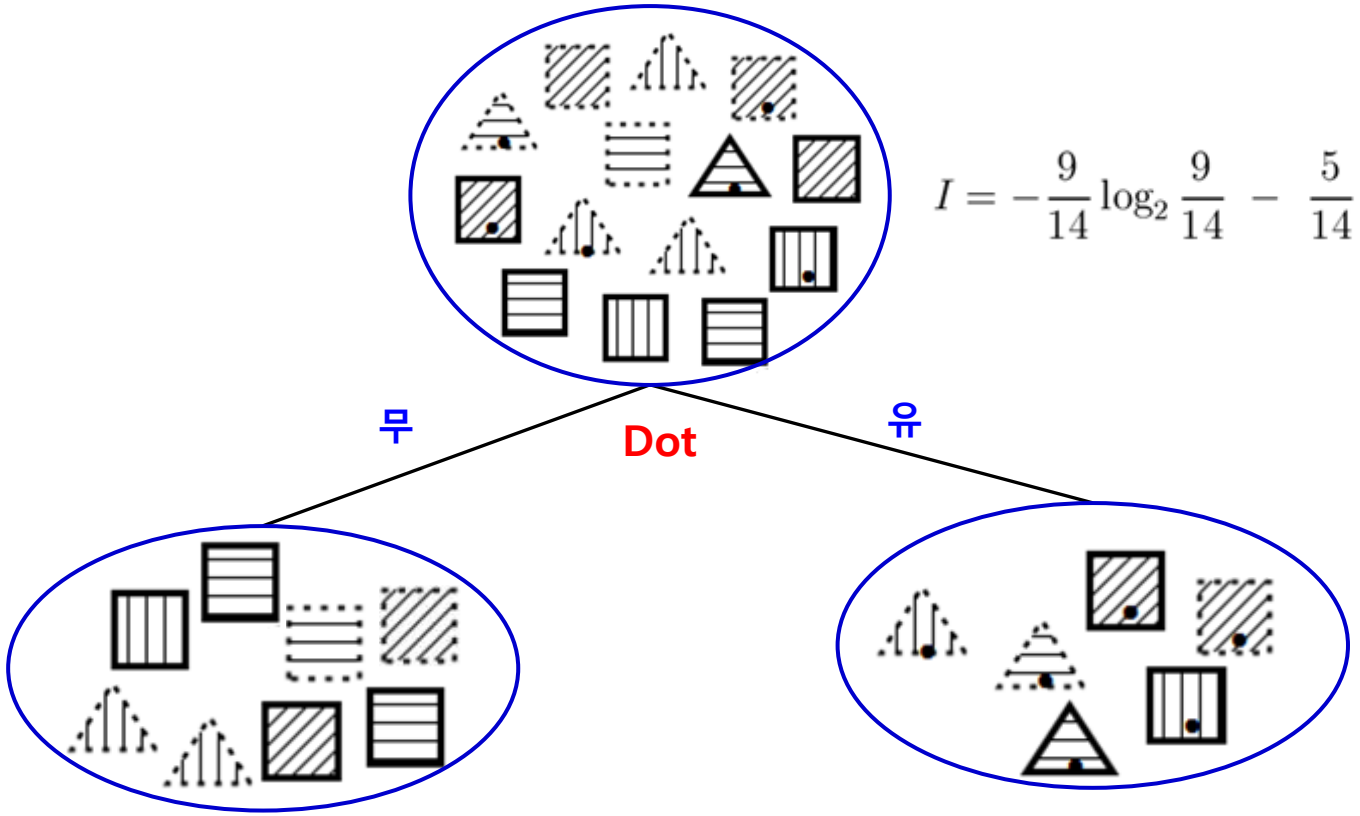
$$I_{solid} = -\frac{6}{7}\log_2\frac{6}{7} - \frac{1}{7}\log_2\frac{1}{7} = 0.592$$

$$I_{res}(Outline) = \sum p(v)I(v) = \frac{7}{14}\cdot 0.985 + \frac{7}{14}\cdot 0.592 = 0.789$$

$$IG(Outline) = I - I_{res}(Outline) = 0.940 - 0.789 = 0.151$$

데이터 집합 분할과 정보이득

- Dot 기준 분할



$$I = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.940$$

$$I_{no} = -\frac{2}{8}\log_2\frac{2}{8} - \frac{6}{8}\log_2\frac{6}{8} = 0.811$$

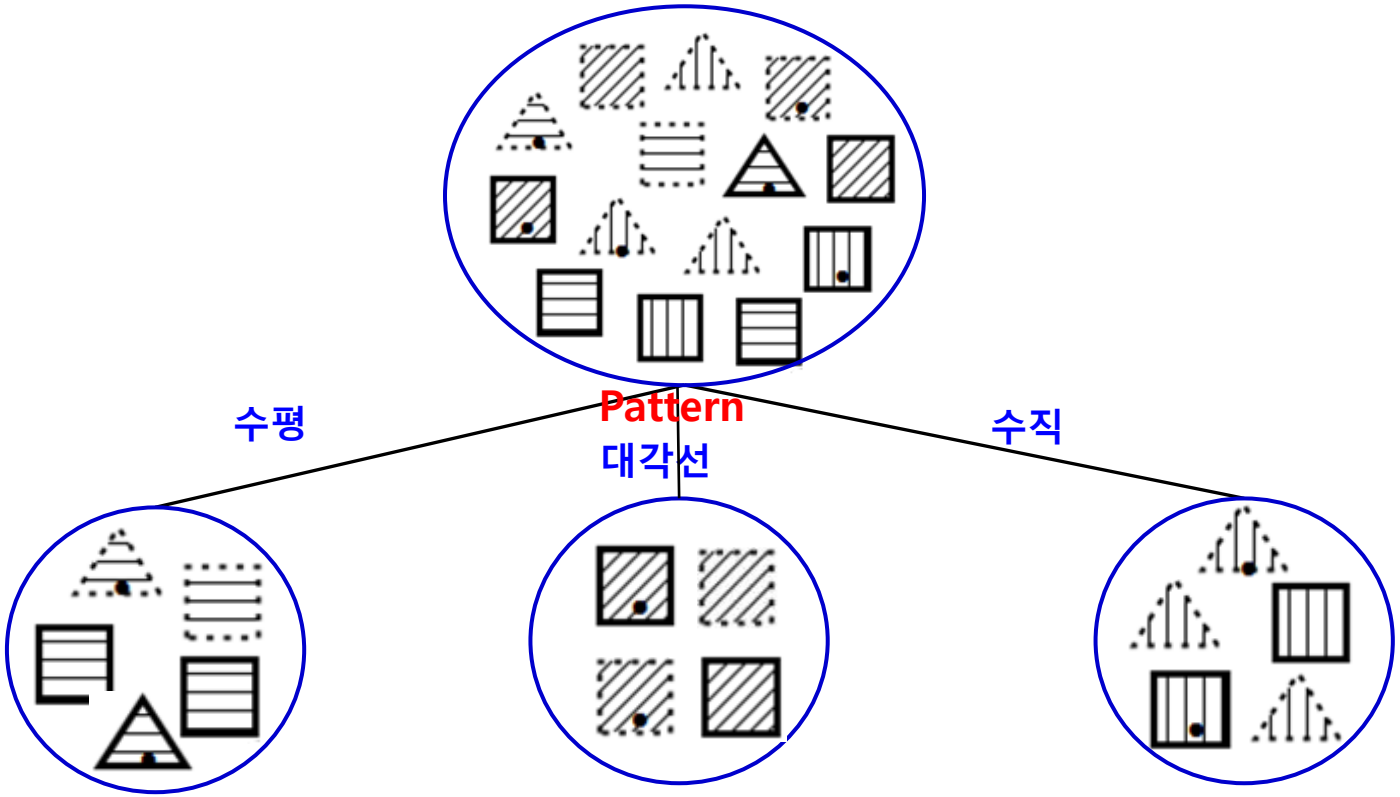
$$I_{yes} = -\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6} = 1.000$$

$$I_{res}(Dot) = \sum p(v)I(v) = \frac{8}{14}\cdot 0.811 + \frac{6}{14}\cdot 1.000 = 0.892$$

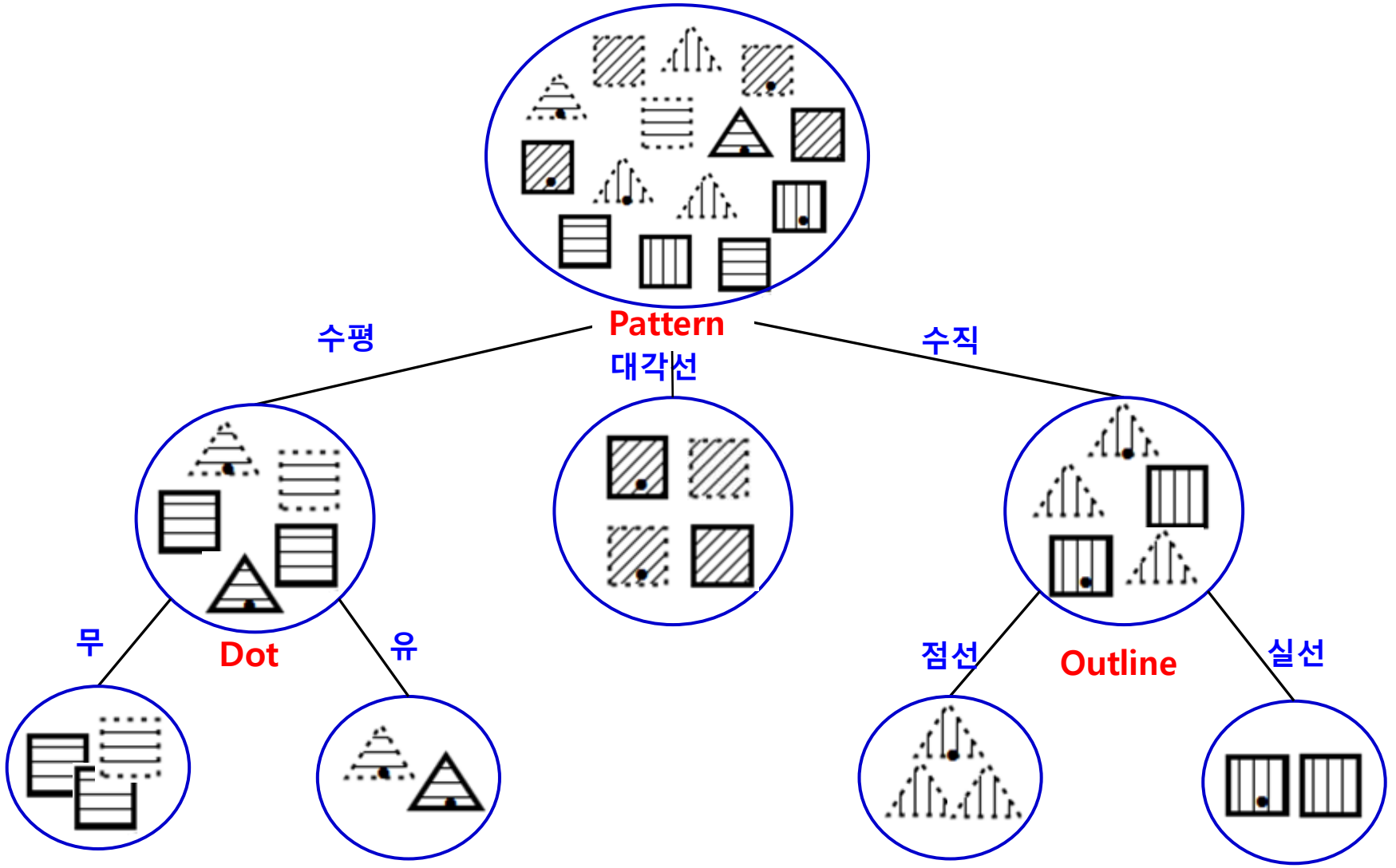
$$IG(Dot) = I - I_{res}(Dot) = 0.940 - 0.892 = 0.048$$

데이터 집합 분할과 정보이득

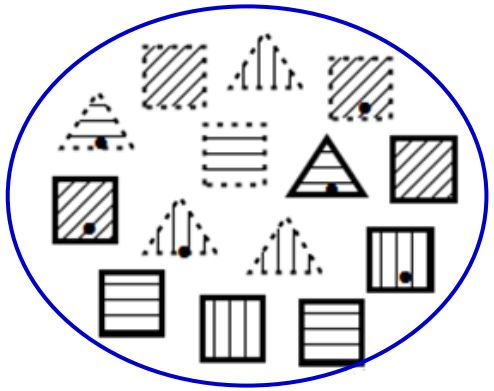
- 속성별 정보 이득
  - IG(Pattern) = 0.246
  - IG(Outline) = 0.151
  - IG(Dot) = 0.048
- 분할속성 선택
  - 정보이득이 큰 것 선택
  - Pattern 선택

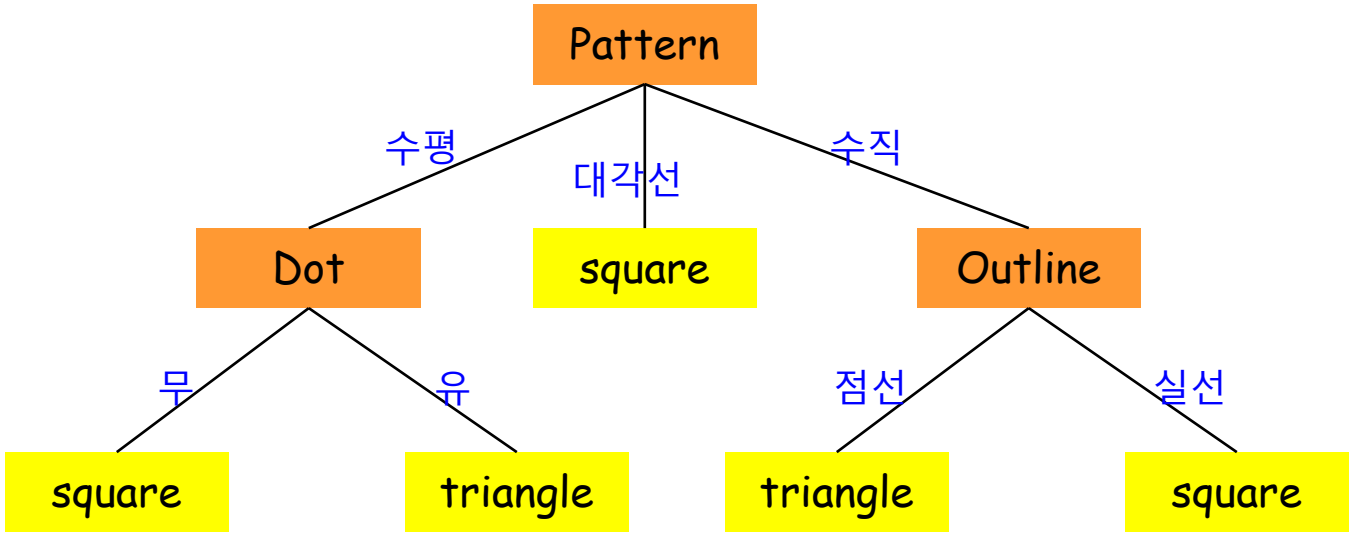


데이터 집합 분할과 정보이득



최종 결정트리




Pattern
수평
대각선
수직
Dot
square
Outline
무
유
점선
실선
square
triangle
triangle
square

정보이득(information gain) 척도의 단점

$$IG = I - I_{res}(A) = -\sum_{c} p(c)\log_2 p(c) + \sum_{v} p(v) \sum_{c} p(c|v)\log_2 p(c|v)$$

- 속성값이 많은 것 선호
  - 예: 학번, 이름 등
- 속성값이 많으면 데이터집합을 많은 부분집합으로 분할
  - 작은 부분집합은 동질적인 경향

- 개선 척도
  - 정보이득비(information gain ratio)
  - 지니 지수(Gini index)

정보이득 비(information gain ratio) 척도

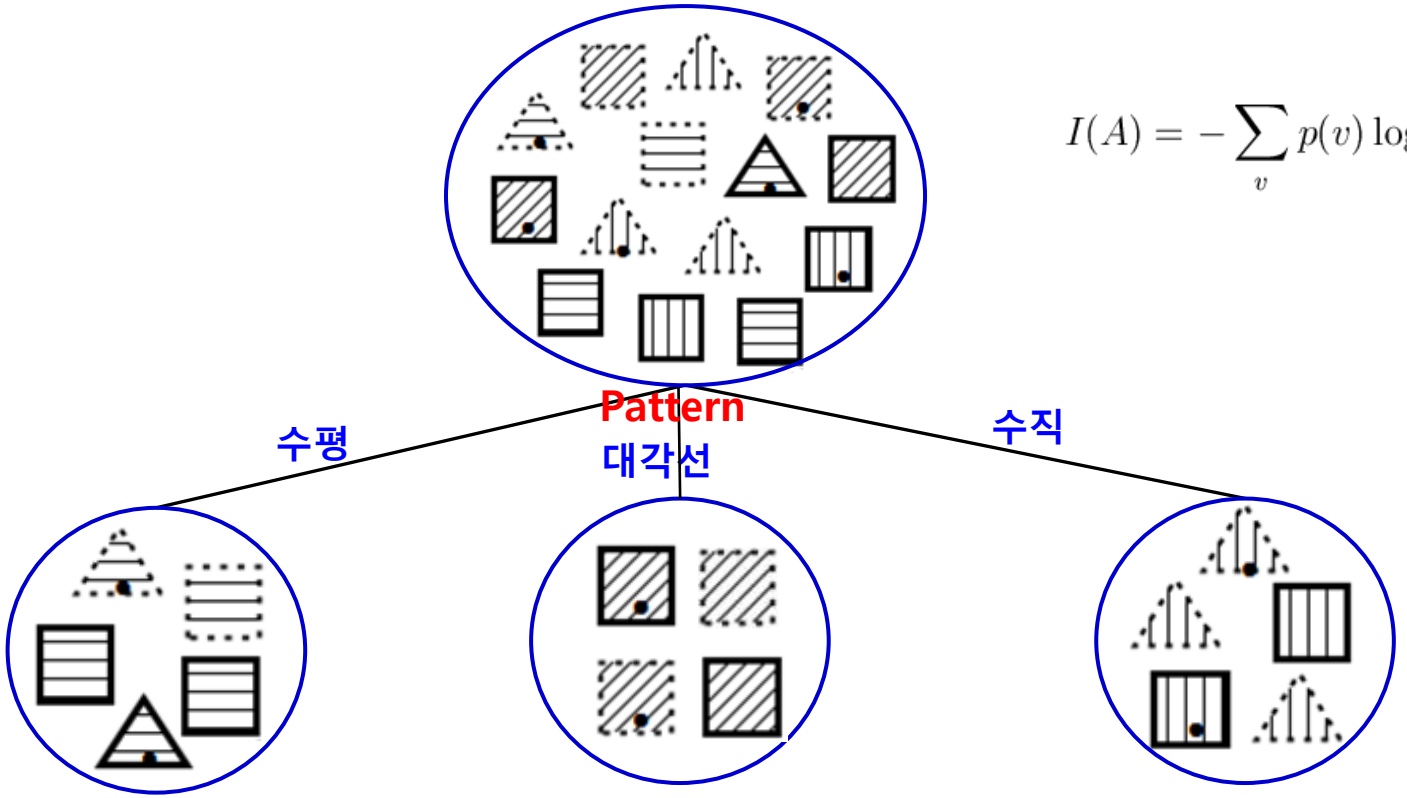
- 정보이득(information gain) 척도를 개선한 것
- 속성값이 많은 속성에 대해 불이익

$$GainRatio(A) = \frac{IG(A)}{I(A)} = \frac{I - I_{res}(A)}{I(A)}$$

- $I(A)$
  - 속성 A의 속성값을 부류(class)로 간주하여 계산한 엔트로피
  - 속성값이 많을수록 커지는 경향

$$I(A) = -\sum_{v} p(v)\, log_2(p(v))$$

정보이득 비



$$I(A) = -\sum_{v} p(v)\log_2(p(v))$$

$$I(Pattern) = -\frac{5}{14}\log_2\frac{5}{14} - \frac{5}{14}\log_2\frac{5}{14} - \frac{4}{14}\log_2\frac{4}{14} = 1.58$$

$$IG(Pattern) = I - I_{res}(Pattern) = 0.940 - 0.694 = 0.246$$

$$GainRatio(Pattern) = \frac{IG(Pattern)}{I(Pattern)} = \frac{0.940 - 0.694}{1.58} = 0.156$$

정보이득 vs 정보이득 비

속성	속성의 개수	정보이득	정보이득비
Pattern	3	0.247	0.156
Outline	2	0.152	0.152
Dot	2	0.048	0.049

지니 지수(Gini index)

- 데이터 집합에 대한 지니 값
- i, j가 부류(class)를 나타낼 때

$$Gini = \sum_{i \neq j} p(i)p(j)$$

- 속성 A에 대한 지니 지수값 가중평균

$$Gini(A) = \sum_{v} p(v) \sum_{i \neq j} p(i|v)p(j|v)$$

- 지니 지수 이득(gini index gain)

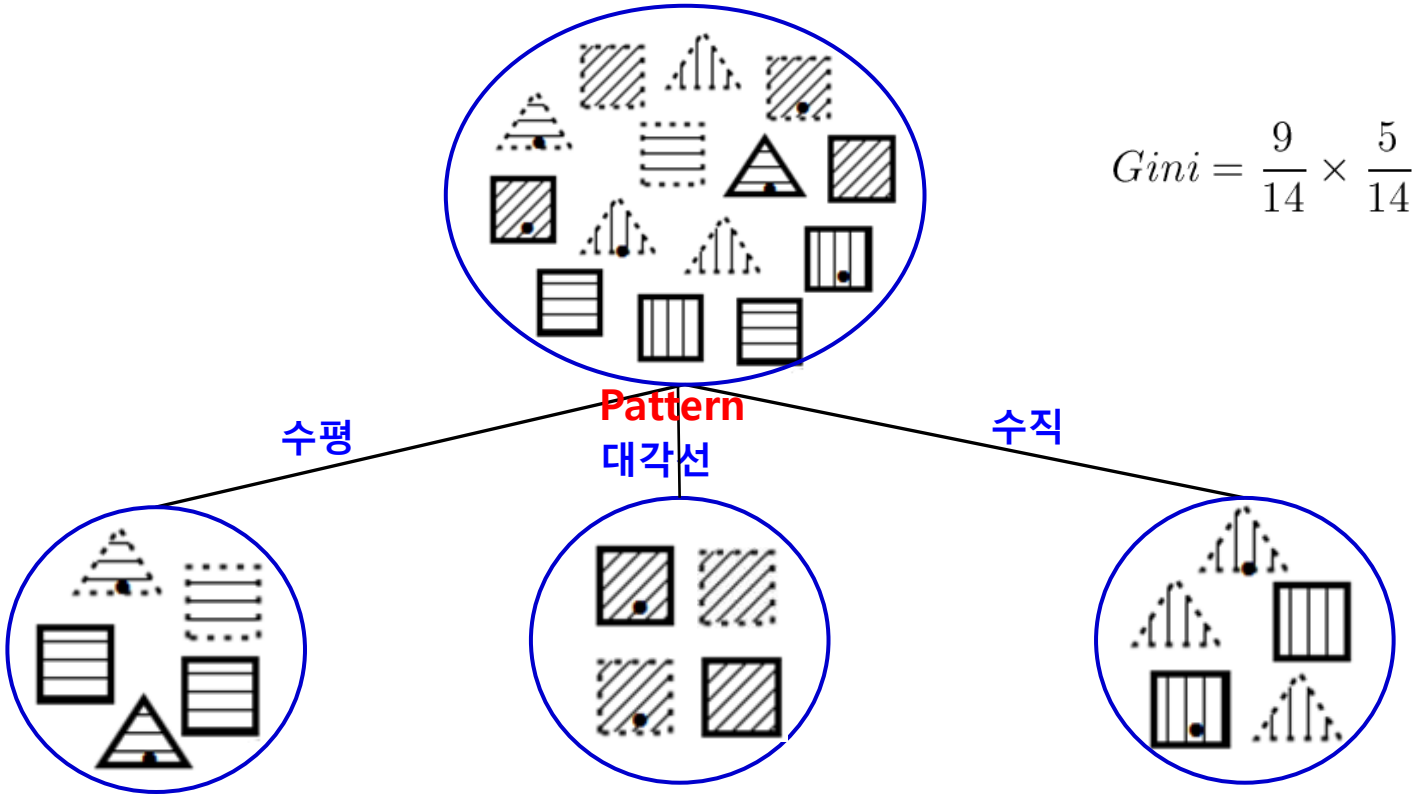
$$GiniGain(A) = Gini - Gini(A)$$

$$p(\square) = \frac{9}{14}$$

$$p(\triangle) = \frac{5}{14}$$

$$Gini = \frac{9}{14} \times \frac{5}{14} = 0.230$$

지니 지수(Gini index)



$$Gini = \frac{9}{14} \times \frac{5}{14} = 0.230$$

$$Gini(A) = \sum_{v} p(v) \sum_{i \neq j} p(i|v)p(j|v)$$

$$Gini(Pattern) = \frac{5}{14} \times (\frac{3}{5} \times \frac{2}{5}) + \frac{5}{14} \times (\frac{2}{5} \times \frac{3}{5}) + \frac{4}{14} \times (\frac{4}{4} \times \frac{0}{4}) = 0.171$$

$$Gini\,Gain(Pattern) = 0.230 - 0.171 = 0.058$$

분할속성 평가 척도 비교

속성	정보이득	정보이득비	지니이득
Pattern	0.247	0.156	0.058
Outline	0.152	0.152	0.046
Dot	0.048	0.049	0.015

결정트리 알고리즘

ID3 알고리즘

- **범주형(categorical) 속성값**을 갖는 데이터에 대한 결정트리 학습
- 예. PlayTennis, 삼각형/사각형 문제

C4.5 알고리즘

- 범주형 속성값과 수치형 속성값을 갖는 데이터로 부터 결정트리 학습
- ID3를 개선한 알고리즘

C5.0 알고리즘

- C4.5를 개선한 알고리즘

CART 알고리즘

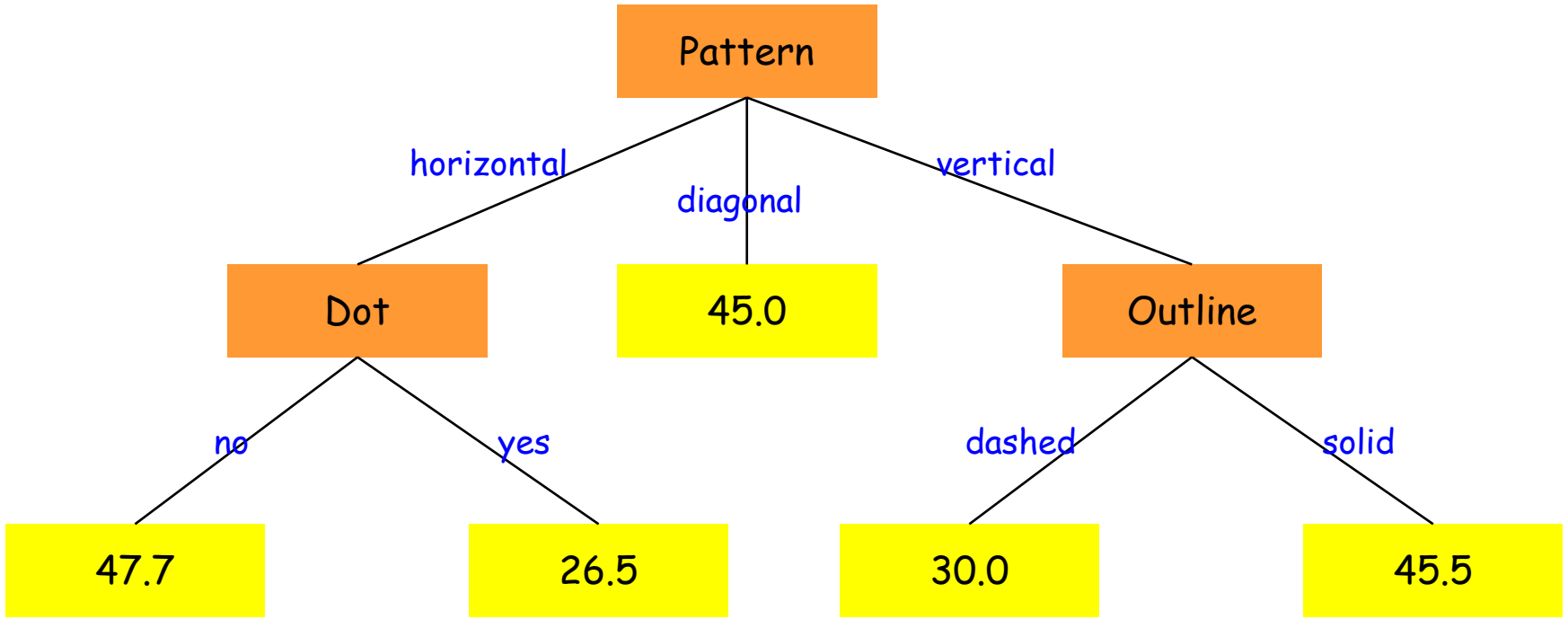
- 수치형 속성을 갖는 데이터에 대해 적용

회귀(regression)를 위한 결정트리

- 출력값이 수치값

	속성			면적
	Pattern	*Outline*	*Dot*	*Area*
1	수직	점선	무	25
2	수직	점선	유	30
3	대각선	점선	무	46
4	수평	점선	무	45
5	수평	실선	무	52
6	수평	실선	유	23
7	수직	실선	무	43
8	수직	점선	무	35
9	대각선	실선	유	38
10	수평	실선	무	46
11	수직	실선	유	48
12	대각선	점선	유	52
13	대각선	실선	무	44
14	수평	점선	유	30

회귀(regression)를 위한 결정트리



회귀(regression)를 위한 결정트리

- 분류를 위한 결정트리와 차이점
  - 단말노드가 부류(class)가 아닌 수치값(numerical value)임
  - 해당 조건을 만족하는 것들이 가지는 대표값
- 분할 속성 선택
  - 표준편차 축소(reduction of standard deviation) $\boldsymbol{SDR}$를 최대로 하는 속성 선택

$$\boldsymbol{SDR}(\boldsymbol{A}) = \boldsymbol{SD} - \boldsymbol{SD}(\boldsymbol{A})$$

  - **표준편차** $\boldsymbol{SD} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x_i} - \boldsymbol{m})^2}$ m : 평균
  - $\boldsymbol{SD}(A)$: 속성 A를 기준으로 **분할 후**의 **부분 집합별 표준표차**의 **가중평균**

회귀(regression)를 위한 결정트리

Area
26
30
48
46
62
23
43
36
38
48
48
62
44
30

$SD = 9.67$

		Area의 표준편차	개수
Pattern	수평	12.15	5
	수직	9.36	5
	대각선	5.77	4
			14

$$SD(Pattern) = \frac{5}{14} \times 12.15 + \frac{5}{14} \times 9.36 + \frac{4}{14} \times 5.77 = 9.05$$

$$SDR(Pattern) = SD - SD(Pattern) = 9.67 - 9.05 = 0.61$$

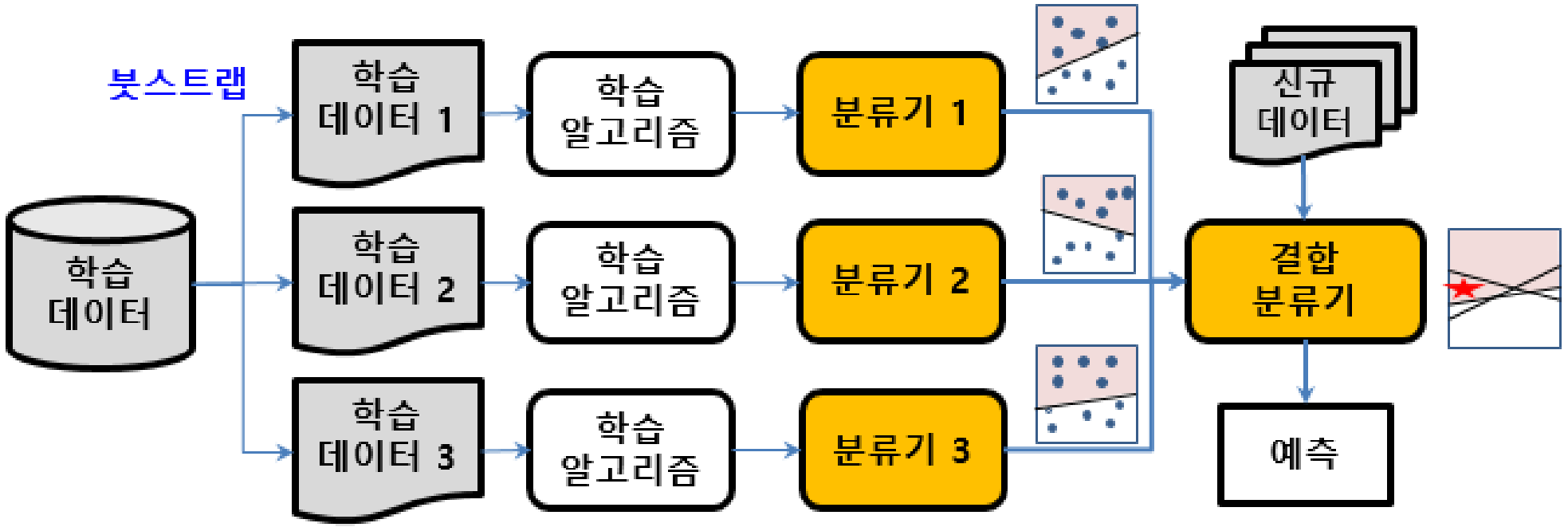
6 앙상블

앙상블 분류기 (ensemble classifier)

- 주어진 학습 데이터 집합에 대해서 여러 개의 서로 다른 분류기를 만들고, 이들 분류기의 판정 결과를 **투표 방식(voting method)**이나 **가중치 투표 방식(weighted voting method)**으로 결합
- 부트스트랩(bootstrap): 주어진 학습 데이터 집합에서 복원추출(resampling with replacement)하여 다수의 학습 데이터 집합을 만들어내는 기법
- 배깅(bagging, bootstrap aggregating)
- 부스팅(boosting)

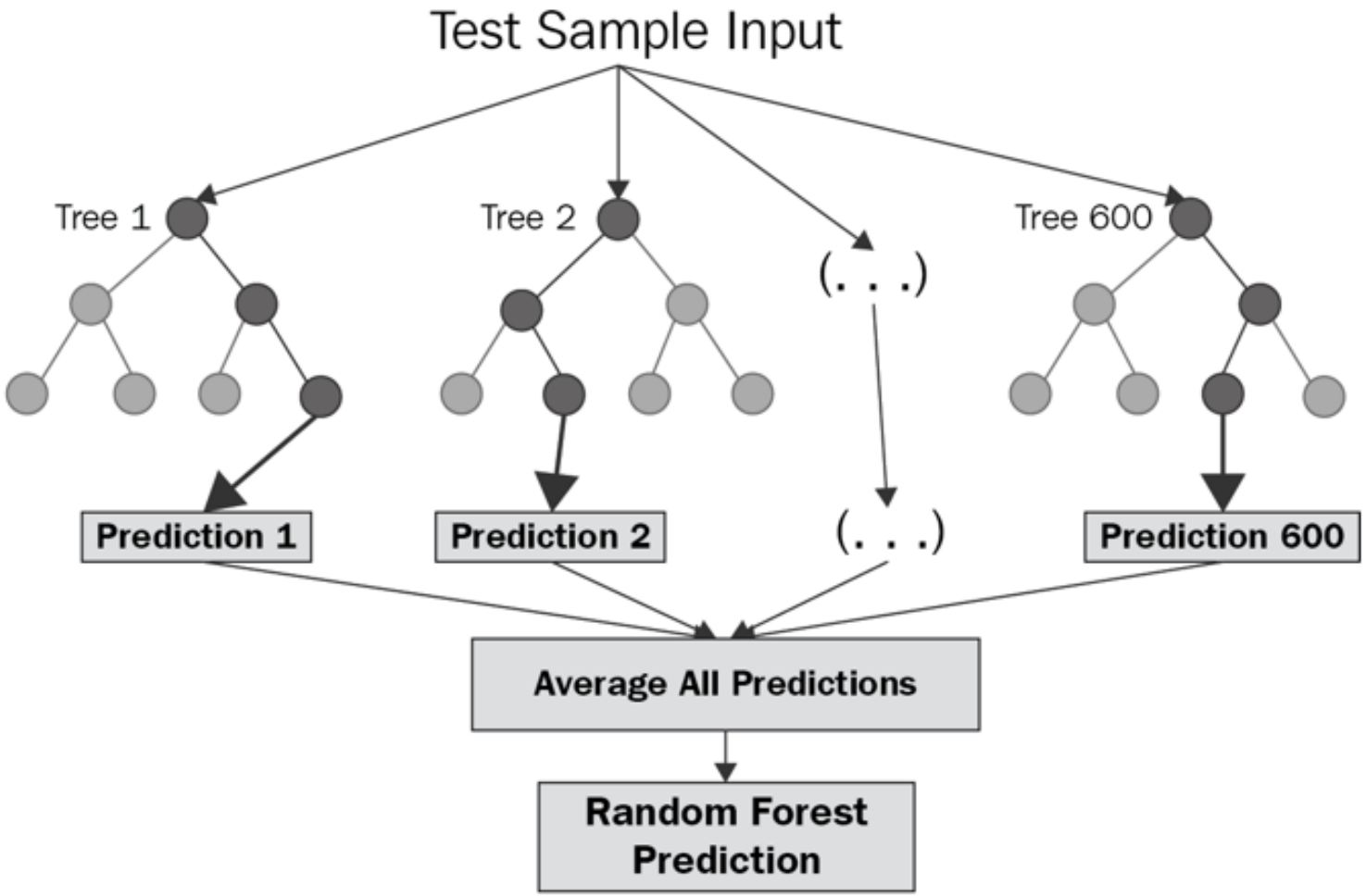
배깅(bagging, bootstrap aggregating)

- 부트스트랩을 통해 여러 개의 학습 데이터 집합을 만들고, 각 학습 데이터 집합별로 분류기를 만들어, 이들이 투표나 가중치 투표를 하여 최종 판정을 하는 기법



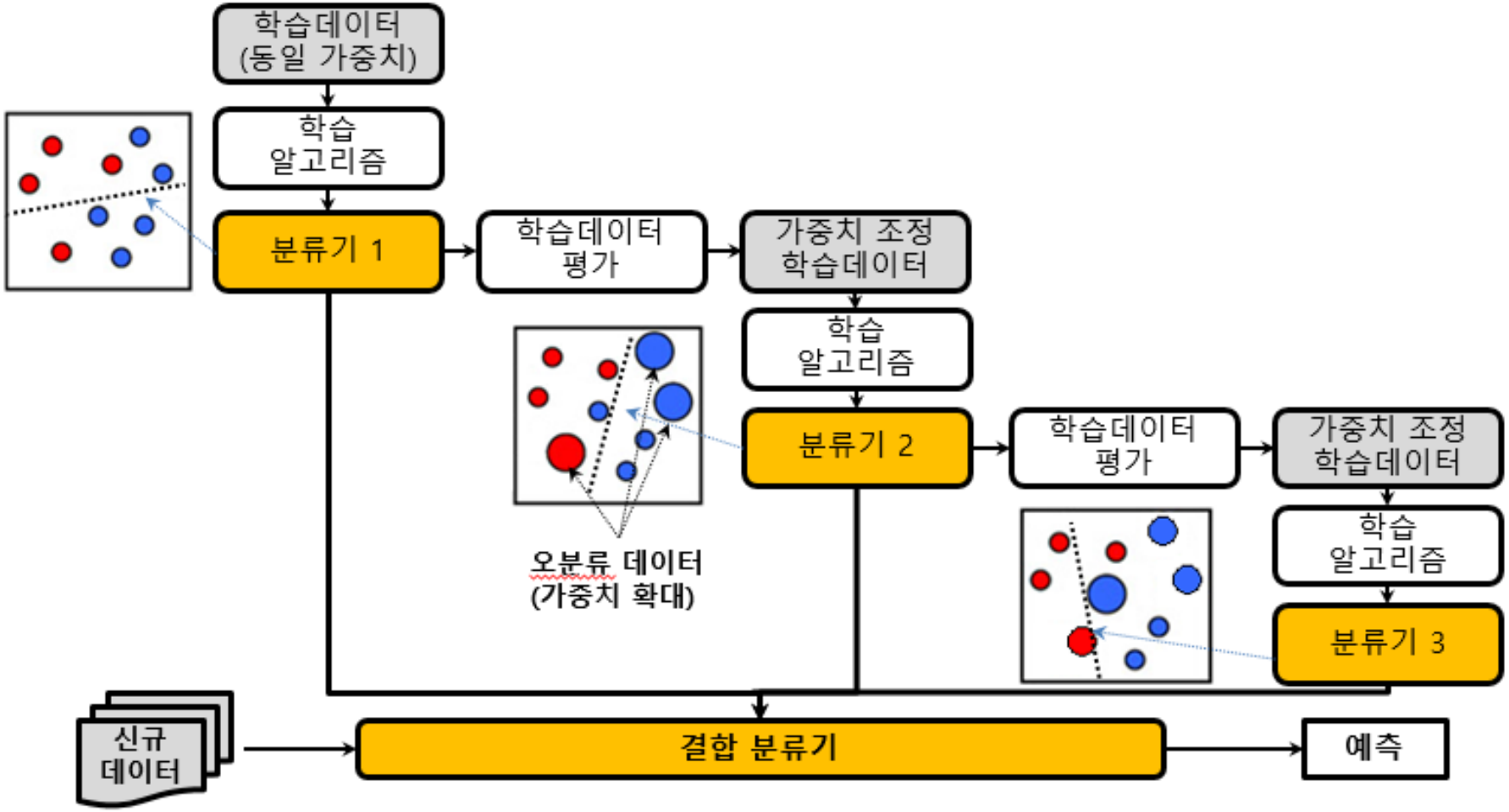
랜덤 포리스트(random forest) 알고리즘

- 분류기로 **결정트리**를 사용하는 배깅 기법



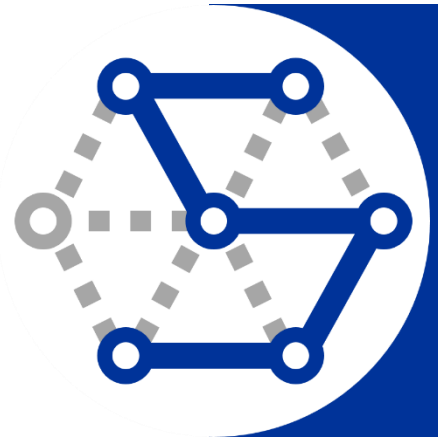
부스팅(boosting)

- k개의 분류기를 순차적으로 만들어 가는 앙상블 분류기 생성 방법
- 분류 정확도에 따라 학습 데이터에 가중치를 변경해가면서 분류기 생성



에이다부스트(AdaBoost)

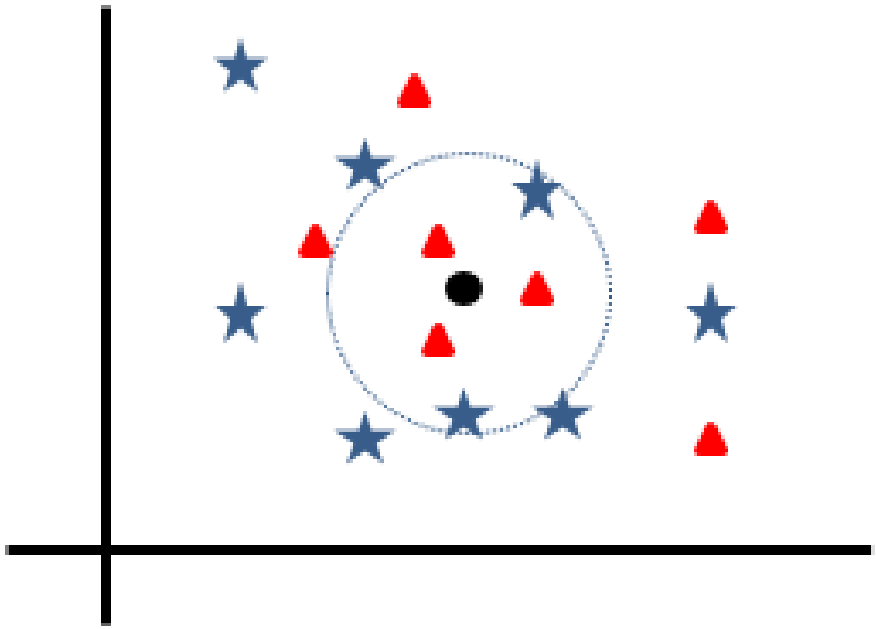
- N개의 학습 데이터 d_i에 대한 **초기 가중치** w_i
  - $w_i = \frac{1}{N}$, 가중치의 합 : 1
- **학습 오류값 ϵ**
  - 잘못 분류한 학습데이터의 가중치의 합으로 표현
  - 값이 0.5미만인 분류기들만을 사용
- **학습**
  - 오류값이 0.5미만인 분류기가 학습되는 경우
  - 분류기 신뢰도 : α
    - $\alpha = 0.5 \ln(\frac{1-\epsilon}{\epsilon})$
  - 잘못 판정한 학습 데이터의 가중치는 증대
    - $w_i \leftarrow w_i e^{\alpha}$
  - 제대로 판정한 학습 데이터의 가중치는 축소
    - $w_i \leftarrow w_i e^{-\alpha}$
  - 가중치의 합이 1이 되도록 정규화



7 최근접 이웃

k-근접이웃 (k-nearest neighbor, KNN) 알고리즘

- (입력, 결과)가 있는 데이터들이 주어진 상황에서, 새로운 입력에 대한 결과를 추정할 때, 결과를 아는 최근접한 k개의 데이터에 대한 결과정보를 이용하는 방법
- 질의(query)와 데이터간의 거리 계산
- 효율적으로 근접이웃 탐색
- 근접 이웃 k개로 부터 결과를 추정

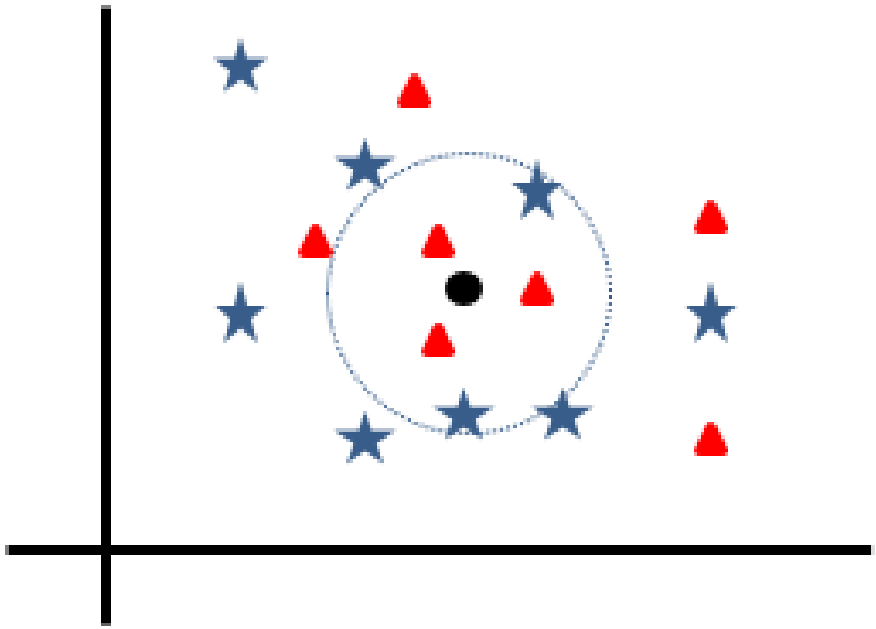


k-근접이웃 (k-nearest neighbor, KNN) 알고리즘

- 데이터간의 거리 계산
- 수치 데이터의 경우
  - 유클리디언 거리(Euclidian distance)

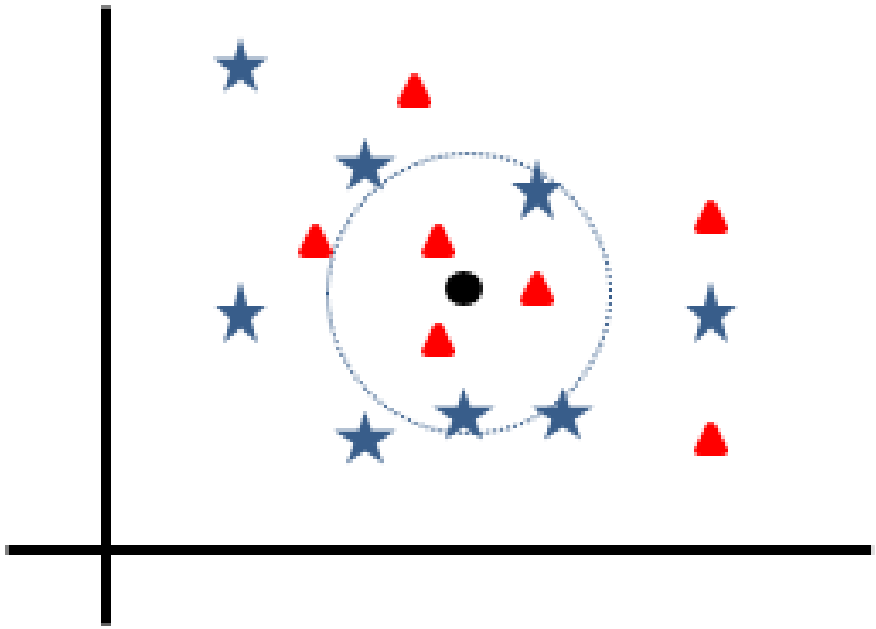
$$X = (x_1, x_2, \ldots, x_n) \quad Y = (y_1, y_2, \ldots, y_n)$$
$$d = (x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2$$

- 범주형 데이터가 포함된 경우
  - 응용분야의 특성에 맞춰 개발



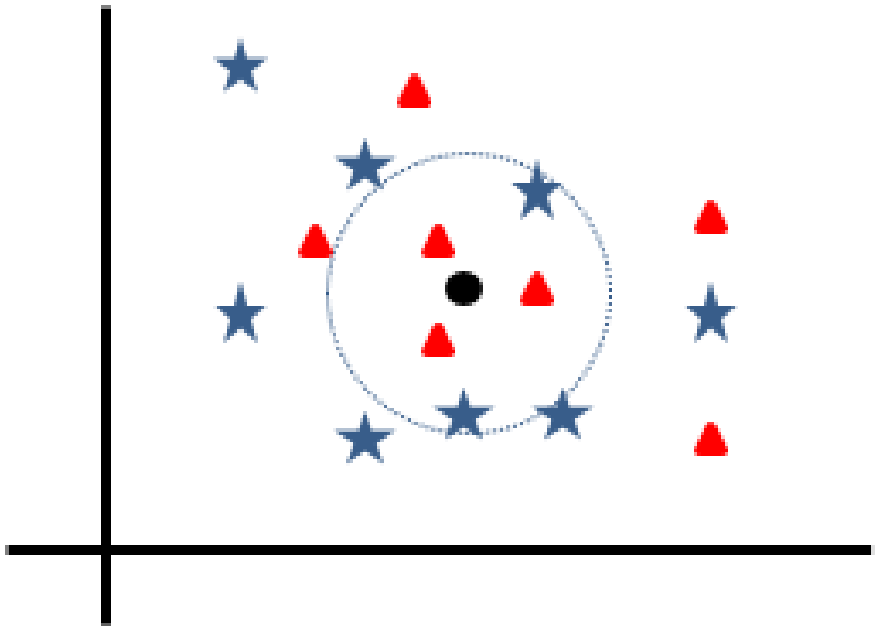
k-근접이웃 (k-nearest neighbor, KNN) 알고리즘

- 효율적인 근접 이웃 탐색
- 데이터의 개수가 많아지면 계산시간 증가 문제
- 색인(indexing) 자료구조 사용
  - R-트리, k-d 트리 등



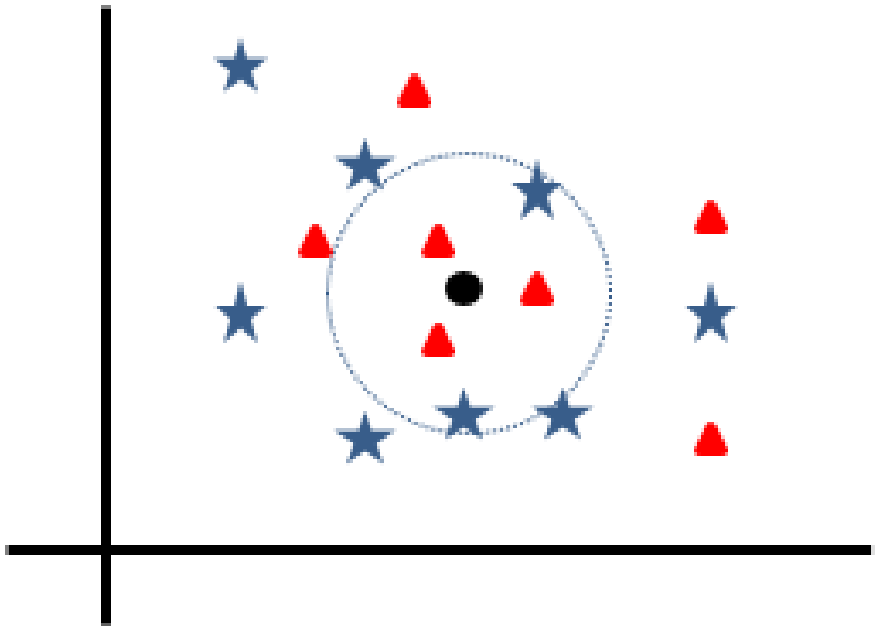
k-근접이웃 (k-nearest neighbor, KNN) 알고리즘

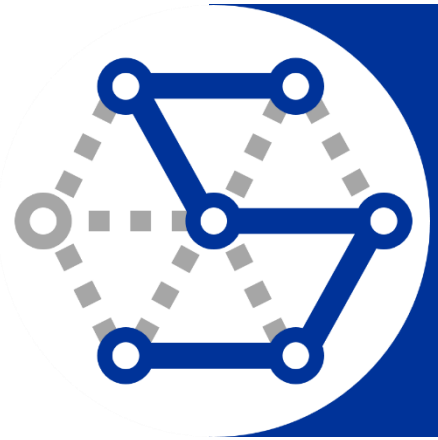
- 최근접 k개로 부터 결과를 추정하는 방법
- 분류
  - 출력이 범주형 값
  - 다수결 투표(majority voting) : 개수가 많은 범주 선택
- 회귀분석
  - 출력이 수치형 값
  - 평균 : 최근접 k개의 평균값
  - 가중합(weighted sum) : 거리에 반비례하는 가중치 사용



k-근접이웃 (k-nearest neighbor, KNN) 알고리즘

- 학습단계에서는 실질적인 학습이 일어나지 않고 데이터만 저장
  - 학습데이터가 크면 메모리 문제
  - 게으른 학습(lazy learning)
- 새로운 데이터가 주어지면 저장된 데이터를 이용하여 학습
  - 시간이 많이 걸릴 수 있음



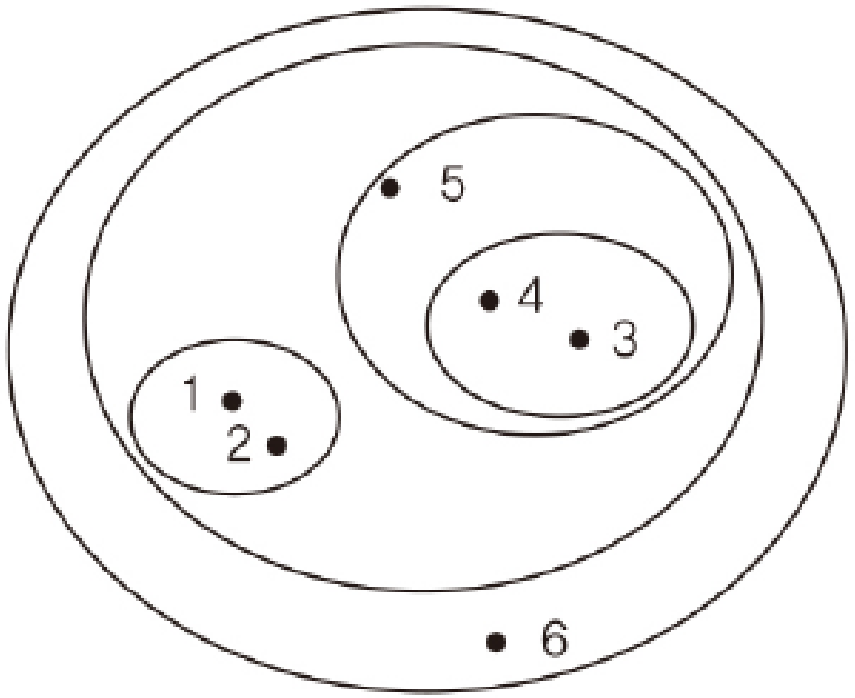


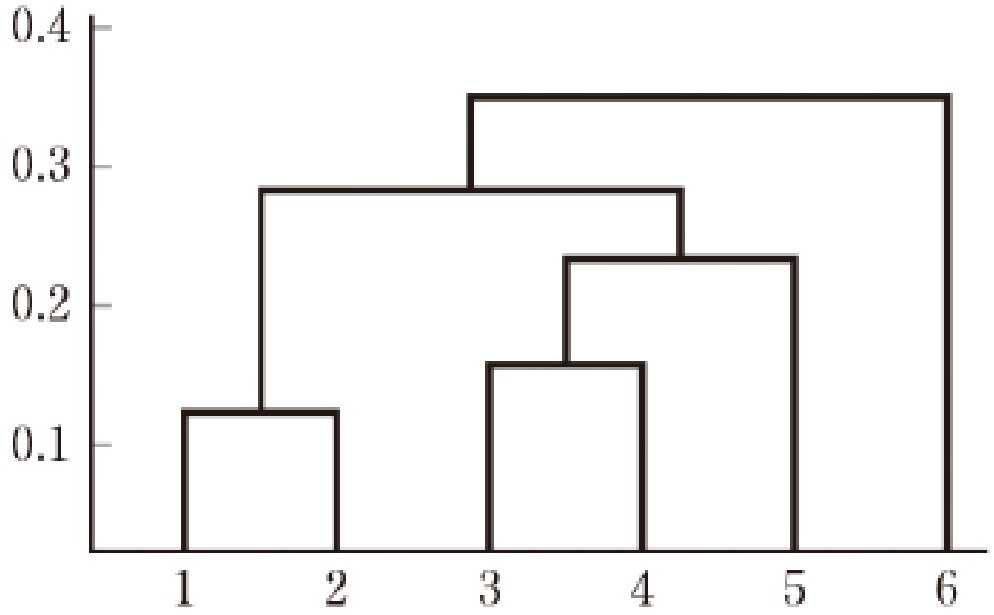
8 군집화

군집화 알고리즘

- 군집화(clustering) 알고리즘
  - 데이터를 유사한 것들끼리 모우는 것
  - 군집 간의 유사도(similarity)는 크게, 군집 내의 유사도는 작게
- 계층적 군집화 (hierarchical clustering)
  - 군집화의 결과가 군집들이 계층적인 구조를 갖도록 하는 것
  - 병합형(agglomerative) 계층적 군집화: 각 데이터가 하나의 군집을 구성하는 상태에서 시작하여, 가까이에 있는 군집들을 결합하는 과정을 반복하여 계층적인 군집 형성
  - 분리형(divisive) 계층적 군집화: 모든 데이터를 포함한 군집에서 시작하여 유사성을 바탕으로 군집을 분리하여 점차 계층적인 구조를 갖도록 구성
- 분할 군집화 (partitioning clustering)
  - 계층적 구조를 만들지 않고 전체 데이터를 유사한 것들끼리 나누어서 묶는 것
  - 예. k-means 알고리즘

계층적 군집화와 덴드로그램(dendrogram)



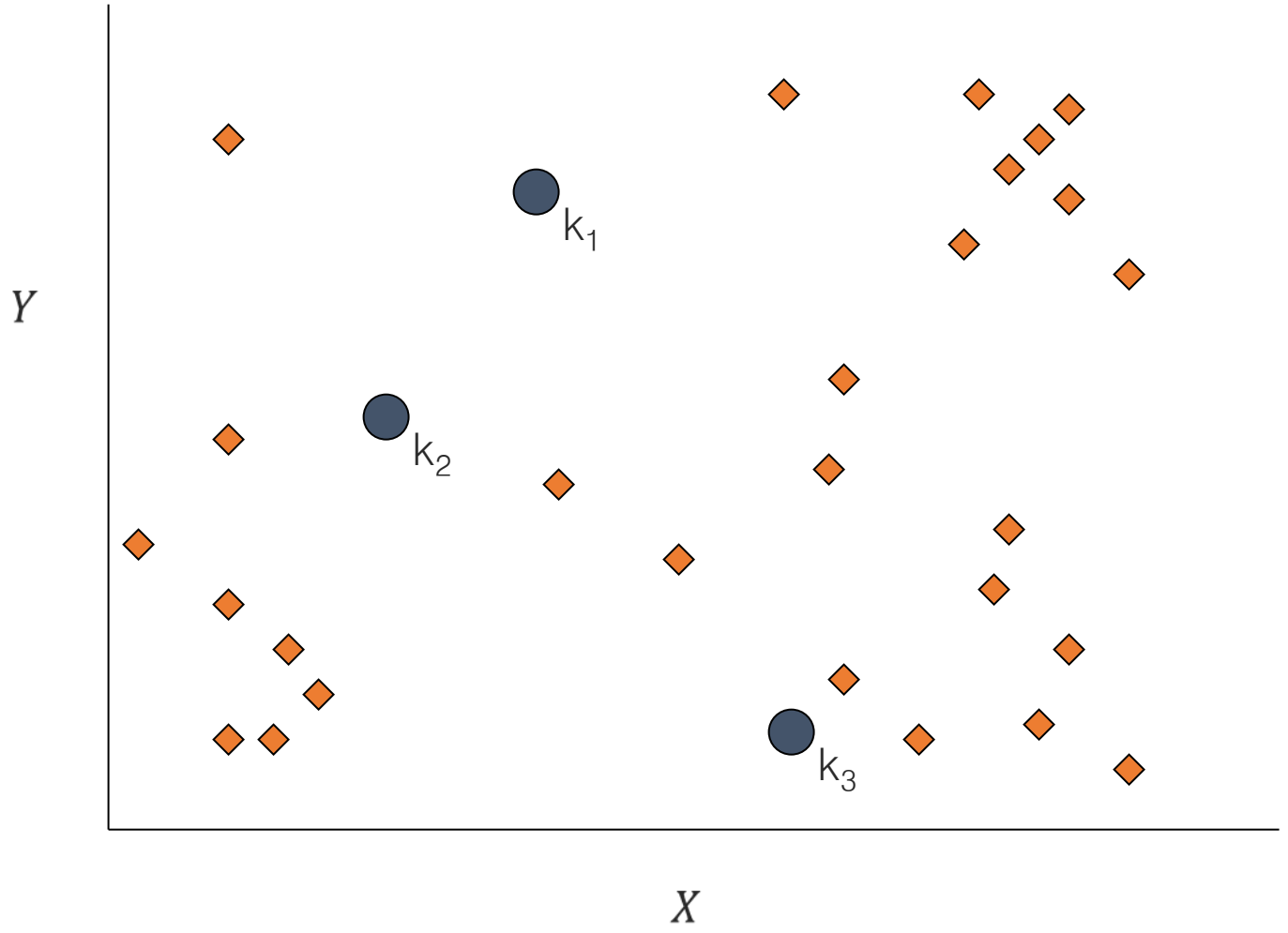


k-means 알고리즘

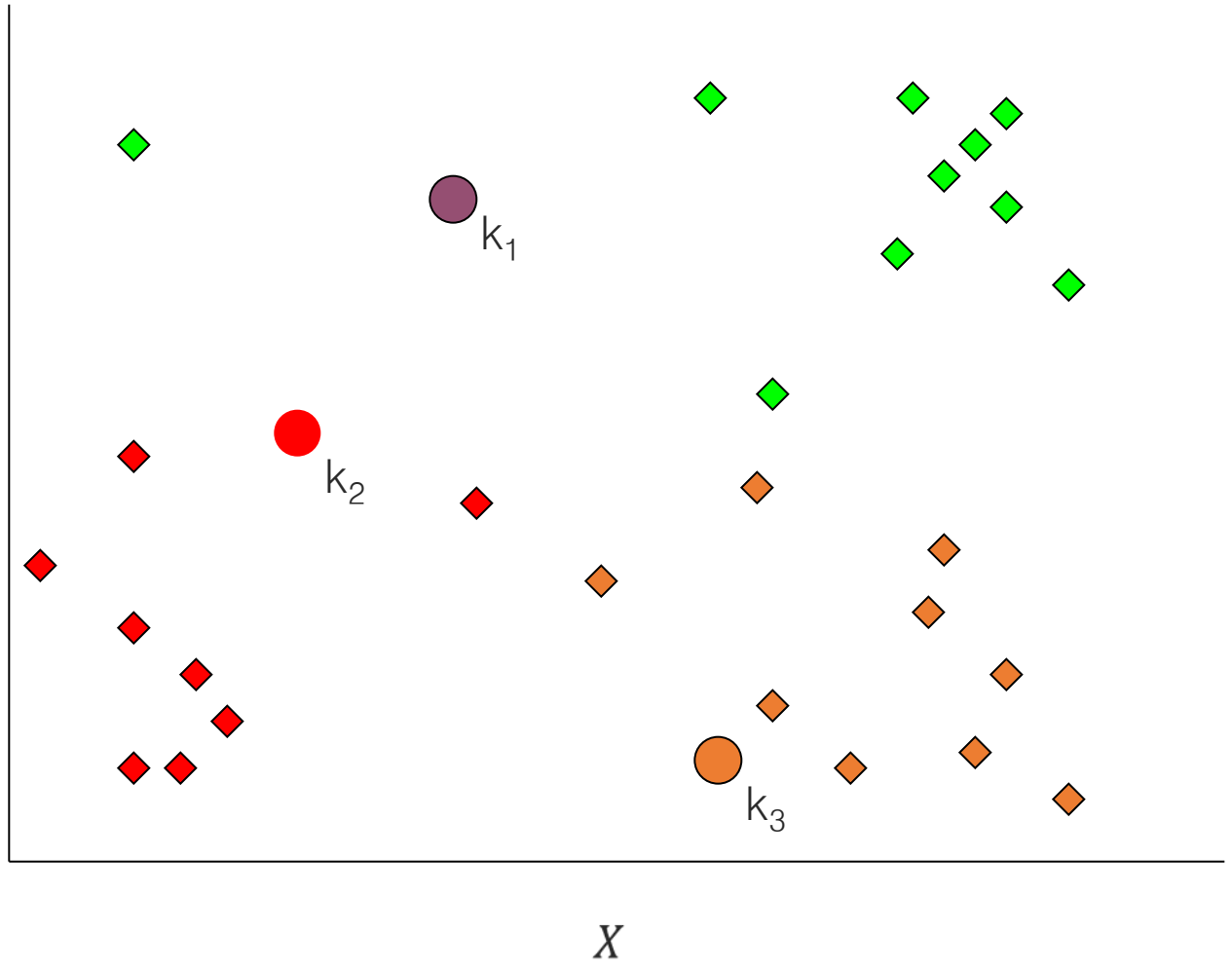
- 군집화 알고리즘
- 군집화 과정
  - 군집의 중심 위치 선정
  - 군집 중심을 기준으로 군집 재구성
  - 군집별 평균 위치 결정
  - 군집 평균 위치로 군집 중심 조정
  - 수렴할 때까지 2-4 과정 반복

k-means 실행과정

무작위로
군집 중심위치
3개를 선택

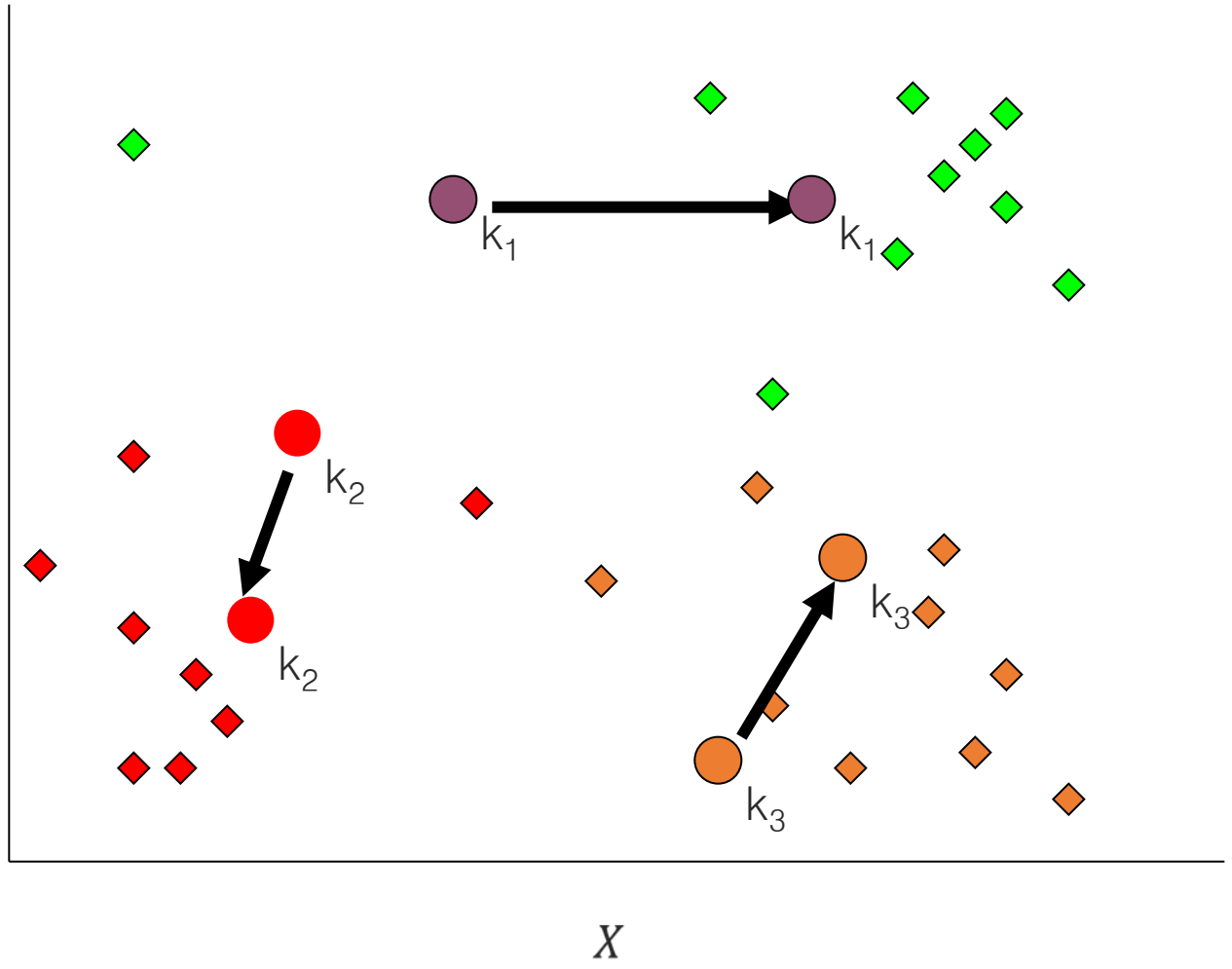


k-means 실행과정



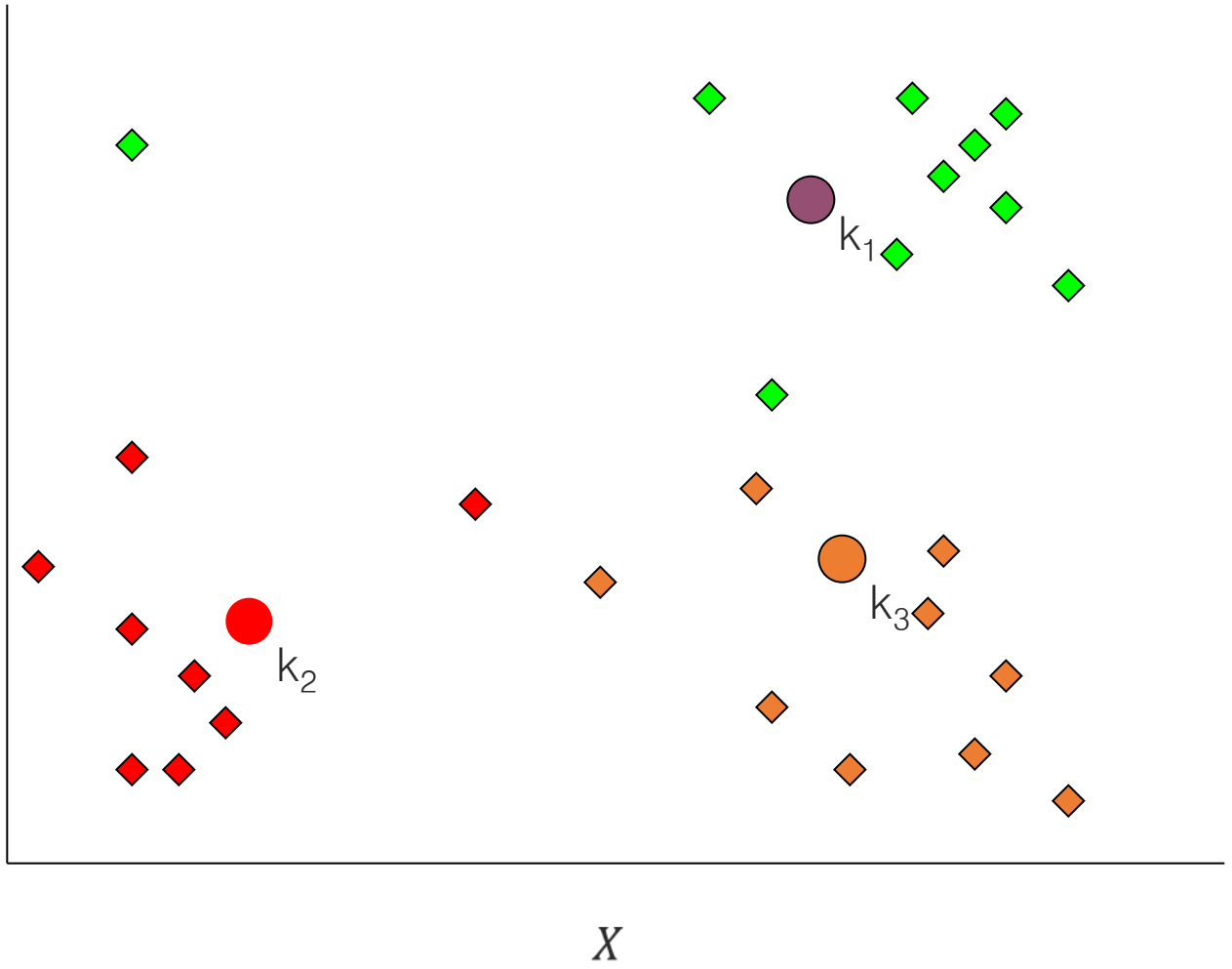
k-means 실행과정

군집 중심 위치를 군집의 평균 위치로 이동

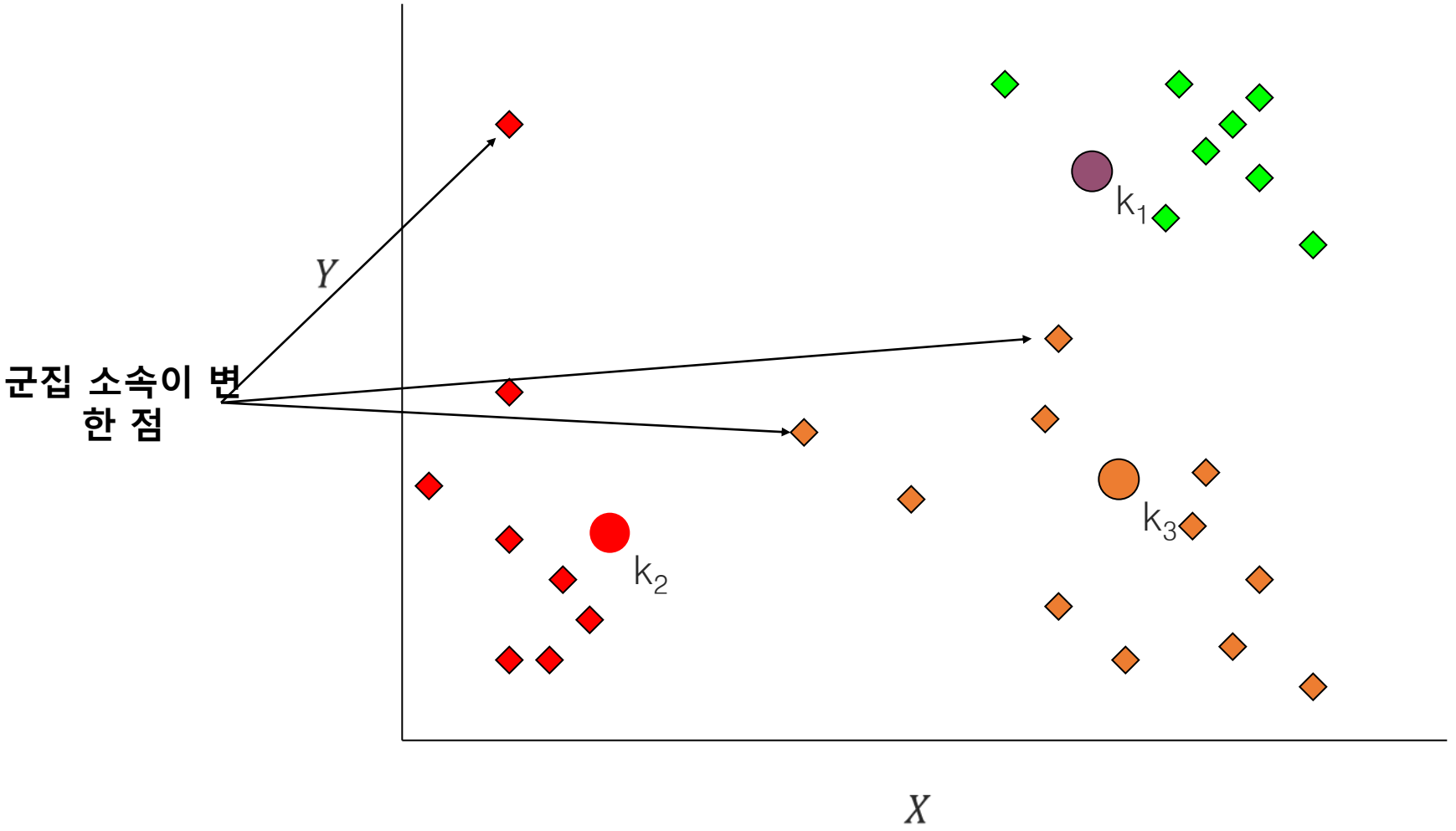


k-means 실행과정

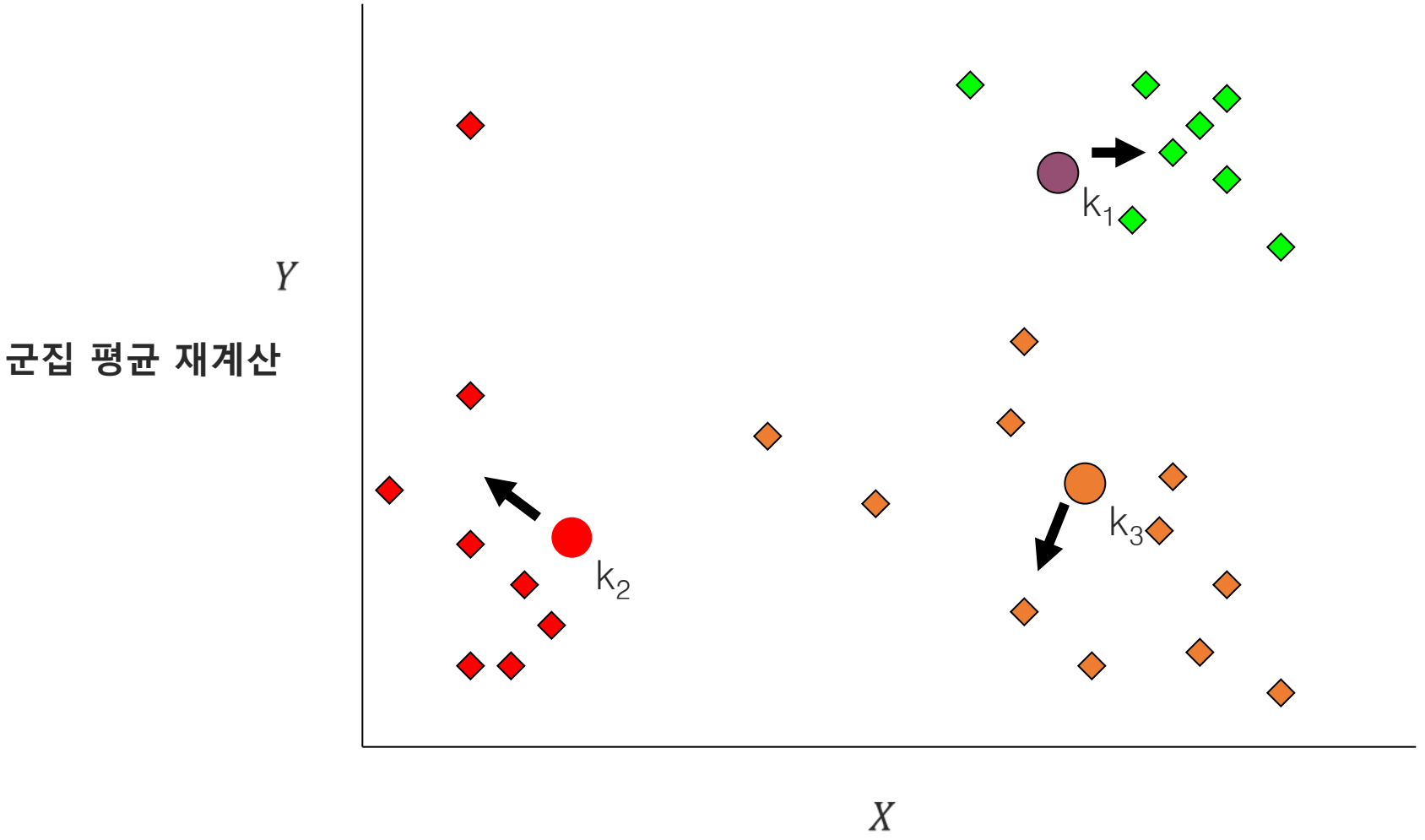
새로운 군집 중심을 기준으로 각 점의 소속을 재할당



k-means 실행과정

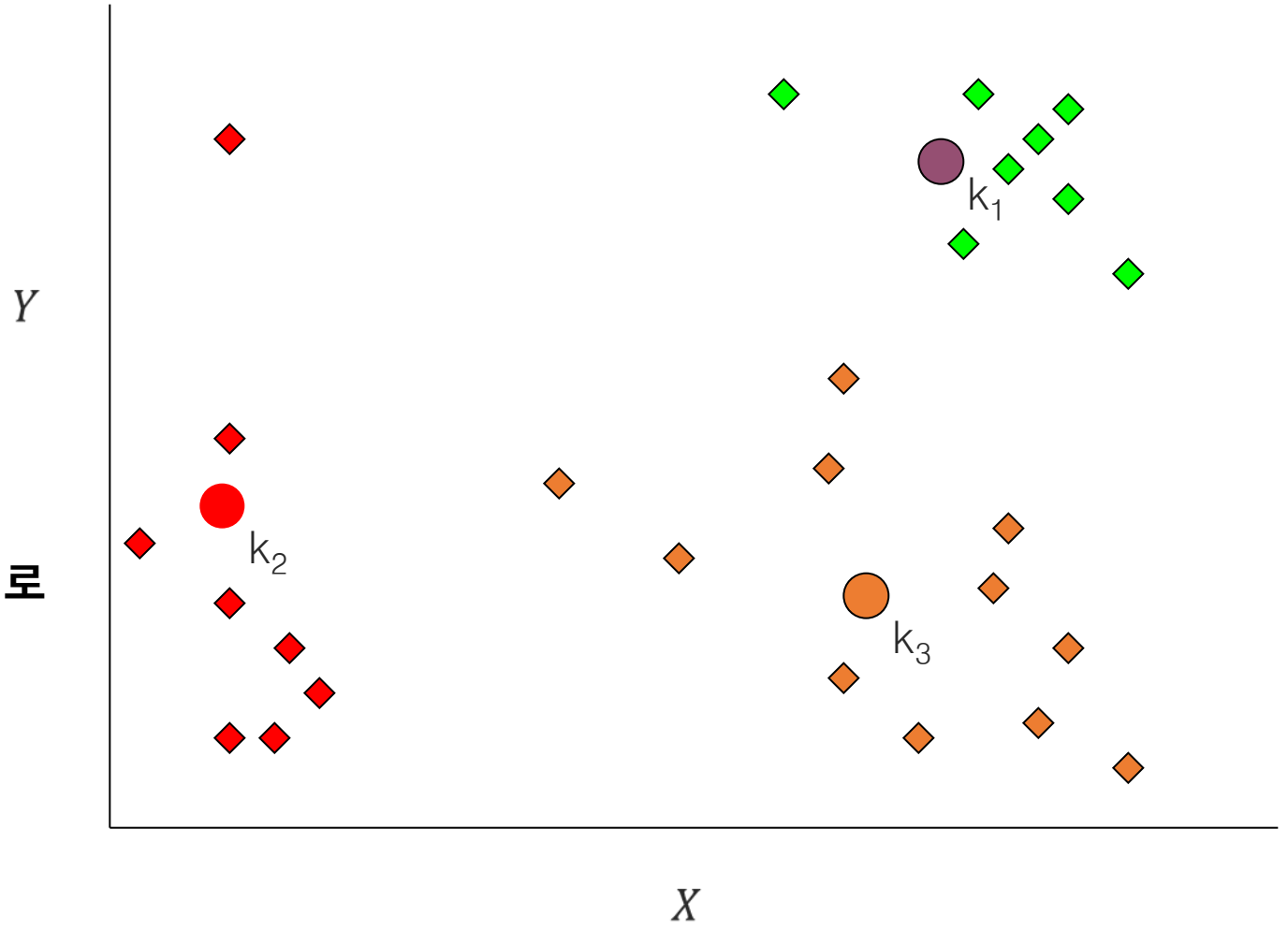


k-means 실행과정



k-means 실행과정

군집 중심을
군집 평균위치로
변경



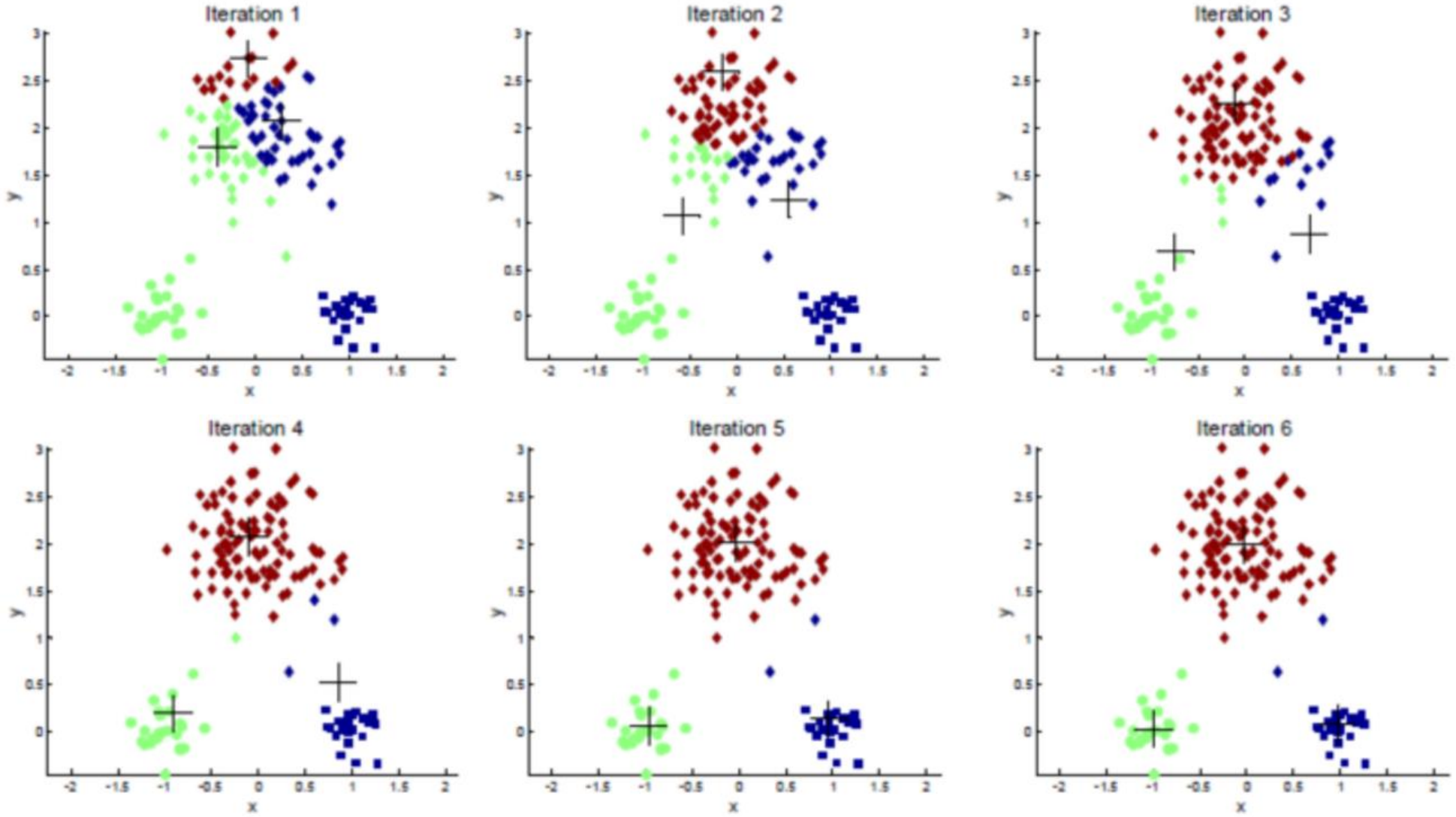
k-means 알고리즘

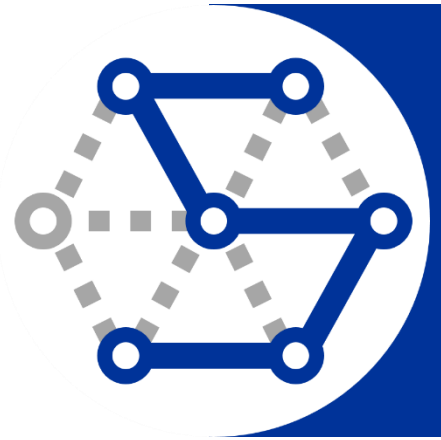
- i 번째 클러스터의 **중심**을 μ_i, 클러스터에 속하는 **점의 집합** S_i을 라고 할 때, **전체 분산**

$$V = \sum_{i=1}^{k} \sum_{j \in S_i} |x_j - \mu_i|^2$$

- **분산값 V**을 **최소화**하는 S_i를 찾는 것이 알고리즘의 목표
- **과정**
  1. 우선 초기의 μ_i를 임의로 설정
  2. 다음 두 단계를 클러스터가 변하지 않을 때까지 반복
     I. **클러스터 설정**: 각 점에 대해, 그 점에서 가장 가까운 클러스터를 찾아 배당한다.
     II. **클러스터 중심 재조정**: μ_i를 각 클러스터에 있는 점들의 평균값으로 재설정해준다.
- **특성**
  - **군집의 개수** k는 **미리 지정**
  - **초기 군집 위치**에 **민감**

초기 중심값에 대해 민감한 군집화 결과





9 기타 머신러닝 방법

나이브 베이즈 분류기(Naïve Bayes Classifier)

- 부류(class) 결정 지식을 조건부 확률(conditional probability)로 결정
  - $P(c|x_1, x_2, \ldots, x_n)$: 속성값에 대한 부류의 조건부 확률
  - c : 부류
  - x_i: 속성값
- 베이즈 정리(Bayes theorem)

가능도 사전확률

$$P(c|x_1, x_2, \ldots, x_n) = \frac{P(x_1, x_2, \ldots, x_n|c)P(c)}{P(x_1, x_2, \ldots, x_n)}$$

사후확률 증거

- 가능도(likelihood)의 조건부 독립(conditional independence) 가정
  - $P(x_1, x_2, \ldots, x_n|c) = P(x_1|c)P(x_2|c) \ldots P(x_n|c)$
  - $P(c|x_1, x_2, \ldots, x_n) = \frac{P(x_1|c)P(x_2|c)\ldots P(x_n|c)P(c)}{P(x_1, x_2, \ldots, x_n)}$

나이브 베이즈 분류기 예제

	속성			부류
	Pattern	*Outline*	*Dot*	*Shape*
1	수직	점선	무	삼각형
2	수직	점선	유	삼각형
3	대각선	점선	무	사각형
4	수평	점선	무	사각형
5	수평	실선	무	사각형
6	수평	실선	유	삼각형
7	수직	실선	무	사각형
8	수직	점선	무	삼각형
9	대각선	실선	유	사각형
10	수평	실선	무	사각형
11	수직	실선	유	사각형
12	대각선	점선	유	사각형
13	대각선	실선	무	사각형
14	수평	점선	유	삼각형

$$P(\text{삼각형}) = \frac{5}{14} \qquad P(\text{사각형}) = \frac{9}{14}$$

$$P(\text{수직}|\text{삼각형}) = \frac{3}{5}$$

$$P(\text{수평}|\text{삼각형}) = \frac{2}{5}$$

$$P(\text{대각선}|\text{삼각형}) = \frac{0}{5}$$

$$P(\text{점선}|\text{삼각형}) = \frac{4}{5}$$

$$P(\text{실선}|\text{삼각형}) = \frac{1}{5}$$

$$P(\text{유}|\text{삼각형}) = \frac{3}{5}$$

$$P(\text{무}|\text{삼각형}) = \frac{2}{5}$$

$$P(\text{수직}, \text{점선}, \text{무}) = \frac{2}{14}$$

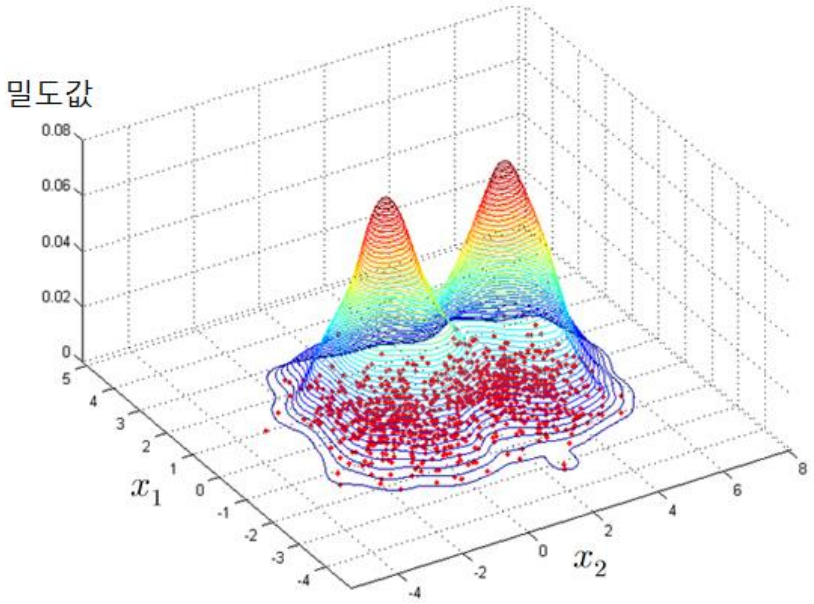
$$P(\text{삼각형}|\text{수직}, \text{점선}, \text{무}) = \frac{P(\text{수직}|\text{삼각형})\,P(\text{점선}|\text{삼각형})P(\text{무}|\text{삼각형})P(\text{삼각형})}{P(\text{수직},\text{점선},\text{무})} = \frac{3/5 \cdot 4/5 \cdot 2/5 \cdot 5/14}{2/14} = 0.48$$

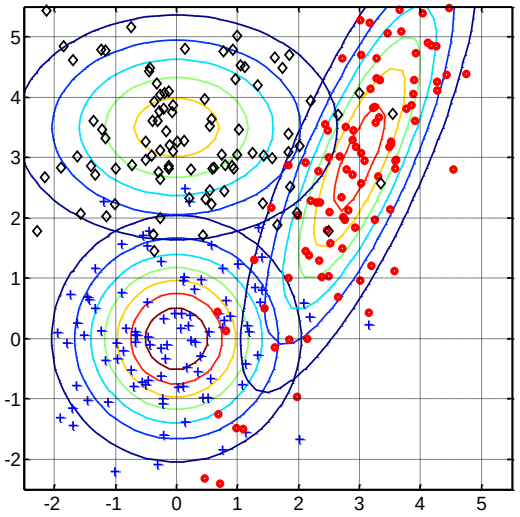
비지도학습(unsupervised learning)

- 결과정보가 없는 데이터들에 대해서 특정 패턴을 찾는 것
  - 데이터에 잠재한 구조(structure), 계층구조(hierarchy) 를 찾아내는 것
  - 숨겨진 사용자 집단(hidden user group)을 찾는 것
  - 문서들을 주제에 따라 구조화하는 것
  - 로그(log) 정보를 사용하여 사용패턴(usage pattern)을 찾아내는 것
- 비지도 학습의 대상
  - 군집화(clustering)
  - 밀도추정(density estimation)
  - 차원축소(dimensionality reduction)

밀도 추정(density estimation)

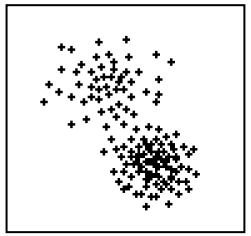
- 분류(class)별 데이터를 만들어 냈을 것으로 추정되는 확률분포을 찾는 것
- 용도
  - 각 부류 별로 주어진 데이터를 발생시키는 확률 계산
  - 가장 확률이 높은 부류로 분류

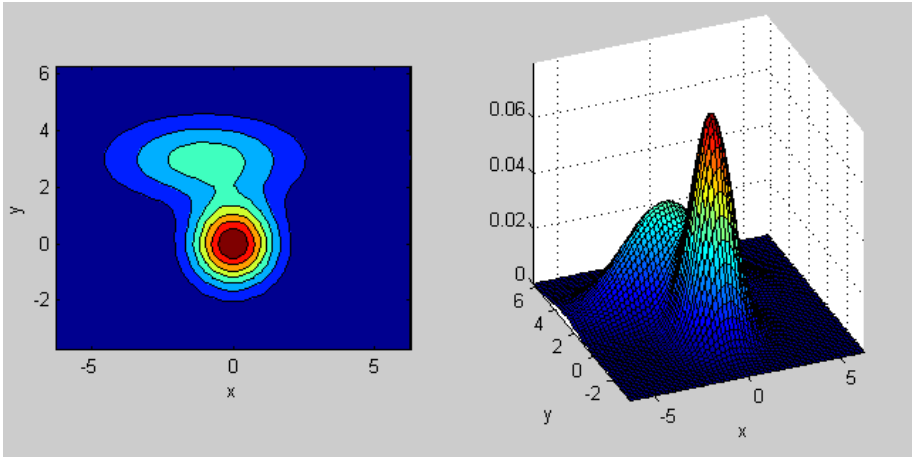




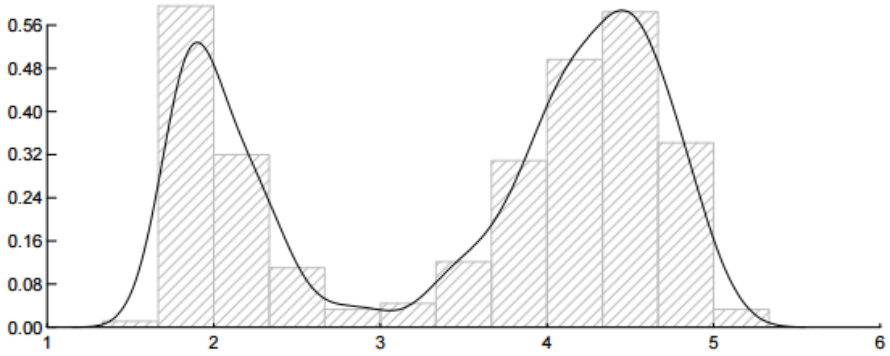
밀도 추정(density estimation)

- 모수적(parametric) 밀도 추정
  - 분포가 특정 수학적 함수의 형태를 가지고 있다고 가정
  - 주어진 데이터를 가장 잘 반영하도록 함수의 파라미터 결정
  - 전형적인 형태 : 가우시안(Gaussian) 함수 또는 여러 개의 가우시안 함수의 혼합(Mixture of Gaussian)
- 비모수적(nonparametric) 밀도 추정
  - 분포에 대한 특정 함수를 가정하지 않고, 주어진 데이터를 사용하여 밀도함수의 형태 표현
  - 전형적인 형태 : 히스토그램(histogram)



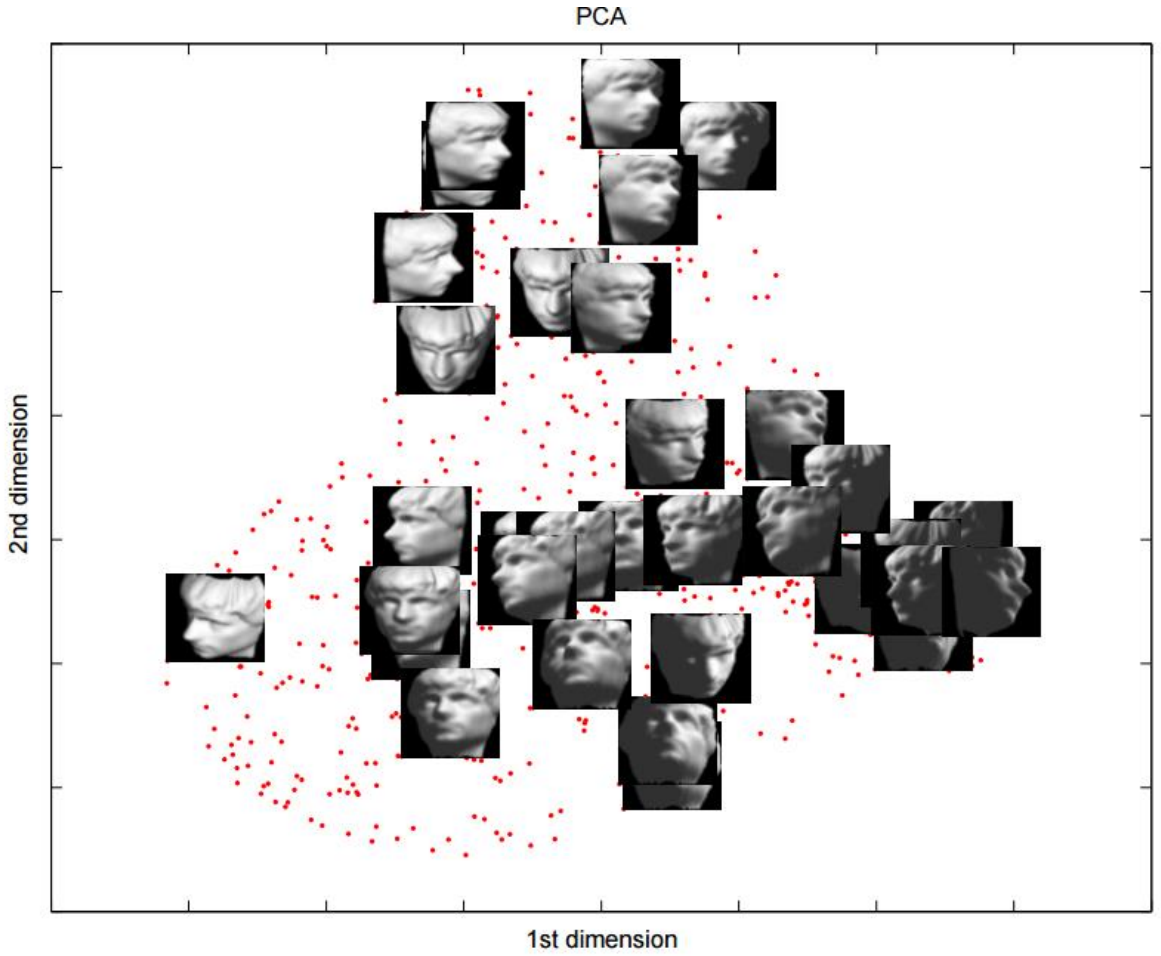


http://i.stack.imgur.com/pE0Xu.gif



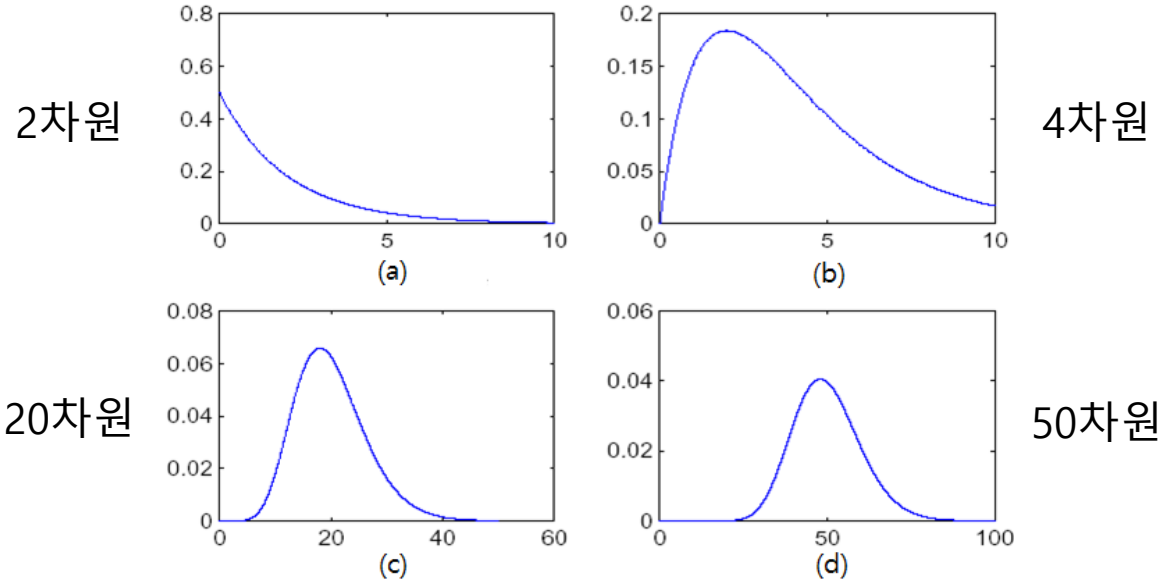
차원축소(dimension reduction)

- 고차원의 데이터를 정보의 손실을 최소화하면서 저차원으로 변환하는 것
- 목적
  - 2,3차원으로 변환해 시각화하면 직관적 데이터 분석 가능
  - 차원의 저주(curse of dimensionality) 문제 완화

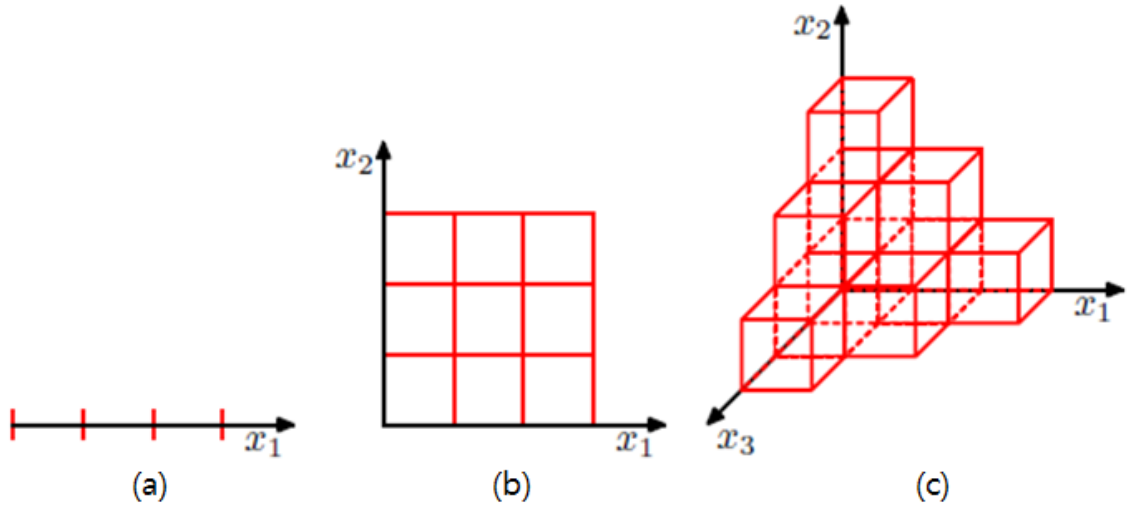


차원의 저주(curse of dimensionality)

- 차원이 커질수록 거리분포가 일정해지는 경향

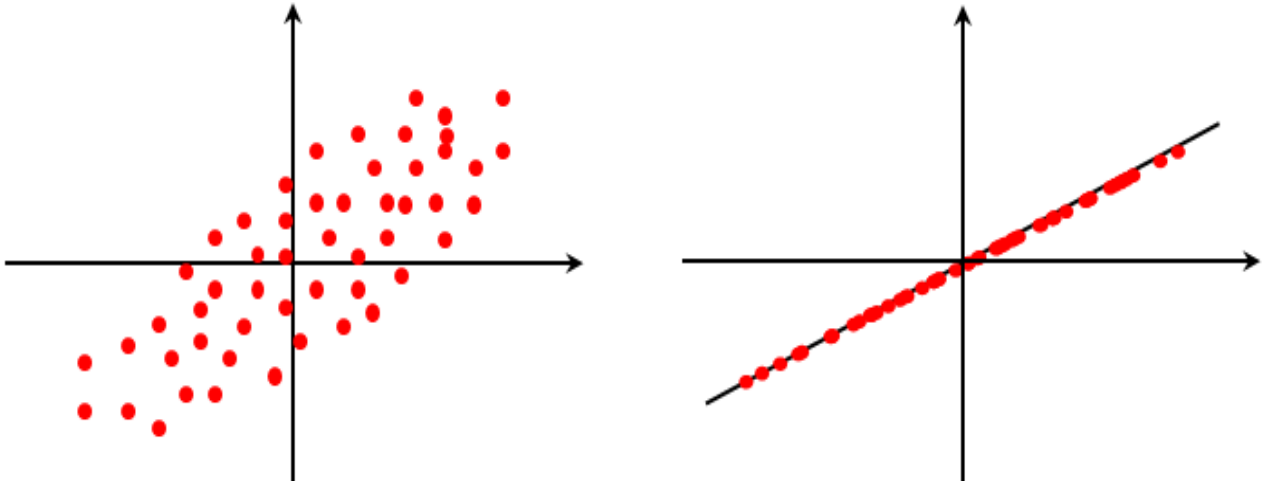


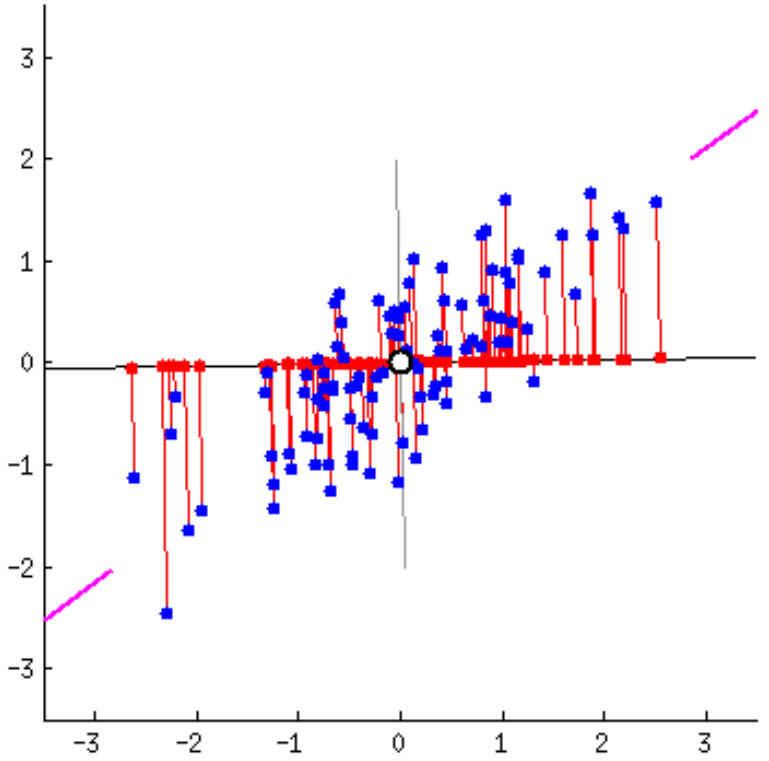
- 차원이 증가함에 따라 부분공간의 개수가 기하급수적으로 증가



주성분 분석 (Principal Component Analysis, PCA)

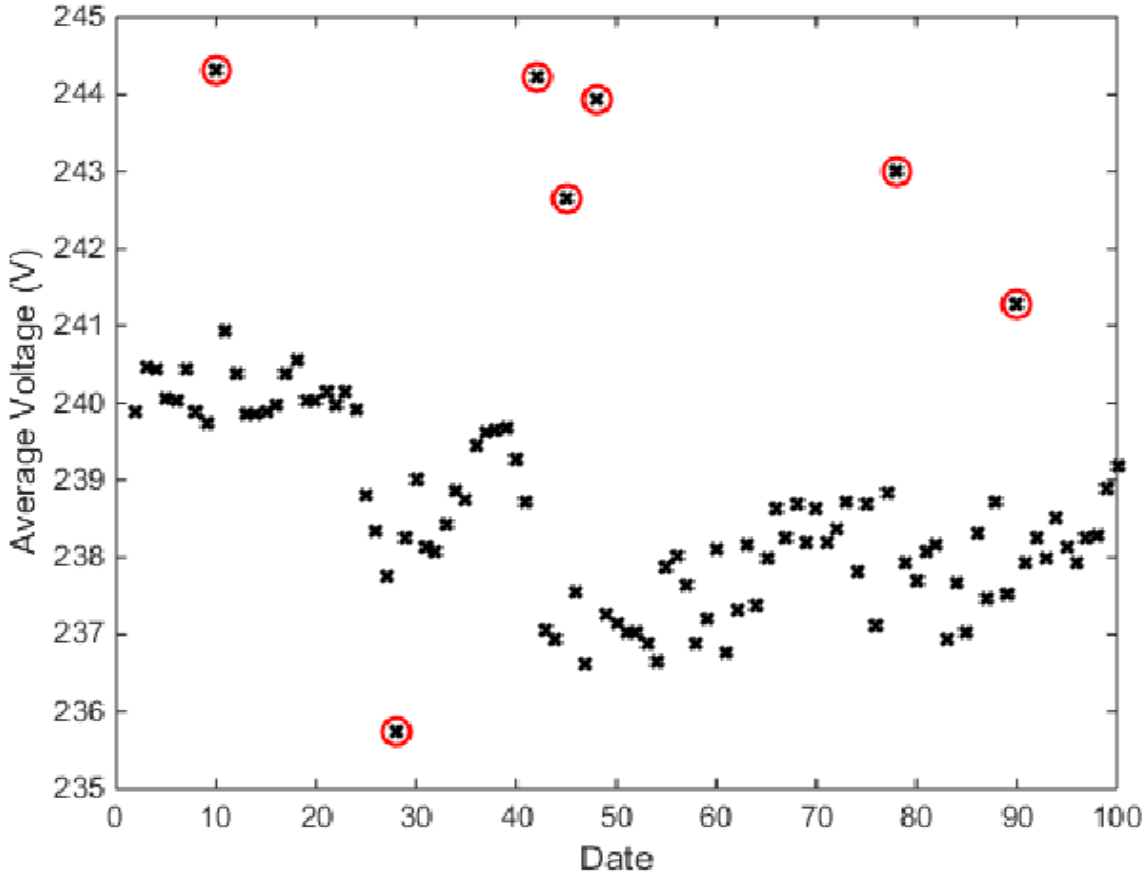
- 분산이 큰 소수의 축들을 기준으로 데이터를 사상(projection)하여 저차원으로 변환
- 데이터의 공분산행렬(covariance matrix)에 대한 고유값(eigenvalue)가 큰 소수의 고유벡터(eigenvector)를 사상 축으로 선택





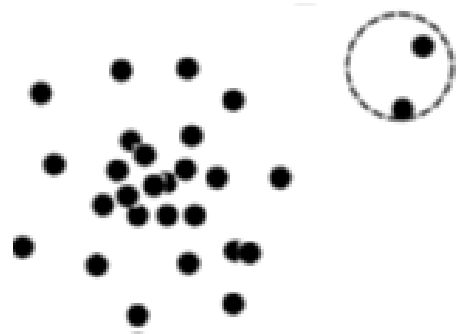
이상치(outlier) 탐지

- 이상치
  - 다른 데이터와 크게 달라서 다른 메커니즘에 의해 생성된 것이 아닌지 의심스러운 데이터
  - 관심 대상
- 잡음(noise)
  - 관측 오류, 시스템에서 발생하는 무작위적인 오차
  - 관심이 없는 제거할 대상
- 신규성 탐지(novelty detection)와 관련

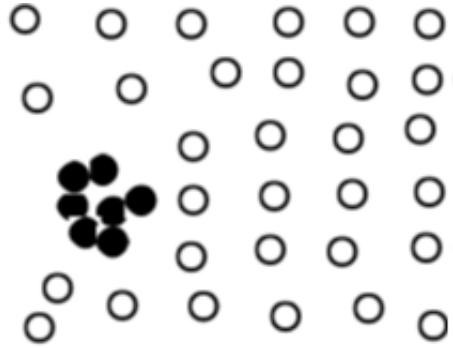


이상치(outlier) 탐지

- 점 이상치(point outlier)
  - 다른 데이터와 비교하여 차이가 큰 데이터



- 상황적 이상치(contextual outlier)
  - 상황에 맞지 않는 데이터
  - 예) 여름철에 25도인 데이터는 정상, 겨울철에 25도는 이상치



- 집단적 이상치(collective outlier)
  - 여러 데이터를 모아서 보면 비정상으로 보이는 데이터들의 집단

... http-web, buffer-overflow, http-web, http-web, smtp-mail, ftp, http-web, ssh, smtp-mail, http-web, **ssh, buffer-overflow, ftp,** http-web, ftp, smtp-mail. http-web ...

이상치(outlier) 탐지

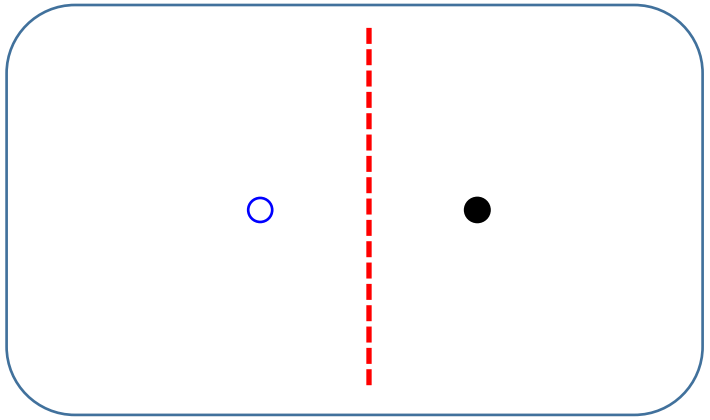
- 부정사용감지 시스템(fraud detection system, FDS)
  - 이상한 거래 승인 요청 시에 카드 소유자에게 자동으로 경고 메시지 전송

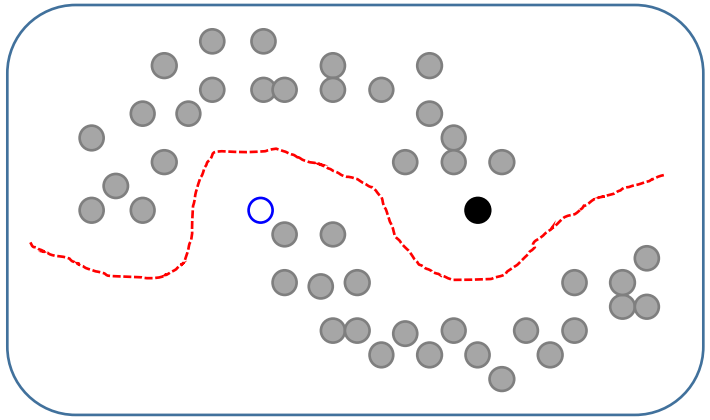
- 침입탐지 시스템(intrusion detection system, IDS)
  - 네트워크 트래픽을 관찰하여 이상 접근 식별

- 시스템의 고장 진단
- 임상에서 질환 진단 및 모니터링
- 공공보건에서 유행병의 탐지
- 스포츠 통계학에서 특이 사건 감지
- 관측 오류의 감지

반지도 학습(semi-supervised learning)

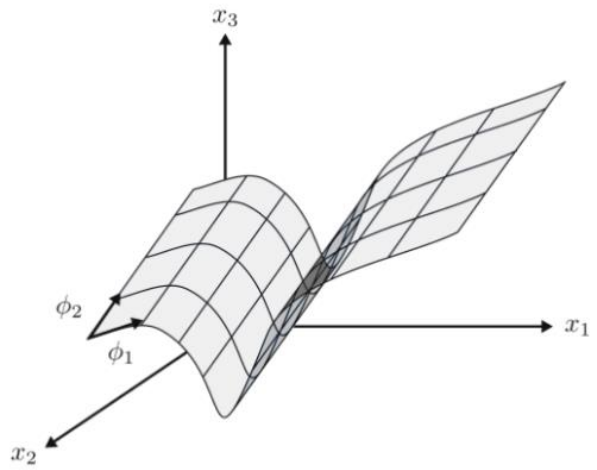
- 입력에 대한 결과값이 없는 미분류 데이터(unlabeled data)를 지도학습에 사용하는 방법
- 분류된 데이터(labeled data)는 높은 획득 비용, 미분류 데이터는 낮은 획득 비용
- 분류 경계가 인접한 미분류 데이터들이 동일한 집단에 소속하도록 학습
- 같은 군집에 속하는 것은 가능한 동일한 부류에 소속하도록 학습

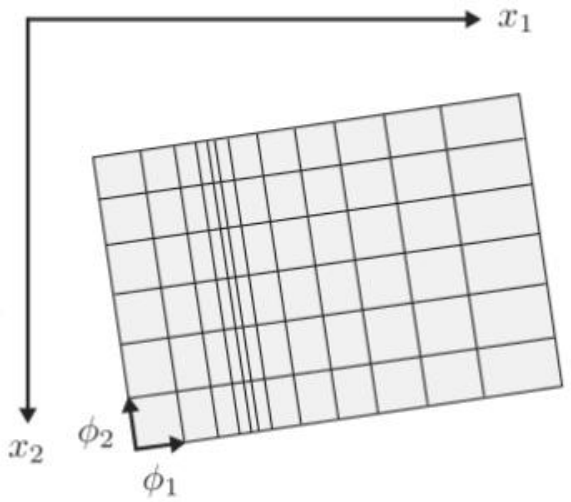




반지도 학습(semi-supervised learning)

- 평활성(smoothness, 平滑性) 가정
  - 가까이 있는 점들은 서로 같은 부류에 속할 가능성이 높음
- 군집(cluster) 가정
  - 같은 군집에 속하는 데이터는 동일한 부류에 속할 가능성이 높음
- 매니폴드(manifold) 가정
  - 원래 차원보다 낮은 차원의 매니폴드에 데이터가 분포할 가능성이 높음






suanlab
이수안 컴퓨터 연구소
suan computer laboratory