머신런닝 개요

Lecture 2 : Issues of ML

College of Information and Electronic Engineering

Kyung Hee Univeristy

Prof. Wonha Kim

(wonha@khu.ac.kr)

Contents

1. Machine Learning Data
   - Features
   - Data Format & Vector space representation
   - Issues of Data
   - Database

2. Machine Learning Model
   - Regression and Training
   - Underfit and Overfit
   - Variance/Resolution
   - Data augment and Regulation
   - Model Verification

3. Classification of Machine Learning
   - Supervised, Unsupervised and Reinforcement learning
   - Online and Offline learning
   - Instance-based and Model based learning

2.1 Machine Learning Data : Feature Space

- Feature Selection :

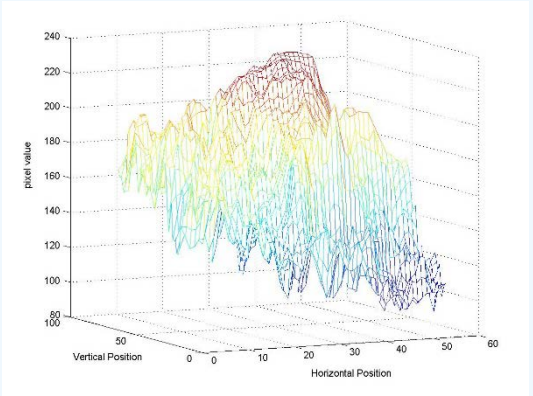
  also known as **variable selection**, **attribute selection** or **variable subset selection**, is the process of selecting a subset of relevant features (variables, predictors) for use in model construction.

- Feature selection techniques are used for several reasons:
  - simplification of models to make them easier to interpret by researchers/users
  - shorter training times
  - to avoid the curse of dimensionality,
  - enhanced generalization by reducing overfitting (formally, reduction of variance)

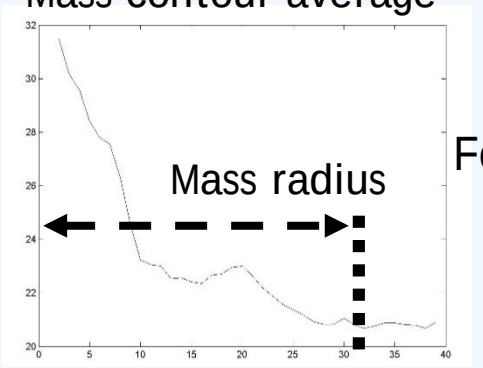
2.1. ML data : Feature Selection -Difficult examples

- **Mass Mammogram (종괴형 유방암)**

Pixel values of mass



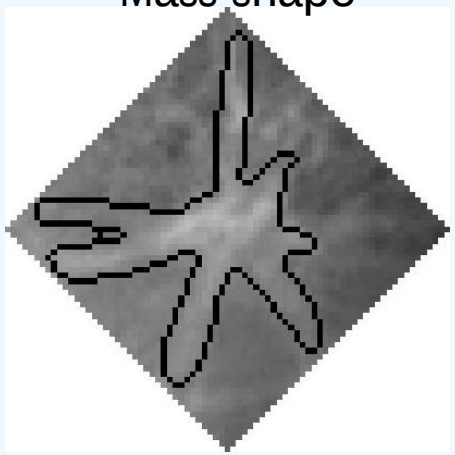
Mass contour average



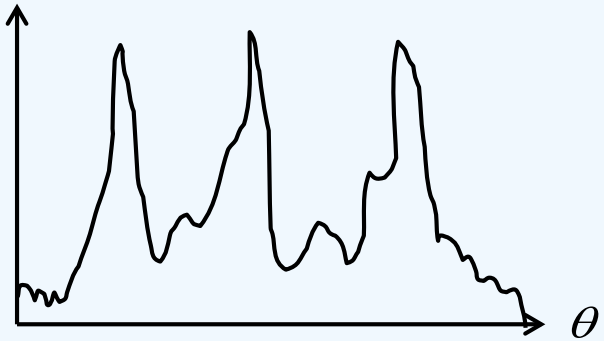
Radius

Feature 1 = the gradient average of contours

Mass shape



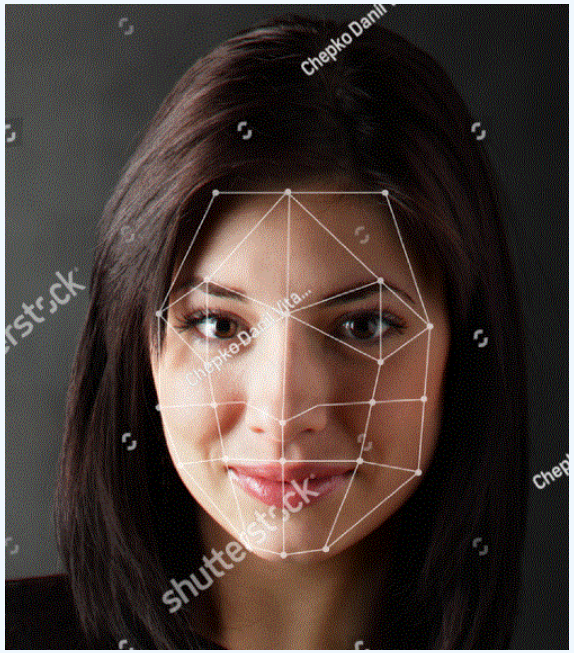
Feature 2 = Length of boundary



- 종괴형 유방암 특징
  - 종양이 빨리 자람에 따라서 중심으로 부터 tissue(조직)의 밀도가 급격히 떨어짐
  - 종양이 빨리 자람에 따라서 별 모양 임.

2.1. ML data : Feature Selection -Difficult examples

- Face Recognition



- Feature 1 : 눈간의 거리
- Feature 2 : 두 눈과 코가 만드는 삼각형의 모양
- Feature 3 : 입과 턱을 구성하는 도형의 모양
- ….

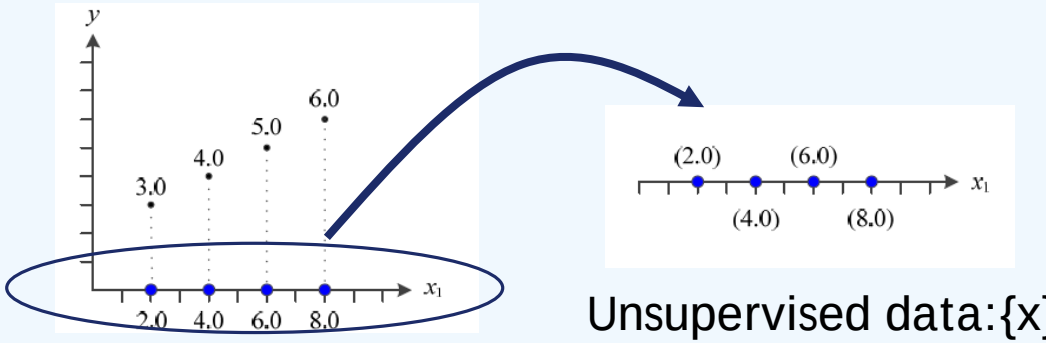
2.1. ML data : Data format

- **Feature data:** $\mathbb{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$, **Corresponding label data :** $\mathbb{Y} = \{y_1, y_2, \cdots, y_n\}$

  where feature data $\mathbf{x}_i$ is a vector and label value y_i is a scalar (value).

- **ML Data Representation :** $\mathbb{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$, $\mathbb{Y} = \{y_1, y_2, \cdots, y_n\}$

  – 1D Feature data



Supervised data:{(x,y)}

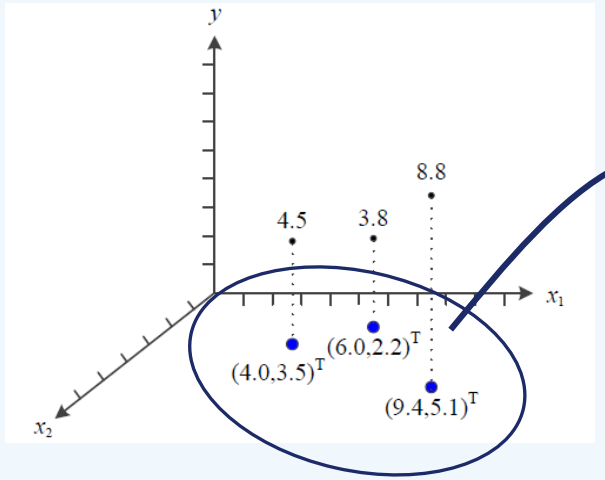
Unsupervised data:{x}

  – 2D Feature data

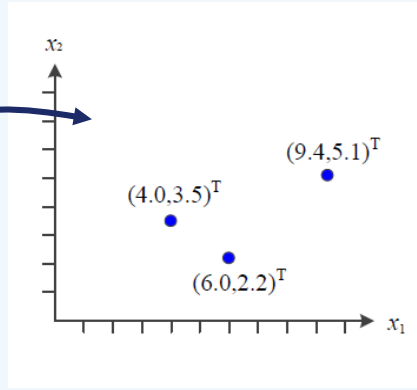
- Feature vector : $\mathbf{x}=(x_1,x_2)^T$
- Examples:

  $\mathbf{x}$=(Weight, Height)^T, y : Driving distance of a ball

  $\mathbf{x}$=(Temperatures, Headache)^T, y : Prob. Of cold



Supervised data:{(**x**,y)}

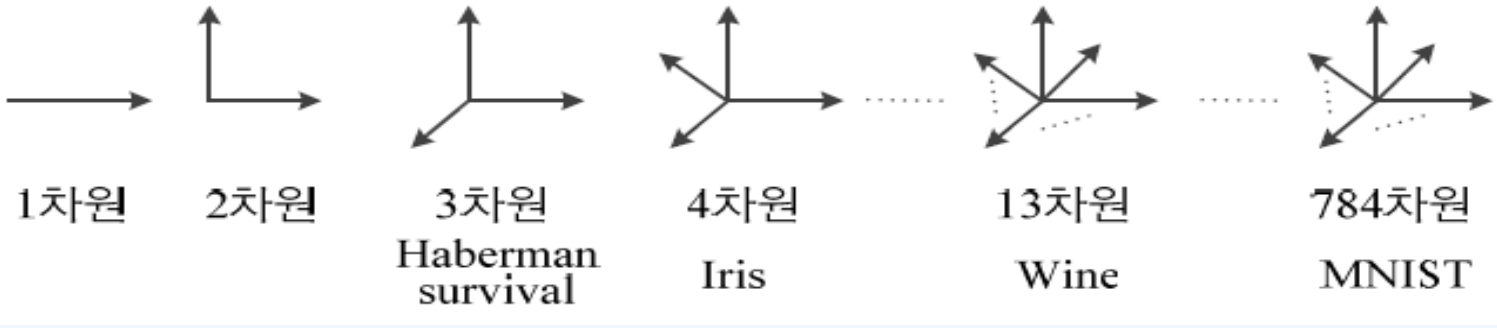


Unsupervised data:{**x**}

2.1 ML data : Multi-dimensional (feature) data

- d-dimensional Feature vector : $\mathbf{x}=(x_1,x_2, \dots ,x_d)^T$

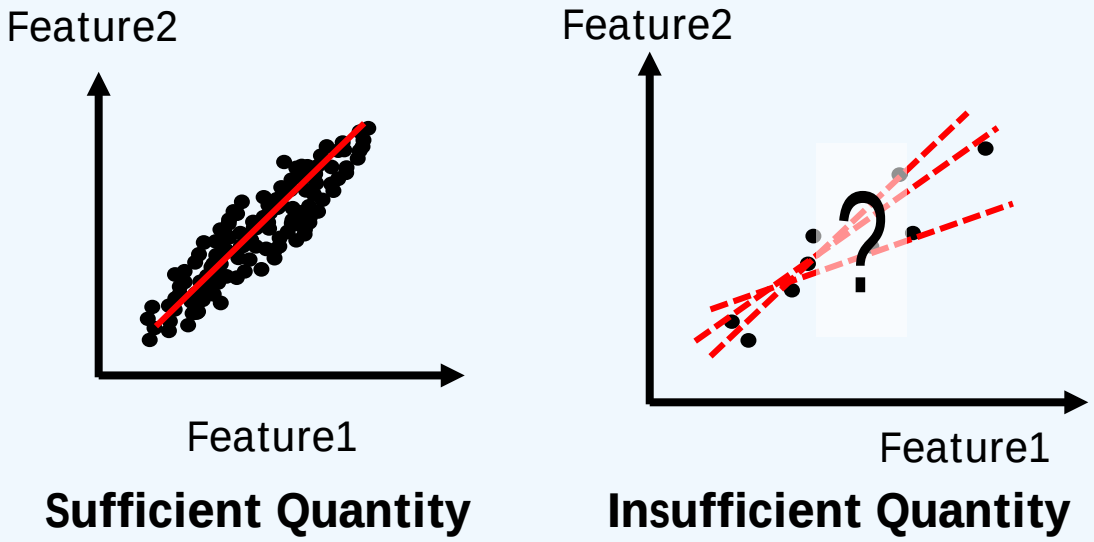
- Haberman survival: $\mathbf{x} = (\text{나이}, \text{수술년도}, \text{양성 림프샘 개수})^T$: 유방암 환자의 수술 후 5년 생존과 관련한 데이터
- Iris: $\mathbf{x} = (\text{꽃받침 길이}, \text{꽃받침 너비}, \text{꽃잎 길이}, \text{꽃잎 너비})^T$: 꽃의 분류에 관한 데이터
- Wine: $\mathbf{x} = (\text{Alcohol}, \text{Malic acid}, \text{Ash}, \text{Alcalinity of ash}, \text{Magnesium}, \text{Total phenols}, \text{Flavanoids}, \text{Nonflavanoid phenols}$
  $\text{Proanthocyanins}, \text{Color intensity}, \text{Hue}, \text{OD280 / OD315 of diluted wines}, \text{Proline})^T$
- MNIST: $\mathbf{x} = (\text{화소1}, \text{화소2}, \cdots, \text{화소784})^T$: 28x28=784 화소로 된 bitmap 영상 데이터. 목포 값은 0,1..9중 하나 선택



- Training Model
  - 1 차원 모델 : $y = w_1x_1 + w_2x_2 + \cdots + w_dx_d + b$ ⟶ Number of parameters : d=1
  - 2 차원 모델 : $y = w_1x_1^2 + w_2x_2^2 + \cdots + w_dx_d^2 + w_{d+1}x_1x_2 + \cdots + w_{d^2}x_{d-1}x_d + w_{d^2+1}x_1 + \cdots + w_{d^2+d}x_d + b$ ⟶ Number of parameters : d^2+d+1

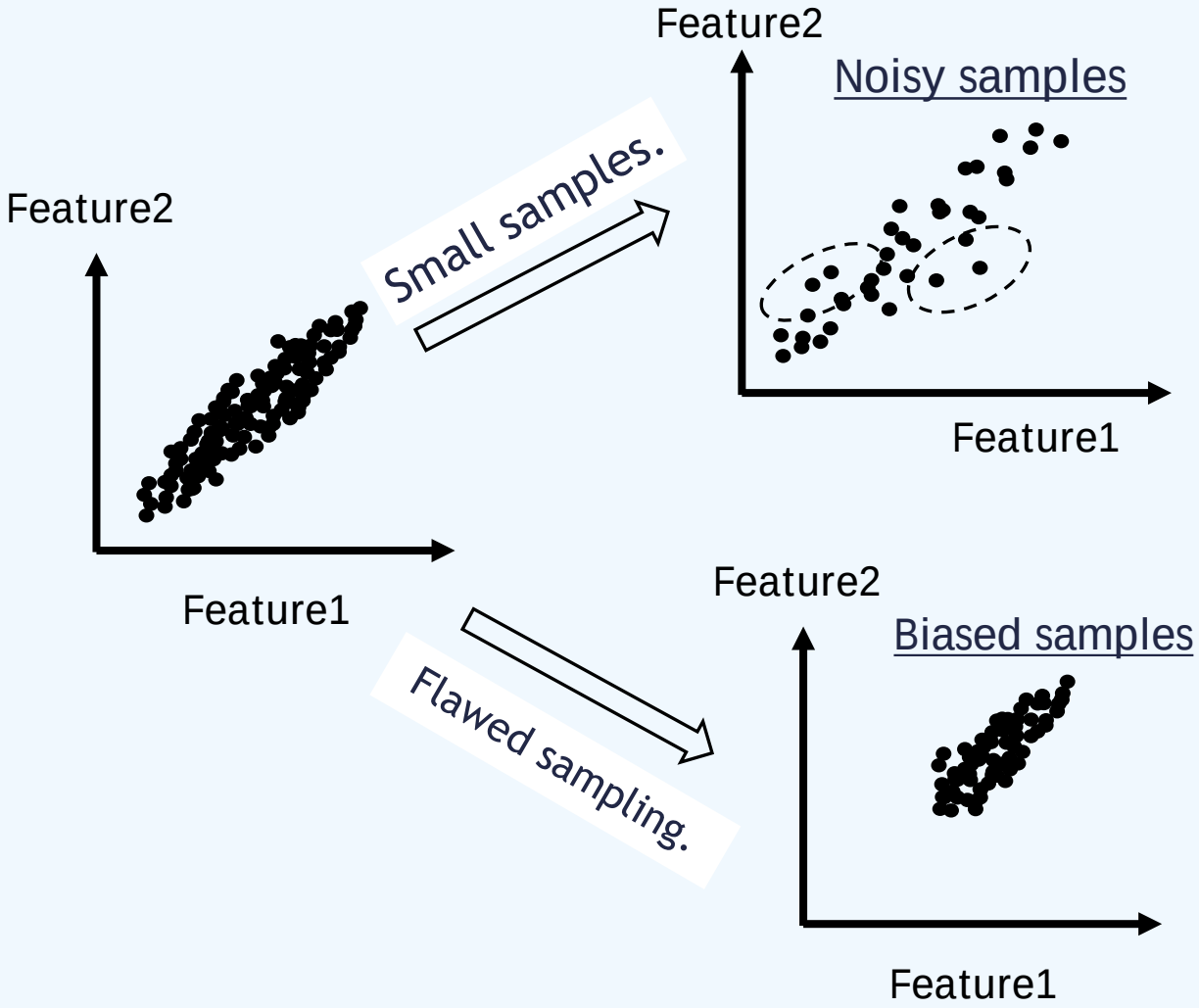
2.1 ML Data : Issues (1/3)

- Insufficient Quantity



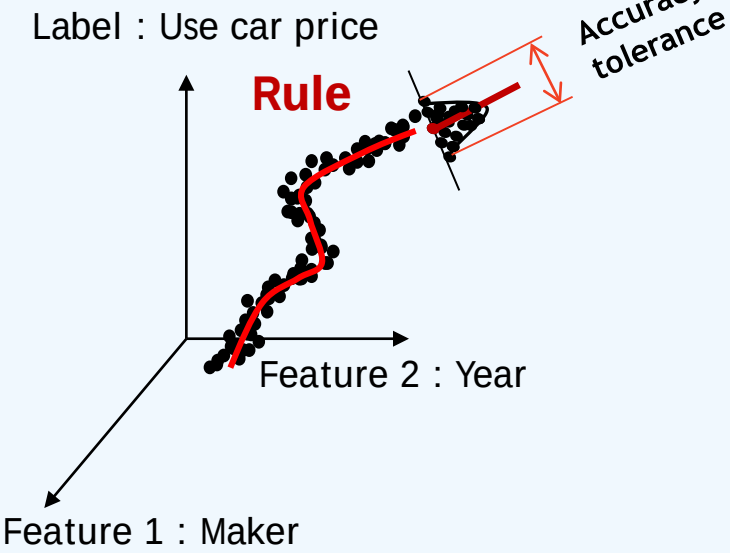
- It takes a lot of data for most Machine Learning algorithms to work properly.
- For more complex problems, more data are needed.

- Non-representative Data

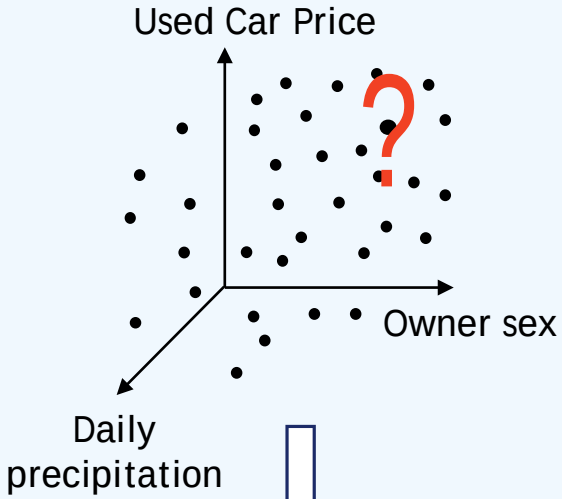


2.1 ML Data : Issues (2/3) – Irrelevant data :Supervised

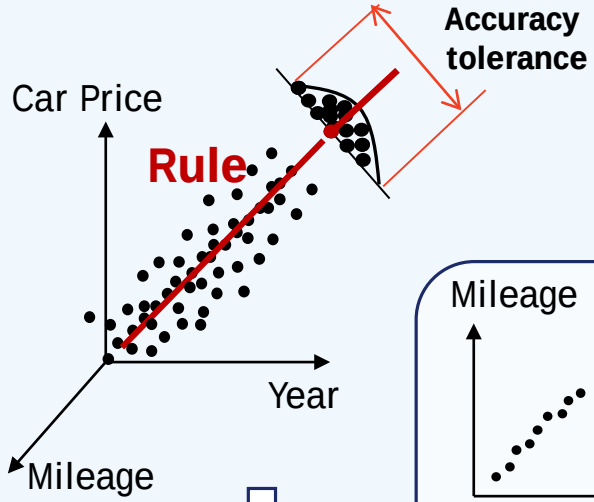
- Relevant and less-correlated features (good features)

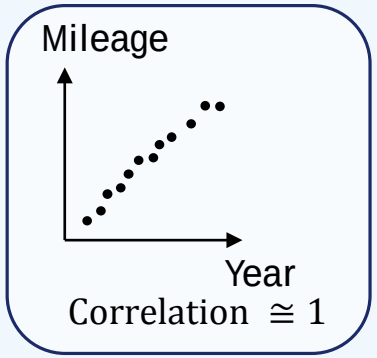


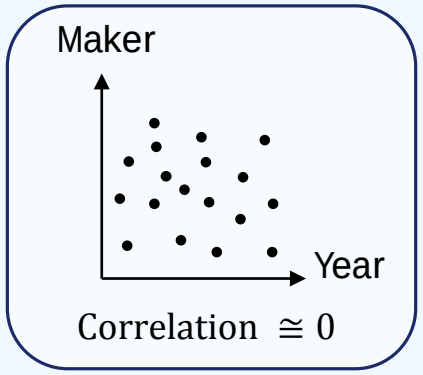
- Irrelevant features



- Correlated (Redundant) features







Feature Engineering

Feature selection

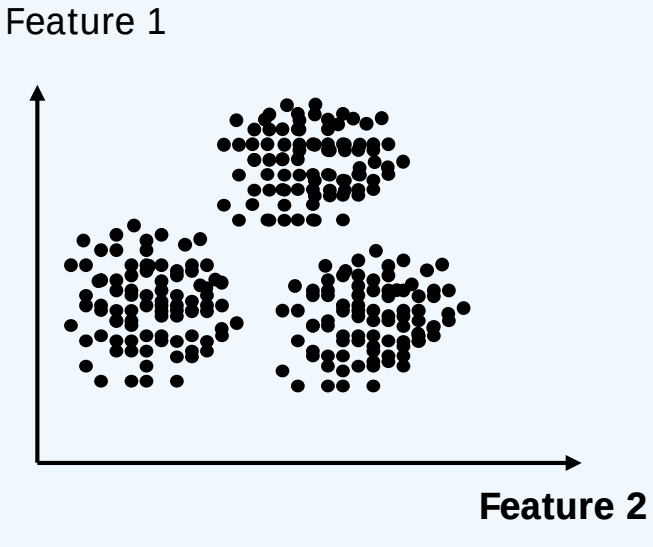
Selecting the most useful features.

Feature Extraction

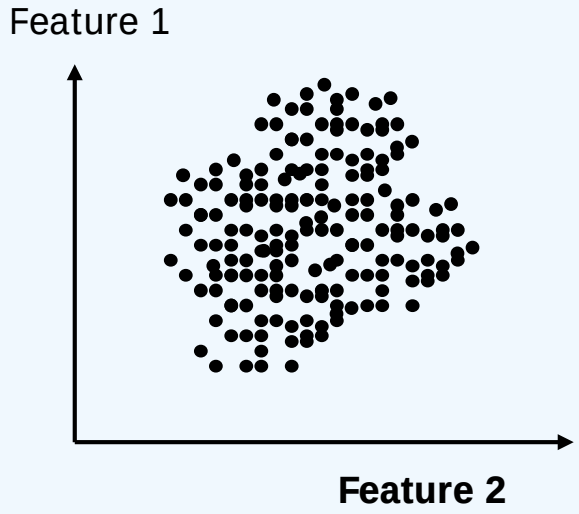
Combining existing features to produce a more useful one.

2.1 ML Data : Issues (3/3)-Irrelevant data -Unsupervised

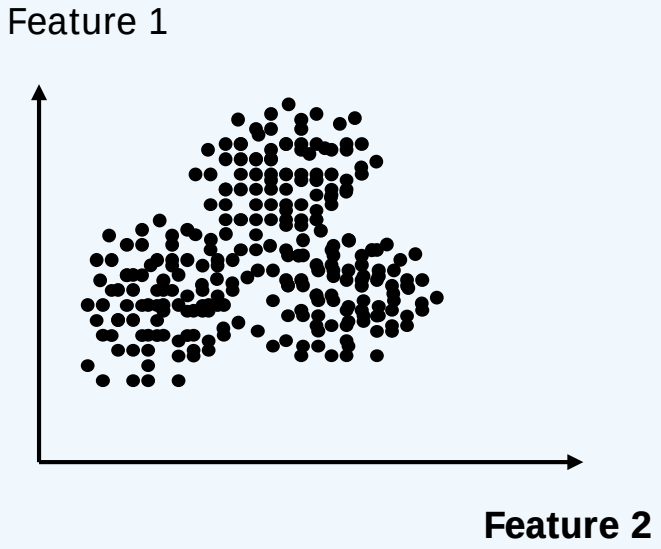
- Relevant and less-correlated features (good features)



- Irrelevant features

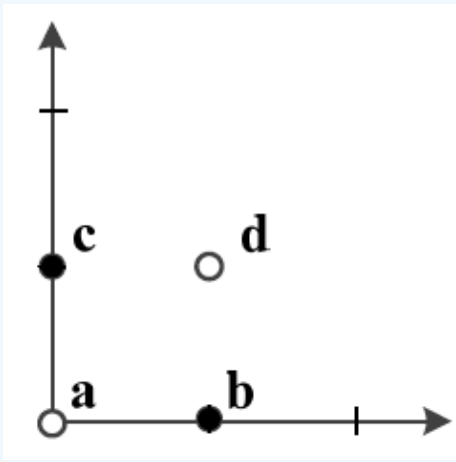


- Correlated (Redundant) features

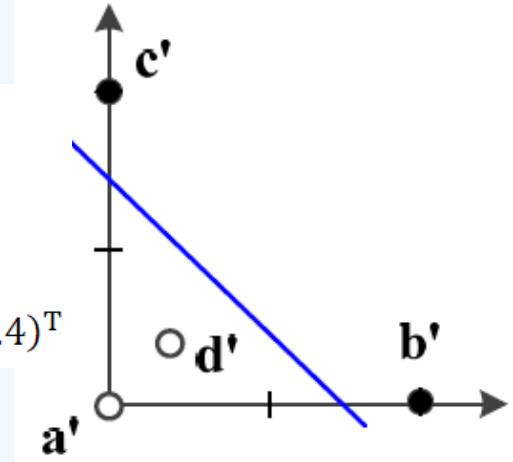


2.1 ML Data : Issues - Data Representation

- **Representation Learning**



$$\mathbf{a} = (0,0)^T \rightarrow \mathbf{a}' = (0,0)^T$$
$$\mathbf{b} = (1,0)^T \rightarrow \mathbf{b}' = (2,0)^T$$
$$\mathbf{c} = (0,1)^T \rightarrow \mathbf{c}' = (0,2)^T$$
$$\mathbf{d} = (1,1)^T \rightarrow \mathbf{d}' = (0.4,0.4)^T$$



75% clustering

100% clustering

원래 특징 벡터 $\mathbf{x} = (x_1, x_2)^T \rightarrow$ 변환된 특징 벡터 $\mathbf{x}' = \left(\frac{x_1}{2x_1x_2 + 0.5}, \frac{x_2}{2x_1x_2 + 0.5} \right)^T$

- **ML의 좋은 특징 공간을 자동으로 찾는 작업이라고 할 수 있음.**
- **딥러닝(Deep learning) 은 다수의 은닉층을 가진 신경망을 이용하여 계층적인 특징 공간을 찾아냄**
  - **왼쪽 은닉층은 low-level feature (에지, 구석점 등), 오른쪽은 high-level feature (얼굴, 바퀴 등) 추출**

2.1 ML Data : Issues - Curse of dimensionality

- 차원의 저주 (curse of dimensionality)
  - 차원이 높아짐에 따라 거대한 특징 공간이 발생하는 현실적인 문제들
  - 예) d=4인 Iris 데이터에서 축마다 100개 구간으로 나누면 총 100^4=1억 개의 칸
  - 예) d=784인 MNIST 샘플의 화소가 0과 1값을 가진다면 2^{784}개의 칸. 이 거대한 공간에 고작 6만 개의 샘플을 흩뿌린 매우 희소한 분포

2.1 ML data : Database (1/5)

- Database(DB) : an organized collection of data, generally stored and accessed electronically from a computer system.

- 기계학습에서의 DB의 필요성
  - 기계 학습이 푸는 문제는 훨씬 복잡함. Ex: 8’ 숫자 패턴과 ‘단추’ 패턴의 다양한 변화 양상
  - 단순한 수학 공식으로 표현 불가능함에 따라서 데이터의 집합으로 부터 예측
  - 모델을 설정하고 모델의 parameter를 training하는데 필수 적 임

- DB의 생성과정
  - ML problem은 데이터 생성 과정을 알 수 없고, 주어진 훈련집합 $\mathbb{X}, \mathbb{Y}$로 예측 모델 또는 생성 모델을 근사 추정
  - 데이터 생성 과정, 방법을 알고 있으면 ML problem이 아님. Data의 생성 과정을 알아 내는 것이 ML.
  - 데이터 생성 과정을 알고 있는 경우는 현실에서는 존재 않음.
  - 수학적 문제와 ML 문제를 구분 해야 함.
  - Ex: 두 개 주사위를 던져 나온 눈의 합을 x라 할 때, y=(x-7)2+1 점을 받는 게임. y의 확률

2.1 ML data : Database (2/5)

- 데이터베이스의 품질
  - 주어진 응용에 맞는 충분히 다양한 데이터를 충분한 양만큼 수집 → 추정 정확도 높아짐
  - 주어진 응용 환경에 맞는 데이터베이스 확보가 중요함.

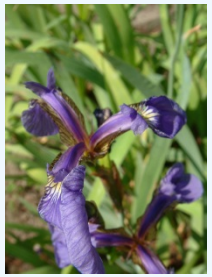
    EX: 정면 얼굴만 가진 데이터베이스로 학습하고 나면, 기운 얼굴은 매우 낮은 성능

- 공개 데이터베이스
  - 위키피디아에서 'list of datasets for machine learning research' 로 검색
  - UCI 리퍼지토리 (2017년11월 기준으로 394개 데이터베이스 제공)
  - 기계 학습의 연습을 위한 3가지 데이터베이스: Iris, MNIST, ImageNet

2.1 ML data : Database (3/5)

- Iris 데이터베이스는 통계학자인 피셔 교수가 1936년에 캐나다 동부 해안의 가스페 반도에 서식하는 3종의 붓꽃(setosa, versicolor, virginica)을 50송이씩 채취하여 만들었다[Fisher1936]. 150개 샘플 각각에 대해 꽃받침 길이, 꽃받침 너비, 꽃잎 길이, 꽃잎 너비를 측정하여 기록하였다. 따라서 4차원 특징 공간이 형성되며 목푯값은 3종을 숫자로 표시함으로써 1, 2, 3 값 중의 하나이다. http://archive.ics.uci.edu/ml/datasets/Iris에 접속하여 내려받을 수 있다.

Sepal length	Sepal width	Petal length	Petal width	Species
5.2	3.5	1.4	0.2	*I. setosa*
4.9	3.0	1.4	0.2	*I. setosa*
4.7	3.2	1.3	0.2	*I. setosa*
4.6	3.1	1.5	0.2	*I. setosa*
7.0	3.2	4.7	1.4	*I. versicolor*
6.4	3.2	4.5	1.5	*I. versicolor*
6.9	3.1	4.9	1.5	*I. versicolor*
5.5	2.3	4.0	1.3	*I. versicolor*
6.3	3.3	6.0	2.5	*I. virginica*
5.8	2.7	5.1	1.9	*I. virginica*
7.1	3.0	5.9	2.1	*I. virginica*
6.3	2.9	5.6	1.8	*I. virginica*



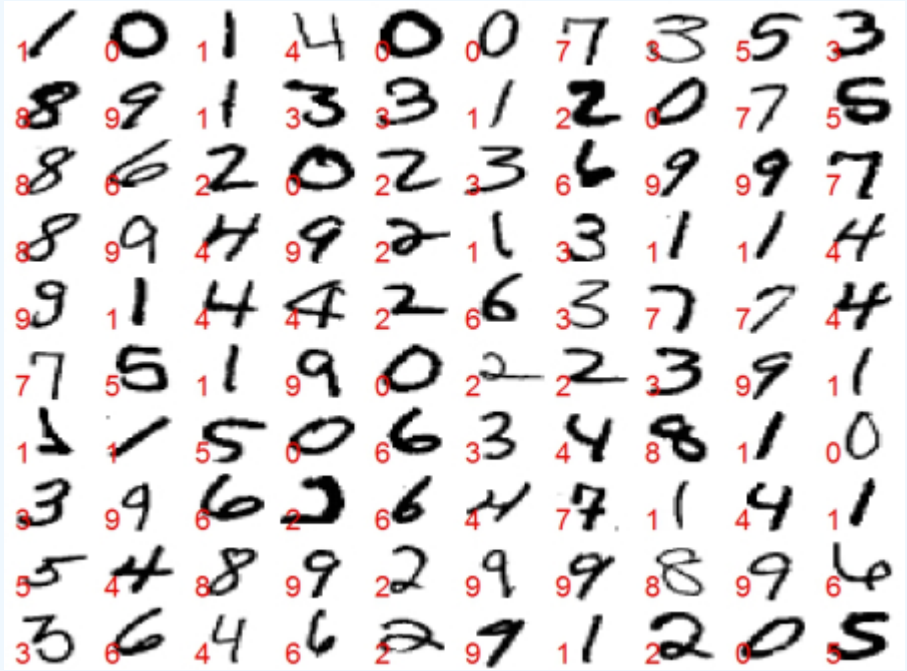
Setosa

Versicolor

Virginica

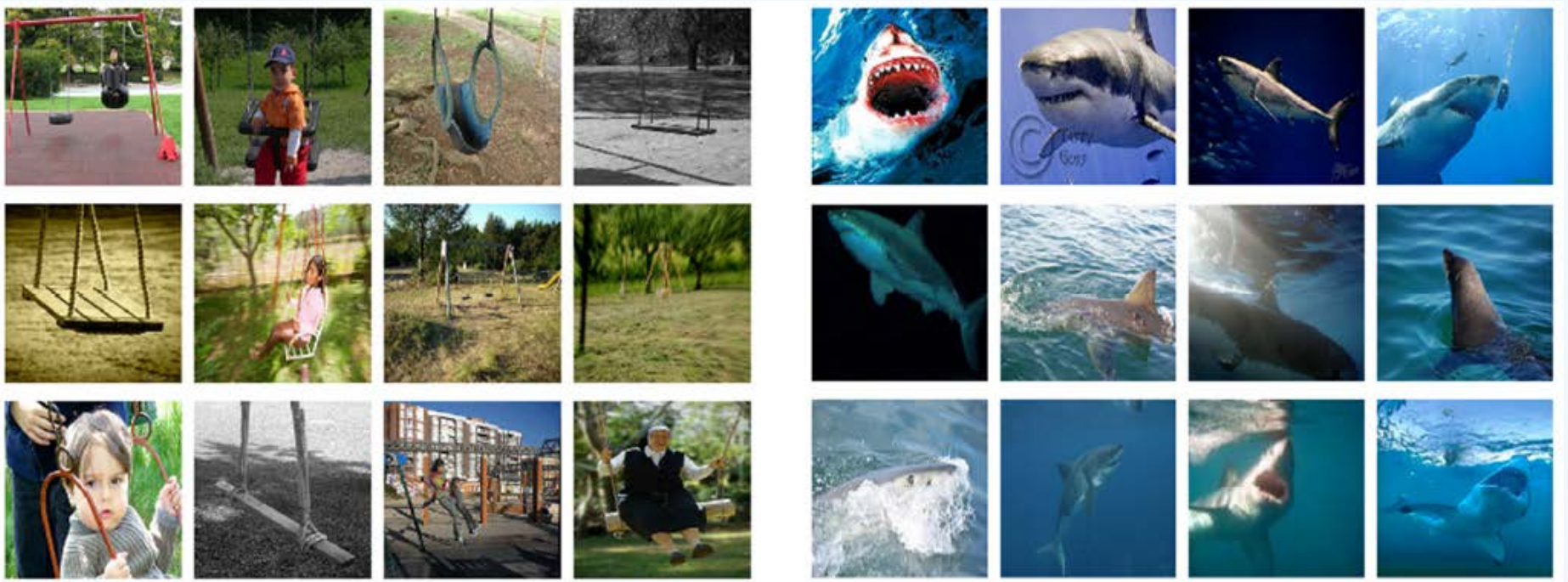
2.1 ML data : Database (4/5)

- MNIST 데이터베이스는 미국표준국(NIST)에서 수집한 필기 숫자 데이터베이스로, 훈련집합 60,000자, 테스트집합 10,000자를 제공한다. http://yann.lecun.com/exdb/mnist에 접속하면 무료로 내려받을 수 있으며, 1988년부터 시작한 인식률 경쟁 기록도 볼 수 있다. 2017년 8월 기준으로는 [Ciresan2012] 논문이 0.23%의 오류율로 최고 자리를 차지하고 있다. 테스트집합에 있는 10,000개 샘플에서 단지 23개만 틀린 것이다.



2.1 ML data : Database (5/5)

- ImageNet 데이터베이스는 정보검색 분야에서 만든 WordNet의 단어 계층 분류를 그대로 따랐고, 부류마다 수백에서 수천 개의 영상을 수집하였다[Deng2009]. 총 21,841개 부류에 대해 총 14,197,122개의 영상을 보유하고 있다. 그중에서 1,000개 부류를 뽑아 ILSVRC(ImageNet Large Scale Visual Recognition Challenge)라는 영상인식 경진대회를 2010년부터 매년 개최하고 있다. 대회 결과에 대한 자세한 내용은 4.4절을 참조하라. http:// image-net.org에서 내려받을 수 있다.



(a) 'swing' 부류 (b) 'Great white shark' 부류

2.2. Machine Learning Model : Regression

- **Feature Data set : X** = $[x_1, x_2 \cdots x_n]^T$, **Target value set : Y** = $[y_1, y_2 \cdots y_n]^T$
- **Polynomial predicting the target value :** $f_\theta(x)$ => **For an input feature** x_i , $f_\theta(x_i)$ **predicts its target get value** y_i.

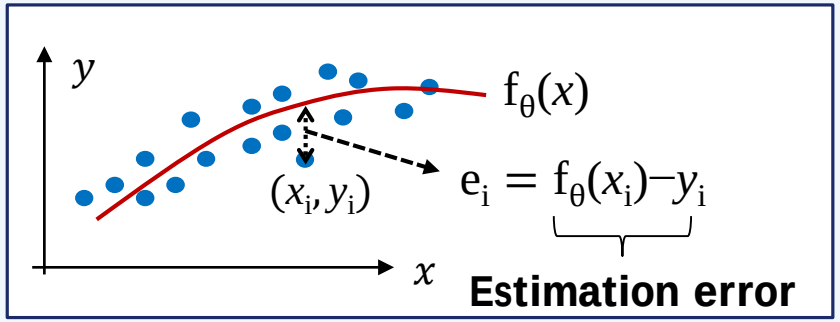
$$f_\theta(x_i) = \theta_0 + \theta_1 x_i + \theta_2 {x_i}^2 \cdots + \theta_p {x_i}^p = \sum_{k=0}^{p} \theta_k {x_i}^k \quad \text{where parameter set } \boldsymbol{\theta} = [\theta_0, \theta_1, \theta_2, \cdots, \theta_p]^T,$$

$$= \mathrm{x_i}^T \cdot \boldsymbol{\theta} \qquad \mathrm{x_i} = [1,\ x_i,\ {x_i}^2, \cdots, {x_i}^p]^T$$

Objective (Cost) function : $J(\Theta) = \frac{1}{n}\sum_{i=1}^{n} |e_i|^2 = \frac{1}{n}\sum_{i=1}^{n} (f_\Theta(x_i) - y_i)^2$

(where $\mathbf{X_n} = [\mathrm{x_0} + \mathrm{x_1} + \mathrm{x_2}, \cdots, +\mathrm{x_n}]^T$)

$= (\mathbf{X_n} \cdot \boldsymbol{\theta} - \boldsymbol{Y})^T (\mathbf{X_n} \cdot \boldsymbol{\theta} - \boldsymbol{Y})$



- ML Training : Decide parameters so that the model minimizes MSE.

$\boldsymbol{Y}$

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmin}} J(\Theta) \Rightarrow$$

$$\boldsymbol{J(\theta)} = (\mathbf{X_n} \cdot \boldsymbol{\theta} - \boldsymbol{Y})^T (\mathbf{X_n} \cdot \boldsymbol{\theta} - \boldsymbol{Y}) = \boldsymbol{\theta}^T \mathbf{X_n}^T \mathbf{X} \boldsymbol{\theta} - 2(\mathbf{X_n}\boldsymbol{\theta})^T \boldsymbol{Y} + \boldsymbol{Y}^T \boldsymbol{Y}$$

$$\frac{1}{\partial \boldsymbol{\theta}} \boldsymbol{J(\theta)} = 2\mathbf{X_n}^T \mathbf{X_n} \boldsymbol{\theta} - 2\mathbf{X_n}^T \boldsymbol{Y} = \mathbf{0} \Rightarrow \hat{\theta} = (\mathbf{X_n}^T \cdot \mathbf{X_n})^{-1} \cdot \mathbf{X_n}^T \cdot \mathbf{Y}$$

2.2 ML Model: 1 order Regression- learning example

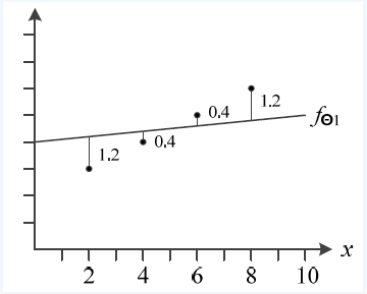
- Model : y= wx+b (1st order)
- Parameters : $\Theta = (w, b)^{\mathrm{T}}$

- 훈련집합
  $\mathbb{X} = \{x_1 = (2.0), x_2 = (4.0), x_3 = (6.0), x_4 = (8.0)\},$
  $\mathbb{Y} = \{y_1 = 3.0,\ y_2 = 4.0,\ y_3 = 5.0,\ y_4 = 6.0\}$

- 초기 $\Theta_1 = (0.1, 4.0)^{\mathrm{T}}$

$$\mathbf{x}_1, y_1 \rightarrow \left(f_{\Theta_1}(2.0) - 3.0\right)^2 = \left((0.1 * 2.0 + 4.0) - 3.0\right)^2 = 1.44$$
$$\mathbf{x}_2, y_2 \rightarrow \left(f_{\Theta_1}(4.0) - 4.0\right)^2 = \left((0.1 * 4.0 + 4.0) - 4.0\right)^2 = 0.16$$
$$\mathbf{x}_3, y_3 \rightarrow \left(f_{\Theta_1}(6.0) - 5.0\right)^2 = \left((0.1 * 6.0 + 4.0) - 5.0\right)^2 = 0.16$$
$$\mathbf{x}_4, y_4 \rightarrow \left(f_{\Theta_1}(8.0) - 6.0\right)^2 = \left((0.1 * 8.0 + 4.0) - 6.0\right)^2 = 1.44$$

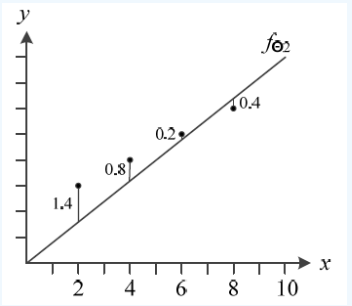
⇨ $J(\Theta_1) = 0.8$ ⇨



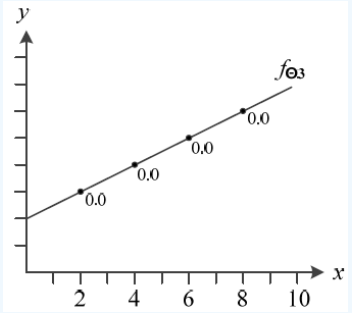
- Θ_1을 개선하여 $\Theta_2 = (0.8, 0.0)^{\mathrm{T}}$ 하면

$$\mathbf{x}_1, y_1 \rightarrow \left(f_{\Theta_2}(2.0) - 3.0\right)^2 = \left((0.8 * 2.0 + 0.0) - 3.0\right)^2 = 1.96$$
$$\mathbf{x}_2, y_2 \rightarrow \left(f_{\Theta_2}(4.0) - 4.0\right)^2 = \left((0.8 * 4.0 + 0.0) - 4.0\right)^2 = 0.64$$
$$\mathbf{x}_3, y_3 \rightarrow \left(f_{\Theta_2}(6.0) - 5.0\right)^2 = \left((0.8 * 6.0 + 0.0) - 5.0\right)^2 = 0.04$$
$$\mathbf{x}_4, y_4 \rightarrow \left(f_{\Theta_2}(8.0) - 6.0\right)^2 = \left((0.8 * 8.0 + 0.0) - 6.0\right)^2 = 0.16$$

⇨ $J(\Theta_2) = 0.7$ ⇨

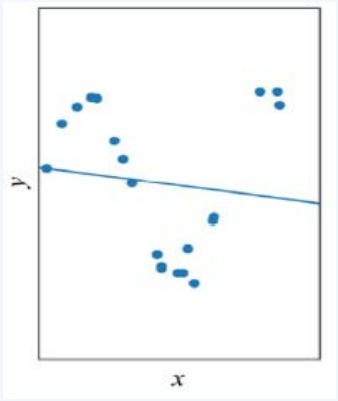


- Θ_2를 개선하여 $\Theta_3 = (0.5, 2.0)^{\mathrm{T}}$ 하면, $J(\Theta_3) = 0.0$ 최적값 $\hat{\Theta}$는 Θ_3. ⇨

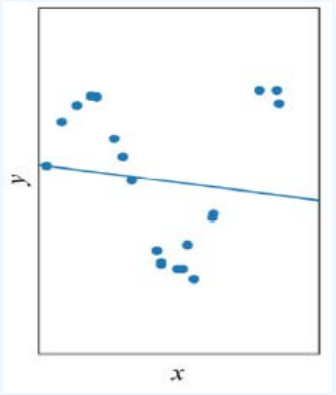


2.2 ML Model : Model Selection

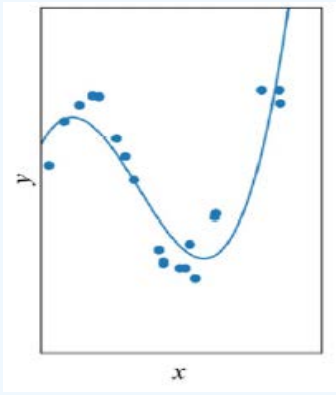
- Underfit : Biased Model



$f_\theta(x) = \theta_0 + \theta_1 x$



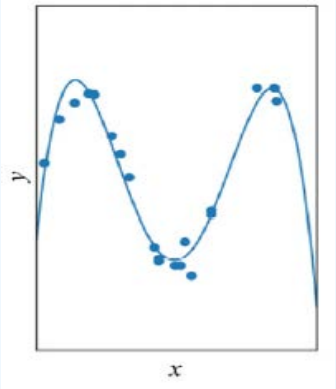
$f_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$



$f_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_2 x^3$

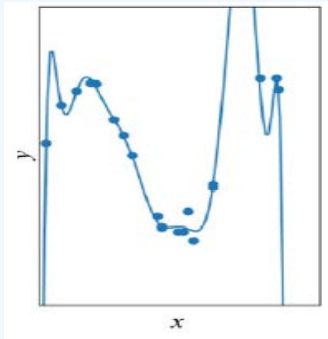
- A model is too simple to learn the underlying structure of the data.
- A model shows low variance (resolution) but high bias.
- Possible solution to Overfitting
  - ✓Change the model to have more parameter.
  - ✓Select the better features (feature engineering).

- Just right



$f_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_2 x^3 + \theta_2\, x^4$

- Overfit : High variance (resolution) Model

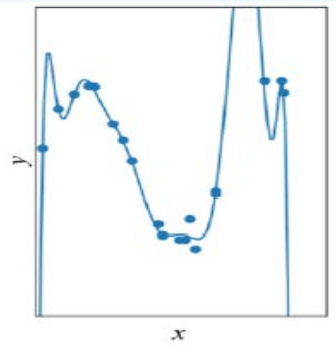


$f_\theta(x) = \theta_0 + \theta_1 x \cdots + \theta_p x^{12}$

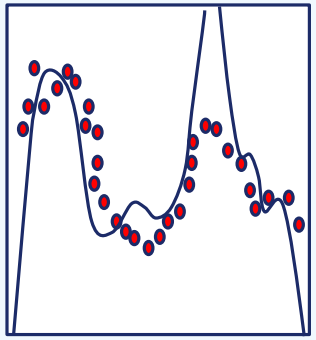
- A model is too much complex. That is, a model has too many parameters relative to data quantity.
- A model begins to memorize training data rather than learn the trend of data.
- Possible solution to Overfitting
  - ✓ Select simpler model with fewer parameters,
  - ✓ Gather more training data
  - ✓ Constraining model by regularization.
  - ✓ Remove noisy data

2.2 ML Model : Looking at Under- and Over-fitting

- Underfitting
  - 훈련집합과 테스트집합 모두 낮은 성능
- Overffing
  - 12차 다항식 곡선을 채택한다면 훈련집합 (Training data set) 에 대해 거의 완벽하게 근사화
  - 하지만 '새로운' 데이터 (Test data set)를 예측한다면 오류가 산만하게 발생함.
  - 이유는 '용량이 크기' 때문. 학습 과정에서 잡음까지 수용 => Overfitting



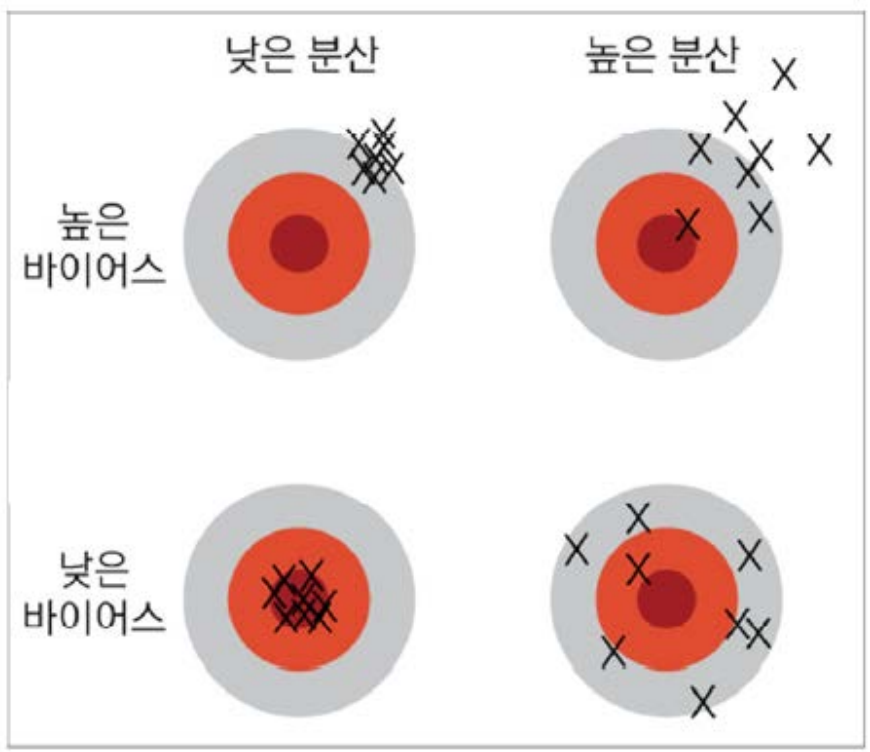
Model for training data
=> Almost perfect estimation



Prediction for test data
=> Scattering estimation errors

- Optimal fitting
  - 훈련집합에 대해 overfitting modle 보다 예측 성능이 낮지만 테스트집합에는 높은 성능
  - 높은 일반화 능력

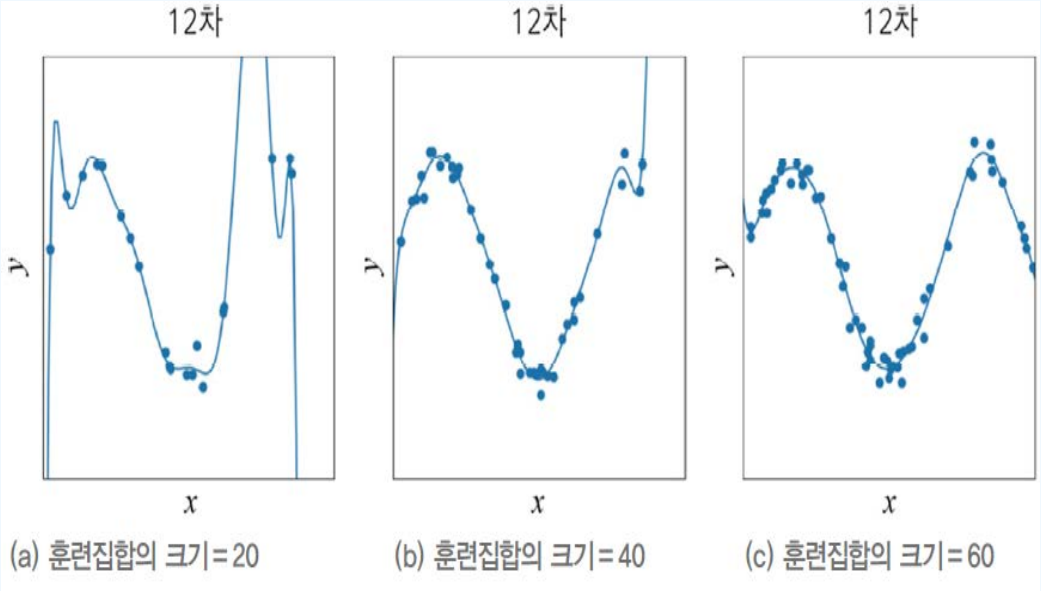
2.2 ML Model : Estimation errors -Biase and Variance



- 기계 학습의 목표
  - 낮은 바이어스와 낮은 분산을 가진 예측기 제작이 목표.
  - 하지만 바이어스와 분산은 트레이드오프 관계
  - 따라서 바이어스 희생을 최소로 유지하며 분산을 최대로 낮추는 전략 필요

2.2 ML Model : 모델 선택의 해결책 – Data 확대

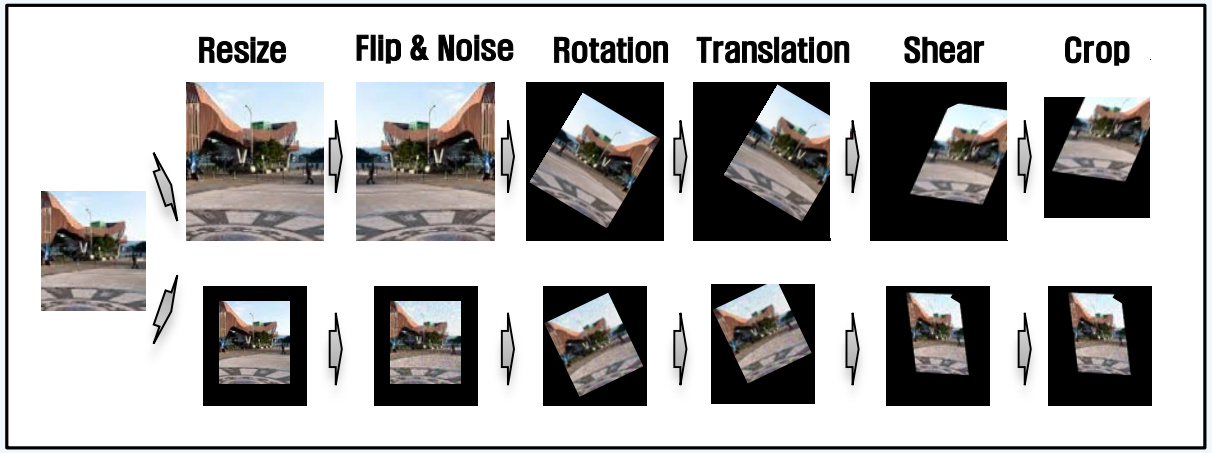
- Data 확대 :
  - 데이터를 더 많이 수집하면 일반화 능력이 향상됨



- 일반적으로 데이터를 수집에 현실적으로 어려움.
- Ground truth data 수집을 위한 labeling 작업등으로 인하여 데이터 수집 비용이 많이 소용됨.

- Data Augmentation
  - 인위적으로 training data 변환하여 data 확대
  - 약간 회전 또는 와핑 (부류 소속이 변하지 않게 주의)

그림 5–24 필기 숫자 데이터의 다양한 변형[8]



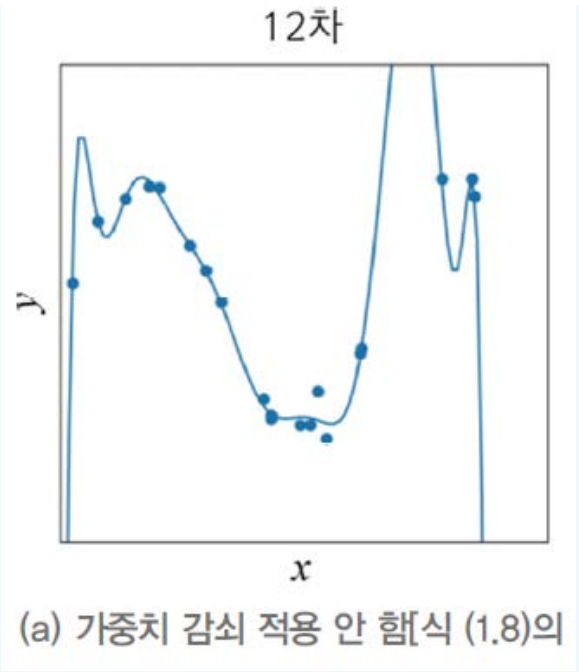
2.2 ML Model : 모델 선택의 해결책 - Regulation

- **Overfitting이 되면 계수들의 크기가 큼 :** $y = 1005.7x^{12} - 27774.4x^{11} + \cdots - 22852612.5x^{1} - 12.8$
- **목적함수에 Regualtion term을 더하여 가중치를 작게 조절**

$$J(\Theta) = \frac{1}{n}\sum_{i=1}^{n}(f_\Theta(\mathbf{x}_i) - y_i)^2 + \lambda\|\Theta\|_2^2$$

Regulation term : 계수들을 작게 유지해 줌

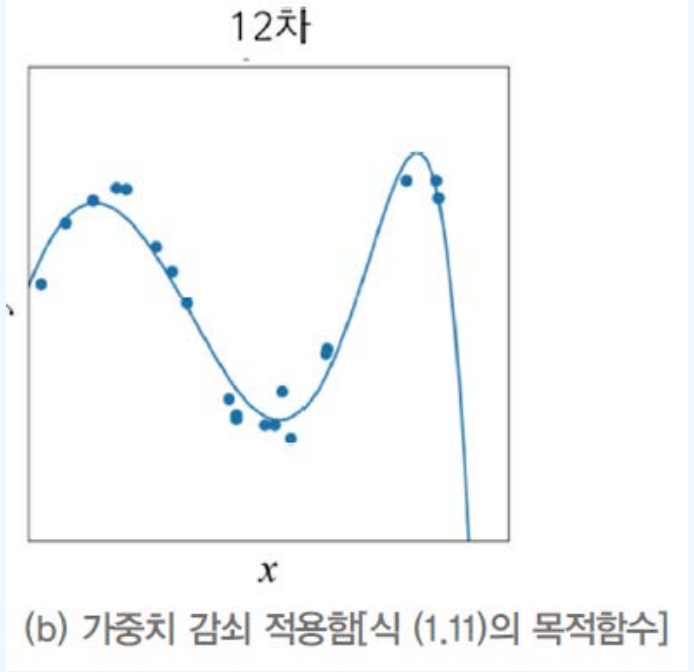
$y = 1005.7x^{12} - 27774.4x^{11} + \cdots - 22852612.5x^{1} - 12.8$



(a) 가중치 감쇠 적용 안 함[식 (1.8)의

Regulation

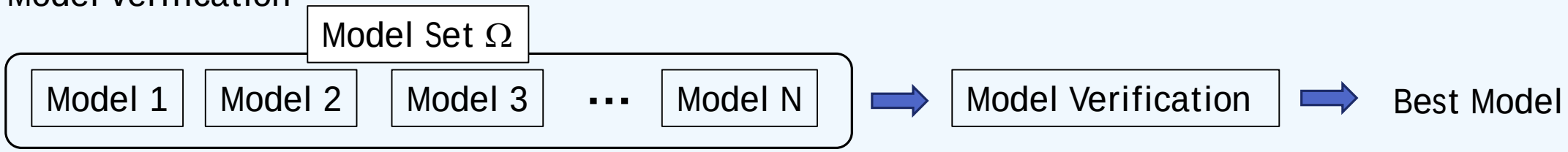
$y = 10.779x^{12} - 42.732x^{11} + \cdots - 2.379x^{1} + 0.119$



(b) 가중치 감쇠 적용함[식 (1.11)의 목적함수]

2.2 ML Model : Model Verification

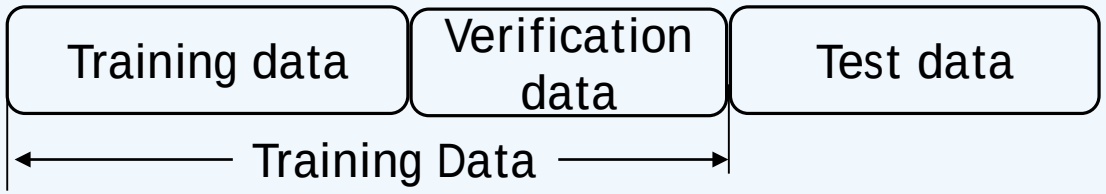
- Model verification



- Method using verification set

```
1  for (Ω에 있는 각각의 모델)
2      모델을 훈련집합으로 학습시킨다.
3      검증집합으로 학습된 모델의 성능을 측정한다.  // 검증 성능 측정
4  가장 높은 성능을 보인 모델을 선택한다.
5  테스트집합으로 선택된 모델의 성능을 측정한다.
```

Useful only when data set is large enough.

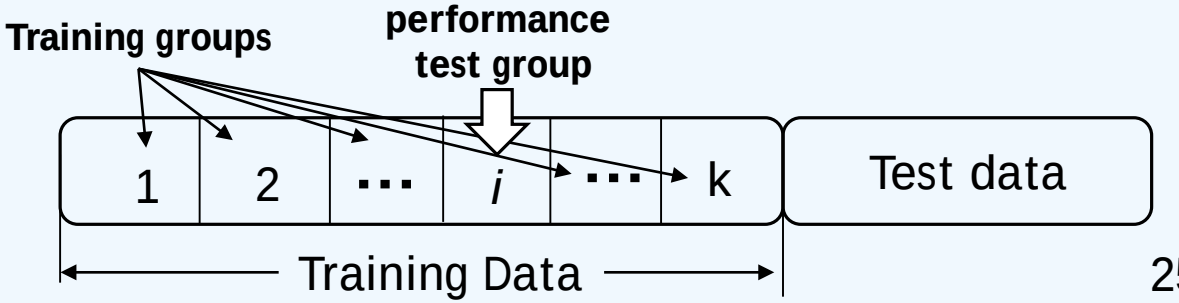


Do not touch test data

- Cross verification method

```
1  훈련집합을 k개의 그룹으로 등분한다.
2  for (Ω에 있는 각각의 모델)
3      for (i=1 to k)
4          i번째 그룹을 제외한 k-1개 그룹으로 모델을 학습시킨다.
5          학습된 모델의 성능을 i번째 그룹으로 측정한다.
6      k개 성능을 평균하여 해당 모델의 성능으로 취한다.
7  가장 높은 성능을 보인 모델을 선택한다.
8  테스트집합으로 선택된 모델의 성능을 측정한다.
```

For Model *p*, at *i*,

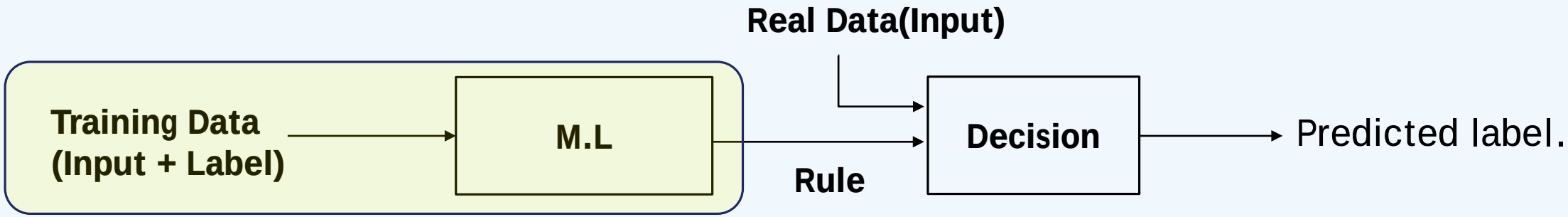


2.3 Classifications of Machine Learning : Overview

Criteria	Method	Description
Training method	Supervised	Learning with the training data with label or target. The system is learned with human supervision (labels).
	Unsupervised	Learning with data without label or target. The system is learned without a teacher.
	Reinforcement	Learn from states what is the best strategy, called a policy, to get the most reward over time.
Data Sequencing	Batch	The system is trained using all the available data at once.
	Online	Data is sequentially fed either by individually or by small groups called mini-batch.
Design	Instance-based	The system learns the examples by heart, then generalizes to new cases using a similarity measure
	Model-based	The system is to build a model of examples, then use that model to make predictions.

2.3 Classifications of ML: Supervised Learning (1/2)

- Training Data format:
  - Label : Description for identification of a phenomenon, a physical quantity
  - Feature : Individual measurable properties (quantities) of a label.
    (* There could be multiple features attributed to a label.)
  - Rule : label = rule[feature1, feature2, ….] => a RULE decides Label from given Features.
  - Vector data (visualization) format : [feature1, …., featuer-N, Target Value] or
    Label <- [feature1, …., feature-N]

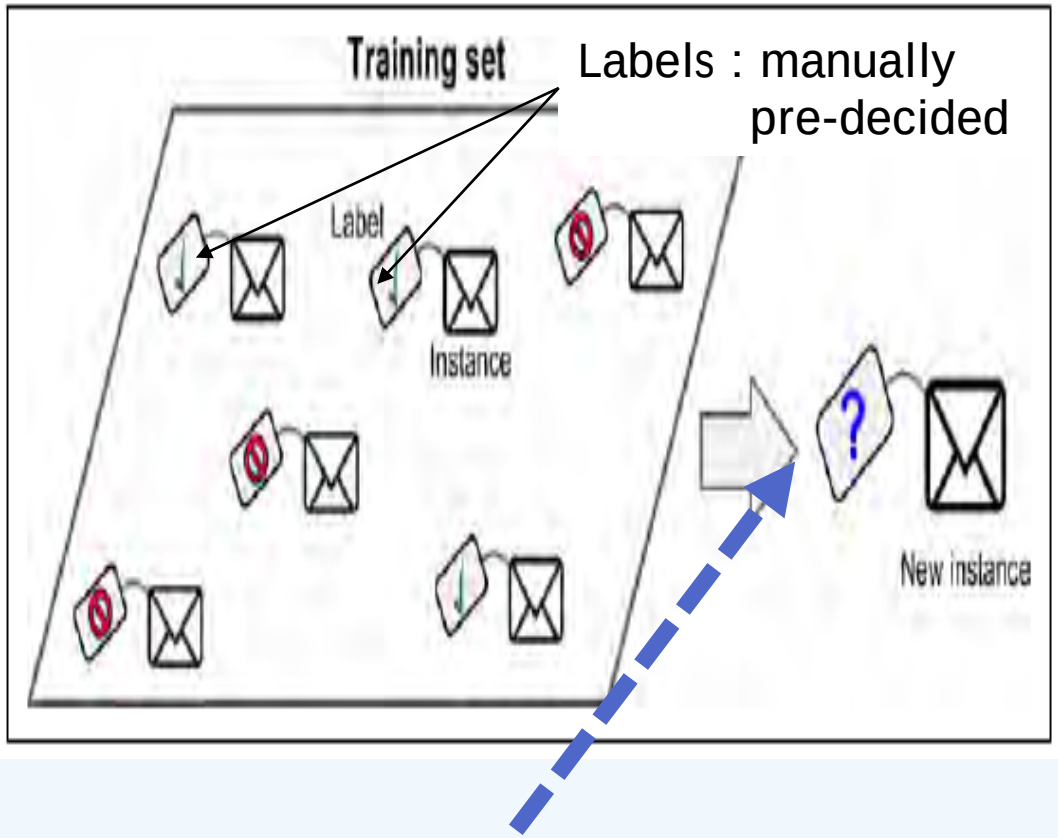


From feature data with label (target), find the optimal rule to predict the label proper to input feature data.

- Typical tasks : Classification, Regression

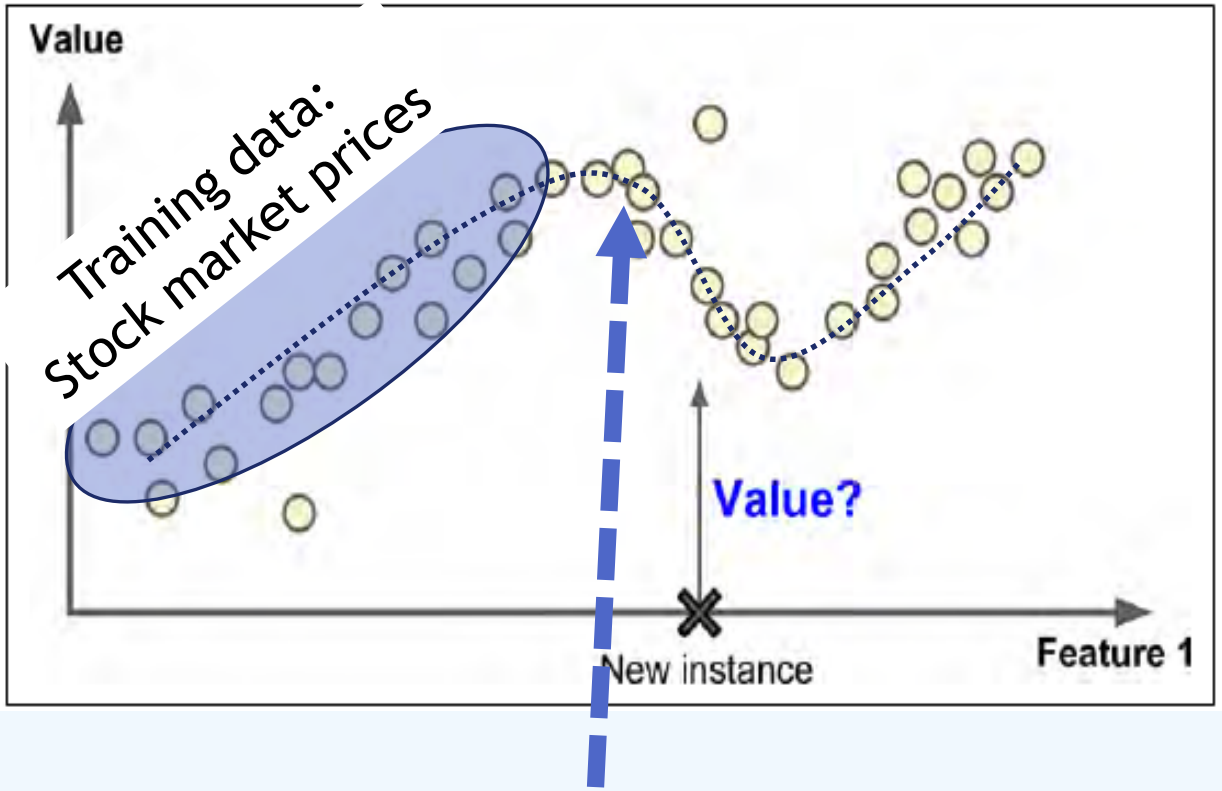
2.3 Classifications of ML : Supervised Learning(2/2) - Examples

- Classification : Spam mail filter



- Learn how to classify new emails.

- Regression : Predicting stock market



- Predicting a target numeric value, given a set of features (training data).

2.3 Classifications of ML : Unsupervised Learning(1/3)

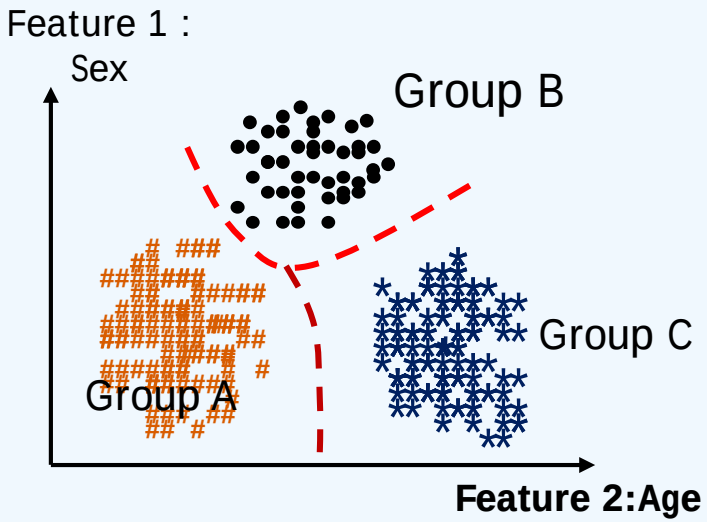
- Training data : No label or target / Vector data format : [feature1, feature2, …., feature N]
- The system tries to learn without a teacher.

Training data
(w/o labels)

Typical unsupervised ML Algorithms	→	Typical tasks
k-Means, Hierarchical Cluster Analysis (HCA), Expectation Maximization (EM)	→	Clustering
Principal Component Analysis (PCA), Locally-Linear Embedding (LLE), t-distributed Stochastic Neighbor Embedding (t-SNE)	→	Visualization & Dimensionality Reduction
Apriori, Eclat	→	Association rule learning /Data Mining
Local Outlier Factor(LOF)	→	Anomaly detection

2.3 Classifications of ML : Unsupervised Learning – Examples (2/3)

• Clustering

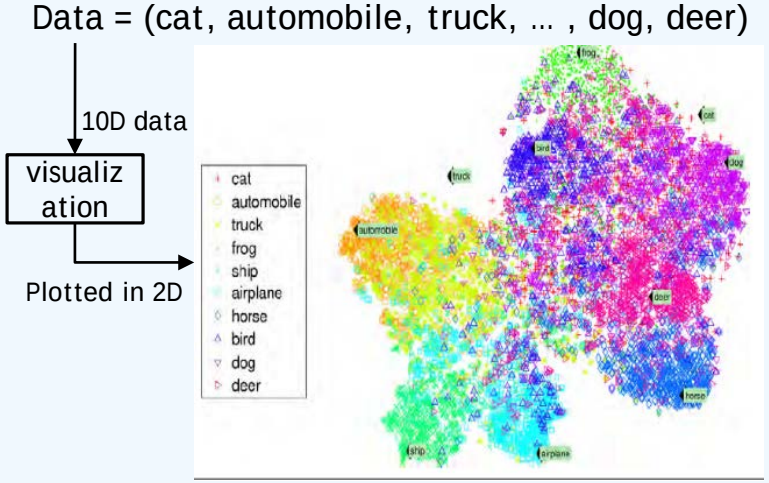


Blog's Visitor Analysis

Data Analysis

- G-A: Age 20s, Male like sports.
- G-B: Age 30s, Male like action movie.
- G-C: Age 20~30s, Female like soap operas.

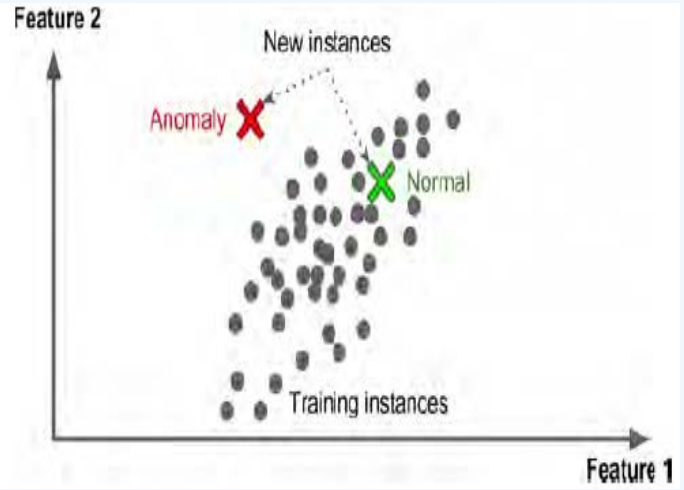
• Visualization



Regional Analysis

- Easily plotted 2D or 3D data representation.
- Trying to keep separate clusters in the input space from overlapping in the visualization
- Provide understanding of how the data is organized.

• Anomaly Detection

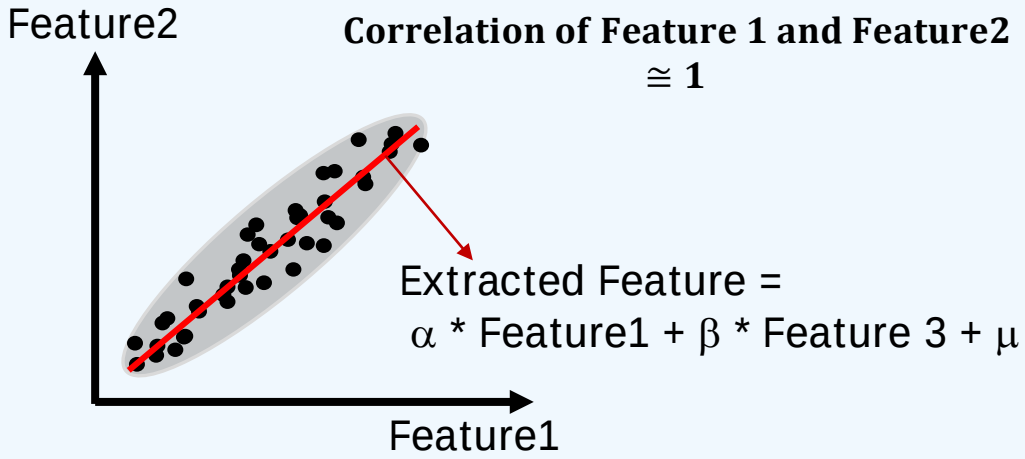


Fraud Card Detection

- Automatically removing or detecting outliers from dataset .
- Trained with normal instance data.

2.3 Classifications of ML : Unsupervised Learning – Examples (3/3)

- Feature Extraction /Dimension Reduction



(Feature1, Feature2) -> Extract Feature

- Merge several correlated features into one.
- Simplify the data without losing too much information.
- 2 features are merged to 1 feature.
- Example: Mileage and Year of used car
  => F=0.8*M+0.2*Year + 3.

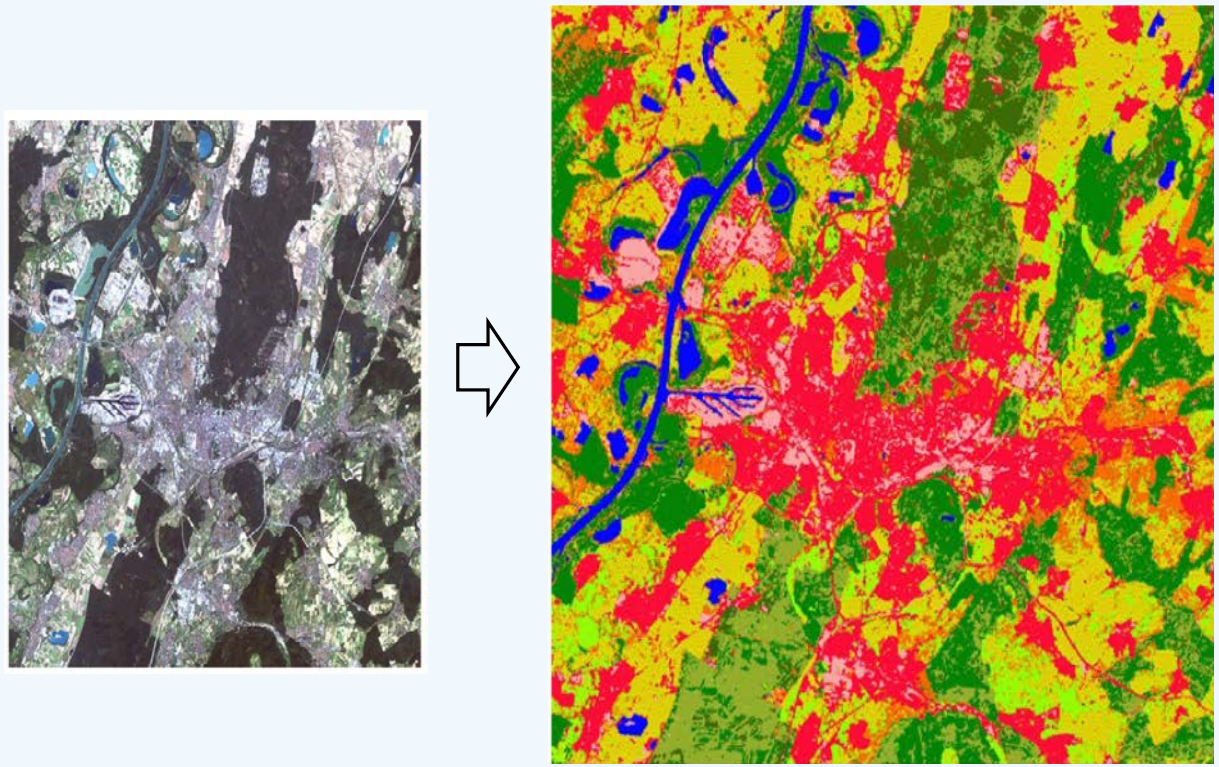
- Association rule learning

- Dig into large amounts of data and discover interesting relations between features.
- Example : Running an association rule on sales logs may reveal that people who purchase barbecue sauce and potato chips also tend to buy steak. Thus, it pays to place these items close to each other.

2.3 Classifications of ML : Classification vs Clustering

Supervised : Classification

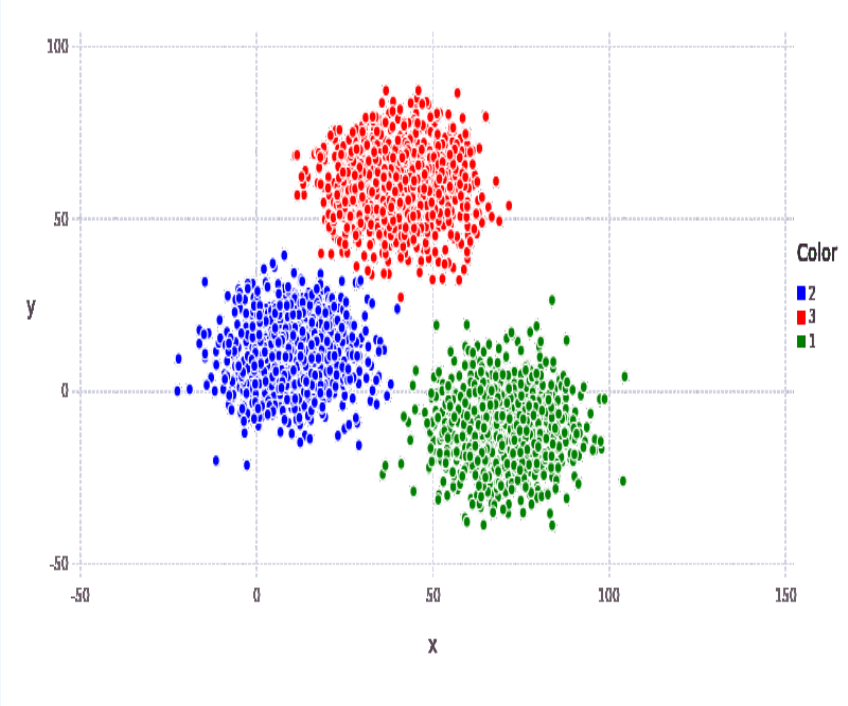
Groups are labeled or known.



: water :buildings :woods :field

Unsupervised : Clustering

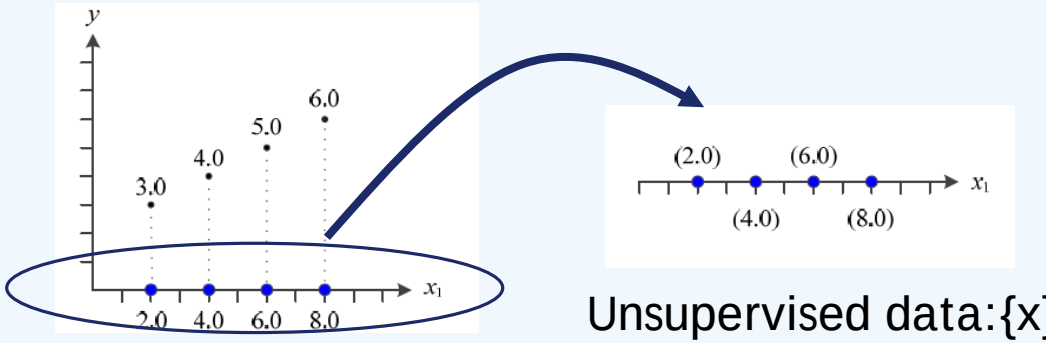
Features are grouped. But groups are not known.
Need investigation to extract information on groups.



2.3 Classifications of ML : Supervised and Unsupervised data

- **Feature data:** $\mathbb{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$, **Corresponding label data :** $\mathbb{Y} = \{y_1, y_2, \cdots, y_n\}$
  where feature data $\mathbf{x}_i$ is a vector and label value y_i is a scalar (value).
- **ML Data Representation :** $\mathbb{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$, $\mathbb{Y} = \{y_1, y_2, \cdots, y_n\}$

– 1D Feature data

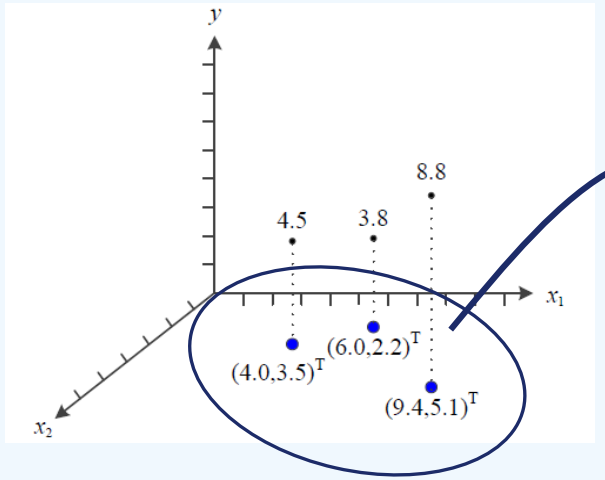


Supervised data:{(x,y)}

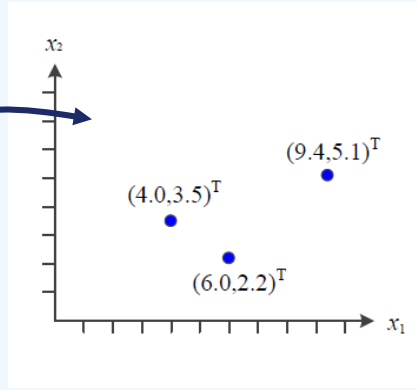
Unsupervised data:{x}

– 2D Feature data

- Feature vector : $\mathbf{x}=(x_1,x_2)^T$
- Examples:
  $\mathbf{x}$=(Weight, Height)^T, y : Driving distance of a ball
  $\mathbf{x}$=(Temperatures, Headache)^T, y : Prob. Of cold

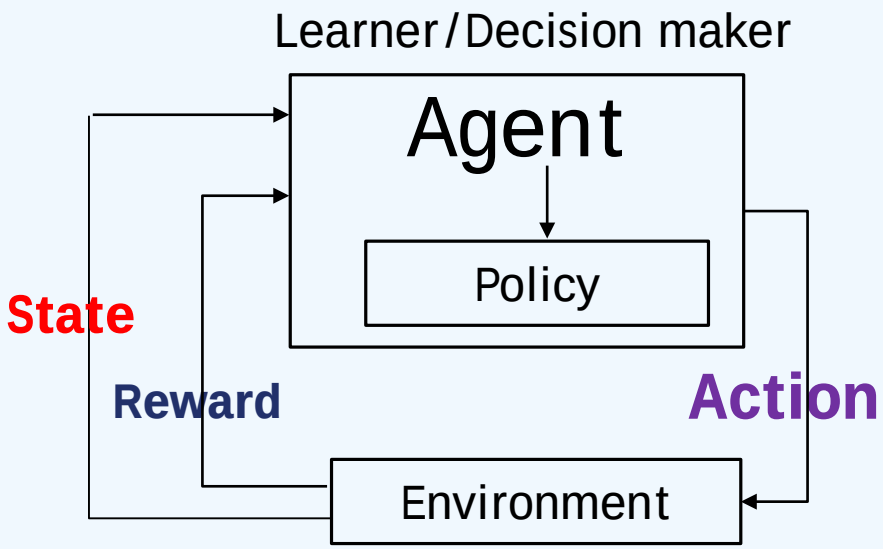


Supervised data:{(**x**,y)}



Unsupervised data:{**x**}

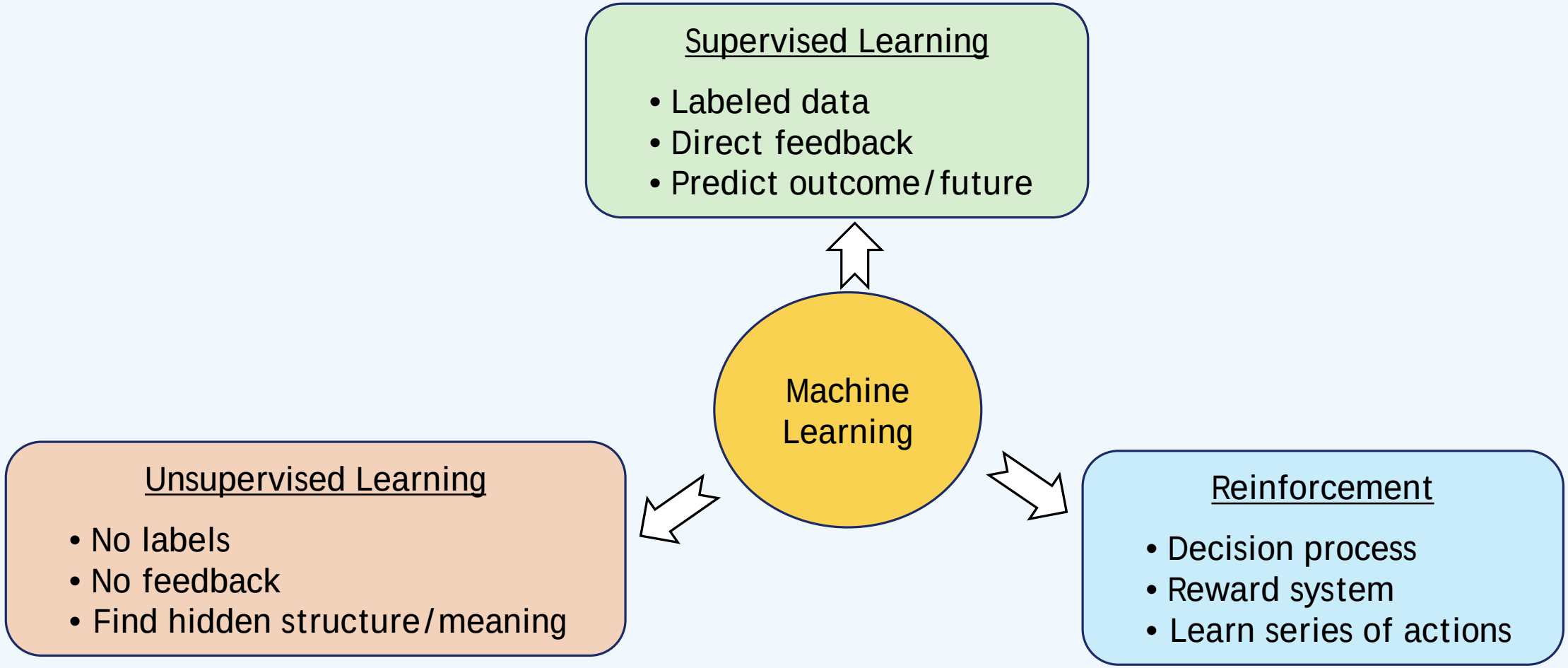
2.3 Classifications of ML : Reinforcement Learning



- (Discrete) *State* describes or determines environment.
- *Reward* is a measurable indicator of better or worse environment.
- *Action* changes state.
- Policy is the best action strategy to get the most average reward over time.

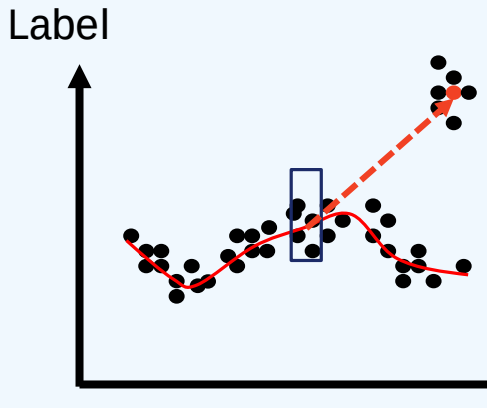
- Possible actions at a state are probabilistically prescribed. The agent takes an action among the possible actions, then
  - the action changes the state,
  - the possible actions are scribed with the new state,
  - the reward computed at environment is returned to the agent.
- Agent learns a *policy (or behavior) through trial-and- error interaction with rewards* over time.

2.3 Classifications of ML : Training method Summary

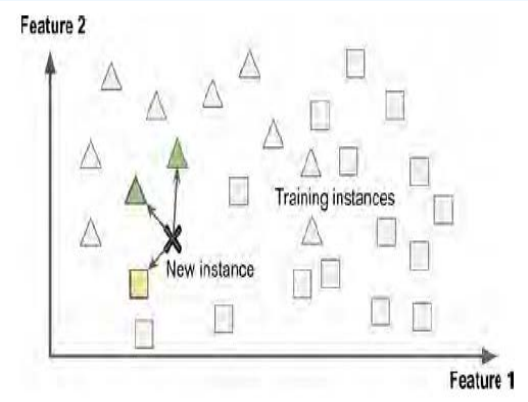


2.3 Classifications of ML : Instance-based vs Model based

- Instance-based Learning (Non-parametric model)

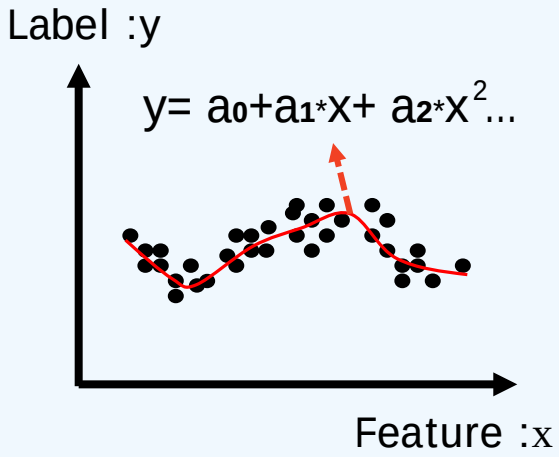


Predict by similarity measure of neighboring data

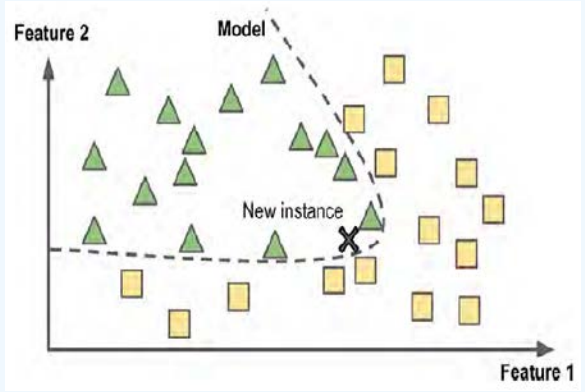


Cluster by similarity measure of neighboring data

- Model-based Learning (Parametric model)



Build data model and then training the model to make prediction.



Build data model and then training the model for clustering.

2.3 Classifications of ML : Online / Batch learning (1/2)

Online learning	Batch learning
• The system is incrementally trained by feeding it data instances sequentially, either individually or by small groups called mini-batches. • Online learning is great for systems that receive data as a continuous flow (e.g., stock prices) and need to adapt to change rapidly or autonomously. • One important parameter of online learning systems is how fast they should adapt to changing data (Learning Rate) • A big challenge with online learning is that if bad data is fed to he system, the system's performance will gradually decline.	• The system is trained using all the available data. • First the system is trained, and then it is launched into production and runs without learning anymore. It just applies what it has learned. This is called *Offline Learning*

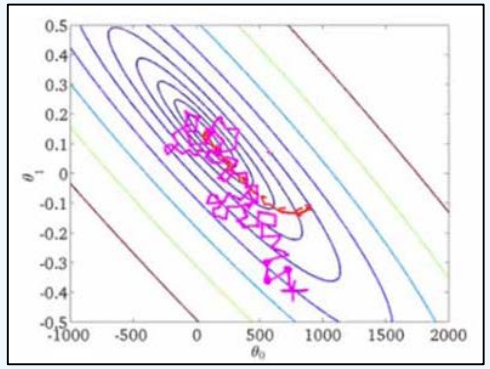
2.3 Classifications of ML : On-line vs Batch learning (2/2)

- Training for regression : $\hat{\Theta} = \underset{\Theta}{\operatorname{argmin}} J(\Theta)$

- On-line learning : data가 계속 입력됨. 작은 개선을 반복하여 최적해를 찾아가는 수치적 방법

입력 : 훈련집합 $\mathbb{X}$와 $\mathbb{Y}$
출력 : 최적의 매개변수 $\hat{\Theta}$

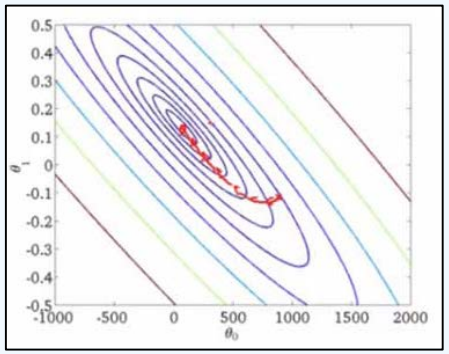
1. 난수를 생성하여 초기 해 Θ_1을 설정한다.
2. t=1
3. while ($J(\Theta_t)$가 0.0에 충분히 가깝지 않음) // 수렴 여부 검사
4. $J(\Theta_t)$가 작아지는 방향 $\Delta\Theta_t$를 구한다. // $\Delta\Theta_t$는 주로 미분을 사용하여 구함
5. $\Theta_{t+1} = \Theta_t + \Delta\Theta_t$
6. t=t+1
7. $\hat{\Theta} = \Theta_t$



- Faster
- Online learning
- Heavy fluctuation
- Capability to jump to new (potentially better local minima)
- Complicated convergence (overshooting)

- Batch learning : data가 묶어서 (data set)으로 들어옴. Analytical 방법으로 solution을 구함

$$\hat{\theta}=(X_n^T \cdot X_n)^{-1} \cdot X_n^T \cdot Y$$



- Gently converges to the (local) minimum
- Very slow
- Intractable for datasets that don't fit in memory
- No online learning