The Introduction To Artificial Intelligence

Yuni Zeng yunizeng@zstu.edu.cn

2022-2023-1

The Introduction to Artificial Intelligence

- Part I Brief Introduction to AI & Different AI tribes
- Part II Knowledge Representation & Reasoning
- Part III AI GAMES and Searching
- Part IV Model Evaluation and Selection
- Part V Machine Learning

Machine Learning

Supervised learning

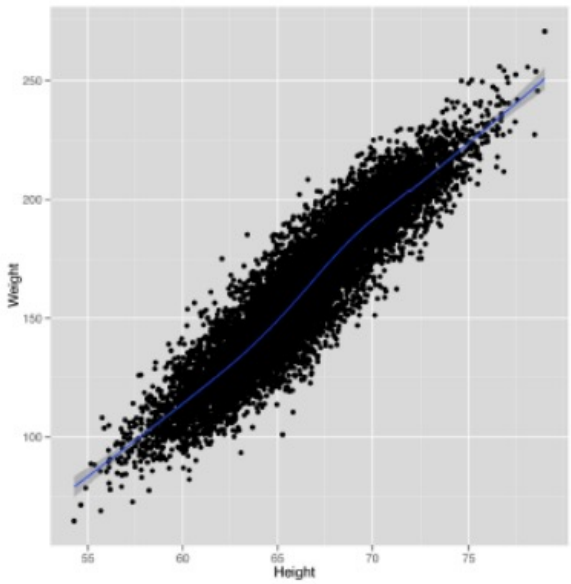
Unsupervised learning

Reinforcement learning

Linear Regression

- What is regression?

Regression is to relate input variables to the output variable, to either predict outputs for new inputs and/or to interpret the effect of the input on the output.



Height is correlated with weight.

Supervised learning

- *Linear Regression*
- Logistic Regression
- Classification
  - Distance-based algorithms
  - Linear classifiers
  - Other classifiers
- ……

Linear Regression

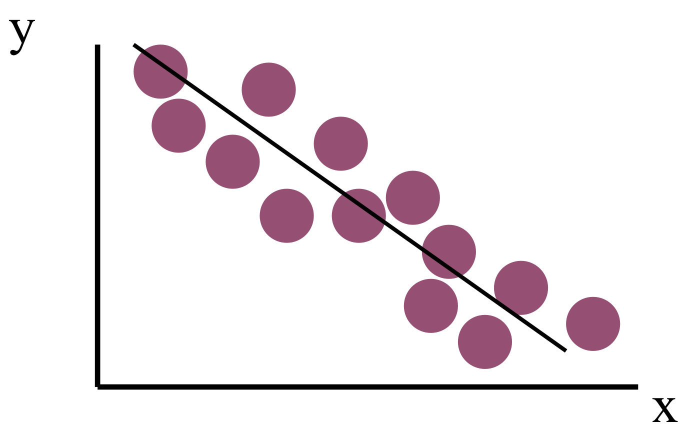
Linear Regression Model

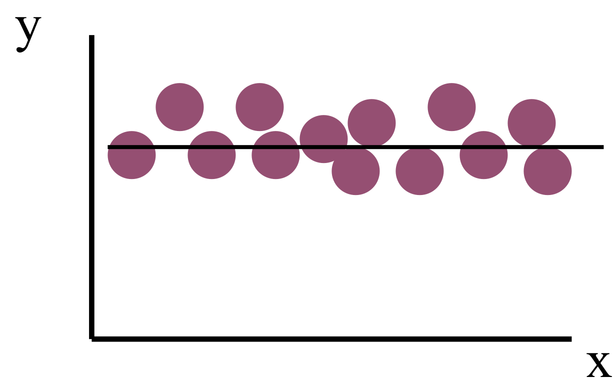
- Only **one** independent variable, x
- Relationship between x and y is described by a linear function
- Changes in y are assumed to be related to changes in x

Linear Regression

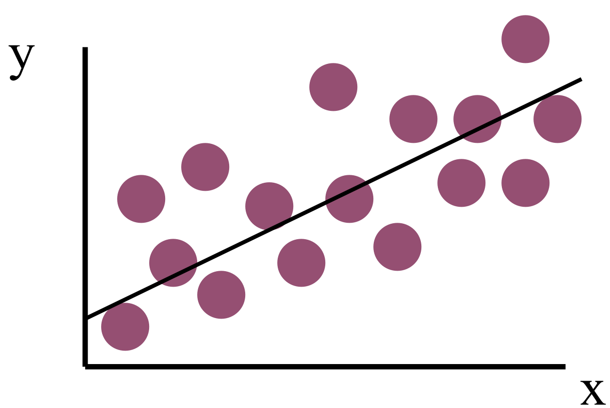
Linear Regression Model

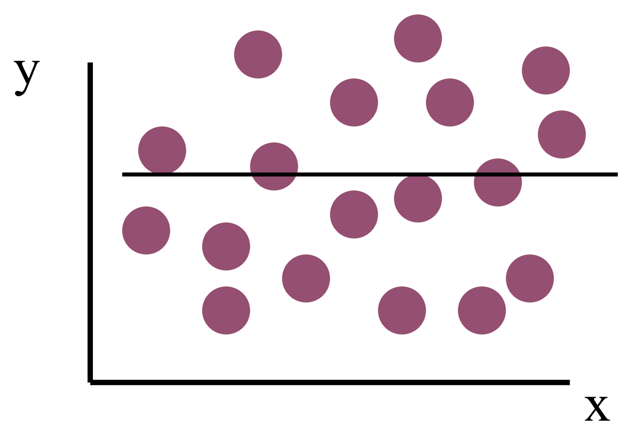
Linear relationships





Question: How to describe the linear relationships?





No relationship

Linear Regression

- Linear Regression Model

Dependent Variable

intercept

Slope Coefficient

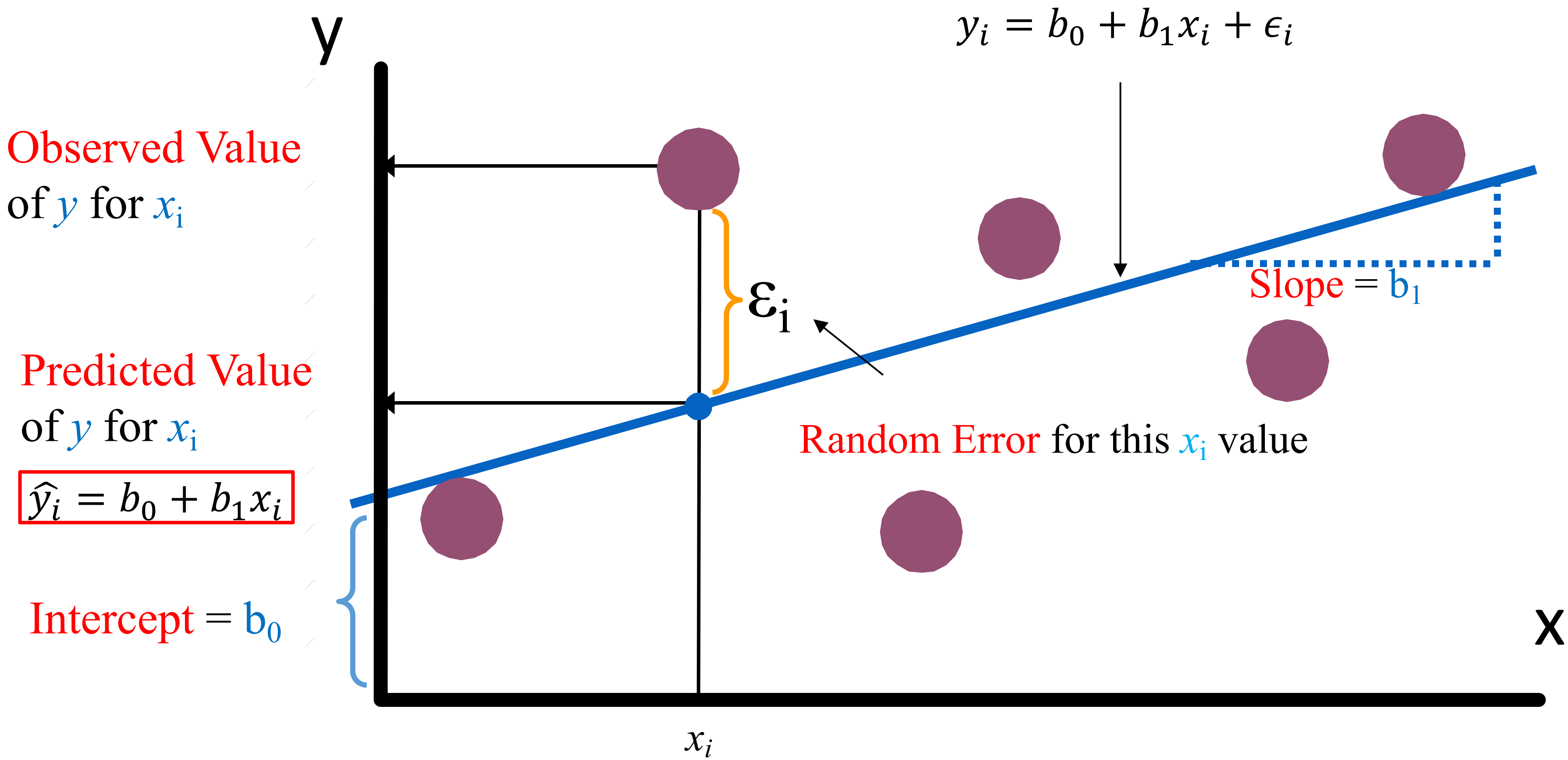
Independent Variable

Random Error term

$$y_i = \underbrace{b_0 + b_1 x_i}_{\text{Linear component}} + \underbrace{\epsilon_i}_{\text{Random Error component}}$$

Linear Regression

- Linear Regression Model



Question: How to obtain the best line?

Linear Regression

- The Least Squares Method

b_0 and b_1 are obtained by finding the values of that minimize the sum of the squared differences between y_i and $\hat{y}_i$ for all i :

$$\min \sum (y_i - \hat{y}_i)^2$$

$\hat{y}_i = b_0 + b_1 x_i$

$$\min \sum (y_i - (b_0 + b_1 x_i))^2$$

Objective function

Question: How to calculate b_0 and b_1?

derivative$[\sum (y_i - (b_0 + b_1 x_i))^2] = 0$ → solve for b_0, b_1

Linear Regression

The Least Squares Method

- Considering the objective function:

$$\mathrm{J} = \sum(y_i - (b_0 + b_1 x_i))^2$$

- Rewrite it in matrix form as:

$$\mathrm{J} = \|Y - \theta^T X\|_2^2$$

where $Y = [y_1, \cdots, y_n]$, $X = \begin{bmatrix} 1 & \cdots & 1 \\ x_1 & \cdots & x_n \end{bmatrix}$, and $\theta = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}$

$$\frac{\partial J}{\partial \theta} = -2(Y - \theta^T X)X^T = 0$$

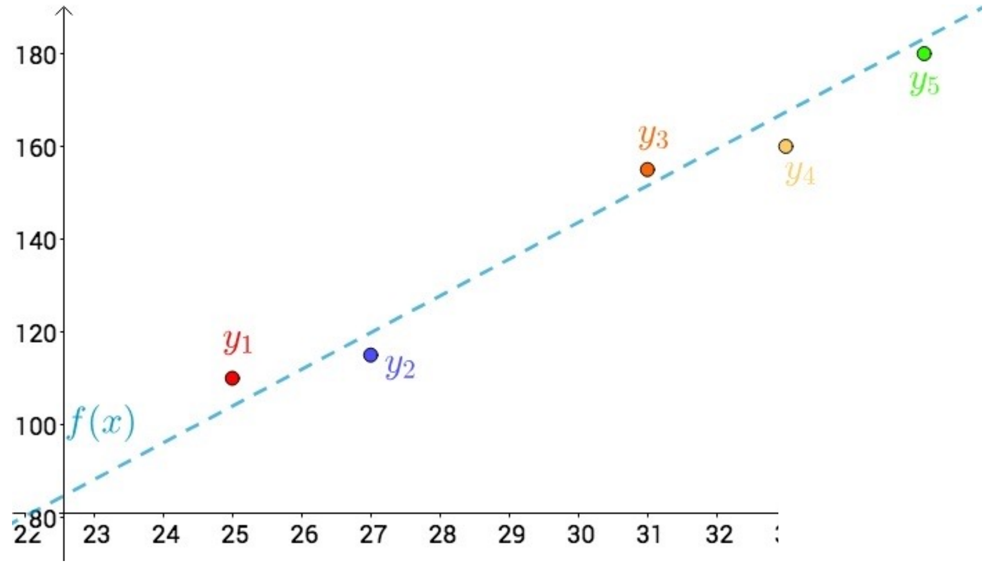
$$\theta^* = (XX^T)^{-1} X Y^T$$

Linear Regression

An Example

- between temperature and ice cream sales:

Temperature	Sales
25°	110
27°	115
31°	155
33°	160
35°	180



Seems like a linear relationship

Linear Regression

An Example

- between temperature and ice cream sales:
- Set: $y = ax + b$

Temperature	Sales
25°	110
27°	115
31°	155
33°	160
35°	180

$\longleftrightarrow$

i	x	y
1	25	110
2	27	115
3	31	155
4	33	160
5	35	180

Linear Regression

An Example

- between temperature and ice cream sales:
- Set: $y = ax + b$
- $J = \sum(f(x_i) - y_i)^2 = \sum(ax_i + b - y_i)^2$
- $\begin{cases} \frac{\partial}{\partial a} J = 2\sum(ax_i + b - y)x_i = 0 \\ \frac{\partial}{\partial b} J = 2\sum(ax_i + b - y) = 0 \end{cases}$
- $\begin{cases} a \approx 7.2 \\ b \approx -73 \end{cases}$

i	x	y
1	25	110
2	27	115
3	31	155
4	33	160
5	35	180

Linear Regression

- Another Example
  - A real estate agent wishes to examine the relationship between the selling price of a houses and its size (measured in square feet)
  - A random sample of 10 houses is selected
    - Dependent variable (y) = house price in $1000s
    - Independent variable (x) = square feet

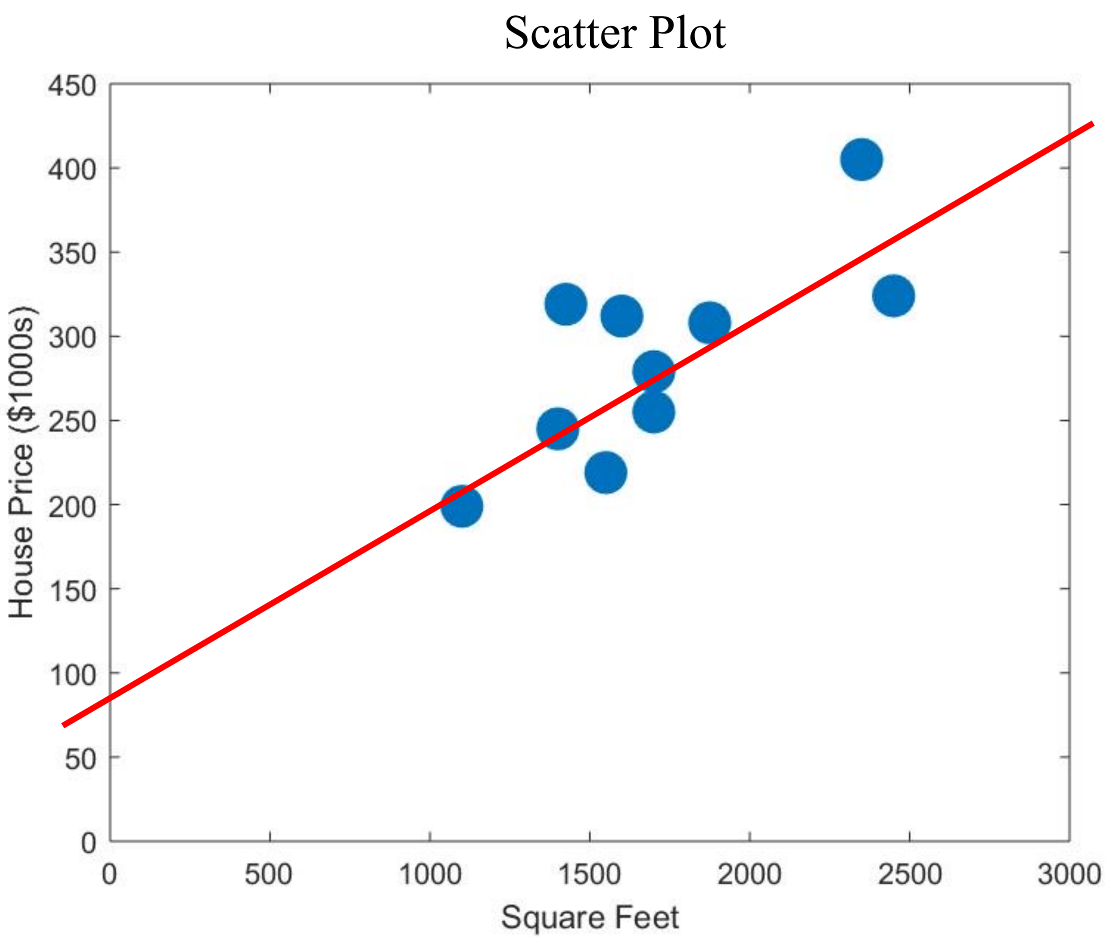
Linear Regression

- An Example

House Price (y) in $1000s	Square Feet (x)
245	1400
312	1600
279	1700
308	1875
199	1100
219	1550
405	2350
324	2450
319	1425
255	1700

Linear Regression

- An Example $\theta^* = (XX^T)^{-1}XY^T$




```
>> theta = inv(X*X')*X*Y'

theta =

   98.2483
    0.1098
```

```
>> [epsilon,b1,b0] = regression(X,Y)

epsilon =

    0.7621


b1 =

    0.1098


b0 =

   98.2483
```

Linear Regression

- <u>Conclusion: Linear Regression</u>
- Uses least squares estimation to estimate parameters
  - Finds the line that minimizes total squared error around the line:
  - Sum of Squared Error (SSE)= $\Sigma(y_i-(b_0 + b_1x))^2$
  - Minimize the squared error function:

    derivative$[\Sigma(y_i-(b_0 + b_1x))^2]=0 \rightarrow$ solve for b_0, b_1

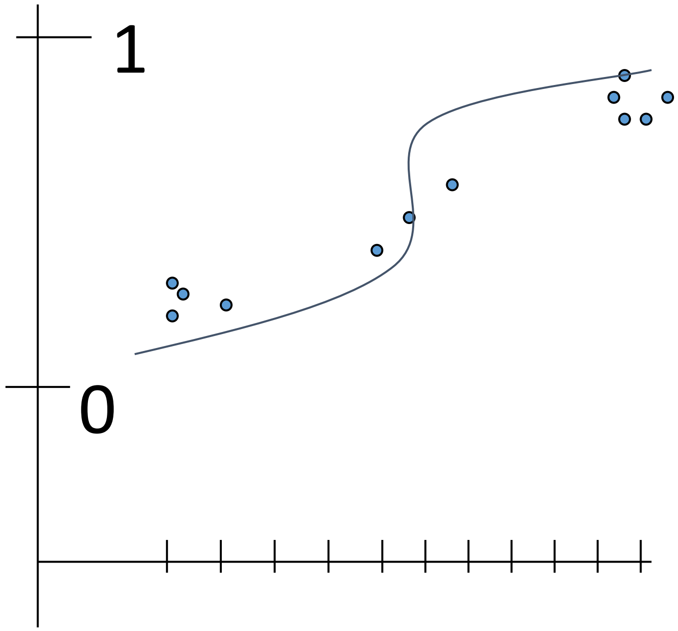
Linear Regression

- Thinking…

Could model probability of lung cancer…

$$P \leftarrow b_0 + b_1 x_i$$

The probability of lung cancer (p)



Smoking (cigarettes/day)

But why might this not be best modeled as linear?

Supervised learning

- Linear Regression
- *Logistic Regression*
- Classification
    - Distance-based algorithms
    - Linear classifiers
    - Other classifiers
- ……

Logistic Regression

Logistic Regression Model

- In medical research, it is often necessary to analyze which factors are related to the outcome of a certain outcome.
- How do we find out which factors have a significant impact on the outcome?
- Logistic regression analysis can solve these problems better.

Logistic Regression

- Linear regression is written as:

$$y = b_0 + b_1 X \qquad -\infty \le y \le +\infty$$

- If we define y as disease or normal, it can not be modeled by the above equation.

- How about apply the probability to represent it?

$$p \leftarrow b_0 + b_1 X$$

Logistic Regression

- Logistic Regression Model

Think about the probability…

probability of disease : p $0 \leq p \leq 1$

probability of no-disease : $1\text{-}p$ $0 \leq p \leq 1$

odds: $\frac{p}{1-p}$ $0 \leq \frac{p}{1-p} < +\infty$

$\ln(\frac{p}{1-p})$ $-\infty < \ln(\frac{p}{1-p}) < +\infty$

Logistic Regression

Logistic Regression Model

Define logistic model as

$$\ln \frac{p}{1-p} = b_0 + b_1 X$$

We obtained that,

$$p = \frac{1}{1 + e^{-(b_0 + b_1 X\ )}}$$

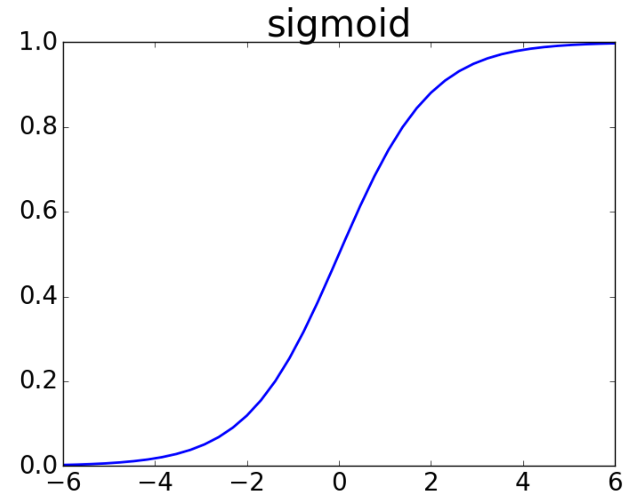
$$h_\theta(X) = \frac{1}{1 + e^{-\theta^T X}}$$

Therefore,

$$P(class = 1|x;\theta) = h_\theta(X)$$

$$P(class = 0|x;\theta) = 1 - h_\theta(X)$$

$$g(z) = \frac{1}{1 + e^{-z}}$$



The output of sigmoid function could be used to indicate the probability.

Logistic Regression

Logistic Regression Model

$$P(class = 1|x;\theta) = h_\theta(X)$$

$$P(class = 0|x;\theta) = 1 - h_\theta(X)$$

$$P(class = y|x;\theta) = h_\theta(X)^y\big(1 - h_\theta(X)\big)^{1-y}$$

Considering all the given data (training set):

$X = [x_1, \cdots, x_n]$, $Y = [y_1, \cdots, y_n]$,

$$L(\theta) = \prod_i^n h_\theta(x_i)^{y_i}\big(1 - h_\theta(x_i)\big)^{1-y_i}$$

The cost function： $J = -\frac{1}{n}\log\left(L(\theta)\right)$

Logistic Regression

Conclusion

- <u>Logistic regression</u>
  - Uses sigmoid and log function and to estimate the parameters
  - According to the Maximum Likelihood Estimate, construct the loss function:

$$J = -\frac{1}{m}\log\left(L(\theta)\right)$$

where,

$$L(\theta) = \prod_{i}^{n} h_\theta(x_i)^{y_i}\left(1 - h_\theta(x_i)\right)^{1-y_i}$$

  - Minimize the cost:

$$\frac{\partial J}{\partial \theta} = 0 \quad \rightarrow \quad \text{solve for } \theta \qquad \text{HOW?}$$

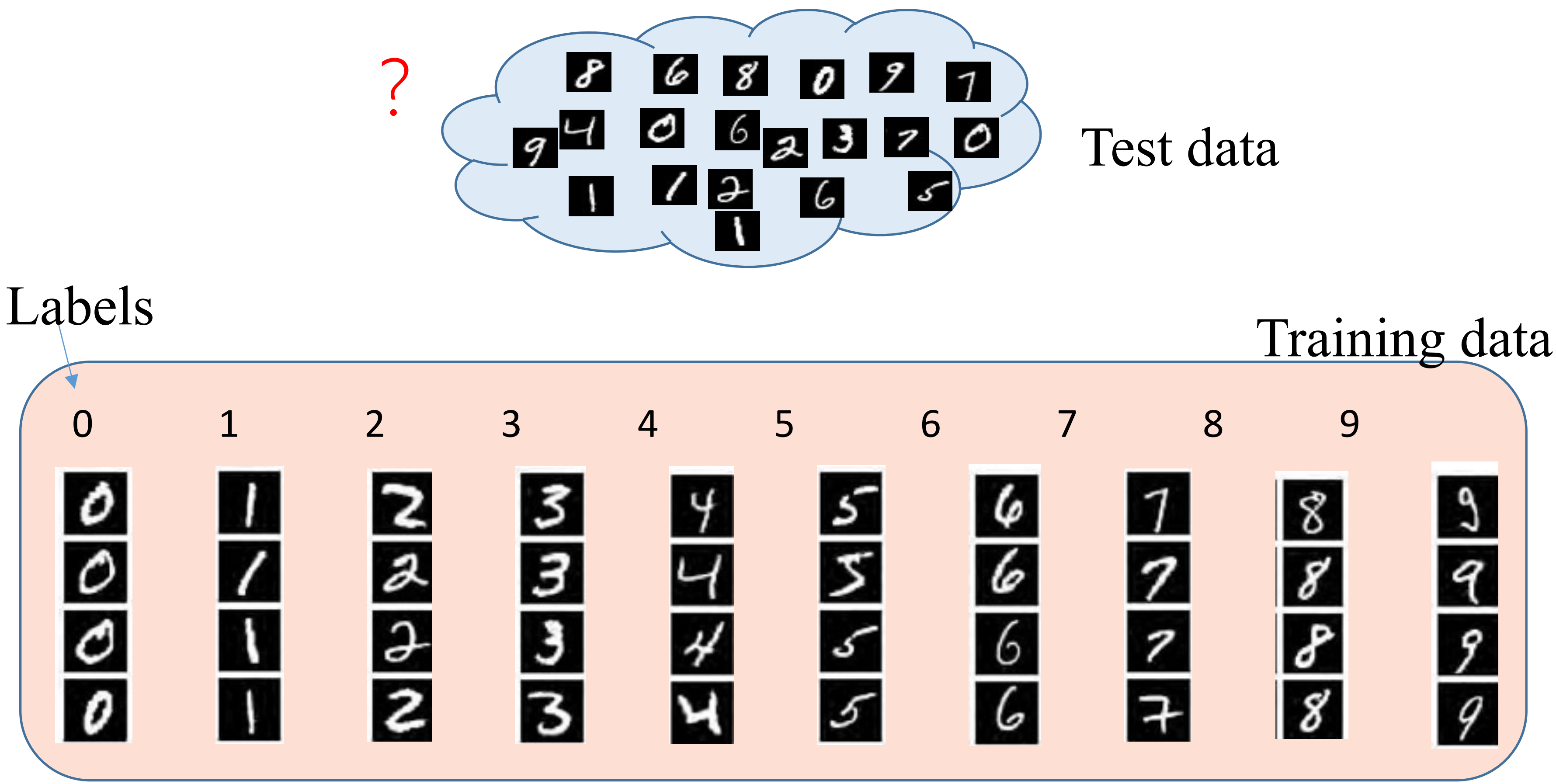
Try to solve it by yourself.

Supervised learning

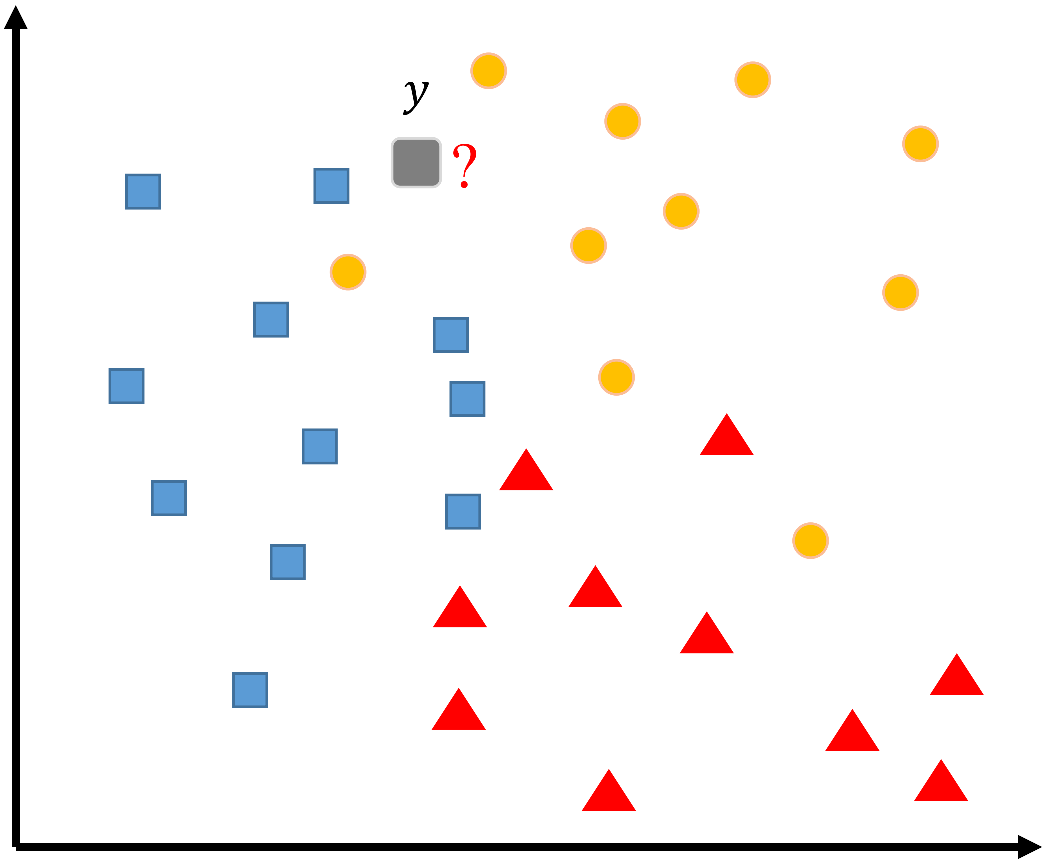
- Linear Regression
- Logistic Regression
- *Classification*
  - *Distance-based algorithms*
  - Linear classifiers
  - Other classifiers
- ……

Classification

Multi-class classification assigns test samples to a certain class.



Classification



Training data:

$X = \{x^{(1)}, x^{(2)}, \dots, x^{(N)}\}$

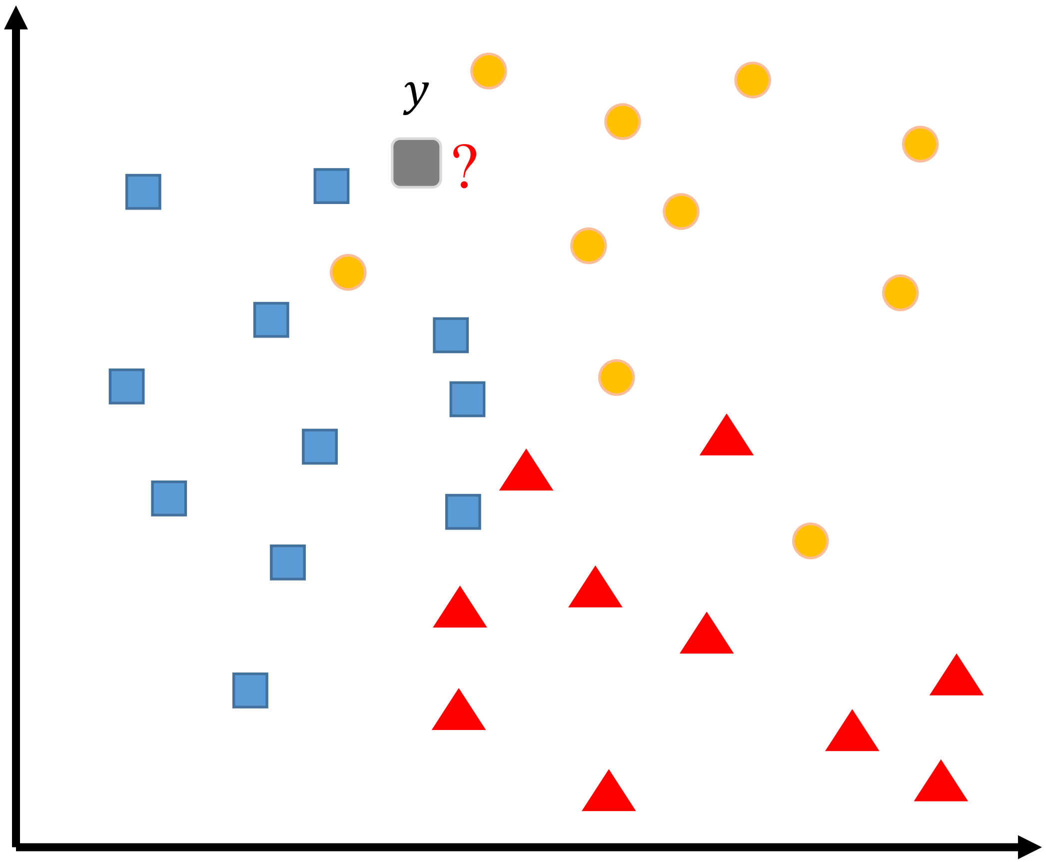
and
training labels:

$L = \{l^{(1)}, l^{(2)}, \dots, l^{(N)}\}$

N: *the number of training data*

Classification

- Nearest neighbor



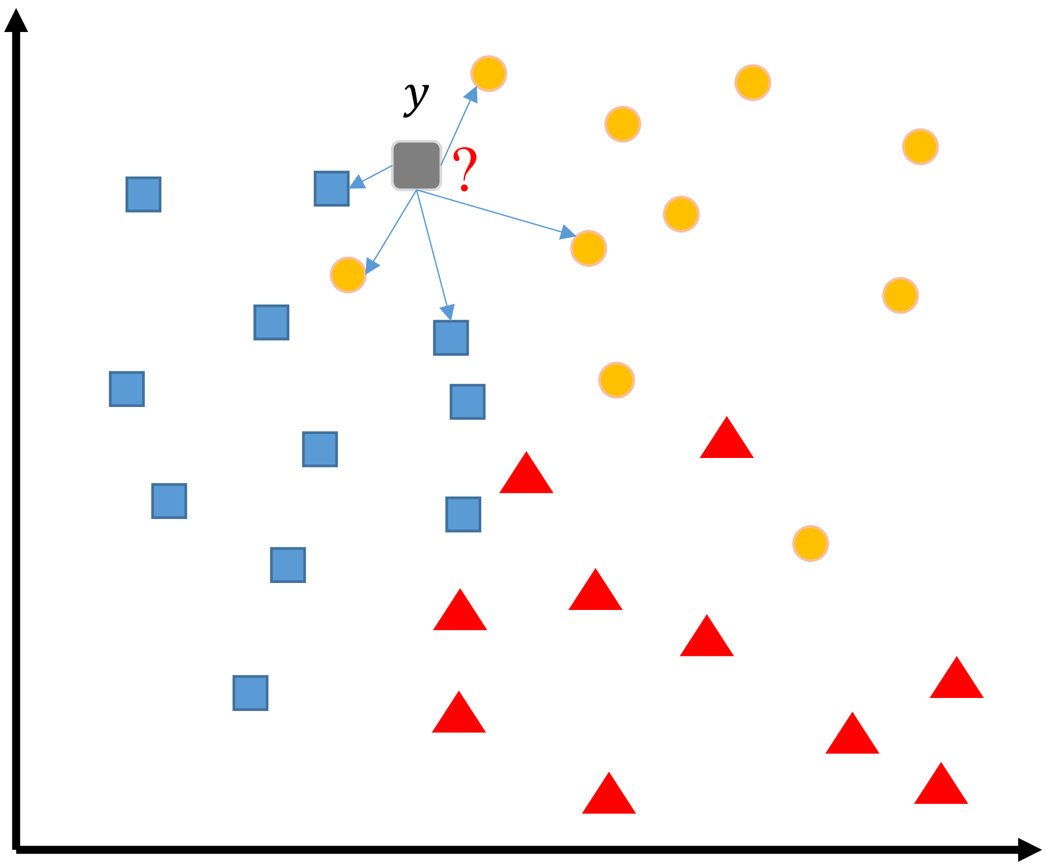
How to decide which is the nearest one?

The distance $d(\mathbf{x}, \mathbf{y})$ between two points $\mathbf{x} \in R^n$ and $\mathbf{y} \in R^n$ can for example be measured by the Euclidean distance.

$$d\left(x^{(1)}, x^{(2)}\right) = \sqrt{\sum_{i=1}^{n}\left(x_i^{(1)} - x_i^{(2)}\right)^2}$$

Classification

Nearest neighbor



How to decide which is the nearest

$$d^j(x^{(y)}, y) = \sqrt{\sum_{i=1}^{n}\left(x_i^{(j)} - y\right)^2}$$

Calculate all the distances from the training data to the test data y, and we obtain:

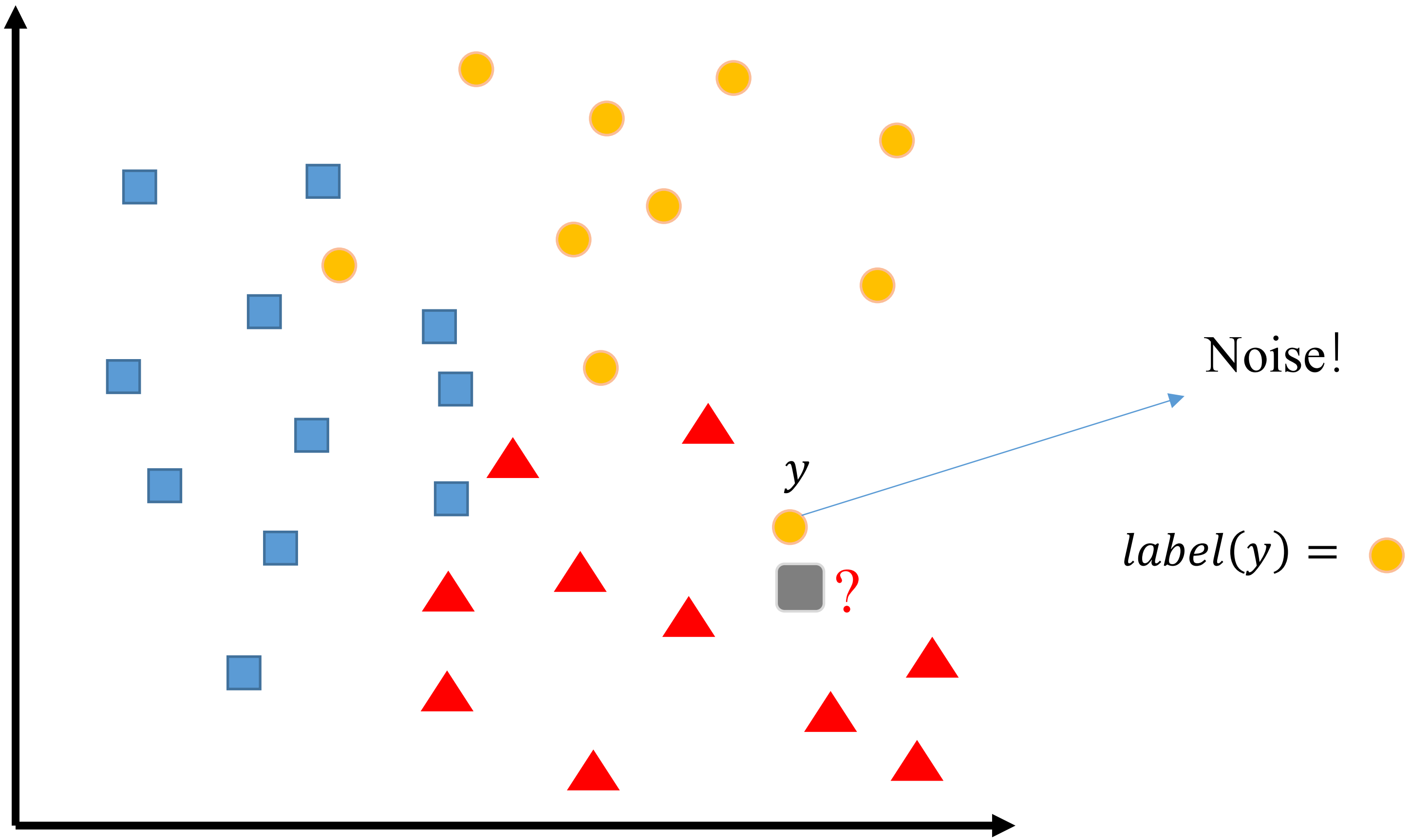
$$D = [d^{(1)}, d^{(2)}, \ldots, d^{(N)}]$$

$$s = argmin_i \, d^{(i)}$$

$label(y) = label(x^{(s)})=$ 

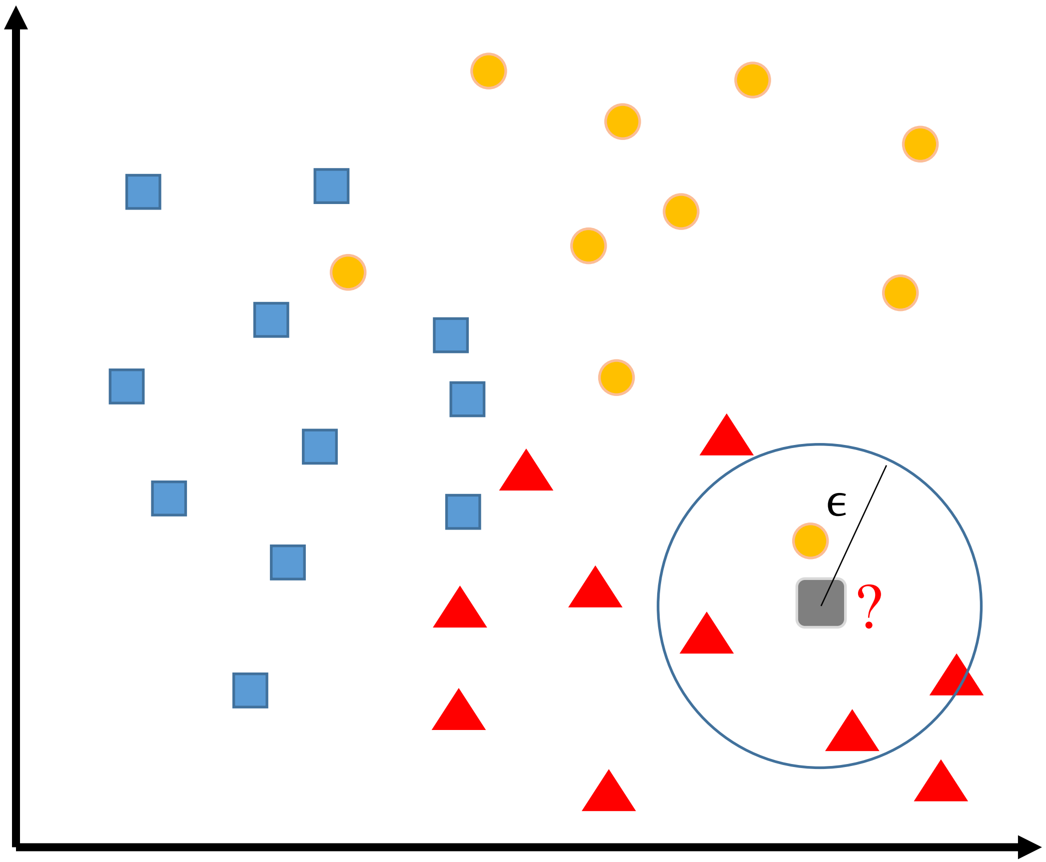
Classification

- Nearest neighbor



Classification

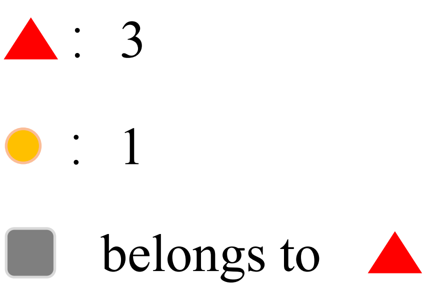
ϵ-ball Nearest neighbor



Select a value ϵ, then draw a ball in R^n with y as the center and ϵ as the radius.

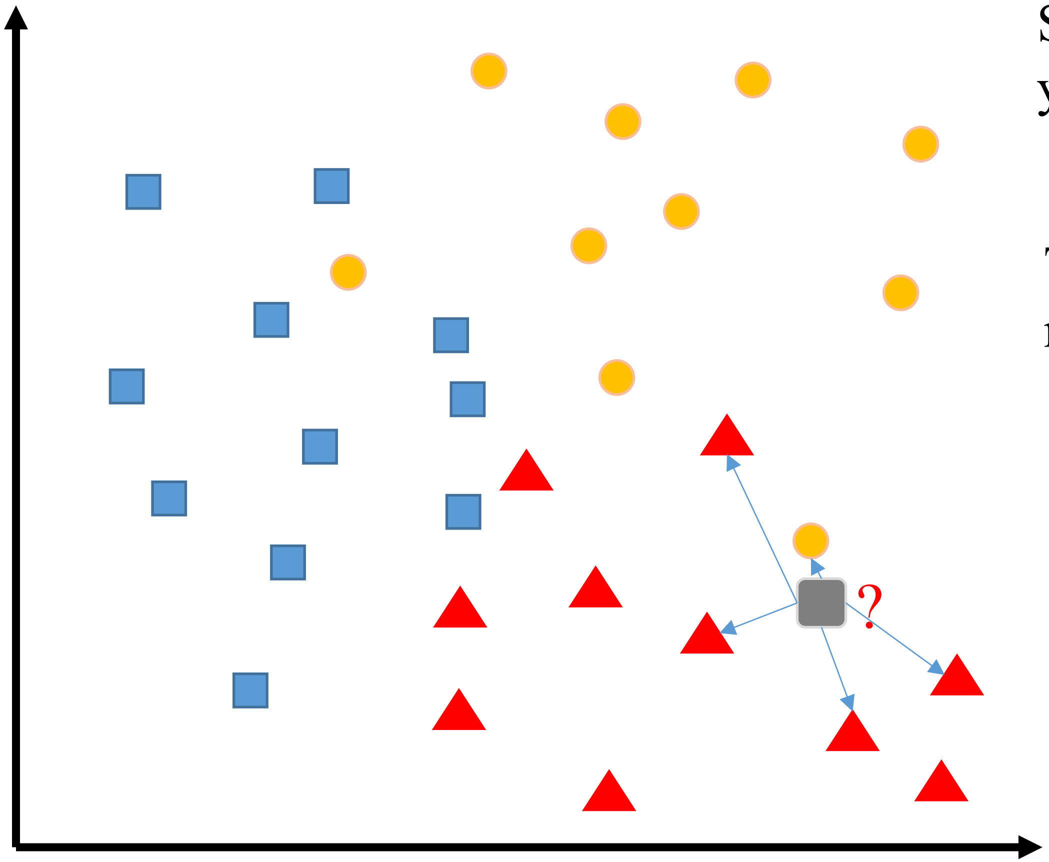
The label of y is decided by majority labels of points in this ball.

In this ball:



Classification

- K Nearest neighbor



Select a value *k*, then find y's k nearest neighbor.

The label of y is decided by majority labels of y's k neighbors.

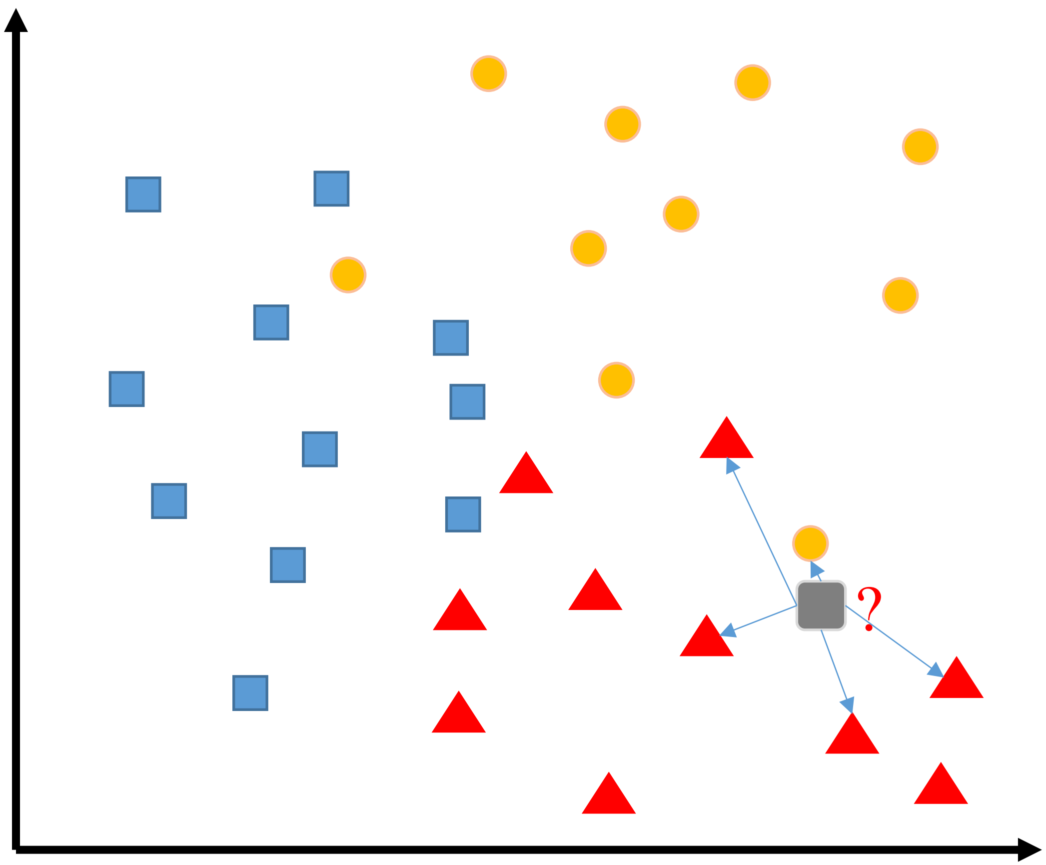
Let k be 5,

▲ : 5 ● : 1

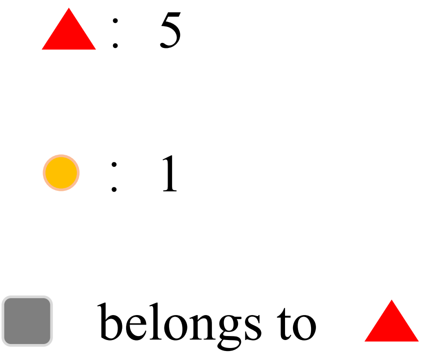
■ belongs to ▲

Classification

- K Nearest neighbor



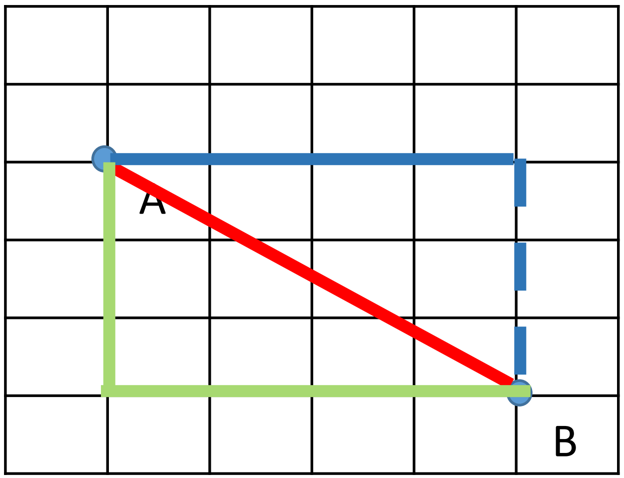
Question:
How to decide k?
Which algorithm achieve better performance?



Classification

Distance Metrics

- Euclidean distance
- $d_e(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
- Sum of squared distance
- $d_q(x, y) = \sum_{i=1}^{n}(x_i - y_i)^2$
- Manhattan distance
- $d_m(x, y) = \sum_{i=1}^{n}|x_i - y_i|$
- Chebyshev distance
- $d_c(x, y) \max_{i=1,\cdots,n} |x_i - y_i|$



Classification

- Nearest neighbor classifier

Problem:

- Need to determine value of parameter K
- Distance based learning is not clear which type of distance to use and which attribute to use to produce the best results.
- Computation cost is quite high because we need to compute distance of each query instance to all training samples.

Classification

Example

- Each image is represented by a vector of dimension 784.

The matrix indicates the pairwise distances.

0	2.8735	2.1766	2.6559	2.2201	2.2500	2.0893	2.4795	2.8443	2.1202
2.8735	0	2.5055	2.8681	2.9475	2.6062	2.8493	2.8330	2.9434	3.1619
2.1766	2.5055	0	2.9024	2.3556	0.7858	2.3561	2.2060	2.5274	2.4331
2.6559	2.8681	2.9024	0	2.7428	2.9531	3.0539	2.8362	2.8488	2.6425
2.2201	2.9475	2.3556	2.7428	0	2.5284	2.1733	2.4262	2.3432	2.5895
2.2500	2.6062	0.7858	2.9531	2.5284	0	2.4679	2.2906	2.5549	2.3900
2.0893	2.8493	2.3561	3.0539	2.1733	2.4679	0	2.5580	2.7456	2.3759
2.4795	2.8330	2.2060	2.8362	2.4262	2.2906	2.5580	0	2.8885	2.5823
2.8443	2.9434	2.5274	2.8488	2.3432	2.5549	2.7456	2.8885	0	2.9773
2.1202	3.1619	2.4331	2.6425	2.5895	2.3900	2.3759	2.5823	2.9773	0

The distance between the data is inconsistent with similarity of the content of the image .

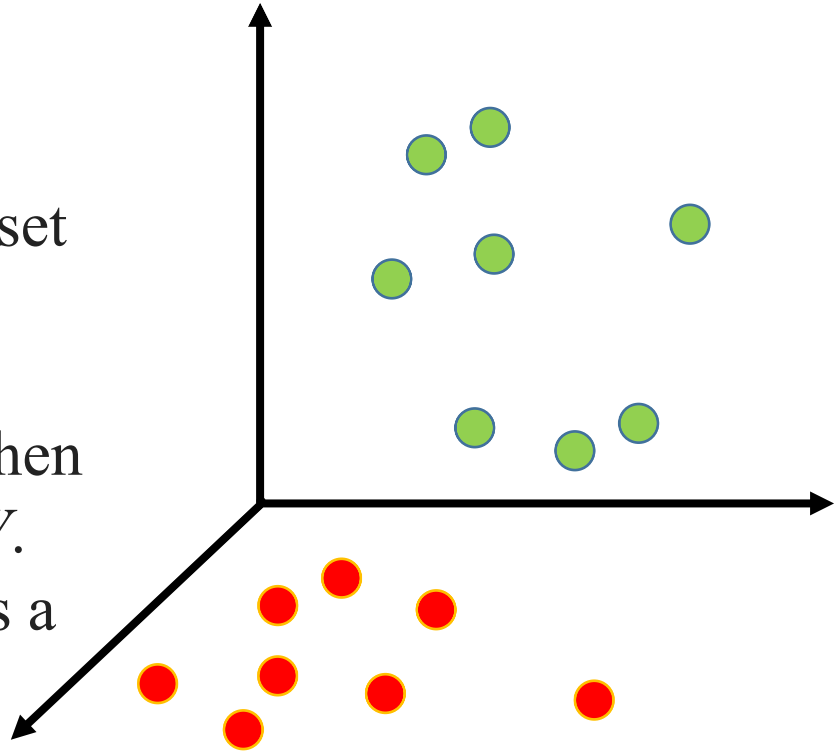
Classification

- Nearest subspace classifier

What is subspace?

Let K be a field (such as the real numbers), V be a vector space over K, and let W be a subset of V. Then W is a **subspace** if:

1. The zero vector, $\mathbf{0}$, is in W.
2. If $\mathbf{u}$ and $\mathbf{v}$ are elements of W, then the sum $\mathbf{u} + \mathbf{v}$ is an element of W.
3. If $\mathbf{u}$ is an element of W and c is a scalar from K, then the scalar product $c\mathbf{u}$ is an element of W.

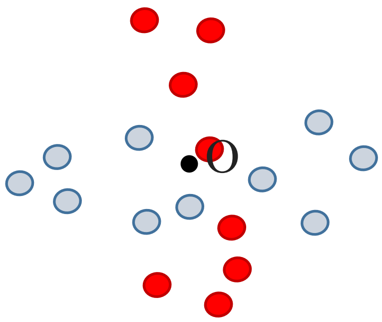


Classification

- Nearest subspace classifier

Assume that data in R^n which belong to the same class lie on the same subspace of R^n

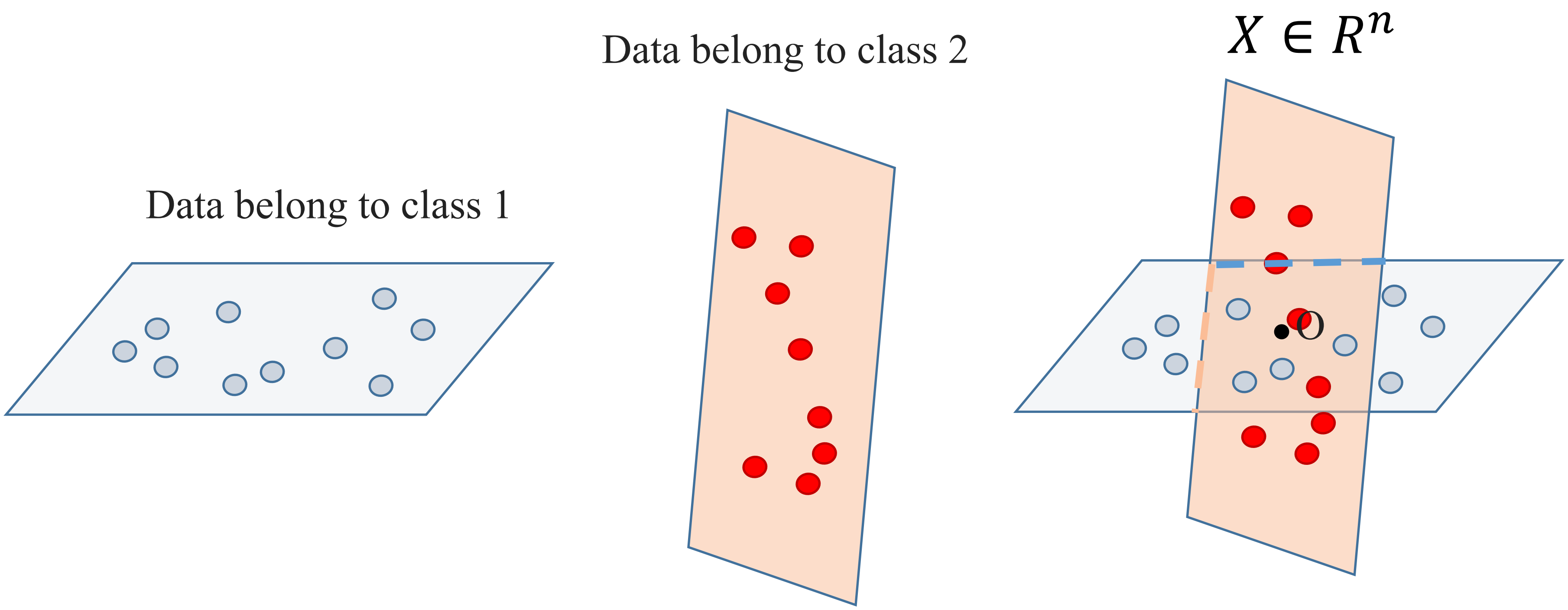
$$X \in R^n$$



Classification

Nearest subspace classifier

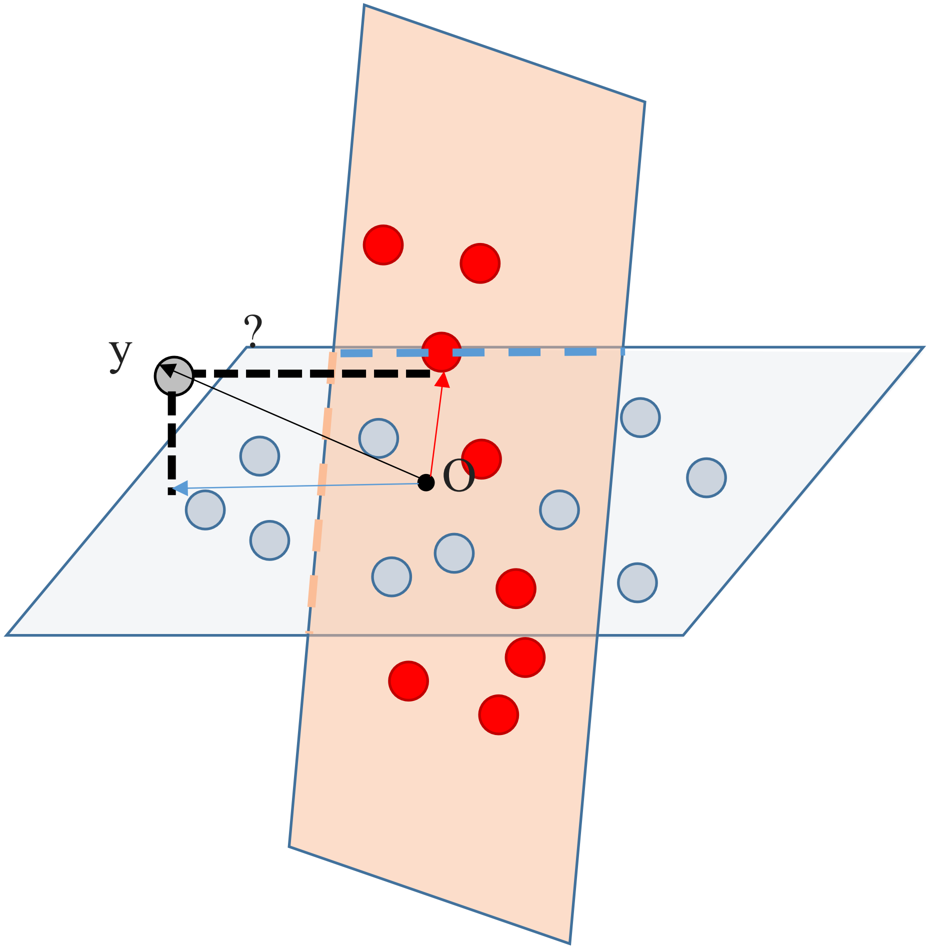
Assume that data in R^n which belong to the same class lie on the same subspace of R^n



Classification

- Nearest subspace classifier

Assume that data points in each class lie in the same subspace, nearest subspace classifier assign the given data to the class whose related subspace is nearest.

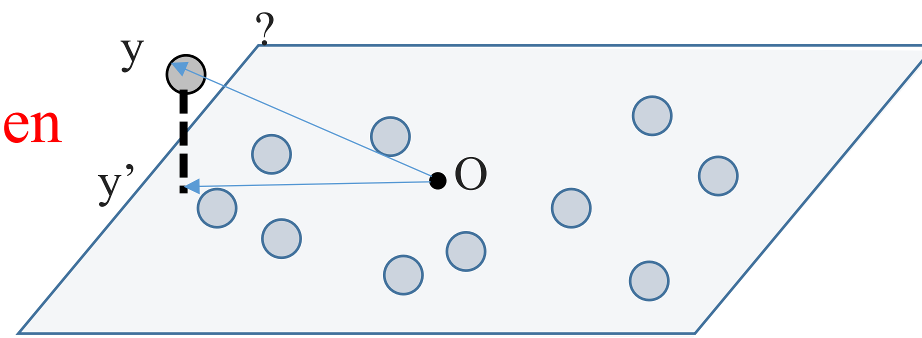


Classification

- Nearest subspace classifier

Assume that data in R^n which belong to the same class lie on the same subspace of R^n

How to calculate the distance between a point and a certain subspace?



The test sample $y \in R^n$ can be represented by the give data $X \in R^m$, which is a subspace of R^n. The distance between y and the subspace R^m can be calculated as the reconstruction error:

$$d_{NS} = \|y - Xa\|_2$$

where a is the coefficient of representing y by X linearly.

Classification

- Nearest subspace classifier

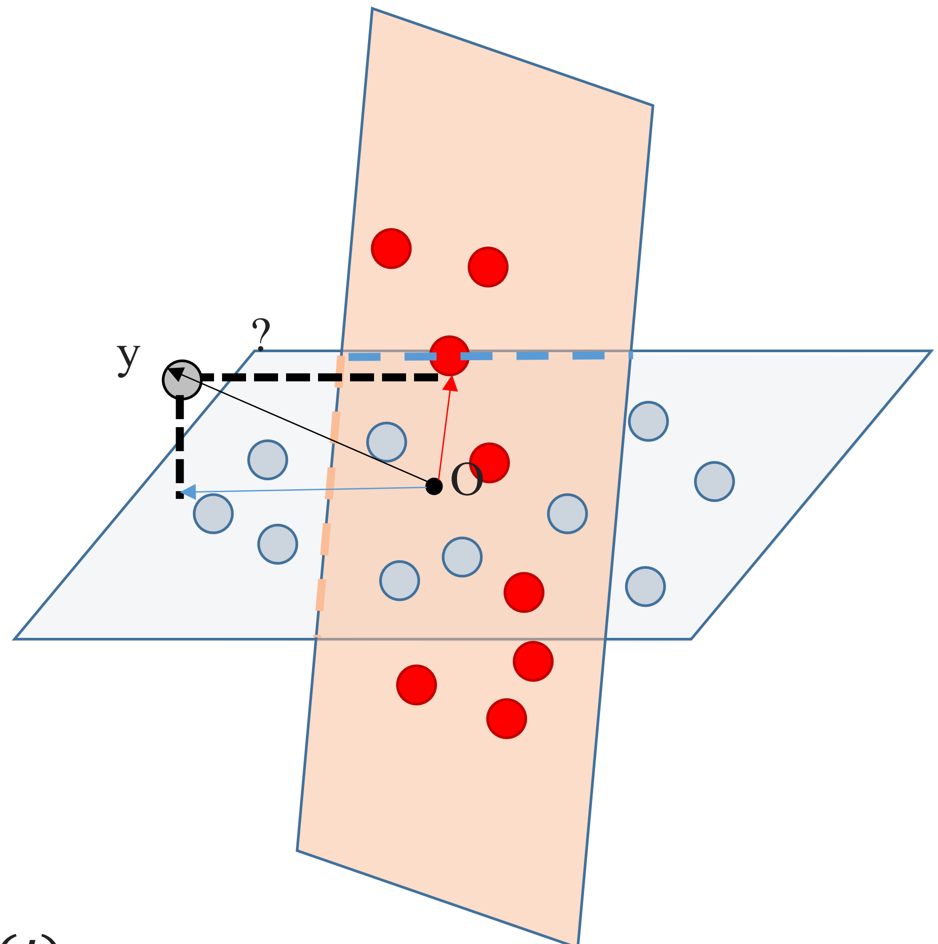
Therefore, the algorithm of nearest subspace classifier is described as:

1. Calculate the distances from y to each subspace composed by data points that belong to different class.

$$d_{NS}(i) = \|y - X_i a_i\|_2$$

2. Find the smallest distance, and assign y to the related class.

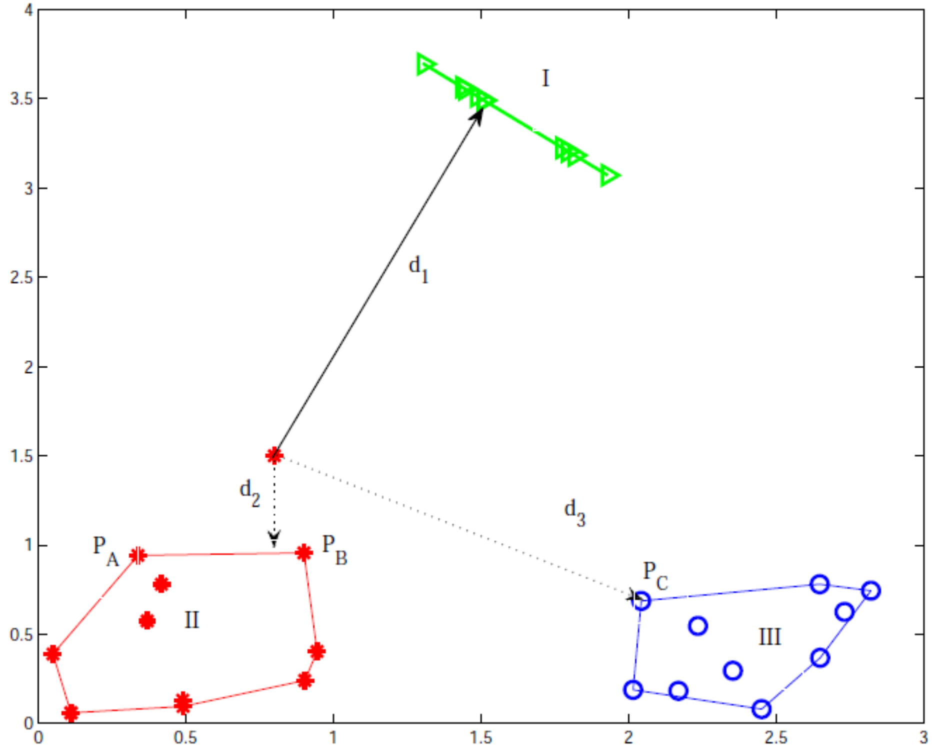
$$classify(y) = \text{argmin}_i \, d_{NS}(i)$$



Classification

Other distance based algorithm

Some other distance based methods use different similarity measurement.



- e.g. Nearest convex hull classifier