01 INTRODUCTION

Fall 2020

CS5439 Machine Learning

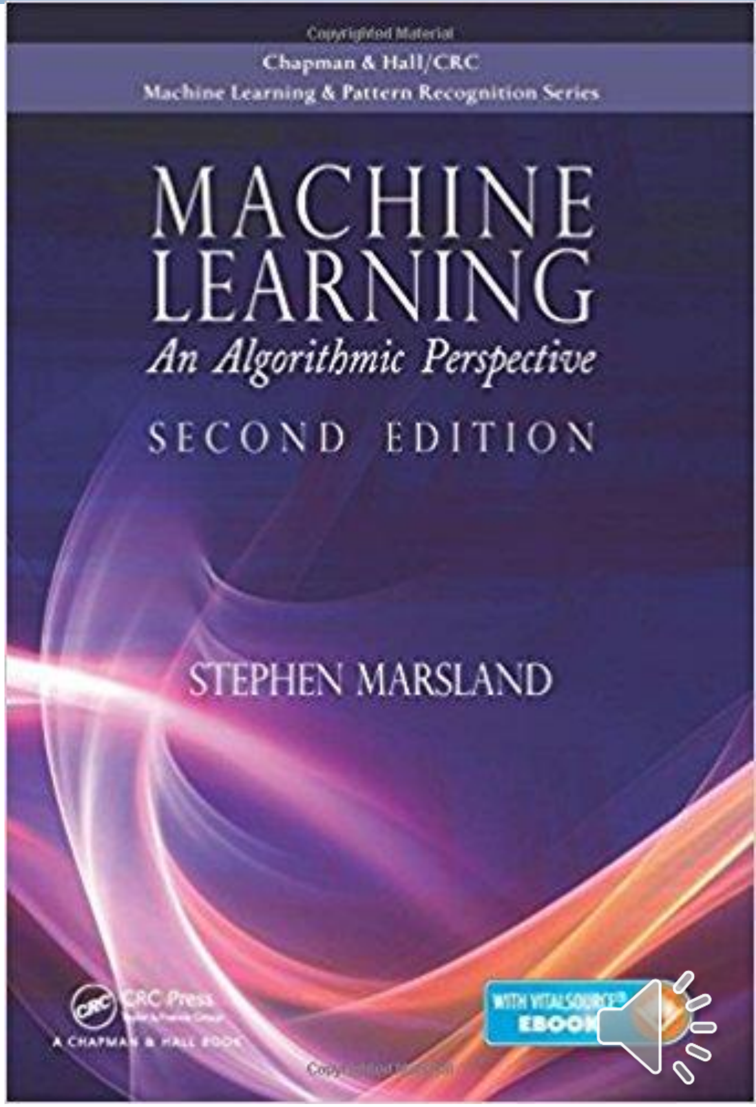
B1:

Machine learning: an algorithmic perspective.

2nd Edition

Marsland, Stephen.

CRC press, 2015.



Credits

1. B1
2. https://en.wikipedia.org/wiki/Curse_of_dimensionality
3. Keogh, Eamonn, and Abdullah Mueen. "Curse of Dimensionality." *Encyclopedia of Machine Learning*. Springer US, 2011. 257-258.
4. https://en.wikipedia.org/wiki/Precision_and_recall
5. http://scott.fortmann-roe.com/docs/BiasVariance.html
6. https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff
7. http://homepages.cae.wisc.edu/~ece539/project/s01/qi.ppt
8. https://medium.com/@UdacityINDIA/difference-between-machine-learning-deep-learning-and-artificial-intelligence-e9073d43a4c3

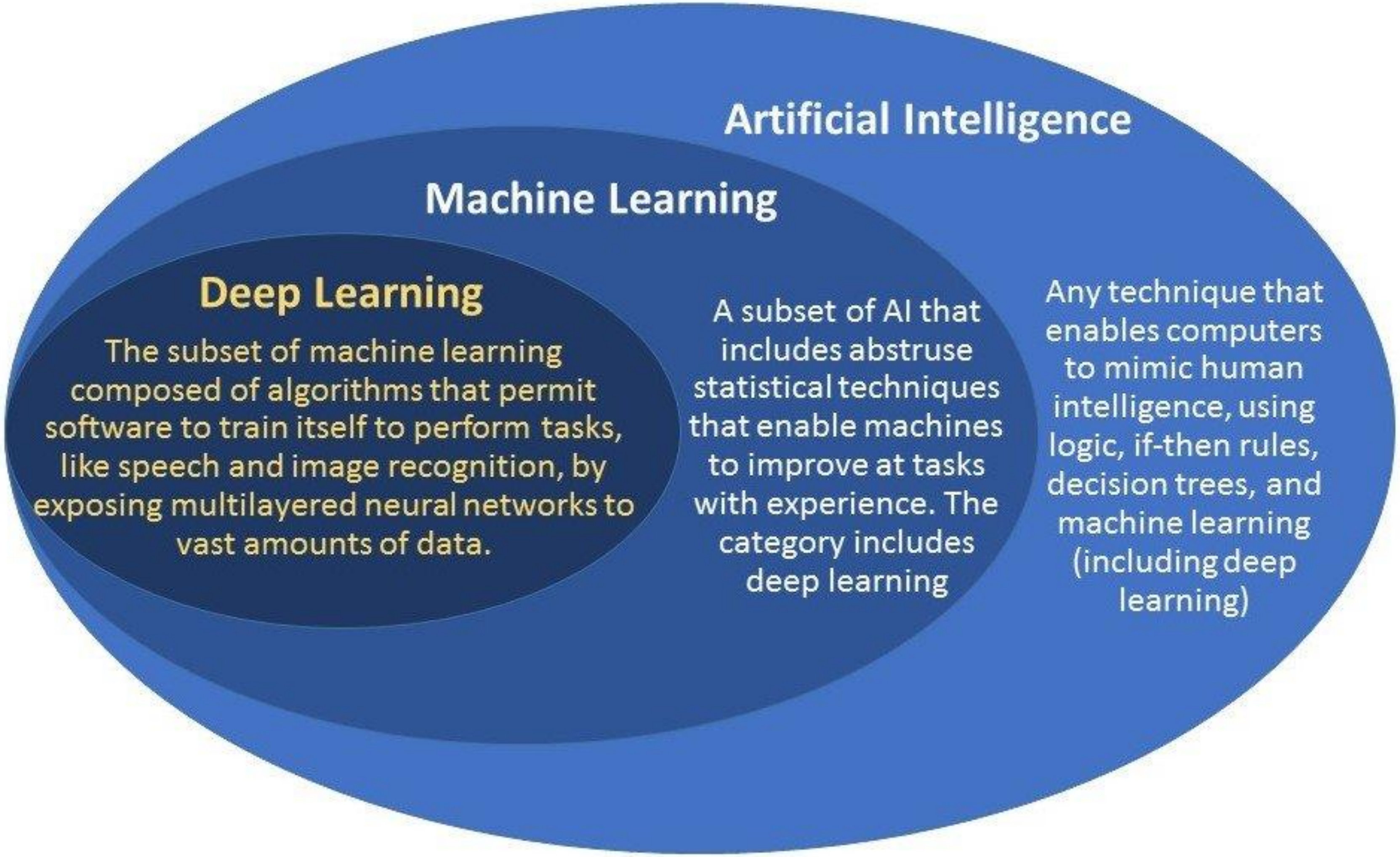
Assignment

Read:

B1: Ch1, Ch 2

Problems:

AI vs. Machine Learning vs. Deep Learning



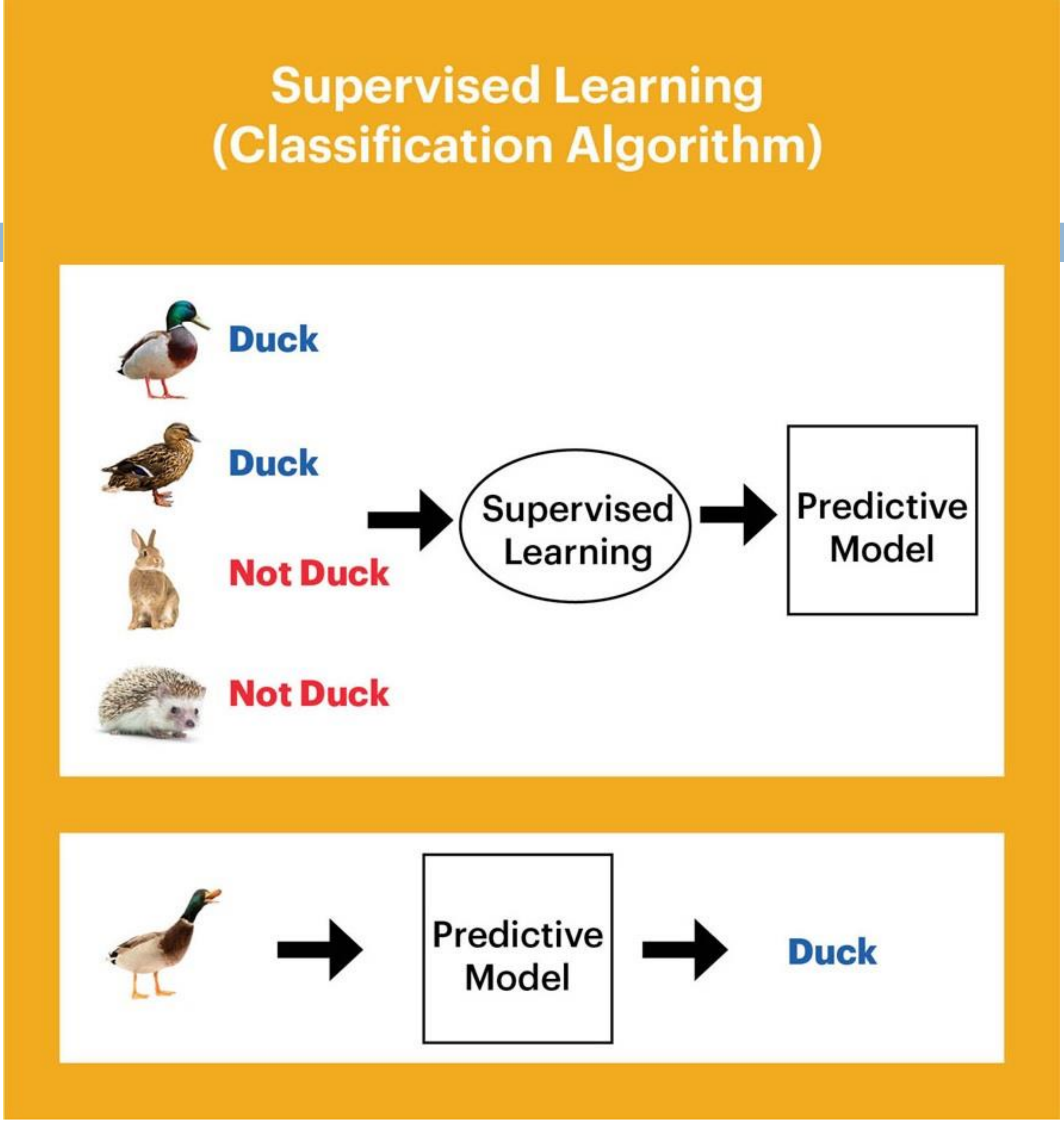
Credit: *https://www.geospatialworld.net/blogs/difference-between-ai%EF%BB%BF-machine-learning-and-deep-learning/*

Machine Learning

- Computer algorithms that allow computer programs to automatically improve through experience…
  - …without explicit programming for the problem at hand.

Types of Machine Learning

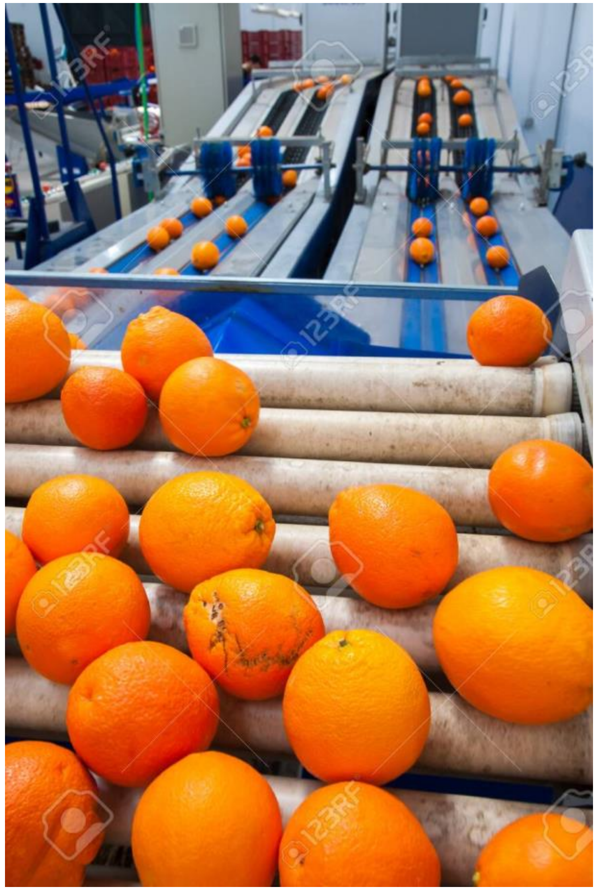
- Supervised learning
- Unsupervised learning
- Reinforcement learning
- Evolutionary learning



Credit: https://learncuriously.wordpress.com/2018/12/22/machine-learning-vs-human-learning-part-1/

A Supervised Learning Example

- Apples vs. Oragnes



Helpful instruments and attributes!

- Camera
  - Color, Shape
- Weighing machine
- Freshness: pH sensor, Moisture sensor, and Gas sensor

  (*http://www.ijsce.org/wp-content/uploads/papers/v8i3/C3146078318.pdf*)
- Texture profile analysis: using visible and near infrared hyperspectral imaging

  https://pubmed.ncbi.nlm.nih.gov/24128497/
- ..and more???

	Apple	Orange
Redness (from Color) [0,1] : 1-> completely red	More towards red	More towards orange/yellow
Roughness (from texture) [0,1]: 0-> completely smooth	Smooth surface	Rough surface
Moisture [0:1] : 1-> 100% water	Less % of water	High % of water

If *redness* $>$ .85

And If *roughness* $<$ .1

And If moisture $<$ .4

Then Output "Apple"

Else Output "Orange"

$$\alpha = .85$$

$$\beta = .1$$

$$\gamma = .4$$

If *redness* $> \alpha$ $\quad \alpha = .85$

And If *roughness* $< \beta$ $\quad \beta = .1$

And If moisture $< \gamma$ $\quad \gamma = .4$

Then Output "Apple"

Else Output "Orange"

Consider the following sample

Redness	Roughness	Moisture	Fruit type
.9	.08	.35	Apple

✓

If *redness* $> \alpha$ $\alpha = .85$

And If *roughness* $< \beta$ $\beta = .1$

And If moisture $< \gamma$ $\gamma = .4$

Then Output "Apple"

Else Output "Orange"

Need to update threshold parameter $\beta = .1$, need to push towards .15

$$\Delta\beta \leftarrow \text{a fraction of } (\beta_{\text{expected}} - \beta_{\text{existing}})$$

$$\Delta\beta \leftarrow \mu\,(\beta_{\text{expected}} - \beta_{\text{existing}})$$

$$(\mu = .3, \text{learning rate})$$

$$\beta_{\text{new}} \leftarrow \beta_{\text{existing}} + \mu\left(\beta_{\text{expected}} - \beta_{\text{existing}}\right)$$

$$= .1 + .3(.15 - .1) = .115$$

Consider the following sample

Redness	Roughness	Moisture	Fruit type
.91	.15	.31	Apple

✗

If *redness* $>\alpha$ $\alpha = .85$

And If *roughness* $< \beta$ $\beta = .115$

And If moisture $< \gamma$ $\gamma = .4$

Then Output "Apple"

Else Output "Orange"

Need to update threshold parameter $\alpha = .85$, need to push towards .7

$$\Delta\alpha \leftarrow \text{a fraction of } (\alpha_{\text{expected}} - \alpha_{\text{existing}})$$

$$\Delta\alpha \leftarrow \mu\,(\alpha_{\text{expected}} - \alpha_{\text{existing}})$$

$$(\mu = .3, \text{learning rate})$$

$$\alpha_{\text{new}} \leftarrow \alpha_{\text{existing}} + \mu\left(\alpha_{\text{expected}} - \alpha_{\text{existing}}\right)$$

$$= .85 + .3(.7 - .85) = .805$$

…and so on.

Consider the following sample

Redness	Roughness	Moisture	Fruit type
.7	.11	.33	Apple

- Features
- Algorithm
- Parameters
  - Hyper (μ)
  - Model (α, β, γ)
- Model = Algorithms + Trained Parameters
- Training, Testing

- The algorithm varies across different machine learning techniques

$$f(\alpha, \beta, \gamma) = \frac{\sin\alpha + \cos\frac{\beta}{2}}{e^{\gamma}}$$

If $f(\alpha, \beta, \gamma) \geq .3$

Then Output "Apple"

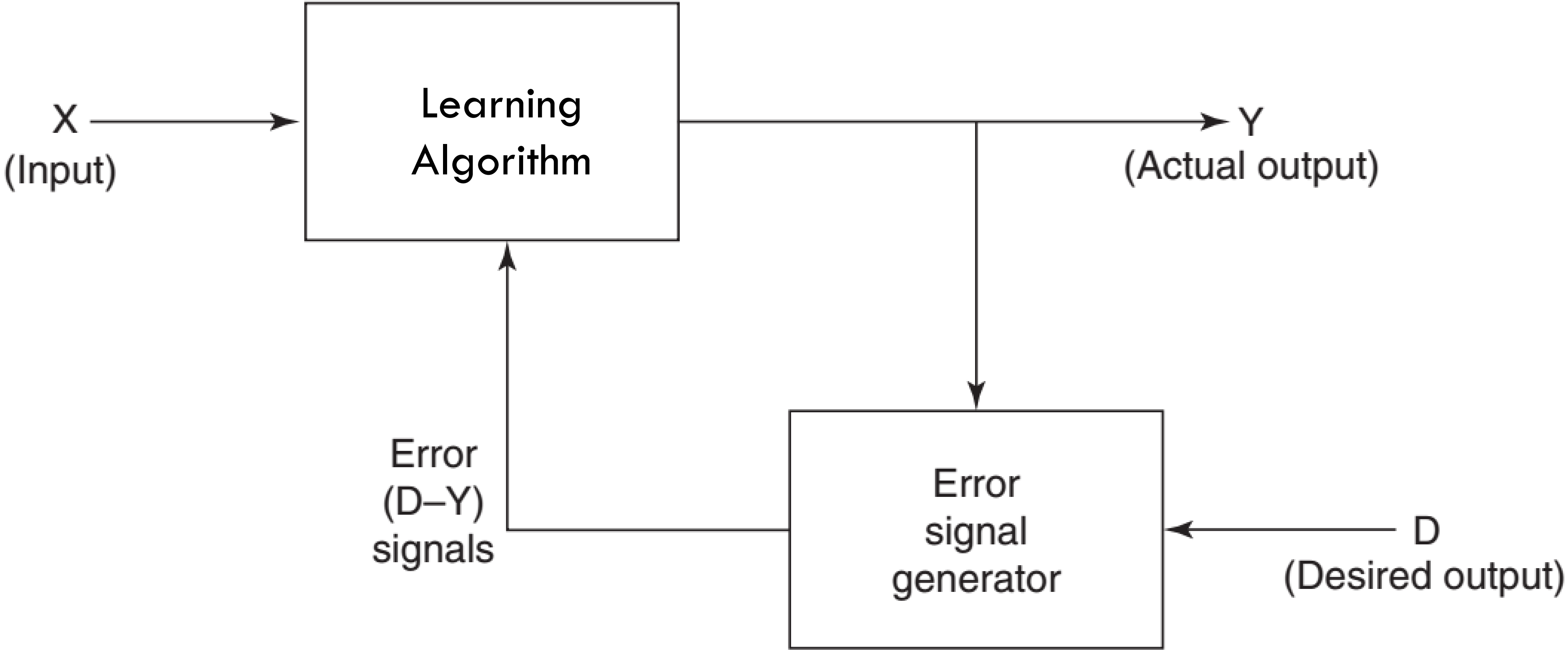
Else Output "Orange"

The above algorithm may not make sense, but it is an algorithm nonetheless.

Key Terms

- Features
- Algorithm
- Parameter
  - Hyper parameter
  - Model parameter
- Model
- Training, Testing

Supervised learning



Classification Problem

x_1	x_2	Class
0.1	1	1
0.15	0.2	2
0.48	0.6	3
0.1	0.6	1
0.2	0.15	2
0.5	0.55	3
0.2	1	1
0.3	0.25	2
0.52	0.6	3
0.3	0.6	1
0.4	0.2	2
0.52	0.5	3

x_1 *and* x_2 are called "features"

Class is the output

- Unsupervised learning
  - Correct responses are not provided, but instead the algorithm tries to identify similarities between the inputs so that inputs that have something in common are categorized together.

An Un-Supervised Learning Example

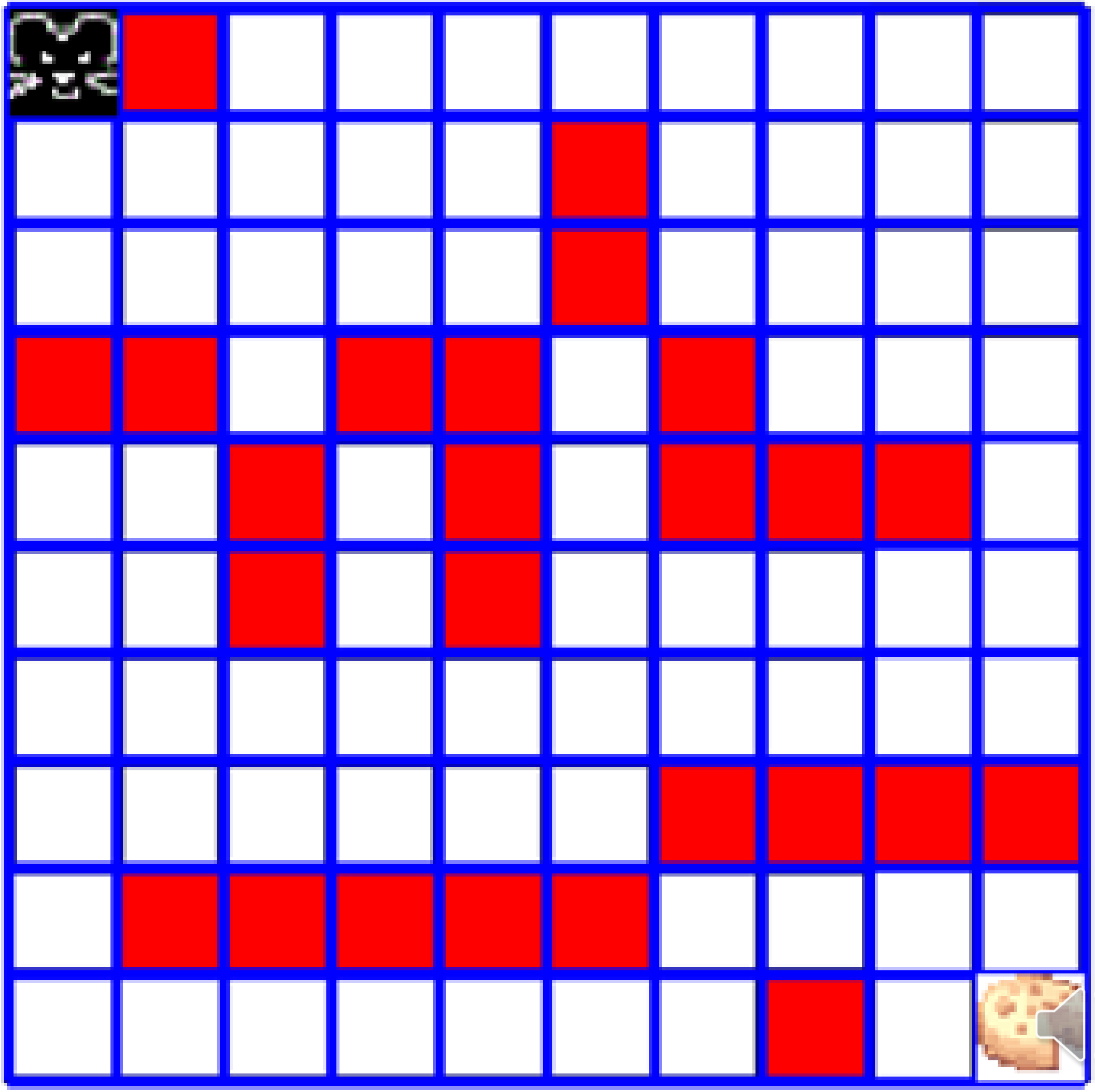
- Good apples
  - Very fresh, the best of the lot
- Average apples
  - Averagely fresh
- Below average
  - About to go bad, slightly damaged

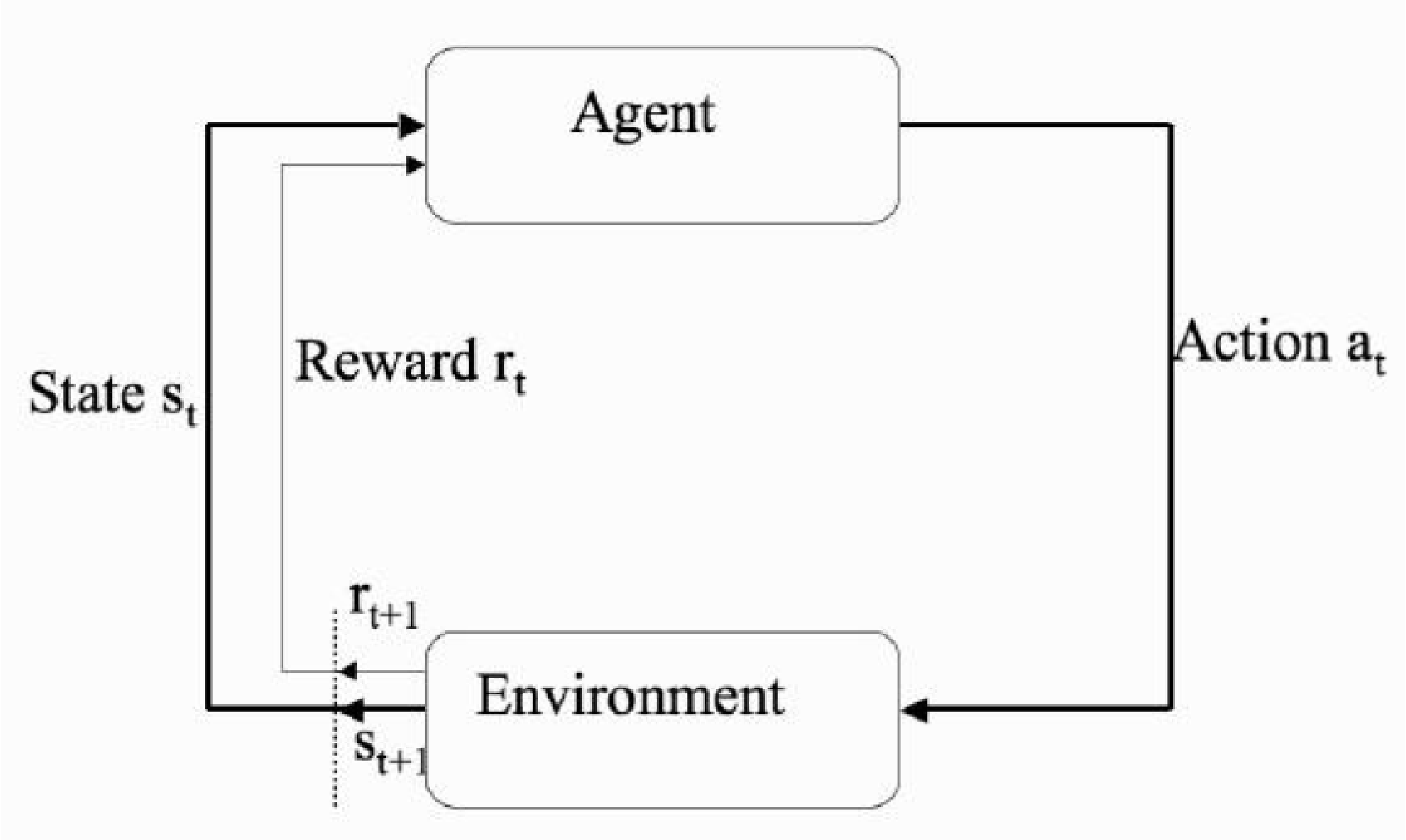
Reinforcement Learning

- Somewhere between supervised and unsupervised learning.
- The algorithm gets a “reward score” for each prediction (action), but does not get told how to better the score.
- It has to explore and try out different possibilities until it maximizes the reward score.

An Example

- Robot mouse has no idea of the room layout – obstructions and ways
- Robot mouse only knows how to recognize block with cheese – the end goal.



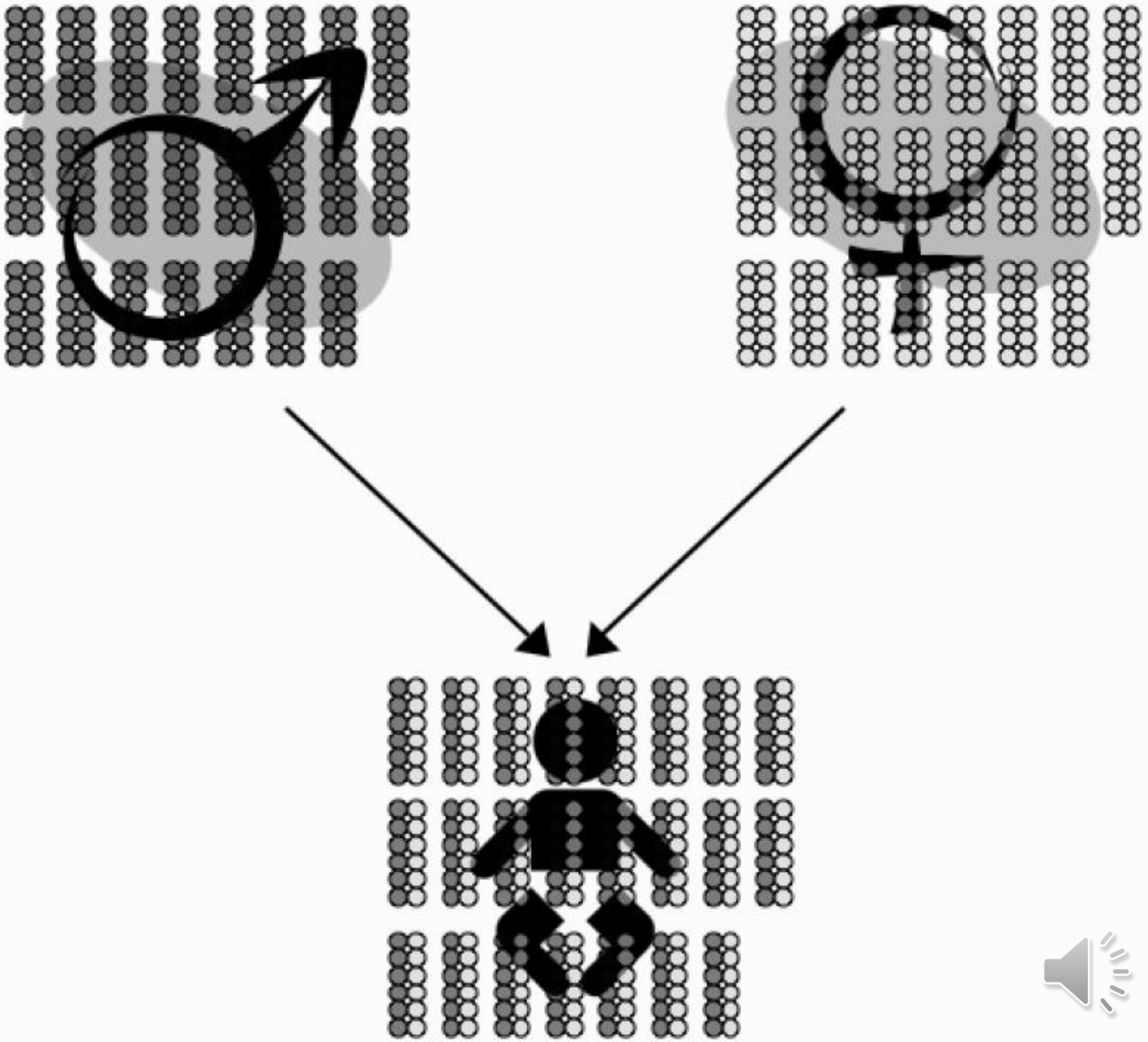


Agent
State s_t
Reward r_t
Action a_t
r_{t+1}
s_{t+1}
Environment

Evolutionary Learning

- Evolution as a search problem
- Competing animals and “Survival of the fittest”
  - “Fittest” animals
    - Live longer
    - Stronger
    - More attractive
  - Hence, they get more mates and produce more and “healthier” off springs

- Nature is biased towards “fitter” animals for sexual reproduction
  - Basis for Genetic Algorithms
- A child inherits chromosome from its parents
  - Survival of fittest means child can be better than its parents



Genetic Algorithm (GA)

Modelling a problem as a GA

- A method for representing solutions as chromosomes (or string of characters)
- A way to calculate the fitness of a solution
- A selection method to choose parents
- A way to generate offspring by breeding the parents

One generation

Select, Produce, Repeat!

Regression Problem

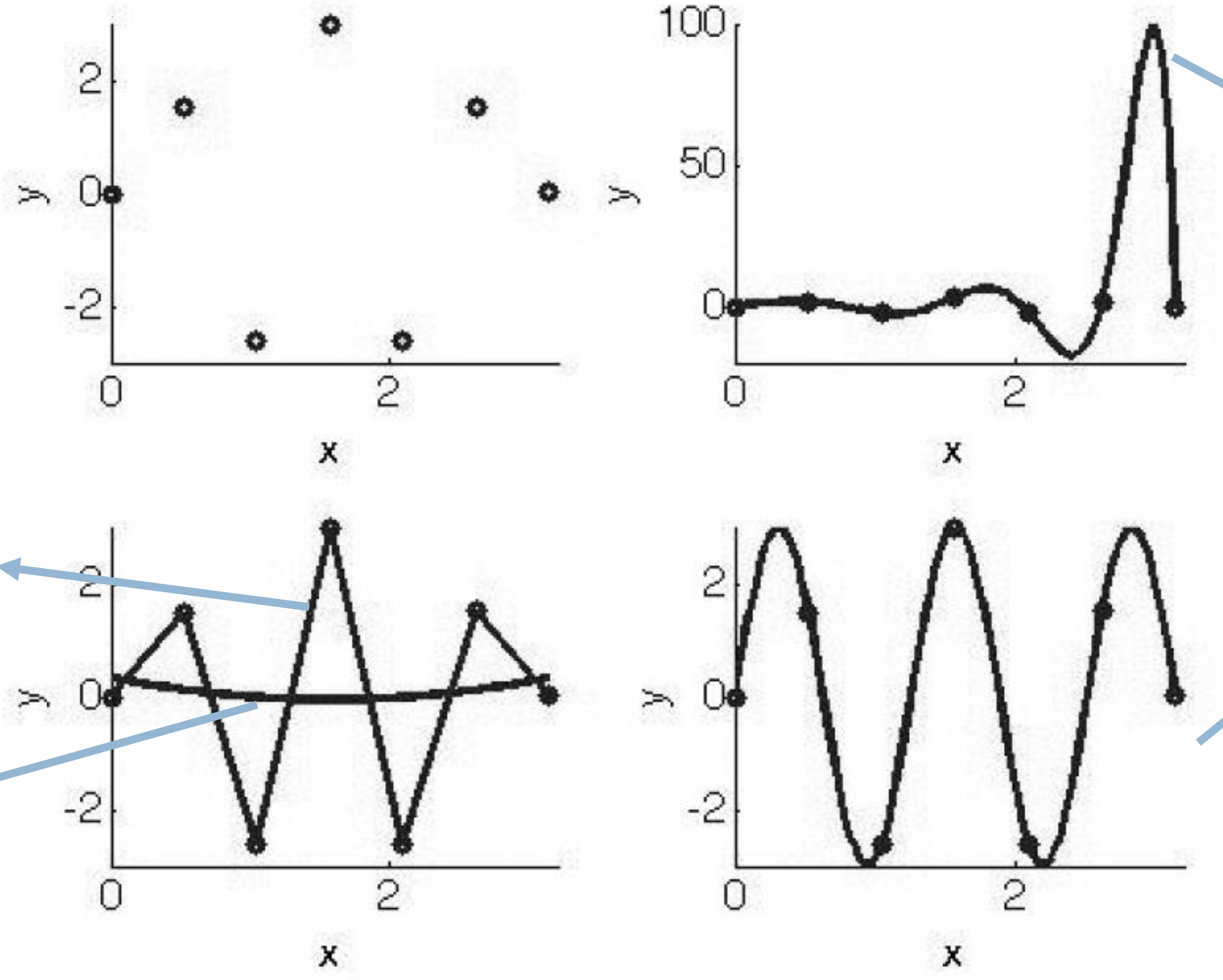
x	y
0	0
0.5236	1.5
1.0472	-2.5981
1.5708	3.0
2.0944	-2.5981
2.6180	1.5
3.1416	0

What is the value of y when $x = 0.44$?

Function Approximation

Points plotted in 2D

Goes through all data points but the spike looks out of place

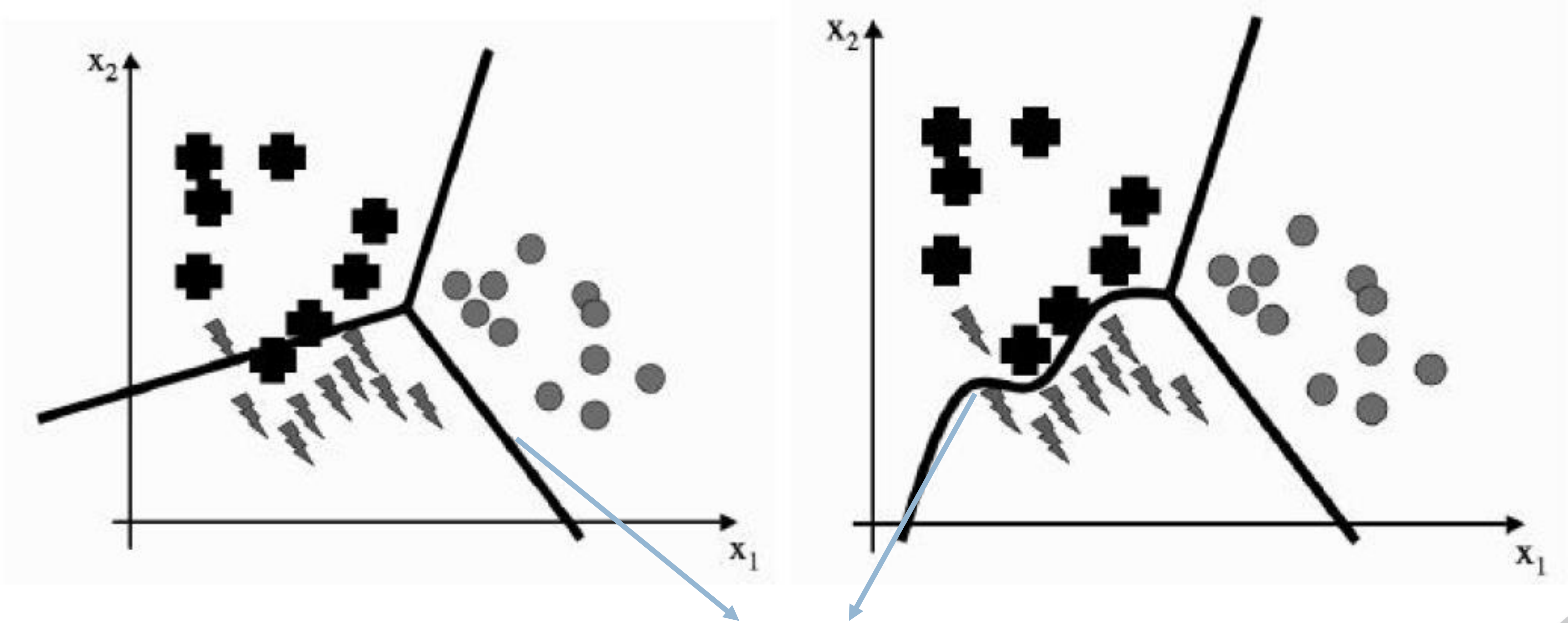


Joining with straight lines

Approximated using cubic function

Plotted using y = 3 sin(5x)

Classification Problem



Decision Boundaries

Machine Learning Process

1. Data Collection and Preparation
   - #Samples, error-free, etc.
2. Feature Selection
   - #features, which features to select, etc.
3. Algorithm Choice/Model Selection
   - Which algorithm to select
4. Parameters
   - Algorithms can be parametrized

5. Training
   - Generalizing model based on training dataset to predict outputs of unseen data
6. Evaluation
   - How good our trained model fares on unseen data

Input

Input is a vector of *n* dimensions

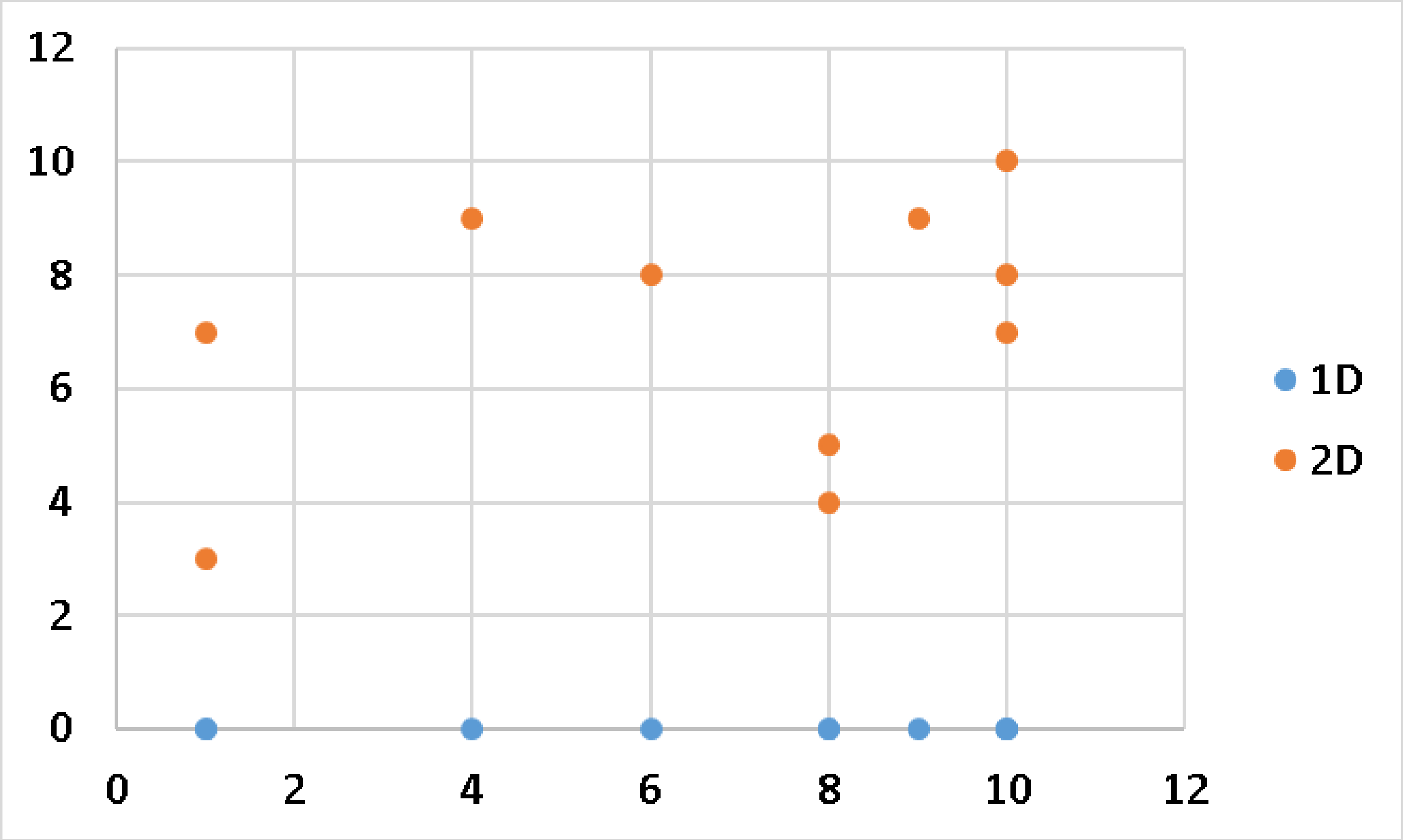
x $= \{x_0, x_1, \dots, x_{n-1}\}$

Sometimes, it is represented as a column vector.

Curse of Dimensionality

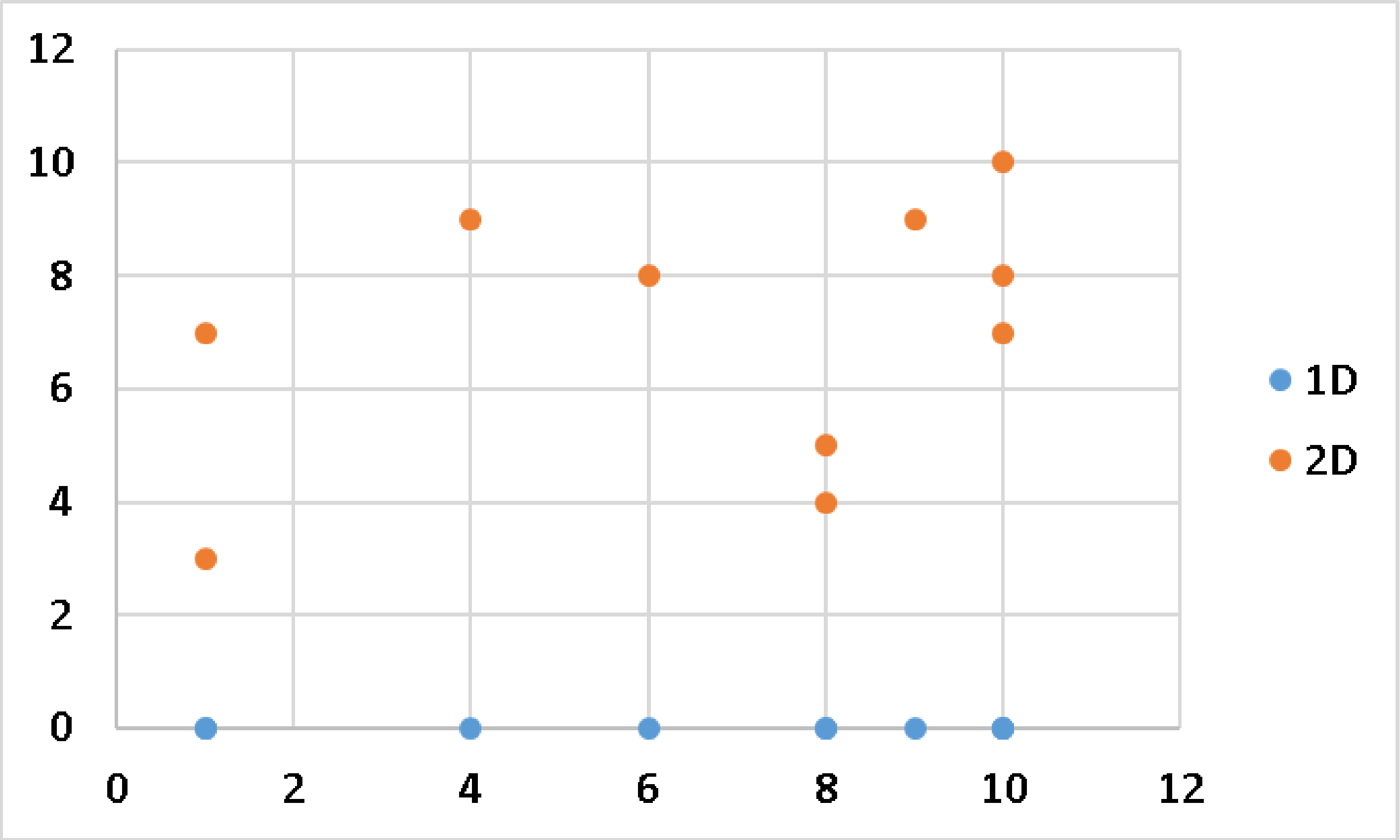
- As dimensionality increases, the volume of the space increases so fast that the available data become sparse.
- In order to obtain a statistically sound and reliable result, the amount of data needed to support the result often grows exponentially with the dimensionality.

X	Y
4	9
8	4
8	5
9	9
1	7
6	8
10	8
10	10
10	7
1	3



- Organizing and searching data often relies on detecting areas where objects form groups with similar properties
  - In high dimensional data, however, all objects appear to be sparse and dissimilar in many ways, which prevents common data organization strategies from being efficient.

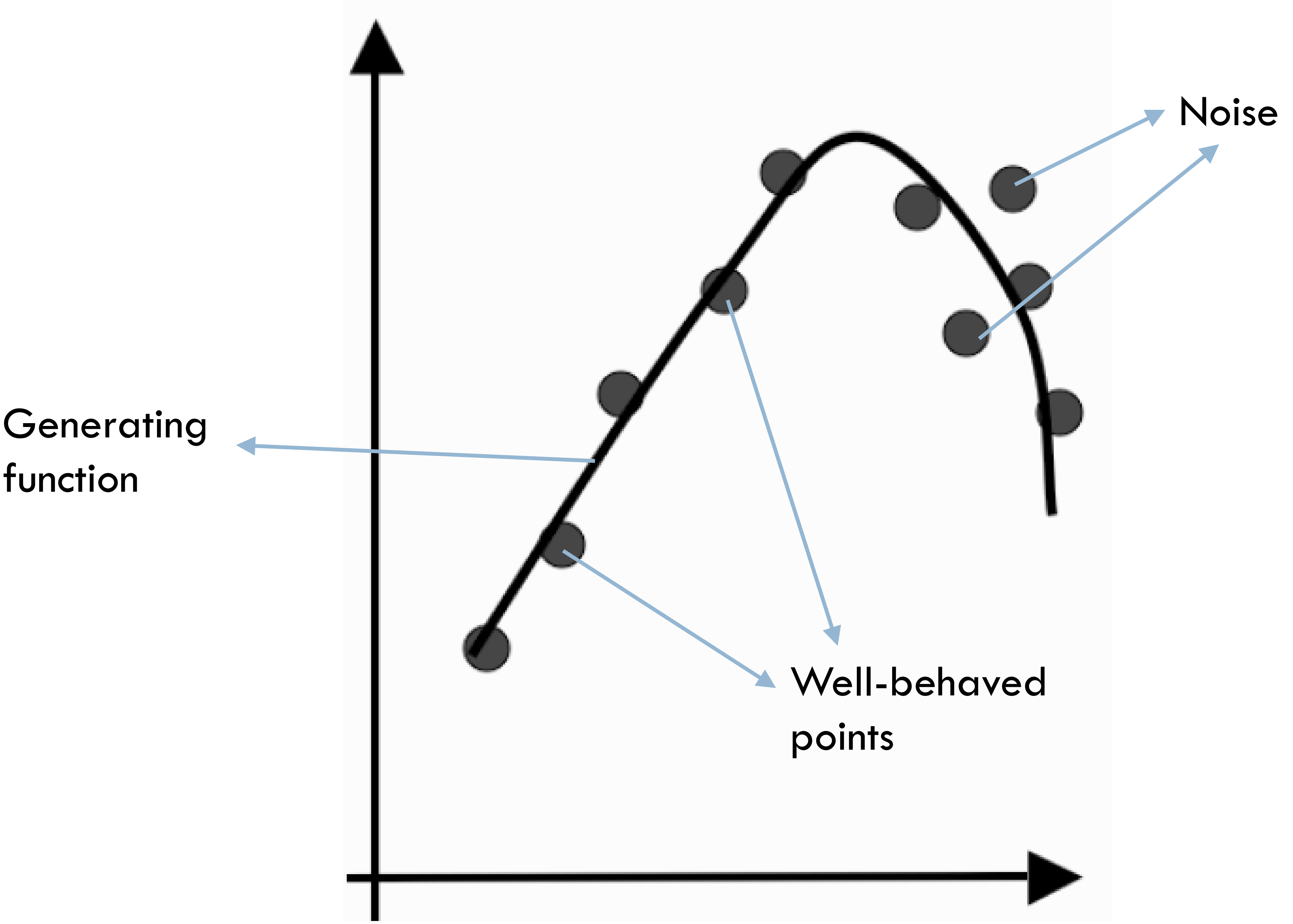
X	Y
4	9
8	4
8	5
9	9
1	7
6	8
10	8
10	10
10	7
1	3



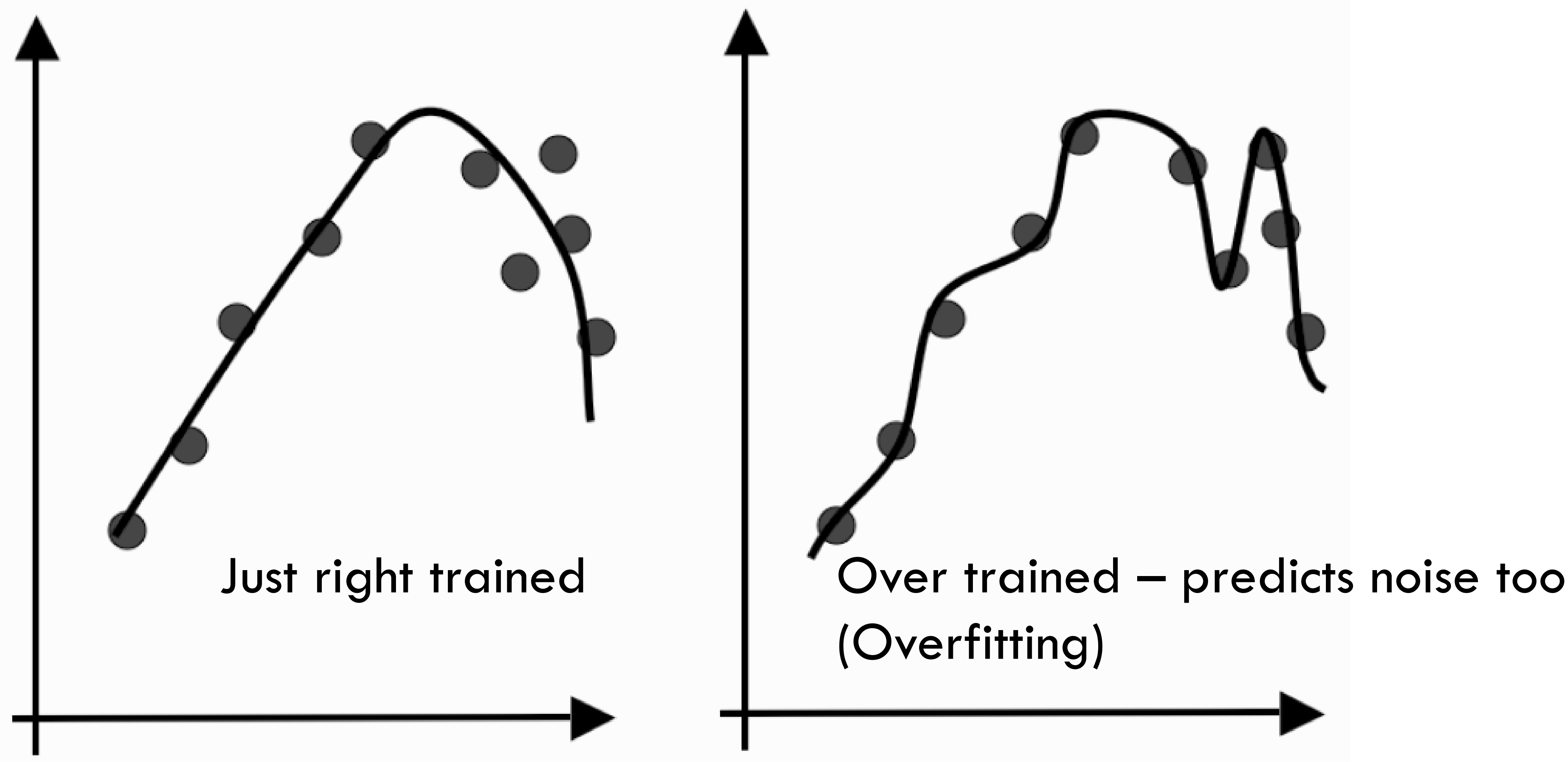
- Having higher dimension is not necessarily a good thing for machine learning algorithms.
- The same can hold for very few dimensions as well.
- Generally:
  - Results improve with increasing the number of dimensions
  - Then reach their best at “ideal #dimension”
  - And then start deteriorating with further higher dimensions.

Training, Testing, & Validation Sets (Supervised Learning)

- We "train" the algorithm on a certain data set called "Training Set"
  - The algorithm generalizes or adapts itself to predict target labels of yet to be seen data based on training dataset.
- Once trained, the algorithm is evaluated on "Testing Set" for how well it can predict the unseen data.
- Every data set is assumed to contain data points based on certain pattern or generating function – say *F*.
  - Most of the points can be generated nicely by *F*.
  - Some cannot ! They are ***noise***.

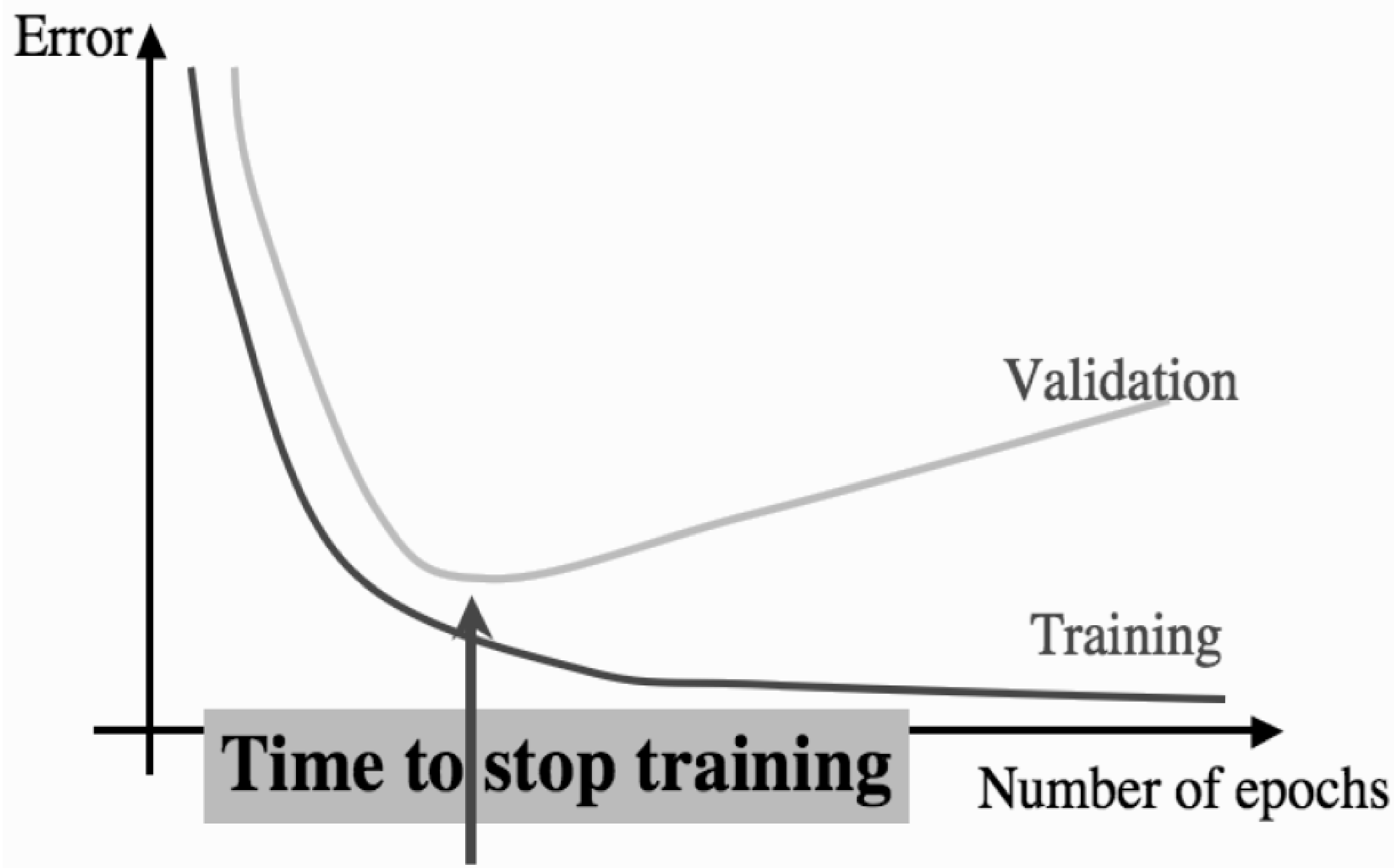

Noise
Generating
function
Well-behaved
points

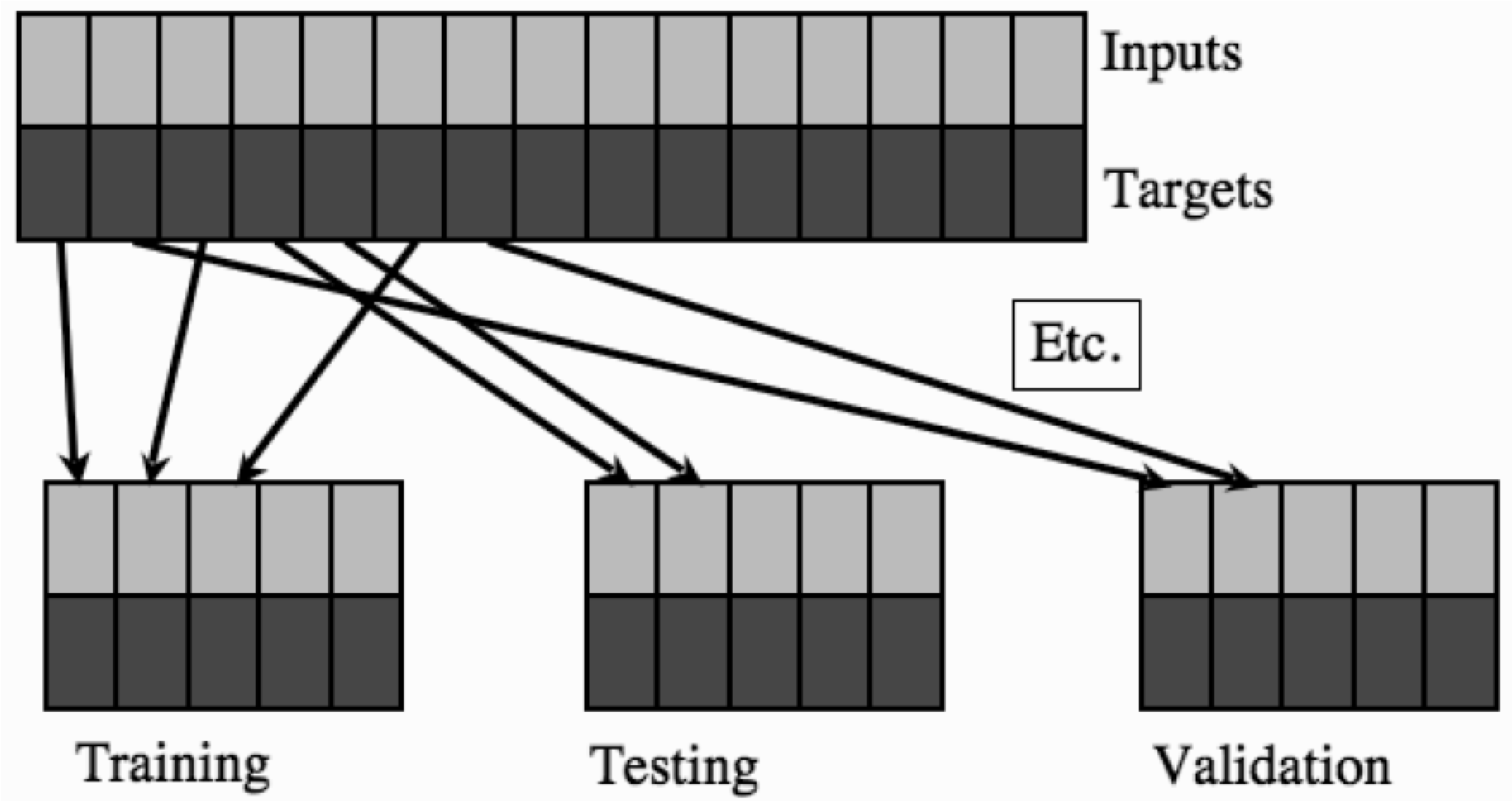
Overfitting



Prediction snapshots at two different points of training

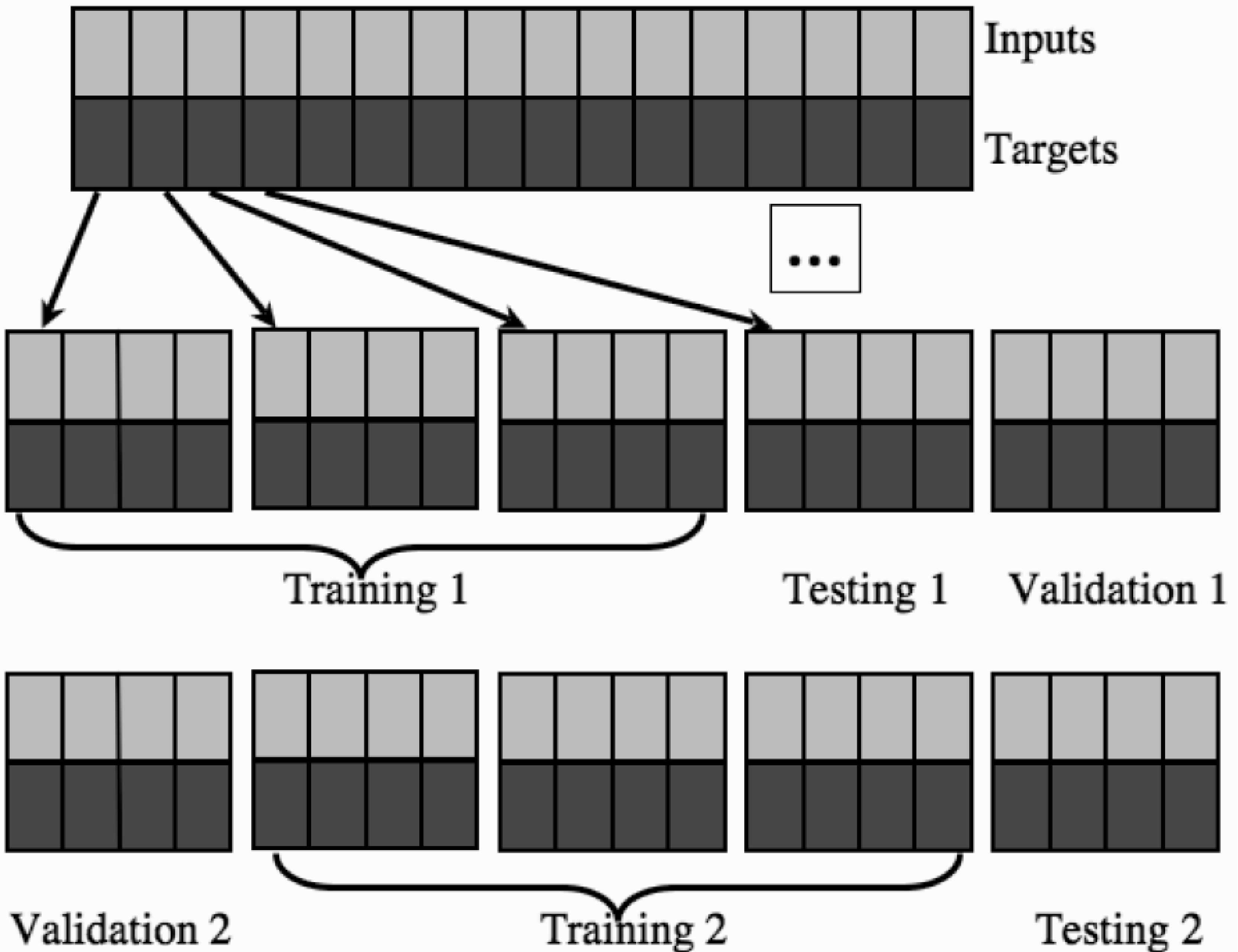
- We want to stop training before overfitting happens
  - We need to evaluate how well the training algorithm is generalizing (predicting) an unseen data set.
  - Training data set cannot be used here – wont detect overfitting
  - Test data set cannot be used – saved for final evaluation
  - We use "Validation set" – a third data set, which is different from training set and test set.
    - The process is called cross validation.


Error
Validation
Training
Time to stop training
Number of epochs



50: 25:25 (if you have plenty of data)
60:20:20 (if you don't have plenty of data)

Multi-fold Cross Validation



Model with lowest validation error is selected, OR, Average error is considered

Model 1

Model 2

Testing Output Results

- Confusion Matrix: a $(k * k)$ matrix, where $k =$ #target labels
  - $(i, j)^{th}$ entry denotes #samples having target label i, but labelled as j by the algorithm.

	Outputs		
	C_1	C_2	C_3
C_1	5	1	0
C_2	1	4	1
C_3	2	0	4

C_3 has most misclassifications, i.e., two.

Accuracy

- Assume a binary classification (Class I and Class II)
- Consider results from Class I perspective:
  - True Positive (TP): An observation correctly classified into Class I
  - False Positive (FP): An observation incorrectly classified into Class I
  - True Negative (TN): An observation correctly classified into the other class (i.e., Class II)
  - False Negative (FN): An observation incorrectly classified into the other class (i.e., Class II)

- Accuracy is defined as:

$$\frac{\#\text{Correct Predictions}}{\#\text{Total Predictions}} = \frac{\#TP + \#TN}{\#TP + \#FP + \#TN + \#FN}$$

Incorrectly given in text book Eq. 2.2

$$\text{Sensitivity} = \frac{\#TP}{\#TP + \#FN} \Rightarrow = \frac{\#\text{Correct Positive Examples}}{\#\text{Total Positive Examples}}$$

$$\text{Specificity} = \frac{\#TN}{\#TN + \#FP} \Rightarrow = \frac{\#\text{Correct Negative Examples}}{\#\text{Total Negative Examples}}$$

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

$$\text{Recall} = \frac{\#TP}{\#TP + \#FN}$$

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

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

$$\text{Precision} = \frac{\#TP}{\#TP + \#FP} \Rightarrow = \frac{\#\text{Correct Positive Examples}}{\#\text{Total Classified as Positive}}$$

$$\text{Recall} = \frac{\#TP}{\#TP + \#FN} \Rightarrow = \frac{\#\text{Correct Positive Examples}}{\#\text{Total Positive Examples}}$$

Incorrectly given in text book: swapped text-based definitions of Precision and Recall

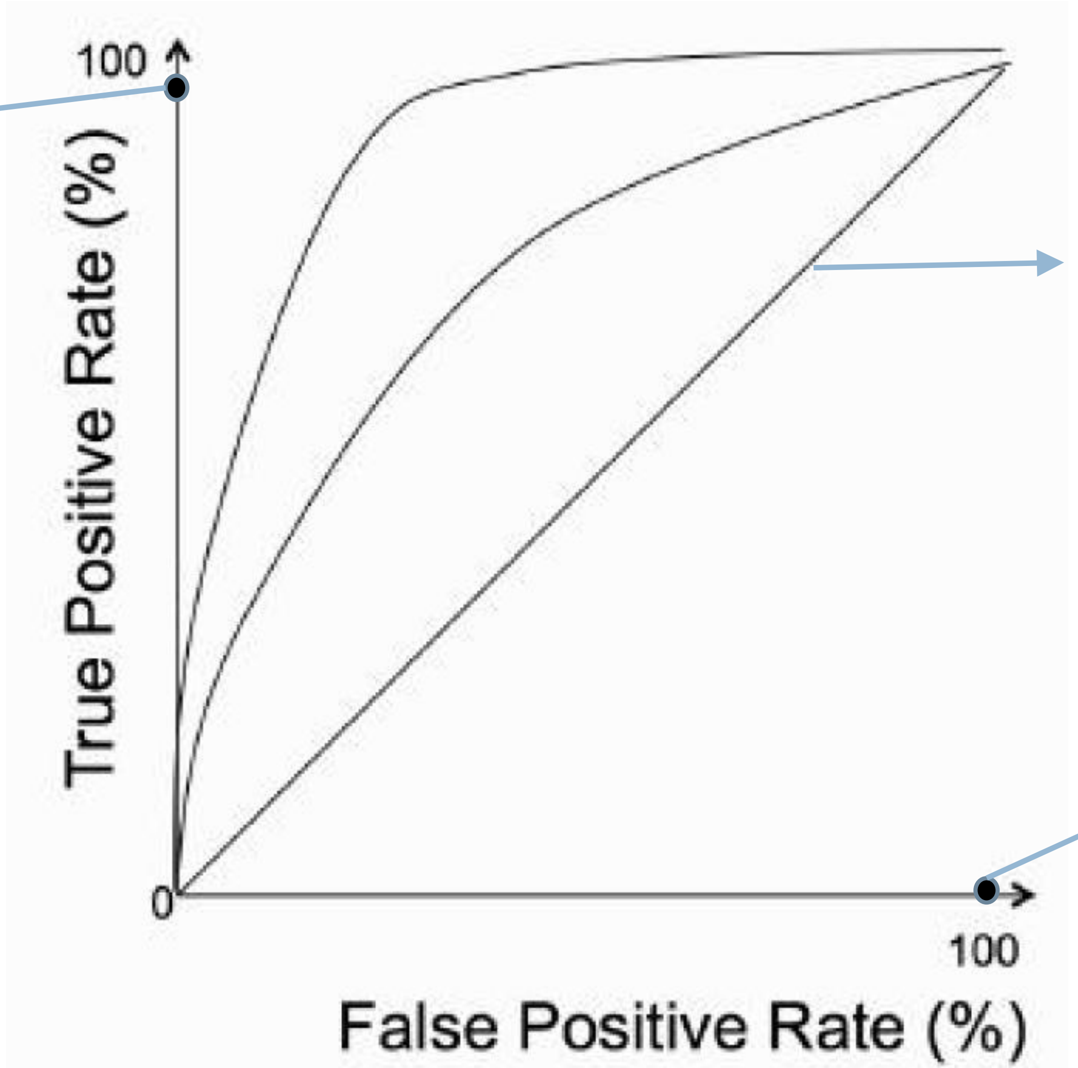
$$\text{Precision} = \frac{\#TP}{\#TP + \#FP} \qquad \text{Recall} = \frac{\#TP}{\#TP + \#FN}$$

- Precision $\propto \frac{1}{\text{Recall}}$ …to an extent
- F_1 score = harmonic mean of precision and recall

$$= 2 \times \left(\frac{1}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} \right) = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Receiver Operator Characteristic (ROC) Curve

Ideal Classifier



By chance – 50-50 ratio

The further away you are from diagonal – the better it is

Anti Classifier

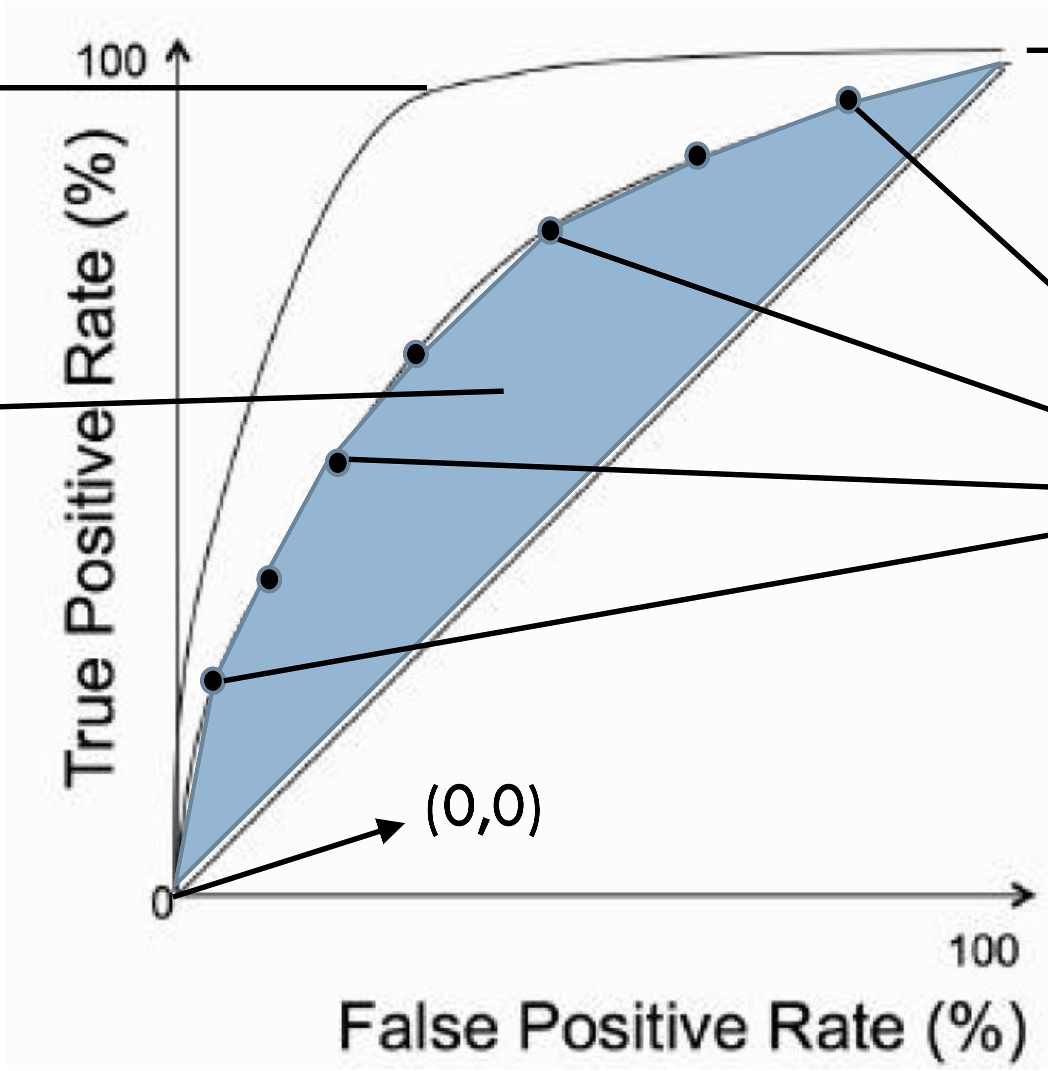
True positive rate = $\dfrac{\#TP}{\#positives}$

False positive rate = $\dfrac{\#FP}{\#negatives}$

Area under ROC curve

Better Result

Area under ROC curve – larger area means better classifier



(1,1)

(0,0)

Results from multi-fold cross validation of a classifier

Matthew's Correlation Coefficient

- So far defined measures of accuracy work best when #positive samples is same as #negative samples in the dataset.
- Otherwise, a more correct measure is Matthew's Correlation Coefficient:

$$MCC = \frac{\#TP \times \#TN - \#FP \times \#FN}{\sqrt{(\#TP + \#FP)(\#TP + \#FN)(\#TN + \#FP)(\#TN + \#FN)}}$$

Case of Multi-Class Classification

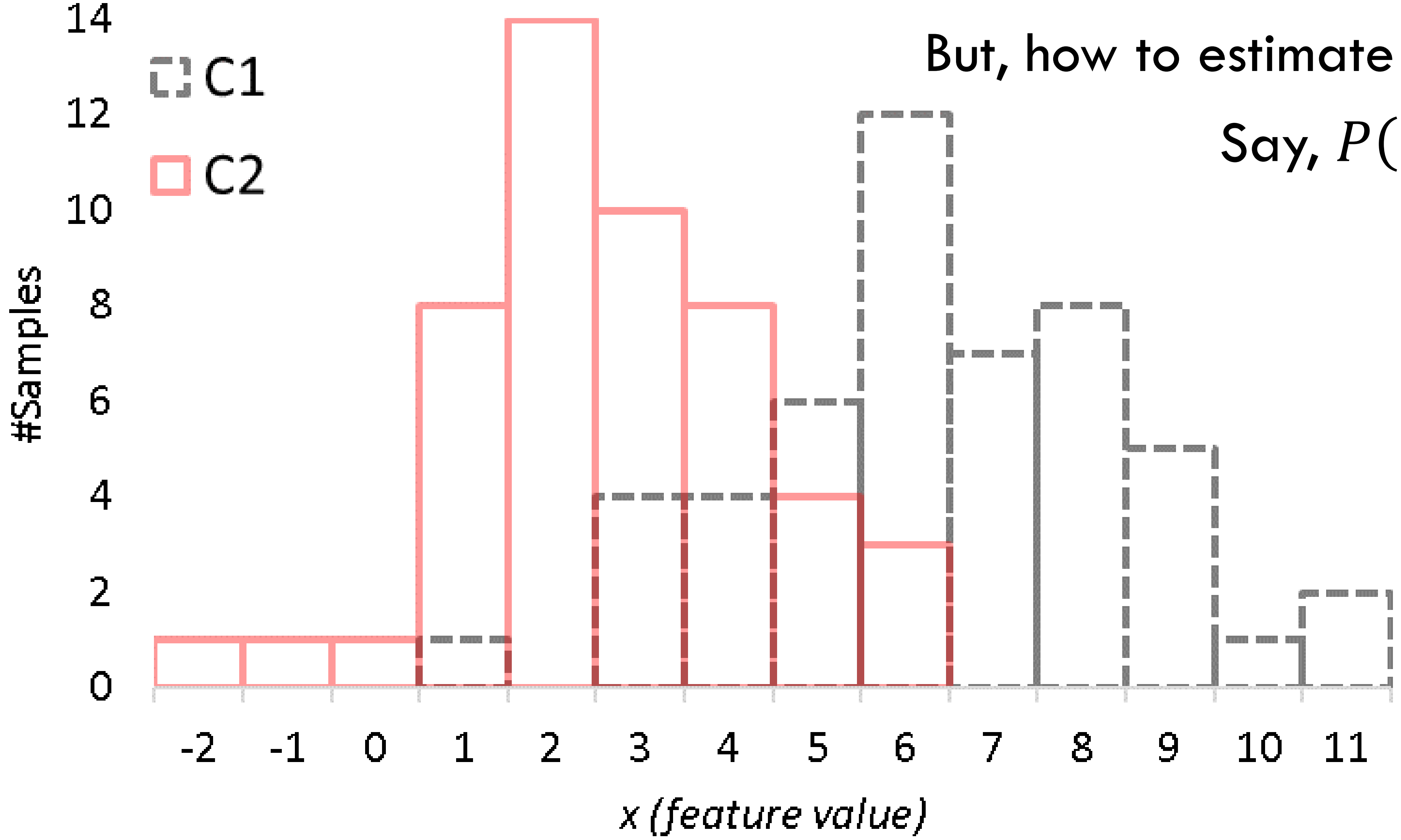
- The above measures can be calculated individually for each class X
  - Considering X as the positive class and clubbing all the other classes together as the negative class.

Maximum a Posteriori (MAP) Hypothesis

- What is the most likely class given the training data?
- Let $X_j = \{X_j^1, X_j^2, \dots, X_j^n\}$ be an input vector. Then, we are interested in class C_x such that

$$P(C_x|X_j) > P(C_y|X_j) \ \forall x, y$$

- But, how to estimate $P(C_x|X_j)$???



But, how to estimate $P(C_x|X_j)$???

Say, $P(C_1|x = 5) =$?

$= \frac{6}{10}$

$$P(C_x|X_j) = \frac{\#\text{Examples in bin } X_j \text{ of class } C_1}{\#\text{Examples in bin } X_j}$$

- #samples=10000, #dimension =1, #bins/dimension=14, #bins=14
  - $E\left(\frac{\#\text{samples}}{\text{bin}}\right) = \frac{10000}{14}$
- #samples=10000, #dimension =2, #bins/dimension=14, #bins= 14^2
  - $E\left(\frac{\#\text{samples}}{\text{bin}}\right) = \frac{10000}{14^2}$
- #samples=10000, #dimension =3, #bins/dimension=14, #bins=14^3
  - $E\left(\frac{\#\text{samples}}{\text{bin}}\right) = \frac{10000}{14^3}$

…

- #samples=10000, #dimension =5, #bins/dimension=14, #bins=14^5
  - $E\left(\frac{\#\text{samples}}{\text{bin}}\right) = \frac{10000}{14^5}$

- For, $X_j = \{X_j^1, X_j^2, \dots, X_j^n\}$, as *n* (#dimensions) increases, #samples in each bin of histogram shrinks
  - Becomes almost 1 for each bin (curse of dimensionality)
  - Severe lack of statistically confidence

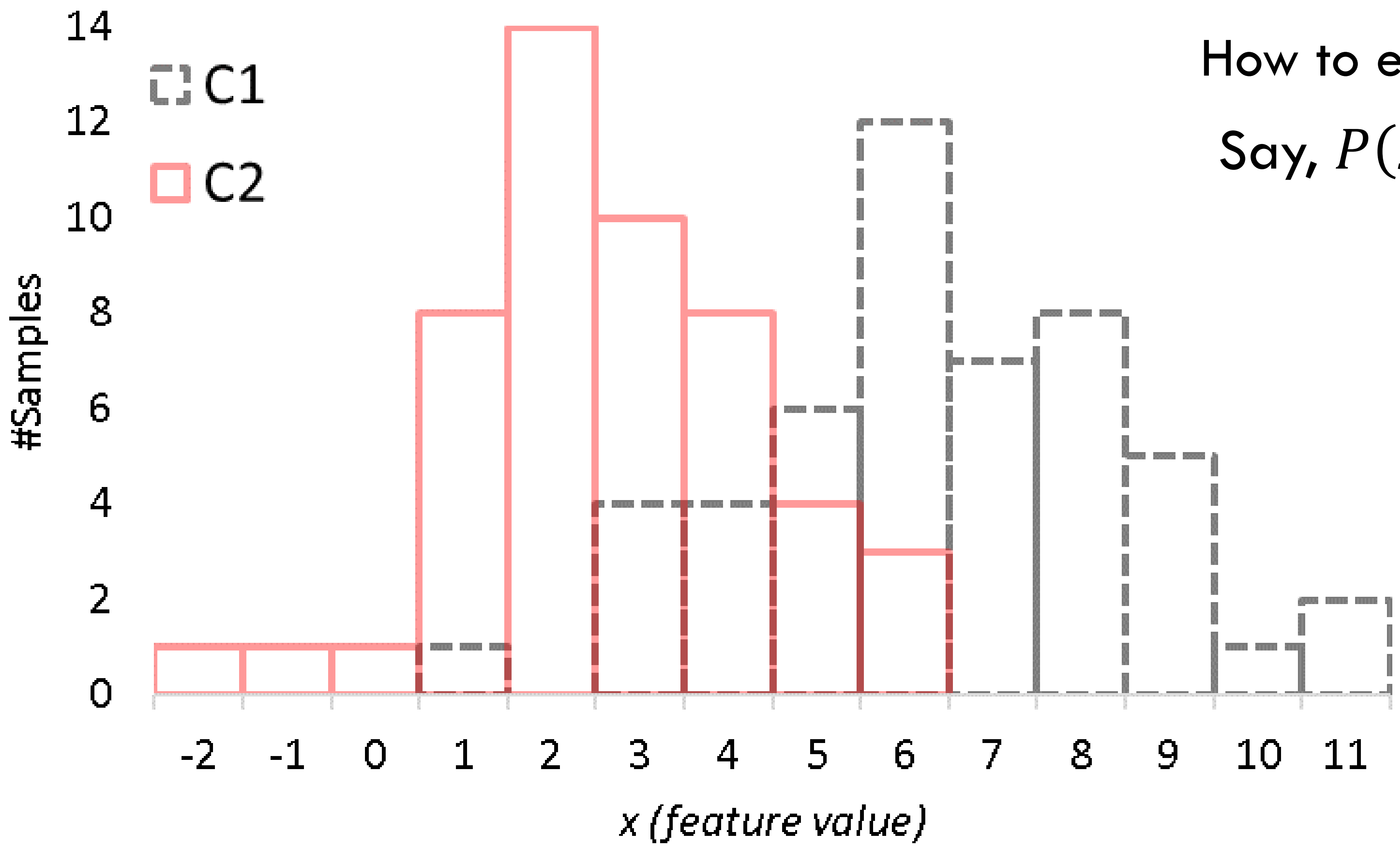
Using Bayes' Rule

$$P(C_i|X_j) = \frac{P(X_j|C_i)P(C_i)}{P(X_j)}$$

Where:

$$P(X_k) = \sum_i P(X_k|C_i)P(C_i)$$

$P(C_i)$ can be estimated easily.
How to estimate $P(X_k|C_i)$?



How to estimate $P(X_k|C_i)$?

Say, $P(x = 5|C_2)$ =?

$=\frac{5}{50}$

For X = X_j, $P(X_j|C_1) = \frac{\text{\#Examples in bin } X_j \text{ of class } C_1}{\text{Toal \#examples of class } C_1}$

For X = X_j, $P(X_j|C_1) = \frac{\text{\#Examples in bin } X_j \text{ of class } C_1}{\text{Toal \#examples of class } C_1}$

- For, $X_j = \{X_j^1, X_j^2, \ldots, X_j^n\}$, as *n* (#dimensions) increases, #samples in each bin of histogram shrinks
  - Becomes almost 1 for each bin (curse of dimensionality)
- Simplifying assumption: "With respect to classification, elements of feature vector are mutually conditionally independent"

$$P(X_j^1 = a_1, X_j^2 = a_2, \ldots, X_j^n = a_n|C_i)$$

$$=$$

$$P(X_j^1 = a_1|C_i) \times P(X_j^2 = a_2|C_i) \times \ldots \times P(X_j^n = a_n|C_i) = \prod_k P(X_j^k = a_k|C_i)$$

$$P(X_j^1 = a_1, X_j^2 = a_2, \ldots, X_j^n = a_n | C_i)$$

$$=$$

$$P(X_j^1 = a_1|C_i) \times P(X_j^2 = a_2|C_i) \times \ldots \times P(X_j^n = a_n|C_i) = \prod_k P(X_j^k = a_k|C_i)$$

Hence, rule for naïve Bayes' classifier is to select C_i for which the following is maximum:

$$P(C_i) \prod_k P(X_j^k = a_k|C_i)$$

Deadline?	Is there a party?	Lazy?	Activity
Urgent	Yes	Yes	Party
Urgent	No	Yes	Study
Near	Yes	Yes	Party
None	Yes	No	Party
None	No	Yes	Pub
None	Yes	No	Party
Near	No	No	Study
Near	No	Yes	TV
Near	Yes	Yes	Party
Urgent	No	No	Study

List of activity you have been doing since last few days

Suppose a deadline is looming, but it is not urgent. Further, there is no ongoing party and you are feeling lazy. Based on naïve Bayes' classifier, what will you do?

Input $X_j = \{\text{Deadline} = \text{Near}, \text{Party} = \text{No}, \text{Lazy} = \text{Yes}\}$

$$\max_i P(C_i)\prod_k P(X_j^k = a_k|C_i)$$

Let us consider class “Party”

$$P(\text{Party}) = ?$$
$$= 5/10$$
$$P(\text{Deadline} = \text{Near}|\text{Party}) = ?$$
$$= \frac{2}{5}$$
$$P(\text{Party} = \text{No}|\text{Party}) = ?$$
$$= \frac{0}{5}$$
$$P(\text{Lazy} = \text{Yes}|\text{Party}) = ?$$
$$= \frac{3}{5}$$

Deadline?	Is there a party?	Lazy?	Activity
Urgent	Yes	Yes	Party
Urgent	No	Yes	Study
Near	Yes	Yes	Party
None	Yes	No	Party
None	No	Yes	Pub
None	Yes	No	Party
Near	No	No	Study
Near	No	Yes	TV
Near	Yes	Yes	Party
Urgent	No	No	Study

$$X_j = \{\text{Deadline} = \text{Near}, \text{Party} = \text{No}, \text{Lazy} = \text{Yes}\}$$

$$P(C_i)\prod_k P(X_j^k = a_k|C_i) = P(\text{Party}) \times P(\text{Deadline} = \text{Near}|\text{Party}) \times P(\text{Party} = \text{No}|\text{Party}) \times P(\text{Lazy} = \text{Yes}|\text{Party}) = 0$$

Basic Statistics

- Average measures: Mean, Median, Mode, Variance
  - Mean: arithmetic average.
  - Median: the middle value (sort and find)
  - Mode: most frequent value
  - Variance: measures how spread out values are

Variance

$$\mathrm{var}(\{x_i\}) = \sigma^2(\{x_i\}) = E\left((\{x_i\} - \mu)^2\right) = \frac{\left(\sum_{i=1}^{N}(x_i - \mu)^2\right)}{N}$$

Where:

- x_i is random variable sampled N times $(x_1, x_2, \ldots, x_n)$
- μ denotes the mean of x_i
- σ is known as standard deviation (square root of variance)
- $E\left((\{x_i\} - \mu)^2\right)$ is expectation of the squared deviation of a random variable from its mean

- Covariance: measures dependency of two (random) variables

$$\mathrm{cov}(\{x_i\},\{y_i\}) = E((\{x_i\}-\mu)(\{y_i\}-v))$$

Where, μ is the mean of x_i and v is the mean of y_i

  - Zero value: both (x_i and y_i) are unrelated
  - Positive value: both increase/decrease at the same time.
  - Negative value: when one increases, the other decreases, and vice-versa

$$\mathrm{cov}(\{x_i\},\{x_i\}) = \sigma^2(\{x_i\}) = \mathrm{var}(\{x_i\})$$

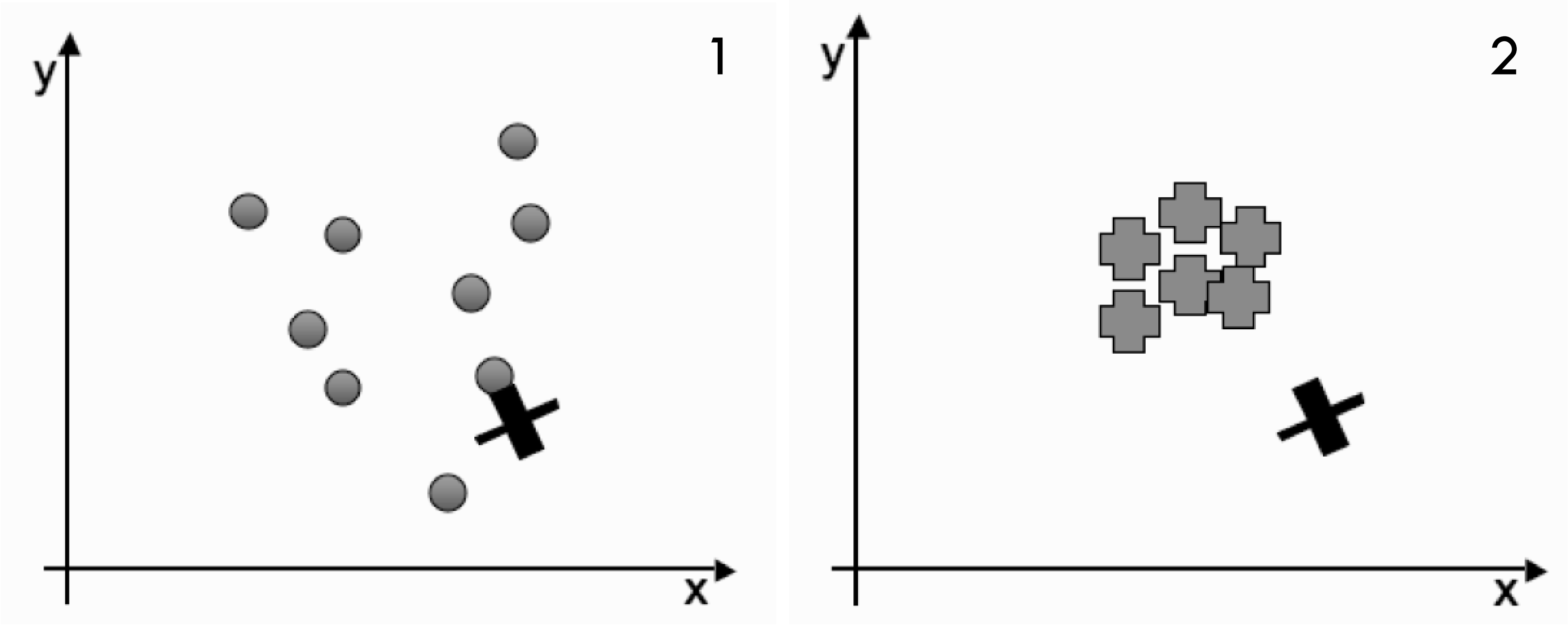
- For multiple variables, **covariance matrix** contains covariance between all pairs of variables.

$$\mathbf{\Sigma} = \begin{pmatrix} E[(\mathbf{x}_1 - \mu_1)(\mathbf{x}_1 - \mu_1)] & E[(\mathbf{x}_1 - \mu_1)(\mathbf{x}_2 - \mu_2)] & \dots & E[(\mathbf{x}_1 - \mu_1)(\mathbf{x}_n - \mu_n)] \\ E[(\mathbf{x}_2 - \mu_2)(\mathbf{x}_1 - \mu_1)] & E[(\mathbf{x}_2 - \mu_2)(\mathbf{x}_2 - \mu_2)] & \dots & E[(\mathbf{x}_2 - \mu_2)(\mathbf{x}_n - \mu_n)] \\ \dots & \dots & \dots & \dots \\ E[(\mathbf{x}_n - \mu_n)(\mathbf{x}_1 - \mu_1)] & E[(\mathbf{x}_n - \mu_n)(\mathbf{x}_2 - \mu_2)] & \dots & E[(\mathbf{x}_n - \mu_n)(\mathbf{x}_n - \mu_n)] \end{pmatrix}$$

Where, x_i is a column vector describing the elements of i^{th} variable, and μ_i is their mean.

- Square and symmetric matrix
- Matrix form:

$$\mathbf{\Sigma} = E[(\mathbf{X} - E[\mathbf{X}])(\mathbf{X} - E[\mathbf{X}])^T]$$



Is test point large ‘X’ part of the data?

- Mahalanobis distance: captures distance between a point and a distribution (set of points)

$$D_M(\mathbf{x}) = \sqrt{(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})}.$$

Where:

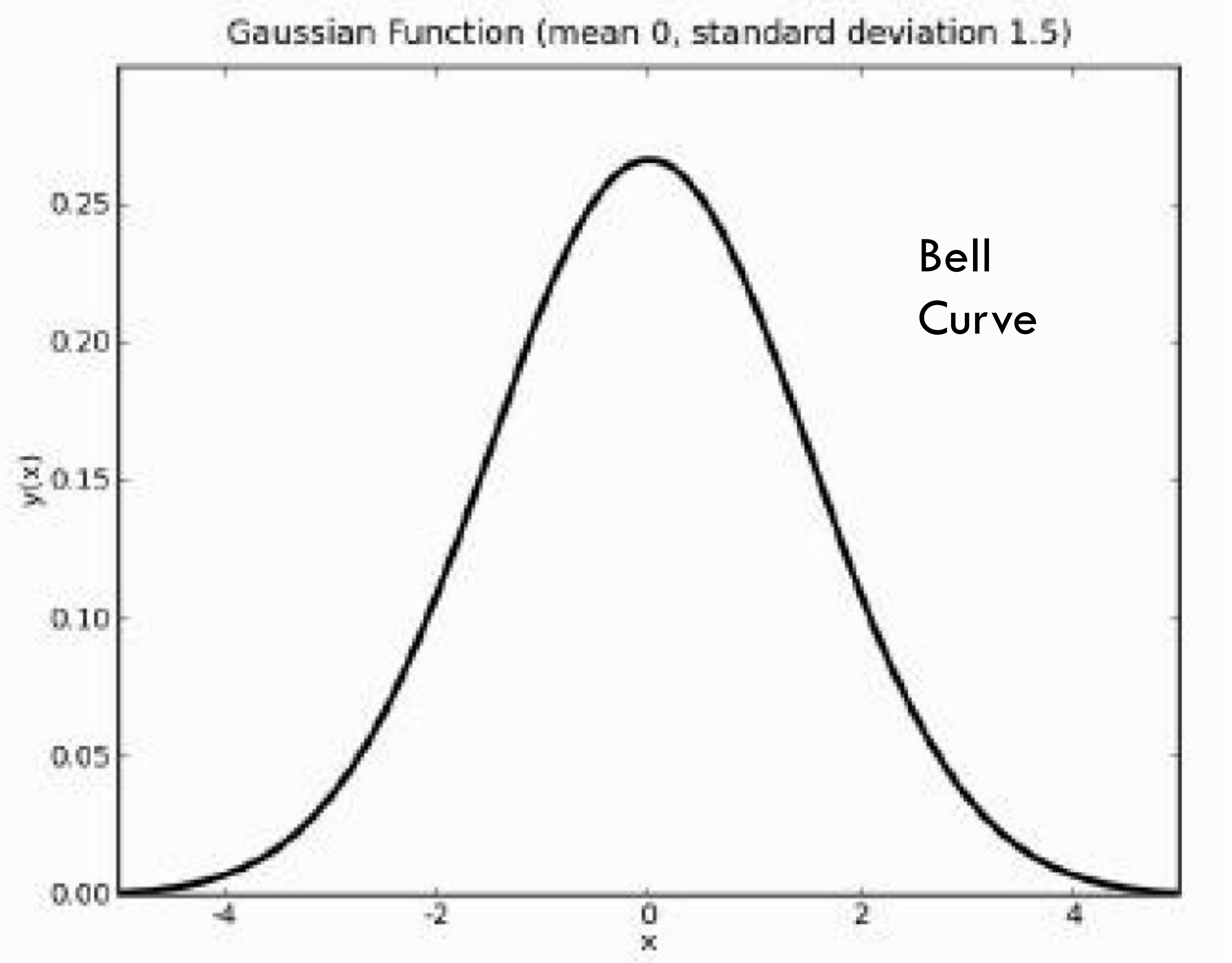
  - x is data (point) arranged as a column vector
  - μ is a column vector representing the mean of the distribution
  - Σ^{-1} is the inverse covariance matrix of the distribution
- When Σ is an identity matrix, $D_M(x)$ reduces to Euclidian distance.
- Intuitively, it measures how many standard deviations away x is from the mean of the distribution.

Gaussian or Normal Distribution

- It is a probability distribution defined as (for two dimension):

$$p(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)$$

Where, μ is the mean and σ the standard deviation.


Gaussian Function (mean 0, standard deviation 1.5)
y(x)
0.00
0.05
0.10
0.15
0.20
0.25
-4
-2
0
2
4
x

Bell
Curve

- For higher dimension, it is defined as

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

Where, Σ is the $n \times n$ covariance matrix