



Introduction to Machine Learning

These slides were assembled by Sujata Pathak, with grateful acknowledgment of the many others who made their course materials freely available online.

Sujata Pathak, IT, KJSCE

What is Machine Learning?

“Learning is any process by which a system improves performance from experience.”

- Herbert Simon

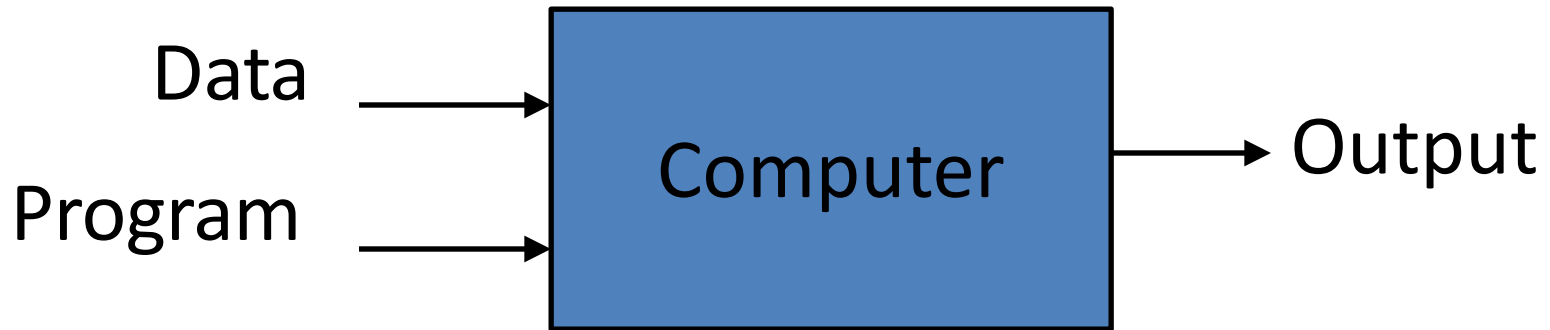
Definition by Tom Mitchell (1998):

Machine Learning is the study of algorithms that

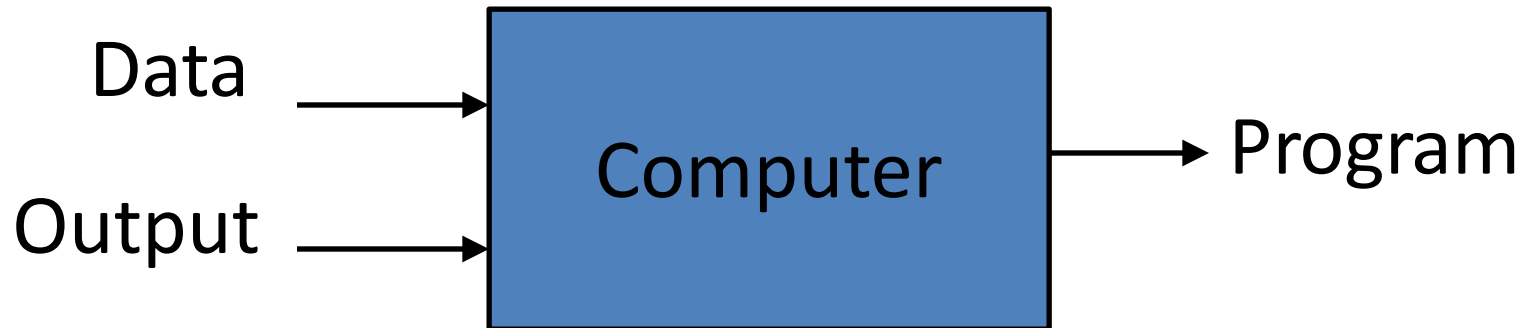
- improve their performance P
- at some task T
- with experience E .

A well-defined learning task is given by $\langle P, T, E \rangle$.

Traditional Programming



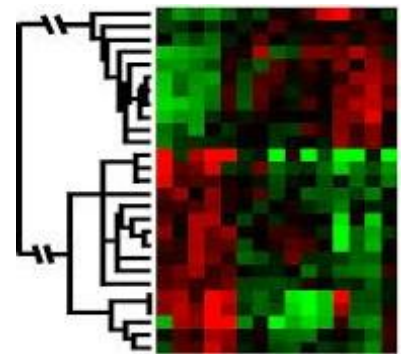
Machine Learning



When Do We Use Machine Learning?

ML is used when:

- Human expertise does not exist (navigating on Mars)
- Humans can't explain their expertise (speech recognition)
- Models must be customized (personalized medicine)
- Models are based on huge amounts of data (genomics)



Learning isn't always useful:

- There is no need to “learn” to calculate payroll

A classic example of a task that requires machine learning:
It is very hard to say what makes a 2

0 0 0 1 1 1 1 1 1 2

2 2 2 2 2 2 2 3 3 3

3 4 4 4 4 4 5 5 5 5

6 6 7 7 7 7 8 8 8 8

9 9 9 9 9 9 9 9 9 9

Some more examples of tasks that are best solved by using a learning algorithm

- Recognizing patterns:
 - Facial identities or facial expressions
 - Handwritten or spoken words
 - Medical images
- Generating patterns:
 - Generating images or motion sequences
- Recognizing anomalies:
 - Unusual credit card transactions
 - Unusual patterns of sensor readings in a nuclear power plant
- Prediction:
 - Future stock prices or currency exchange rates

Sample Applications

- Web search
- Computational biology
- Finance
- E-commerce
- Space exploration
- Robotics
- Information extraction
- Social networks
- Debugging software
- [Your favorite area]

Defining the Learning Task

Improve on task T , with respect to performance metric P , based on experience E

T : Recognizing hand-written words

P : Percentage of words correctly classified

E : Database of human-labeled images of handwritten words

T : Driving on four-lane highways using vision sensors

P : Average distance traveled before a human-judged error

E : A sequence of images and steering commands recorded while observing a human driver.

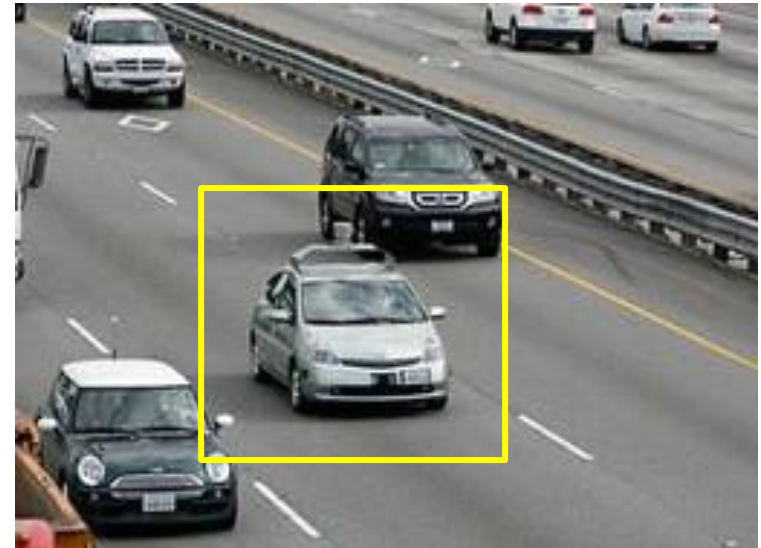
T : Categorize email messages as spam or legitimate.

P : Percentage of email messages correctly classified.

E : Database of emails, some with human-given labels

State of the Art Applications of Machine Learning

Autonomous Cars

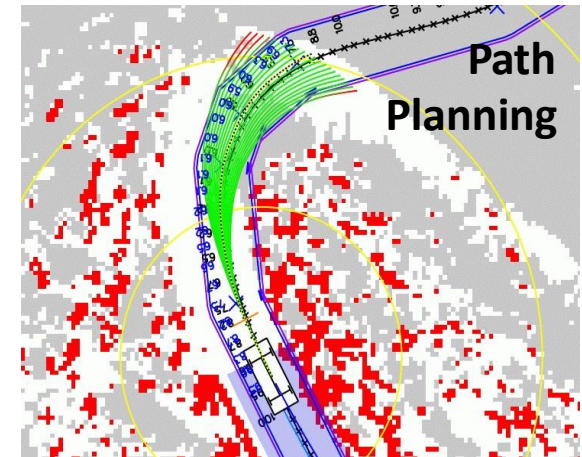


- Nevada made it legal for autonomous cars to drive on roads in June 2011
- As of 2013, four states (Nevada, Florida, California, and Michigan) have legalized autonomous cars

Penn's Autonomous Car →
(Ben Franklin Racing Team)

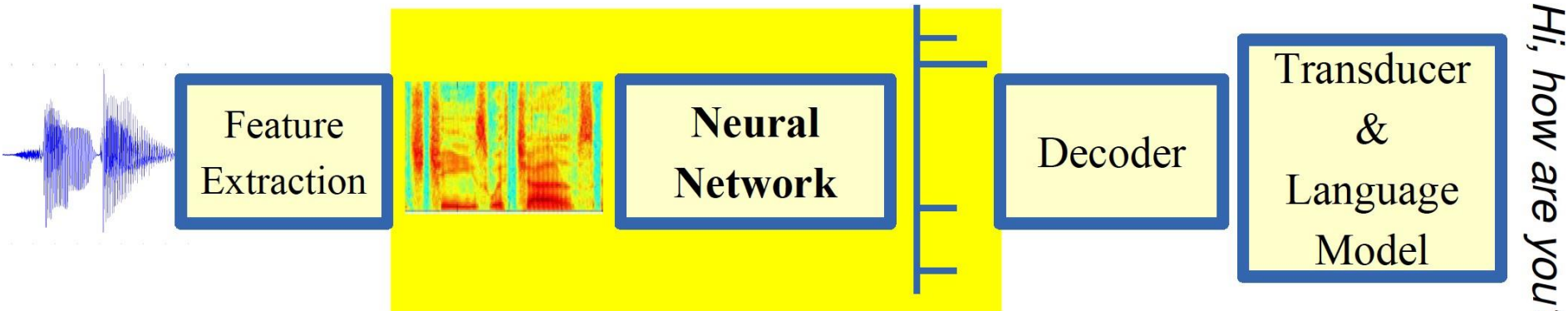


Autonomous Car Technology

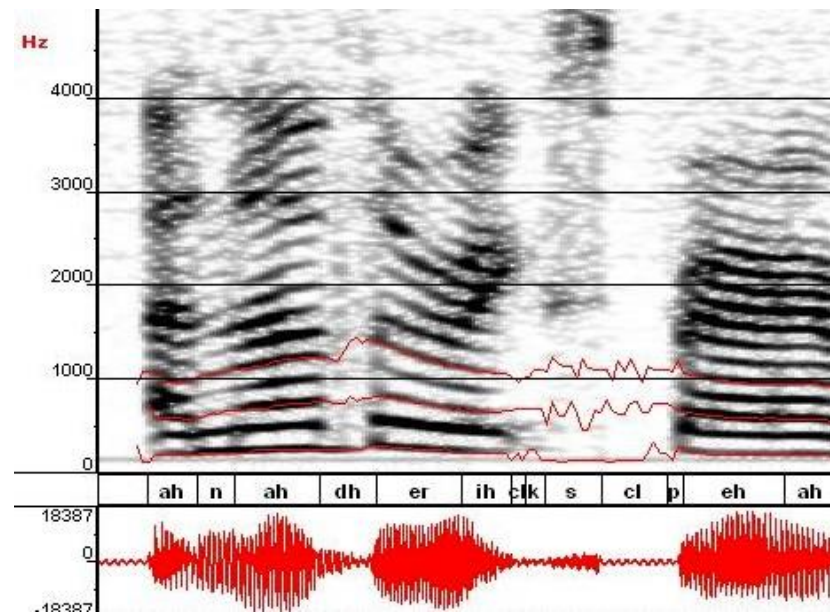


Machine Learning in Automatic Speech Recognition

A Typical Speech Recognition System



ML used to predict the phone states from the sound spectrogram



Deep learning has state-of-the-art results

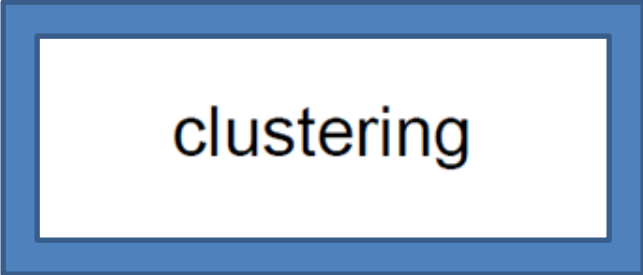
# Hidden Layers	1	2	4	8	10	12
Word Error Rate %	16.0	12.8	11.4	10.9	11.0	11.1

Baseline GMM performance = 15.4%

[Zeiler et al. "On rectified linear units for speech recognition" ICASSP 2013]

Sujata Pathak, IT, KJSCE

Machine Learning Problems

	<i>Supervised Learning</i>	<i>Unsupervised Learning</i>
<i>Discrete</i>	classification or categorization	 clustering
<i>Continuous</i>	regression	dimensionality reduction

The machine learning framework

Apply a prediction function to a feature representation of the image to get the desired output:



$f(x)$ = “apple”

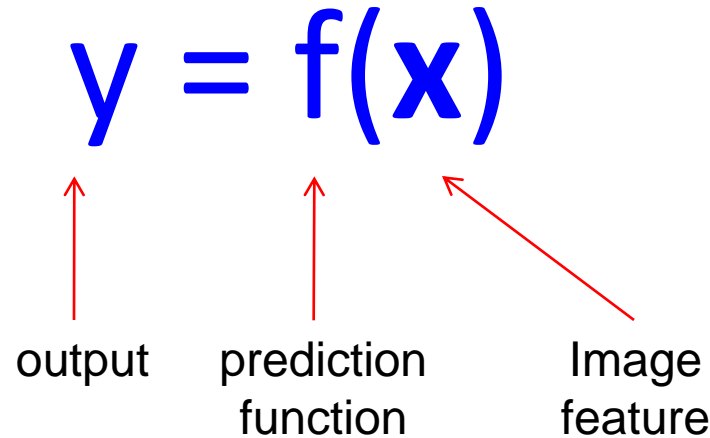


$f(x)$ = “tomato”



$f(x)$ = “cow”

The machine learning framework



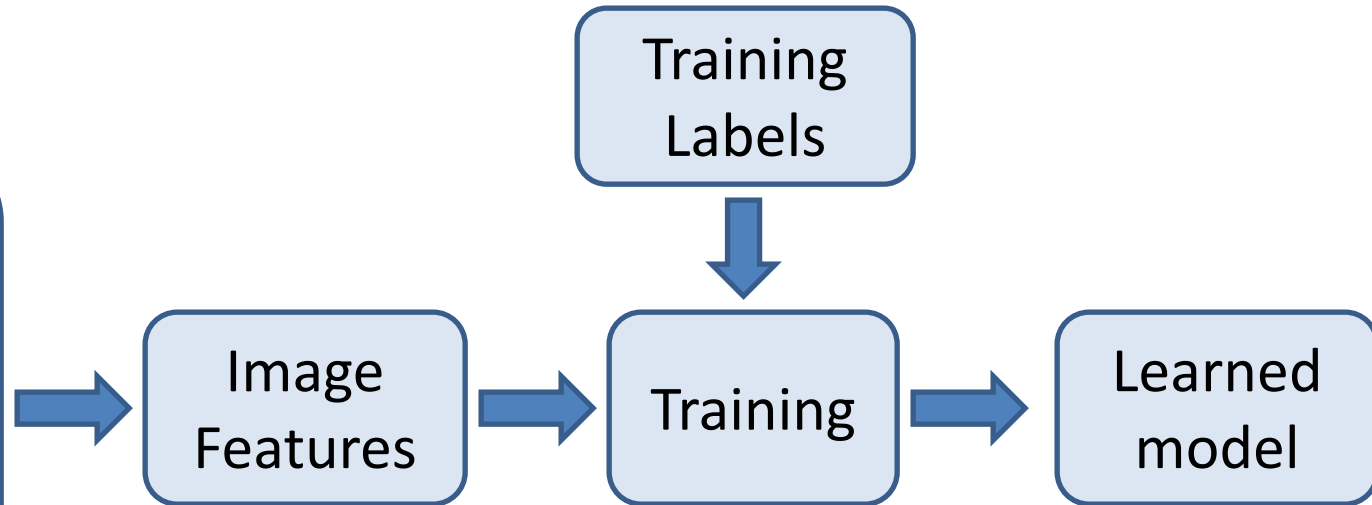
Training: given a *training set* of labeled examples $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$, estimate the prediction function f by minimizing the prediction error on the training set

Testing: apply f to a never before seen *test example* \mathbf{x} and output the predicted value $y = f(\mathbf{x})$

Steps

Training

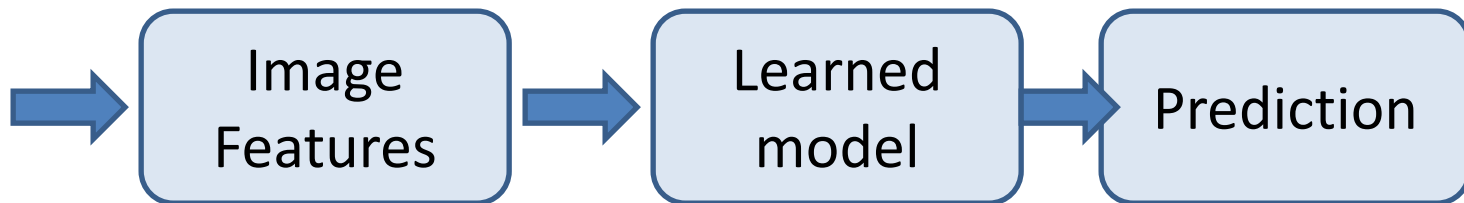
Training
Images



Testing



Test Image



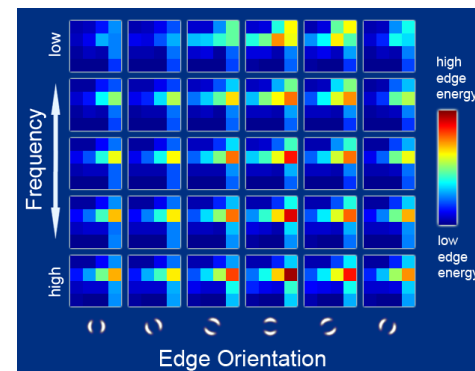
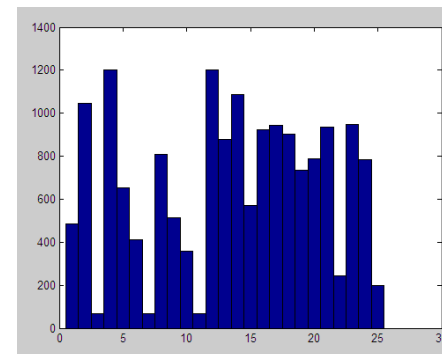
Features

Raw pixels

Histograms

GIST descriptors

...

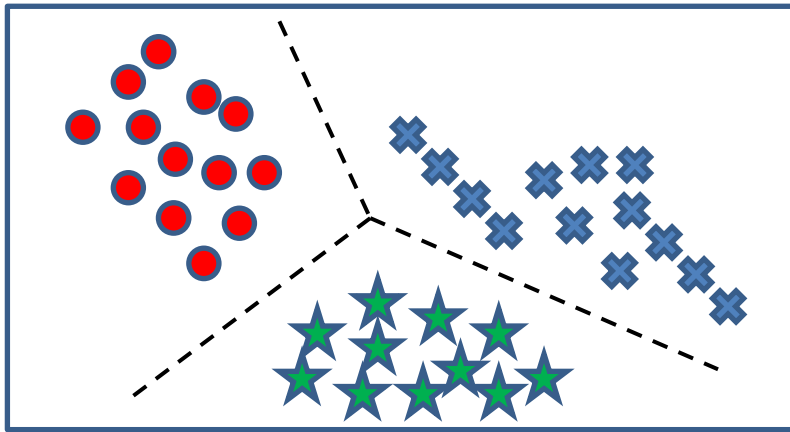


Types of Learning

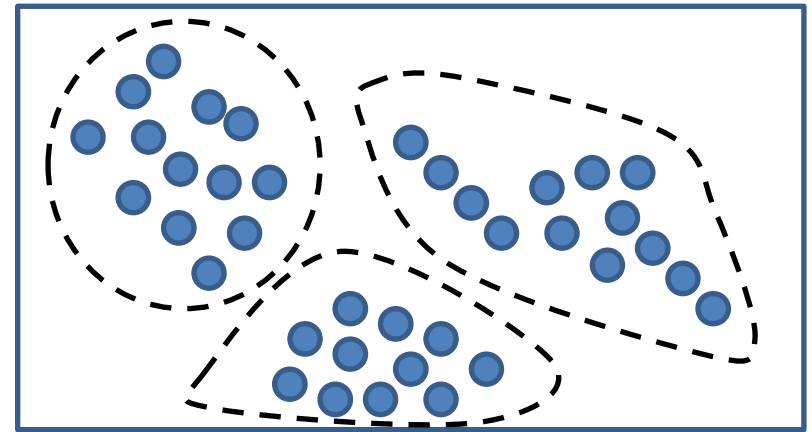
Types of Learning

- **Supervised (inductive) learning**
 - Given: training data + desired outputs (labels)
- **Unsupervised learning**
 - Given: training data (without desired outputs)
- **Semi-supervised learning**
 - Given: training data + a few desired outputs
- **Reinforcement learning**
 - Rewards from a sequence of actions
- **Evolutionary learning**
 - Biological evolution can be seen as a learning process

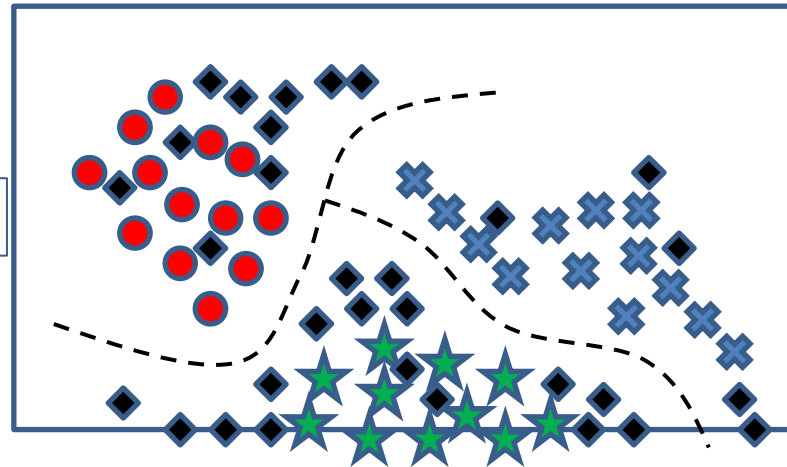
Types of Learning



Supervised learning



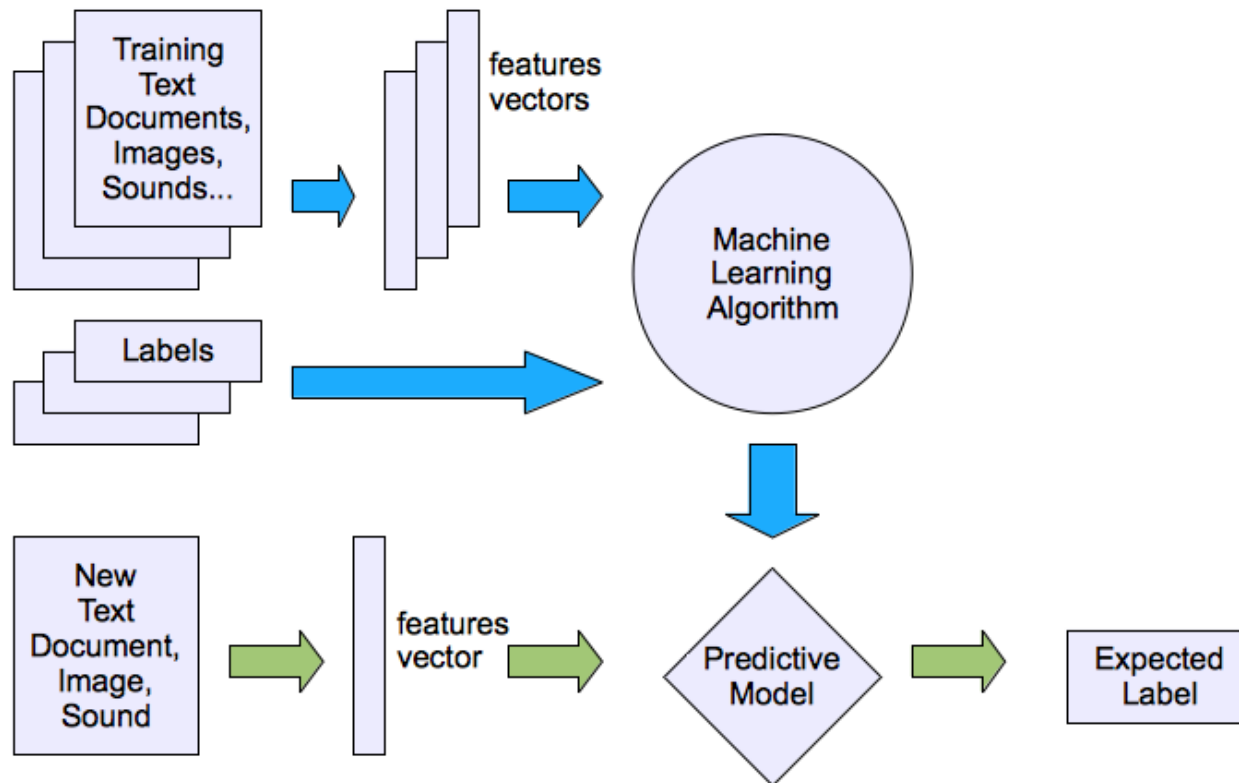
Unsupervised learning



Semi-supervised learning

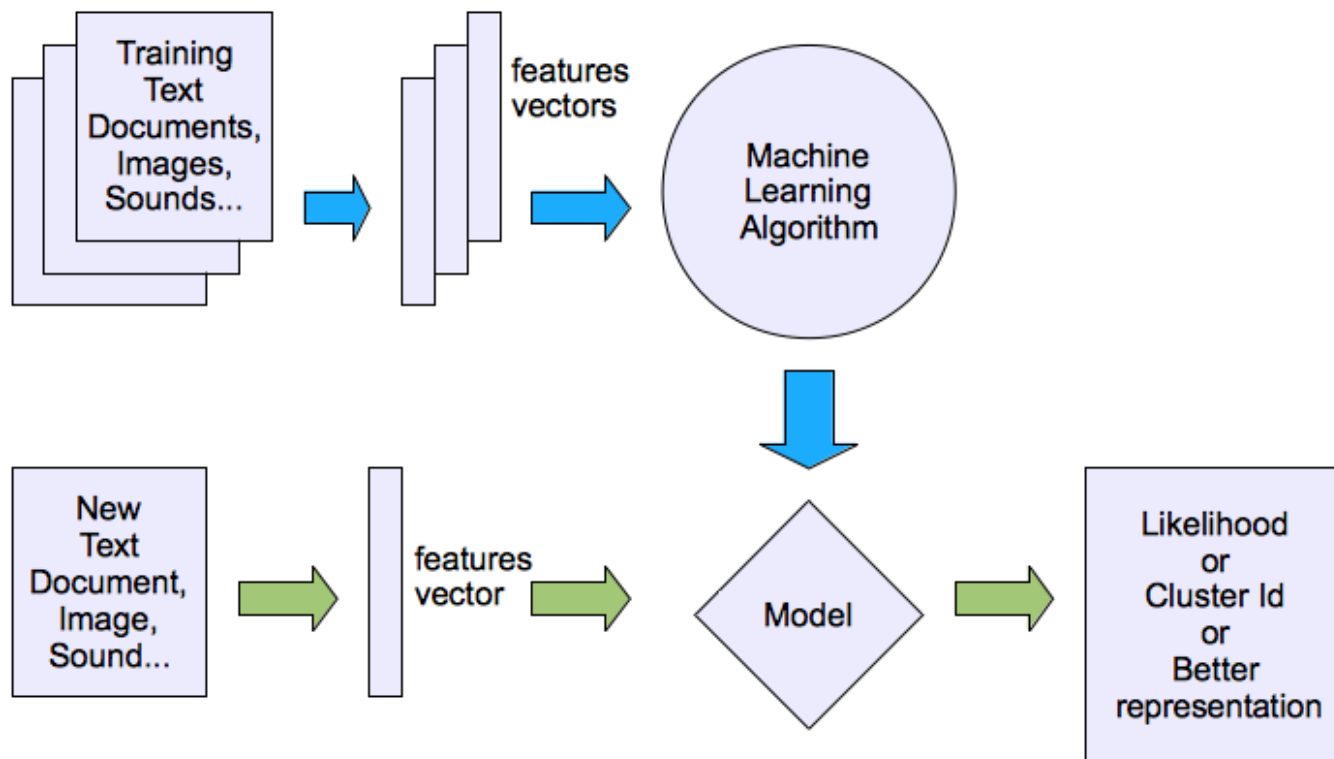
Types of Learning

Supervised learning



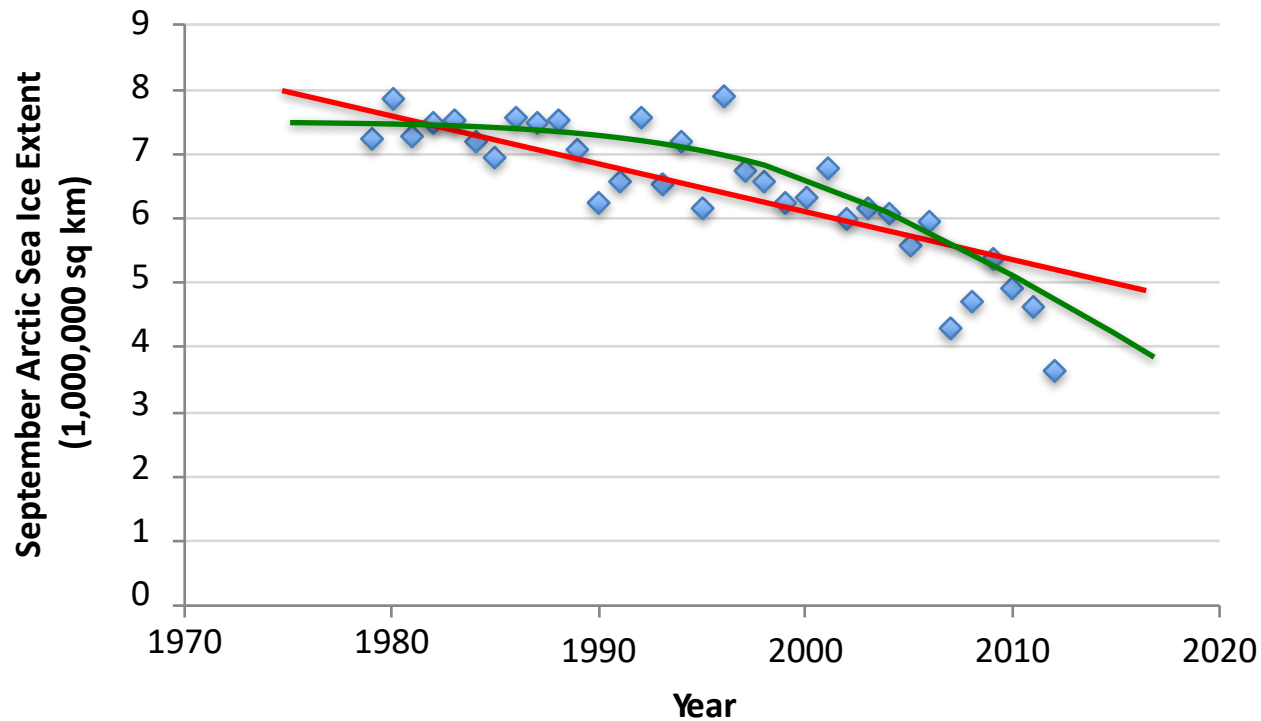
Types of Learning

Unsupervised learning



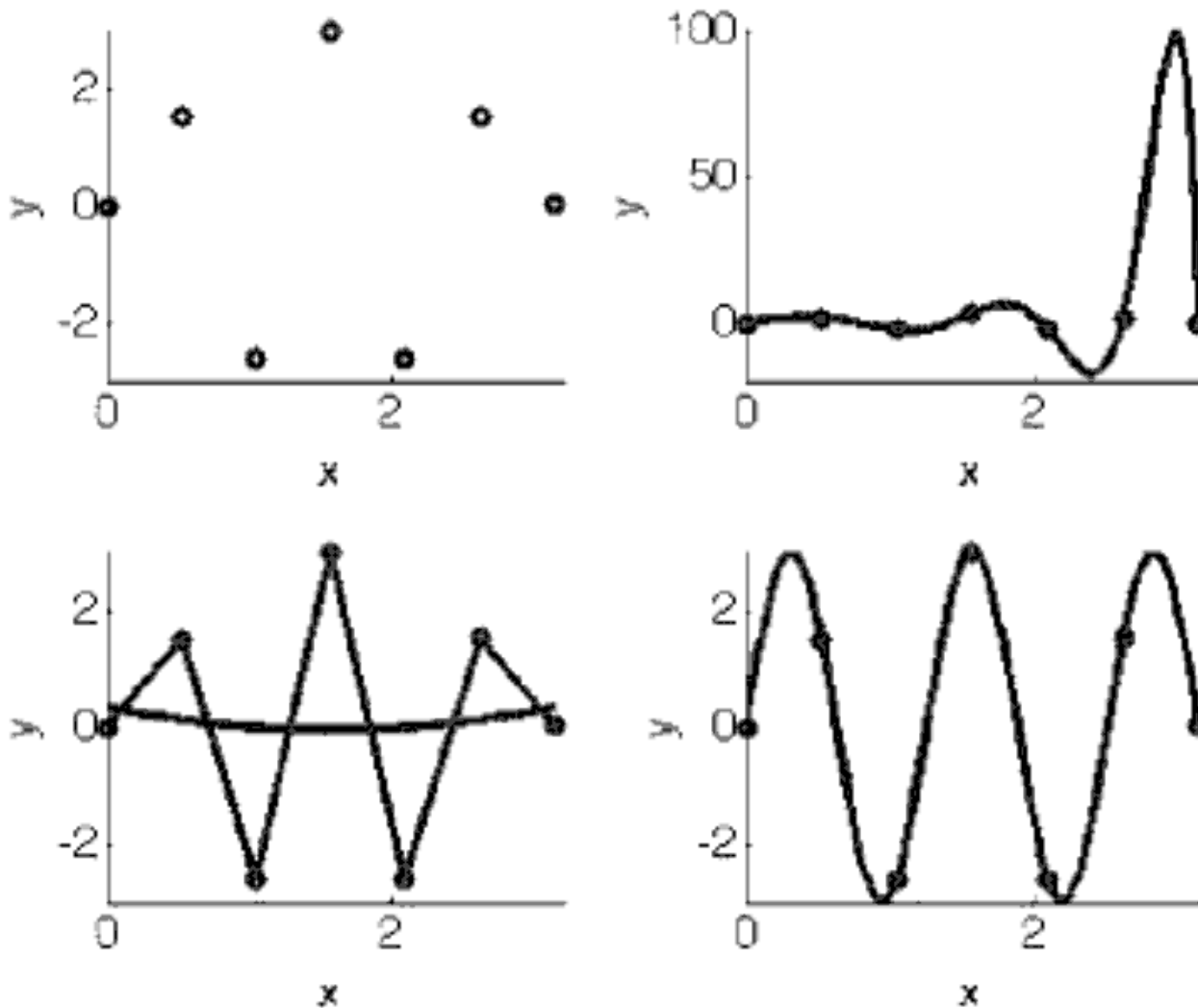
Supervised Learning: Regression

- Given $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$
- Learn a function $f(x)$ to predict y given x
 - y is real-valued == regression



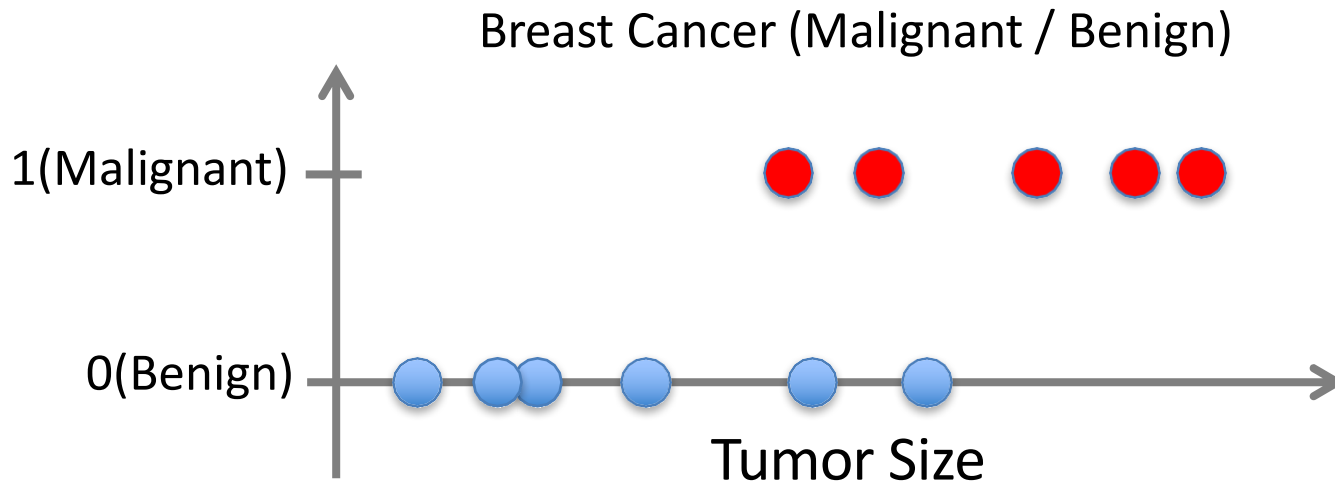
Sujata Pathak, IT, KJSCE

Supervised Learning: Regression



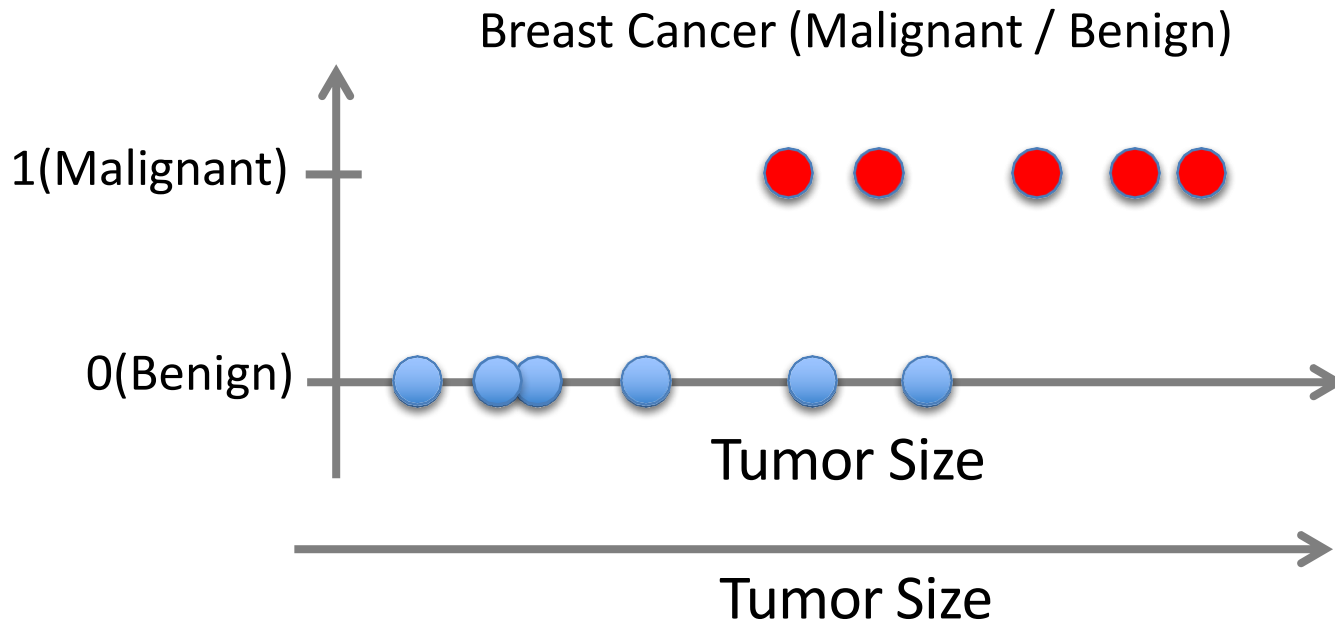
Supervised Learning: Classification

- Given $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$
- Learn a function $f(x)$ to predict y given x
 - y is categorical == classification



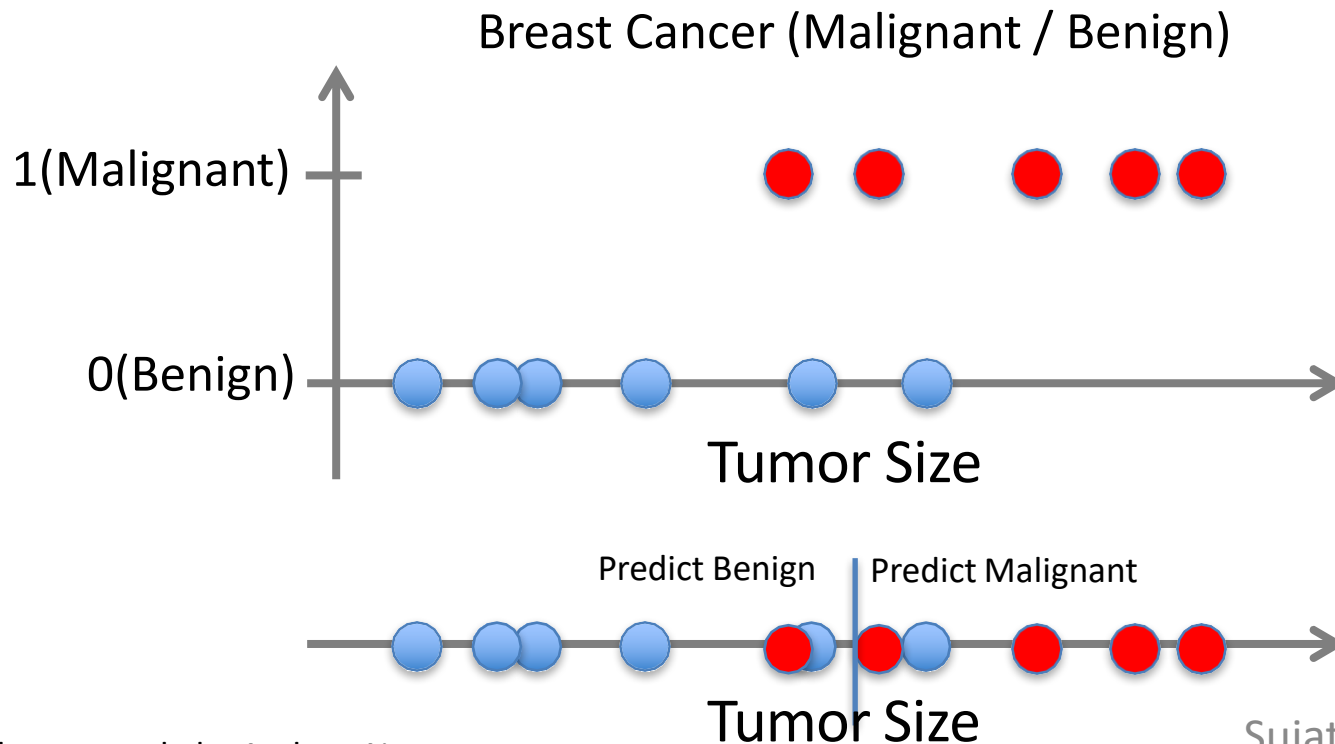
Supervised Learning: Classification

- Given $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$
- Learn a function $f(x)$ to predict y given x
 - y is categorical == classification

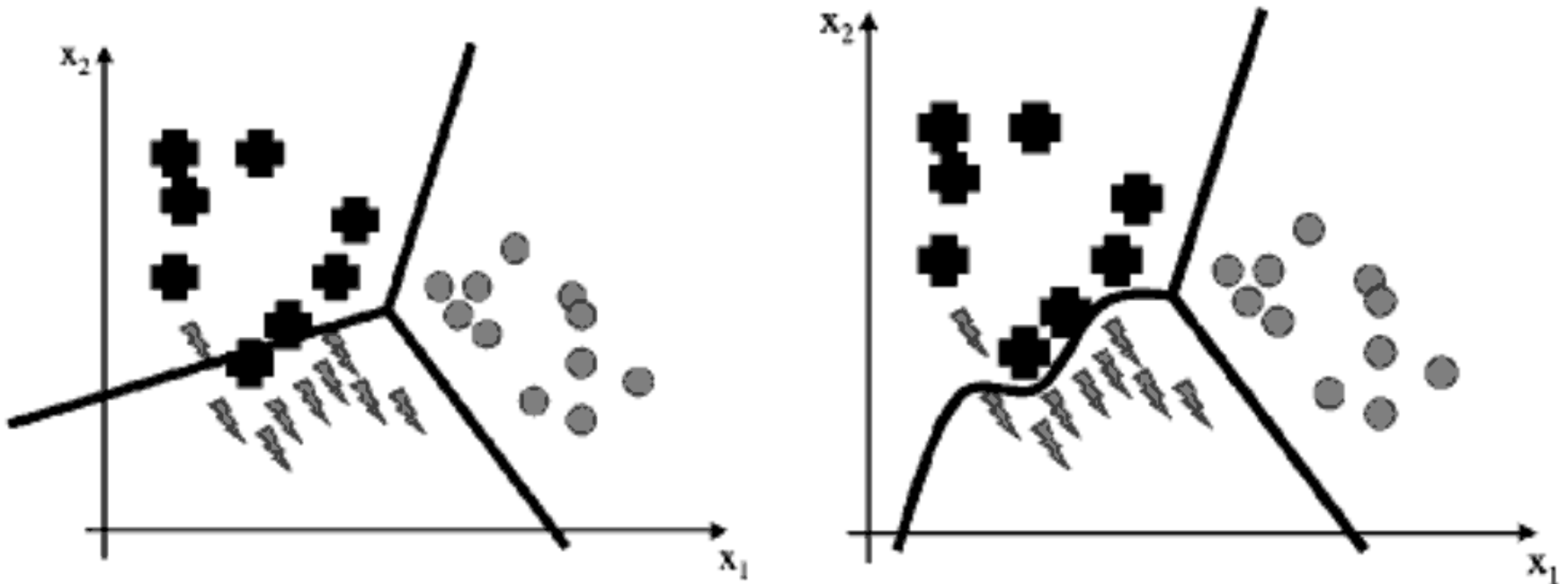


Supervised Learning: Classification

- Given $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$
- Learn a function $f(x)$ to predict y given x
 - y is categorical == classification

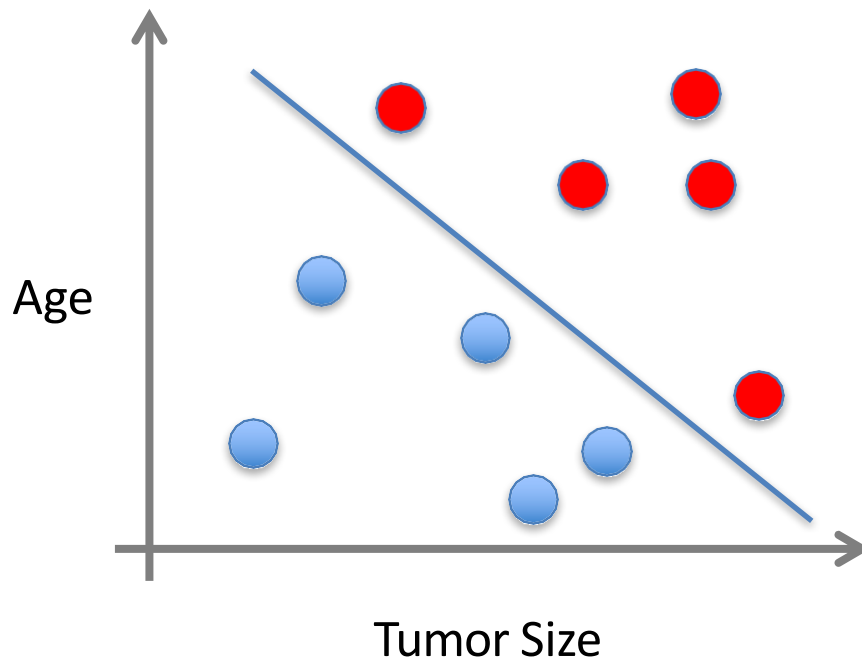


Supervised Learning: Classification



Supervised Learning

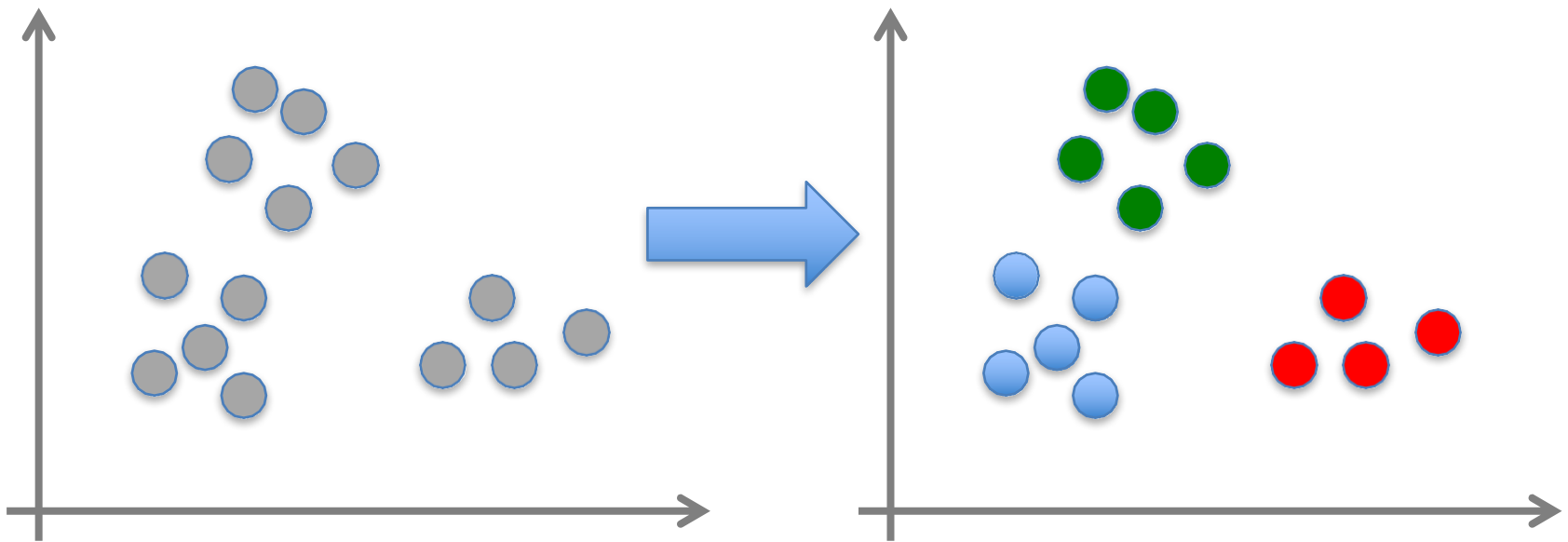
- x can be multi-dimensional
 - Each dimension corresponds to an attribute



- Clump Thickness
- Uniformity of Cell Size
- Uniformity of Cell Shape
- ...

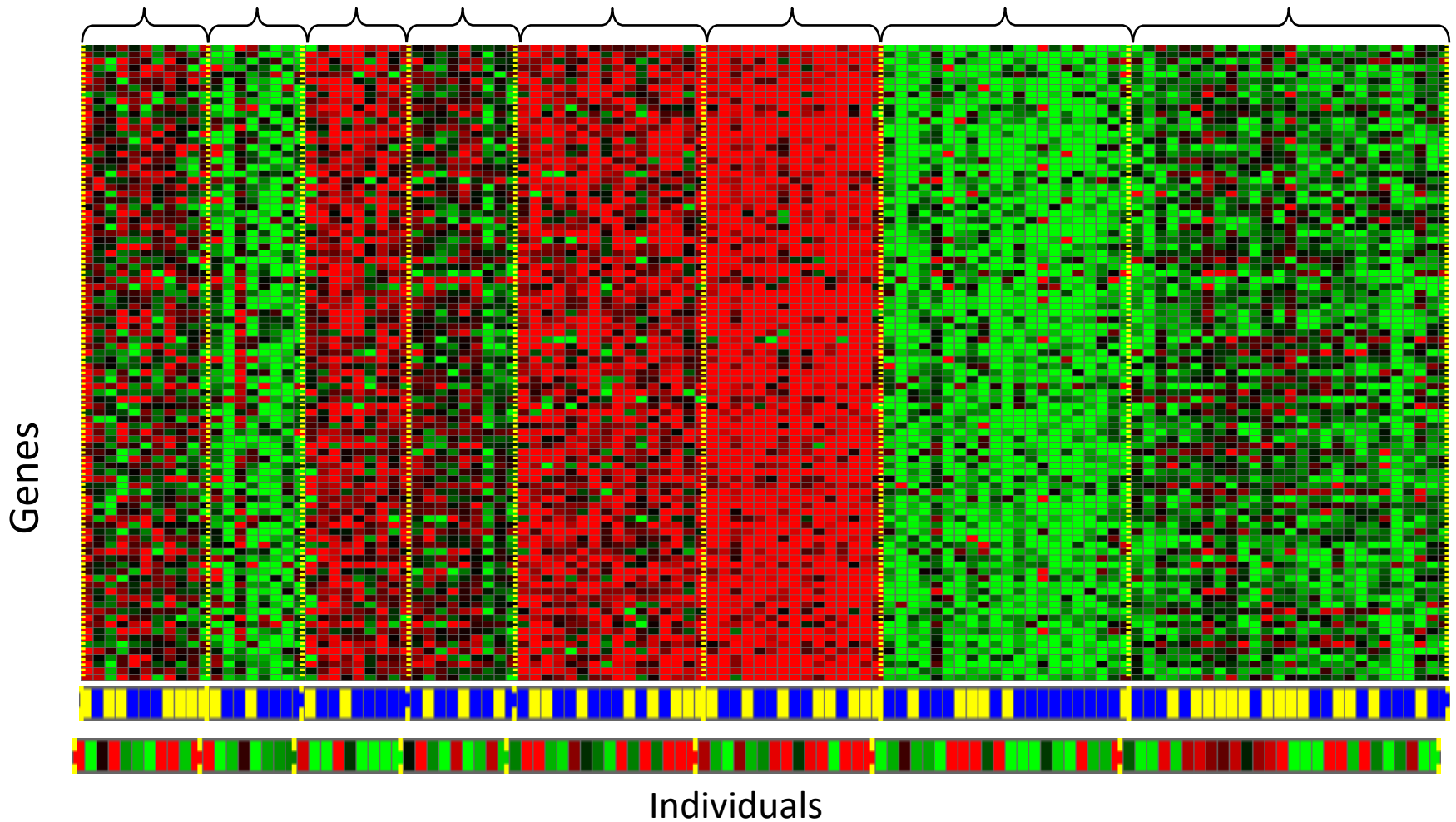
Unsupervised Learning

- Given x_1, x_2, \dots, x_n (without labels)
- Output hidden structure behind the x 's
 - E.g., clustering



Unsupervised Learning

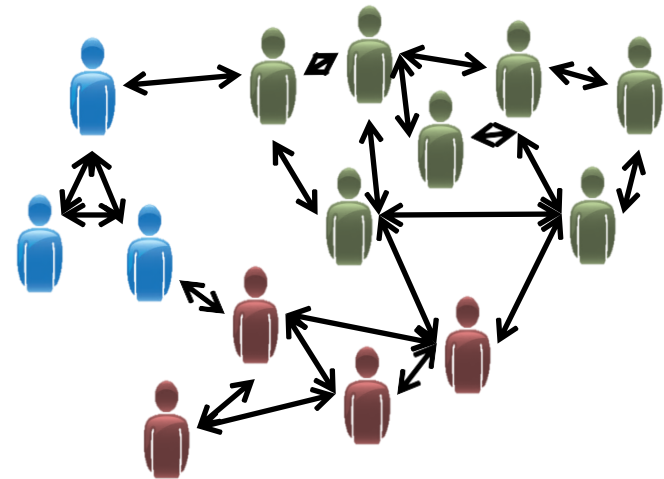
Genomics application: group individuals by genetic similarity



Unsupervised Learning



Organize computing clusters



Social network analysis



Market segmentation

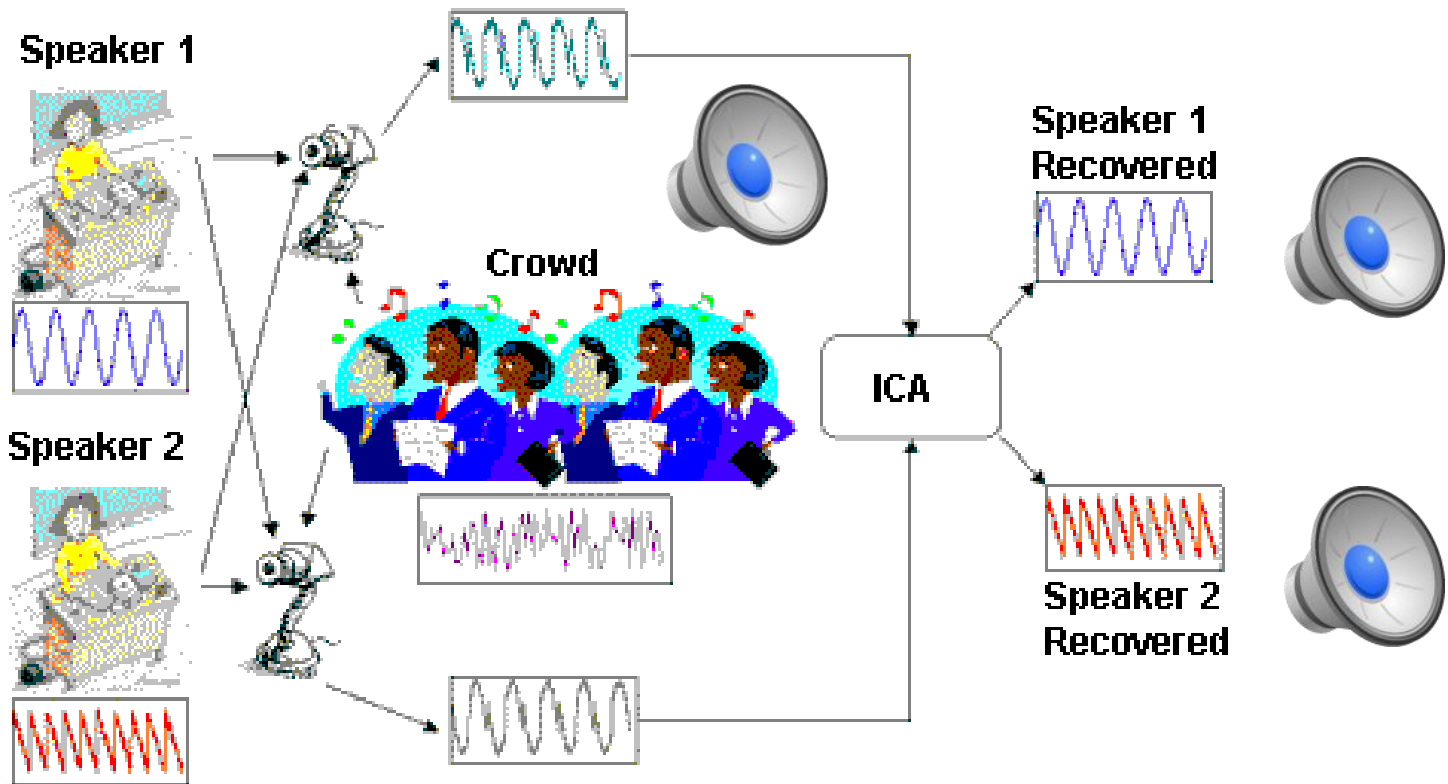


Image credit: NASA/JPL-Caltech/E. Churchwell (Univ. of Wisconsin, Madison)

Astronomical data analysis

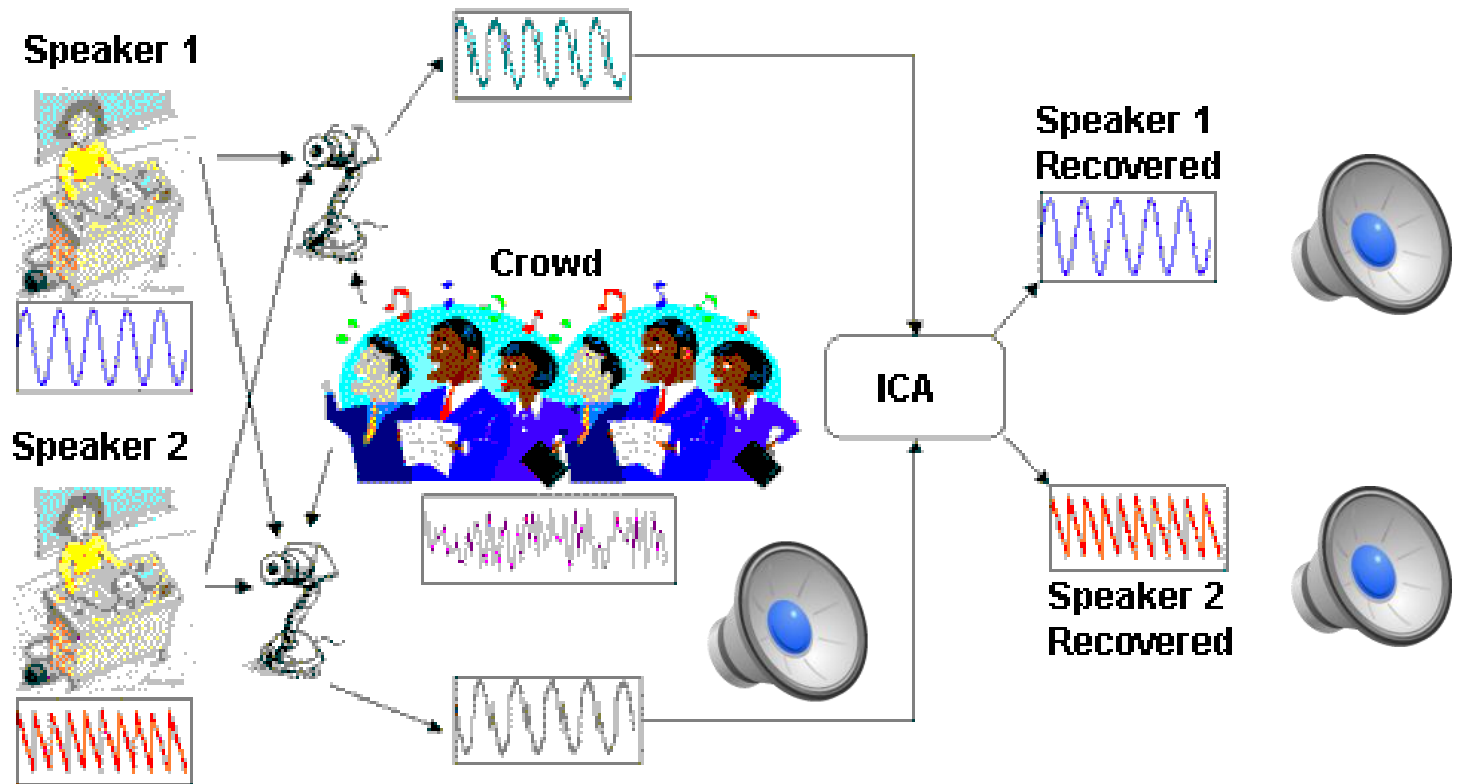
Unsupervised Learning

- Independent component analysis – separate a combined signal into its original sources



Unsupervised Learning

- Independent component analysis – separate a combined signal into its original sources



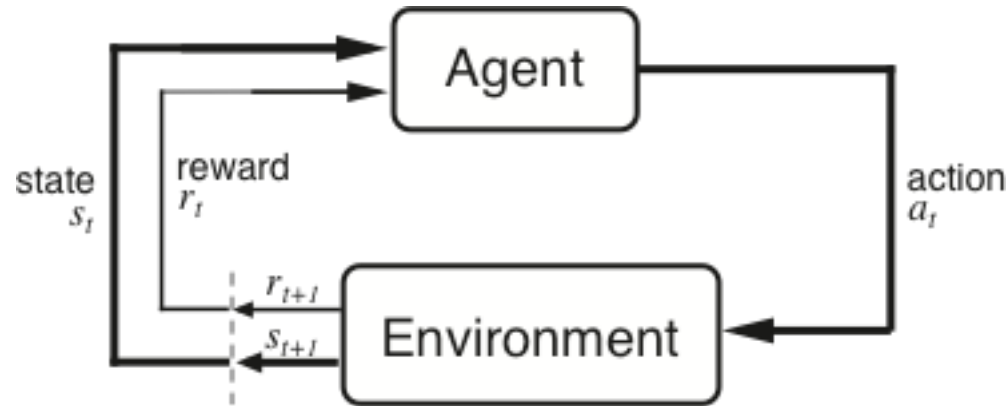
Sujata Pathak, IT, KJSCE

Image credit: statsoft.com Audio from <http://www.ism.ac.jp/~shiro/research/blindsep.html>

Reinforcement Learning

- Given a sequence of states and actions with (delayed) rewards, output a policy
 - Policy is a mapping from states \rightarrow actions that tells you what to do in a given state
- Examples:
 - Credit assignment problem
 - Game playing
 - Robot in a maze
 - Balance a pole on your hand

The Agent-Environment Interface



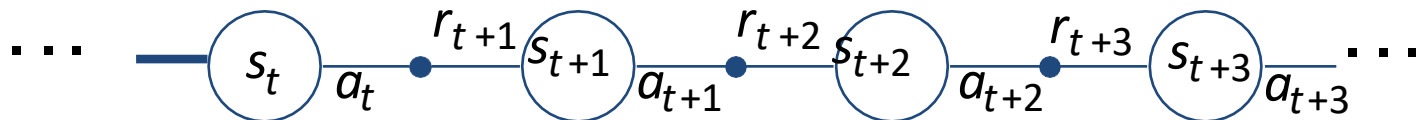
Agent and environment interact at discrete time steps : $t = 0, 1, 2, K$

Agent observes state at step t : $s_t \in \mathcal{S}$

produces action at step t : $a_t \in A(s_t)$

gets resulting reward : $r_{t+1} \in \mathcal{R}$

and resulting next state : s_{t+1}



Sujata Pathak, IT, KJSCE

Reinforcement Learning



<https://www.youtube.com/watch?v=4cgWya-wjgY>

Inverse Reinforcement Learning

- Learn policy from user demonstrations



Stanford Autonomous Helicopter

<http://heli.stanford.edu/>

<https://www.youtube.com/watch?v=VCdxqn0fcnE>

Sujata Pathak, IT, KJSCE

Generalization



Training set (labels known)



Test set (labels unknown)

How well does a learned model generalize from the data it was trained on to a new test set?

Generalization

Components of generalization error

Bias: how much the average model over all training sets differ from the true model?

Error due to inaccurate assumptions/simplifications made by the model

Variance: how much models estimated from different training sets differ from each other

Underfitting: model is too “simple” to represent all the relevant class characteristics

High bias and low variance

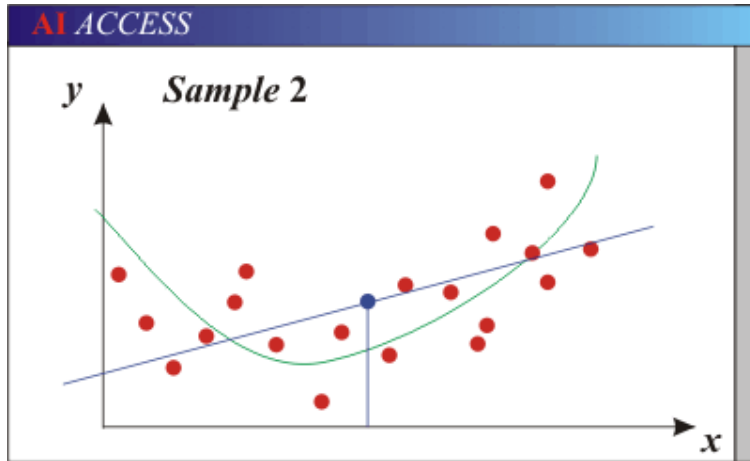
High training error and high-test error

Overfitting: model is too “complex” and fits irrelevant characteristics (noise) in the data

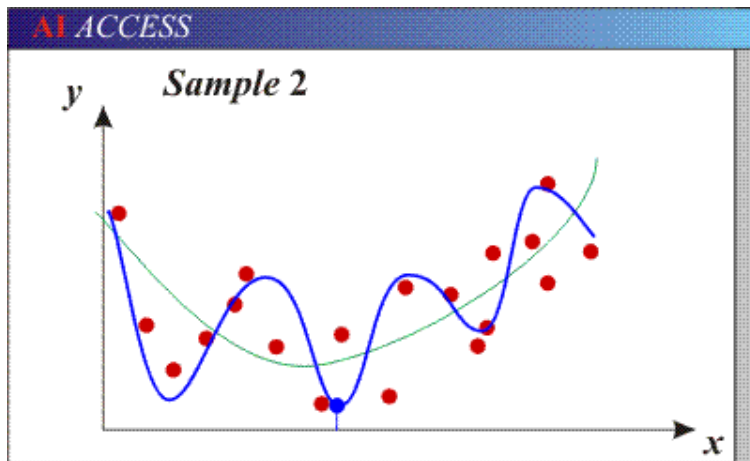
Low bias and high variance

Low training error and high-test error

Bias-Variance Trade-off



- Models with too few parameters are inaccurate because of a large bias (not enough flexibility).

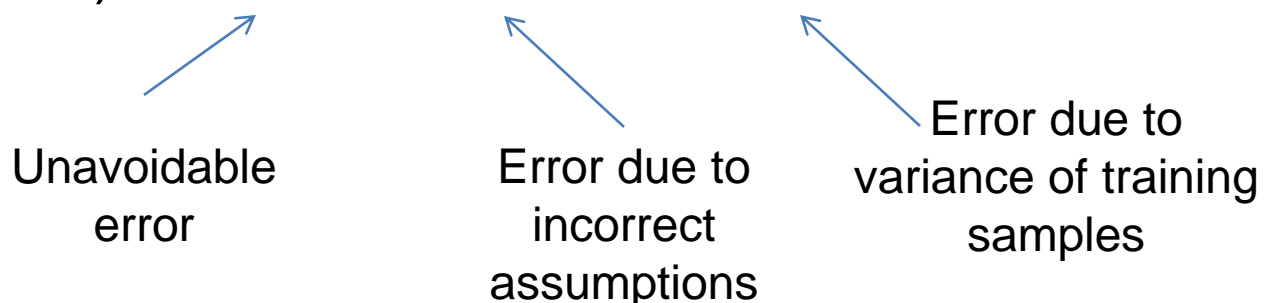


- Models with too many parameters are inaccurate because of a large variance (too much sensitivity to the sample).

Bias-Variance Trade-off

$$E(\text{MSE}) = \text{noise}^2 + \text{bias}^2 + \text{variance}$$

Unavoidable
error



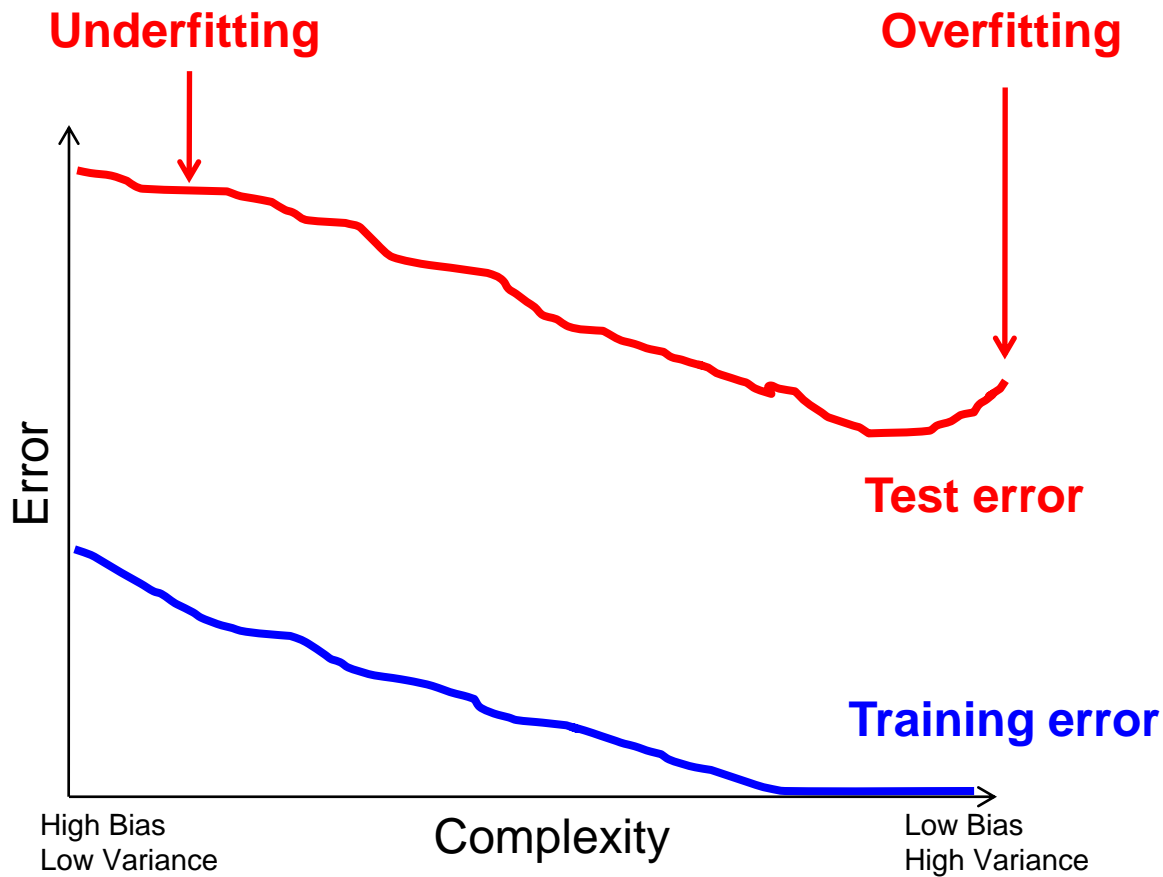
Error due to
incorrect
assumptions

Error due to
variance of training
samples

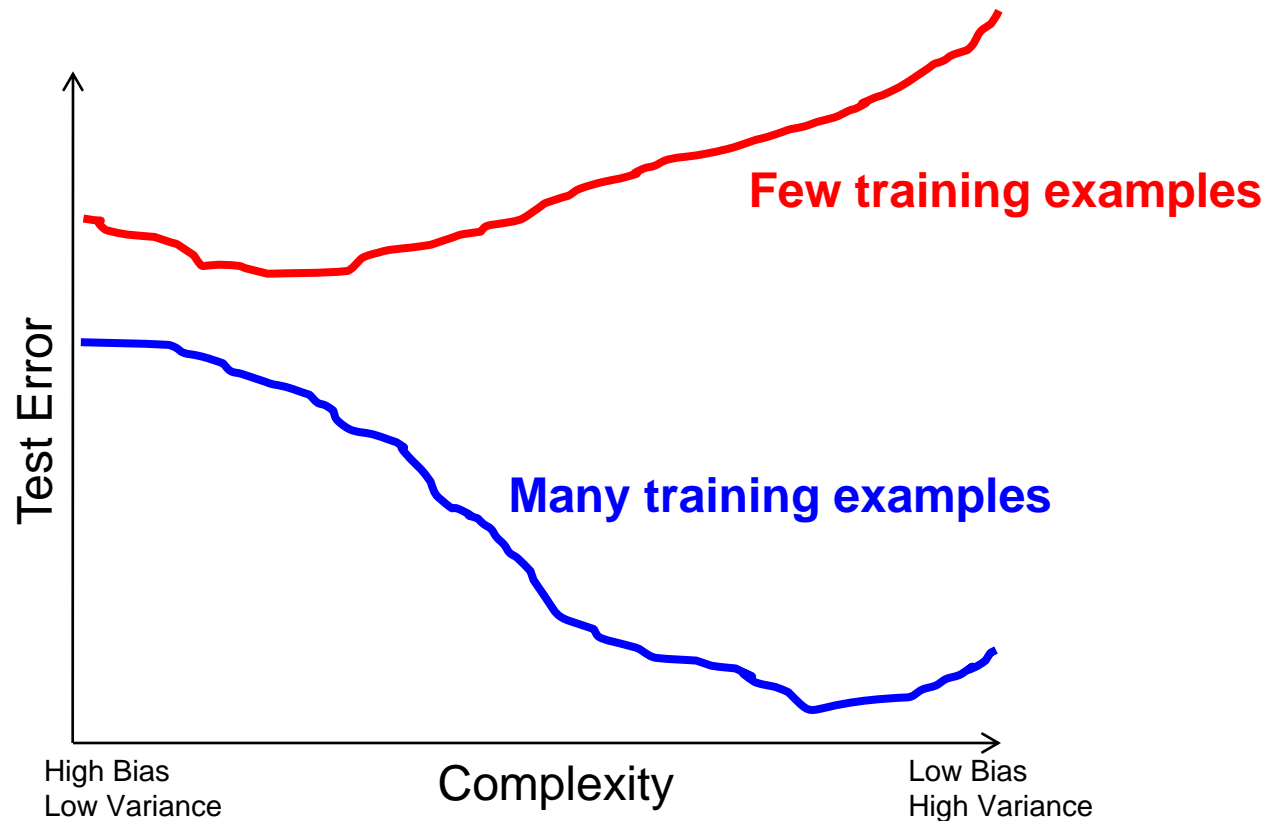
See the following for explanations of bias-variance (also Bishop's "Neural Networks" book):

- <http://www.inf.ed.ac.uk/teaching/courses/mlsc/Notes/Lecture4/BiasVariance.pdf>

Bias-variance tradeoff

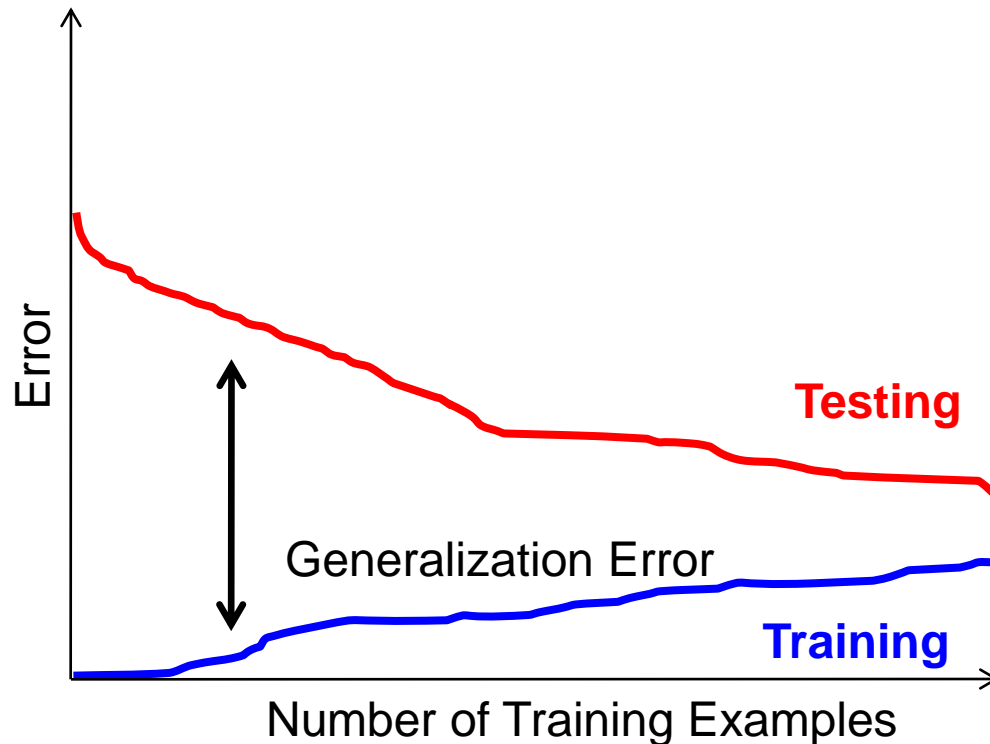


Bias-variance tradeoff

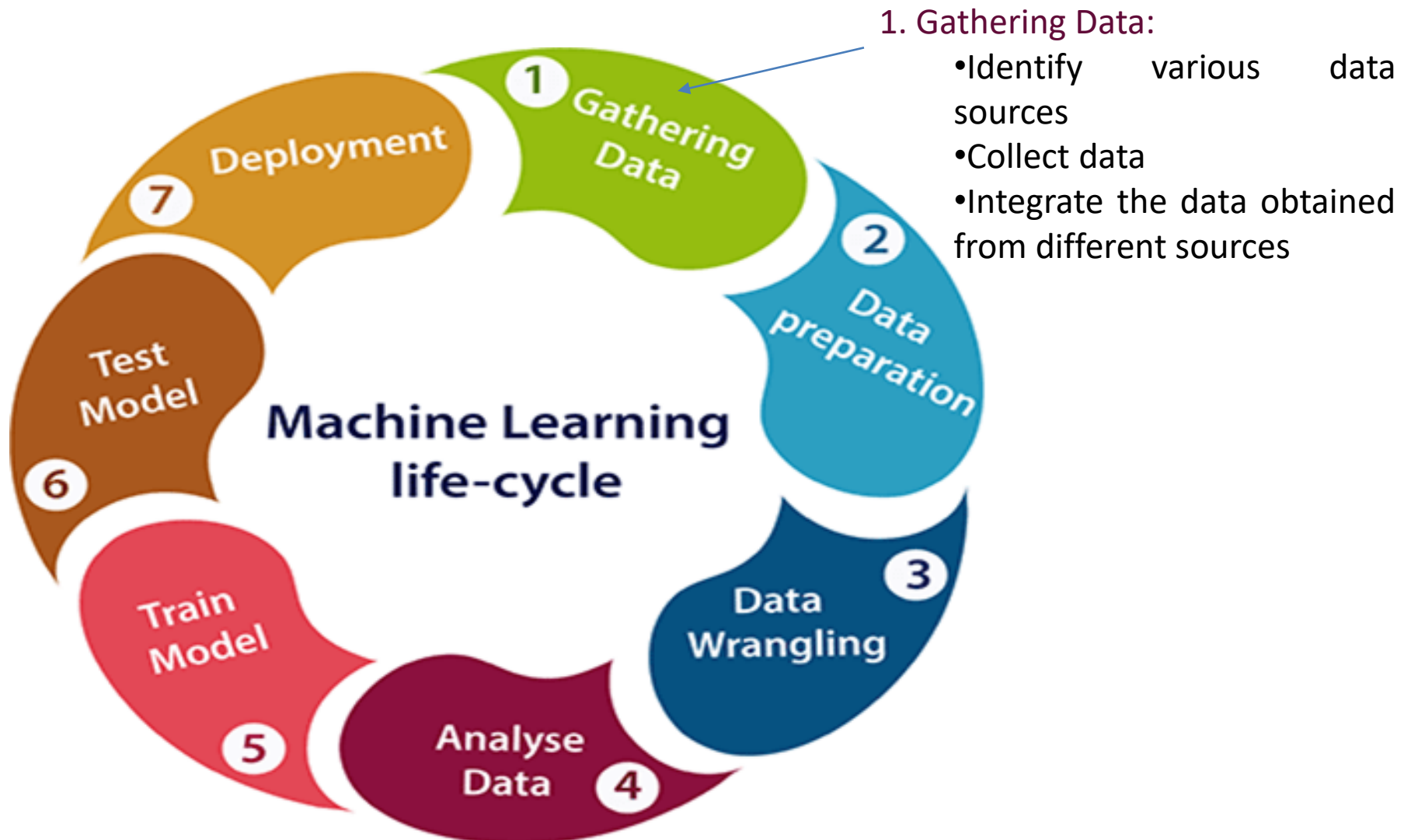


Effect of Training Size

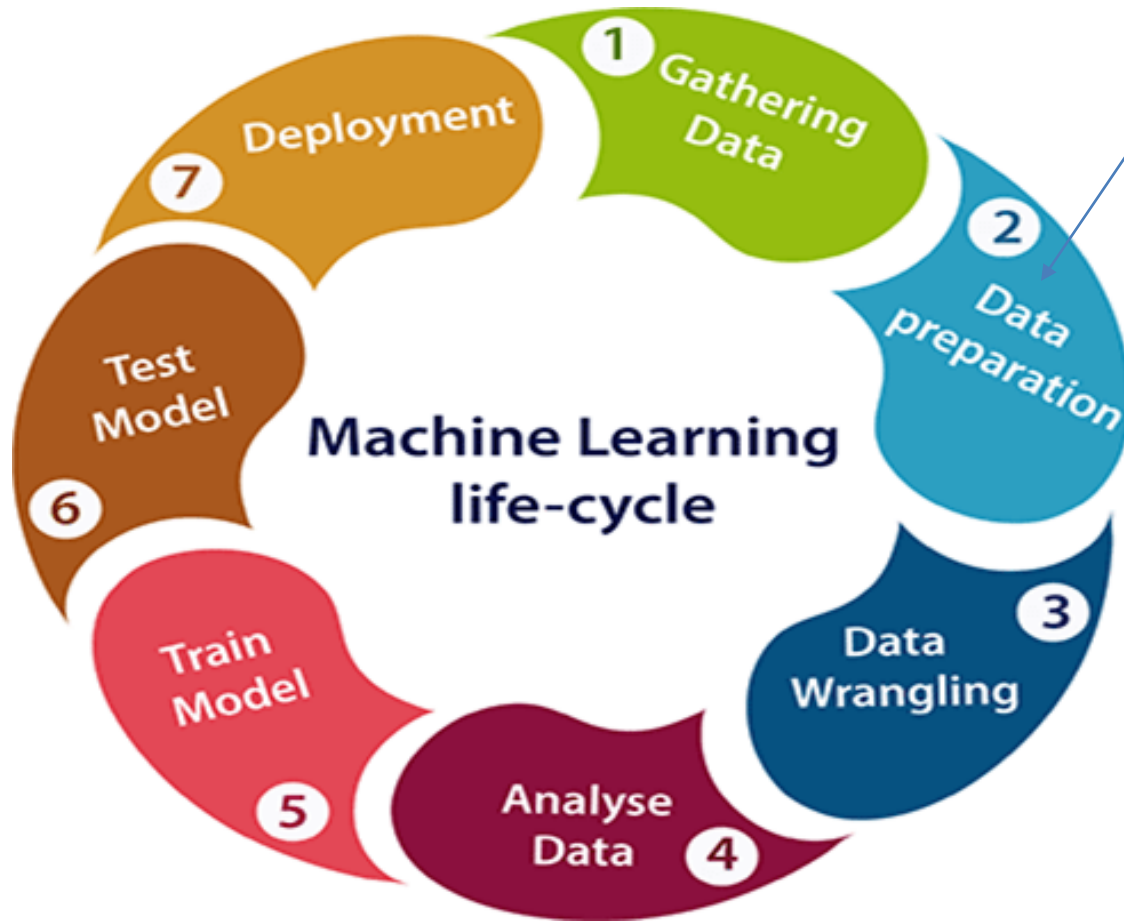
Fixed prediction model



Process of Machine learning



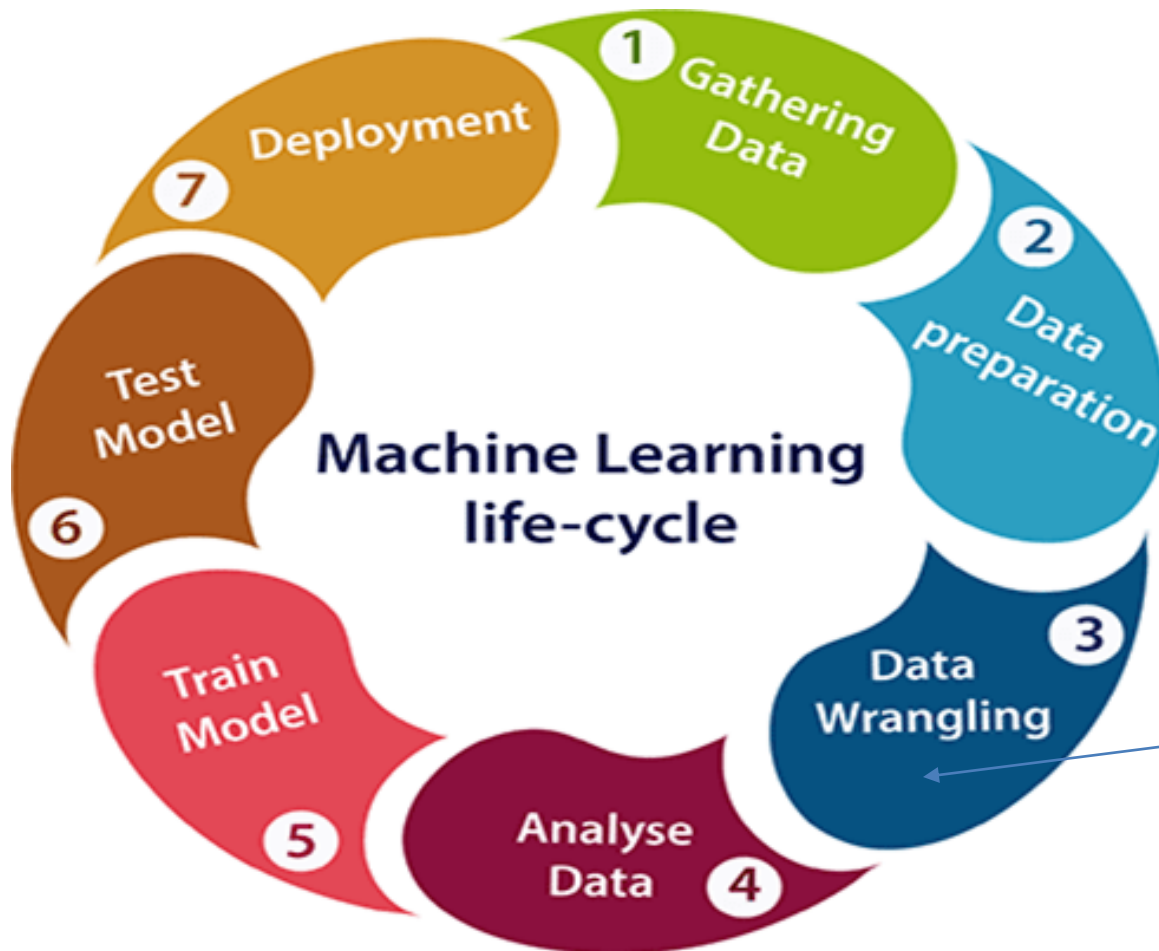
Process of Machine learning



Data exploration:

- Understand the nature of data.
- Understand the characteristics, format, and quality of data.
- find Correlations, general trends, and outliers.

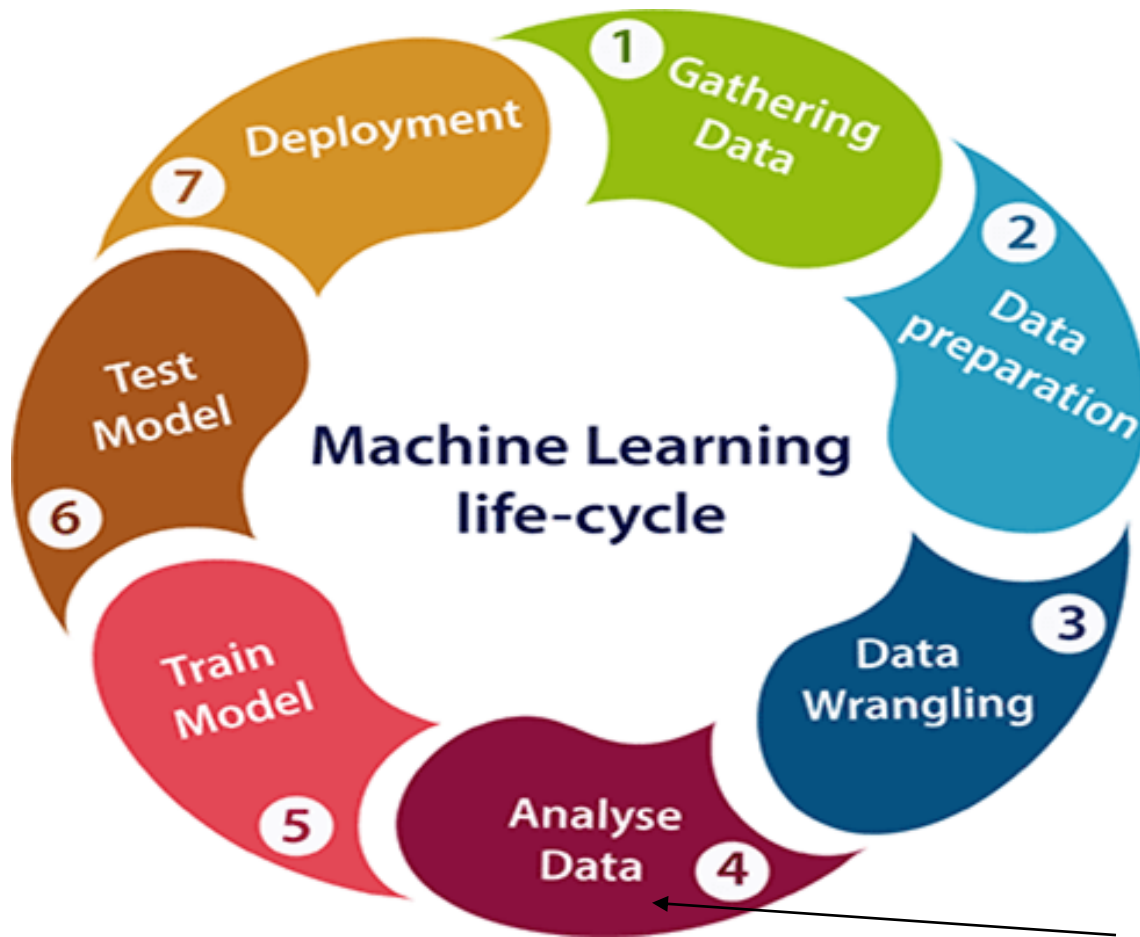
Process of Machine learning



Data Wrangling:

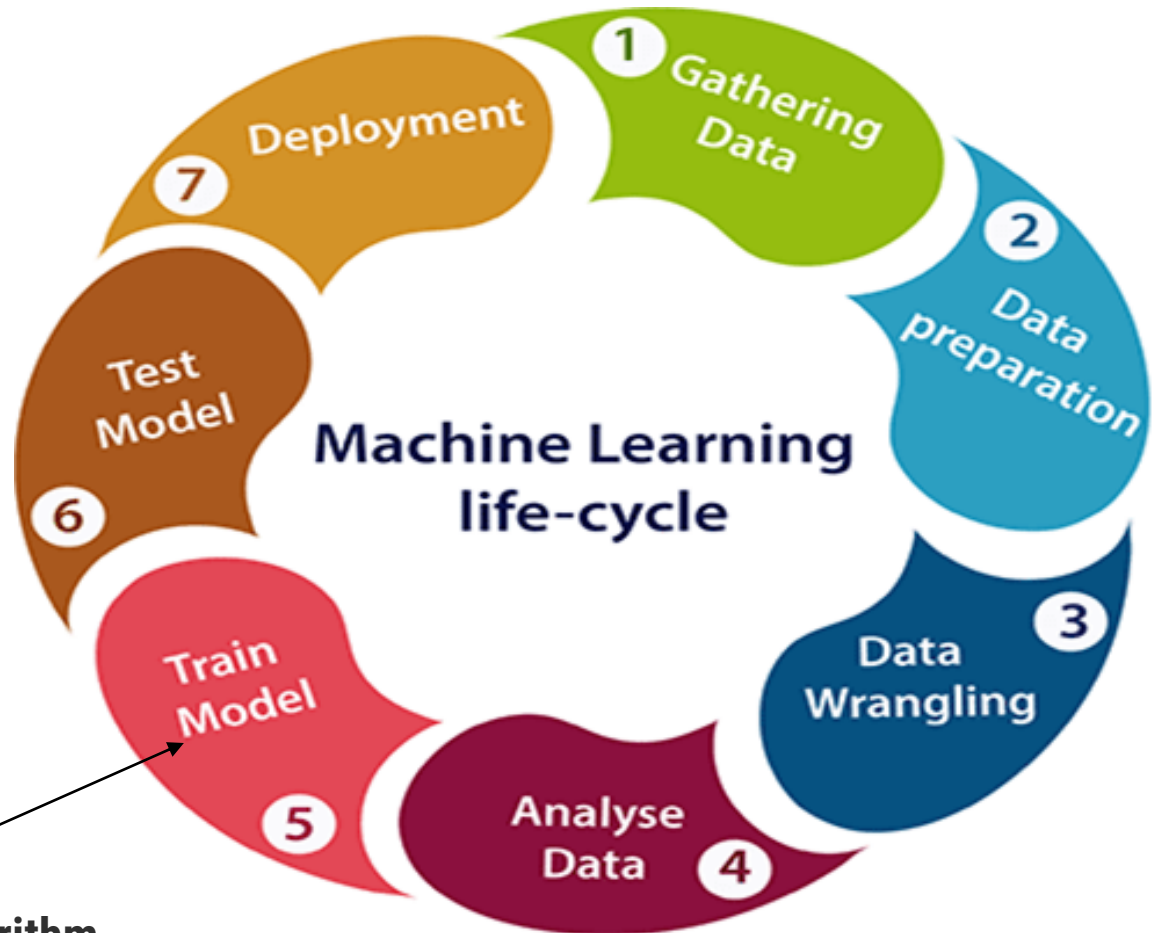
- Ignoring the missing values
- Removing instances having missing values from the dataset.
- Estimating the missing values of instances using mean, median or mode.
- Removing duplicate instances from the dataset.
- Normalizing the data in the dataset.

Process of Machine learning



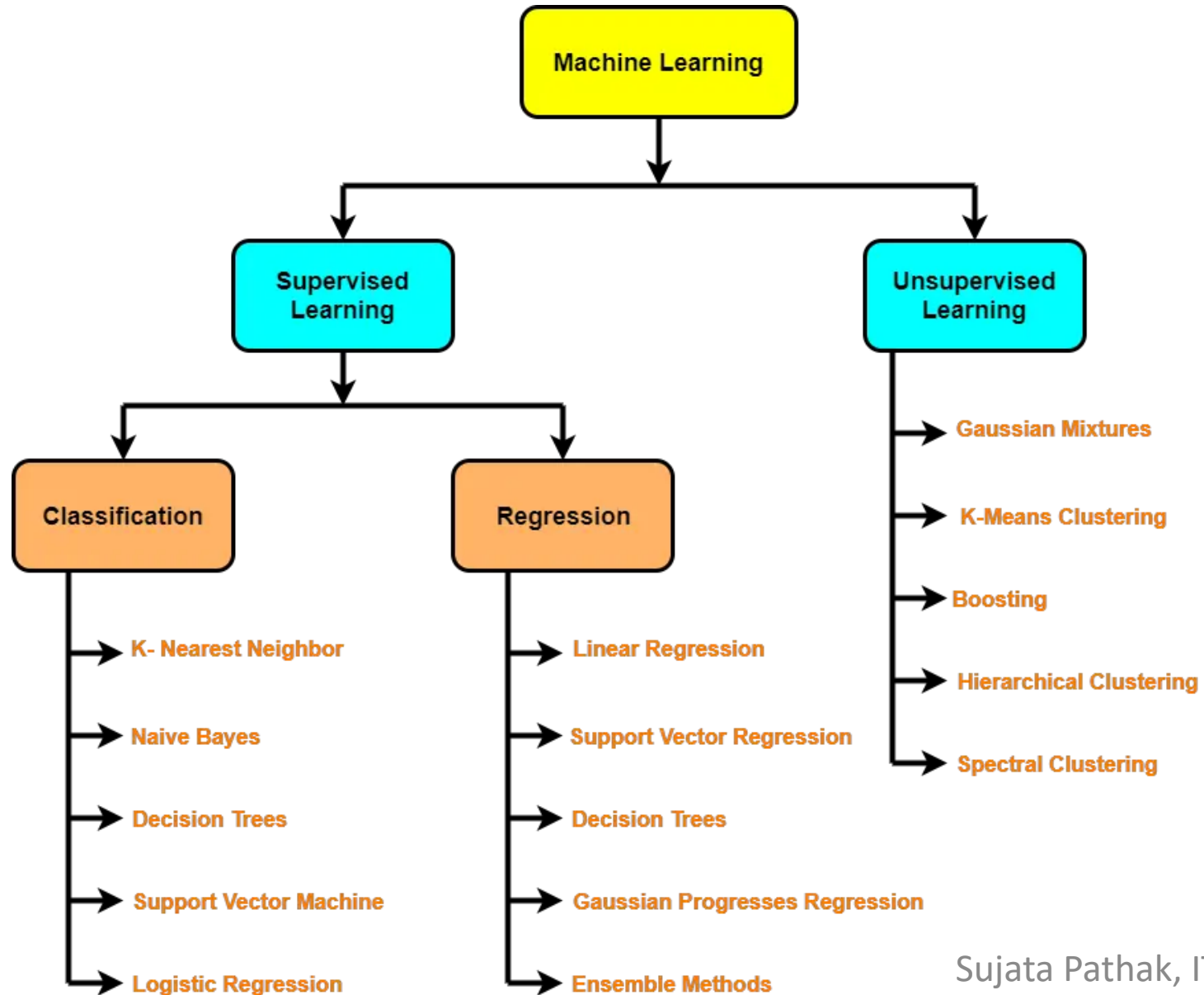
- Selection of analytical techniques
- Building models
- Review the result

Process of Machine learning



Choosing Learning Algorithm

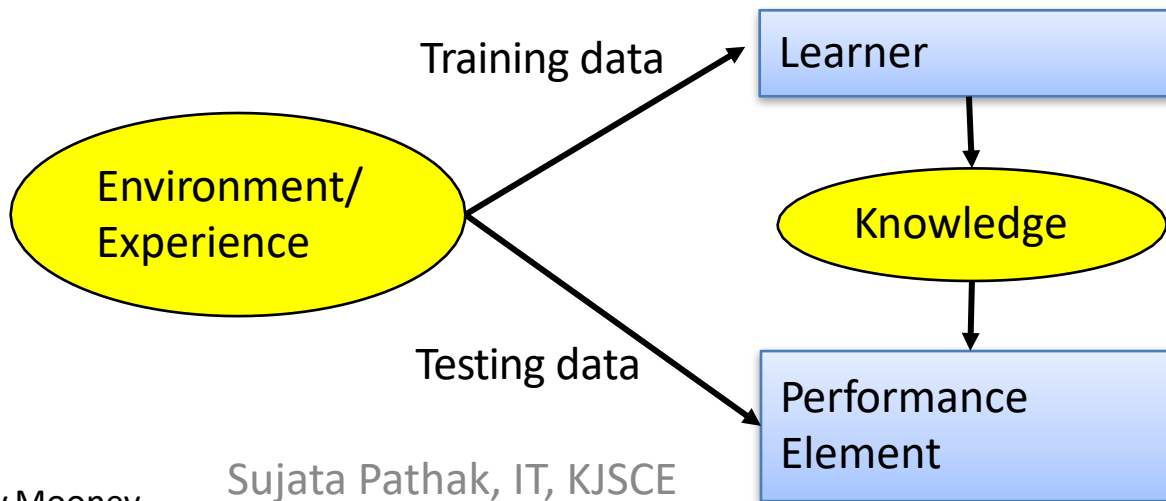
Process of Machine Learning



Framing a Learning Problem

Designing a Learning System

- Choose the training experience
- Choose exactly what is to be learned
 - i.e. the **target function**
- Choose how to represent the target function
- Choose a learning algorithm to infer the target function from the experience



Training vs. Test Distribution

- We generally assume that the training and test examples are independently drawn from the same overall distribution of data
 - We call this “i.i.d” which stands for “independent and identically distributed”
- If examples are not independent, requires *collective classification*
- If test distribution is different, requires *transfer learning*

Various Function Representations

- Numerical functions
 - Linear regression
 - Neural networks
 - Support vector machines
- Symbolic functions
 - Decision trees
 - Rules in propositional logic
 - Rules in first-order predicate logic
- Instance-based functions
 - Nearest-neighbor
 - Case-based
- Probabilistic Graphical Models
 - Naïve Bayes
 - Bayesian networks
 - Hidden-Markov Models (HMMs)
 - Probabilistic Context Free Grammars (PCFGs)
 - Markov networks

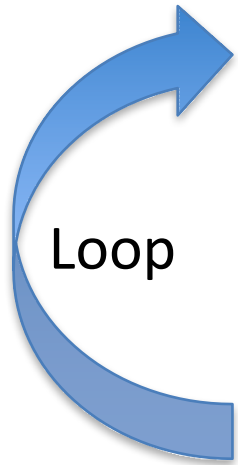
Various Search/Optimization Algorithms

- Gradient descent
 - Perceptron
 - Backpropagation
- Dynamic Programming
 - HMM Learning
 - PCFG Learning
- Divide and Conquer
 - Decision tree induction
 - Rule learning
- Evolutionary Computation
 - Genetic Algorithms (GAs)
 - Genetic Programming (GP)
 - Neuro-evolution

Evaluation

- Accuracy
- Precision and recall
- Squared error
- Likelihood
- Posterior probability
- Cost / Utility
- Margin
- Entropy
- K-L divergence
- etc.

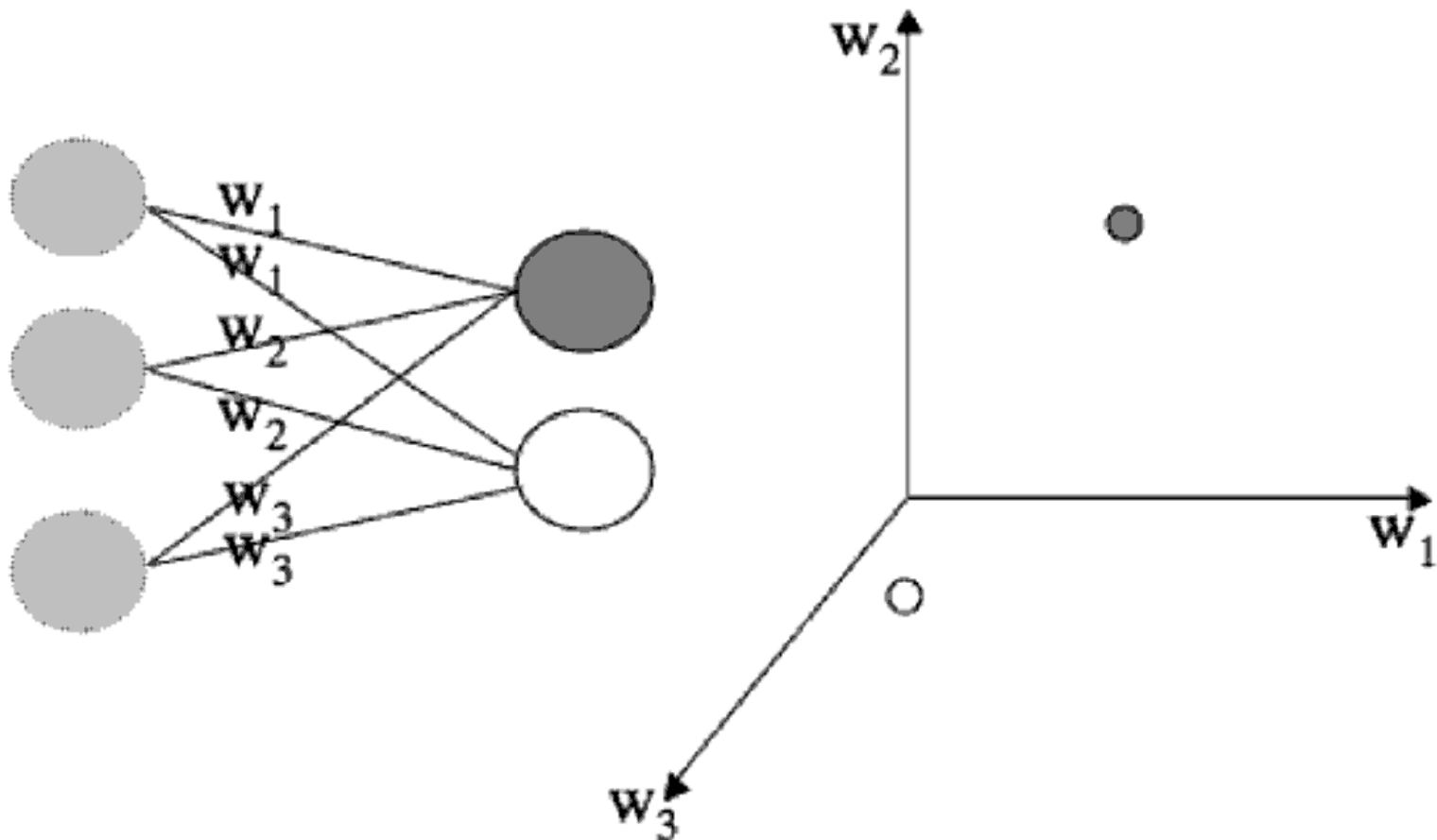
ML in Practice



- Understand domain, prior knowledge, and goals
- Data integration, selection, cleaning, pre-processing, etc.
- Learn models
- Interpret results
- Consolidate and deploy discovered knowledge

Introduction to terminologies

Weight Space



Introduction to terminologies

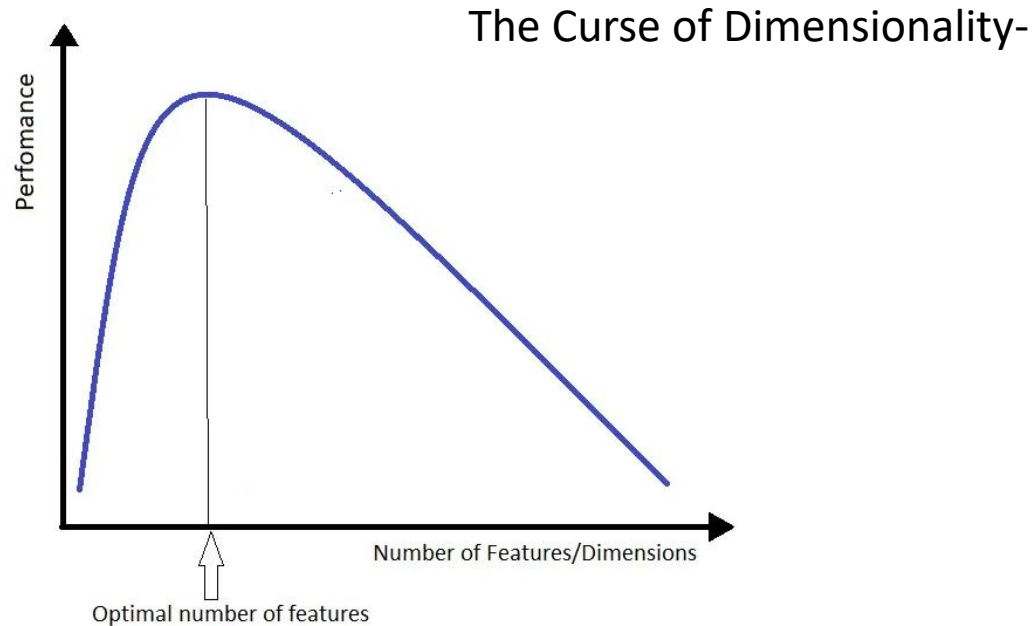
The Curse of Dimensionality-

- As the number of dimensions or features increases, the amount of data needed to generalize the machine learning model accurately increases exponentially.

Need for Data Points with Increase in Dimensions

1 Binary feature	→	2^1 unique values	→	$2^1 \times 10 = 20$ data points
2 Binary features	→	2^2 unique values	→	$2^2 \times 10 = 40$ data points
3 Binary features	→	2^3 unique values	→	$2^3 \times 10 = 80$ data points
.		.		.
.		.		.
.		.		.
k Binary features	→	2^k unique values	→	$2^k \times 10$ data points

Introduction to terminologies



In one-dimensional, 2D, or even 3D data space

$$(dist_{\max(A)} - dist_{\min(A)}) / (dist_{\min(A)}) > 0$$

But, as the dimensions increase, that is as $\dim \rightarrow \infty$;

$$\lim_{\dim \rightarrow \infty} (dist_{\max(A)} - dist_{\min(A)}) / (dist_{\min(A)}) \rightarrow 0$$

Sujata Pathak, IT, KJSCE

Introduction to terminologies

The Curse of Dimensionality-

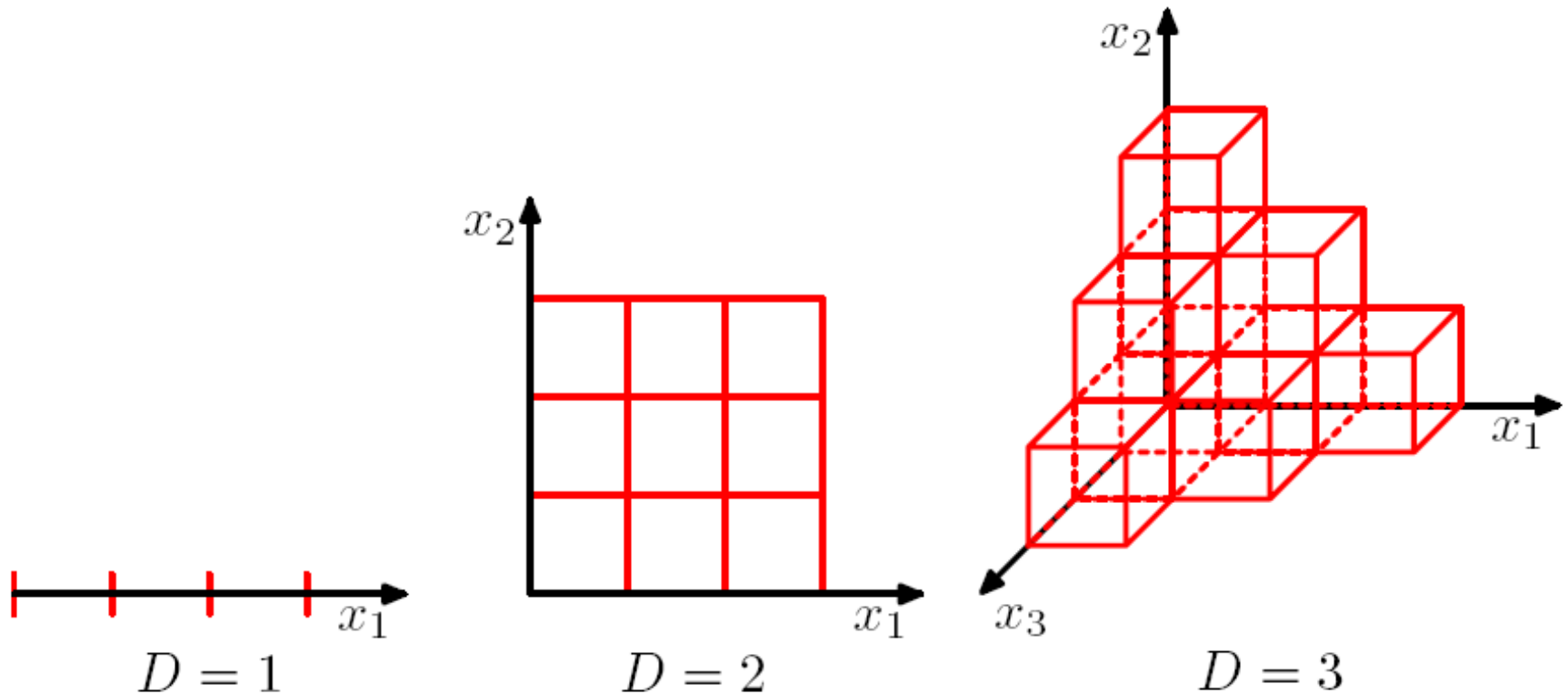


Illustration of the curse of dimensionality, showing how the number of regions of a regular grid grows exponentially with the dimensionality D of the space. For clarity, only a subset of the cubical regions are shown for $D = 3$.

Lessons Learned about Learning

- Learning can be viewed as using direct or indirect experience to approximate a chosen target function.
- Function approximation can be viewed as a search through a space of hypotheses (representations of functions) for one that best fits a set of training data.
- Different learning methods assume different hypothesis spaces (representation languages) and/or employ different search techniques.

Classification

Classification and Prediction

- What is classification? What is prediction?
- Issues regarding classification and prediction
- Classification by decision tree induction
- Bayesian Classification
- Other Classification Methods
- Prediction
- Classification accuracy
- Summary

Classification vs. Prediction

□ Classification:

- predicts categorical class labels
- classifies data (constructs a model) based on the training set and the values (**class labels**) in a classifying attribute and uses it in classifying new data

□ Prediction:

- models continuous-valued functions, i.e., predicts unknown or missing values

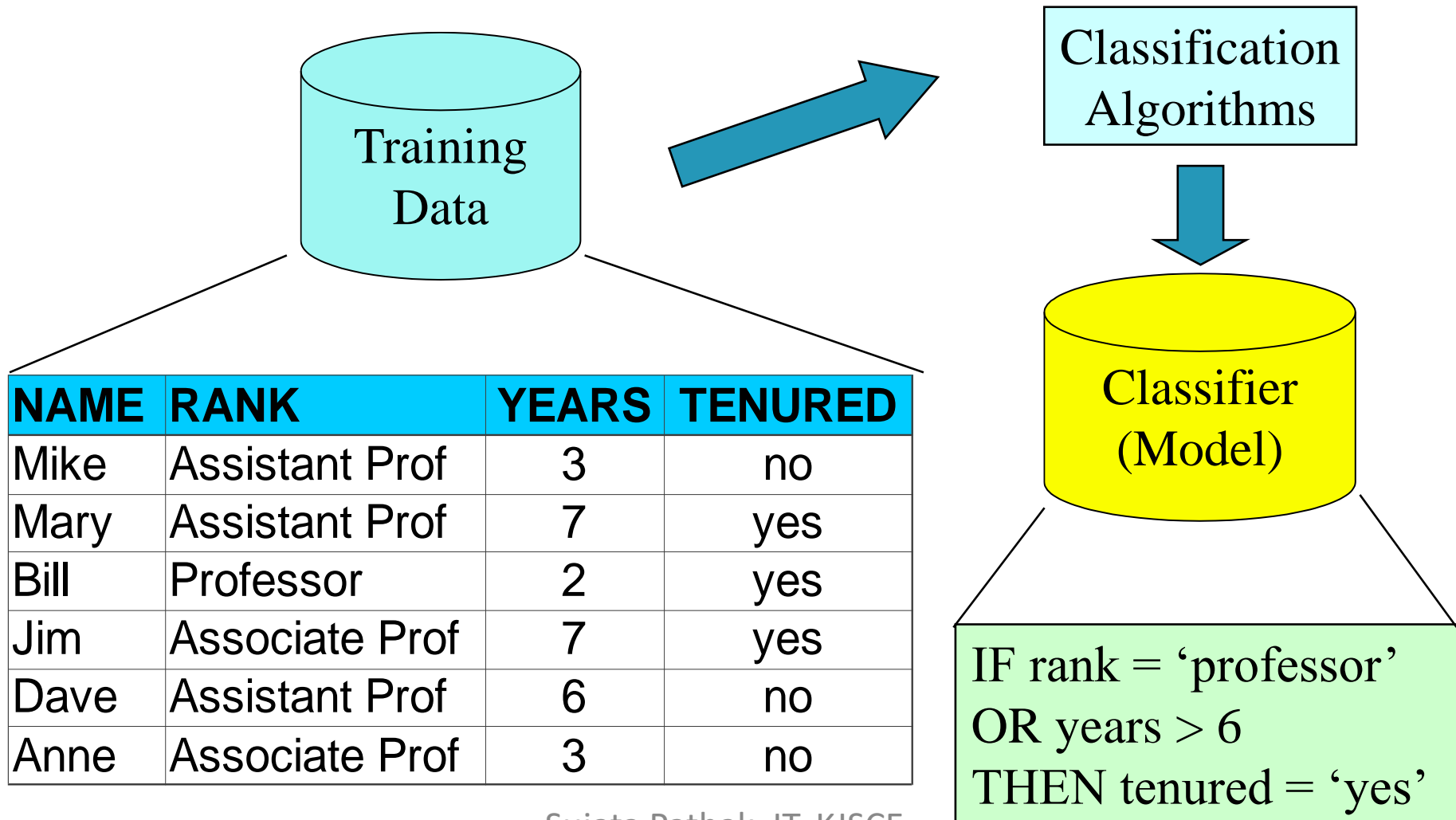
□ Typical Applications

- credit approval
- target marketing
- medical diagnosis
- treatment effectiveness analysis

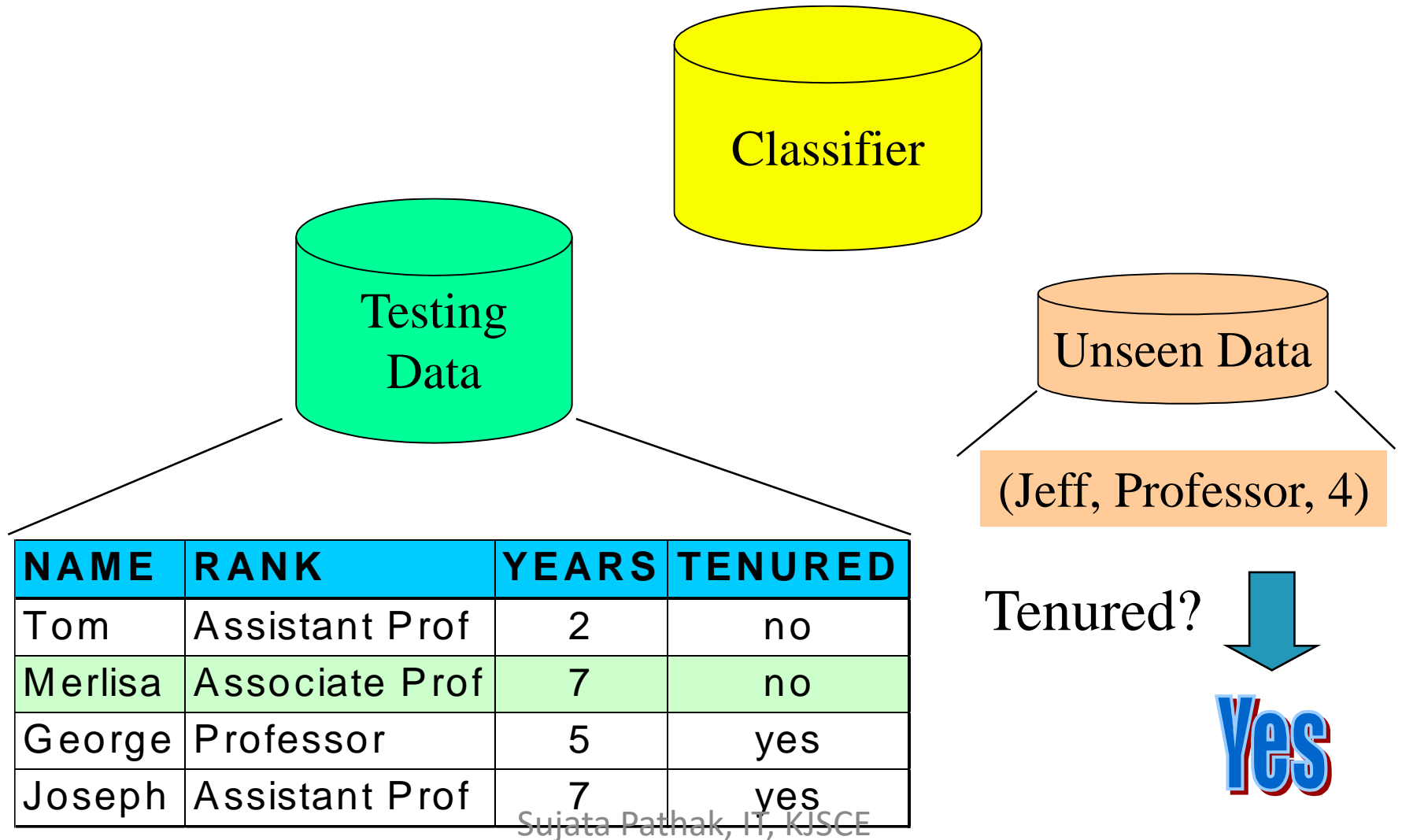
Classification—A Two-Step Process

- Model construction: describing a set of predetermined classes
 - Each tuple/sample is assumed to belong to a predefined class, as determined by the class label attribute
 - The set of tuples used for model construction: training set
 - The model is represented as classification rules, decision trees, or mathematical formulae
- Model usage: for classifying future or unknown objects
 - Estimate accuracy of the model
 - The known label of test sample is compared with the classified result from the model
 - Accuracy rate is the percentage of test set samples that are correctly classified by the model
 - Test set is independent of training set, otherwise over-fitting will occur

Classification Process (1): Model Construction



Classification Process (2): Use the Model in Prediction



Classification and Prediction

- What is classification? What is prediction?
- Issues regarding classification and prediction
- Classification by decision tree induction
- Bayesian Classification
- Classification by backpropagation
- Classification based on concepts from association rule mining
- Other Classification Methods
- Prediction
- Classification accuracy
- Summary

Classification Techniques

A number of classification techniques are known, which can be broadly classified into the following categories:

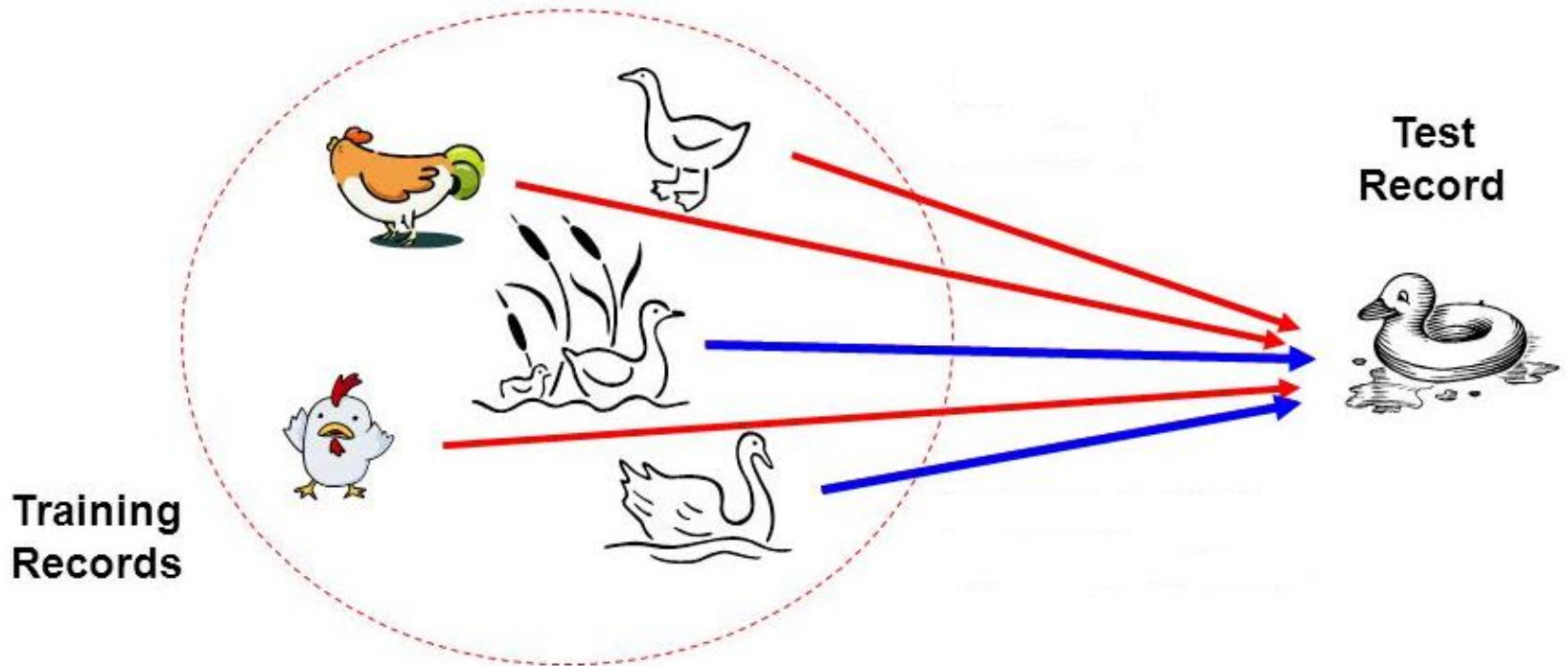
1. Statistical-Based Methods
 - Regression
 - Bayesian Classifier
2. Distance-Based Classification
 - K-Nearest Neighbours
3. Decision Tree-Based Classification
 - ID3, C 4.5, CART
5. Classification using Machine Learning (SVM)
6. Classification using Neural Network (ANN)

Bayesian Classifier

Bayesian Classifier

Principle

If it walks like a duck, quacks like a duck, then it is **probably** a duck



Bayesian Classifier

A statistical classifier

Performs *probabilistic prediction*, i.e., predicts class membership probabilities

Foundation

Based on Bayes' Theorem.

Assumptions

- 1.The classes are *mutually exclusive and exhaustive*.
- 2.The attributes are *independent* given the class.

Called “Naïve” classifier because of these assumptions

Empirically proven to be useful.

Scales very well.

Prior and Posterior Probabilities

$P(A)$ and $P(B)$ are called prior probabilities

$P(A|B)$, $P(B|A)$ are called posterior probabilities

Example 8.6: Prior versus Posterior Probabilities

This table shows that the event Y has two outcomes namely A and B , which is dependent on another event X with various outcomes like x_1 , x_2 and x_3 .

Case1: Suppose, we don't have any information of the event A . Then, from the given sample space, we can calculate $P(Y = A) = \frac{5}{10} = 0.5$

Case2: Now, suppose, we want to calculate $P(X = x_2/Y = A) = \frac{2}{5} = 0.4$.

The later is the conditional or posterior probability, where as the former is the prior probability.

X	Y
x_1	A
x_2	A
x_3	B
x_3	A
x_2	B
x_1	A
x_1	B
x_3	B
x_2	B
x_2	A

Naïve Bayesian Classifier

Suppose, Y is a class variable and $X = \{X_1, X_2, \dots, X_n\}$ is a set of attributes, with instance of Y .

INPUT (X)	CLASS(Y)
... ..	
...
x_1, x_2, \dots, x_n	y_i
...

The classification problem, then can be expressed as the class-conditional probability

$$P(Y = y_i | (X_1 = x_1) \text{ AND } (X_2 = x_2) \text{ AND } \dots (X_n = x_n))$$

Naïve Bayesian Classifier

Naïve Bayesian classifier calculate this posterior probability using Bayes' theorem, which is as follows.

From Bayes' theorem on conditional probability, we have

$$P(Y|X) = \frac{P(X|Y) \cdot P(Y)}{P(X)}$$
$$= \frac{P(X|Y) \cdot P(Y)}{P(X|Y = y_1) \cdot P(Y = y_1) + \dots + P(X|Y = y_k) \cdot P(Y = y_k)}$$

where,

$$P(X) = \sum_{i=1}^k P(X|Y = y_i) \cdot P(Y = y_i)$$

Note:

- $P(X)$ is called the evidence (also the total probability) and it is a constant.
- The probability $P(Y|X)$ (also called class conditional probability) is therefore proportional to $P(X|Y) \cdot P(Y)$.
- Thus, $P(Y|X)$ can be taken as a measure of Y given that X .

$$P(Y|X) \approx P(X|Y) \cdot P(Y)$$

Naïve Bayesian Classifier

Suppose, for a given instance of X (say $x = (X_1 = x_1)$ and $(X_n = x_n)$).

There are any two class conditional probabilities namely $P(Y = y_i | X=x)$ and $P(Y = y_j | X=x)$.

If $P(Y = y_i | X=x) > P(Y = y_j | X=x)$, then we say that y_i is more stronger than y_j for the instance $X = x$.

The strongest y_i is the classification for the instance $X = x$.

Example

- Example: Play Tennis

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Example

- Test Phase

- Given a new instance,

$\mathbf{x}' = (\text{Outlook}=\text{Sunny}, \text{Temperature}=\text{Cool}, \text{Humidity}=\text{High}, \text{Wind}=\text{Strong})$

- Look up tables

$$P(\text{Outlook}=\text{Sunny} \mid \text{Play}=\text{Yes}) = 2/9$$

$$P(\text{Outlook}=\text{Sunny} \mid \text{Play}=\text{No}) = 3/5$$

$$P(\text{Temperature}=\text{Cool} \mid \text{Play}=\text{Yes}) = 3/9$$

$$P(\text{Temperature}=\text{Cool} \mid \text{Play}=\text{No}) = 1/5$$

$$P(\text{Humidity}=\text{High} \mid \text{Play}=\text{Yes}) = 3/9$$

$$P(\text{Humidity}=\text{High} \mid \text{Play}=\text{No}) = 4/5$$

$$P(\text{Wind}=\text{Strong} \mid \text{Play}=\text{Yes}) = 3/9$$

$$P(\text{Wind}=\text{Strong} \mid \text{Play}=\text{No}) = 3/5$$

$$P(\text{Play}=\text{Yes}) = 9/14$$

$$P(\text{Play}=\text{No}) = 5/14$$

- MAP rule

$$P(\text{Yes} \mid \mathbf{x}'): [P(\text{Sunny} \mid \text{Yes})P(\text{Cool} \mid \text{Yes})P(\text{High} \mid \text{Yes})P(\text{Strong} \mid \text{Yes})]P(\text{Play}=\text{Yes}) = 0.0053$$

$$P(\text{No} \mid \mathbf{x}'): [P(\text{Sunny} \mid \text{No})P(\text{Cool} \mid \text{No})P(\text{High} \mid \text{No})P(\text{Strong} \mid \text{No})]P(\text{Play}=\text{No}) = 0.0206$$

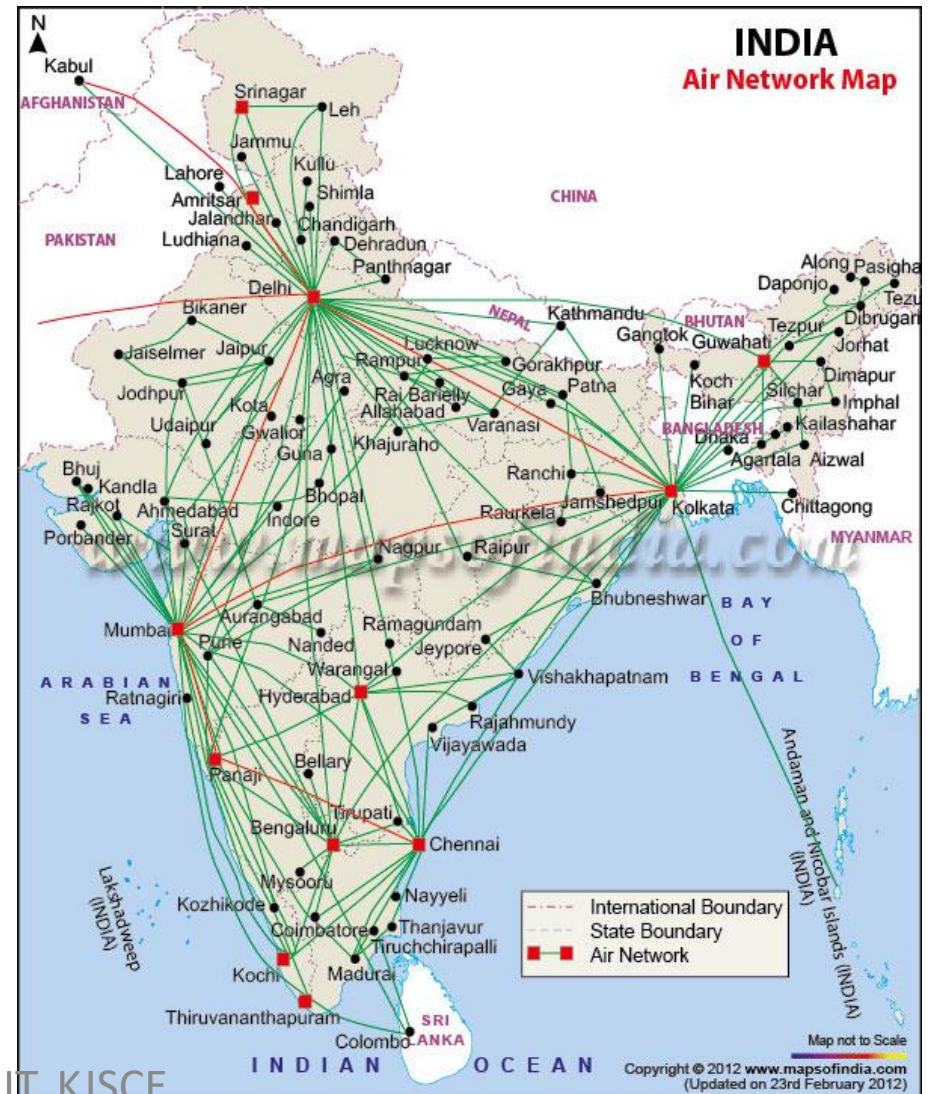
Given the fact $P(\text{Yes} \mid \mathbf{x}') < P(\text{No} \mid \mathbf{x}')$, we label \mathbf{x}' to be “No”.

Example: Bayesian Classification

Example 7.2: Air Traffic Data

Let us consider a set observation recorded in a database

Regarding the arrival of airplanes in the routes from any airport to New Delhi under certain conditions.



Air-Traffic Data

Days	Season	Fog	Rain	Class
Weekday	Spring	None	None	On Time
Weekday	Winter	None	Slight	On Time
Weekday	Winter	None	None	On Time
Holiday	Winter	High	Slight	Late
Saturday	Summer	Normal	None	On Time
Weekday	Autumn	Normal	None	Very Late
Holiday	Summer	High	Slight	On Time
Sunday	Summer	Normal	None	On Time
Weekday	Winter	High	Heavy	Very Late
Weekday	Summer	None	Slight	On Time

Cond. to next slide...

Air-Traffic Data

Cond. from previous slide...

Days	Season	Fog	Rain	Class
Saturday	Spring	High	Heavy	Cancelled
Weekday	Summer	High	Slight	On Time
Weekday	Winter	Normal	None	Late
Weekday	Summer	High	None	On Time
Weekday	Winter	Normal	Heavy	Very Late
Saturday	Autumn	High	Slight	On Time
Weekday	Autumn	None	Heavy	On Time
Holiday	Spring	Normal	Slight	On Time
Weekday	Spring	Normal	None	On Time
Weekday	Spring	Normal	Heavy	On Time

Air-Traffic Data

In this database, there are four attributes

A = [Day, Season, Fog, Rain]

with 20 tuples.

The categories of classes are:

C= [On Time, Late, Very Late, Cancelled]

Given this is the knowledge of data and classes, we are to find most likely classification for any other **unseen instance**, for example:

Week Day	Winter	High	None	???
-----------------	---------------	-------------	-------------	------------

Classification technique eventually to map this tuple into an accurate class.

Naïve Bayesian Classifier

Example: With reference to the Air Traffic Dataset mentioned earlier, let us tabulate all the posterior and prior probabilities as shown below.

		Class			
Attribute		On Time	Late	Very Late	Cancelled
Day	Weekday	$9/14 = 0.64$	$\frac{1}{2} = 0.5$	$3/3 = 1$	$0/1 = 0$
	Saturday	$2/14 = 0.14$	$\frac{1}{2} = 0.5$	$0/3 = 0$	$1/1 = 1$
	Sunday	$1/14 = 0.07$	$0/2 = 0$	$0/3 = 0$	$0/1 = 0$
	Holiday	$2/14 = 0.14$	$0/2 = 0$	$0/3 = 0$	$0/1 = 0$
Season	Spring	$4/14 = 0.29$	$0/2 = 0$	$0/3 = 0$	$0/1 = 0$
	Summer	$6/14 = 0.43$	$0/2 = 0$	$0/3 = 0$	$0/1 = 0$
	Autumn	$2/14 = 0.14$	$0/2 = 0$	$1/3 = 0.33$	$0/1 = 0$
	Winter	$2/14 = 0.14$	$2/2 = 1$	$2/3 = 0.67$	$0/1 = 0$

Naïve Bayesian Classifier

		Class			
Attribute		On Time	Late	Very Late	Cancelled
Fog	None	$5/14 = 0.36$	$0/2 = 0$	$0/3 = 0$	$0/1 = 0$
	High	$4/14 = 0.29$	$1/2 = 0.5$	$1/3 = 0.33$	$1/1 = 1$
	Normal	$5/14 = 0.36$	$1/2 = 0.5$	$2/3 = 0.67$	$0/1 = 0$
Rain	None	$5/14 = 0.36$	$1/2 = 0.5$	$1/3 = 0.33$	$0/1 = 0$
	Slight	$8/14 = 0.57$	$0/2 = 0$	$0/3 = 0$	$0/1 = 0$
	Heavy	$1/14 = 0.07$	$1/2 = 0.5$	$2/3 = 0.67$	$1/1 = 1$
Prior Probability		$14/20 = 0.70$	$2/20 = 0.10$	$3/20 = 0.15$	$1/20 = 0.05$

Naïve Bayesian Classifier

Instance:

Week Day	Winter	High	Heavy	???
----------	--------	------	-------	-----

Case1: Class = On Time : $0.70 \times 0.64 \times 0.14 \times 0.29 \times 0.07 = 0.0013$

Case2: Class = Late : $0.10 \times 0.50 \times 1.0 \times 0.50 \times 0.50 = 0.0125$

Case3: Class = Very Late : $0.15 \times 1.0 \times 0.67 \times 0.33 \times 0.67 = 0.0222$

Case4: Class = Cancelled : $0.05 \times 0.0 \times 0.0 \times 1.0 \times 1.0 = 0.0000$

Case3 is the strongest; Hence correct classification is **Very Late**

Naïve Bayesian Classifier

Algorithm: Naïve Bayesian Classification

Input: Given a set of k mutually exclusive and exhaustive classes $C = \{c_1, c_2, \dots, c_k\}$, which have prior probabilities $P(C_1), P(C_2), \dots, P(C_k)$.

There are n -attribute set $A = \{A_1, A_2, \dots, A_n\}$, which for a given instance have values $A_1 = a_1, A_2 = a_2, \dots, A_n = a_n$

Step: For each $c_i \in C$, calculate the class condition probabilities, $i = 1, 2, \dots, k$

$$p_i = P(C_i) \times \prod_{j=1}^n P(A_j = a_j | C_i)$$

$$p_x = \max\{p_1, p_2, \dots, p_k\}$$

Output: C_x is the classification

Note: $\sum p_i \neq 1$, because they are not probabilities rather proportion values (to posterior probabilities)

Naïve Bayesian Classifier

Pros and Cons

The Naïve Bayes' approach is a very popular one, which often works well.

However, it has a number of potential problems

It relies on all attributes being **categorical**.

If the data is **less**, then it **estimates poorly**.

Case Study - 1

Example 7.5

Class:

C1:buys_computer = 'yes'

C2:buys_computer = 'no'

Data instance

X = (age <=30,

Income = medium,

Student = yes

Credit_rating = fair)

age	income	student	credit_rating	comp
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

Solution to Case Study -1

- $P(C_i)$: $P(\text{buys_computer} = \text{"yes"}) = 9/14 = 0.643$
 $P(\text{buys_computer} = \text{"no"}) = 5/14 = 0.357$
 - Compute $P(X|C_i)$ for each class
 - $P(\text{age} = \text{"<=30"} | \text{buys_computer} = \text{"yes"}) = 2/9 = 0.222$
 - $P(\text{age} = \text{"<= 30"} | \text{buys_computer} = \text{"no"}) = 3/5 = 0.6$
 - $P(\text{income} = \text{"medium"} | \text{buys_computer} = \text{"yes"}) = 4/9 = 0.444$
 - $P(\text{income} = \text{"medium"} | \text{buys_computer} = \text{"no"}) = 2/5 = 0.4$
 - $P(\text{student} = \text{"yes"} | \text{buys_computer} = \text{"yes"}) = 6/9 = 0.667$
 - $P(\text{student} = \text{"yes"} | \text{buys_computer} = \text{"no"}) = 1/5 = 0.2$
 - $P(\text{credit_rating} = \text{"fair"} | \text{buys_computer} = \text{"yes"}) = 6/9 = 0.667$
 - $P(\text{credit_rating} = \text{"fair"} | \text{buys_computer} = \text{"no"}) = 2/5 = 0.4$
 - **$X = (\text{age} \leq 30, \text{income} = \text{medium}, \text{student} = \text{yes}, \text{credit_rating} = \text{fair})$**
 - $P(X|C_i) : P(X|\text{buys_computer} = \text{"yes"}) = 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044$
 $P(X|\text{buys_computer} = \text{"no"}) = 0.6 \times 0.4 \times 0.2 \times 0.4 = 0.019$
 - $P(X|C_i) * P(C_i) : P(X|\text{buys_computer} = \text{"yes"}) * P(\text{buys_computer} = \text{"yes"}) = 0.028$
 $P(X|\text{buys_computer} = \text{"no"}) * P(\text{buys_computer} = \text{"no"}) = 0.007$
- Therefore, X belongs to class ("buys_computer = yes")**

Example of Naïve Bayesian:

Unknown sample---- { Red, SUV, Domestic, ? }

Example No.	Color	Type	Origin	Stolen?
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	Imported	Yes

Color	
$P(\text{Red} \text{Yes}) = 3/5$	$P(\text{Red} \text{No}) = 2/5$
$P(\text{Yellow} \text{Yes}) = 2/5$	$P(\text{Yellow} \text{No}) = 3/5$
Type	
$P(\text{SUV} \text{Yes}) = 1/5$	$P(\text{SUV} \text{No}) = 3/5$
$P(\text{Sports} \text{Yes}) = 4/5$	$P(\text{Sports} \text{No}) = 2/5$
Origin	
$P(\text{Domestic} \text{Yes}) = 2/5$	$P(\text{Domestic} \text{No}) = 3/5$
$P(\text{Imported} \text{Yes}) = 3/5$	$P(\text{Imported} \text{No}) = 2/5$

$v = \text{Yes}$ -

$$P(\text{Yes}) * P(\text{Red} | \text{Yes}) * P(\text{SUV} | \text{Yes}) * P(\text{Domestic} | \text{Yes})$$

$$= 5/10 * 3/5 * 2/5 * 1/5 = 0.024$$

and for

$v = \text{No}$ -

$$P(\text{No}) * P(\text{Red} | \text{No}) * P(\text{SUV} | \text{No}) * P(\text{Domestic} | \text{No})$$

$$= 5/10 * 2/5 * 3/5 * 3/5 = 0.072$$

Since $0.072 > 0.024$, our example gets classified as 'NO'

Bayesian Classification: Why?

- Probabilistic learning: Calculate explicit probabilities for hypothesis, among the most practical approaches to certain types of learning problems
- Incremental: Each training example can incrementally increase/decrease the probability that a hypothesis is correct. Prior knowledge can be combined with observed data.
- Probabilistic prediction: Predict multiple hypotheses, weighted by their probabilities
- Standard: Even when Bayesian methods are computationally intractable, they can provide a standard of optimal decision making against which other methods can be measured

Issues (1): Data Preparation

- Data cleaning
 - Preprocess data in order to reduce noise and handle missing values
- Relevance analysis (feature selection)
 - Remove the irrelevant or redundant attributes
- Data transformation
 - Generalize and/or normalize data

Issues (2): Evaluating Classification Methods

- ❑ Predictive accuracy
- ❑ Speed and scalability
 - ❑ time to construct the model
 - ❑ time to use the model
- ❑ Robustness
 - ❑ handling noise and missing values
- ❑ Scalability
 - ❑ efficiency in disk-resident databases
- ❑ Interpretability:
 - ❑ understanding and insight provided by the model
- ❑ Goodness of rules
 - ❑ decision tree size
 - ❑ compactness of classification rules

Evaluating classification algorithms

- You have designed a new classifier.
- You give it to me, and I try it on my image dataset

Evaluating classification algorithms

- I tell you that it achieved 95% accuracy on my data.
- Is your technique a success?

Types of errors

But suppose that

- The 95% is the correctly classified pixels

- Only 5% of the pixels are actually edges

- It misses all the edge pixels

How do we count the effect of different types of error?

Evaluating Models

False positive - “type I error”

False negative - “type II error”

Types of errors

		Prediction	
		Edge	Not edge
Ground Truth	Edge	True Positive	False Negative
	Not Edge	False Positive	True Negative

Two parts to each: whether you got it correct or not, and what you guessed. For example for a particular pixel, our guess might be labelled...

True Positive

Did we get it correct?
True, we did get it correct.

What did we say?
We said 'positive', i.e. edge.

or maybe it was labelled as one of the others, maybe...

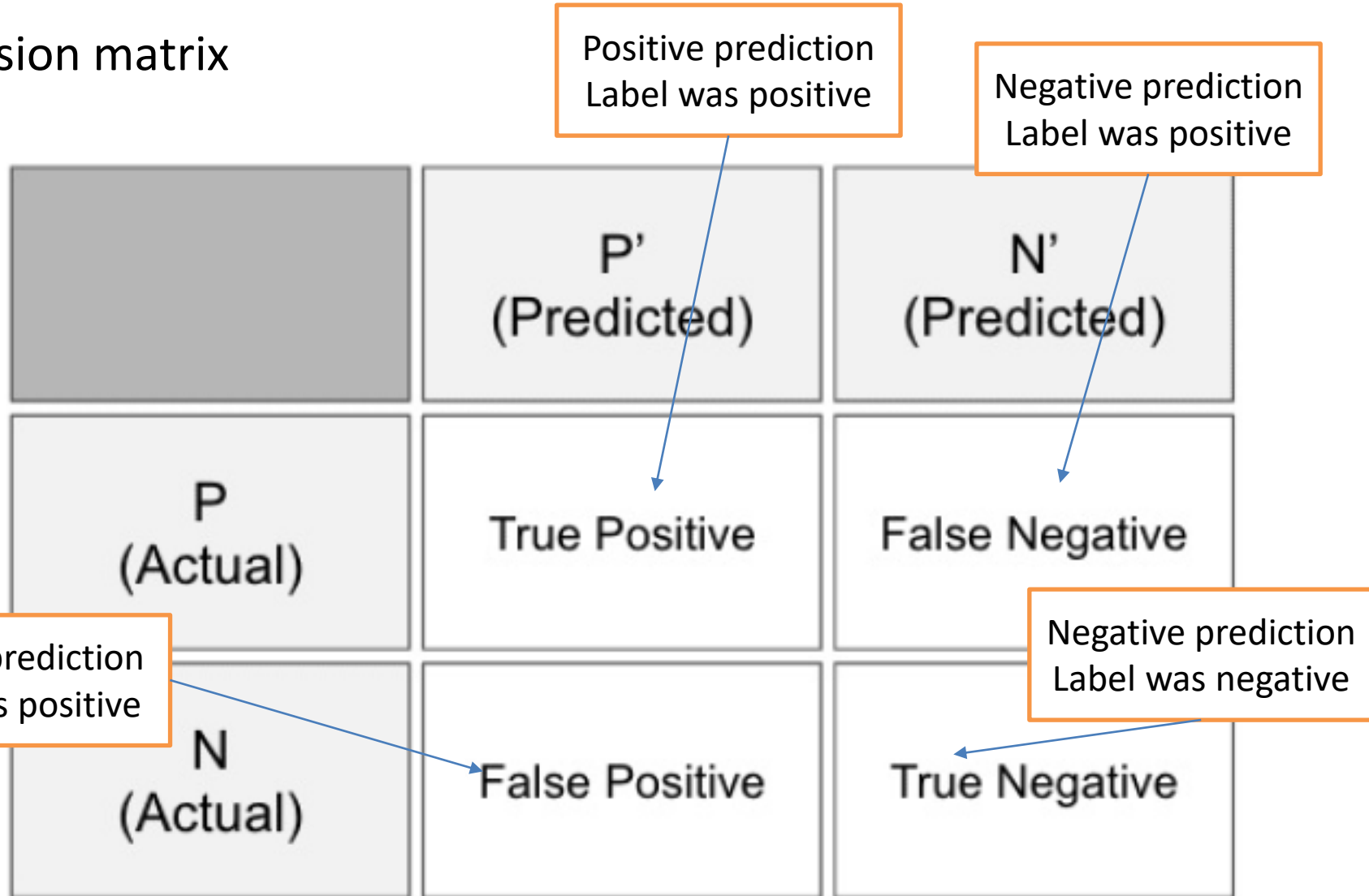
False Negative

Did we get it correct?
False, we did not get it correct.

What did we say?
We said 'negative, i.e. not edge.

Evaluating Models

Confusion matrix



Evaluating Models

Sensitivity versus specificity-

two different measures of a binary classification model. -

Sensitivity-

The true positive rate measures how often we classify an input record as the positive class and its correct classification.

This also is called sensitivity , or recall;

Sensitivity quantifies how well the model avoids false negatives.

$\text{Sensitivity} = \text{TP} / (\text{TP} + \text{FN})$

Evaluating Models

Sensitivity versus specificity-

Specificity-Specificity quantifies how well the model avoids false positives.

$$\text{Specificity} = \text{TN} / (\text{TN} + \text{FP})$$

Evaluating Models

Accuracy-

Accuracy is the degree of closeness of measurements of a quantity to that quantity's true value.

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{FP} + \text{FN} + \text{TN})$$

Accuracy can be misleading in the quality of the model when the class imbalance is high.

Evaluating Models

Precision

The degree to which repeated measurements under the same conditions give us the same results in the context of science and statistics.

Positive prediction value.

$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$

A measurement can be accurate yet not precise, not accurate but still precise, neither accurate nor precise, or both accurate and precise.

We consider a measurement to be valid if it is both accurate and precise.

Evaluating Models

Recall-

Same as sensitivity and is also known as the true positive rate or the hit rate.

F1 score-

In binary classification we consider the F1 score (or F-score, F-measure) to be a measure of a model's accuracy.

Harmonic mean of both the precision and recall measures (described previously) into a single score:

$$F1 = 2TP / (2TP + FP + FN)$$

Sensitivity and Specificity

Count up the total number of each label (TP, FP, TN, FN) over a large dataset. In ROC analysis, we use two statistics:

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

Can be thought of as the likelihood of spotting a positive case when presented with one.

Or... the proportion of edges we find.

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

Can be thought of as the likelihood of spotting a negative case when presented with one.

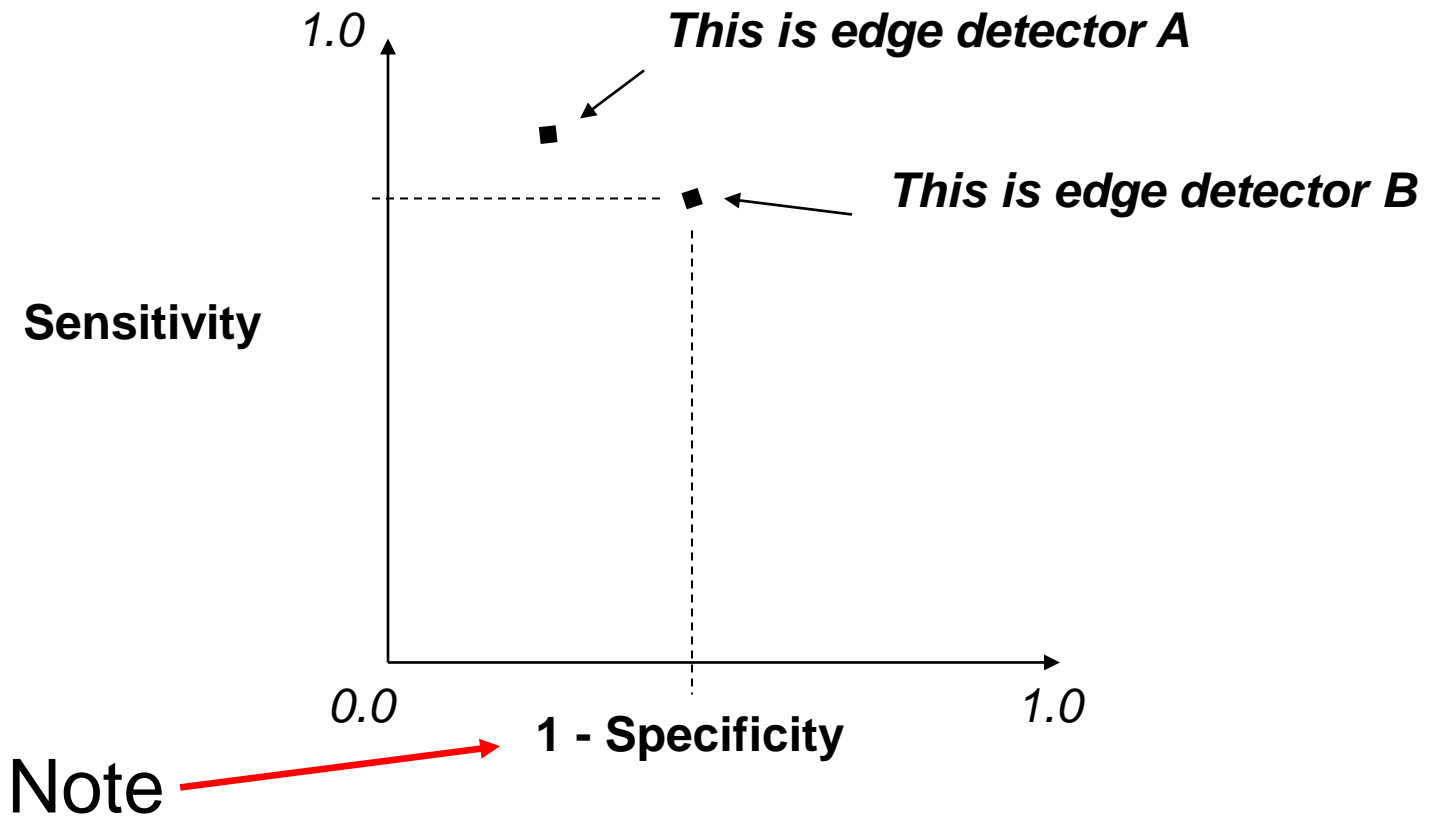
Or... the proportion of non-edges that we find

$$\text{Sensitivity} = \frac{TP}{TP+FN} = ? \quad \text{Specificity} = \frac{TN}{TN+FP} = ?$$

		Prediction		
		1	0	
Ground Truth	1	60	30	60+30 = 90 cases in the dataset were class 1 (edge)
	0	80	20	80+20 = 100 cases in the dataset were class 0 (non-edge)

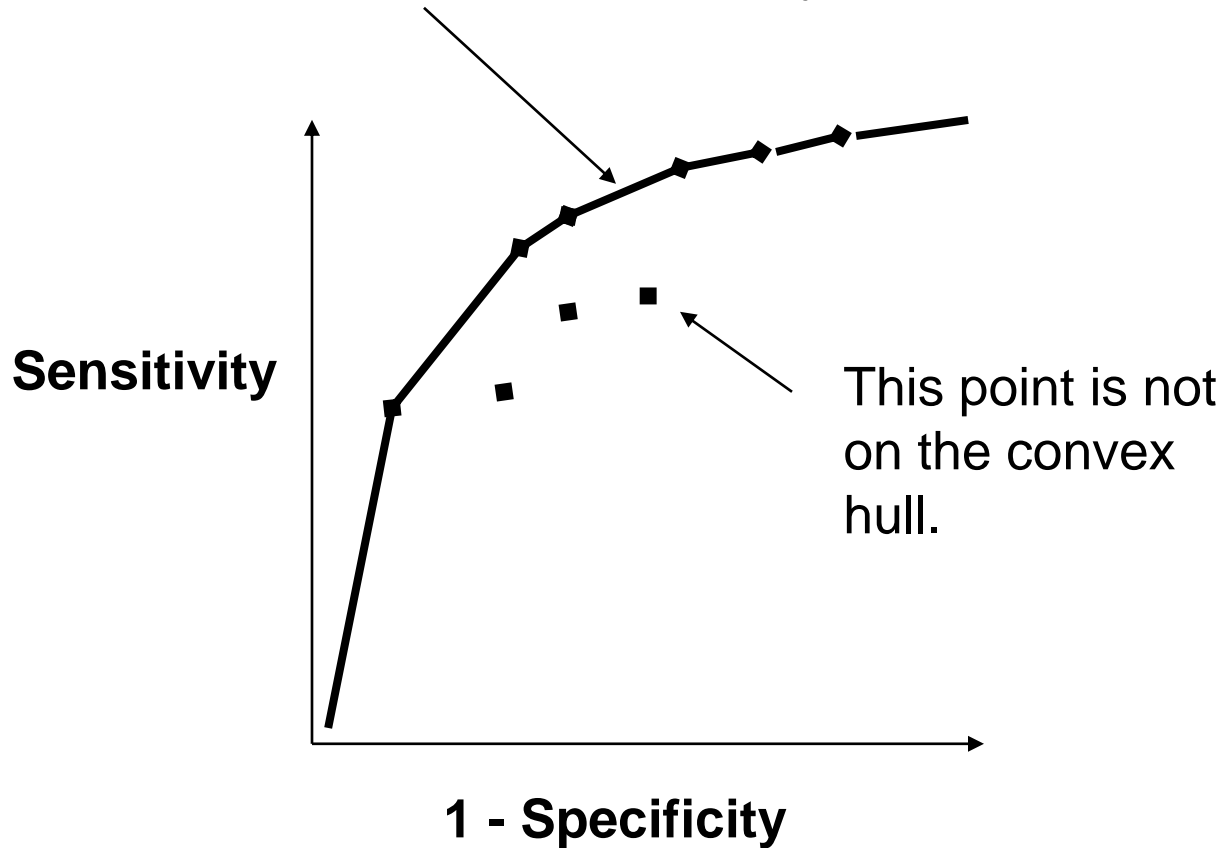
90+100 = 190 examples (pixels) in the data overall

The ROC space

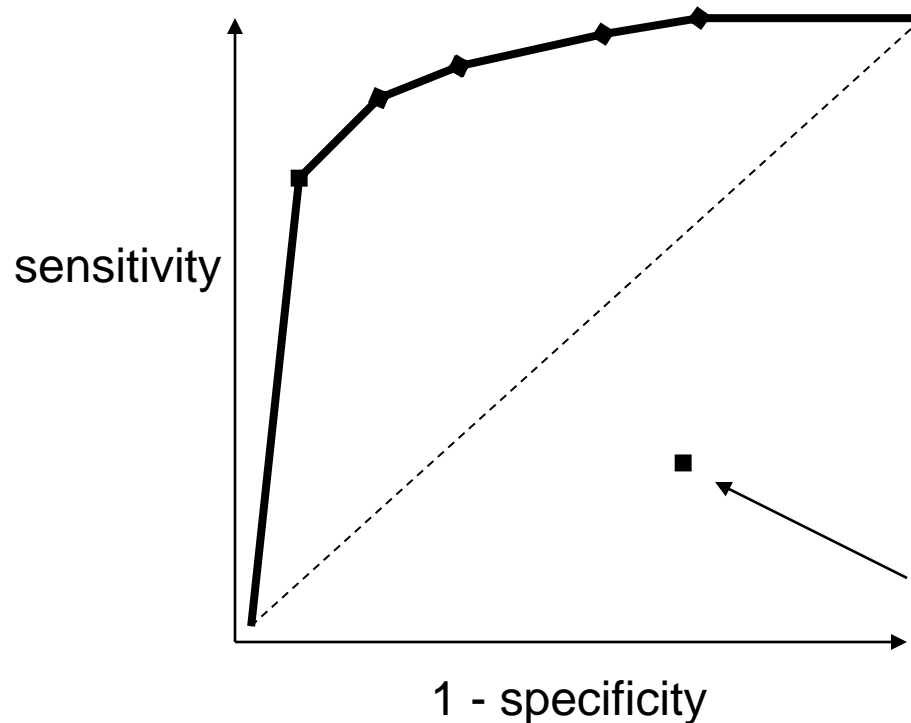


The ROC Curve

Draw a 'convex hull' around many points:



ROC Analysis



All the optimal detectors lie on the convex hull.

Which of these is best depends on the ratio of edges to non-edges, and the different cost of misclassification

Any detector on this side can lead to a better detector by flipping its output.

Take-home point : You should always quote sensitivity and specificity for your algorithm, if possible plotting an ROC graph. Remember also though, any statistic you quote should be an average over a suitable range of tests for your algorithm.

References

Data Mining: Concepts and Techniques, (3rd Edn.), Jiawei Han, Micheline Kamber, Morgan Kaufmann, 2015.

Introduction to Data Mining, Pang-Ning Tan, Michael Steinbach, and Vipin Kumar, Addison-Wesley, 2014

<https://www.cse.sc.edu>

<https://cse.iitkgp.ac.in/~dsamanta/courses/da/resources>

<https://sites.astro.caltech.edu>