UVA CS 6316: Machine Learning

Lecture 1: Introduction

Dr. Yanjun Qi

University of Virginia
Department of Computer Science

Today

- ❑ Course Logistics
- ❑ Machine Learning Basics
- ❑ A Rough Plan of Course Content
- ❑ Machine Learning History

Welcome

- Course Website:
  - https://qiyanjun.github.io/2019f-UVA-CS6316-MachineLearning/

We focus on learning fundamental principles, mathematical formulation, algorithm design and learning theory.

Objective

- To help students able to build simple machine learning tools
  - (not just a tool user!!!)

- Key Results:
  - Able to build a few simple machine learning methods from scratch
  - [illegible]
  - [illegible]
  - [illegible] paper and understand the pipeline

We focus on developing skills, more than feeding knowledge

Course Staff

- Instructor: Prof. Yanjun Qi
  - QI: /ch ee/
  - You can call me “professor”, “professor Qi”;
  - I have been teaching Graduate-level and Under-Level Machine Learning course for years!
  - My research is about machine learning

- TA and Office Hour information @ CourseWeb

Course Material

- Text books for this class is:
  - NONE
- My slides – **if it is not mentioned in my slides, it is not an official topic of the course**
- Your UVA Collab for Assignments and Project
- Google Forms for Quizzes

Course Background Needed

- **Background Needed**
  - Calculus, Basic linear algebra, Basic probability and Basic Algorithm
  - Statistics is recommended.
  - Students should already have good programming skills, i.e. python is required for all programming assignments

Today

- ❑ Course Logistics
- ❑ Machine Learning Basics
- ❑ Machine Learning History
- ❑ Rough Plan of Course Content

OUR DATA-RICH WORLD

- Biomedicine
  - Patient records, brain imaging, MRI & CT scans, …
  - Genomic sequences, bio-structure, drug effect info, …
- Science
  - Historical documents, scanned books, databases from astronomy, environmental data, climate records, …

- Social media
  - Social interactions data, twitter, facebook records, online reviews, …
- Business
  - Stock market transactions, corporate sales, airline traffic, …
- Entertainment
  - Internet images, Hollywood movies, music audio files, …

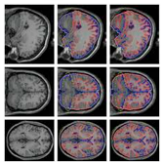
What can we do with the data wealth?

➔ REAL-WORLD IMPACT

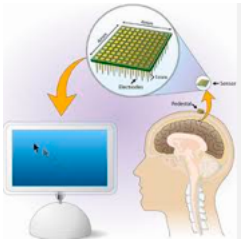
Transportation Data

Genomic Data

Medical Images



Brain computer interaction (BCI)



Device sensor data

- Business efficiencies
- Scientific breakthroughs
- Improve quality-of-life:
  - healthcare,
  - energy saving / generation,
  - environmental disasters,
  - nursing home,
  - transportation,
  - …

BIG DATA CHALLENGES

- Data capturing (sensor, smart devices, medical instruments, et al.)
- Data transmission
- Data storage
- Data management
- High performance data processing
- Data visualization
- Data security & privacy (e.g. multiple ind
- ……

e.g. cloud computing

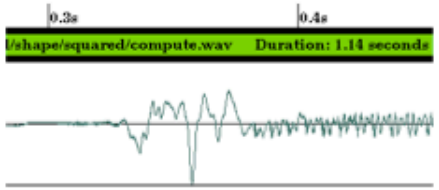
e.g. HCI

this course

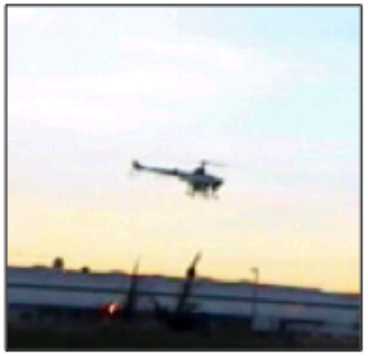
- Data analytics
  - How can we analyze this big data wealth ?
  - E.g. Machine learning and data mining

MACHINE LEARNING IS CHANGING THE WORLD

Mining Databases



Speech Recognition



Control learning

Object recognition

Text analysis

Peter H. van Oppen , Chairman of the Board & Chief Executive Officer
Mr. van Oppen has served as chairman of the board and chief executive officer of ADIC since its acquisition by Interpoint in 1994 and a director of ADIC since 1986. Until its acquisition by Crane Co. in October 1996, Mr. van Oppen served as chairman of the board of directors, president and chief executive officer of Interpoint . Prior to 1985, Mr. van Oppen worked as a consulting manager at Price Waterhouse LLP and at Bain & Company in Boston and London. He has additional experience in medical electronics and venture capital. Mr. van Oppen also serves as a director of Seattle FilmWorks Inc. and Spacelabs Medical, Inc.. He holds a B.A. from Whitman College and an M.B.A. from Harvard Business School, where he was a Baker Scholar.

Many more !

BASICS OF MACHINE LEARNING

- "The goal of machine learning is to build computer systems that can learn and adapt from their experience." – Tom Dietterich

- "Experience" in the form of available data examples (also called as instances, samples)

- Available examples are described with properties (data points in feature space X)

e.g. SUPERVISED LEARNING

- Find function to map input space X to output space Y

$$f : X \longrightarrow Y$$

- So that the difference between y and $f(\boldsymbol{x})$ of each example $\boldsymbol{x}$ is small.

e.g.

x	I believe that this book is not at all helpful since it does not explain thoroughly the material . it just provides the reader with tables and calculations that sometimes are not easily understood …

Input X : e.g. a piece of English text

→

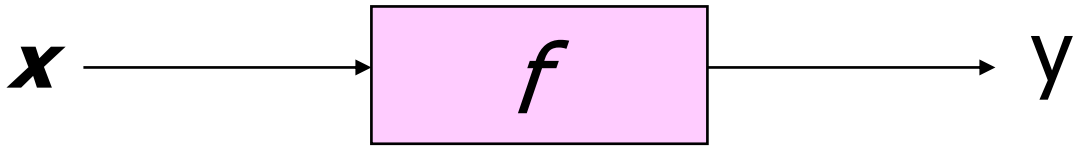
y	-1

Output Y: {1 / Yes , -1 / No }

e.g. Is this a positive product review ?

SUPERVISED Linear Binary Classifier

- Now let us check out a VERY SIMPLE case of

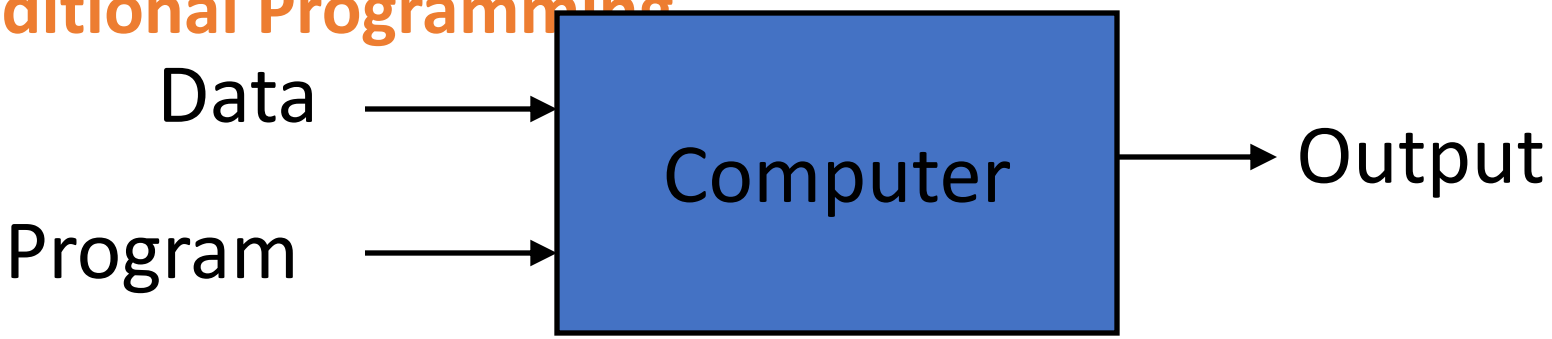


e.g.: Binary *y* / Linear *f* / X as R^2

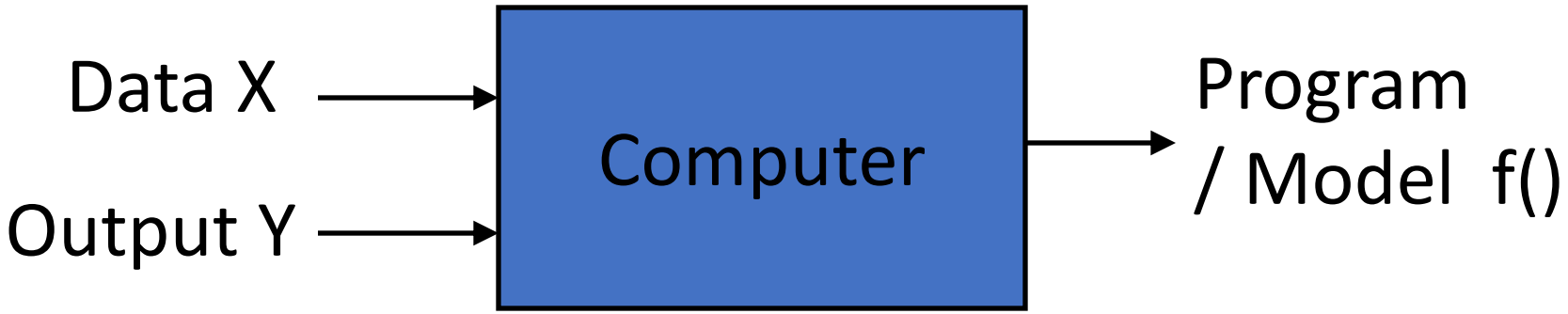
$$f(\mathbf{x}, \mathbf{w}, b) = sign(\mathbf{w}^T \mathbf{x} + b)$$

$$\mathbf{x} = (x\_1, x\_2)$$

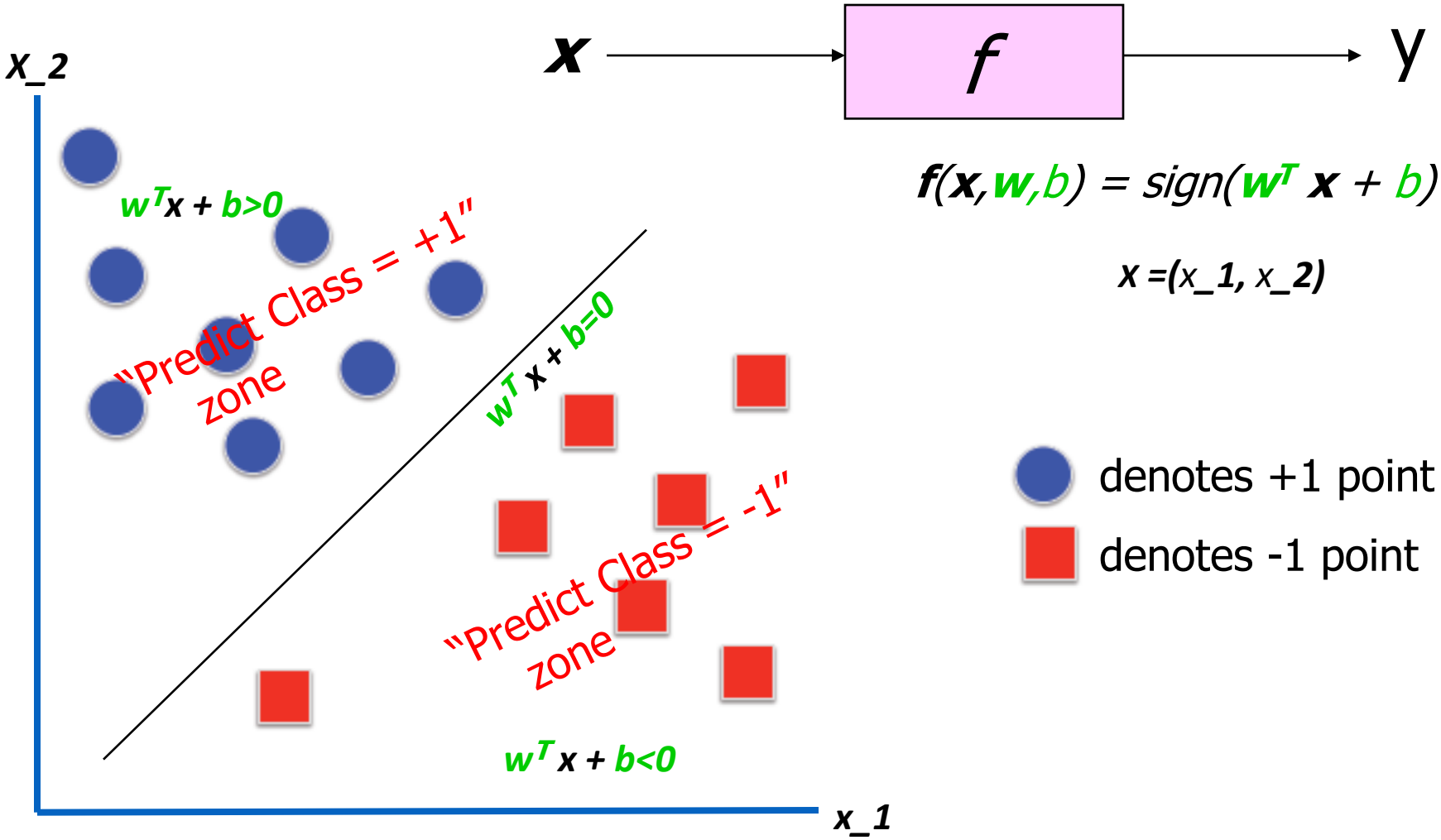
Traditional Programming



Machine Learning (training phase)

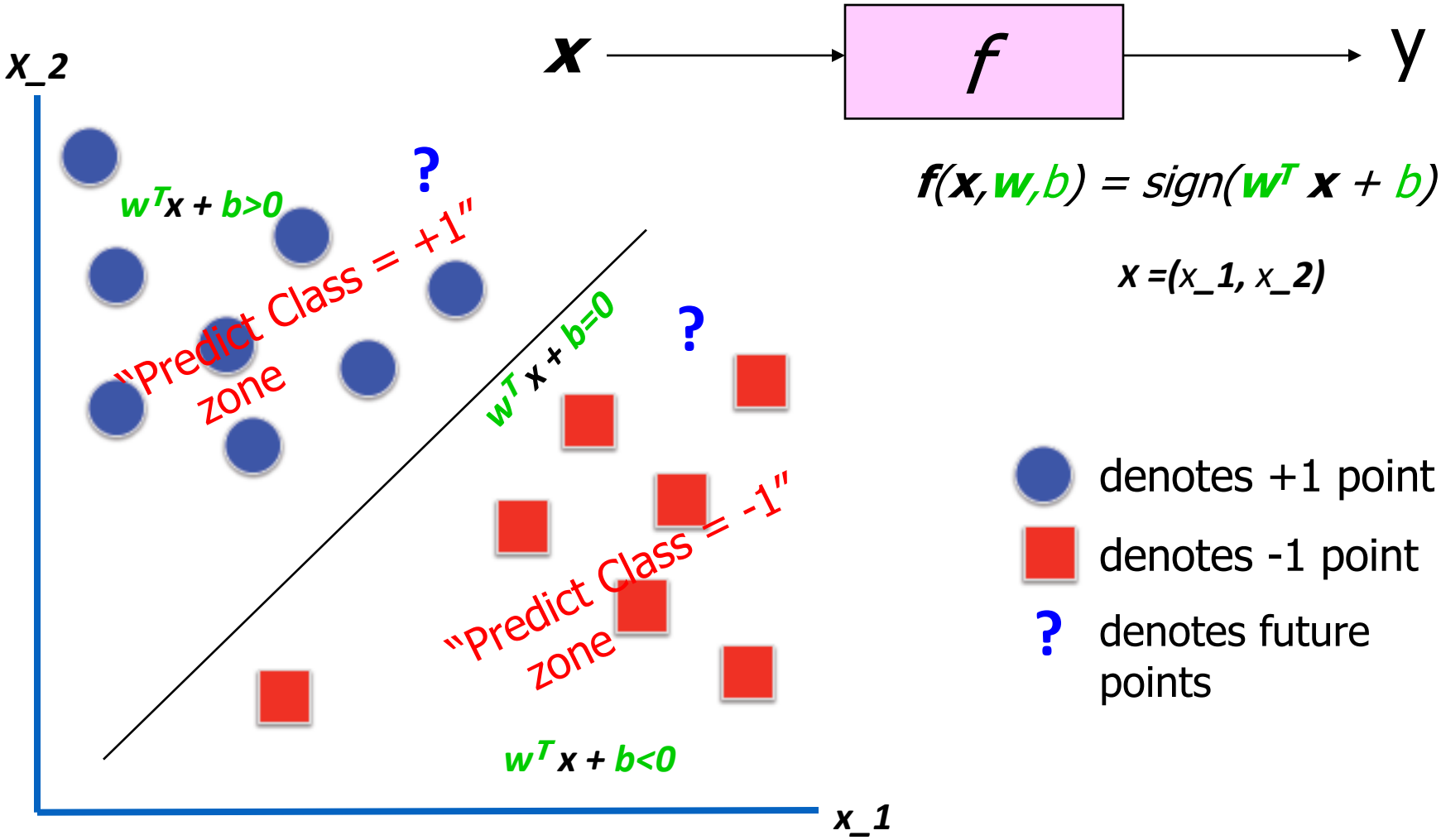


SUPERVISED Linear Binary Classifier



Courtesy slide from Prof. Andrew Moore's tutorial

SUPERVISED Linear Binary Classifier



Courtesy slide from Prof. Andrew Moore's tutorial

Basic Concepts

- Training (i.e. learning parameters ***w,b***)
  - Training set includes
    - available examples $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_L$
    - available corresponding labels $y_1,\ldots,y_L$
  - Find (***w,b***) by minimizing loss
    - (i.e. difference between y and $f(\boldsymbol{x})$ *on available examples in training set*)

$$(\boldsymbol{W}, b) = \underset{\boldsymbol{W},\, b}{\operatorname{argmin}} \sum_{i=1}^{L} \ell(f(x_i), y_i)$$

Basic Concepts

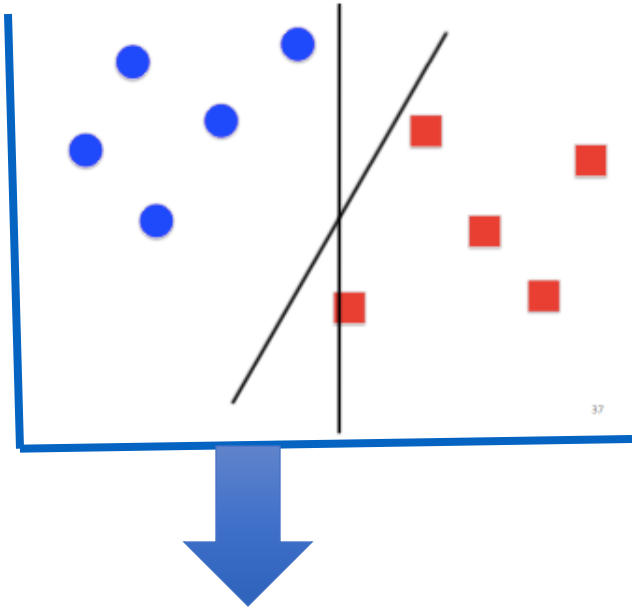
- Testing (i.e. evaluating performance on "future" points)
  - Difference between true $y_?$ and the predicted $f(\boldsymbol{x}_?)$ on a set of testing examples (i.e. *testing set*)
  - Key: example $\boldsymbol{x}_?$ not in the training set

- Generalisation: learn function / hypothesis from past data in order to "explain", "predict", "model" or "control" new data examples

Basic Concepts

- Loss function
  - e.g. hinge loss for binary cl

$$\sum_{i=1}^{L} \ell(f(x_i), y_i) = \sum_{i=1}^{L} \max(0, 1 - y_i f(x_i))$$

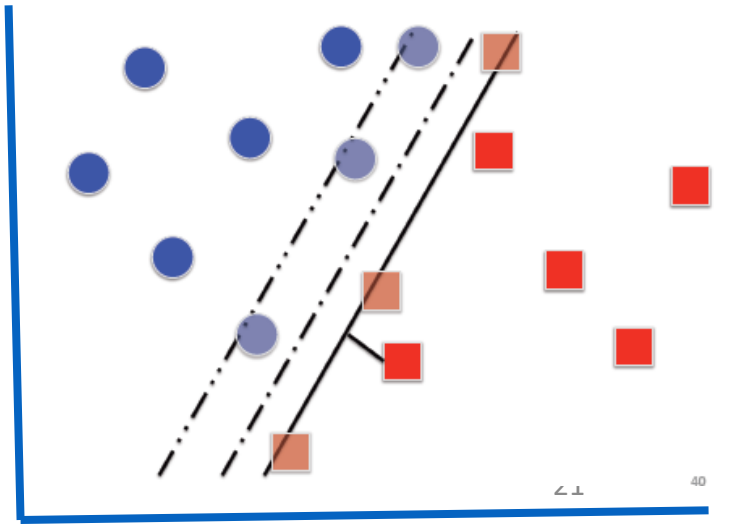


  - e.g. pairwise ranking loss for ranking task (i.e. ordering examples by preference)

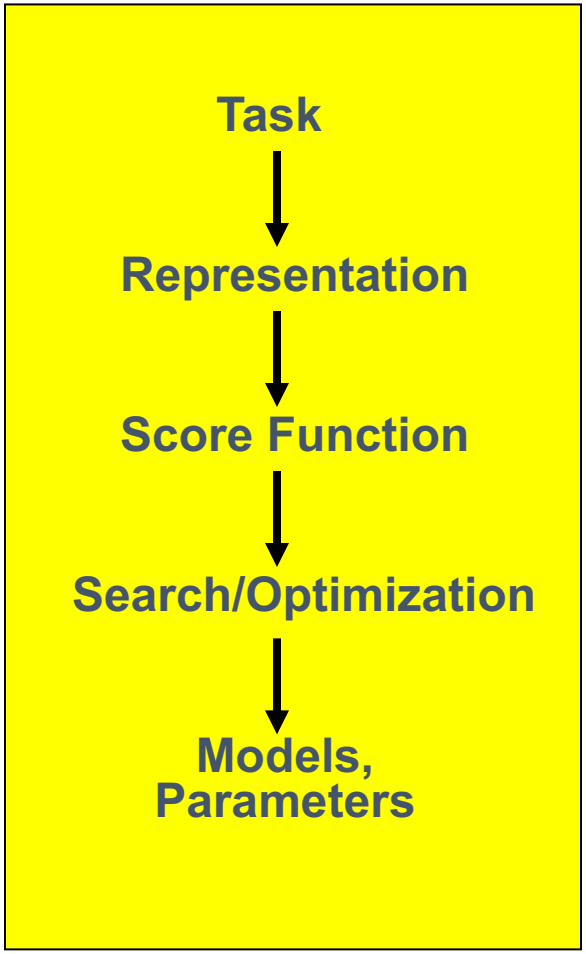
- Regular

$$C \sum_{i=1}^{L} \ell(f(x_i), y_i) + \frac{1}{2}\|w\|^2$$

  - E.g. on added on loss function to control f



Machine Learning in a Nutshell



ML grew out of work in AI

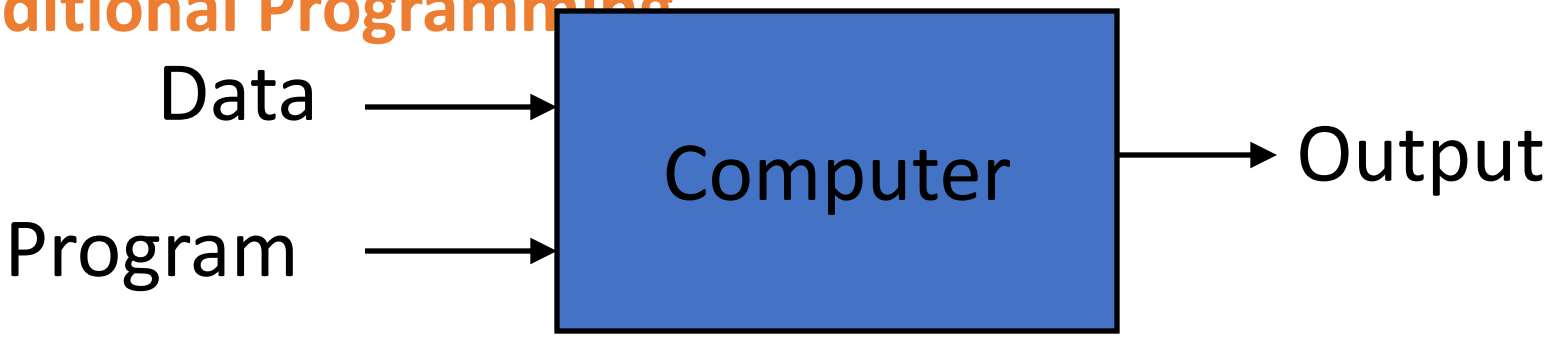
Optimize a performance criterion using example data or past experience,

Aiming to generalize to unseen data

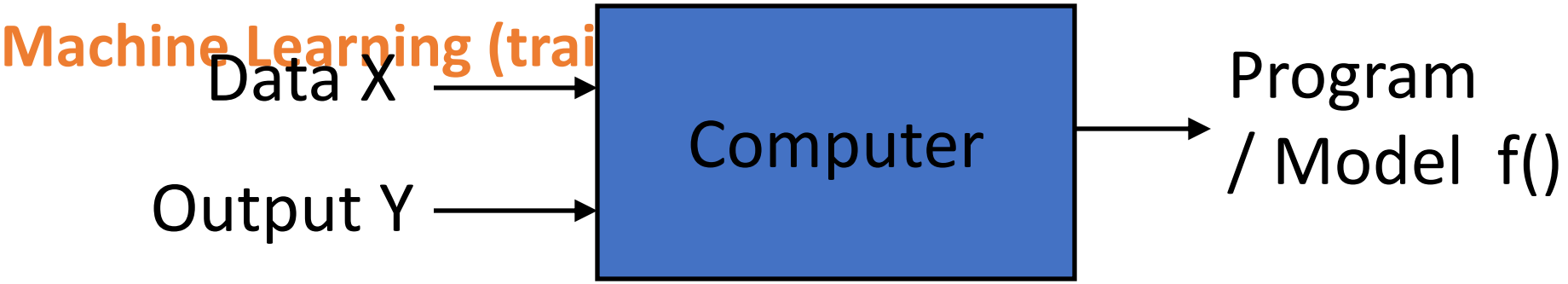
Today

- ❑ Course Logistics
- ❑ Machine Learning Basics
- ❑ A Rough Plan of Course Content
- ❑ Machine Learning History

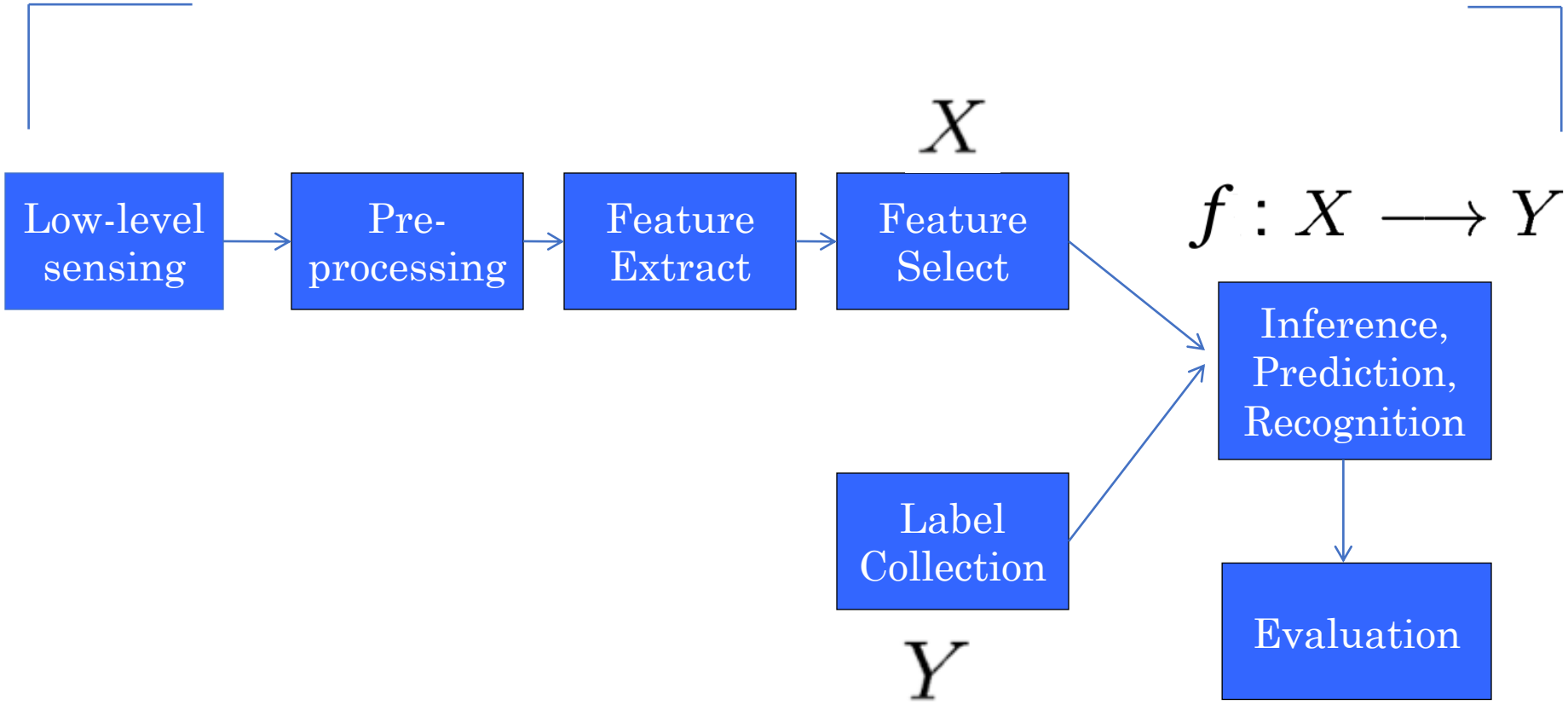
Traditional Programming



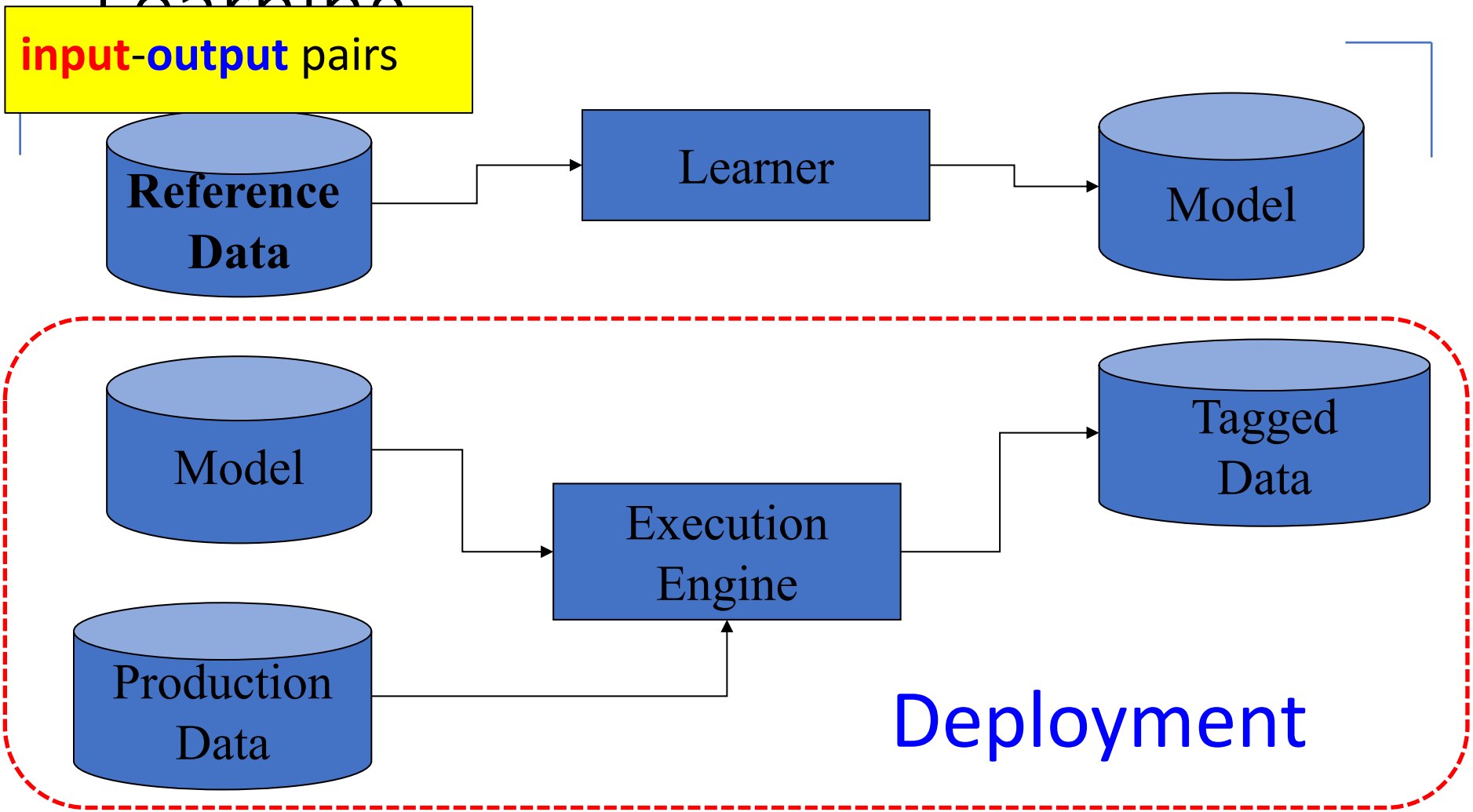
Machine Learning (trai



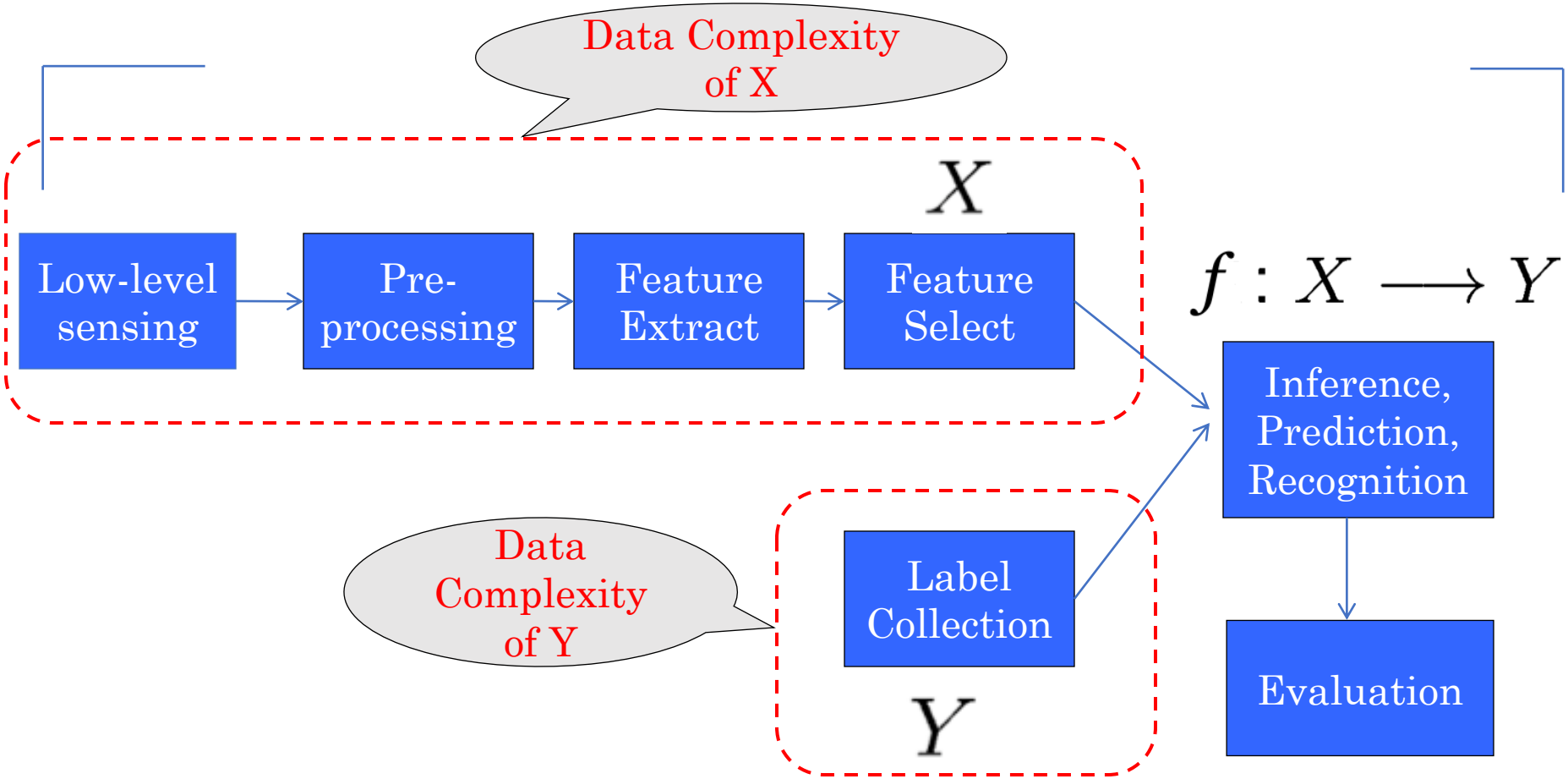
TYPICAL MACHINE LEARNING SYSTEM (Training Mode)



An Operational Model of Machine Learning

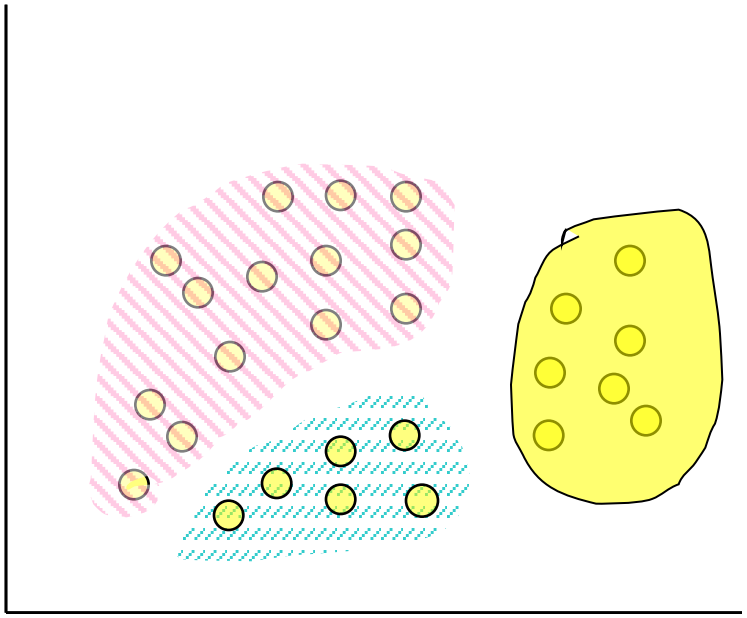


TYPICAL MACHINE LEARNING SYSTEM



UNSUPERVISED LEARNING : COMPLEXITY in Y]

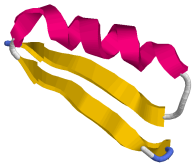
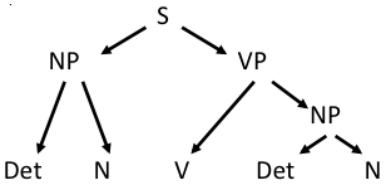
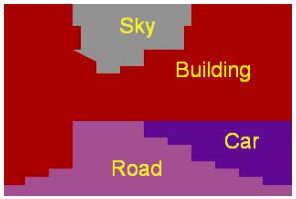
- No labels are provided (e.g. No Y provided)
- Find patterns from unlabeled data, e.g. clustering



e.g. clustering => to find "natural" grouping of instances given un-labeled data

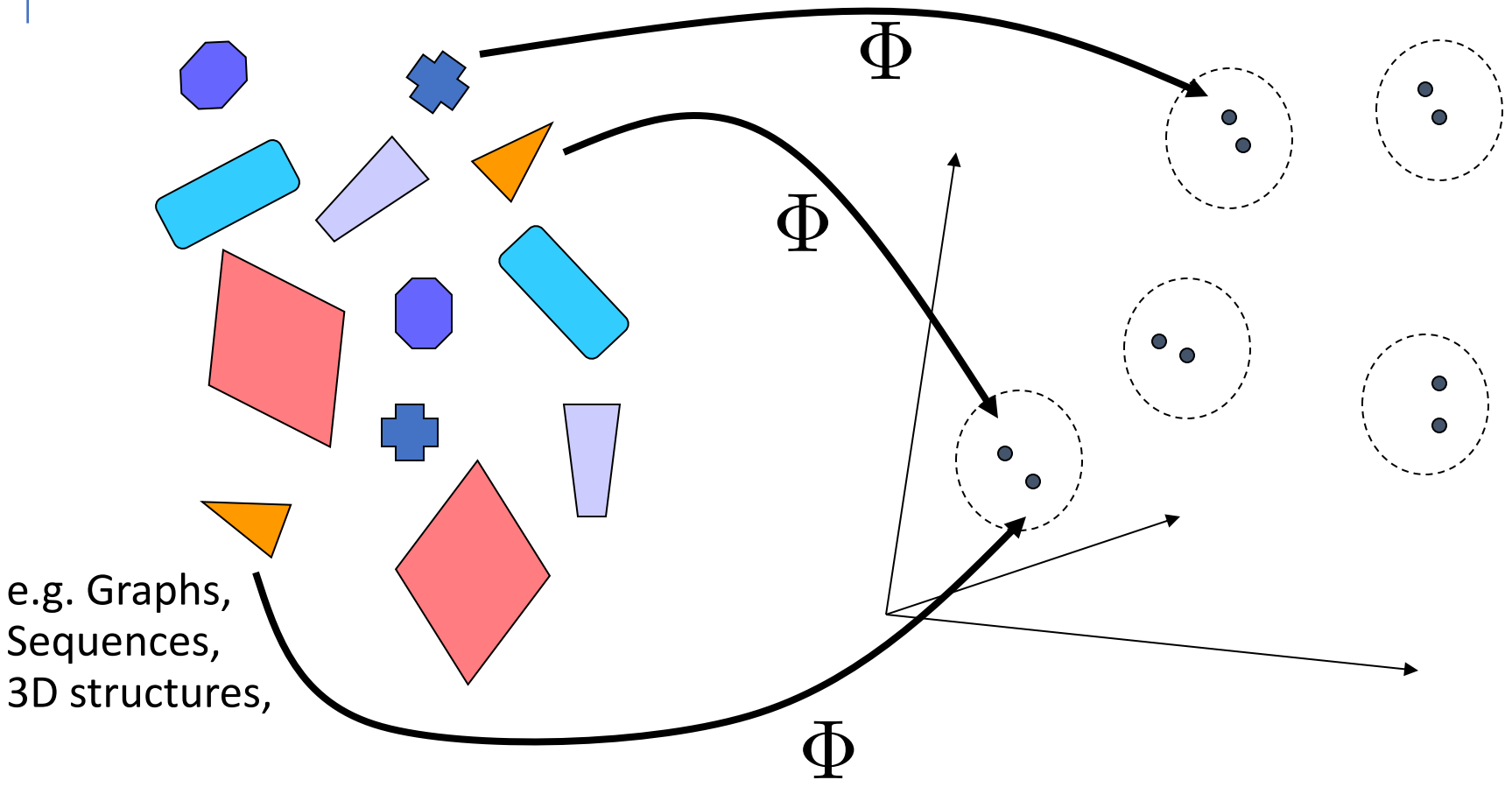
STRUCTURAL OUTPUT LEARNING : [COMPLEXITY OF Y]

- Many prediction tasks involve output labels having structured correlations or constraints among instances

Structured Dependency between Examples' Y	Sequence	Tree	Grid
Input X	APAFSVSPASGACGPECA…	The dog chased the cat	
Output Y	 CCEEEEECCCCCHHHCCC…		

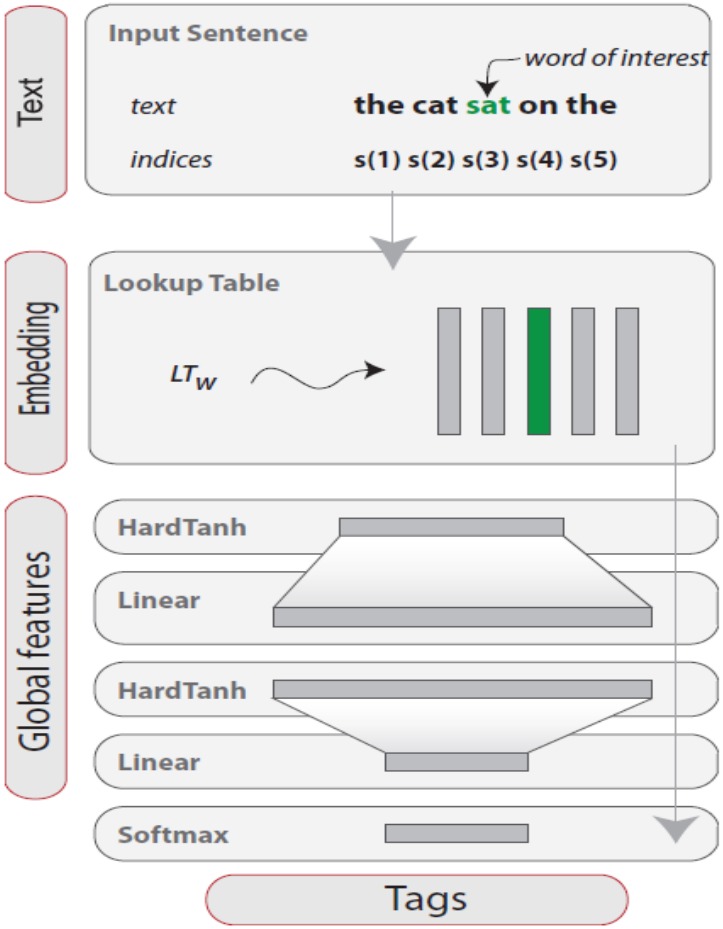
Many more possible structures between y_i , e.g. spatial , temporal, relational …

STRUCTURAL INPUT : Kernel Methods
[COMPLEXITY OF X]

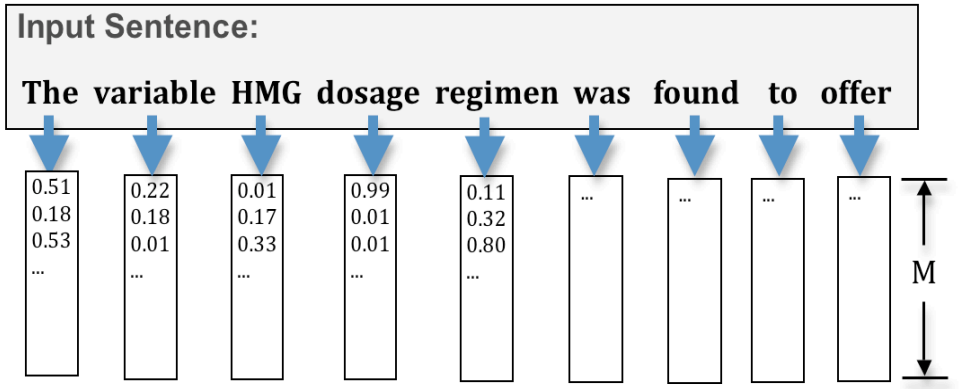


MORE RECENT: FEATURE LEARNING
[COMPLEXITY OF X]

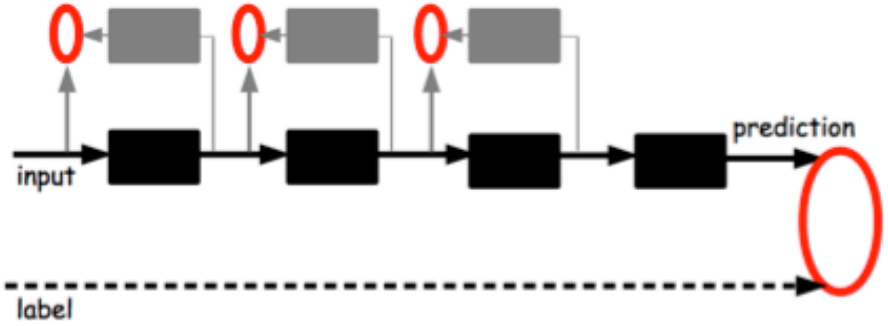
Deep Learning



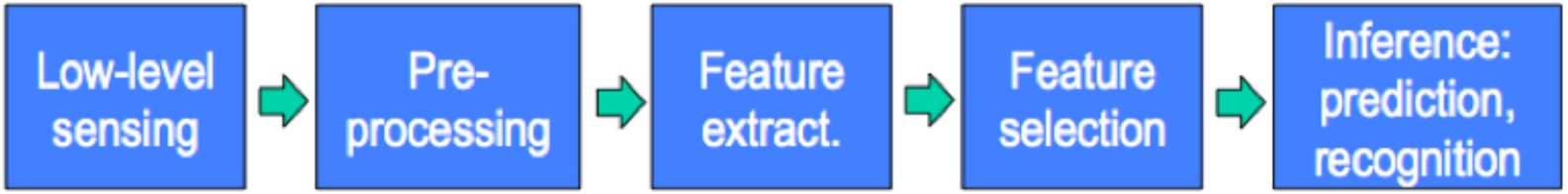
Supervised Embedding



Layer-wise Pretraining



DEEP LEARNING / FEATURE LEARNING : [COMPLEXITY OF X]

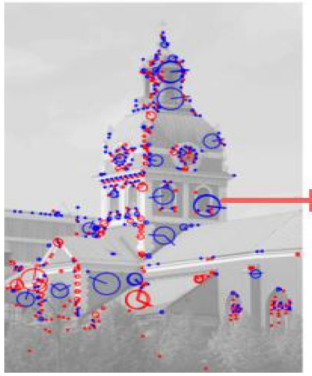


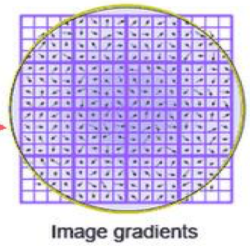
Feature Engineering

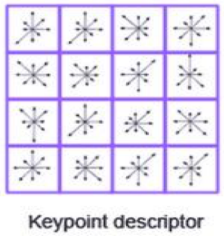
- ✓ Most critical for accuracy
- ✓ Account for most of the computation for testing
- ✓ Most time-consuming in development cycle
- ✓ Often hand-craft and task dependent in practice

Feature Learning

- ✓ Easily adaptable to new similar tasks
- ✓ Layerwise representation
- ✓ Layer-by-layer unsupervised training
- ✓ Layer-by-layer supervised training



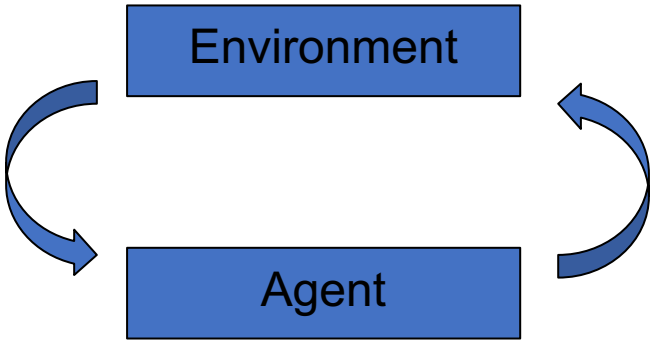




Why learn features

Reinforcement Learning (RL)

- What's Reinforcement Learning?

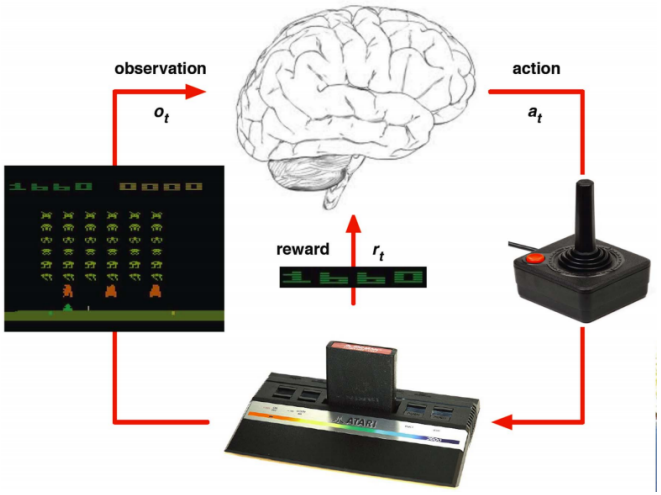


- Agent interacts with an environment and learns by maximizing a scalar reward signal
- No labels or any other supervision signal.
- Previously suffering from hand-craft states or representation.

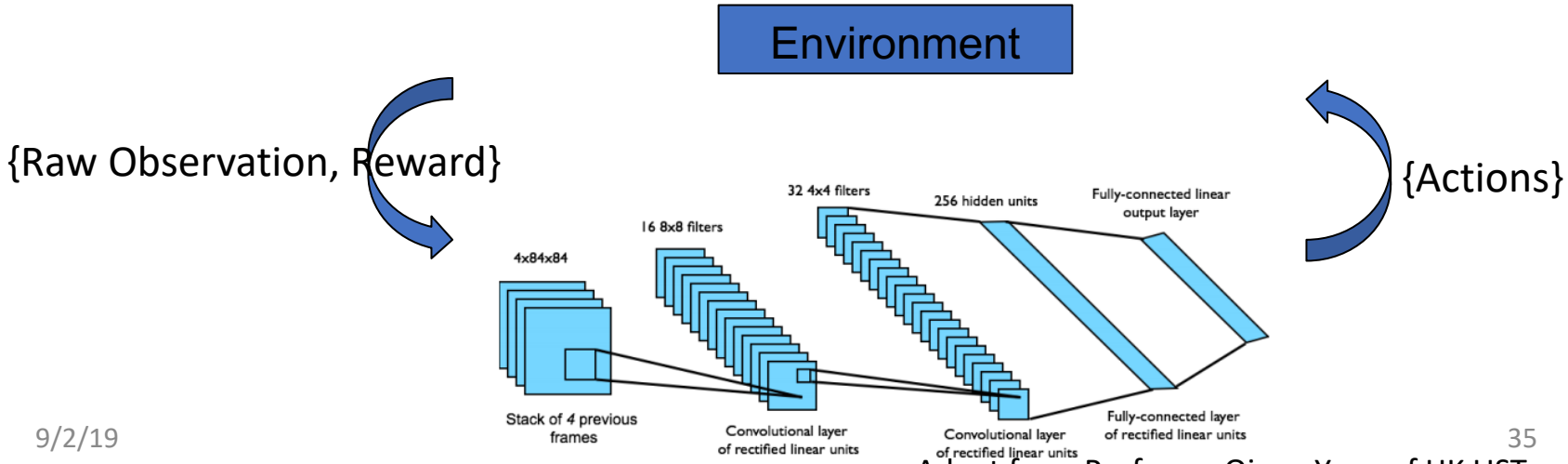
Adapt from Professor Qiang Yang of HK UST

Deep Reinforcement Learning

- Human



- So what's **DEEP** RL?



Adapt from Professor Qiang Yang of HK UST

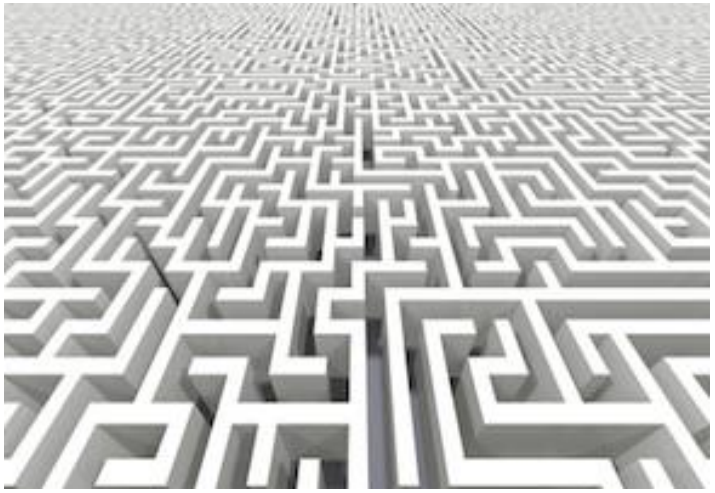
When to use Machine Learning (Adapt to / learn from data) ?

- 1. Extract knowledge from data
  - Relationships and correlations can be hidden within large amounts of data
  - The amount of knowledge available about certain tasks is simply too large for explicit encoding (e.g. rules) by humans
- 2. Learn tasks that are difficult to formalise
  - Hard to be defined well, except by examples, e.g., face recognition
- 3. Create software that improves over time
  - New knowledge is constantly being discovered.
  - Rule or human encoding-based system is difficult to continuously re-design "by hand".

“Big Data” Challenges for Machine Learning

LARGE-SCALE

HIGH-COMPLEXITY



- ✓Large size of samples
- ✓High dimensional features

Not the focus, being covered in my advanced-level course

Large-Scale Machine Learning: SIZE MATTERS

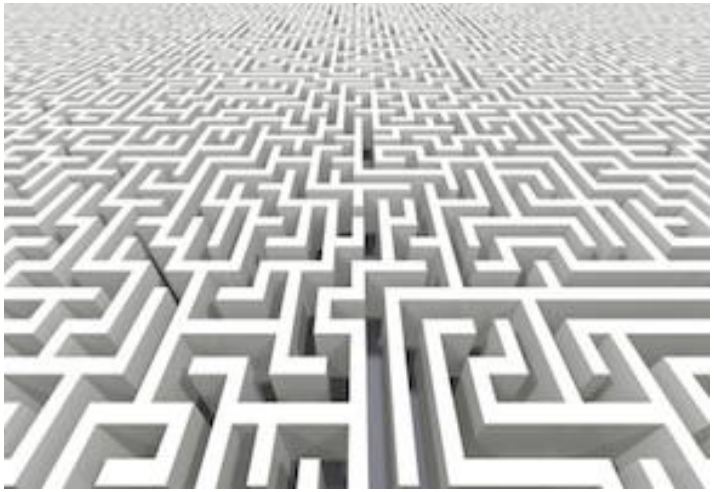
- One thousand data instances
- One million data instances
- One billion data instances
- One trillion data instances

Those are not different numbers, those are different mindsets !!!

BIG DATA CHALLENGES FOR MACHINE LEARNING

LARGE-SCALE

Highly Complex



Most of this course

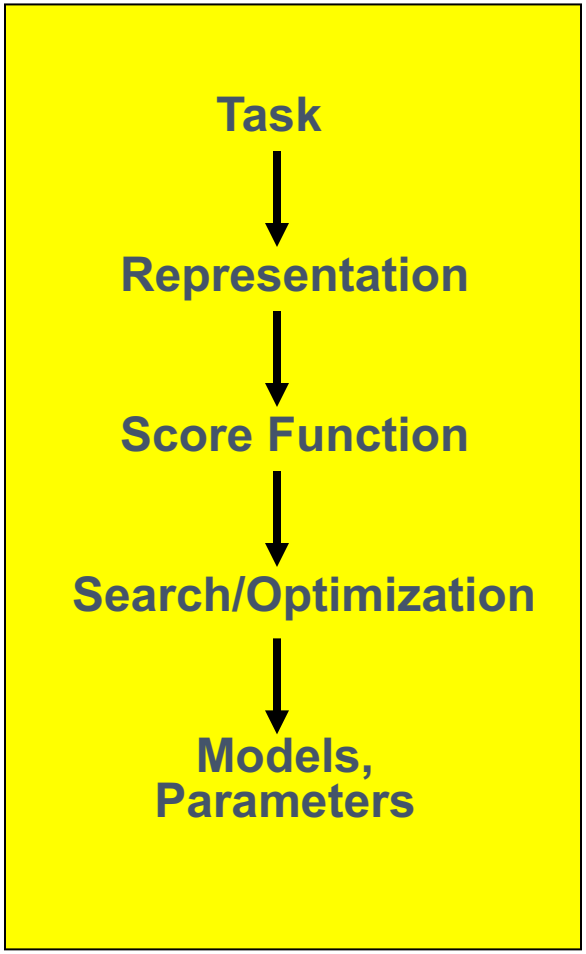
The variations of both X (feature, representation) and Y (labels) are complex !

- ✓Complexity of X
- ✓Complexity of Y

Course Content Plan ➔
Six major sections of this course

- ❑ Regression (supervised) — Y is a continuous
- ❑ Classification (supervised) — Y is a discrete
- ❑ Unsupervised models — NO Y
- ❑ Learning theory — About f()
- ❑
- ❑ Graphical models — About interactions among X1,… Xp
- ❑ Reinforcement Learning — About interactions among X1,… Xp

Machine Learning in a Nutshell



ML grew out of work in AI

Optimize a performance criterion using example data or past experience,

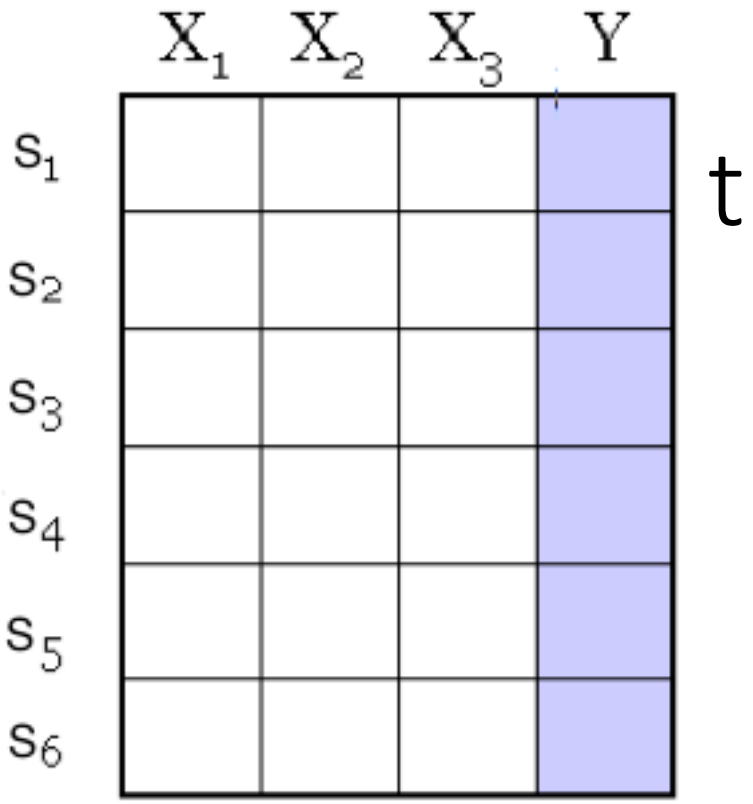
Aiming to generalize to unseen data

What we have covered

Task	
Representation	
Score Function	
Search/Optimization	
Models, Parameters	

What we will cover

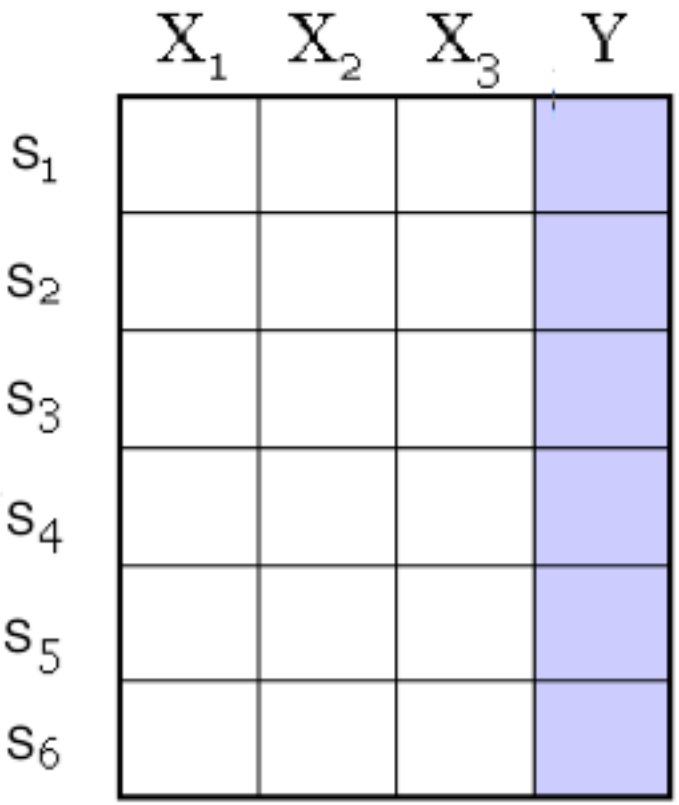
Task	Regression, classification, clustering, dimen-reduction
Representation	Linear func, nonlinear function (e.g. polynomial expansion), local linear, logistic function (e.g. p(c|x)), tree, multi-layer, prob-density family (e.g. Bernoulli, multinomial, Gaussian, mixture of Gaussians), local func smoothness, kernel matrix, local smoothness, partition of feature space,
Score Function	MSE, Margin, log-likelihood, EPE (e.g. L2 loss for KNN, 0-1 loss for Bayes classifier), cross-entropy, cluster points distance to centers, variance, conditional log-likelihood, complete data-likelihood, regularized loss func (e.g. L1, L2) , goodness of inter-cluster similar
Search/ Optimization	Normal equation, gradient descent, stochastic GD, Newton, Linear programming, Quadratic programming (quadratic objective with linear constraints), greedy, EM, asyn-SGD, eigenDecomp, backprop
Models, Parameters	Linear weight vector, basis weight vector, local weight vector, dual weights, training samples, tree-dendrogram, multi-layer weights, principle components, member (soft/hard) assignment, cluster centroid, cluster covariance (shape), …



$$f : X \longrightarrow Y$$

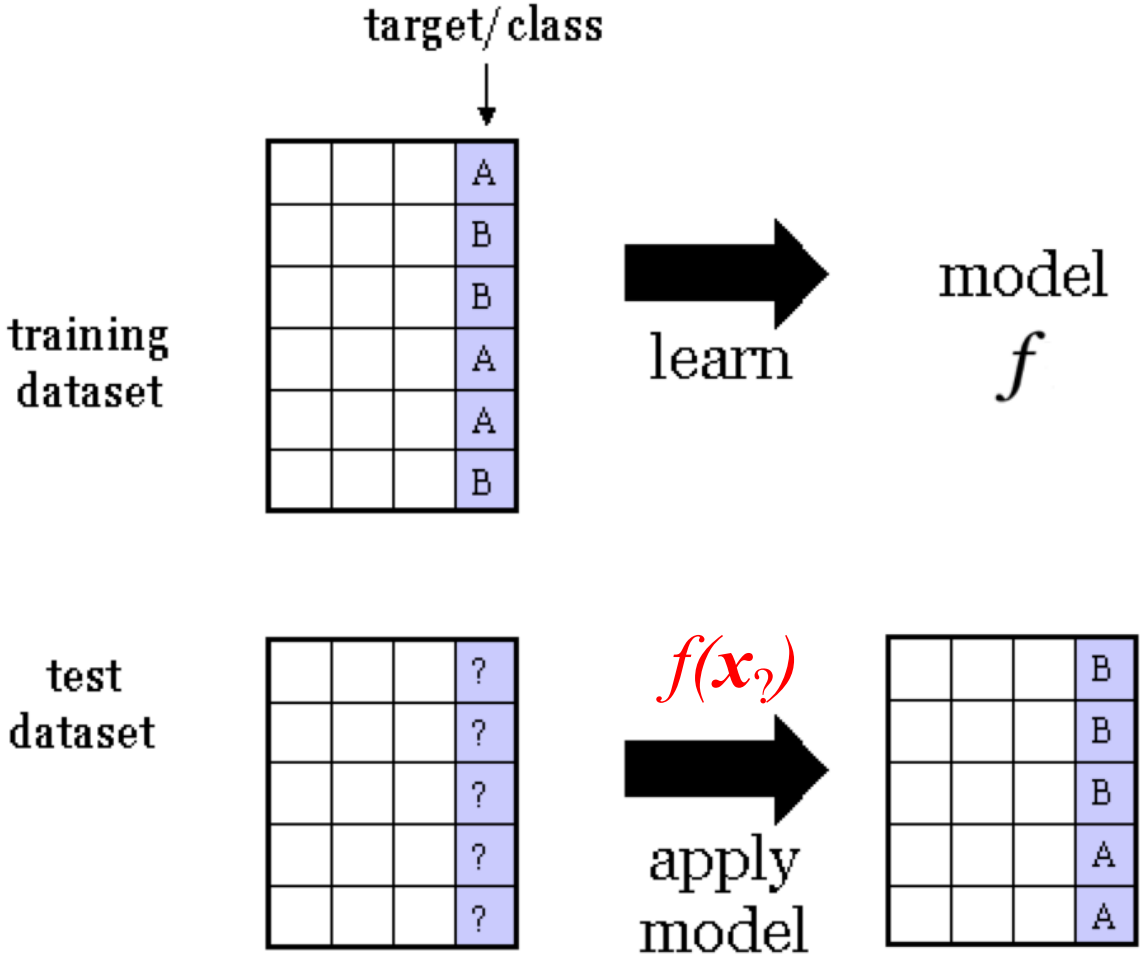
- **Data***/points/instances/examples/samples/records*: [rows]
- **Features***/attributes/dimensions/independent variables/covariates/predictors/regressors*: [columns, except the last]
- **Target***/outcome/response/label/dependent variable*: special column to be predicted [last column]

Main Types of Columns



- *Continuous*: a real number, for example, weight
- *Discrete*: a symbol, like "Good" or "Bad"

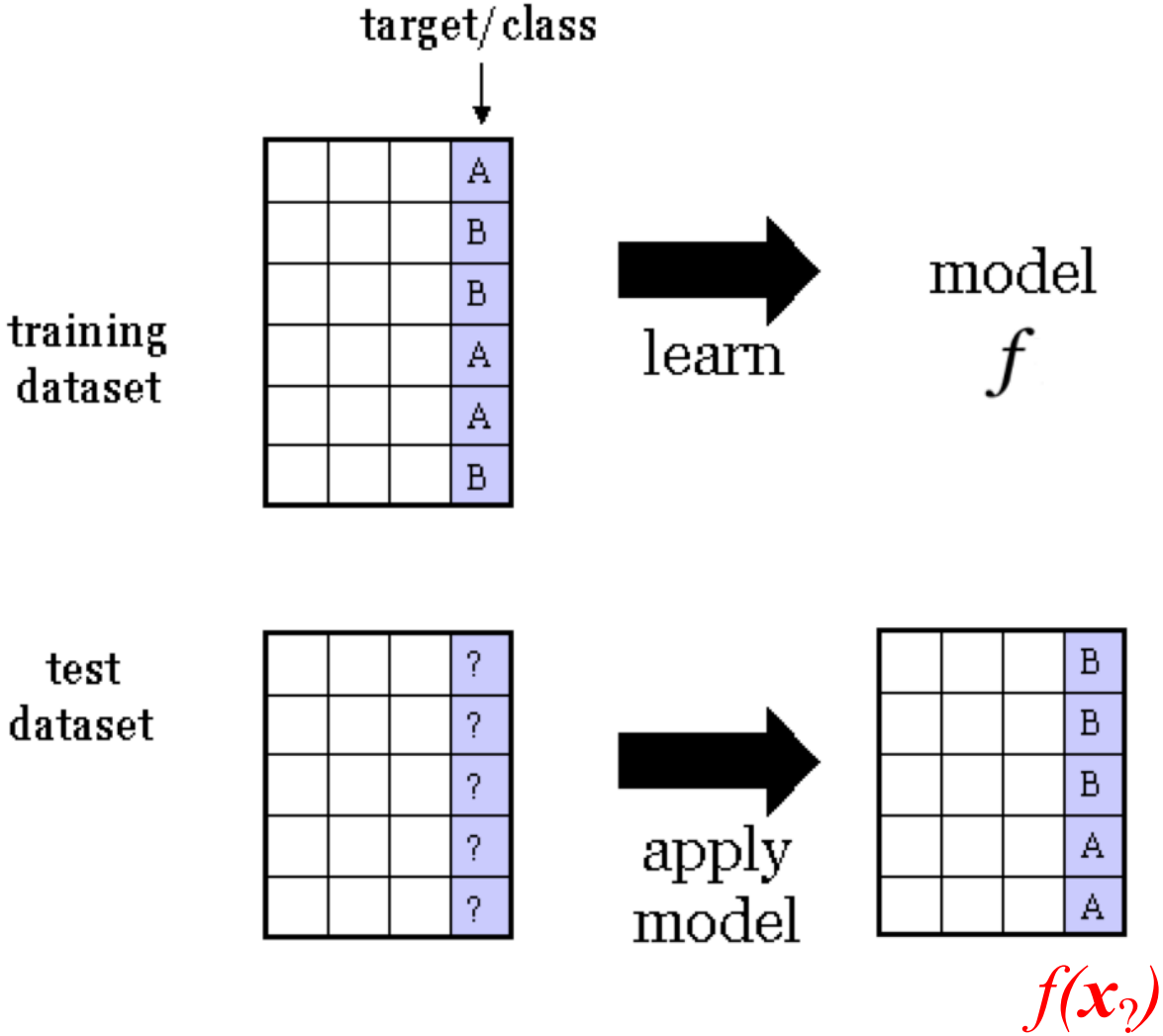
e.g. SUPERVISED Classification



Training dataset consists of **input**-**output** pairs

- e.g. target Y can be a discrete target variable

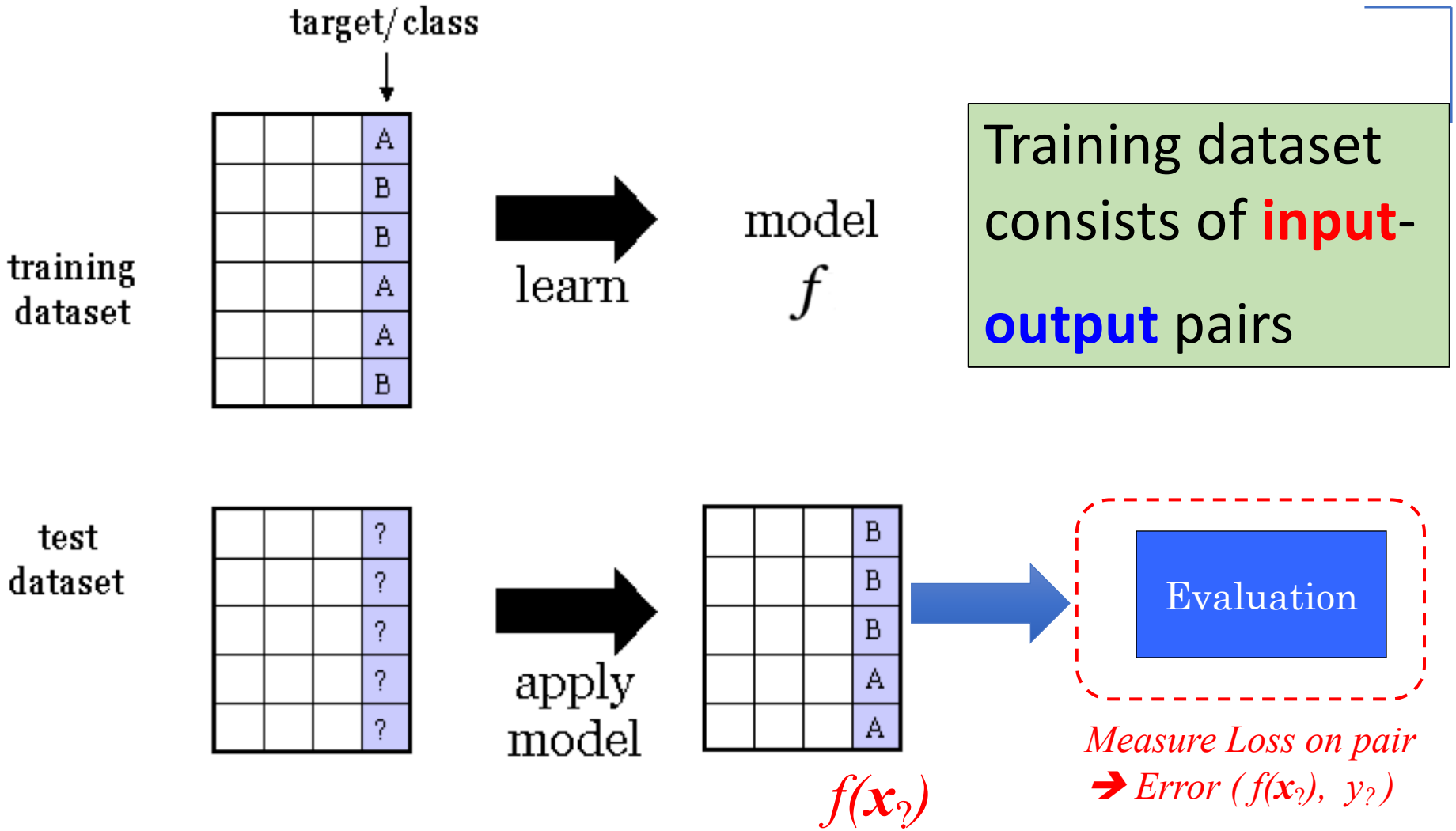
e.g. SUPERVISED Regression

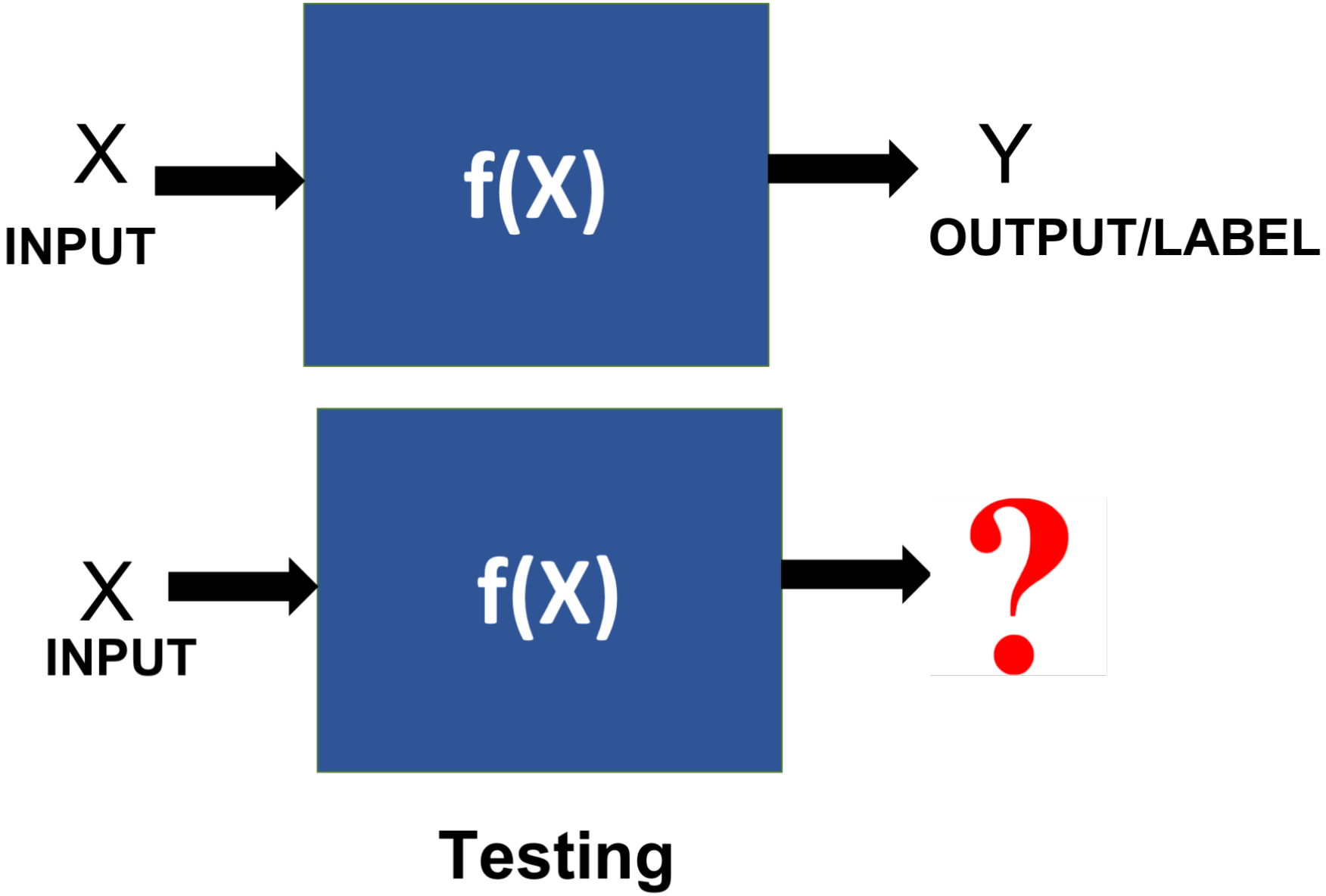


Training dataset consists of **input**-**output** pairs

- e.g. target Y can be a continuous target variable

How to know the program works well?




X
INPUT
f(X)
Y
OUTPUT/LABEL
X
INPUT
f(X)
?
Testing

training dataset

$$\mathbf{X}_{train} = \begin{bmatrix} -- & \mathbf{x}_1^T & -- \\ -- & \mathbf{x}_2^T & -- \\ \vdots & \vdots & \vdots \\ -- & \mathbf{x}_n^T & -- \end{bmatrix} \quad \vec{y}_{train} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

test dataset

$$\mathbf{X}_{test} = \begin{bmatrix} -- & \mathbf{x}_{n+1}^T & -- \\ -- & \mathbf{x}_{n+2}^T & -- \\ \vdots & \vdots & \vdots \\ -- & \mathbf{x}_{n+m}^T & -- \end{bmatrix} \quad \vec{y}_{test} = \begin{bmatrix} y_{n+1} \\ y_{n+2} \\ \vdots \\ y_{n+m} \end{bmatrix}$$

Notation

- Inputs
  - X, X_j (jth element of vector X) : random variables written in capital letter
  - p #input variables, n #observations
  - $\boldsymbol{X}$ **:** matrix written in bold capital
  - Vectors are assumed to be column vectors
  - Discrete inputs often described by characteristic vector (dummy variables)
- Outputs
  - quantitative Y
  - qualitative C (for categorical)
- Observed variables written in lower case
  - The i-th observed value of X is x_i and can be a scalar or a vector

a few sample slides

$$J(\theta) = (X\theta - y)^T (X\theta - y) \frac{1}{2}$$

$$= ((X\theta)^T - y^T)(X\theta - y)\frac{1}{2}$$

$$= (\theta^T X^T - y^T)(X\theta - y)\frac{1}{2}$$

$$= (\theta^T X^T X\theta - \theta^T X^T y - y^T X\theta + y^T y)\frac{1}{2}.$$

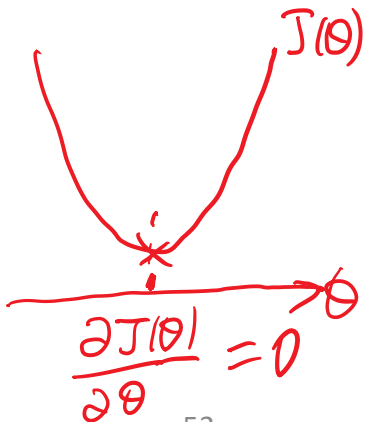
Since $\theta^T X^T y = y^T X\theta$

$\langle X\theta, y \rangle \quad \langle y, X\theta \rangle$

$$= (\theta^T X^T X\theta - 2\theta^T X^T y + y^T y)\frac{1}{2}$$

$\Rightarrow$ $J(\theta)$ quadratic func of θ;

1d case



Deriving the Maximum Likelihood Estimate for Bernoulli

maximize

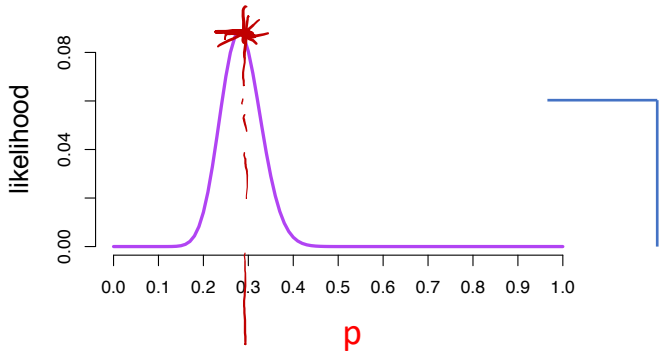
$$L(p) = p^x (1-p)^{n-x}$$

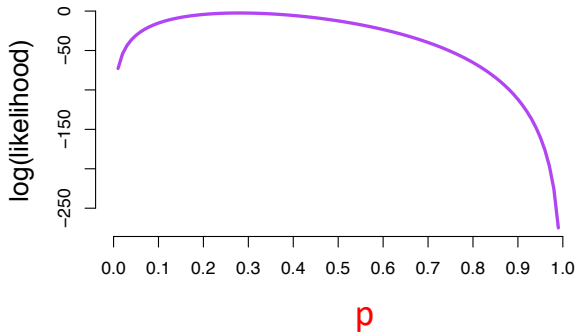
maximize

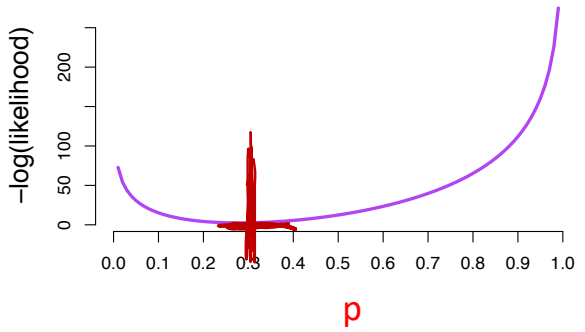
$$\log(L(p)) = \log\left[p^x (1-p)^{n-x} \right]$$

Minimize the negative log-likelihood

$$-l(p) = -\log\left[p^x (1-p)^{n-x} \right]$$





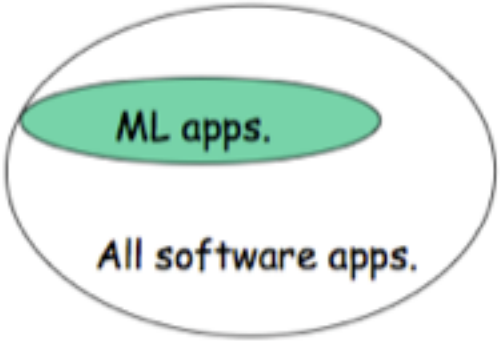


Today

- ❑ Course Logistics
- ❑ Machine Learning Basics
- ❑ A Rough Plan of Course Content
- ❑ Machine Learning History

MACHINE LEARNING IN COMPUTER SCIENCE

- Machine learning is already the preferred approach for
  - Speech recognition, natural language processing
  - Computer vision
  - Medical outcome analysis
  - Robot control …



- Why growing ?
  - Improved machine learning algorithms
  - Improved CPU / GPU powers
  - Increased data capture, new sensors, networking
  - Systems/Software too complex to control manually
  - Demand to self-customization for user, environment, ….

HISTORY OF MACHINE LEARNING

- 1950s
  - Samuel's checker player
  - Selfridge's Pandemonium
- 1960s:
  - Neural networks: Perceptron
  - Pattern recognition
  - Learning in the limit theory
  - Minsky and Papert prove limitations of Perceptron
- 1970s:
  - Symbolic concept induction
  - Winston's arch learner
  - Expert systems and the knowledge acquisition bottleneck
  - Quinlan's DT ID3
  - Michalski's AQ and soybean diagnosis
  - Scientific discovery with BACON
  - Mathematical discovery with AM

Adapted From Prof. Raymond J. Mooney's slides

HISTORY OF MACHINE LEARNING (CONT.)

- 1980s:
  - Advanced decision tree and rule learning
  - Explanation-based Learning (EBL)
  - Learning and planning and problem solving
  - Utility problem
  - Analogy
  - Cognitive architectures
  - Resurgence of neural networks (connectionism, backpropagation)
  - Valiant's PAC Learning Theory
  - Focus on experimental methodology
- 1990s
  - Data mining
  - Adaptive software agents and web applications
  - Text learning
  - Reinforcement learning (RL)
  - Inductive Logic Programming (ILP)
  - Ensembles: Bagging, Boosting, and Stacking
  - Bayes Net learning

Adapted From Prof. Raymond J. Mooney's slides

HISTORY OF MACHINE LEARNING (CONT.)

- 2000s
  - Support vector machines
  - Kernel methods
  - Graphical models
  - Statistical relational learning
  - Transfer learning
  - Sequence labeling
  - Collective classification and structured outputs
  - Computer Systems Applications
    - Compilers
    - Debugging
    - Graphics
    - Security (intrusion, virus, and worm detection)
  - Email management
  - Personalized assistants that learn
  - Learning in robotics and vision

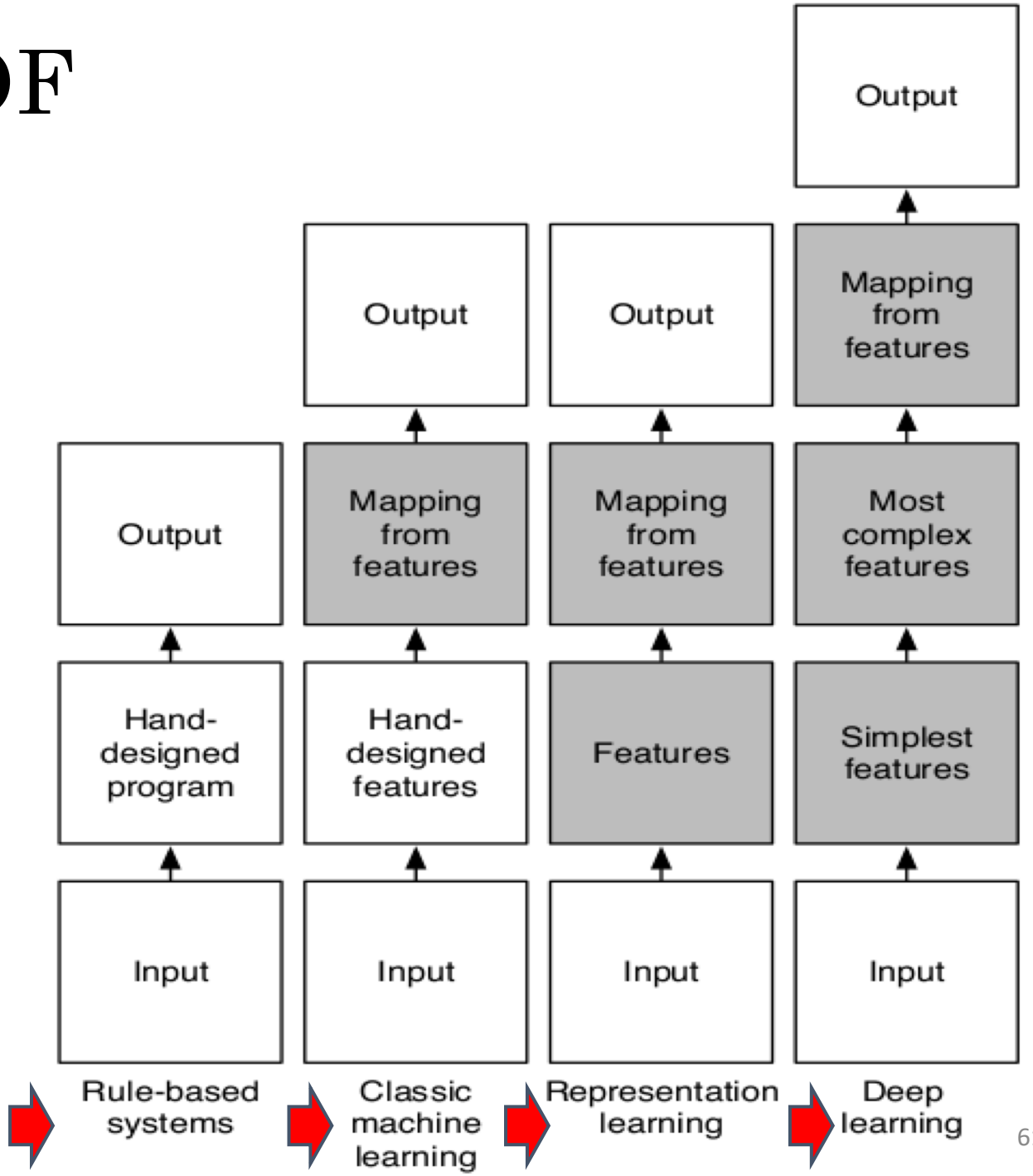
Adapted From Prof. Raymond J. Mooney's slides

HISTORY OF MACHINE LEARNING (CONT.)

- 2010s
  - Speech translation, voice recognition (e.g. SIRI)
  - Google search engine uses numerous machine learning techniques (e.g. grouping news, spelling corrector, improving search ranking, image retrieval, …..)
  - 23 and me (scan sample of person genome, predict likelihood of genetic disease, …)
  - DeepMind, Google Brain, …
  - IBM waston QA system
  - Machine Learning as a service (e.g. google prediction API, bigml.com, cloud autoML .)
  - IBM healthcare analytics
  - ……

Adapted From Prof. Raymond J. Mooney's slides

HISTORY OF MACHINE LEARNING (CONT.)



RELATED DISCIPLINES

- Artificial Intelligence
- Data Mining
- Probability and Statistics
- Information theory
- Numerical optimization
- Computational complexity theory
- Control theory (adaptive)
- Psychology (developmental, cognitive)
- Neurobiology
- Linguistics
- Philosophy

What are the goals of AI research?

Artifacts that THINK like HUMANS	Artifacts that THINK RATIONALLY
Artifacts that ACT like HUMANS	Artifacts that ACT RATIONALLY

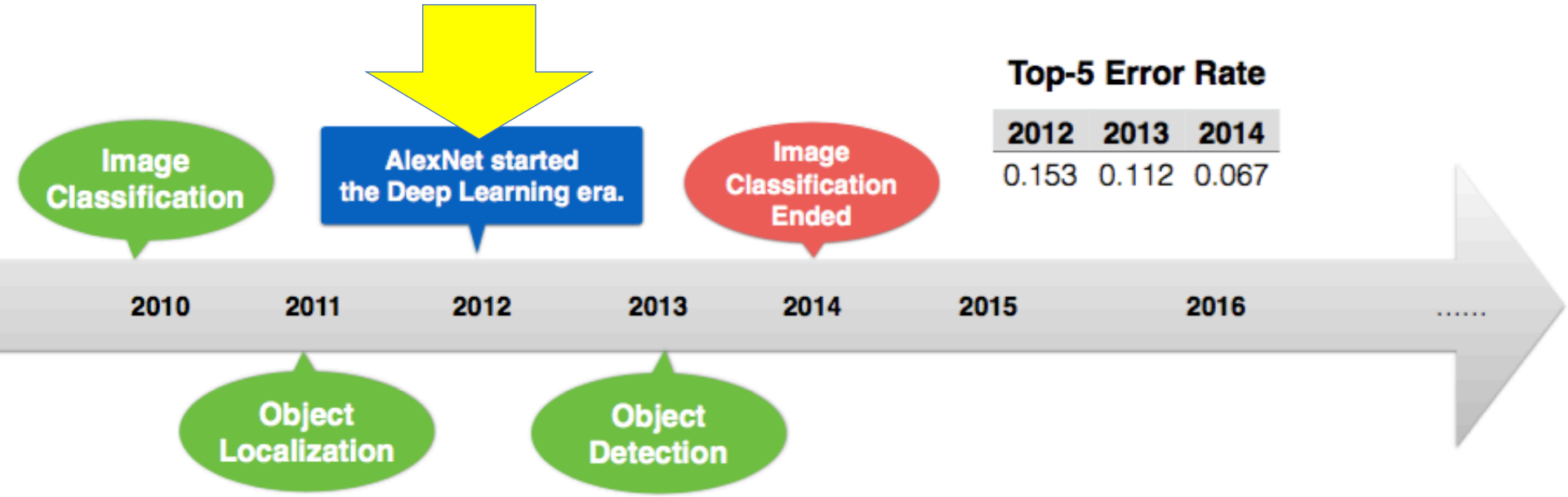
From: M.A. Papalaskar

How can we build more intelligent computer / machine ?

- Able to
  - **perceive the world**
  - **understand the world**
  - **react to the world**

- This needs
  - Basic speech capabilities
  - Basic vision capabilities
  - Language/semantic understanding
  - User behavior / emotion understanding
  - Able to act
  - Able to think ??

How can we build more intelligent computer / machine ? : Milestones in Recent Vision/AI Fields



72%, 2010

74%, 2011

85%, 2012

ImageNet Competition:

[Training on 1.2 million images [X] vs. 1000 different word labels [Y]]

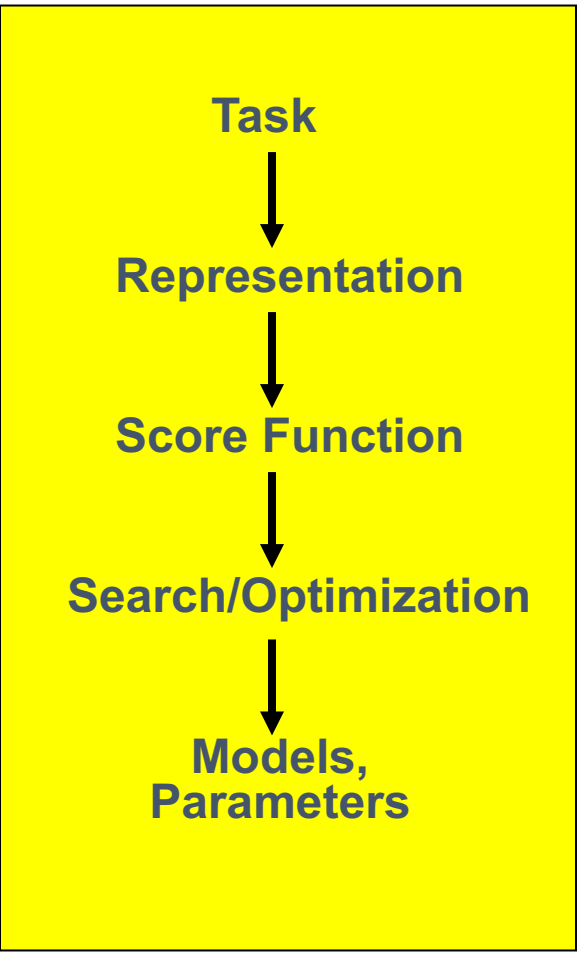
Detour: three planned programming assignments about AI tasks

- HW: Semantic language understanding (sentiment classification on movie review text)

- HW: Visual object recognition (labeling images about handwritten digits)

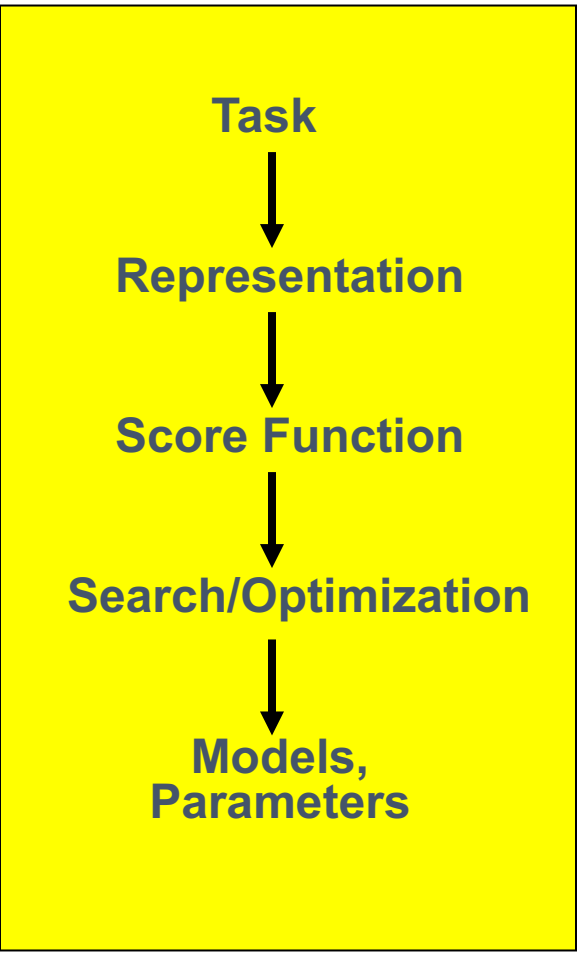
- HW: Audio speech recognition (unsupervised learning based speech recognition task)

HW1



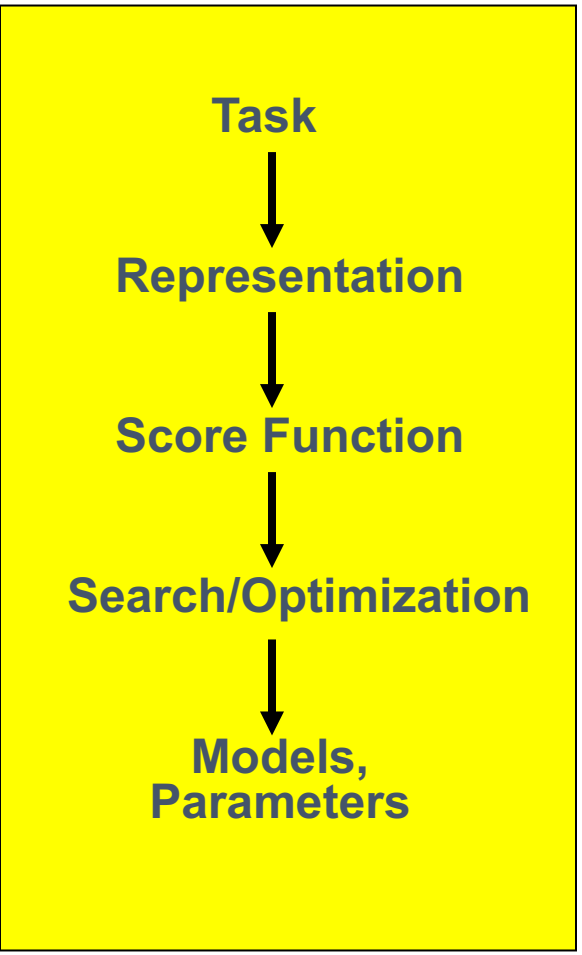
- Q1: Linear algebra review
- Q2: Linear regression + LOOCV
  - Regression
  - Evaluation pipeline
- Q3: Machine learning pipeline practice
  - Basic pipeline
  - GUI Toolbox
  - Evaluation

HW2



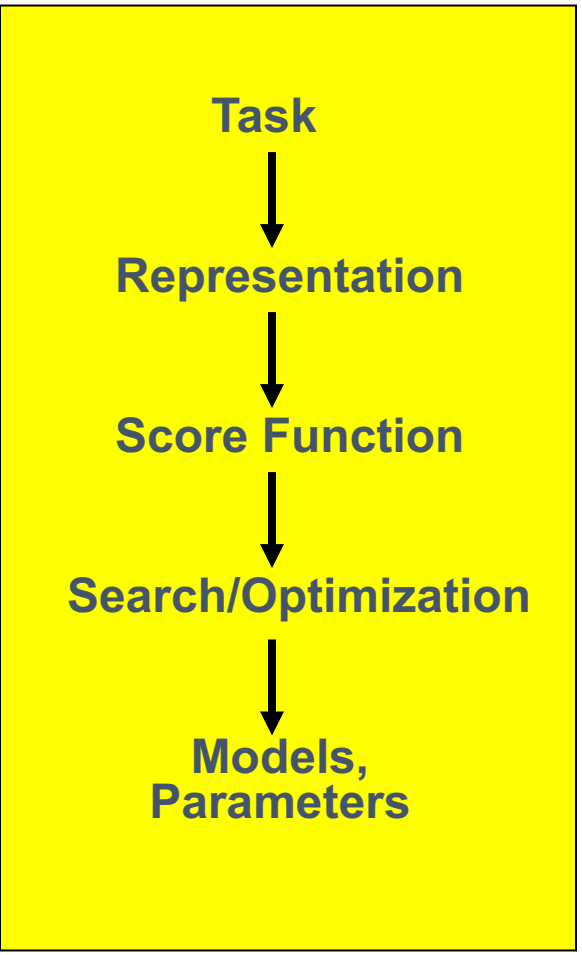
- Q1: Linear regression model fitting
  - Data loading
  - Basic linear regression
  - Three ways to train : Normal equation / SGD / Batch GD
  - Polynomial regression
- Q2: Ridge regression
  - Math derivation of ridge
  - Understand why/how Ridge
  - Model selection of Ridge with K-CV

HW5



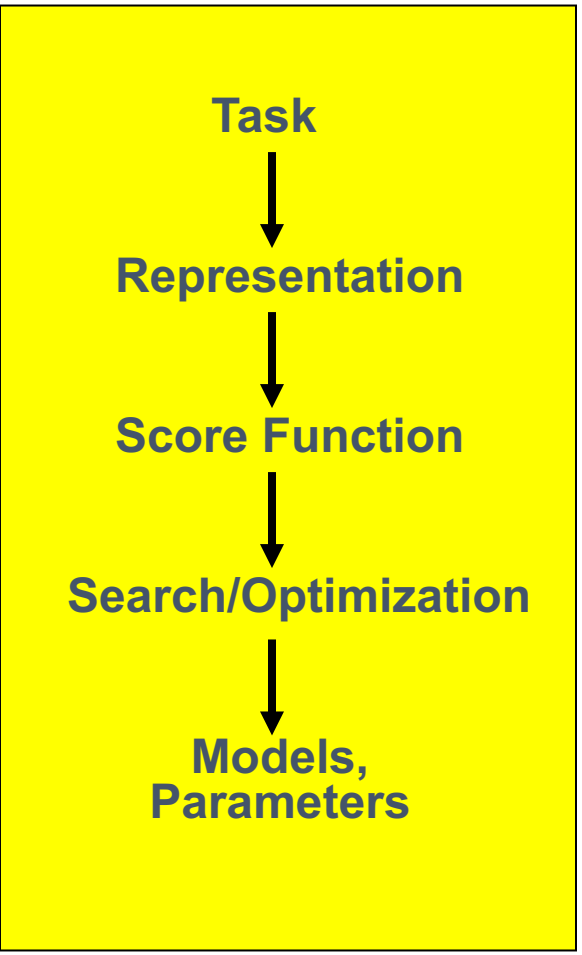
- Q1: Support Vector Machines with Scikit-Learn
  - Data preprocessing
  - How to use SVM package
  - Model selection for SVM
  - Model selection pipeline with train-vali, or train-CV; then test

HW5



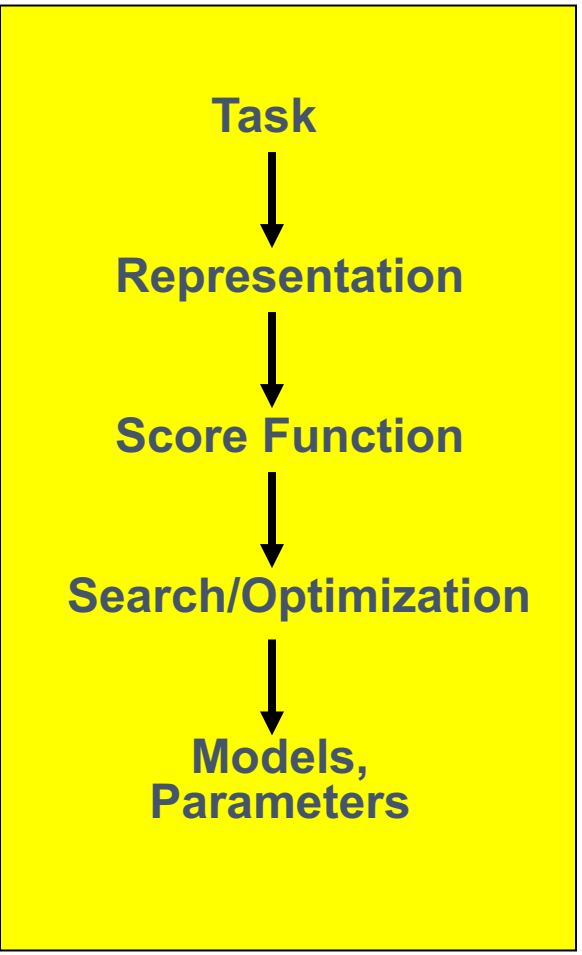
- Q1: Naive Bayes Classifier for Text-base Movie Review Classification
  - Preprocessing of text samples
  - BOW Document Representation
  - Multinomial Naive Bayes Classifier
    - BOW way
    - Language model way
  - Multivariate Bernoulli Naive Bayes Classifier

HW6



- Q1: Neural Network Tensorflow Playground
  - Interactive learning of MLP
  - Feature engineering vs.
  - Feature learning
- Q2: Image Classification
  - Tool using
  - DT / KNN / NN
  - PCA effect for image classification

HW6



- Q3: Unsupervised Clustering of audio data and consensus data
  - Data loading
  - K-mean clustering
  - GMM clustering
  - How to find K: knee-finding plot
  - How to measure clustering: purityMetric

References

- Prof. Andrew Moore's tutorials
- Prof. Raymond J. Mooney's slides
- Prof. Alexander Gray's slides
- Prof. Eric Xing's slides
- http://scikit-learn.org/
- Hastie, Trevor, et al. The elements of statistical learning. Vol. 2. No. 1. New York: Springer, 2009.
- Prof. M.A. Papalaskar's slides