UVA CS 4774: Machine Learning

Lecture 1: Introduction

Dr. Yanjun Qi

University of Virginia
Department of Computer Science

Roadmap

- Course Logistics
- History and Now
- A Rough Plan of Course Content

ATT:

- I will video-record only when I teach
- One TA will act as an coordinator, please type in questions via Chats / He will read out and ask me to answer
- Each session will be arranged as three small modules --- each about 20mins (each into a different recorded video)

Welcome

- Course Website:

- https://qiyanjun.github.io/2020f-UVA-CS-MachineLearningDeep/

We focus on learning fundamental principles, algorithm design and deep learning methods and applications.

Objective

- To help students get 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
  - Able to understand a few complex machine learning methods at the source code level
  - To re-produce / or invent one cutting-edge machine learning algorithm for GOOD USE

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
  - Multiple good reference books are shared via CourseWeb

- 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.
  - Python is required for all programming assignments

Assignments

- Assignments (50%, with five assignments)
- See policy in https://qiyanjun.github.io/2020f-UVA-CS-MachineLearningDeep//About/

Quiz

- Class quizzes (20%): Each takes 10 mins via google form;
  - We will have a total of 12 quizzes
  - Your top 10 scored will be counted into 20%
  - In the middle of our Zoom Synchronous sessions

- See policy in https://qiyanjun.github.io/2020f-UVA-CS-MachineLearningDeep//About/

Course Project

- Final project (30%): Three potential types:

 - a. To produce one machine learning project on cutting-edge data applications with health or social impacts

 - b. Survey and benchmark multiple pytorch library with a shared goal

 - c. To Reproduce a cutting-edge machine learning paper, for instance from Top Venues' most cited recent papers

- See policy in https://qiyanjun.github.io/2020f-UVA-CS-MachineLearningDeep//About/

Thank You

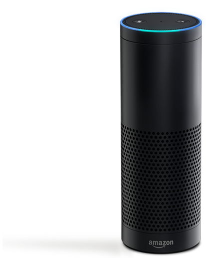
Roadmap

- Course Logistics
- History and Now
- A Rough Plan of Course Content

Artificial Intelligence Today

Watson (IBM)

DeepMind (Google)



Echo
(Amazon)

SIRI
(Apple)

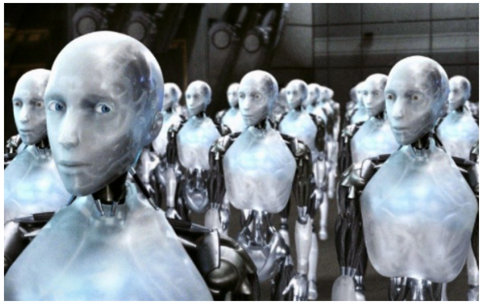
Boston Dynamics

Impact

Economic, cultural, social,
… endless disruption

Labour - McKinsey 58%
of jobs automated

Martin Ford,
Rise of the Robots



Elon Musk, artificial intelligence...
existential threat

Image credits: sibaraki@acm.org

Artificial intelligence (AI)

The study of computer systems that attempt to model and apply the intelligence of the human mind.

What defines “intelligence”?

Why is it that we assume humans are intelligent?

Are monkeys intelligent? Dogs? Ants? Pine trees?

How to build more intelligent computer / machine ?

- Able to perceive the world,
  - e.g., objective recognition, speech recognition, …
- Able to understand the world,
  - e.g., machine translation, text semantic understanding
- Able to Interact with the world,
  - e.g., AlphaGo, AlphaZero, self-driving cars, …
- Able to think / reason / learn,
  - e.g., learn to program programs, learn to search deepNN architecture, …
- Able to imagine / to make analogy,
  - e.g., learn to draw with styles, ……

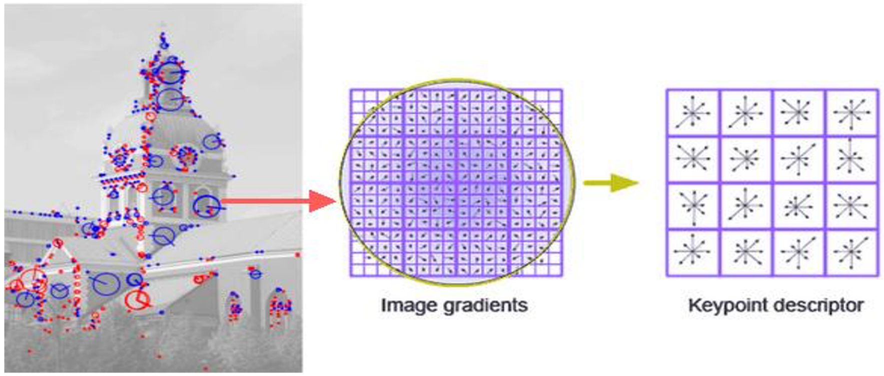
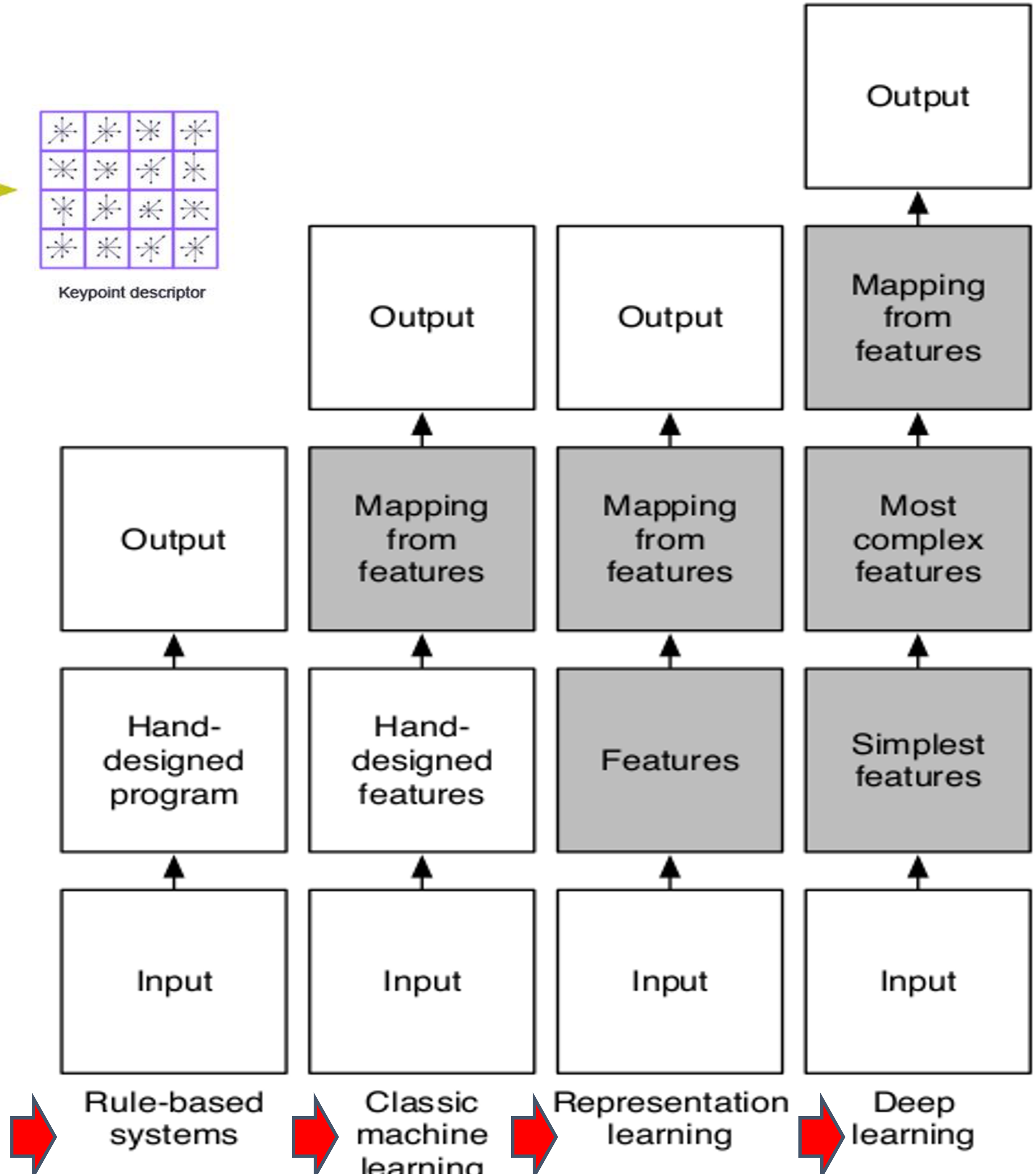


Image gradients
Keypoint descriptor


Output
Hand-designed program
Input
Rule-based systems
Output
Mapping from features
Hand-designed features
Input
Classic machine learning
Output
Mapping from features
Features
Input
Representation learning
Output
Mapping from features
Most complex features
Simplest features
Input
Deep learning

Early History

- In 1950 English mathematician Alan Turing wrote a landmark paper titled "Computing Machinery and Intelligence" that asked the question: "Can machines think?"
- Further work came out of a 1956 workshop at Dartmouth sponsored by John McCarthy. In the proposal for that workshop, he coined the phrase a "study of artificial intelligence"
- Expert systems (70s, 80s)
  - A software system based the knowledge of human experts;
  - Rule-based system
  - processes rules to draw conclusions
  - Idea is to give AI systems lots of information to start with

MIT Technology Review

10 Breakthrough Technologies 2013

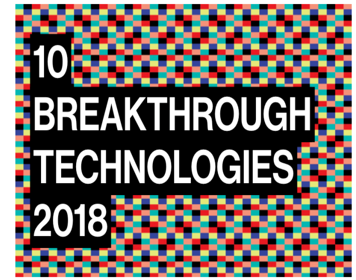
Think of the most frustrating, intractable, or simply annoying problems you can imagine. Now think about what technology is doing to fix them. That's what we did in coming up with our annual list of 10 Breakthrough Technologies. We're looking for technologies that we believe will expand the scope of human possibilities.

Deep Learning

10 Breakthrough Technologies 2017

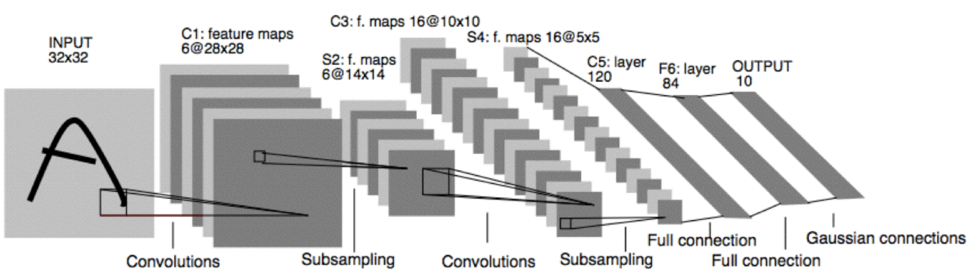
These technologies all have staying power. They will affect the economy and our politics, improve medicine, or influence our culture. Some are unfolding now; others will take a decade or more to develop. But you should know about all of them right now.

Deep Reinforcement Learning



Generative Adversarial Network (GAN)

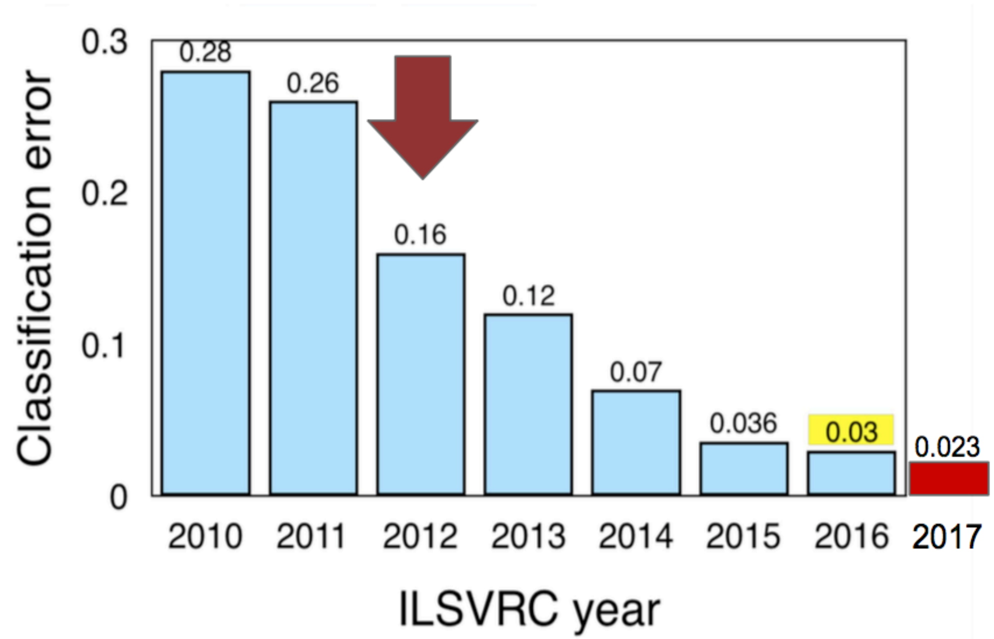
- **1952-1969 Enthusiasm:** Lots of work on neural networks
- **1990s:** Convolutional neural network (CNN) and Recurrent neural network (RNN) were invented



Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86(11): 2278–2324, 1998.

ImageNet Challenge

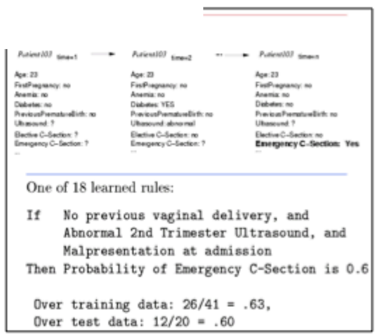
- 2010-11: hand-crafted computer vision pipelines
- 2012-2016: ConvNets
  - 2012: AlexNet
    - major deep learning success
  - 2013: ZFNet
    - improvements over AlexNet
  - 2014
    - VGGNet: deeper, simpler
    - InceptionNet: deeper, faster
  - 2015
    - ResNet: even deeper
  - 2016
    - ensembled networks
  - 2017
    - Squeeze and Excitation Network



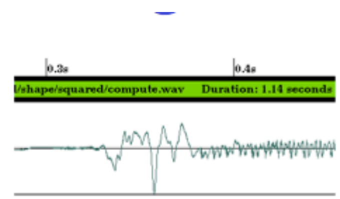
Adapt from From NIPS 2017 DL Trend Tutorial

Deep Learning is Changing the World

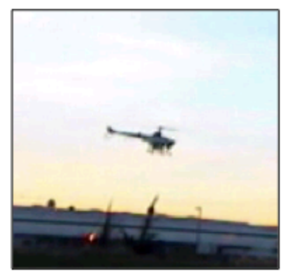
How may I help you, human?



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 !

Reason of Recent Deep Learning breakthroughs:

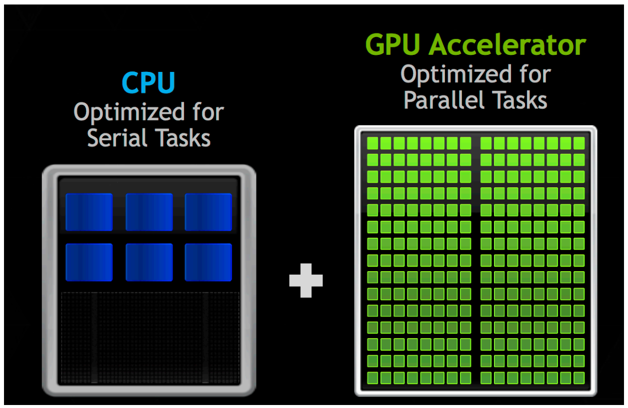
Plenty of (Labeled) Data for ML	Advanced Computer Architecture that fits DNNs	Powerful DNN platforms / Libraries

Reason: Plenty of (Labeled) Data

- Text: trillions of words of English + other languages
- Visual: billions of images and videos
- Audio: thousands of hours of speech per day
- User activity: queries, user page clicks, map requests, etc,
- Knowledge graph: billions of labeled relational triplets
- Genomics data:
- Medical Imaging data:

Dr. Jeff Dean's talk

Reason: Advanced Computer Architecture that fits DNNs



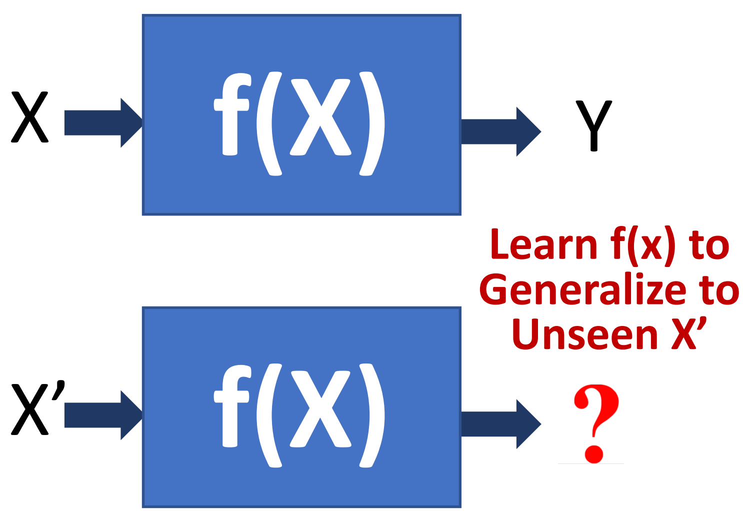
http://www.nvidia.com/content/events/geoInt2015/LBrown_DL.pdf

	Neural Networks	GPUs
Inherently Parallel	✓	✓
Matrix Operations	✓	✓
FLOPS	✓	✓

GPUs deliver --
- *same or better prediction accuracy*
- *faster results*
- *smaller footprint*
- *lower power*

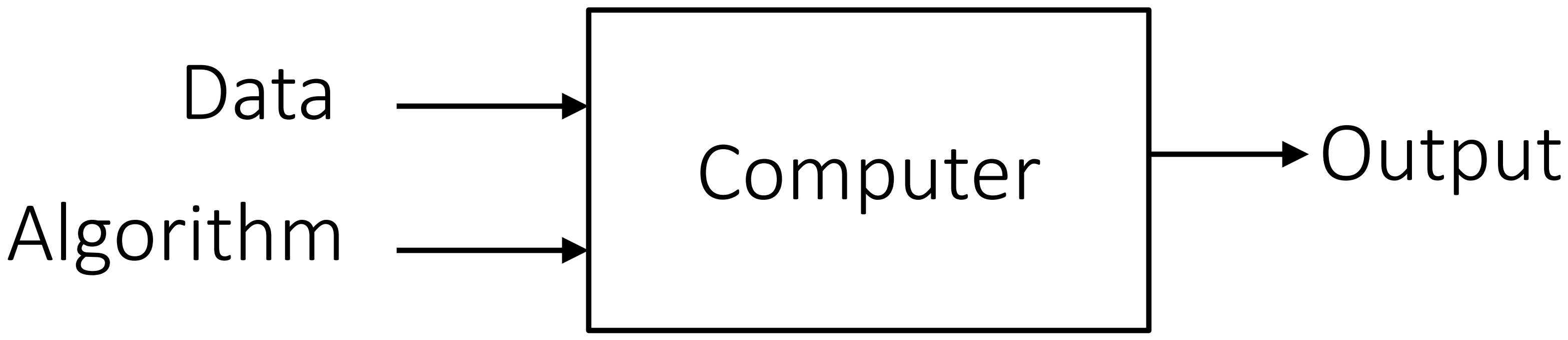
Reason: Data-Driven Machine Learning Algorithms and Platforms

- Need inductive reasoning
  - Generalizations from observed data to unseen data

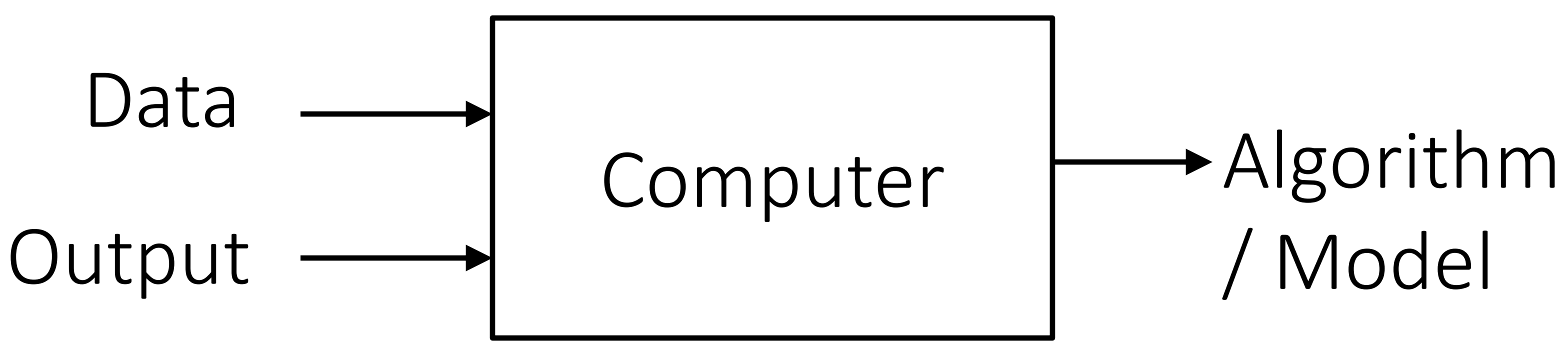


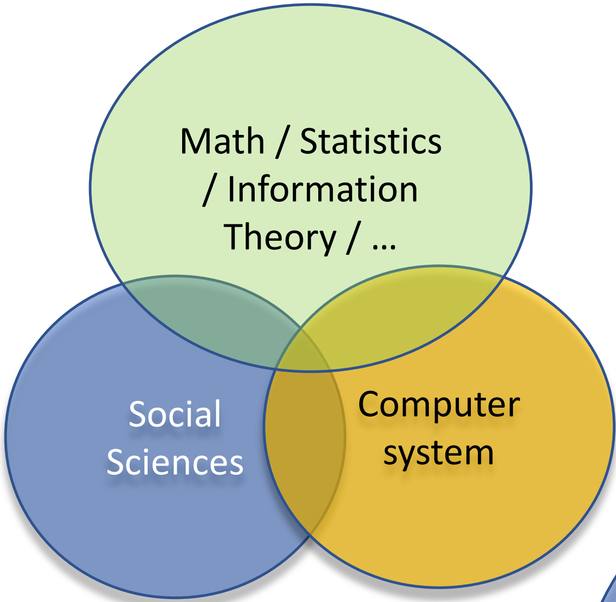
- Able to to build computer systems that can learn and adapt from their experience
- Well-engineered software architectures to build upon
- Provide prediction accuracy
- Create software that improves over time

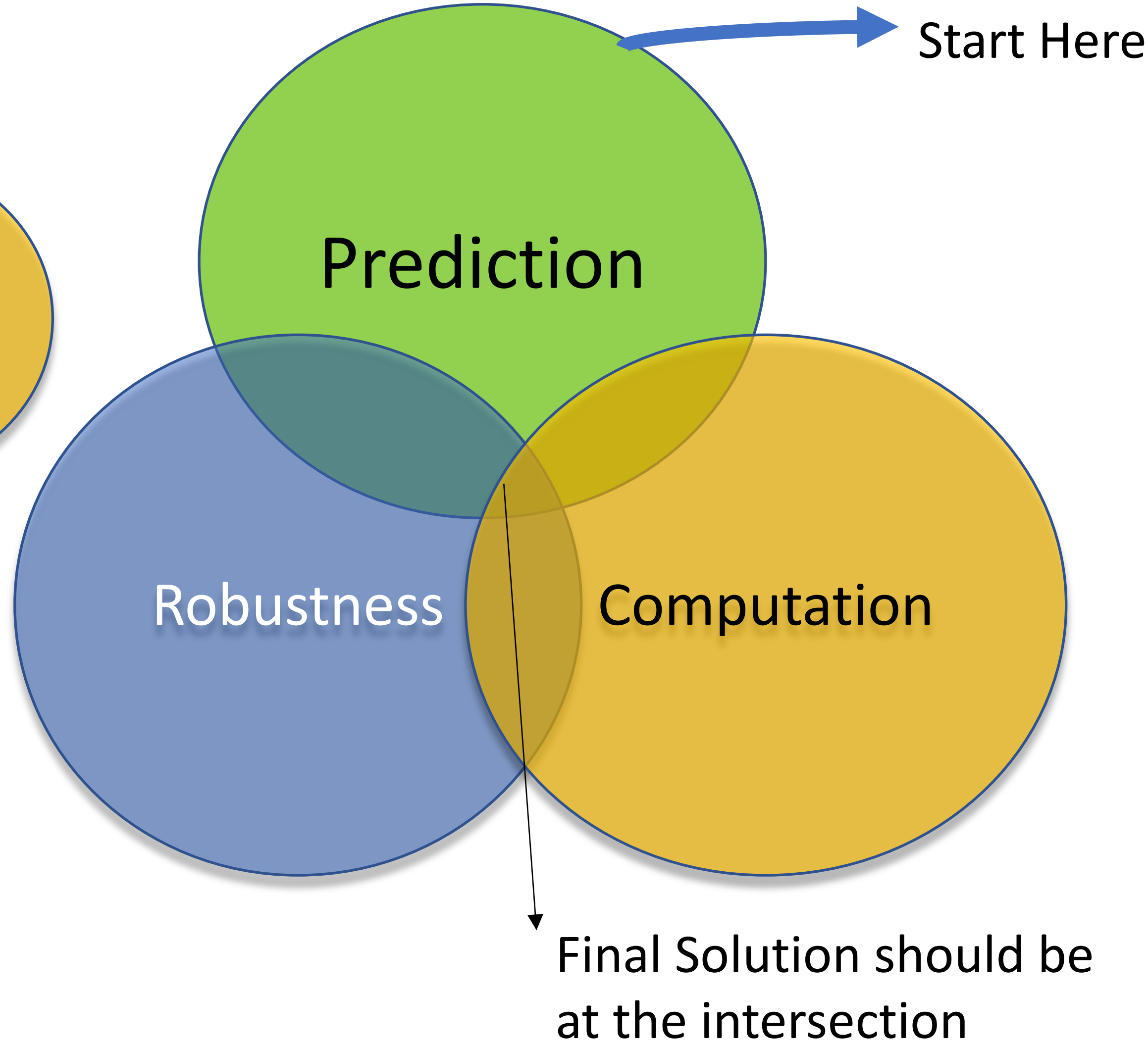
Traditional Programming



Machine Learning




Math / Statistics
/ Information
Theory / …
Social
Sciences
Computer
system


Start Here
Prediction
Robustness
Computation
Final Solution should be
at the intersection

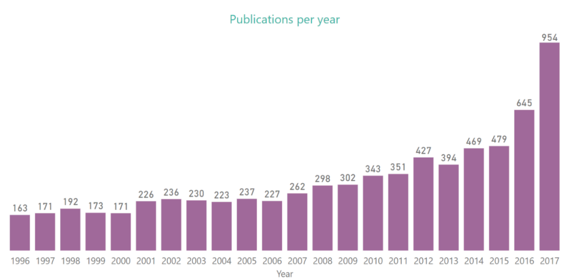
Future?

ML + Digital Data Platforms: Unprecedented Era

Hyper time compression
new disruptive innovations

Extreme convergence
of multiple domains

Exponential accelerating automation
– smart sensors and the billion IoT devices



Universal connectivity linked by a digital mesh

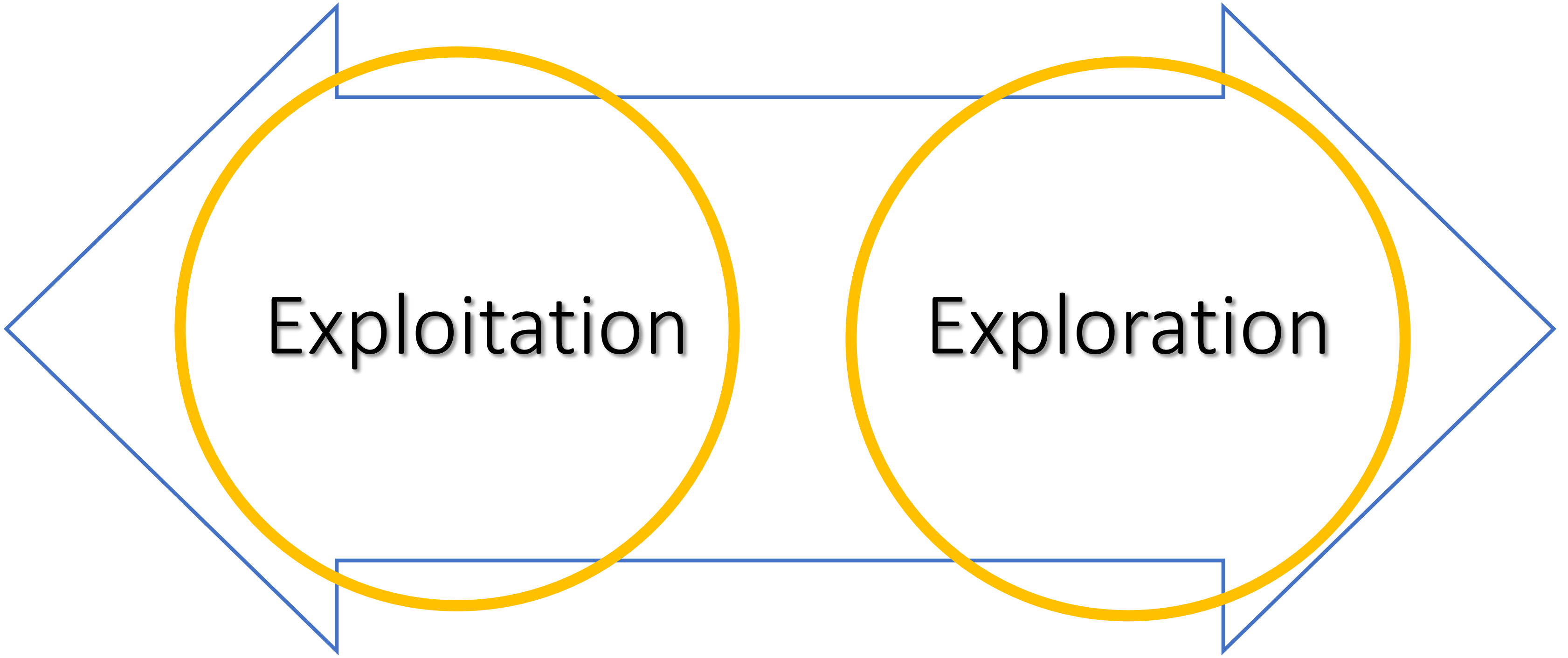
We always overestimate the change
that will occur in the next two years
and underestimate the change
that will occur in the next ten.

- Bill Gates, The Road Ahead, 1996

General Lessons for Excellence

- Good breath in fundamentals is key

- Strength in particular targeted topics help standing out


Exploitation
Exploration

Highly Recommend
Two Extra-curriculum books:

1. Book: By Dr. Domingos: Master Algorithm

So How Do Computers Discover New Knowledge?

1. **Symbolists**--Fill in gaps in existing knowledge
2. **Connectionists**--Emulate the brain
3. **Evolutionists**--Simulate evolution
4. **Bayesians**--Systematically reduce uncertainty
5. **Analogizers**--Notice similarities between old and new

SRC: Pedro Domingos ACM Webinar Nov 2015
http://learning.acm.org/multimedia.cfm

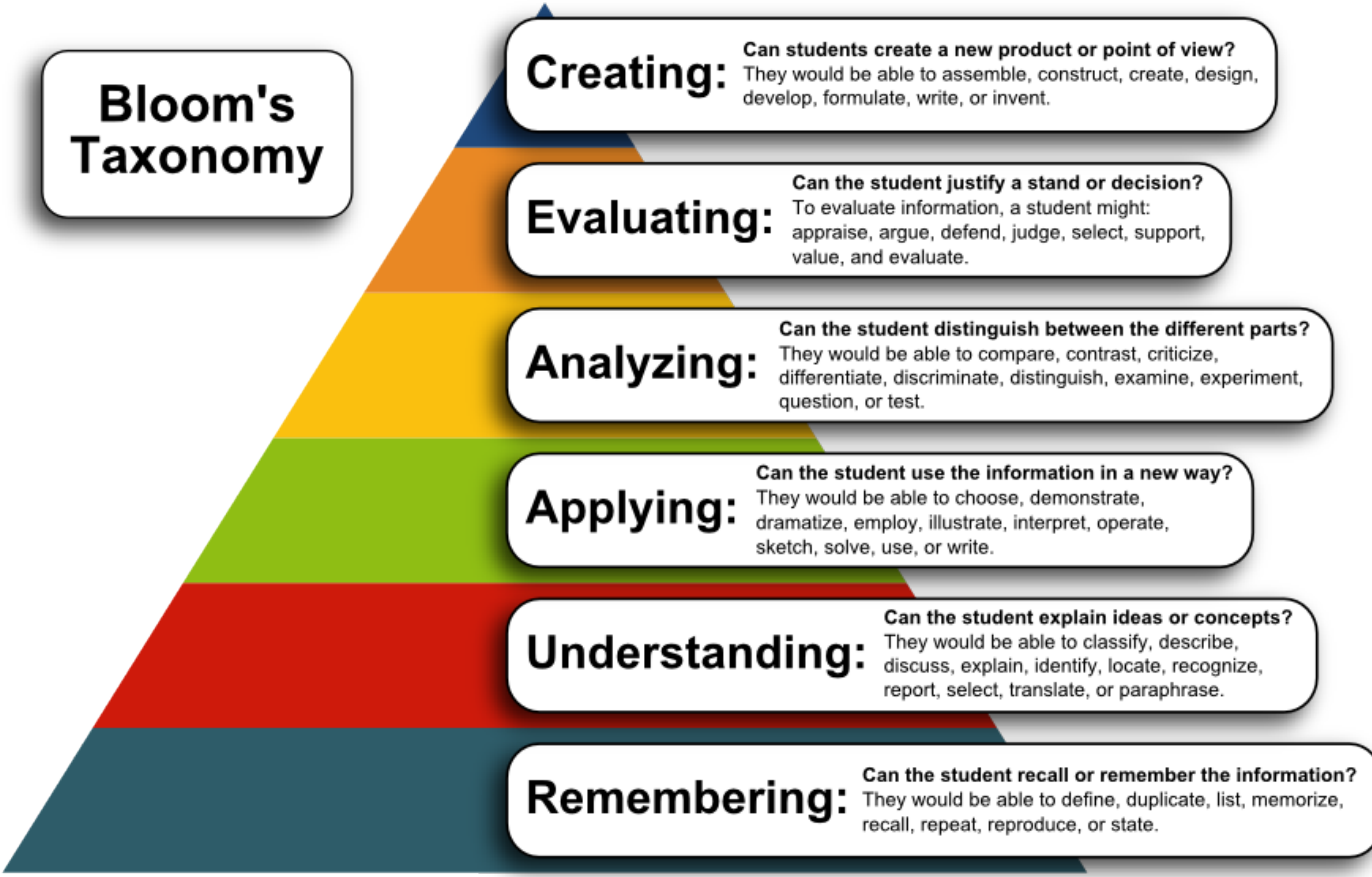
Highly Recommend Two Extra-curriculum books

- 2. Book: [Homo Deus- A Brief History of Tomorrow]
  - https://www.goodreads.com/book/show/31138556-homo-deus
  - “Homo Deus explores the projects, dreams and nightmares that will shape the twenty-first century—from overcoming death to creating artificial life. It asks the fundamental questions: Where do we go from here? And how will we protect this fragile world from our own destructive powers? This is the next stage of evolution. This is Homo Deus.””

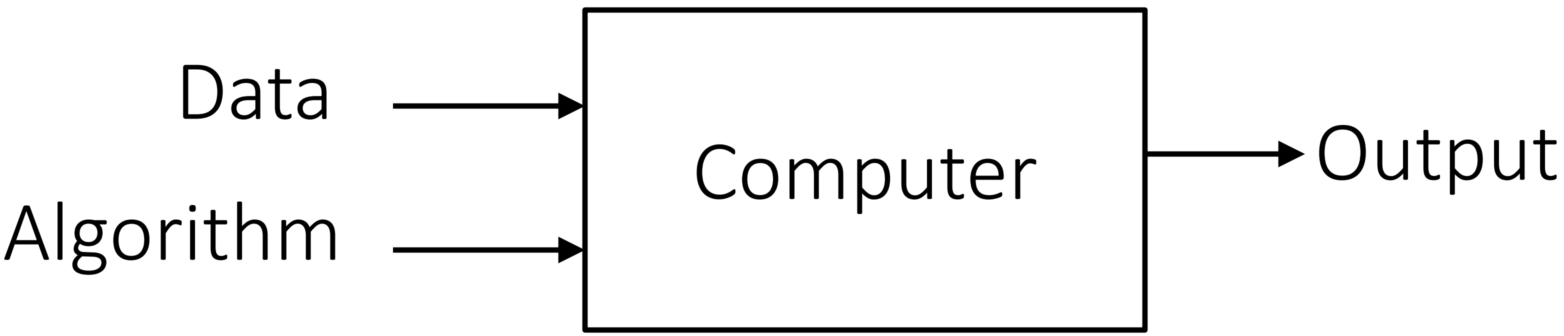
Thank You

Roadmap

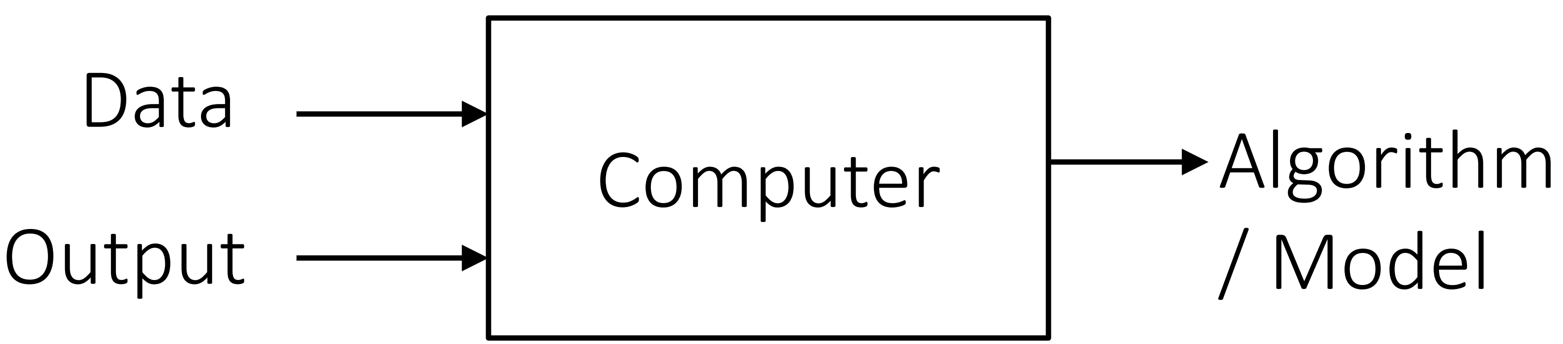
- Course Logistics
- History and Now
- A Rough Plan of Course Content


Bloom's Taxonomy
Creating: Can students create a new product or point of view? They would be able to assemble, construct, create, design, develop, formulate, write, or invent.
Evaluating: Can the student justify a stand or decision? To evaluate information, a student might: appraise, argue, defend, judge, select, support, value, and evaluate.
Analyzing: Can the student distinguish between the different parts? They would be able to compare, contrast, criticize, differentiate, discriminate, distinguish, examine, experiment, question, or test.
Applying: Can the student use the information in a new way? They would be able to choose, demonstrate, dramatize, employ, illustrate, interpret, operate, sketch, solve, use, or write.
Understanding: Can the student explain ideas or concepts? They would be able to classify, describe, discuss, explain, identify, locate, recognize, report, select, translate, or paraphrase.
Remembering: Can the student recall or remember the information? They would be able to define, duplicate, list, memorize, recall, repeat, reproduce, or state.

Traditional Programming



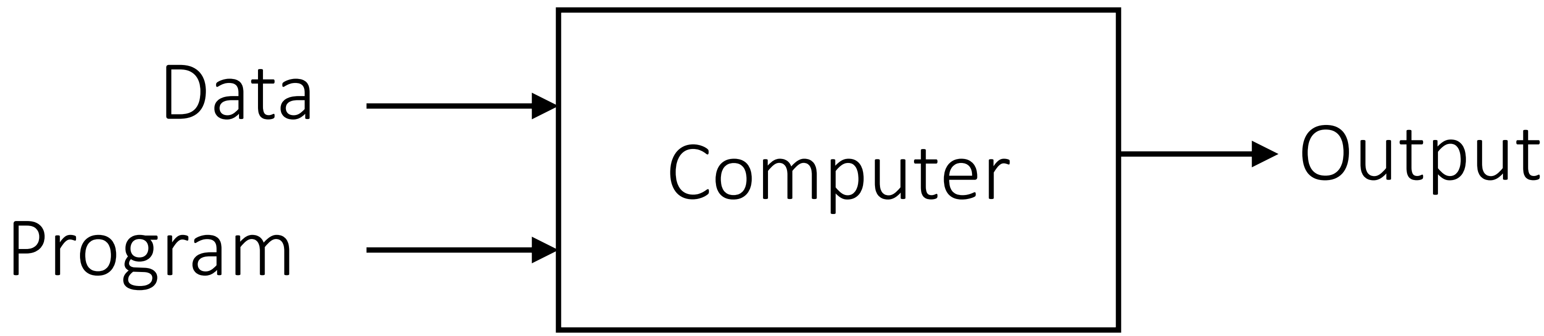
Machine Learning



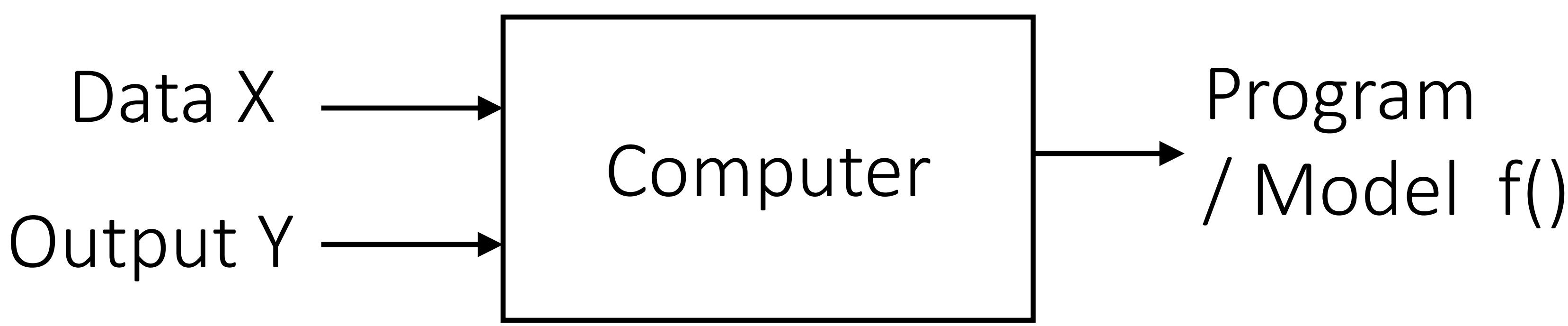
BASICS OF MACHINE LEARNING

- "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)

Traditional Programming



Machine Learning (training phase)



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(x) of each example 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

$\longrightarrow$

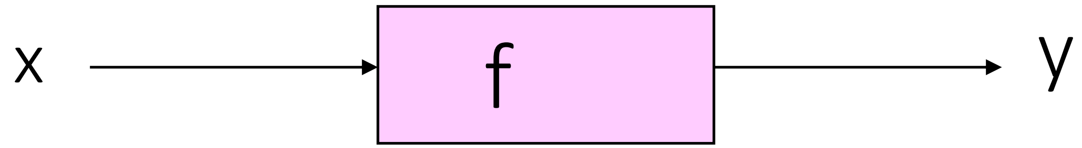
y

-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

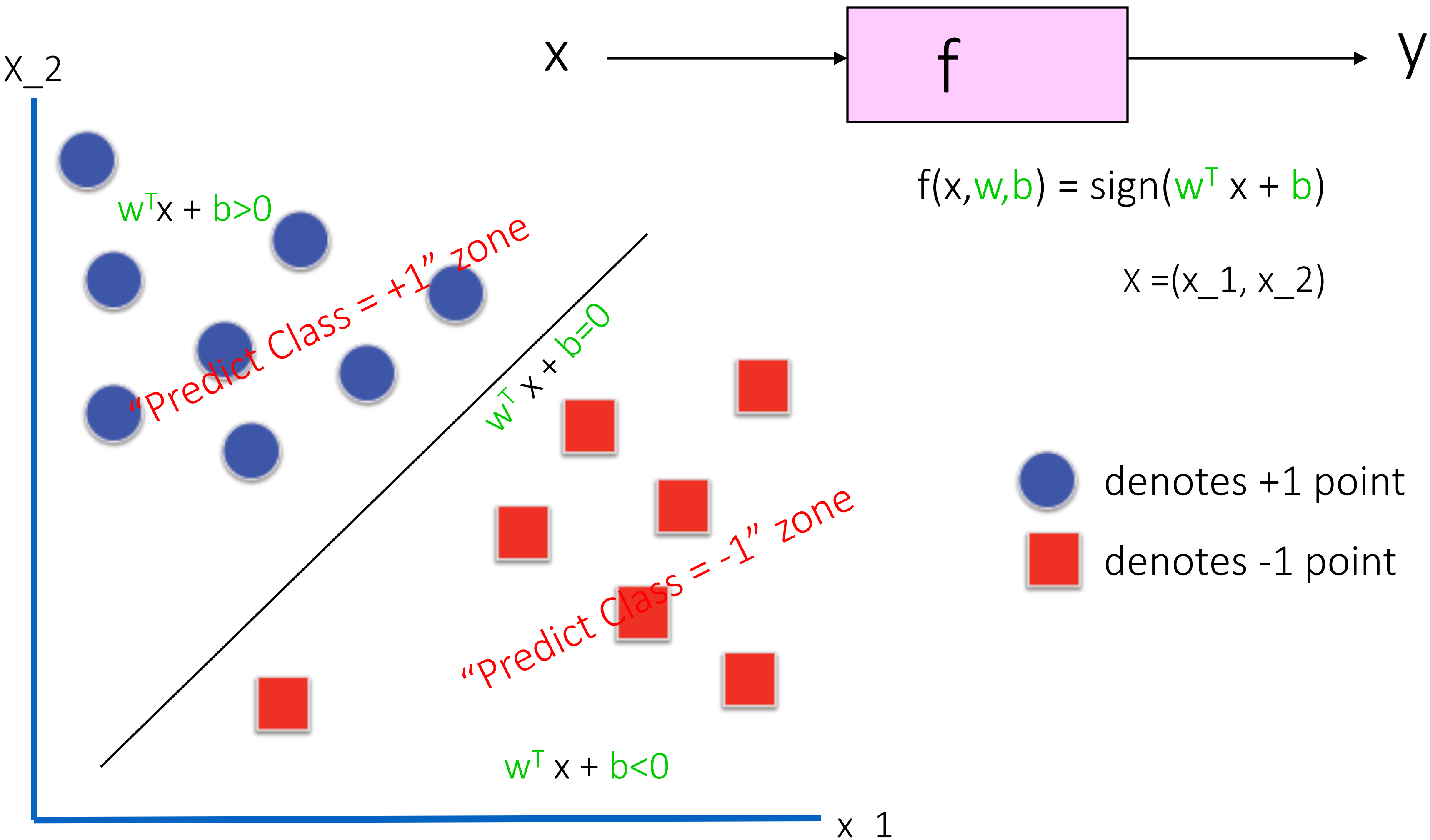


e.g.: Binary y / Linear f / X as R^2

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

X =(x_1, x_2)

SUPERVISED Linear Binary Classifier



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

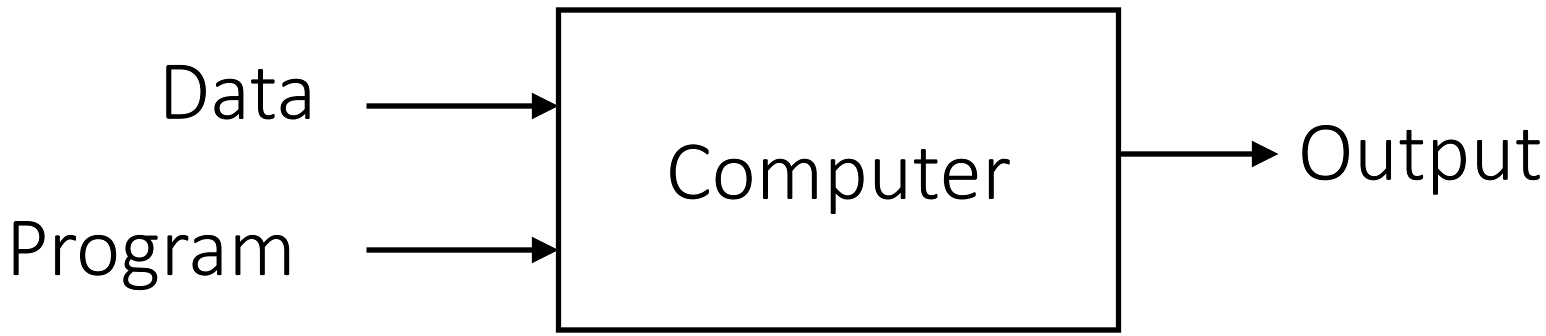
$X = (x\_1, x\_2)$

denotes +1 point

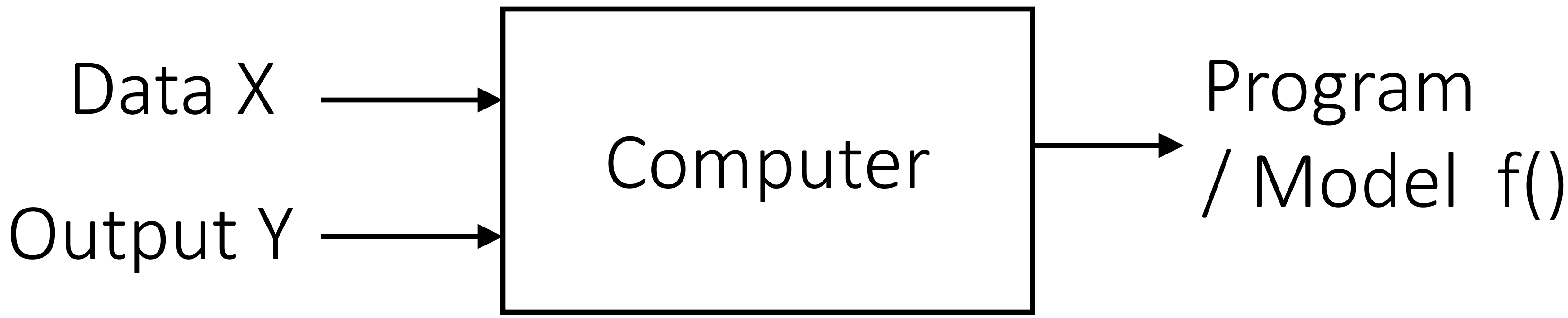
denotes -1 point

Courtesy slide from Prof. Andrew Moore's tutorial

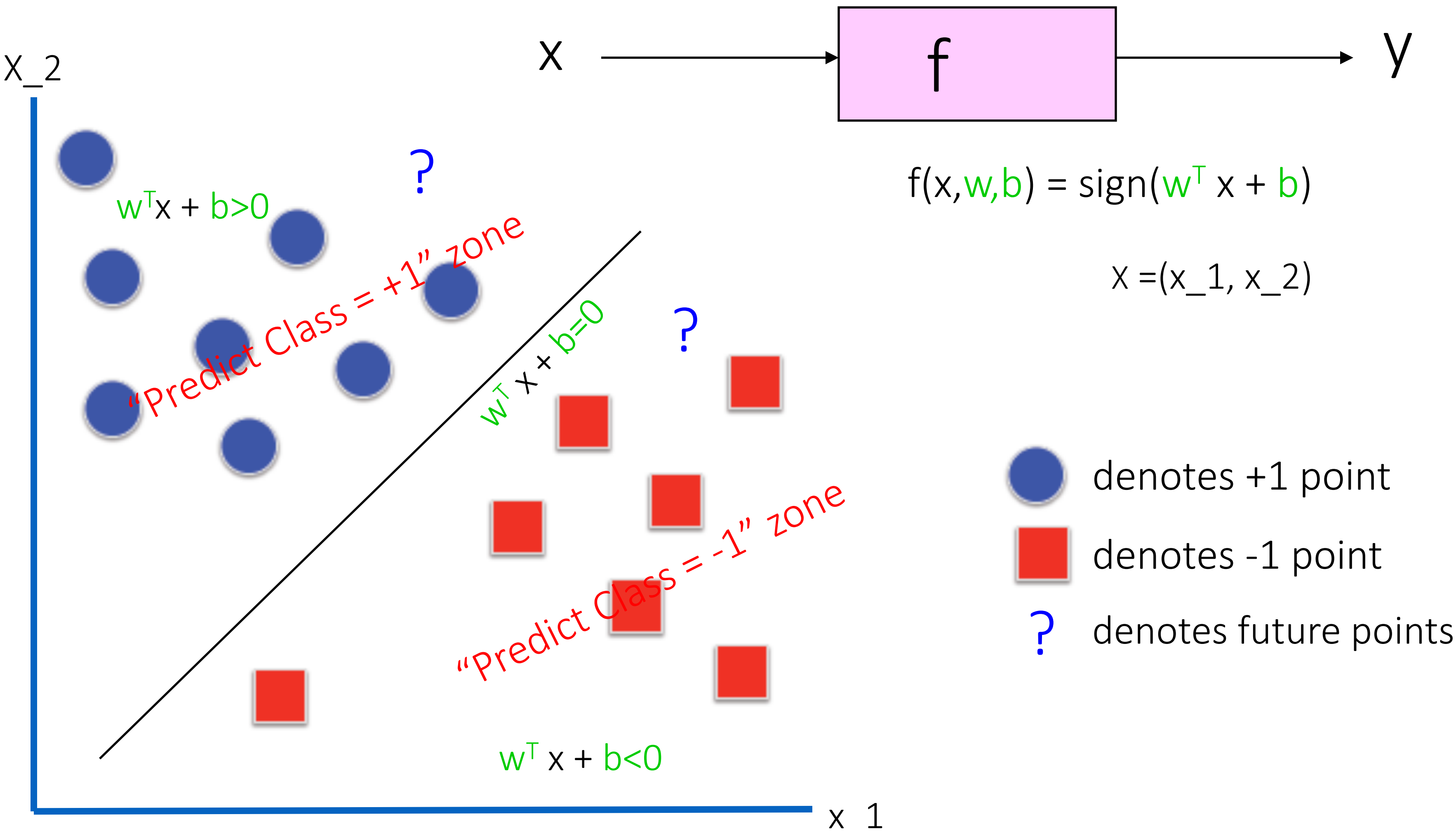
Traditional Programming



Machine Learning (training phase)

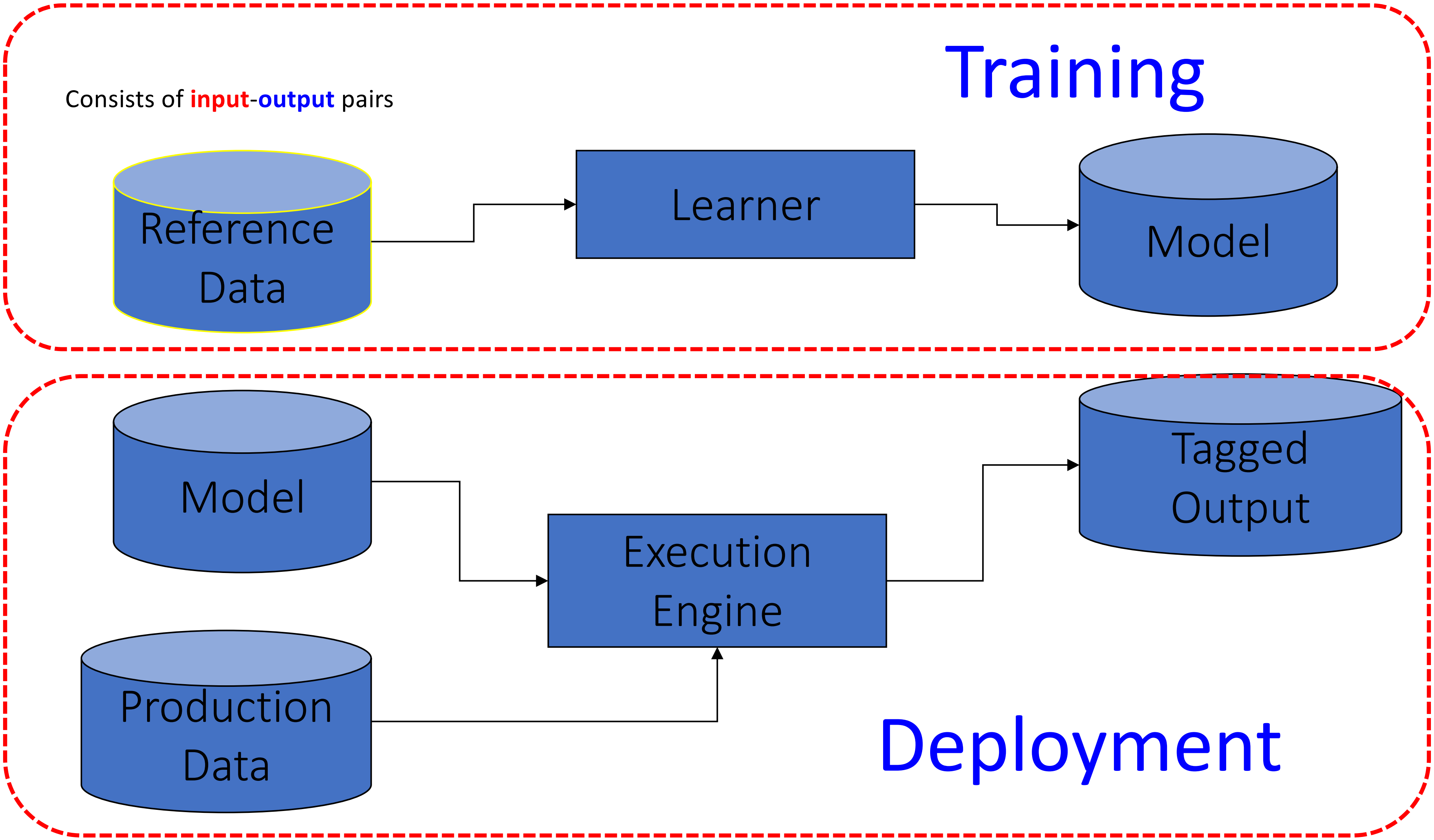


SUPERVISED Linear Binary Classifier



Courtesy slide from Prof. Andrew Moore's tutorial

Two Modes of Machine Learning



Basic Concepts

- Training (i.e. learning parameters w,b)
  - Training set includes
    - available examples $x_1, \ldots, x_L$
    - available corresponding labels $y_1, \ldots, y_L$
  - Find (w,b) by minimizing loss
    - (i.e. difference between y and f(x) on available examples in training set)

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

Basic Concepts

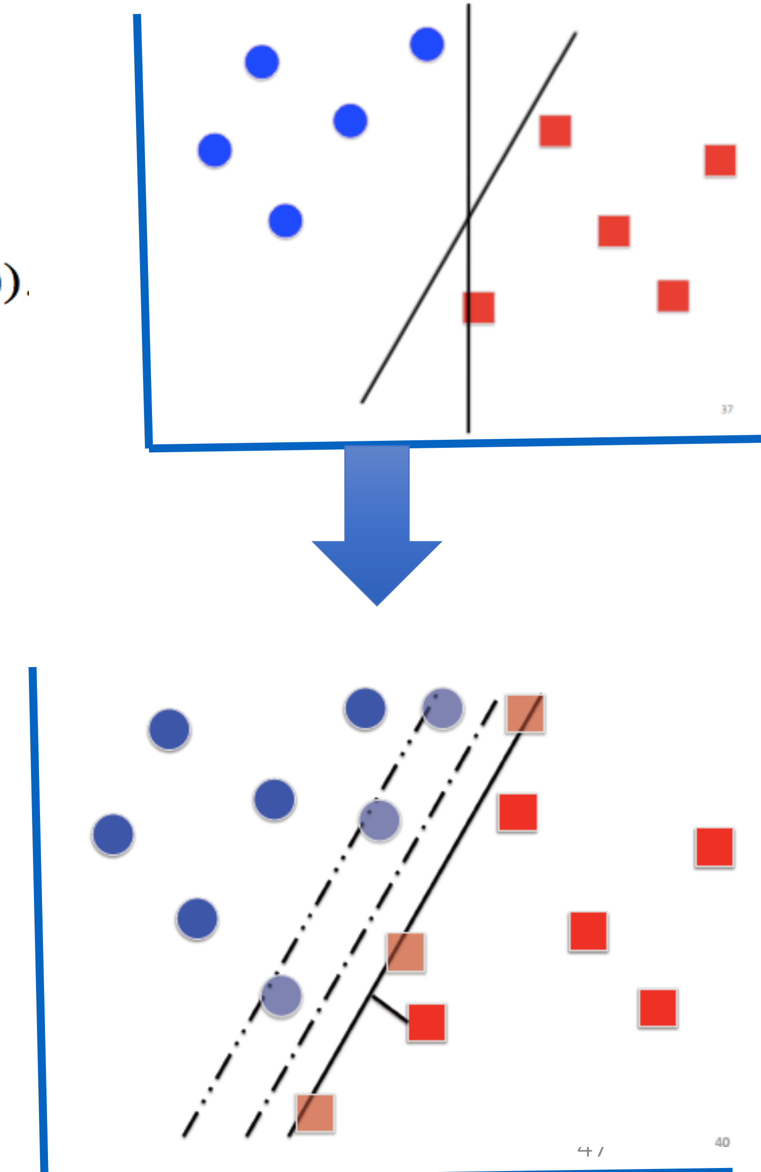
- Loss function
  - e.g. hinge loss for binary classification task

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

- Regularization
  - E.g. additional information added

on loss function to control f

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

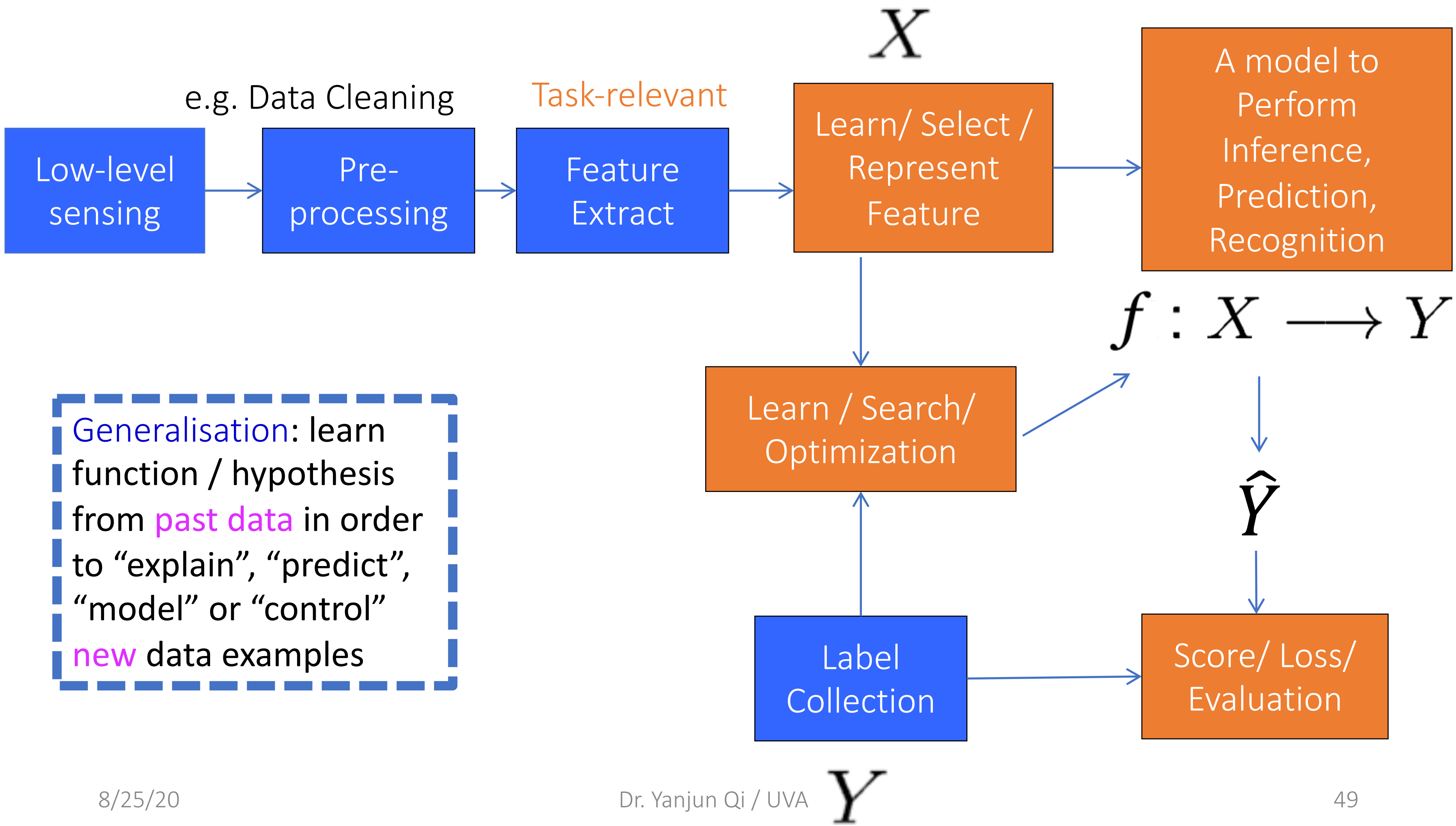


Basic Concepts

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

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

A Typical Machine Learning Application's Pipeline



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

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”.

Machine Learning in a Nutshell

Data

↓

Task

↓

Representation

↓

Score Function

↓

Search/Optimization

↓

Models, Parameters

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

Data	
Task	
Representation	
Score Function	
Search/Optimization	
Models, Parameters	

What we will cover

Data	Tabular, 1-D sequential, 2-D Grid like Imaging, 3-D VR, Graph, Set
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), …**

Machine Learning in a Nutshell

Data

Task

Representation

Score Function

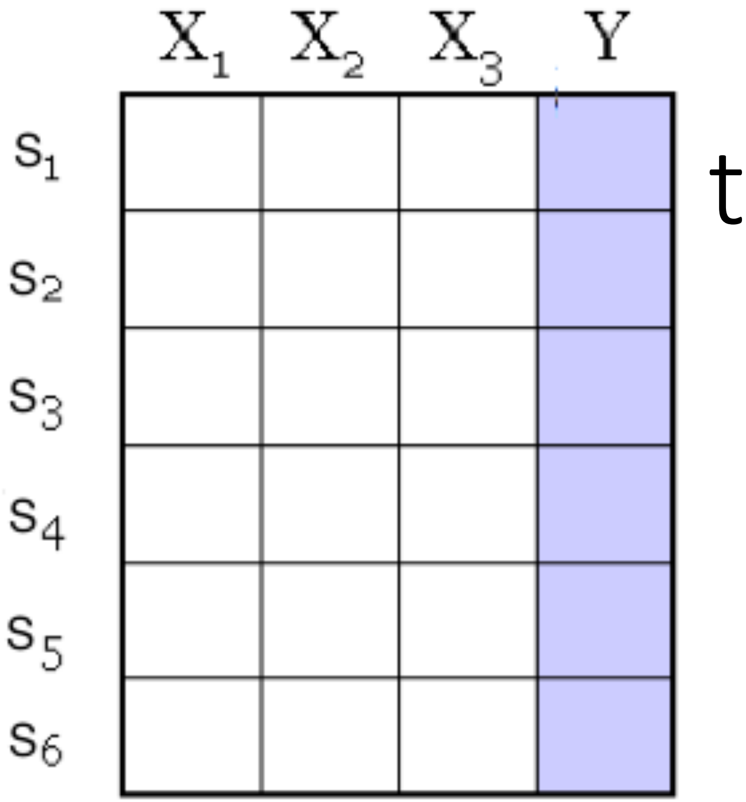
Search/Optimization

Models, Parameters

ML grew out of work in AI

Optimize a performance criterion using example data or past experience,

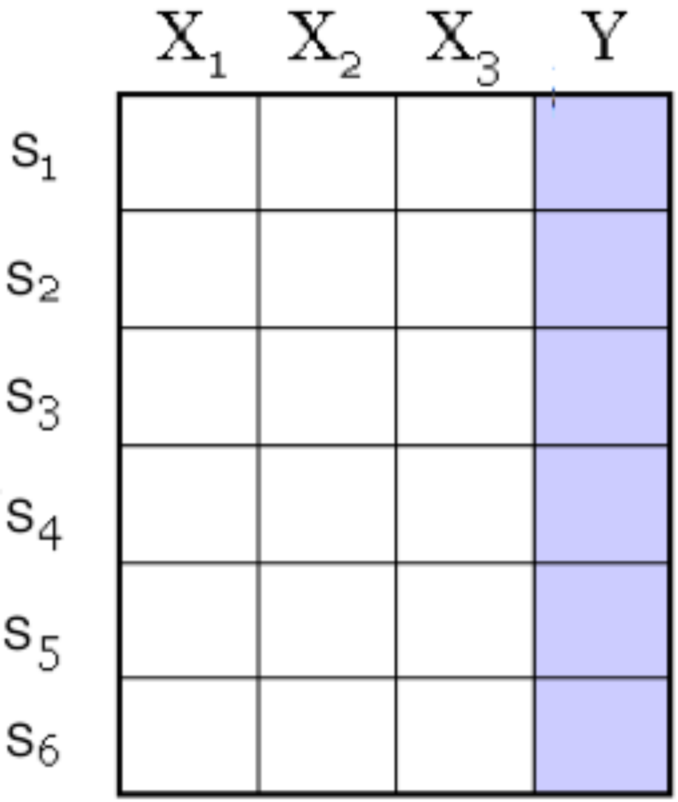
Aiming to generalize to unseen data



$$f : \boxed{X} \longrightarrow \boxed{Y}$$

- **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”

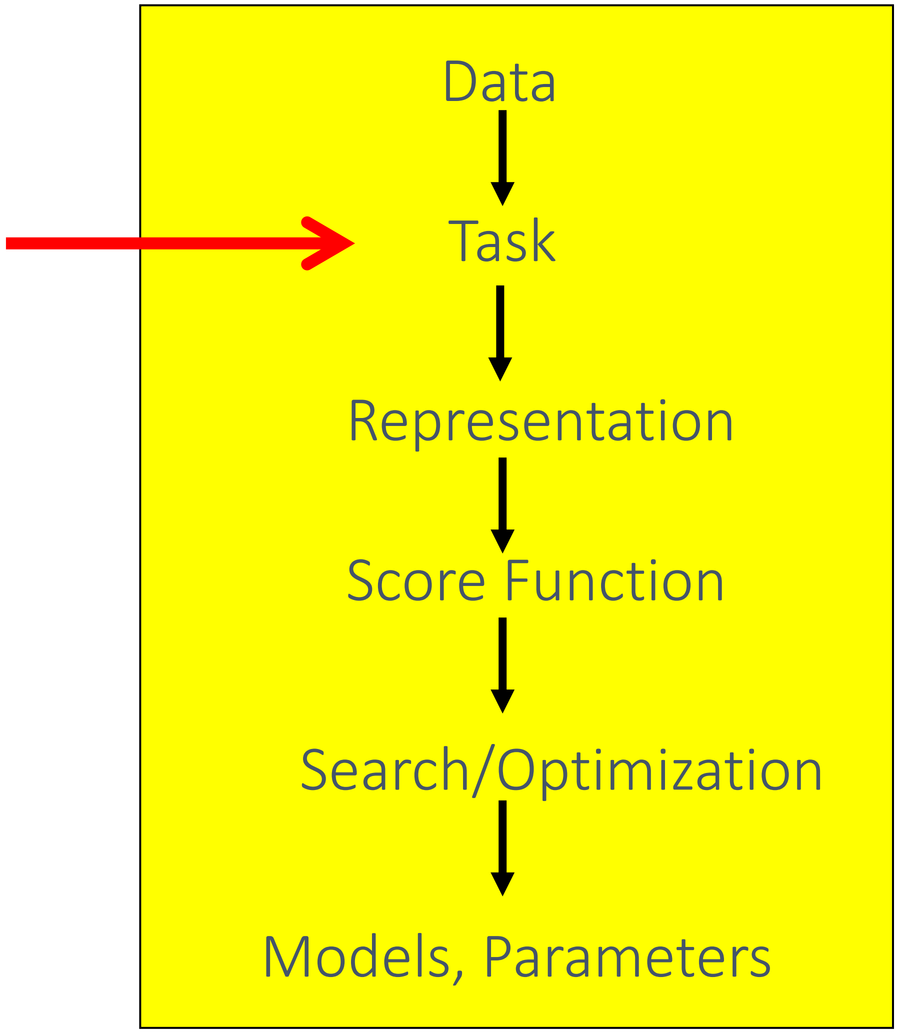
$$\mathbf{X}_{train} = \begin{bmatrix} -- & \mathbf{x}_1^T & -- \\ -- & \mathbf{x}_2^T & -- \\ \vdots & \vdots & \vdots \\ -- & \mathbf{x}_n^T & -- \end{bmatrix} \qquad \vec{y}_{train} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

training dataset

test dataset

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

Machine Learning in a Nutshell

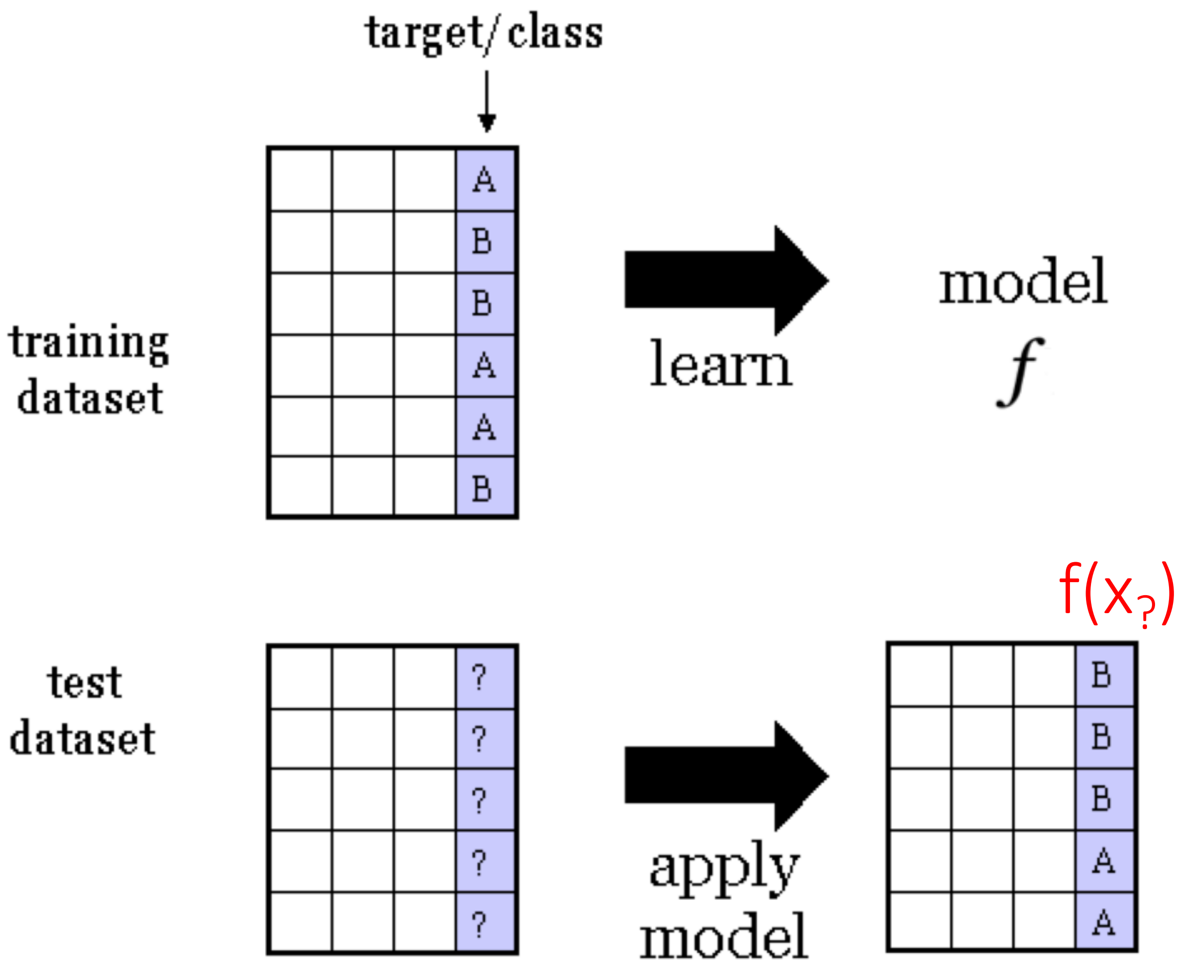


ML grew out of work in AI

Optimize a performance criterion using example data or past experience,

Aiming to generalize to unseen data

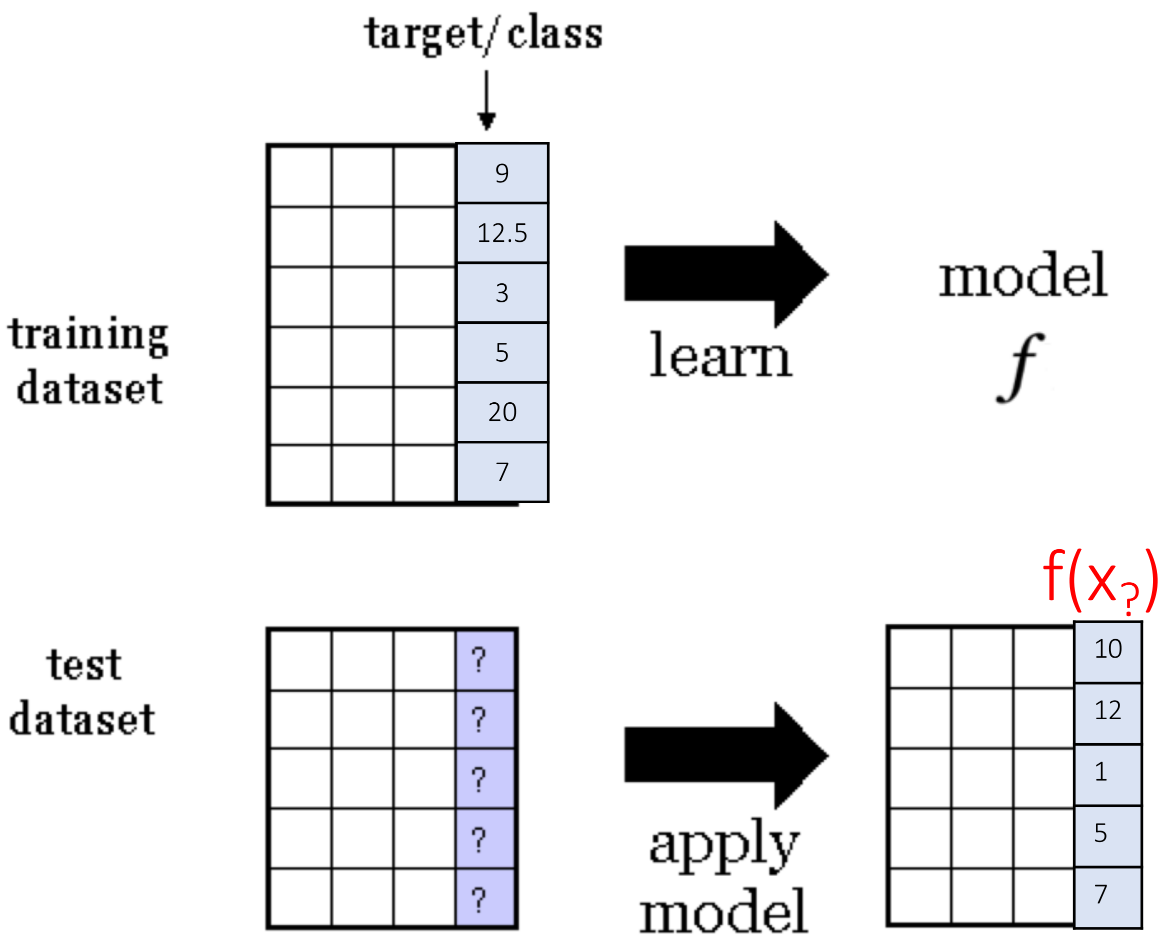
e.g. SUPERVISED Classification



Training dataset consists of input-output pairs

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

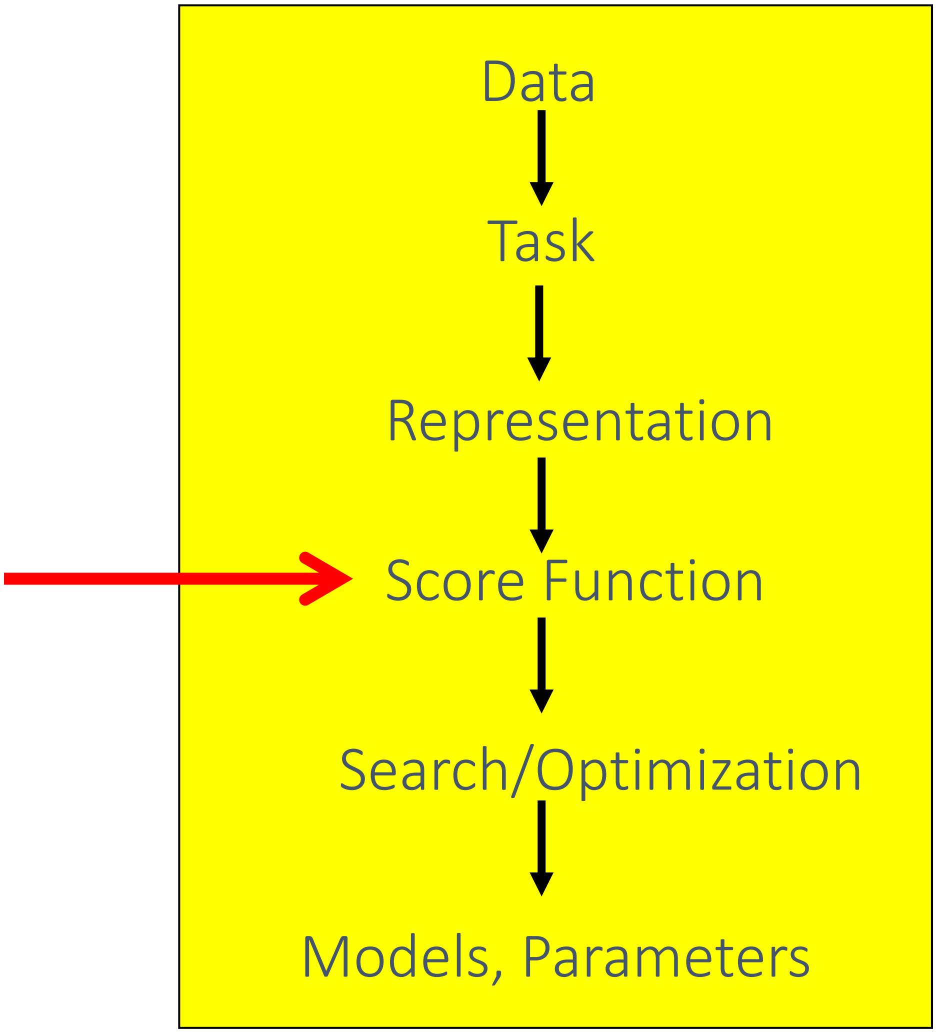
e.g. SUPERVISED Regression



Training dataset consists of input-output pairs

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

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

Basic Concepts

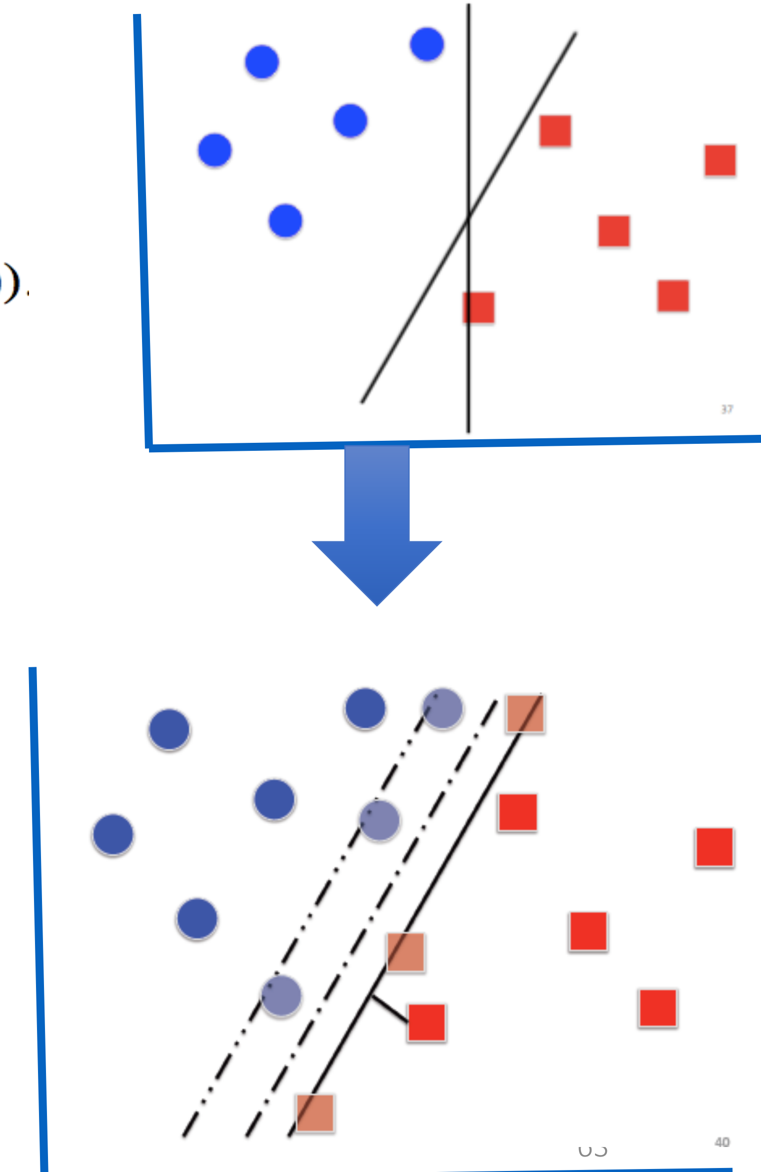
- Training (i.e. learning parameters w,b)
  - Training set includes
    - available examples $x_1,\ldots,x_L$
    - available corresponding labels $y_1,\ldots,y_L$
  - Find (w,b) by minimizing loss
    - (i.e. difference between y and f(x) on available examples in training set)

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

Basic Concepts

- Loss function
  - e.g. hinge loss for binary classification task

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



- Regularization
  - E.g. additional information added on loss function to control f

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

Machine Learning in a Nutshell

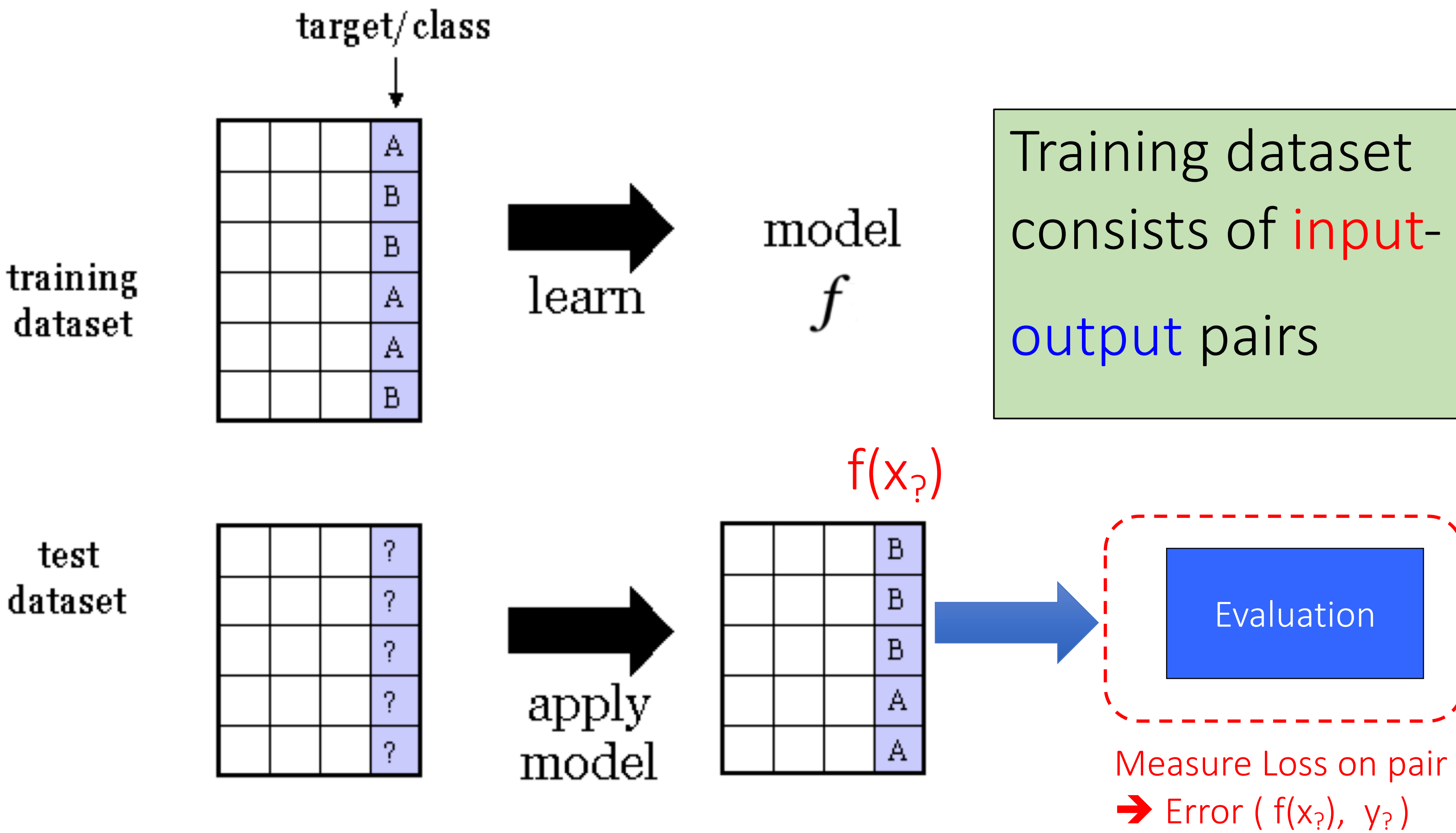
Data
↓
Task
↓
Representation
↓
Score Function
↓
Search/Optimization
↓
Models, Parameter, Metric

ML grew out of work in AI

Optimize a performance criterion using example data or past experience,

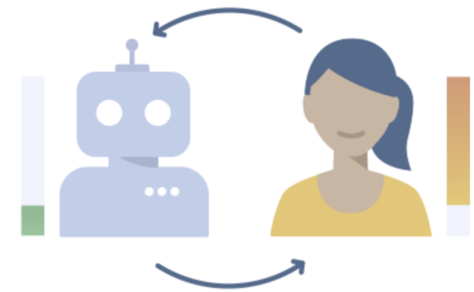
Aiming to generalize to unseen data

How to know the program works well?



Training dataset consists of input-output pairs

ML and AI Research @ UVA CS



Human Centric
Machine Intelligence

Understand Human

Be Communicative

Model Human Intent

Be Explainable

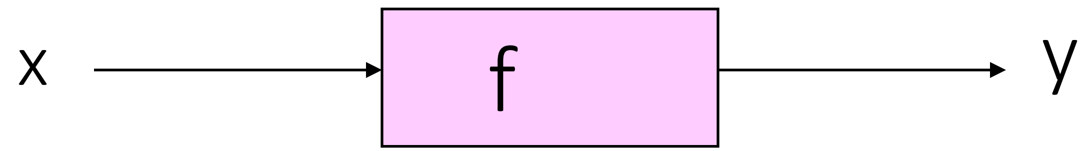
Interact with Human

Efficient & Fast

Theoretically Sound

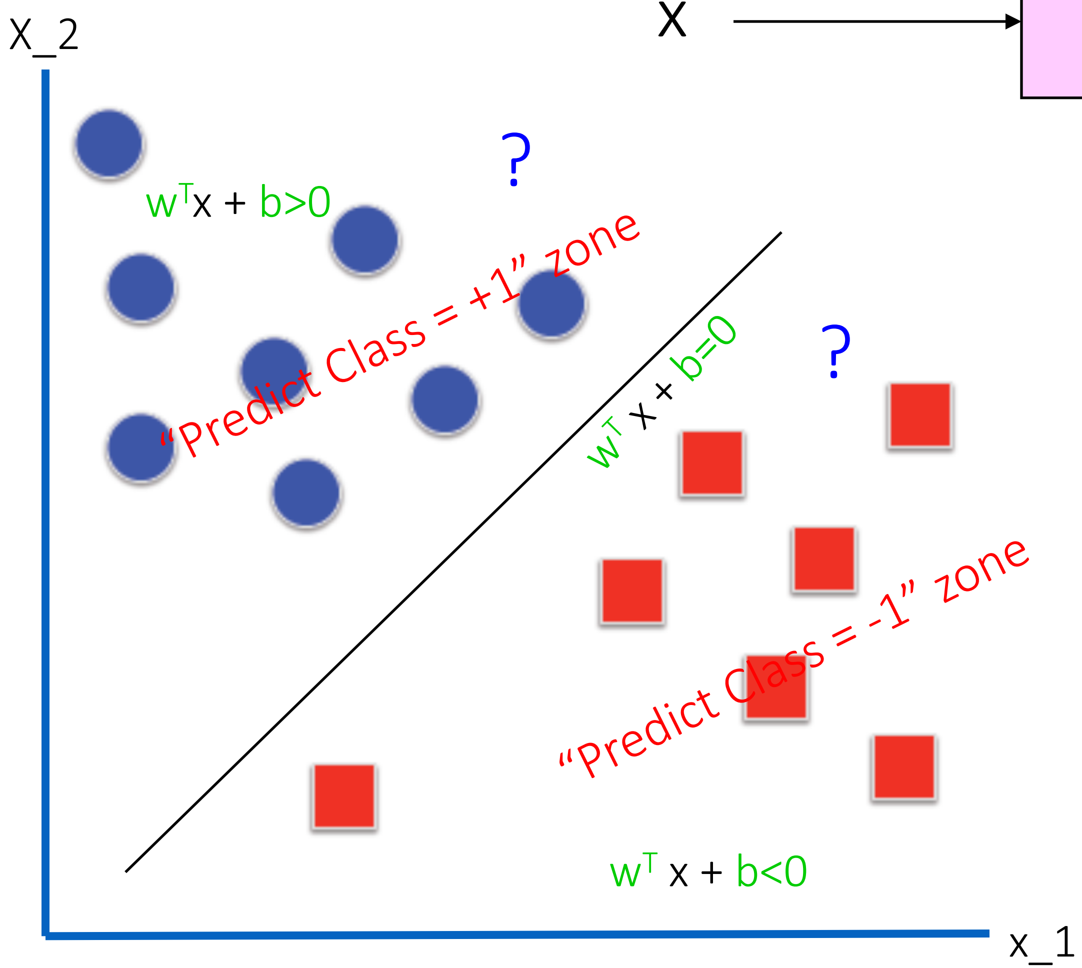
Trustworthy

SUPERVISED Linear Binary Classifier



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

$X = (x\_1, x\_2)$



denotes +1 point

denotes -1 point

? denotes future points

Courtesy slide from Prof. Andrew Moore's tutorial

Nutshell for the simple Linear **Supervised Classifier**

Data
↓
Task
↓
Representation
↓
Score Function
↓
Search/Optimization
↓
Models, Parameters, Metrics

Tabular (R^2)
↓
Classification
Output Y is categorical
↓
Linear Decision Boundary
↓
?
↓
?
↓
Many metric to measure classification on future points


Thank You

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