UVA CS 4774: Machine Learning

Lecture 2: Machine Learning in a Nutshell

Dr. Yanjun Qi

University of Virginia
Department of Computer Science

Roadmap

- Machine Learning in a Nutshell

- Examples of Different Data Types
- Examples of Different Tasks

- Examples of Different Representation Types
- Examples of Different Loss/Cost Types
- Examples of Different Model Properties

Machine Learning in a Nutshell

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

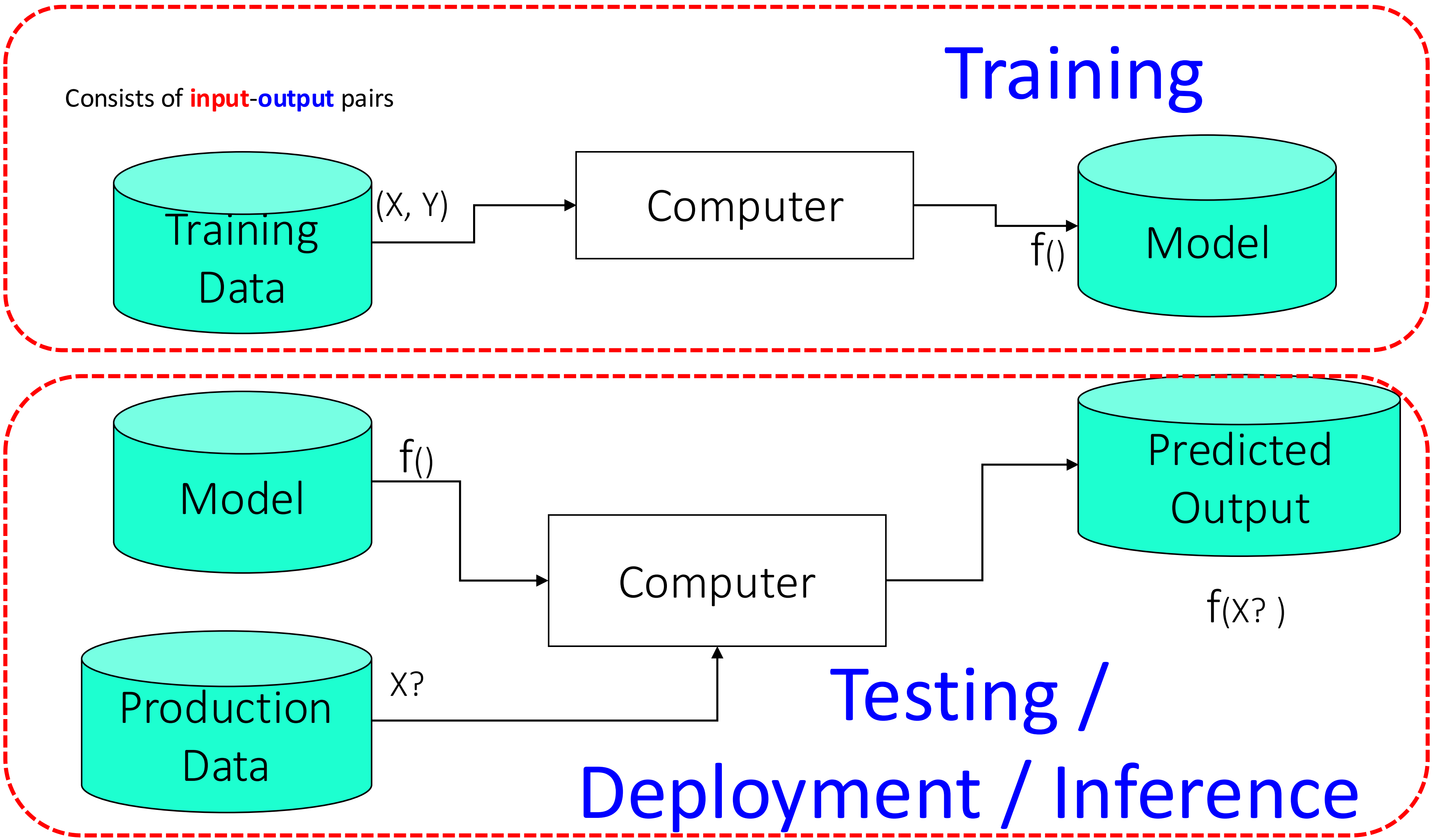
Hyperparameter, Metrics

ML grew out of work in AI

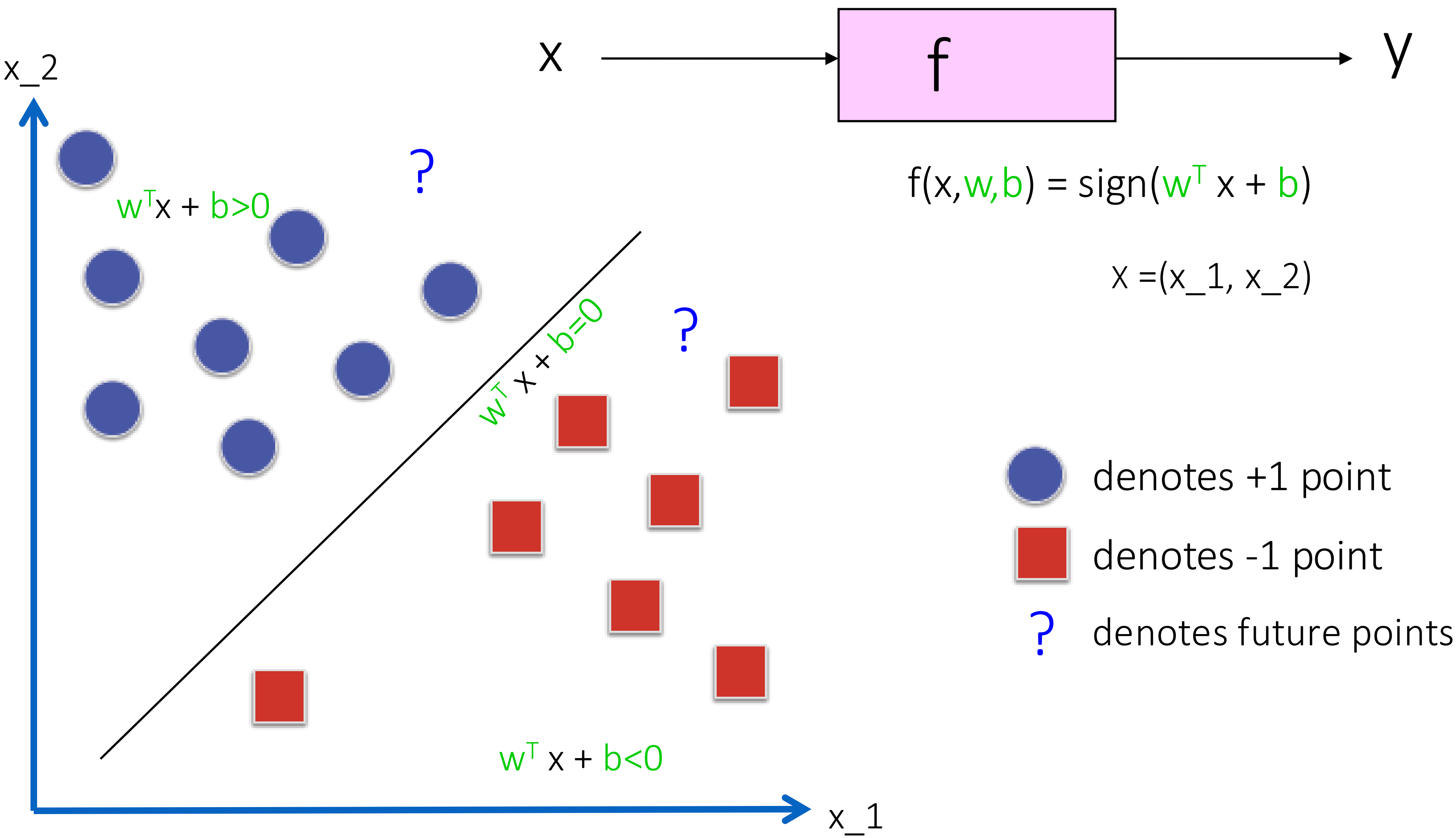
Optimize a performance criterion using example data or past experience,

Aiming to generalize to unseen data

Two Phases (Modes) of Machine Learning

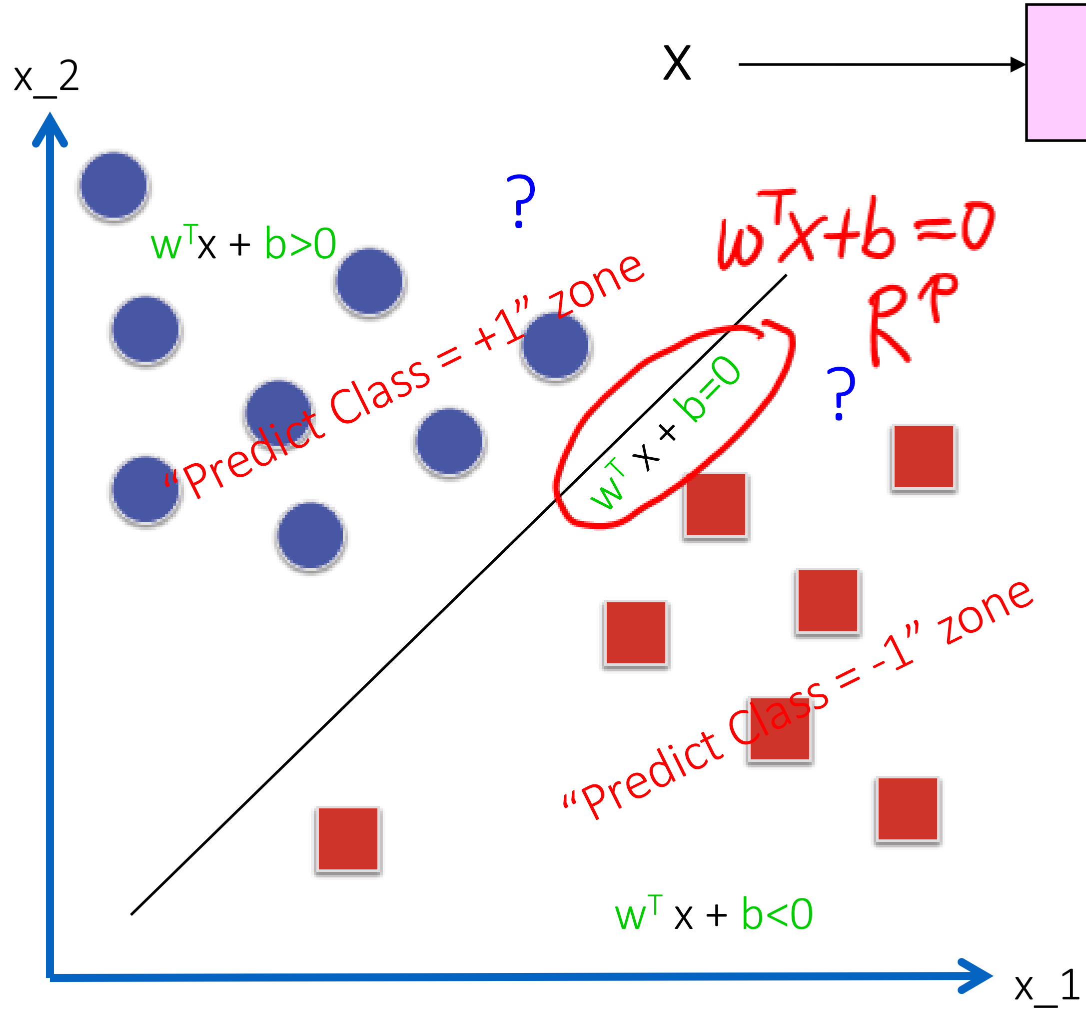


SUPERVISED Linear Binary Classifier
: Binary y / Linear f / X as R^2



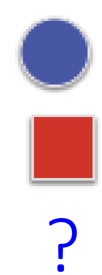
Courtesy slide from Prof. Andrew Moore's tutorial

SUPERVISED Linear Binary Classifier
: Binary y / Linear f / X as R^2



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

X = (x_1, x_2) R^2



Courtesy slide from Prof. Andrew Moore's tutorial

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 / Cost function L()
  - (i.e. difference between y and f(x) on available examples in training set)

ℓ

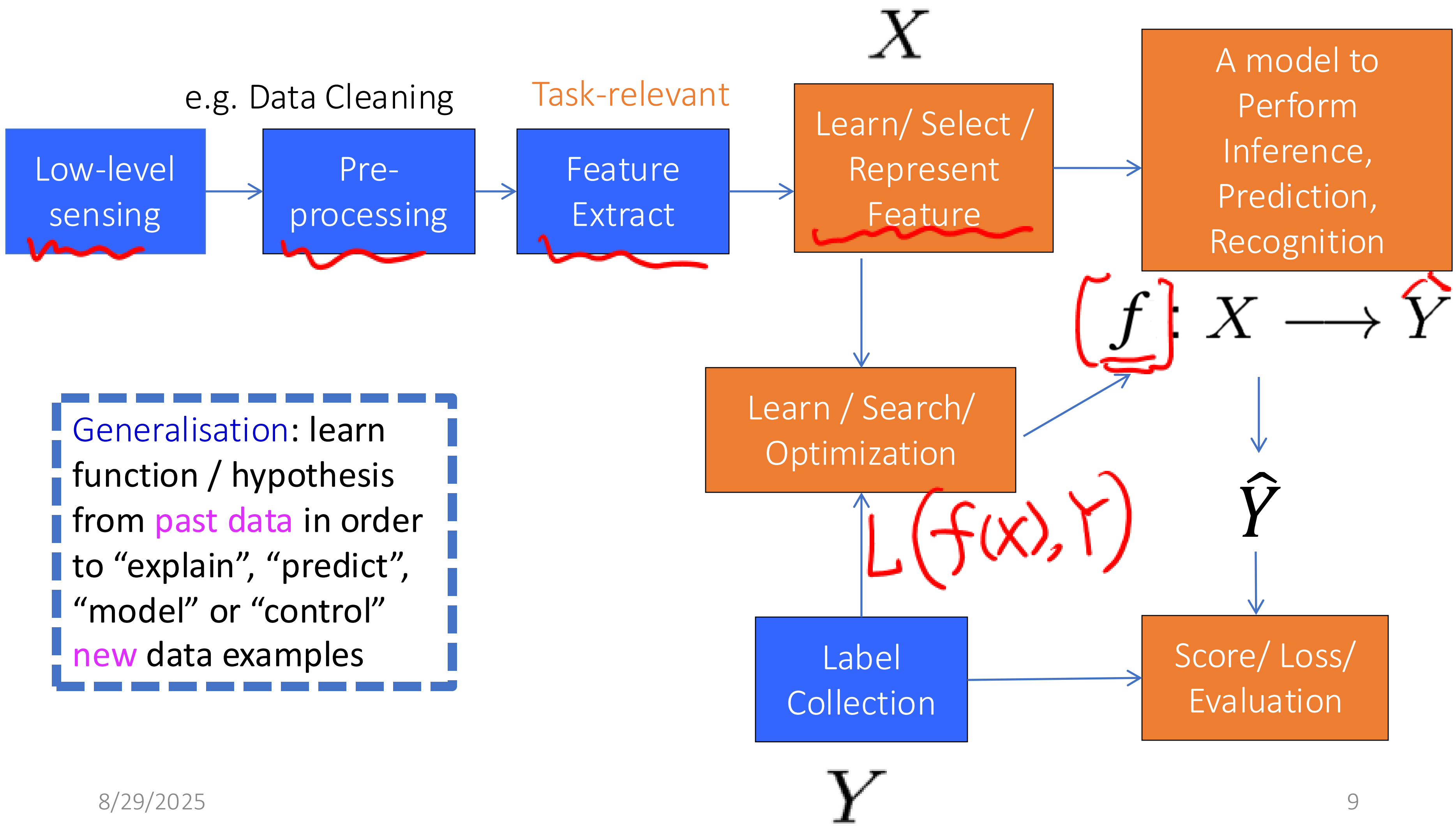
$$(W, b) = \underset{W,\, b}{\operatorname{argmin}} \underbrace{\sum_{i=1}^{N} \ell(f(x_i), y_i)}$$

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

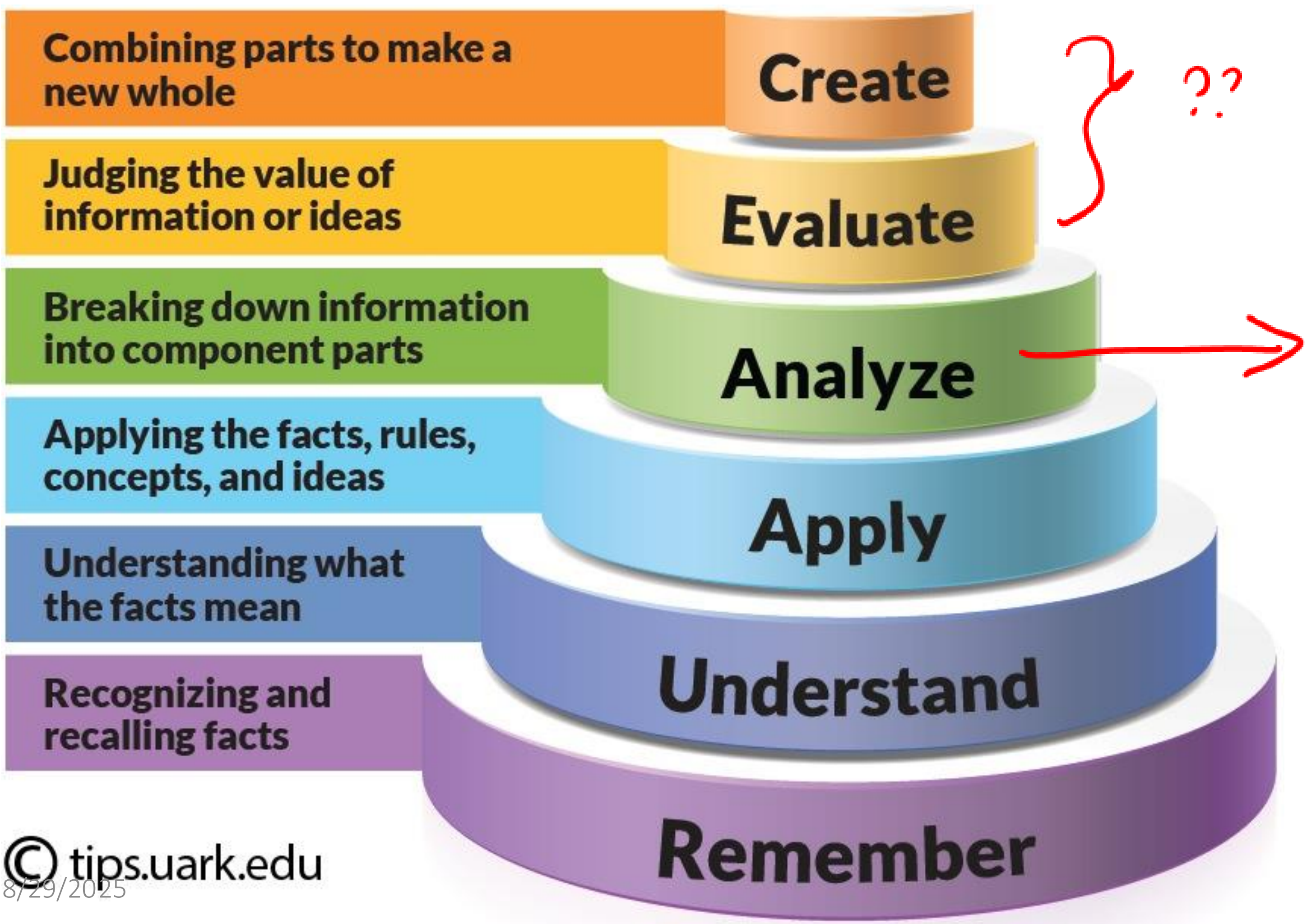
Hyperparameter, Metrics

ML grew out of work in AI

Optimize a performance criterion using example data or past experience,

Aiming to generalize to unseen data

My Teaching Guide: Bloom's Taxonomy on Cognitive Learning



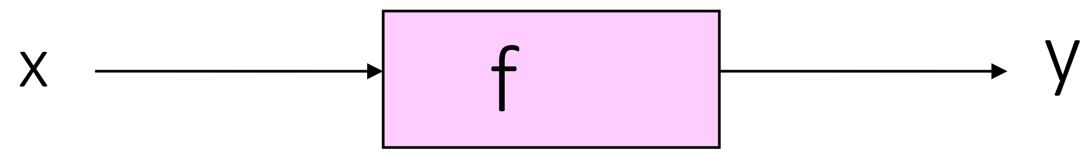
What we have covered

Data	
Task	
Representation	
Score Function	
Search/Optimization	
Models, Parameters	

What we will cover

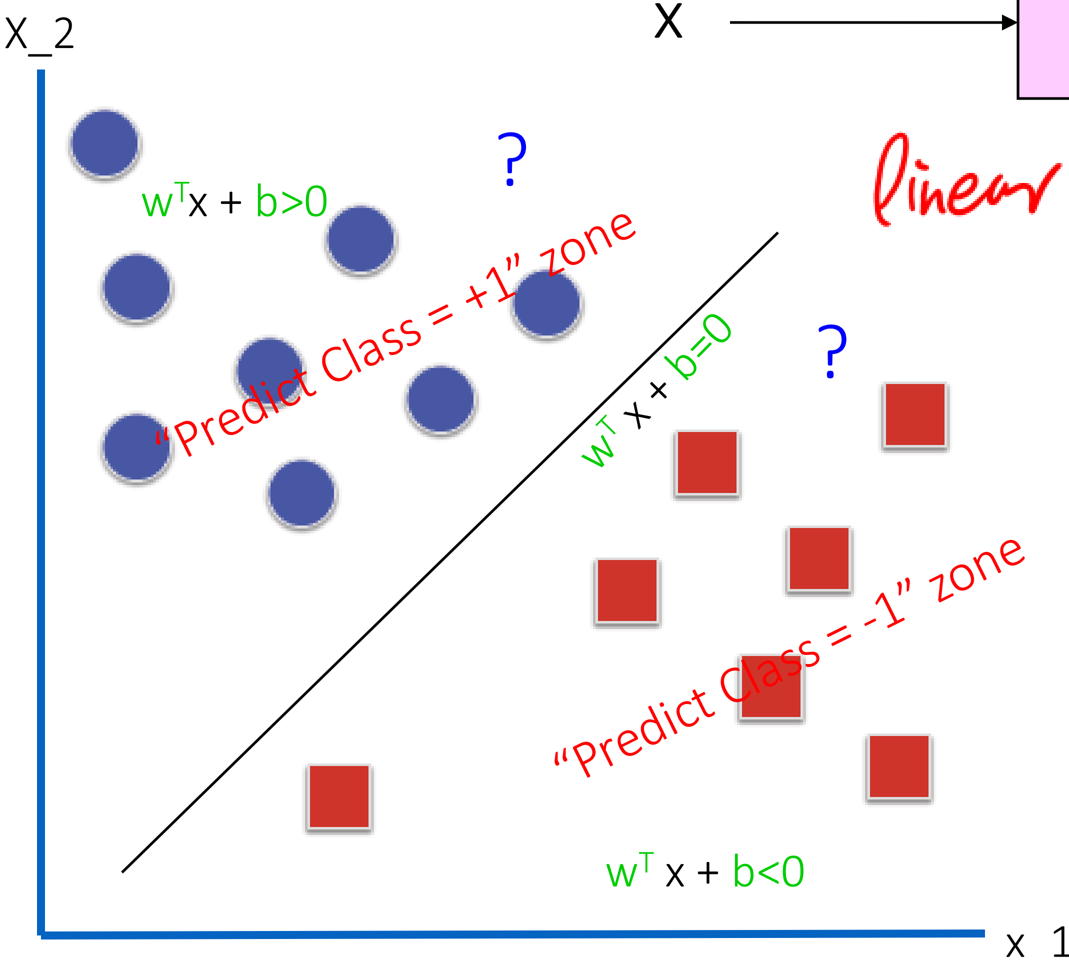
Data	Tabular, 1-D sequential, 2-D Grid like Imaging, 3-D VR, Graph, Set
Task	Regression, classification, clustering, dimen-reduction
Representation	Linear func, nonlinear function (e.g. polynomial expansion), local linear, logistic function (e.g. p(c|x)), tree**, multi-layer, prob-density family (e.g. 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), …**

SUPERVISED Linear Binary Classifier



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

R^2 $x = (x\_1, x\_2)$



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 binary

Linear Decision Boundary

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

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

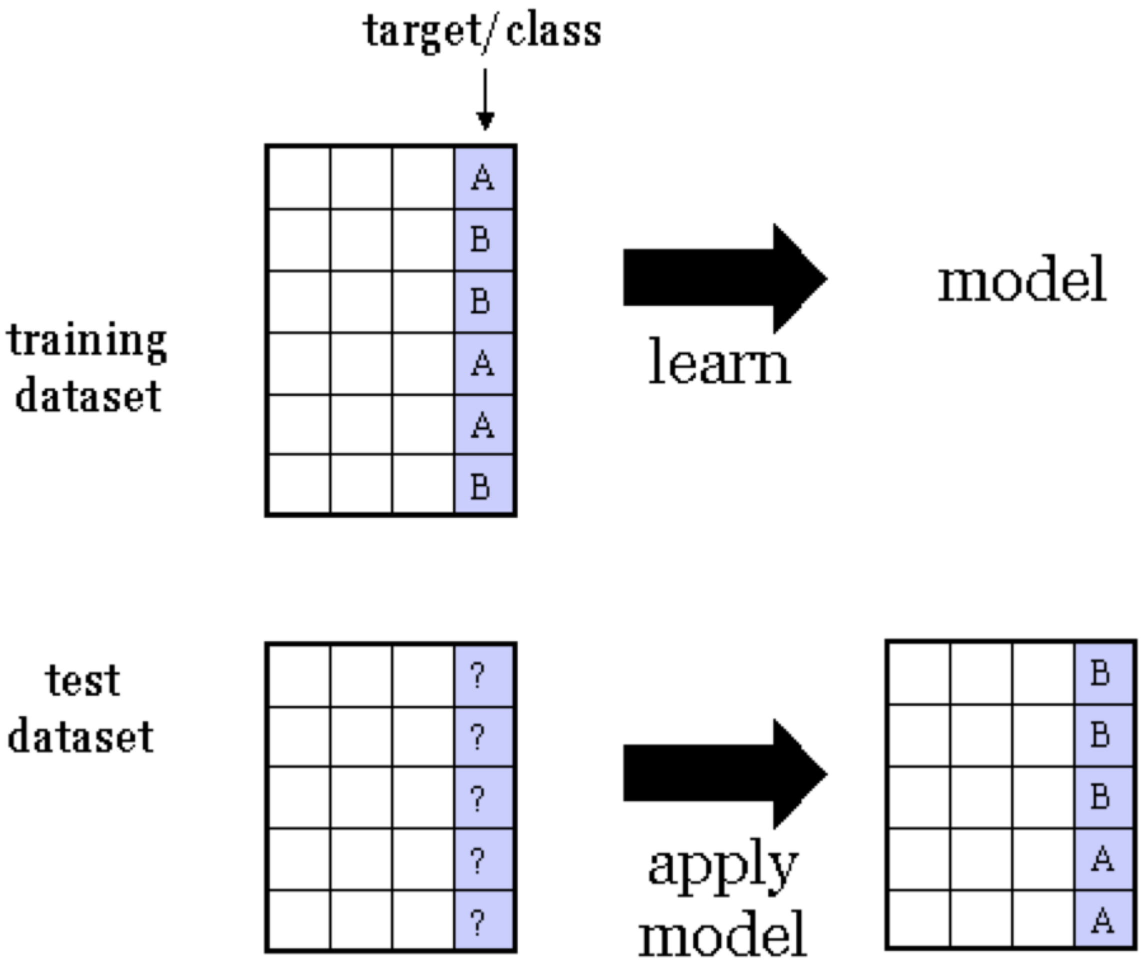
?

e.g., Classification Error on
future points (Testing Data)

https://scikit-learn.org/stable/tutorial/basic/tutorial.html

https://colab.research.google.com/drive/1oEGNhQ55iBNElYqfZpueSE2l_g3tQxSD?usp=sharing

I will code-run through: Recognizing hand-written digits with L2.ipynb
Adapted from: ScikitLearn Tutorial plot_digits_classification.ipynb



Training dataset consists of input-output pairs


Thank You

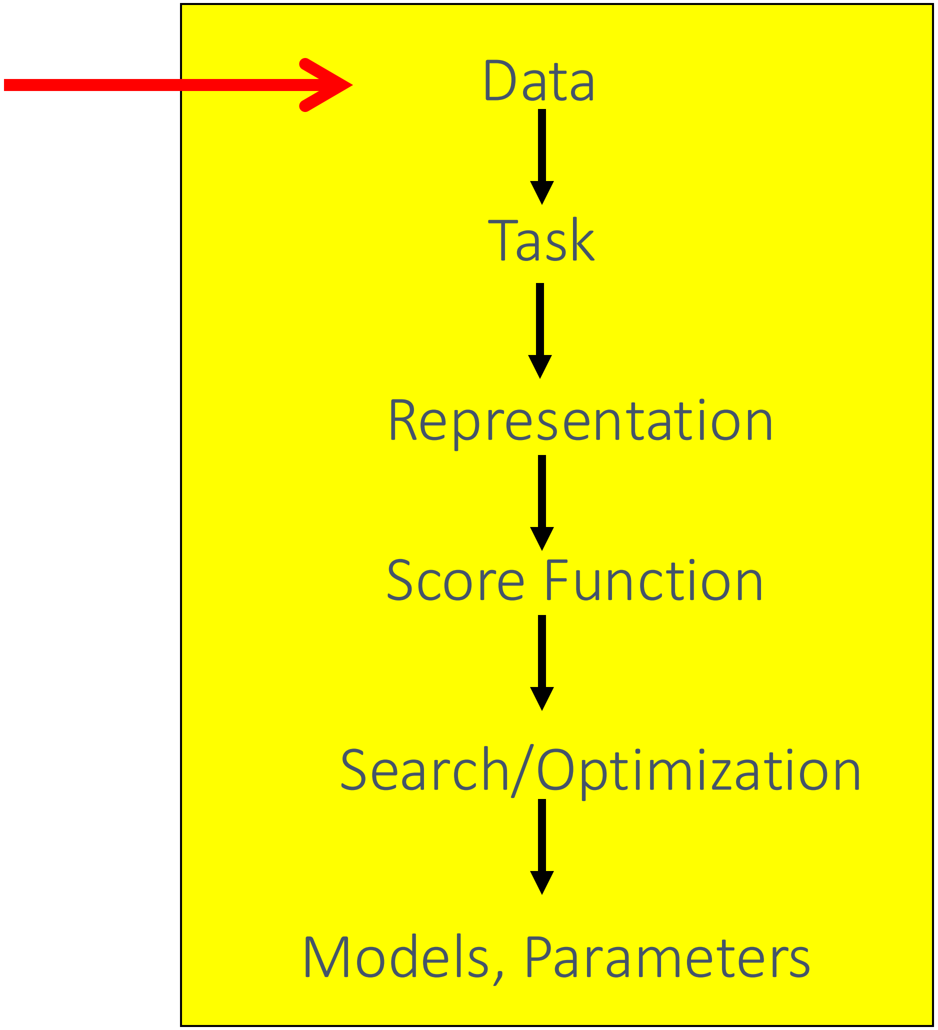
Roadmap

- Machine Learning in a Nutshell

- Examples of Different Data Types
- Examples of Different Tasks

- Examples of Different Representation Types
- Examples of Different Loss/Cost Types
- Examples of Different Model Properties

Machine Learning in a Nutshell

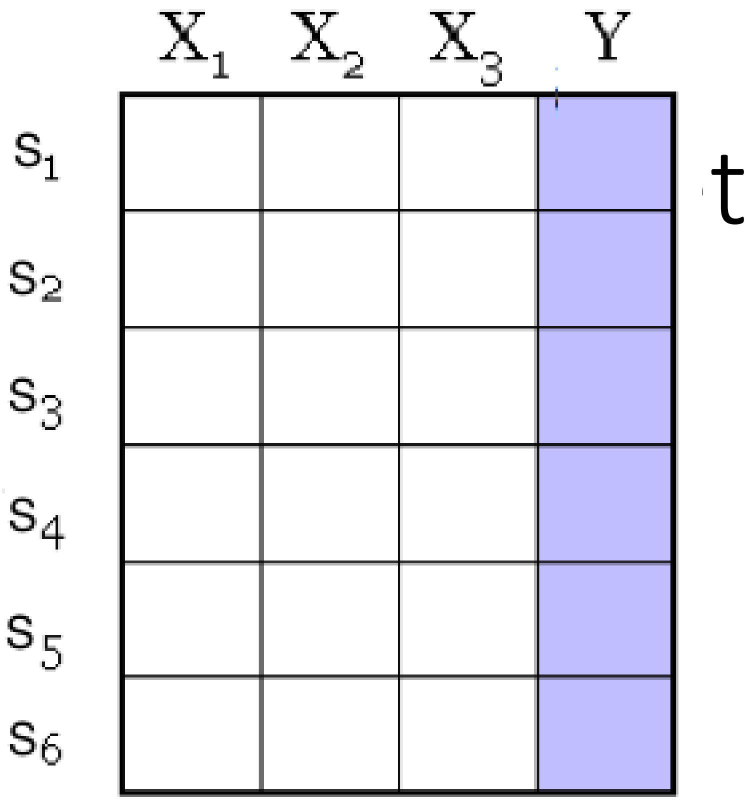


ML grew out of work in AI

Optimize a performance criterion using example data or past experience,

Aiming to generalize to unseen data

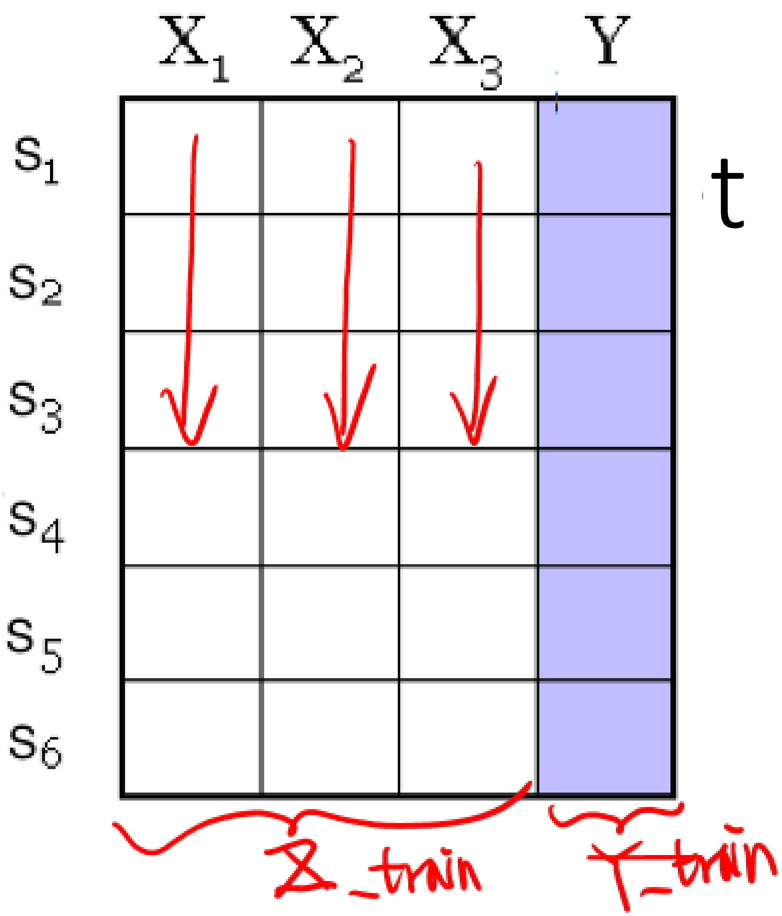
Tabular Data Type



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

- **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]

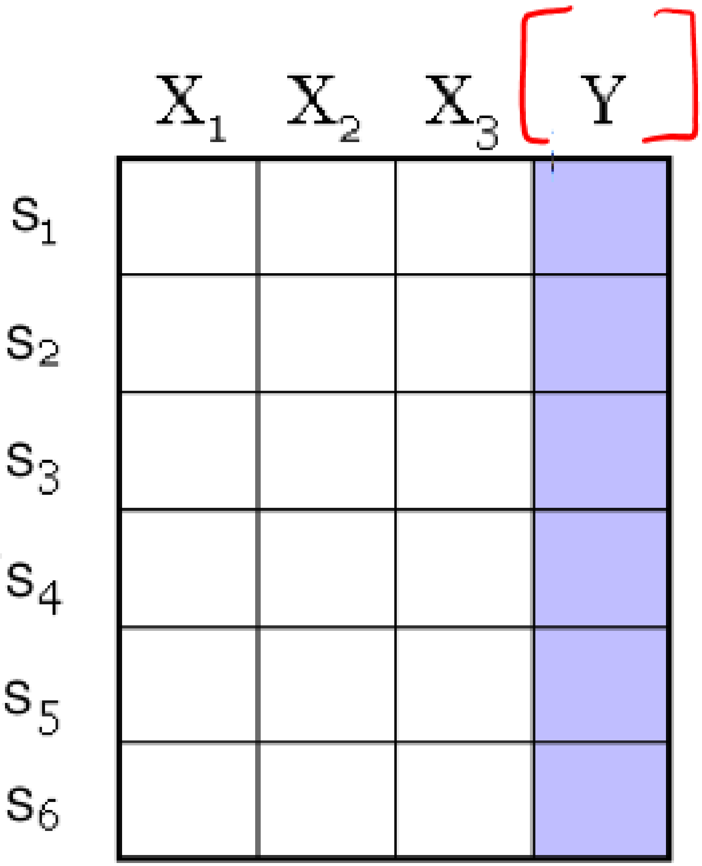
Tabular Data Type



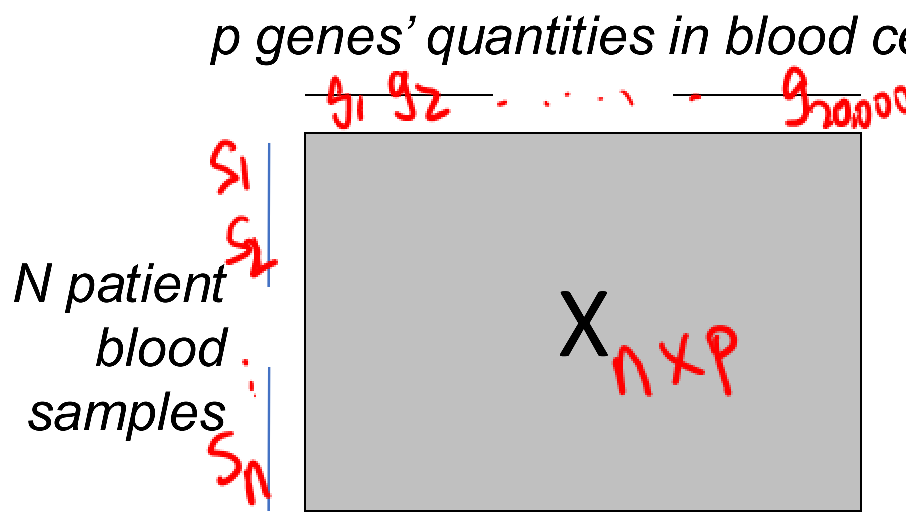
$$f : \boxed{X} \longrightarrow \boxed{Y}$$

- **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"



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

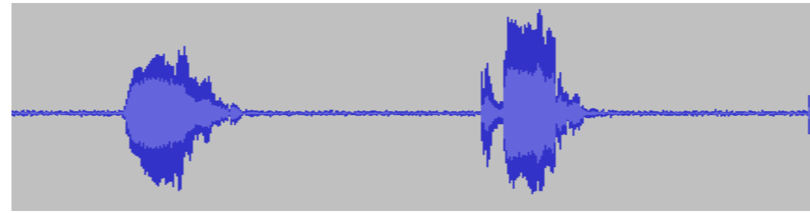
training dataset

test dataset

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

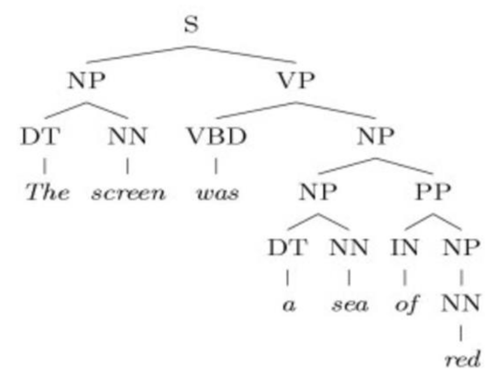
1D Sequence Data Type (eg. Language, Genome, Audio)

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 …

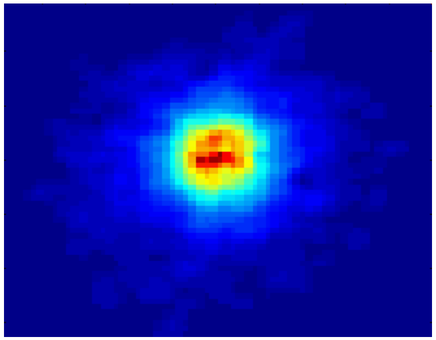


The screen was a sea of red

Language Parsing



2D Grid Data Type (eg. Images)

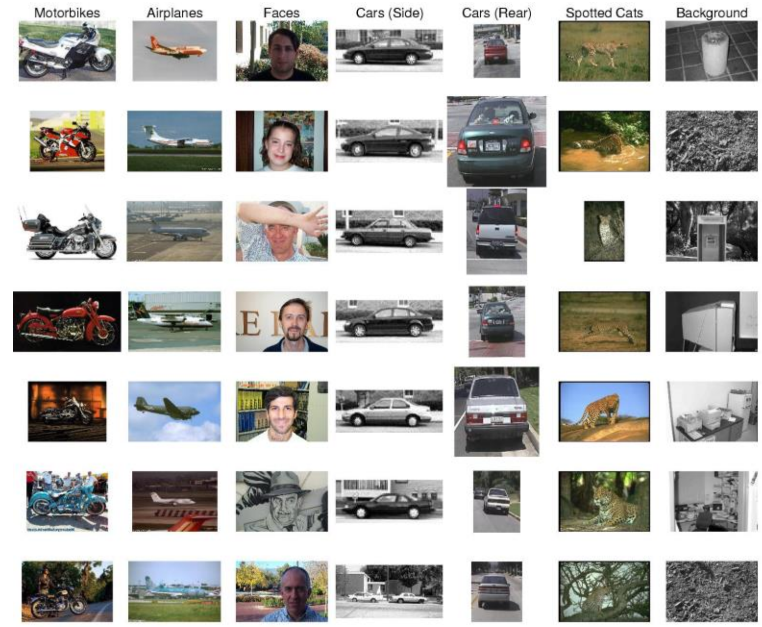


e.g.,

- 72 million stars, 20 million galaxies
- Object Catalog: 9 GB
- Image Database: 150 GB

Figure S6 Illustrative Examples of Chest X-Rays in Patients with Pneumonia,

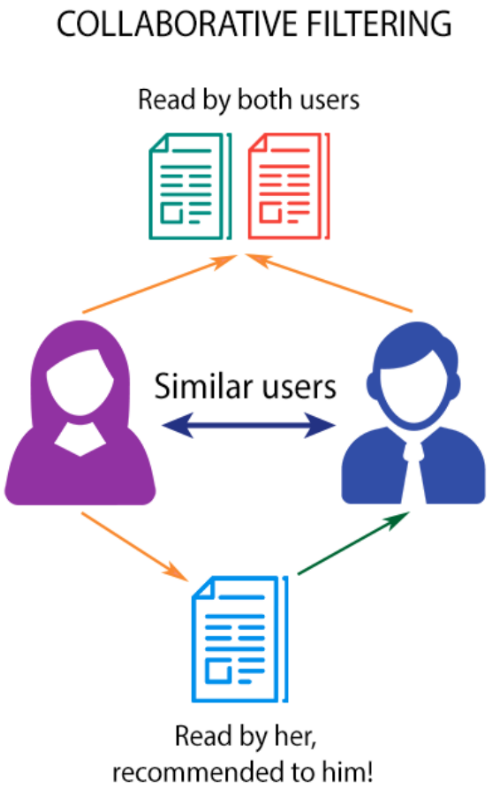
Kaggle: 5,232 chest X-ray images from children, including 3,883 characterized as depicting pneumonia (2,538 bacterial and 1,345 viral) and 1,349 normal, from a total of 5,856 patients



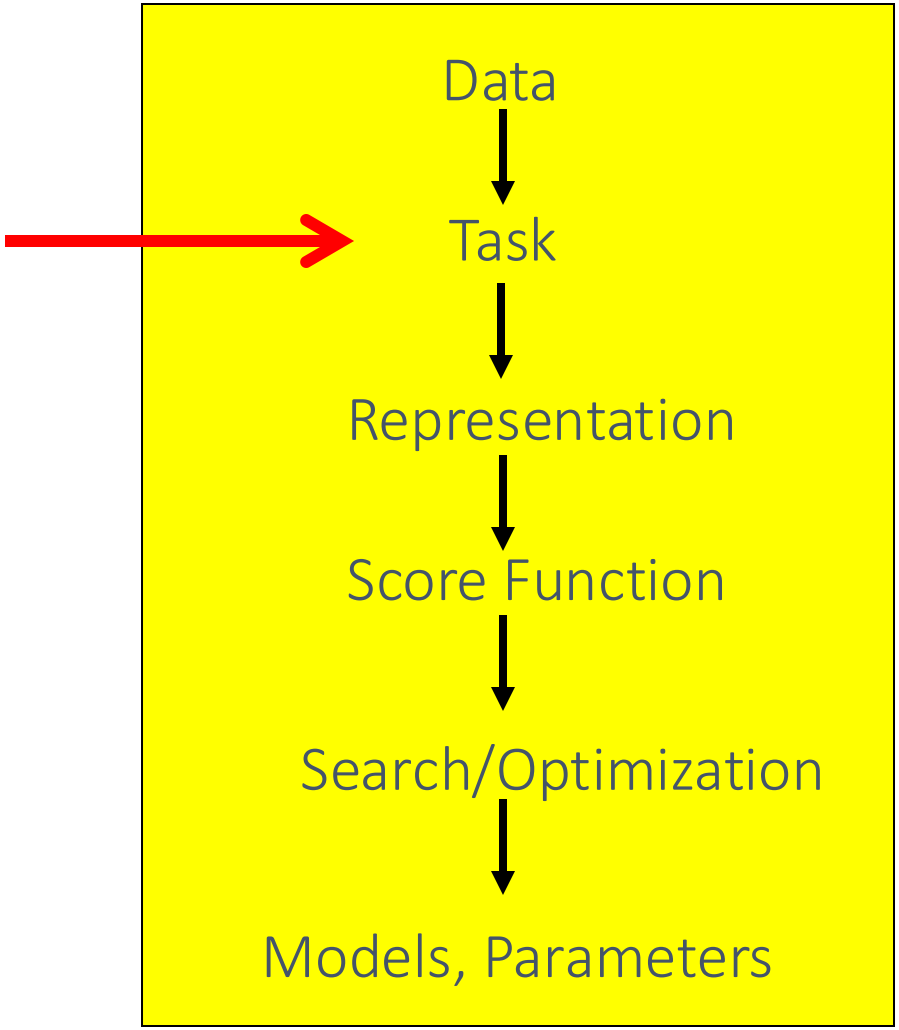
ImageNet Competition:

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

Graph Data Type (eg. Social Network)



Machine Learning in a Nutshell

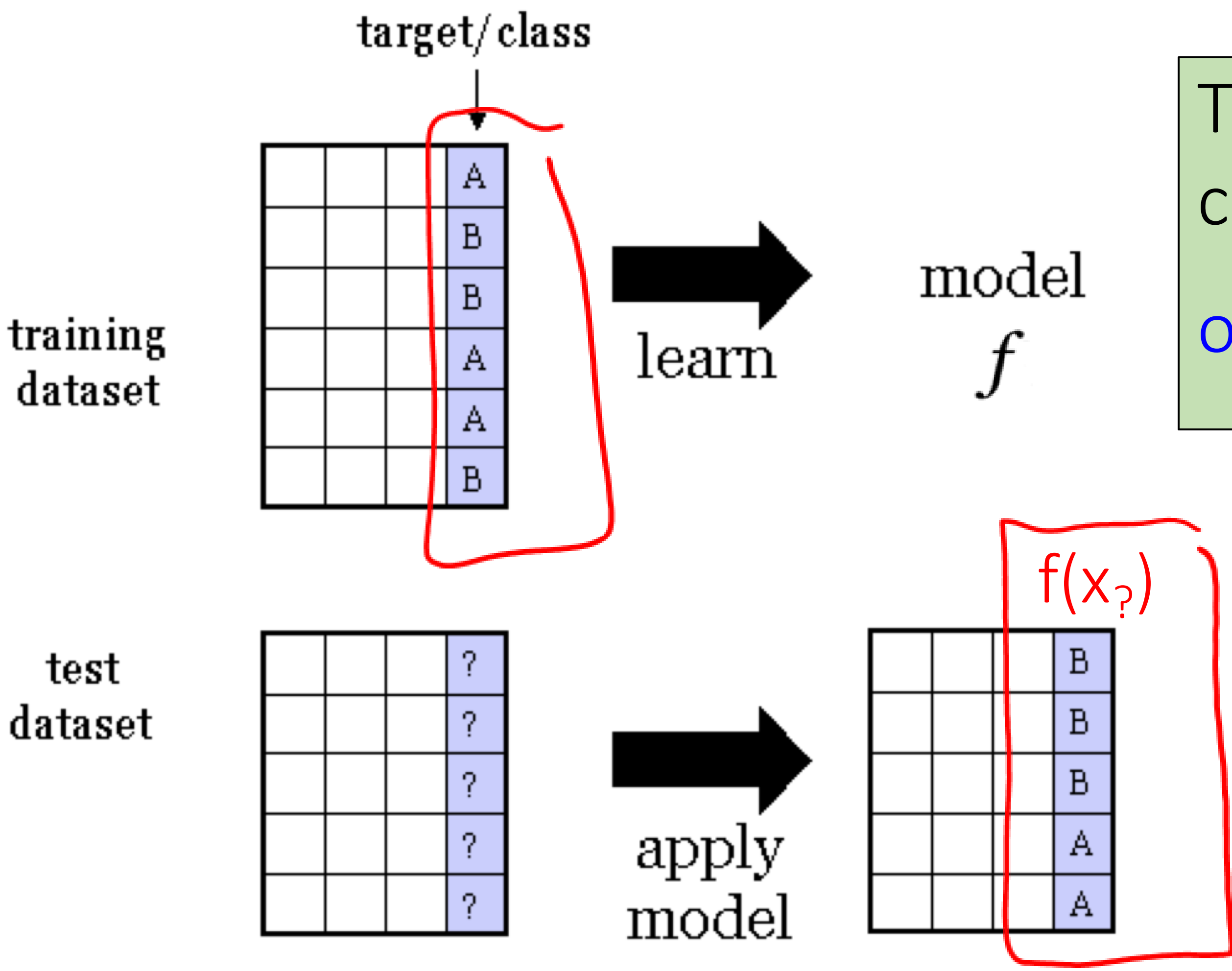


ML grew out of work in AI

Optimize a performance criterion using example data or past experience,

Aiming to generalize to unseen data

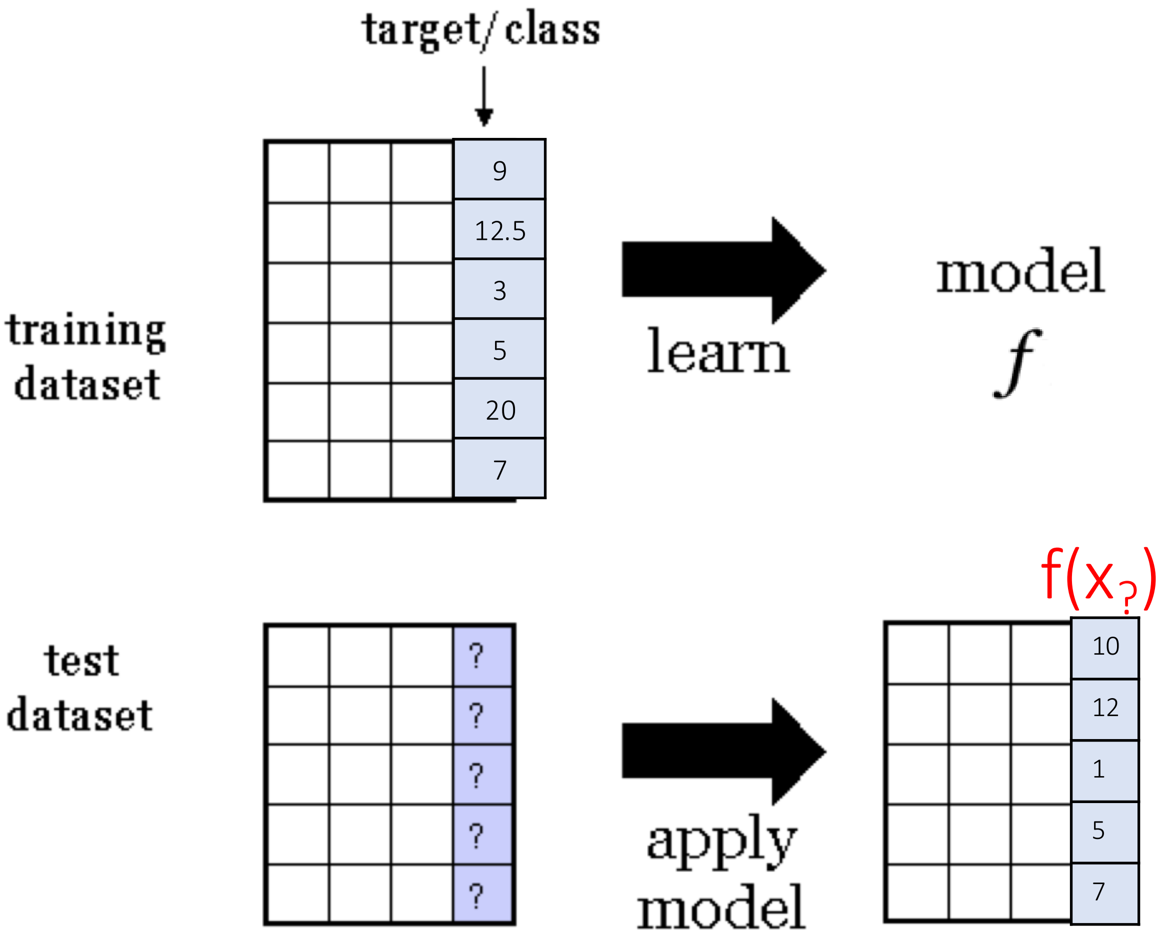
e.g. SUPERVISED Classification



Training dataset consists of input-output pairs

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

e.g. SUPERVISED Regression

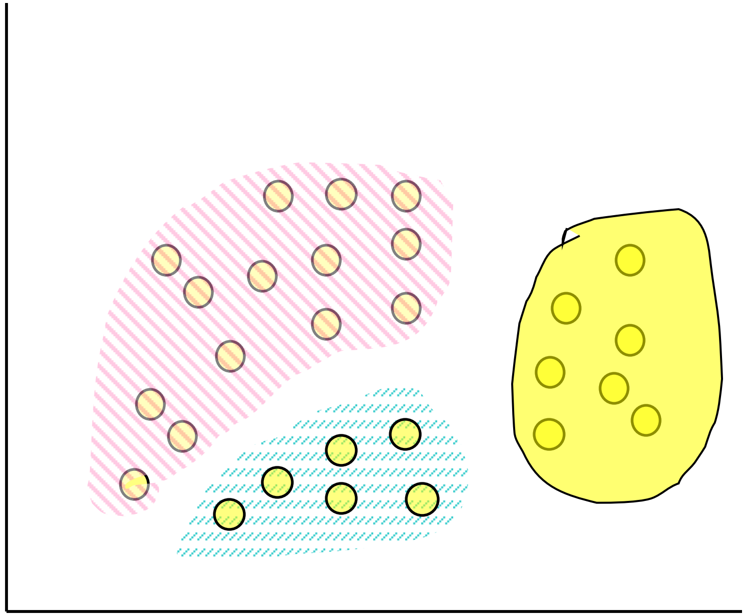


Training dataset consists of input-output pairs

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

Unsupervised LEARNING : [No Given Y]

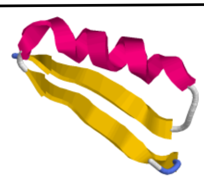
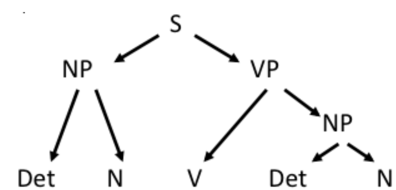
- No labels are provided (e.g. No Y provided)
- Find patterns from unlabeled data, e.g. clustering



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

Structured Output LEARNING : [Complex 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	 CCEEEEEECCCCCHHHCCC...		

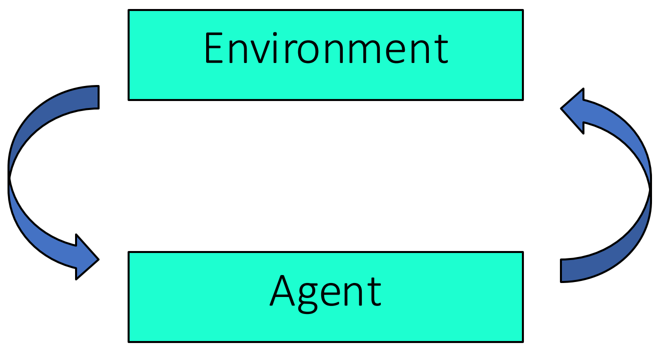
Many more possible structures between y_i , e.g. spatial , temporal, relational ...

Reinforcement Learning (RL)
[Not IID, Sequential]

$\underbrace{X_t \rightarrow Y_t}_{r_t} \rightarrow X_{t+1} \rightarrow Y_{t+1}$

- What's Reinforcement Learning?

{Observation, Reward}

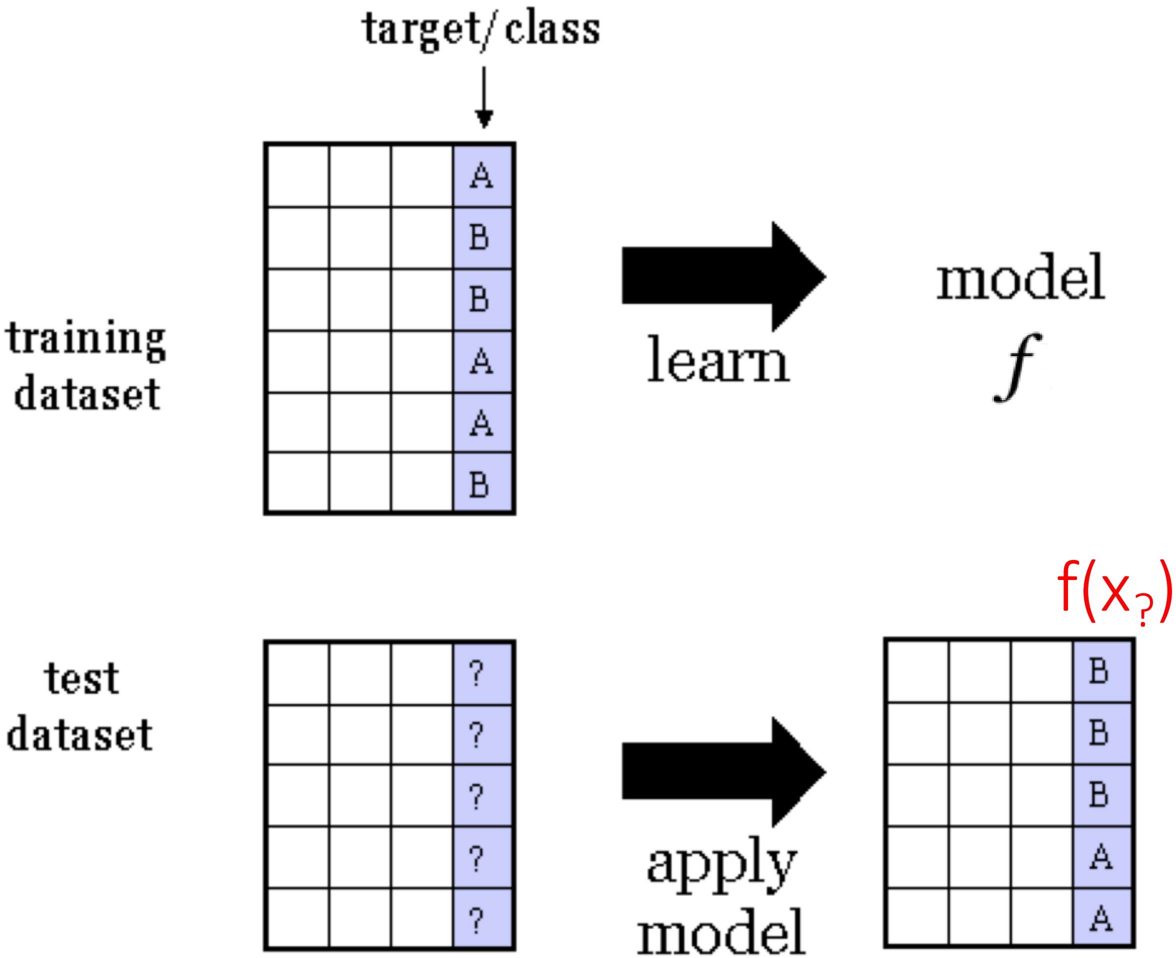


{Actions}

- Agent interacts with an environment and learns by maximizing a scalar reward signal
  - Basic version: No labels or any other supervision signal.
  - Variation like imitation learning: supervised

Adapt from Professor Qiang Yang of HK UST

(Most popular:) SUPERVISED Classification



Training dataset consists of input-output pairs

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

Many Variants of SUPERVISED Classification

- Binary Classification
- Multi-class Classification
- Hierarchical Classification
- Multi-label Classification
- Structured Predictions
- ……

Binary: Text Review-based Sentiment Classification

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

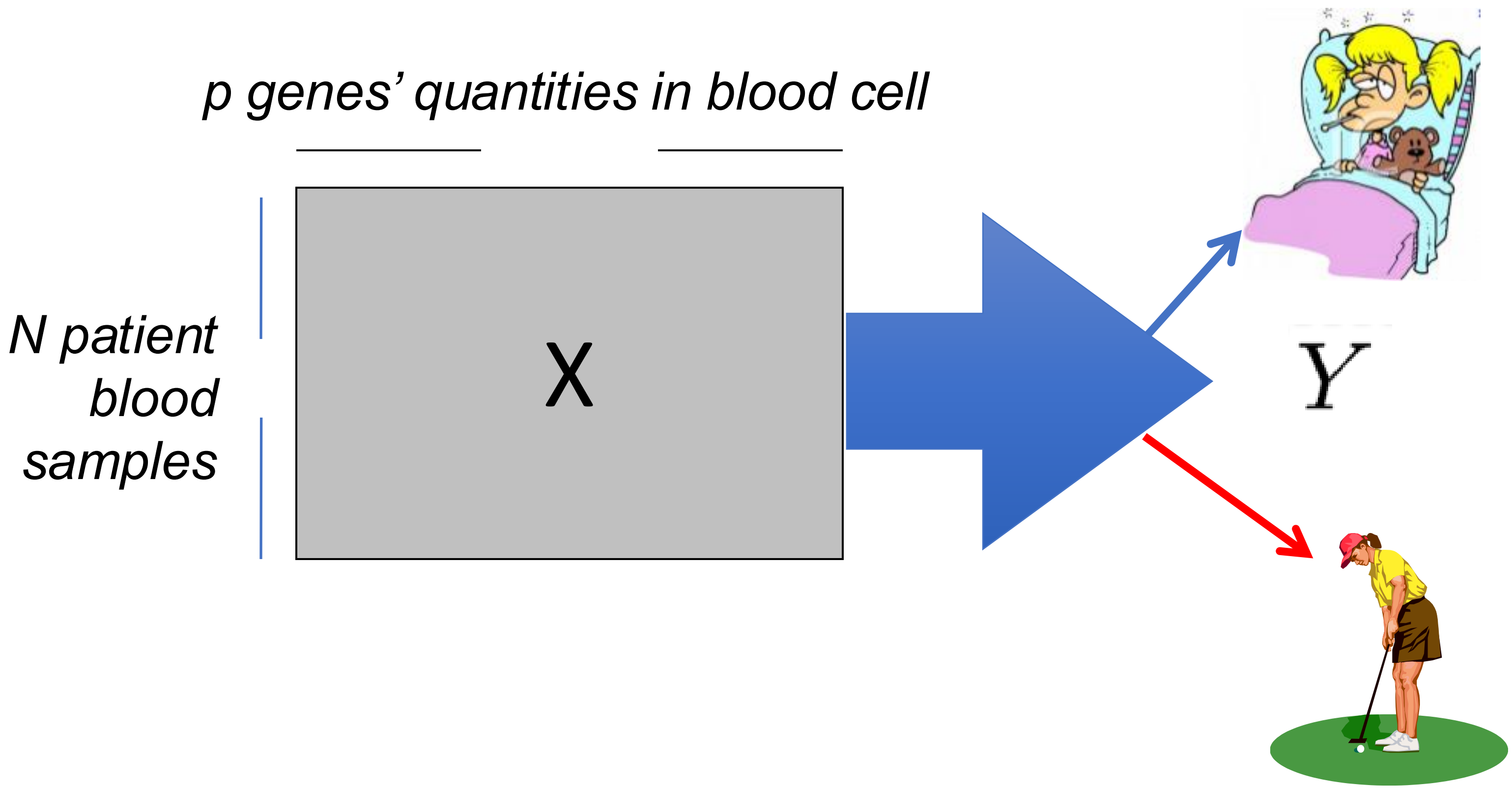
y

-1

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

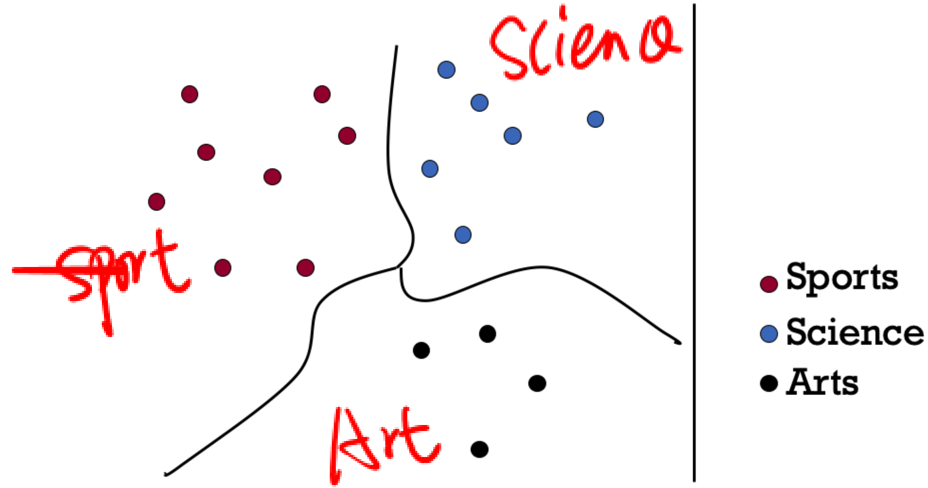
e.g. Is this a positive product review ?

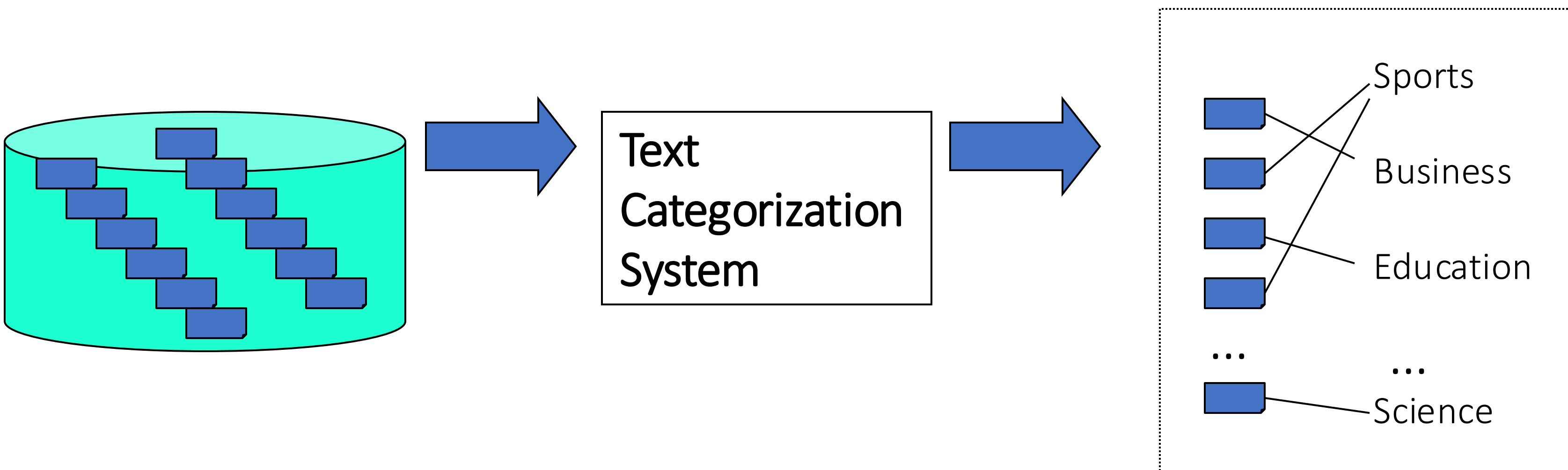
Binary: : Disease Classification using gene expression



Multi-Class: Text Categorization

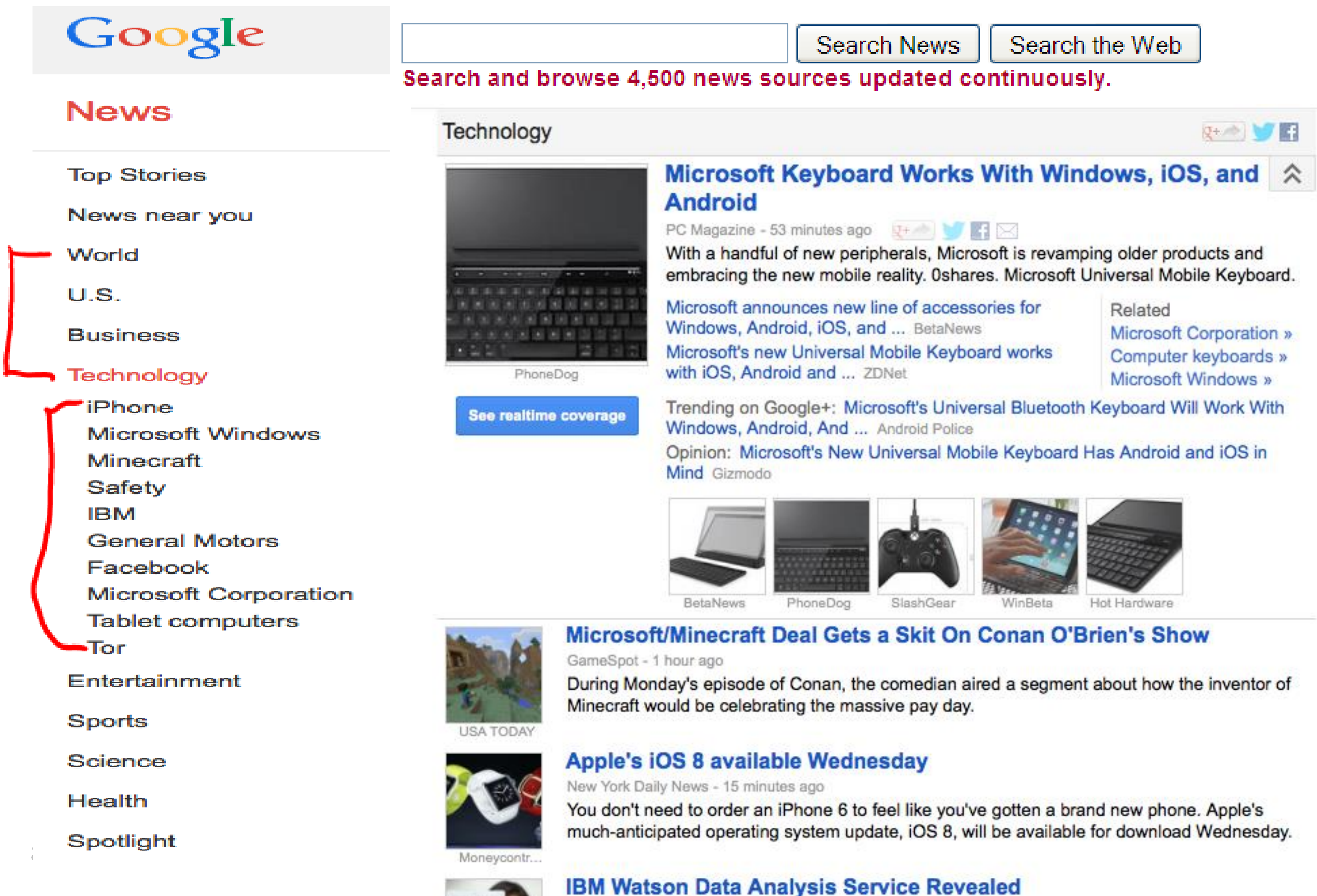
- Almost the most basic/ standard supervised classification problem





$$\hat{y} = f(x)$$

Hierarchical: Text Categorization, e.g. Google News



Multi Label Classification (MLC)

- MLC is the task of assigning a set of target labels for a given sample
- Given input ***x***, predict the set of labels $\{ y_1, y_2, \ldots, y_L \}$, $y_i \in \{0,1\}$

x

y_1 **Castle**

y_2 **City**

y_3 **Mountains**

y_4 **Car**

y_5 **Road**

Generating X: Edges2Image

$z \xrightarrow{f} \underset{\sim}{X}$

Generation

[Isola et al. CVPR 2017]

Generating X: Text2Image

this small bird has a pink breast and crown, and black primaries and secondaries.

this magnificent fellow is almost all black with a red crest, and white cheek patch.

the flower has petals that are bright pinkish purple with white stigma

this white and yellow flower have thin white petals and a round yellow stamen

[Reed et al. ICML 2016]

Open AI recent: DALL·E: Creating Images from Text

TEXT PROMPT

an illustration of a baby daikon radish in a tutu walking a dog

AI-GENERATED IMAGES

Edit prompt or view more images↓

TEXT PROMPT

an armchair in the shape of an avocado. . . .

AI-GENERATED IMAGES


Thank You

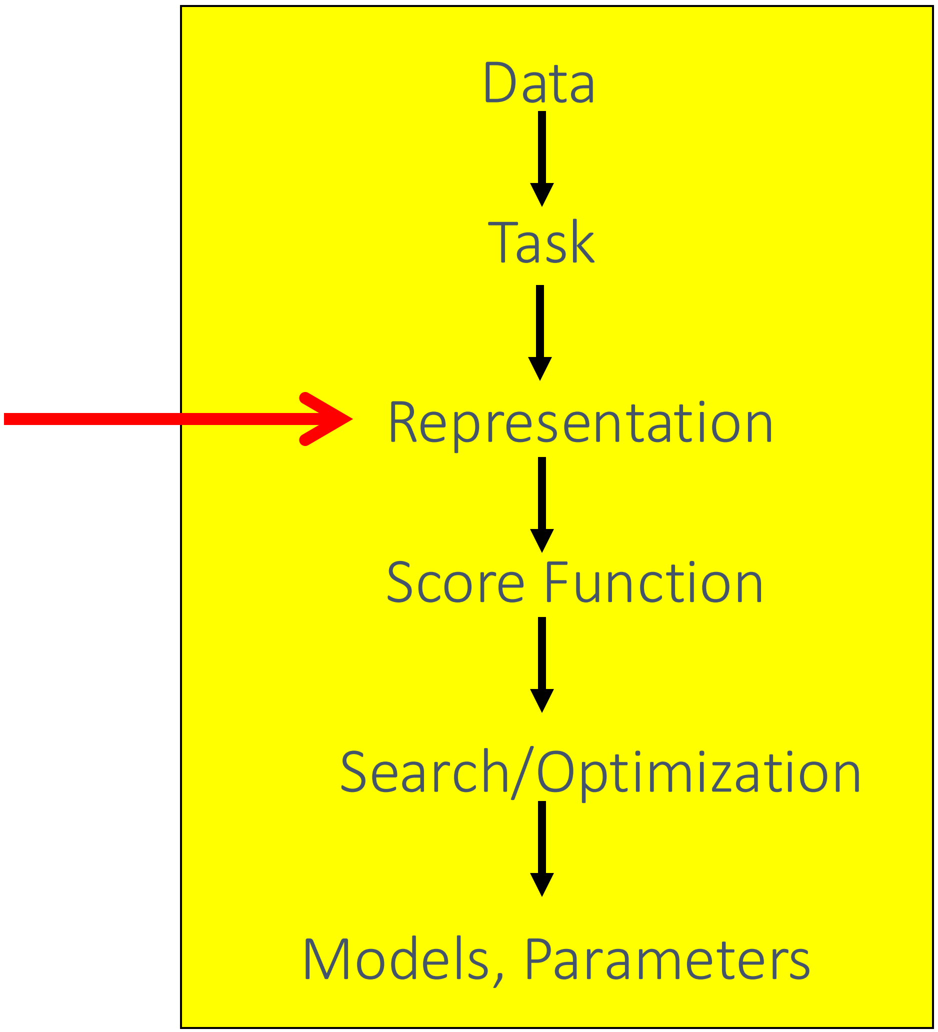
Roadmap

- Machine Learning in a Nutshell

- Examples of Different Tasks
- Examples of Different Data Types

- Examples of Different Representation Types
- Examples of Different Loss/Cost Types
- Examples of Different Model Properties

Machine Learning in a Nutshell

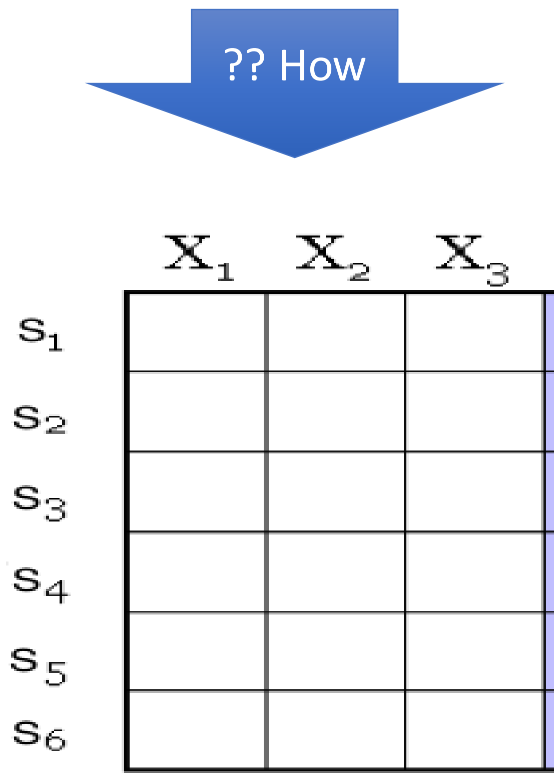


ML grew out of work in AI

Optimize a performance criterion using example data or past experience,

Aiming to generalize to unseen data

- Text / String / Symbolic
- Sequences / Sets / Graph
  - Variable length
  - Discrete
  - Combinatorial
  - Spatial ordering among units



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 …

Vector Space Representation: Bag of Words Trick

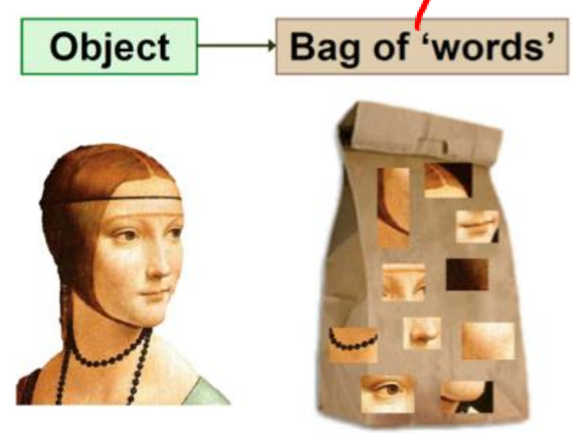
- Each document is a vector, one component for each term (= word).

M

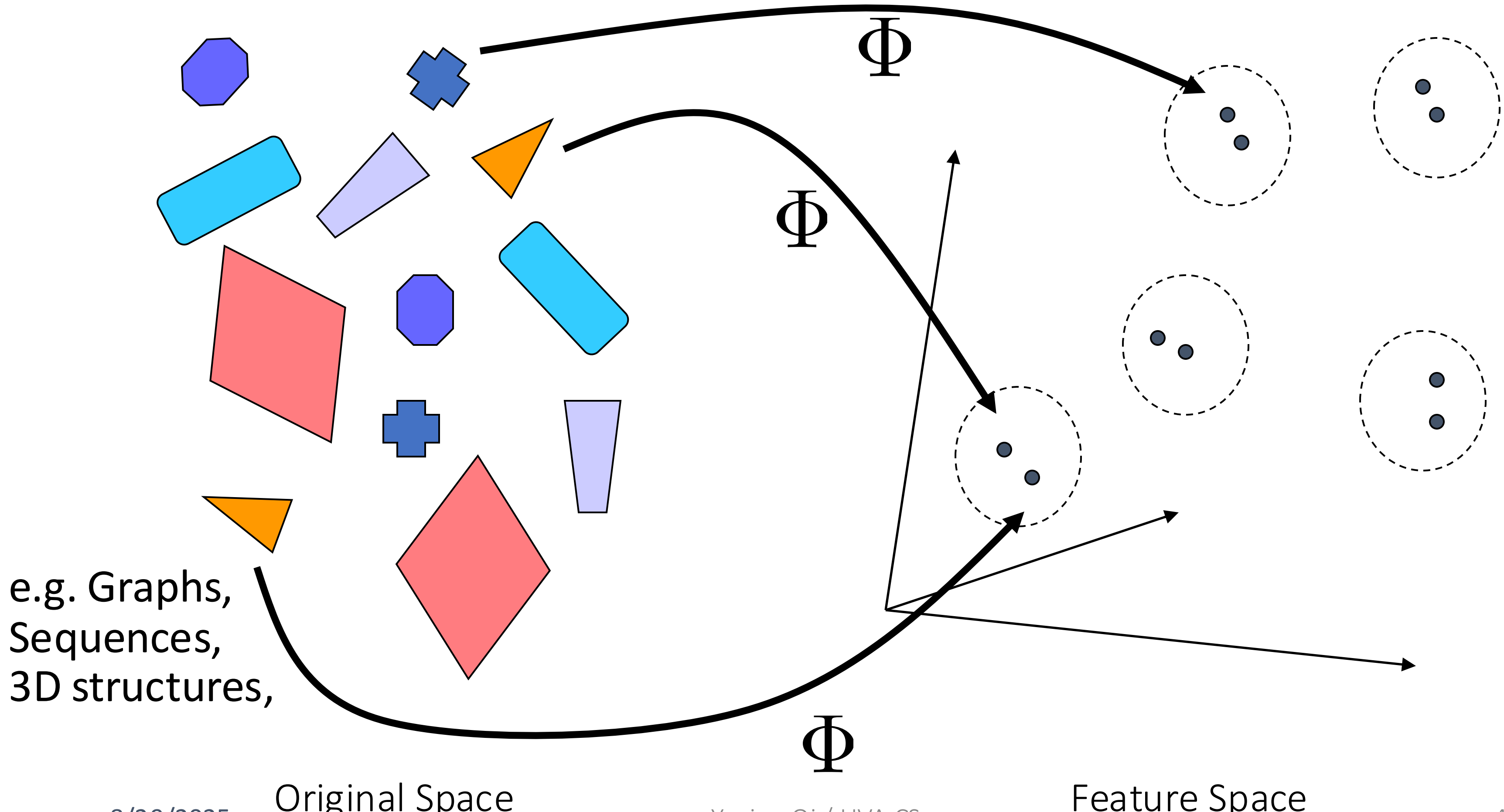
	Doc 1	Doc 2	Doc 3	...
Word 1	3	0	0	...
Word 2	0	8	1	...
Word 3	12	1	10	...
...	0	1	3	...
...	0	0	0	...

- Normalize to unit length.
- High-dimensional vector space:
  - Terms are axes, 10,000+ dimensions, or even 100,000+
  - Docs are vectors in this space

Visual



STRUCTURAL INPUT : Kernel Methods [Complex X]



DEEP LEARNING / FEATURE LEARNING :

Low-level sensing → Pre-processing → Feature extract. → Feature selection → Inference: prediction, recognition

Feature Engineering (before 2012)

- ✓ 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

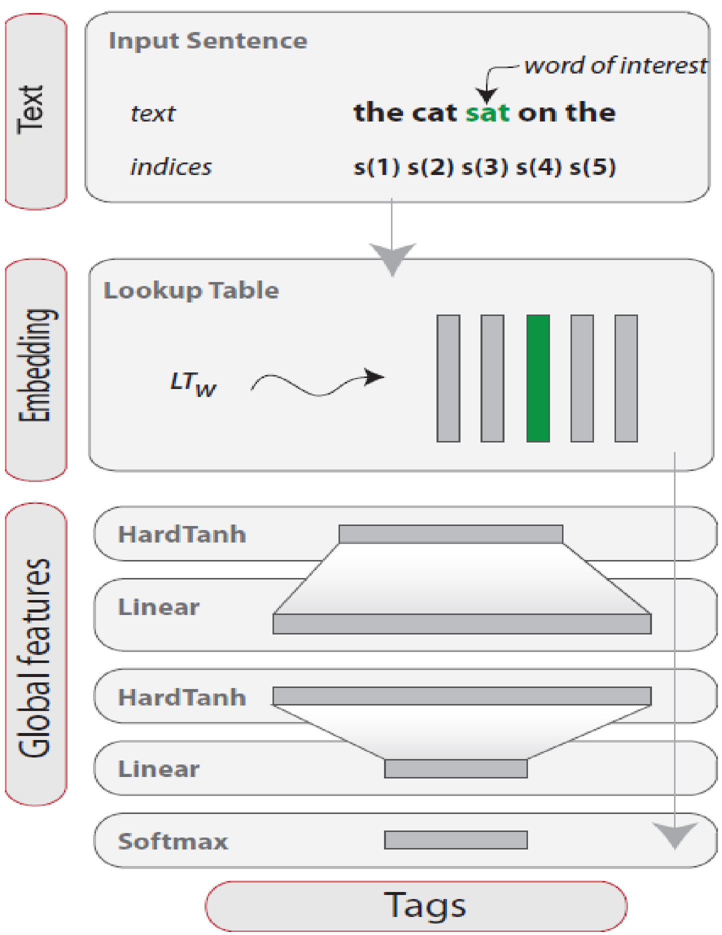
Feature Learning

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

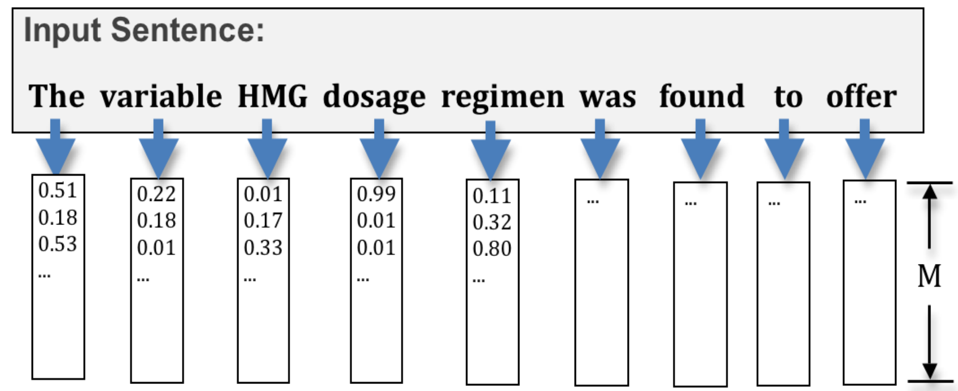
wrt. Data types

MORE RECENT: Deep Learning Based

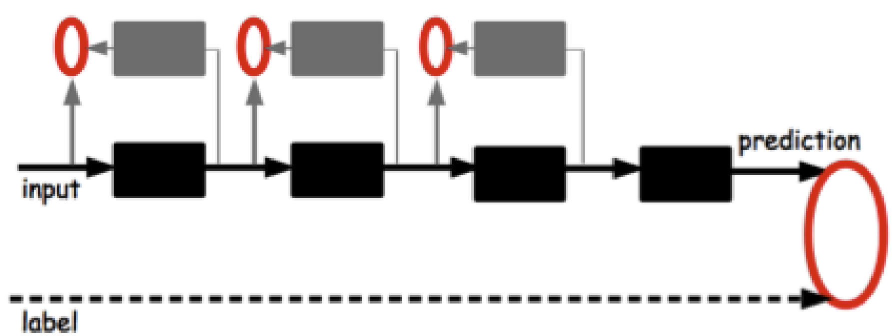
Deep Multi-Layer Learning



Supervised Embedding



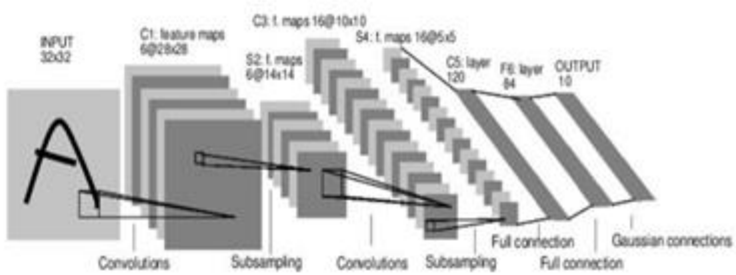
Layer-wise Pretraining



History of ConvNets

1998

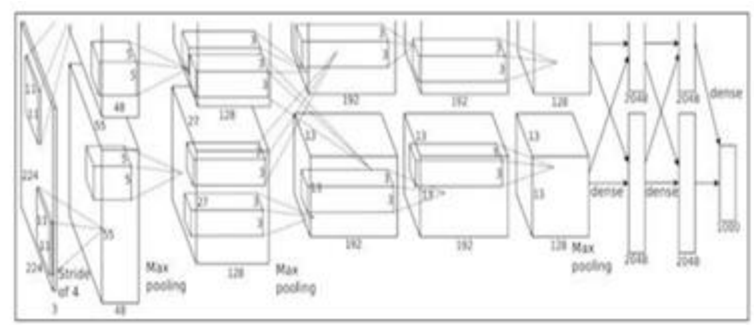
Gradient-based learning applied to document recognition [LeCun, Bottou, Bengio, Haffner]



LeNet-5

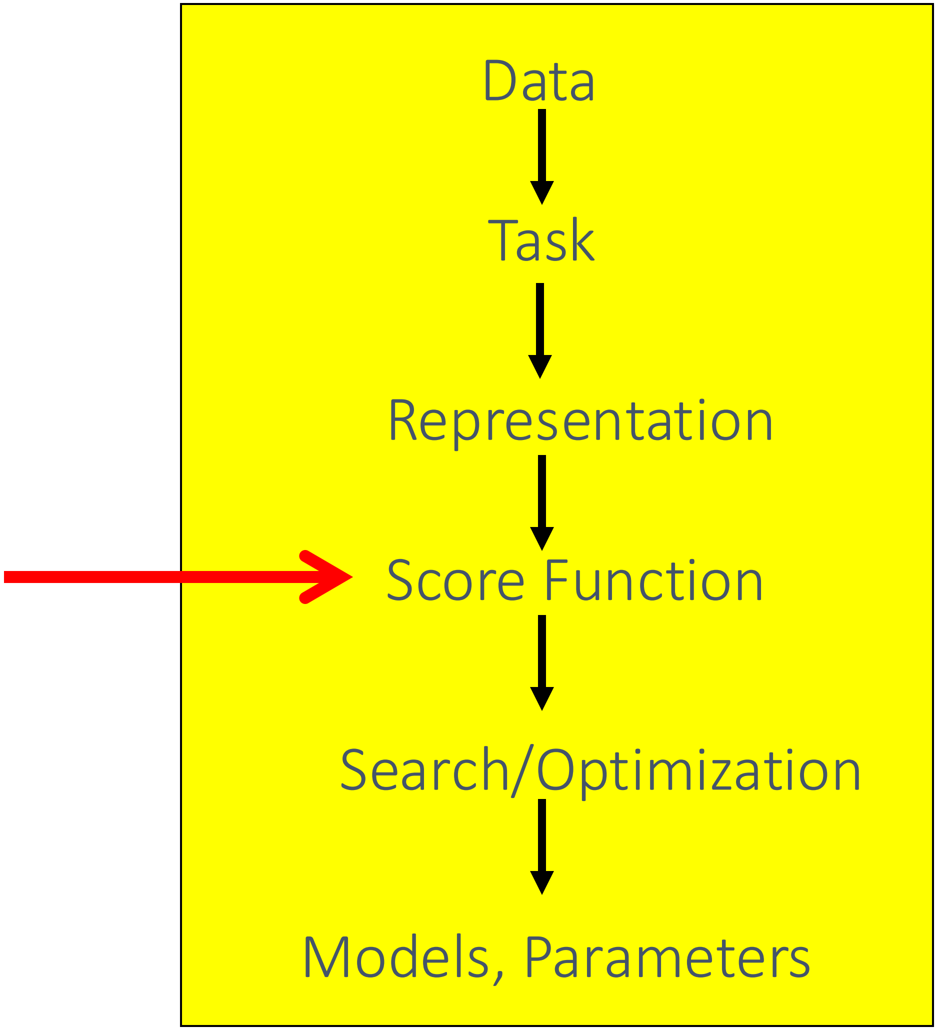
2012

ImageNet Classification with Deep Convolutional Neural Networks [Krizhevsky, Sutskever, Hinton, 2012]



“AlexNet”

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 $x_1,\dots,x_L$
    - available corresponding labels $y_1,\dots,y_L$

  - Find (w,b) by minimizing loss
    - (i.e. difference between y and f(x) on available examples in training set)

??

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

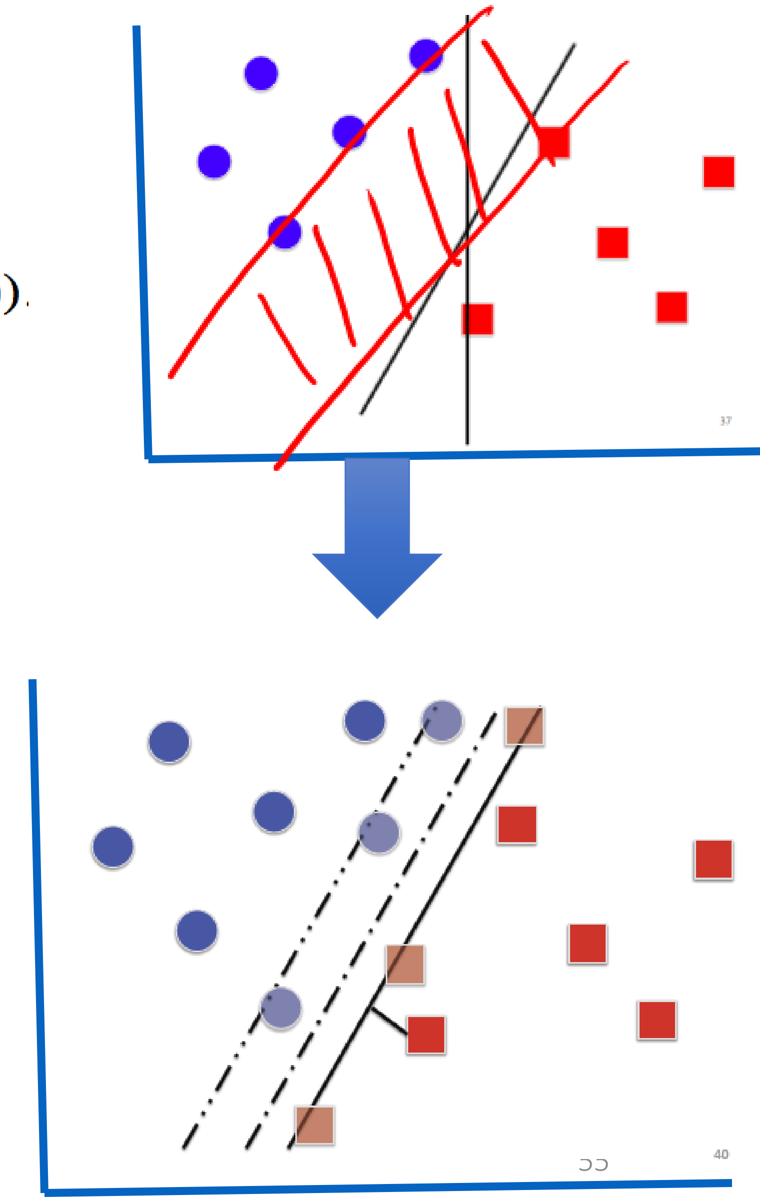
Basic Concepts

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

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

$|f(x_i) - y_i|$

$(f(x_i) - y_i)^2$



- 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

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 !!!

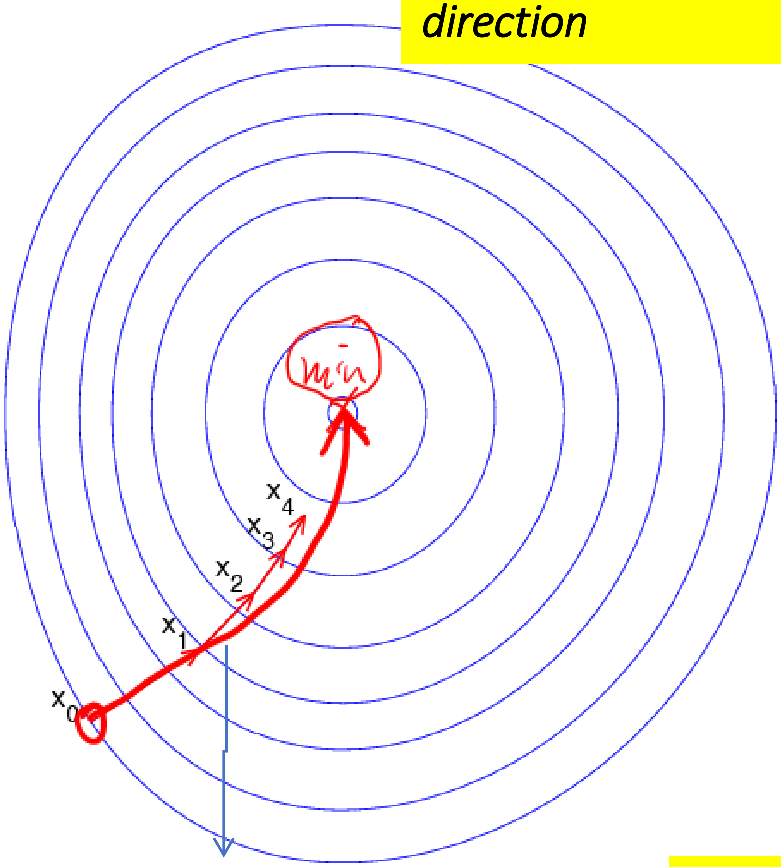
Not the focus, being covered in my advanced-level course

Gradient Descent (Steepest Descent) – contour map view

A first-order optimization algorithm.

To find a local minimum of a function using gradient descent, one takes steps proportional to the *negative* of the gradient of the function at the current point.

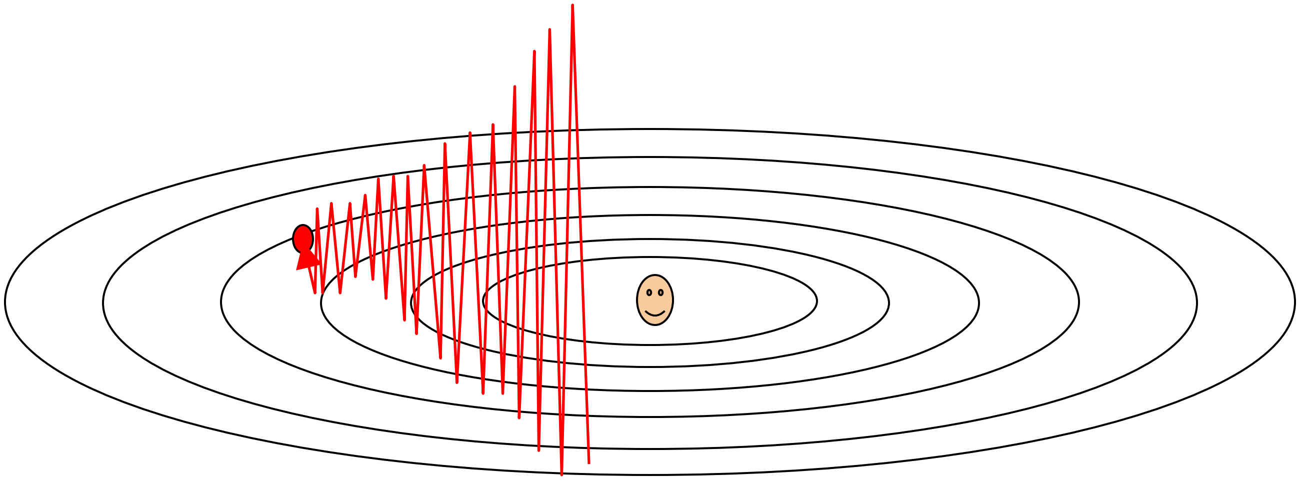
The gradient (in the variable space) points in the direction of the greatest rate of increase of the function and its magnitude is the slope of the surface graph in that direction



$$-\nabla_x F(x_{k-1})$$

Contour map view

Gradient Magnitudes:

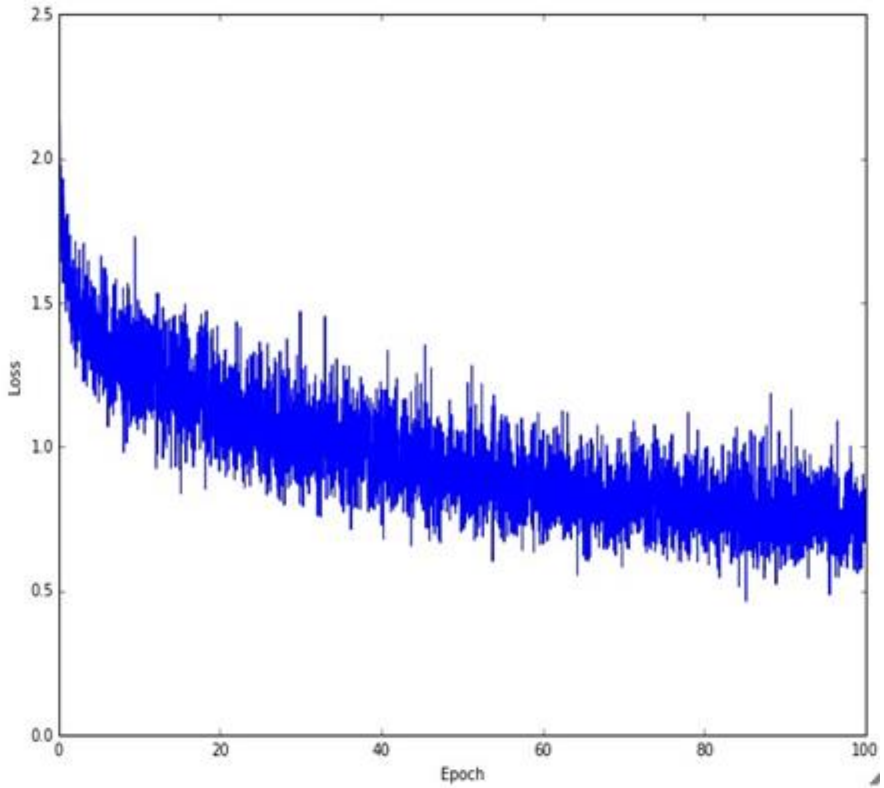


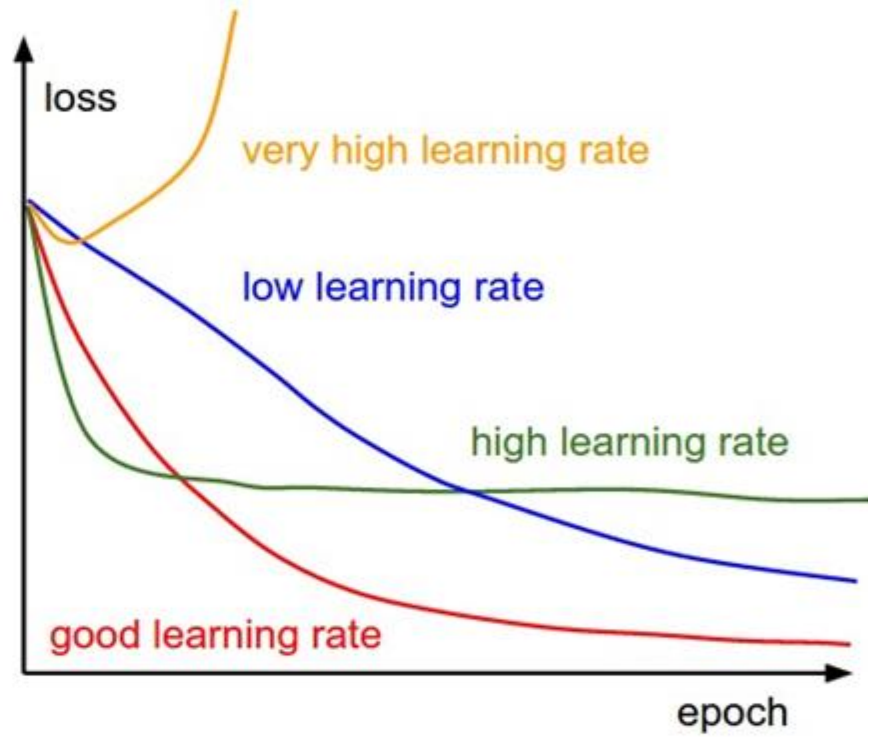
Gradients too big → divergence
Gradients too small → slow convergence

Divergence is much worse!

Many great tools, e.g., Adam
https://arxiv.org/abs/1609.04747

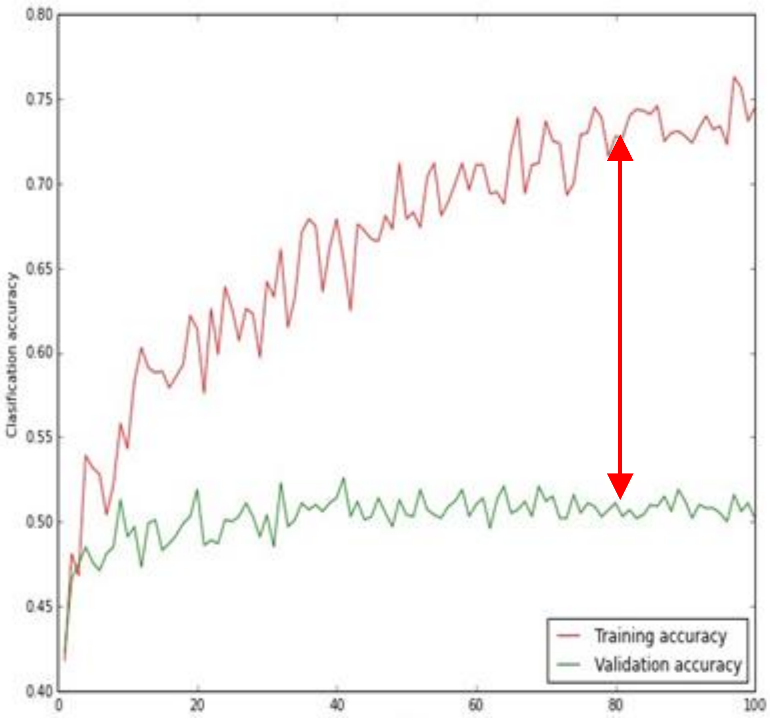
Monitor and visualize the loss curve





From Feifei Li Stanford Cousre

Monitor and visualize the train / validation loss / accuracy: Bias Variance Tradeoff



big gap = overfitting

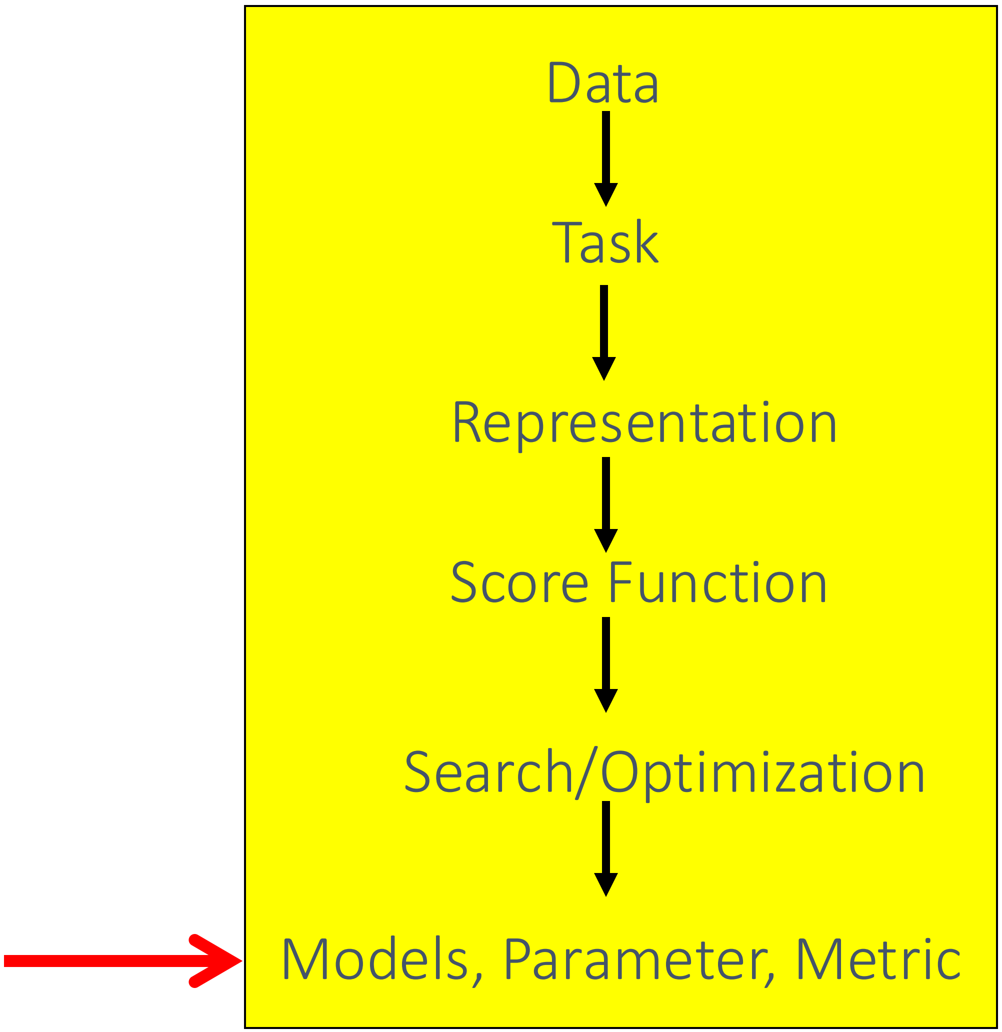
=> increase regularization strength?

no gap, e.g. underfitting / both bad

=> increase model capacity?

From Feifei Li Stanford Cousre

Machine Learning in a Nutshell

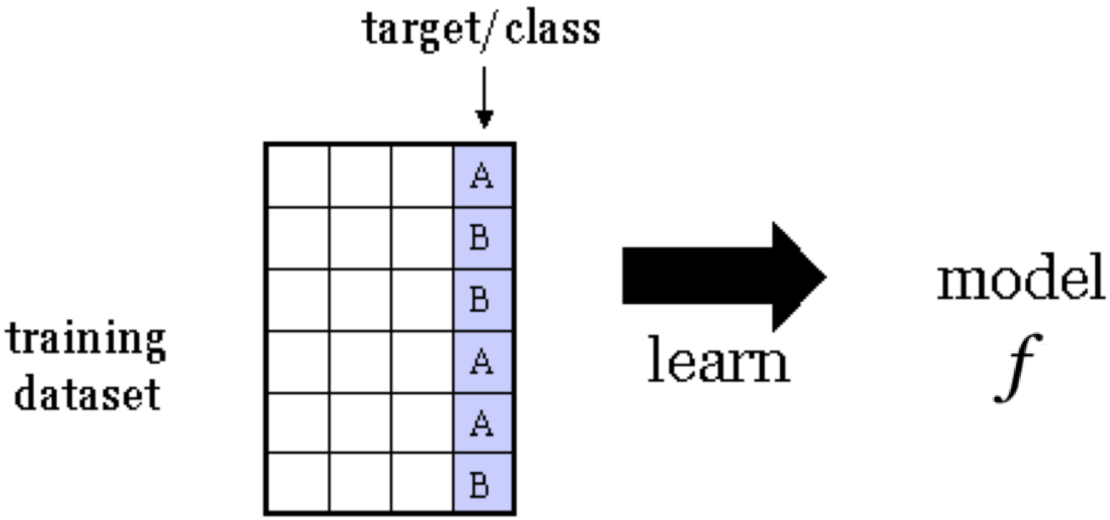


ML grew out of work in AI

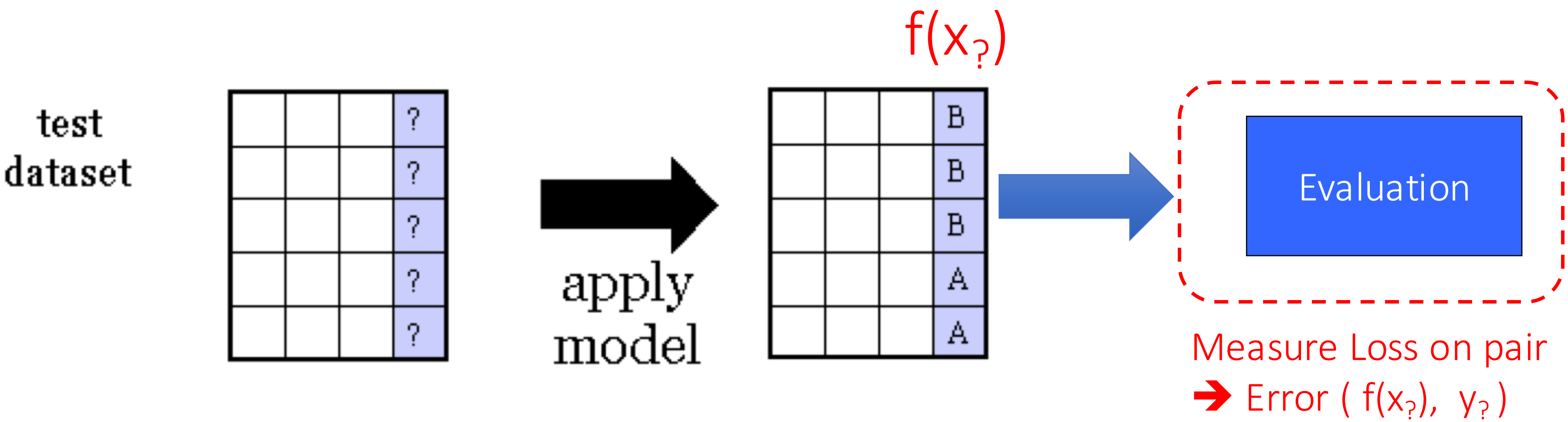
Optimize a performance criterion using example data or past experience,

Aiming to generalize to unseen data

How to know the program works well: Measure Prediction Accuracy on Test Data



Training dataset consists of input-output pairs

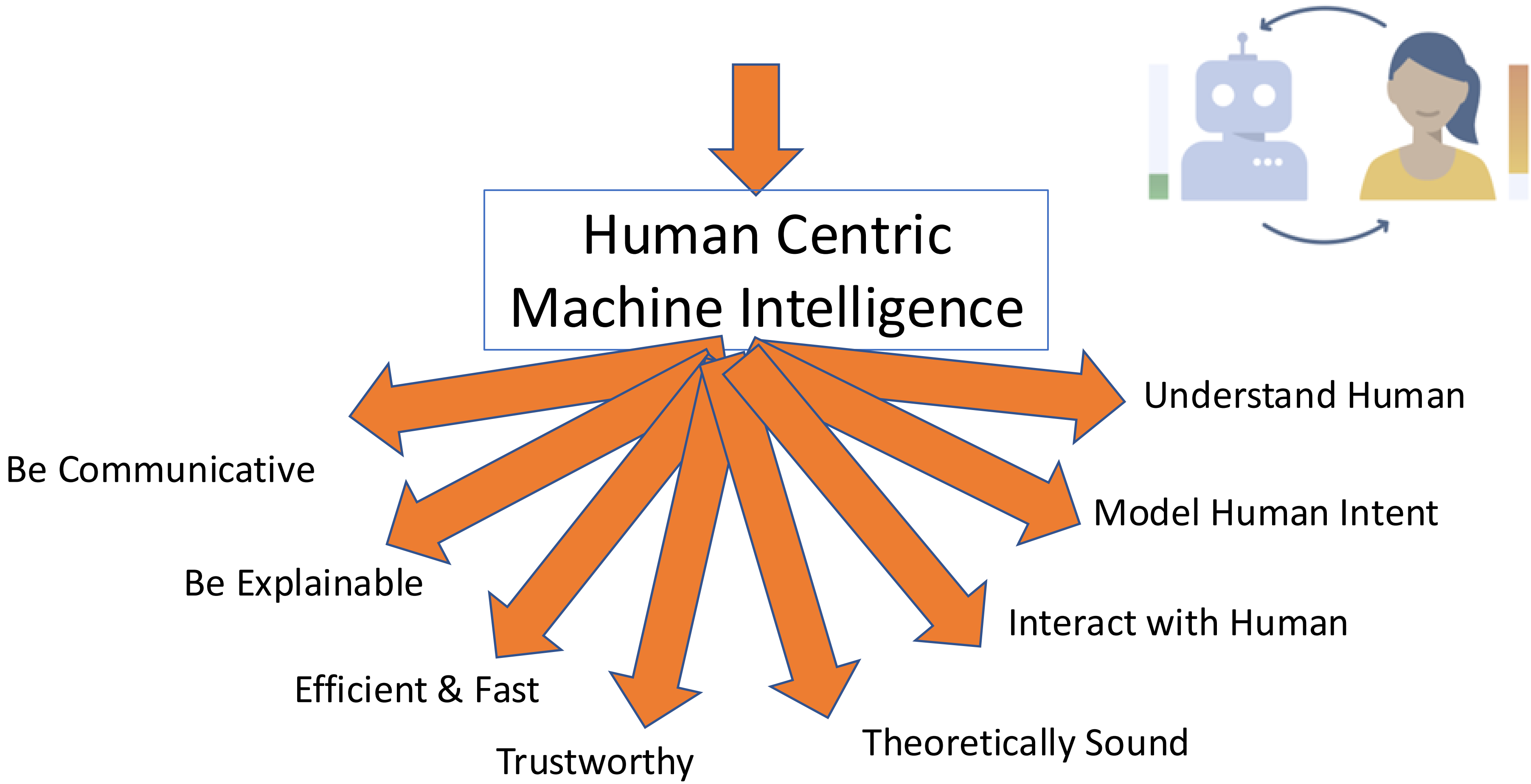


Many Metrics for Supervised Classification

Metric	**Formula**	**Interpretation**
Accuracy	$\dfrac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}$	Overall performance of model
Precision	$\dfrac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$	How accurate the positive predictions are
Recall Sensitivity	$\dfrac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$	Coverage of actual positive sample
Specificity	$\dfrac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}$	Coverage of actual negative sample
F1 score	$\dfrac{2\mathrm{TP}}{2\mathrm{TP}+\mathrm{FP}+\mathrm{FN}}$	Hybrid metric useful for unbalanced classes

Credit: Stanford Machine Learning course

Beyond Prediction Accuracy: e.g., ML and AI Research @ UVA CS



Robustness of DNN, e.g. Adversarial Examples (AE)

f(x) "panda" $+ \quad 0.007 \times [\underbrace{noise}_{\delta}] =$ "gibbon" f(x+δ)

Example from: Ian J. Goodfellow, Jonathon Shlens, Christian Szegedy. Explaining and Harnessing Adversarial Examples. ICLR 2015.

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

Rough Sectioning of this Course

- 1. Basic Supervised Regression + Tabular Data
- 2. Basic Deep Learning + 2D Imaging Data
- 3. Advanced Supervised learning + Tabular Data
- 4. Generative and Deep + 1D Sequence Text Data
- 5. Not Supervised

Now open course website Schedule page ➔


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