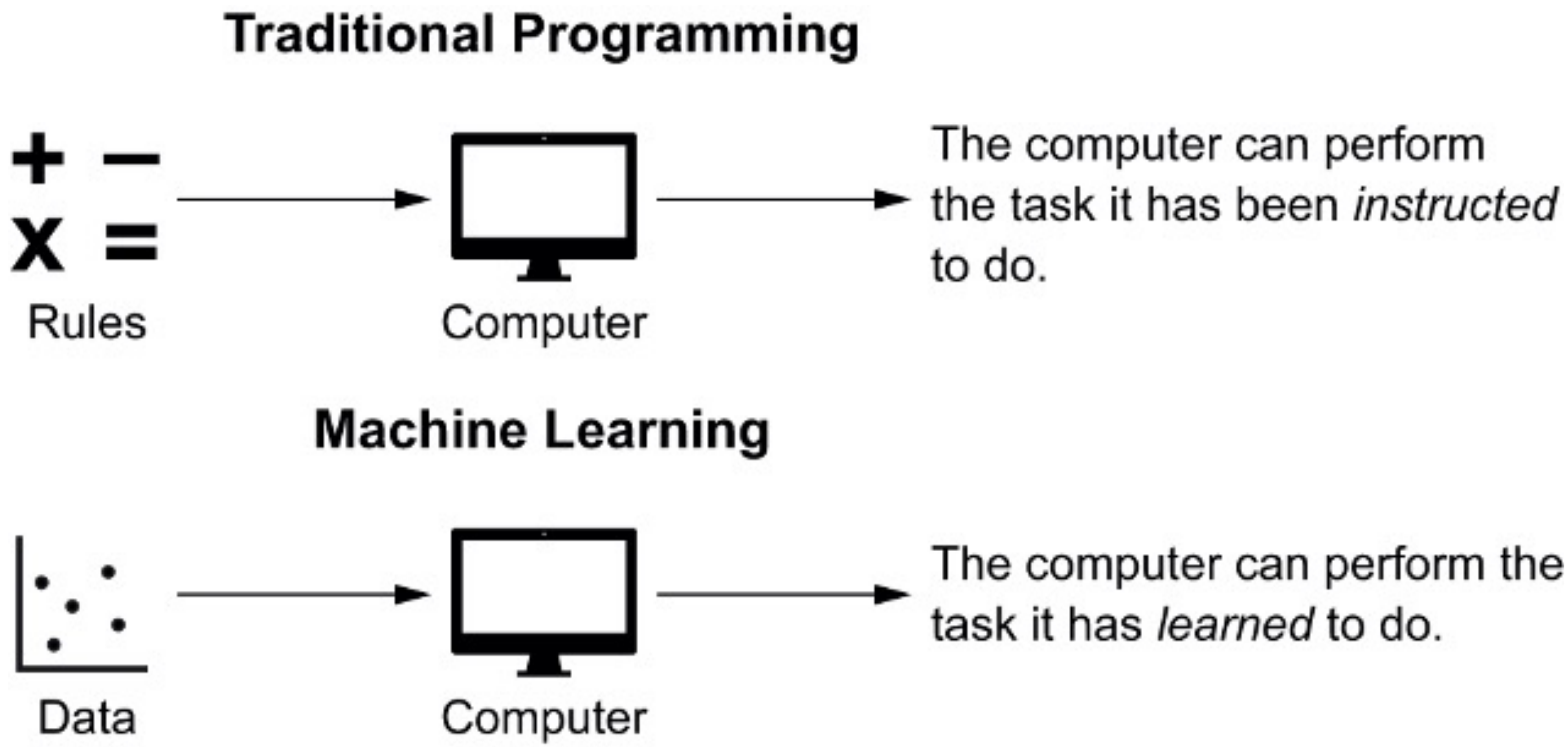
Fundamentals and Landscape of Classical Machine Learning (I)

- Learning? What do we mean?
- Is learning feasible?

https://www.sli.do/
#073374

Learning? What do we mean?



- **Can computer automatically learn these rules from data?**
- **If so, can these rules be applied for new data to guess answers?**

Valigi and Mauro (2020), Zero to AI, Manning Publications.

Machine Learning (**Supervised Learning**): rules can be implicitly

x Data →

y Answers (Labeled) →

$y = f(x)$
Machine Learning

(Model, Implicit Rule)

→ Answers

(e.g., Cat or Dog)

Fun time: which of the following is best suited for machine learning?

(1) Throw a dice and predict its face value
(2) Sort a few points in space
(3) Decide credit card approval for a customer
(4) Predict the next big earthquake

https://www.sli.do/
#073374

Components of Learning: Metaphor Using Credit Approval

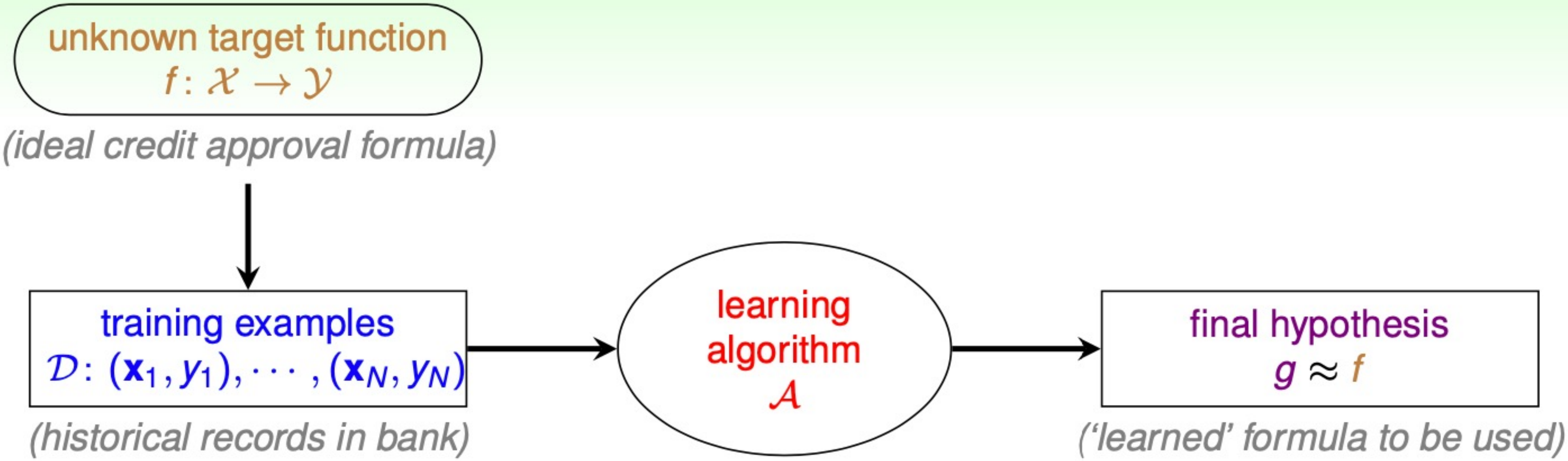
Applicant Information

age	23 years
gender	female
annual salary	NTD 1,000,000
year in residence	1 year
year in job	0.5 year
current debt	200,000

unknown pattern to be learned:

'approve credit card good for bank?'

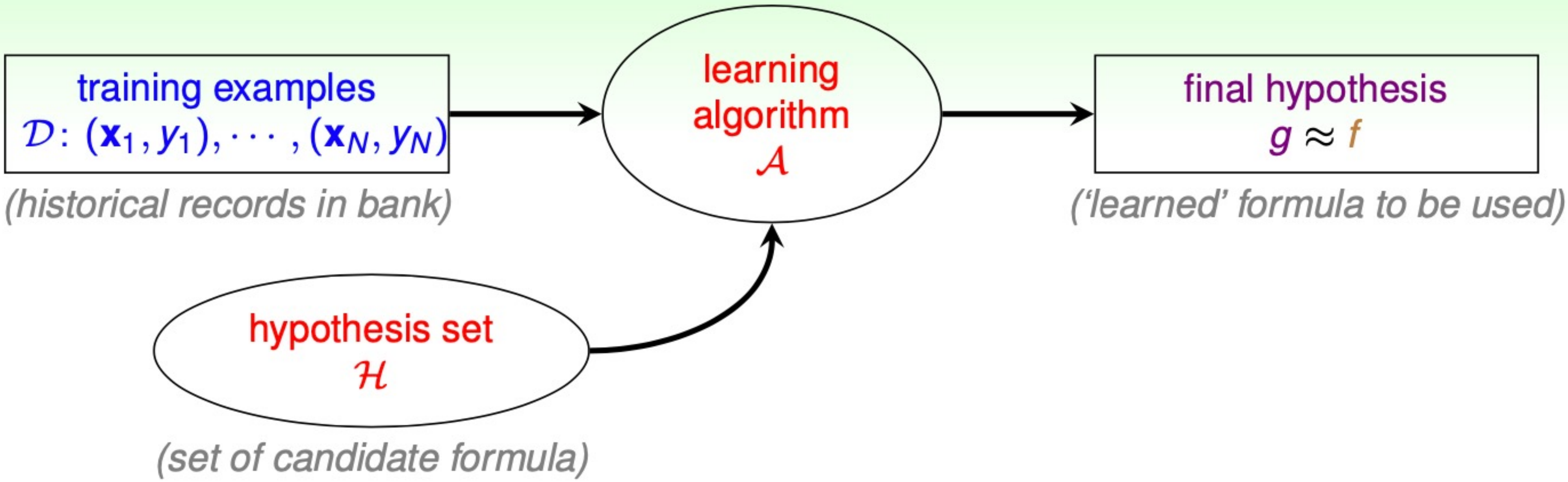
Learning Flow for Credit Approval



- target f **unknown**
  (i.e. no programmable definition)
- hypothesis g hopefully $\approx f$
  but possibly **different** from f
  (perfection 'impossible' when f unknown)

What does g look like?

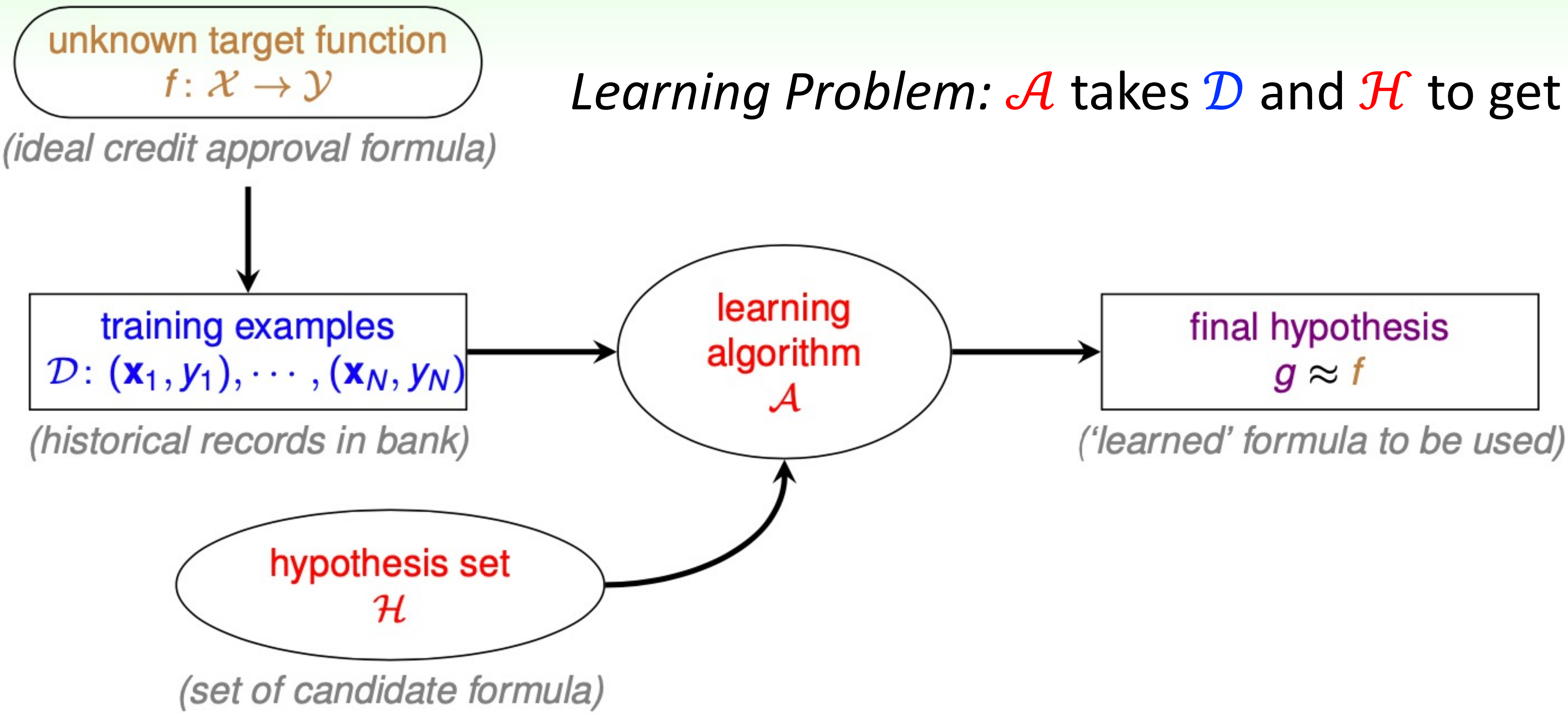
The Learning Model



- assume $g \in \mathcal{H} = \{h_k\}$, i.e. approving if
  - h_1: annual salary > NTD 800,000
  - h_2: debt > NTD 100,000 (really?)
  - h_3: year in job ≤ 2 (really?)
- hypothesis set $\mathcal{H}$:
  - can contain **good or bad hypotheses**
  - up to $\mathcal{A}$ to pick the 'best' one as g

learning model = $\mathcal{A}$ and $\mathcal{H}$

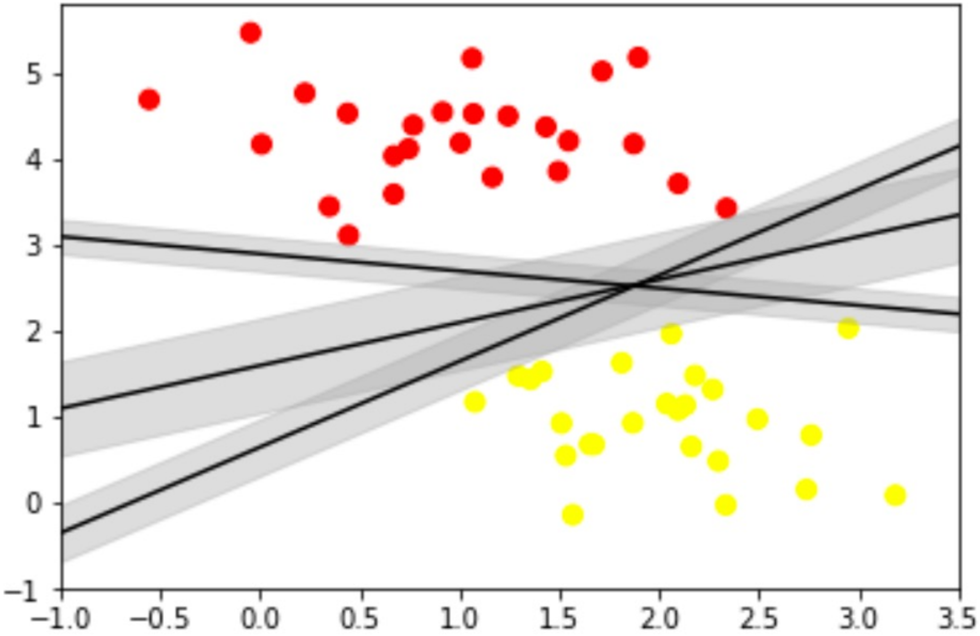
Practical Definition of Machine Learning



Learning Problem: $\mathcal{A}$ takes $\mathcal{D}$ and $\mathcal{H}$ to get g

machine learning:
use data to compute hypothesis g
that approximates target f

Learning Problem in Practice: learning algorithm $\mathcal{A}$ takes training examples $\mathcal{D}$ and hypothesis set $\mathcal{H}$ to get final hypothesis g.



$$w_1 x_1 + w_2 x_2 + b = 0$$

In support vector machines (SVM), the line that maximizes this margin is the one we will choose as the optimal model.

Fun time: Quick Check (who is who)

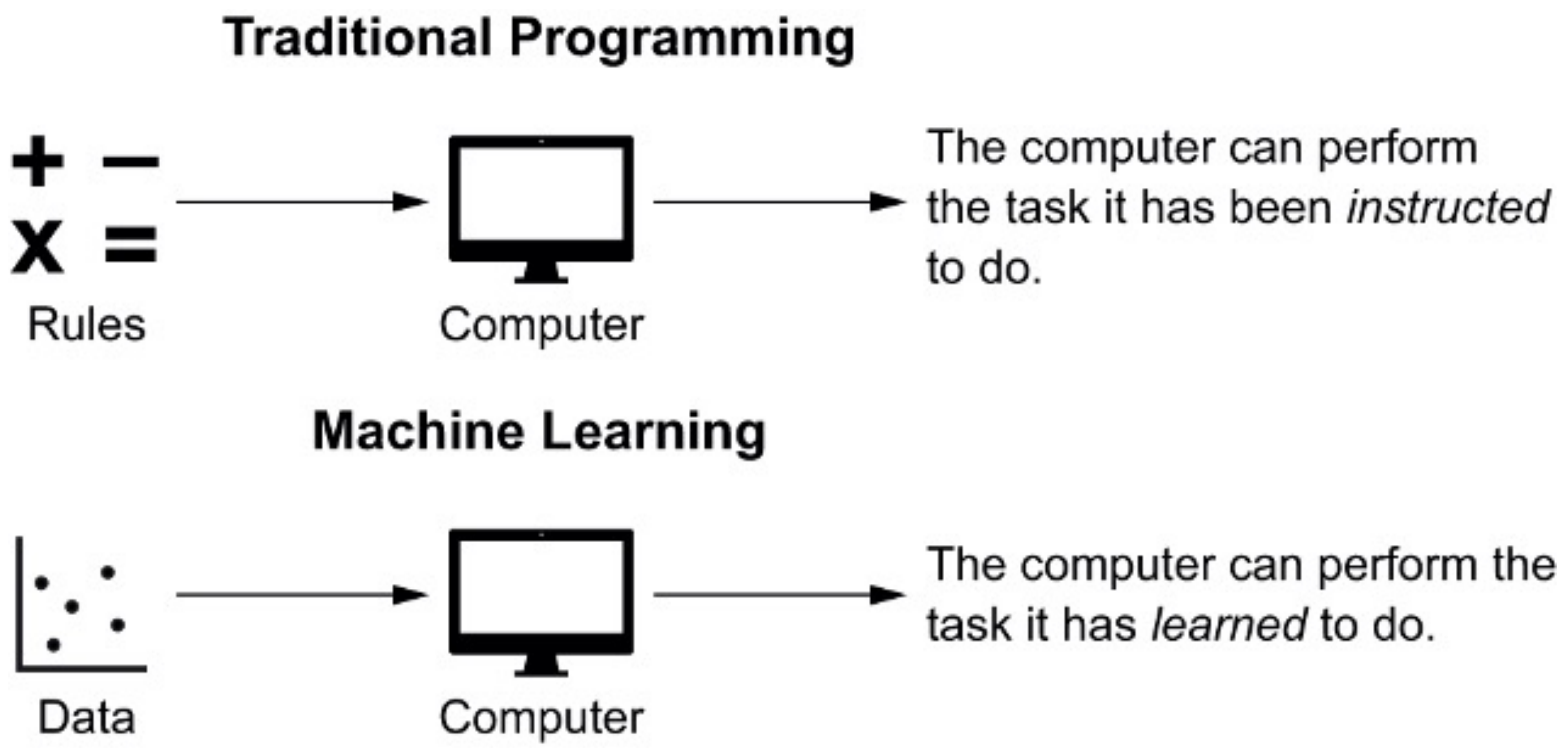
- **Learning algorithm $\mathcal{A}$**
- **Hypothesis set $\mathcal{H}$**
- **Training Examples $\mathcal{D}$**
- **Final hypothesis g**
- **Target function f**

Summary

Learning? What do we mean?

- Machine learning involves building mathematical models to help understand data.
- In practice, we use data to compute hypothesis g that approximate unknown target f.
- In practice, learning algorithm $\mathcal{A}$ takes training examples $\mathcal{D}$ and hypothesis set $\mathcal{H}$ to get final hypothesis g.
- "Learning" enters the picture when we give these models tunable parameters that can be adapted to observed data; in this way the program can be considered to be "learning" from the data.

Is learning feasible?



- **Can computer automatically learn these rules from data?**
  - **We use data to compute hypothesis *g* that approximates target *f***
- **If so, can these rules be applied for new data to guess answers? (generalization)**

Valigi and Mauro (2020), Zero to AI, Manning Publications.

Is learning feasible?

Fun Time: Can final hypothesis *g* predict new data?

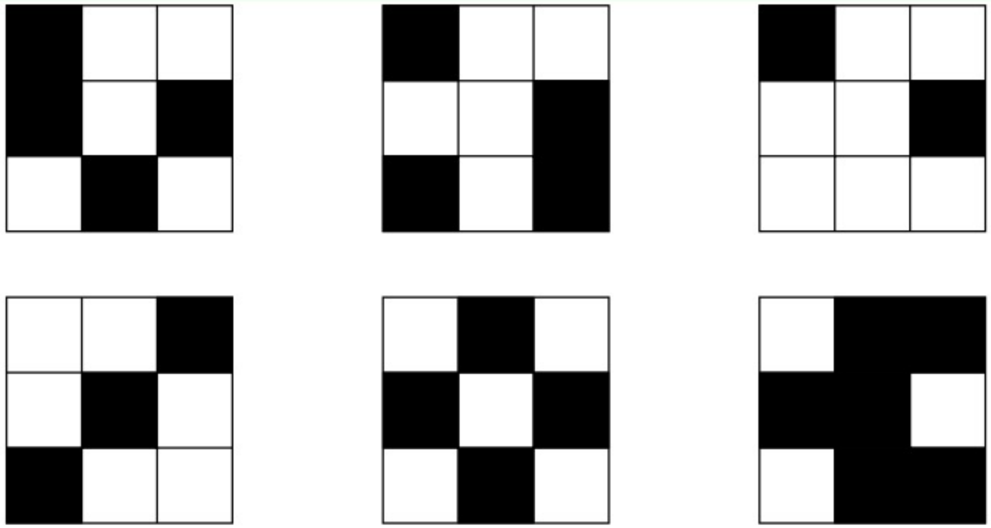
(1) Yes

(2) No

(3) Maybe

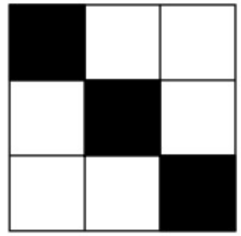
https://www.sli.do/
#073374

A Learning Puzzle



$y_n = -1$

$y_n = +1$



$g(\mathbf{x}) = ?$

Fun Time: *g(x)* prediction of new data

(1) -1

(2) +1

https://www.sli.do/
#073374

let's test your 'human learning' with 6 examples :-)

A 'Simple' Binary Classification Problem

$\mathbf{x}_n$	$y_n = f(\mathbf{x}_n)$
0 0 0	$\circ$
0 0 1	$\times$
0 1 0	$\times$
0 1 1	$\circ$
1 0 0	$\times$

- $\mathcal{X} = \{0, 1\}^3$, $\mathcal{Y} = \{\circ, \times\}$, can enumerate all candidate f as $\mathcal{H}$

Let us do a two-bit case to explore (1) the dimension of the input space (2) all the samples in the input space (3) all the possible hypotheses (and we can pick up one as our target function)

A 'Simple' Binary Classification Problem

	$\mathbf{x}$	y	g	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
$\mathcal{D}$	0 0 0	○	○	○	○	○	○	○	○	○	○
	0 0 1	×	×	×	×	×	×	×	×	×	×
	0 1 0	×	×	×	×	×	×	×	×	×	×
	0 1 1	○	○	○	○	○	○	○	○	○	○
	1 0 0	×	×	×	×	×	×	×	×	×	×
	1 0 1		?	○	○	○	○	×	×	×	×
	1 1 0		?	○	○	×	×	○	○	×	×
	1 1 1		?	○	×	○	×	○	×	○	×

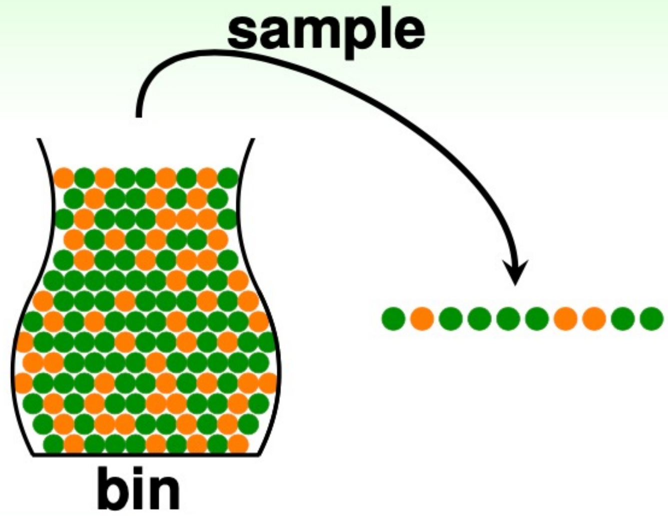
- $g \approx f$ inside $\mathcal{D}$: sure!
- $g \approx f$ outside $\mathcal{D}$: **No!** (but that's really what we want!)

learning from $\mathcal{D}$ (to infer something outside $\mathcal{D}$)
is doomed if **any 'unknown' f can happen**. **:-(**

Is learning doomed (完蛋了)?
If so, this will be a very short course!!!

Probability to recuse!

Statistics 101: Inferring Orange Probability



bin

assume

orange probability $= \mu$,
green probability $= 1 - \mu$,

with μ **unknown**

sample

N marbles sampled independently, with

orange fraction $= \nu$,
green fraction $= 1 - \nu$,

now ν **known**

does **in-sample** ν say anything about out-of-sample μ?

Possible versus Probable

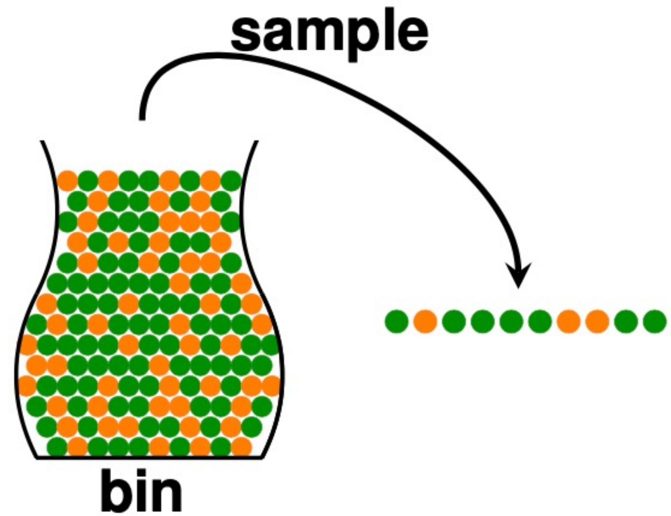
does **in-sample ν** say anything about out-of-sample μ?

No!

possibly not: sample can be mostly green while bin is mostly orange

Yes!

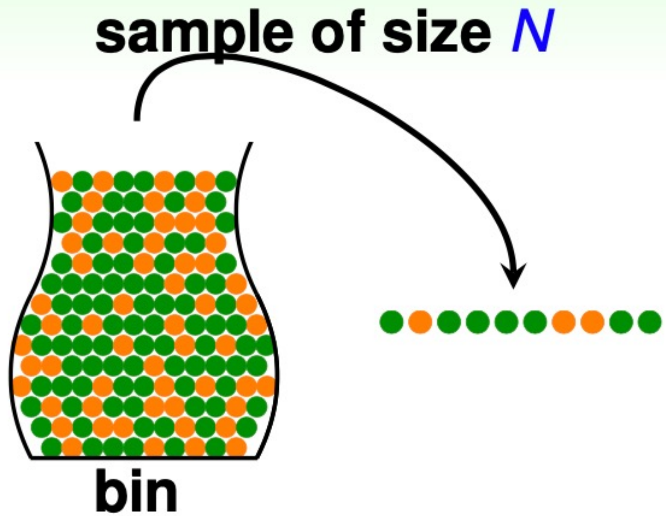
probably yes: in-sample ν likely **close to** unknown μ



formally, **what does ν say about μ?**

Hoeffding's Inequality (1/2)

$\mu =$ orange
probability in bin



$\nu =$ orange
fraction in sample

- in big sample (N large), ν is probably close to μ (within ϵ)

$$\mathbb{P}\left[\left|\nu - \mu\right| > \epsilon\right] \leq 2\exp\left(-2\epsilon^2 N\right)$$

- called **Hoeffding's Inequality**, for marbles, coin, polling, . . .

the statement '$\nu = \mu$' is
probably approximately correct (PAC)

Connection to Learning

bin

- unknown orange prob. μ
- marble $\bullet \in$ bin
- orange $\bullet$
- green $\bullet$
- size-N sample from bin

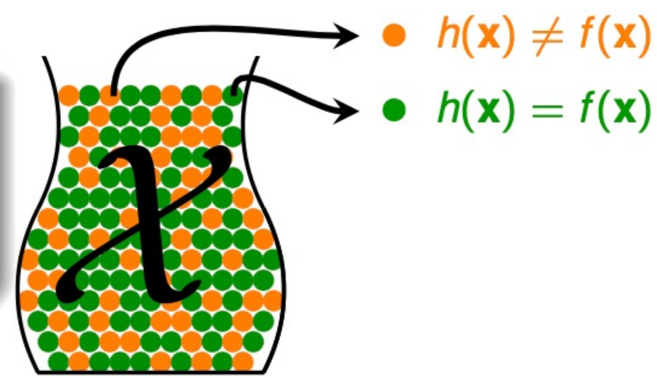
  of i.i.d. marbles

learning

- fixed hypothesis $h(\mathbf{x}) \stackrel{?}{=}$ target $f(\mathbf{x})$
- $\mathbf{x} \in \mathcal{X}$
- h is wrong $\Leftrightarrow h(\mathbf{x}) \neq f(\mathbf{x})$
- h is right $\Leftrightarrow h(\mathbf{x}) = f(\mathbf{x})$
- check h on $\mathcal{D} = \{(\mathbf{x}_n, \underbrace{y_n}_{f(\mathbf{x}_n)})\}$

  with i.i.d. $\mathbf{x}_n$

if **large N** & **i.i.d. $\mathbf{x}_n$**, can **probably** infer
unknown $[\![h(\mathbf{x}) \neq f(\mathbf{x})]\!]$ probability
by known $[\![h(\mathbf{x}_n) \neq y_n]\!]$ fraction



i.i.d. independent and identically distributed

Boolean Learning Example: Take Two

Let us consider a Boolean target function (i.e., $\mathcal{Y} = \{0, 1\}$) over a four-bit vector representation of input space $\{0000, 0001, \ldots, 0111, 1000, 1001, \ldots, 1111\}$.

Q: For this example, what is the dimension of the input space $\mathcal{X}$?

Q: For this example, how big is the entire input space $\mathcal{X}$?

Q: For this example, how big is the entire Boolean hypothesis set $\mathcal{H}$?

See Boolean_Learning_Example.pdf
Boolean_Learning_Example.ipynb

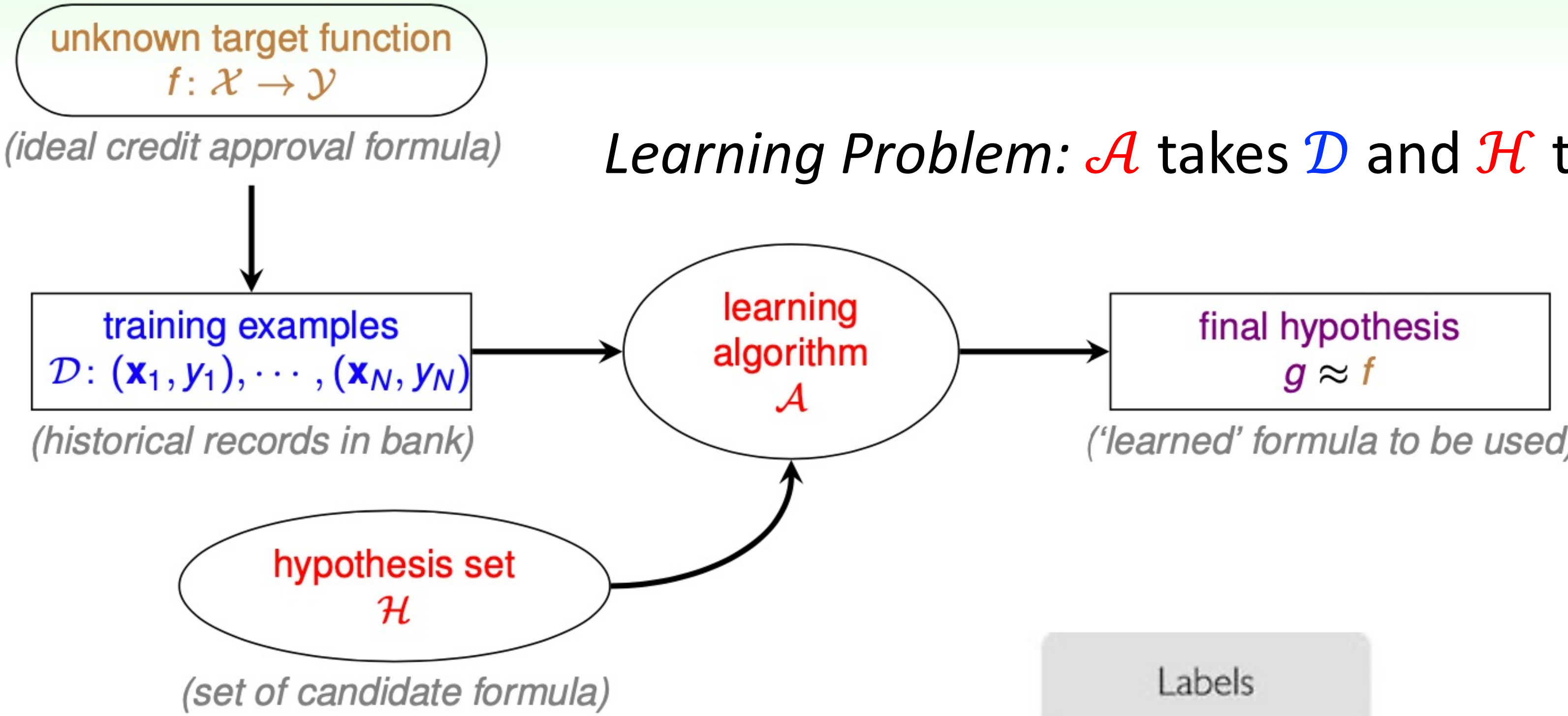
https://www.sli.do/
#073374

Summary

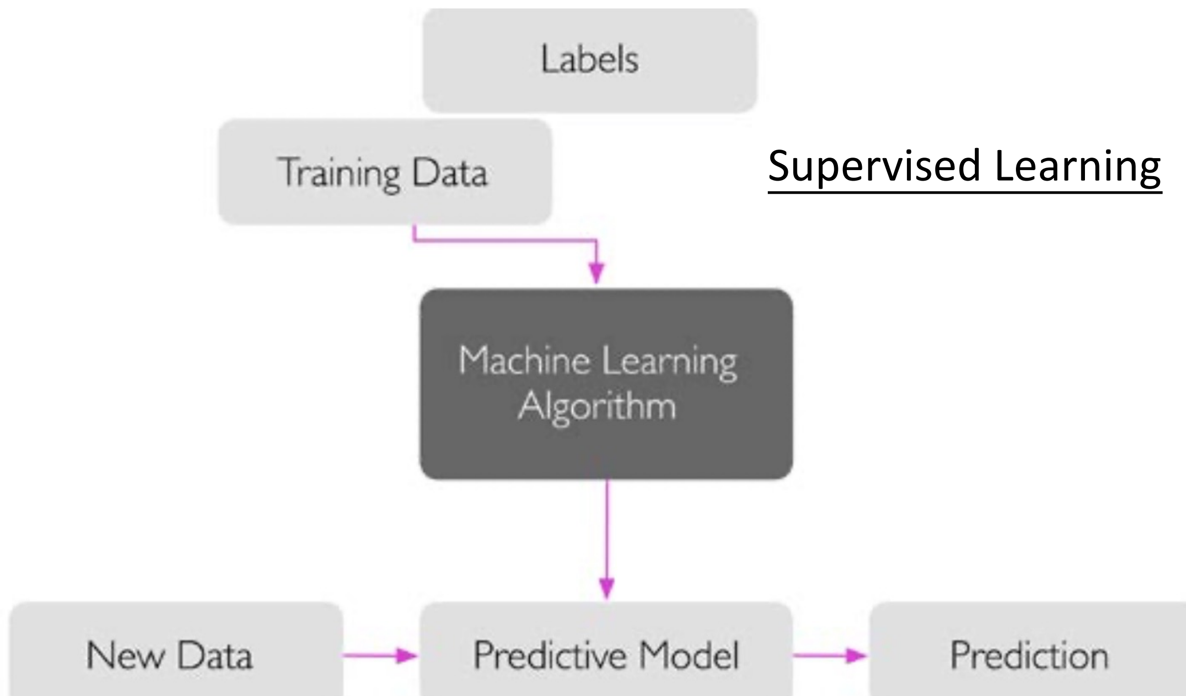
Is learning feasible?

- Learning is only feasible in a *probabilistic* way and we can **predict** something useful outside the training set $\mathcal{D}$ using only $\mathcal{D}$.
- We don't insist on using any particular probability distribution, or even on knowing what distribution is used. However, whatever distribution we use for generating the samples, we must also use when we evaluate how well $\boldsymbol{g}$ approximates the *unknown* target function $\boldsymbol{f}$.
- The hypothesis $\boldsymbol{g}$ is not fixed ahead of time before generating the data, because which hypothesis is selected to be $\boldsymbol{g}$ depends on the data.

Learning Problem: $\mathcal{A}$ takes $\mathcal{D}$ and $\mathcal{H}$ to get g



<u>Supervised Learning</u>



Summary

Learning? What do we mean?

Is learning feasible?

- **Machine learning: use data to compute hypothesis *g* that approximate unknown target *f*.**
- **In practice, learning algorithm $\mathcal{A}$ takes training examples $\mathcal{D}$ and hypothesis set $\mathcal{H}$ to get final hypothesis *g*.**
- **Learning is only feasible in a *probabilistic* way and we can predict something useful outside the training set $\mathcal{D}$ using only $\mathcal{D}$.**