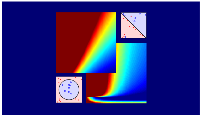
Machine Learning Foundations
(機器學習基石)



Lecture 1: The Learning Problem

Hsuan-Tien Lin (林軒田)

htlin@csie.ntu.edu.tw

Department of Computer Science
& Information Engineering

National Taiwan University
(國立台灣大學資訊工程系)

Course Design (1/2)

Machine Learning: a mixture of theoretical and practical tools

- theory oriented
  - derive everything **deeply** for solid understanding
  - less interesting to general audience
- technique**s** oriented
  - flash over the sexiest techniques **broadly** for shiny coverage
  - too many techniques, hard to choose, hard to use properly

our approach: **foundation oriented**

Course Design (2/2)

Foundation Oriented ML Course

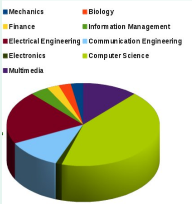
- mixture of philosophical illustrations, key theory, core techniques, usage in practice, and hopefully jokes **:-)**
  —what **every machine learning user** should know
- story-like:
  - **When** Can Machines Learn? (illustrative + technical)
  - **Why** Can Machines Learn? (theoretical + illustrative)
  - **How** Can Machines Learn? (technical + practical)
  - How Can Machines Learn **Better**? (practical + theoretical)

allows students to **learn 'future/untaught' techniques or study deeper theory easily**

Course History

NTU Version

- 15-17 weeks (2+ hours)
- highly-praised with **English and blackboard teaching**



Coursera Version

- 8 weeks of 'foundation' (**this course**) + 7 weeks of 'techniques' (coming course)
- **Mandarin teaching** to reach more audience in need
- **slides teaching** improved with Coursera's quiz and homework mechanisms

goal: **try** making Coursera version even better than NTU version

Roadmap

1. **When** Can Machines Learn?

   Lecture 1: The Learning Problem
   - Course Introduction
   - What is Machine Learning
   - Applications of Machine Learning
   - Components of Machine Learning
   - Machine Learning and Other Fields

2. Why Can Machines Learn?
3. How Can Machines Learn?
4. How Can Machines Learn Better?

From Learning to Machine Learning

learning: acquiring skill
with experience accumulated from observations

observations $\longrightarrow$ $\boxed{\text{learning}}$ $\longrightarrow$ skill

machine learning: acquiring skill
with experience accumulated/**computed** from data

data $\longrightarrow$ $\boxed{\text{ML}}$ $\longrightarrow$ skill

What is skill?

A More Concrete Definition

skill
$\Leftrightarrow$ improve some performance measure (e.g. prediction accuracy)

machine learning: improving some performance measure with experience **computed** from data

data $\longrightarrow$ $\boxed{\text{ML}}$ $\longrightarrow$ improved performance measure

An Application in Computational Finance

stock data $\longrightarrow$ $\boxed{\text{ML}}$ $\longrightarrow$ more investment gain

Why use machine learning?

Yet Another Application: Tree Recognition

- 'define' trees and hand-program: **difficult**
- learn from data (observations) and recognize: a **3-year-old can do so**
- 'ML-based tree recognition system' can be **easier to build** than hand-programmed system

ML: an **alternative route** to build complicated systems

The Machine Learning Route

ML: an **alternative route** to build complicated systems

Some Use Scenarios

- when human cannot program the system manually
  —navigating on Mars
- when human cannot 'define the solution' easily
  —speech/visual recognition
- when needing rapid decisions that humans cannot do
  —high-frequency trading
- when needing to be user-oriented in a massive scale
  —consumer-targeted marketing

Give a **computer** a fish, you feed it for a day;
teach it how to fish, you feed it for a lifetime. **:-)**

Key Essence of Machine Learning

machine learning: improving some performance measure
with experience **computed** from data

data ⟶ ML ⟶ improved performance measure

1. exists some 'underlying pattern' to be learned
—so 'performance measure' can be improved
2. but no programmable (easy) definition
—so 'ML' is needed
3. somehow there is data about the pattern
—so ML has some 'inputs' to learn from

key essence: help decide whether to use ML

Fun Time

Which of the following is best suited for machine learning?

1. predicting whether the next cry of the baby girl happens at an even-numbered minute or not
2. determining whether a given graph contains a cycle
3. deciding whether to approve credit card to some customer
4. guessing whether the earth will be destroyed by the misuse of nuclear power in the next ten years

Fun Time

Which of the following is best suited for machine learning?

1. predicting whether the next cry of the baby girl happens at an even-numbered minute or not
2. determining whether a given graph contains a cycle
3. deciding whether to approve credit card to some customer
4. guessing whether the earth will be destroyed by the misuse of nuclear power in the next ten years

Reference Answer: 3

1. no pattern
2. programmable definition
3. pattern: customer behavior;
   definition: not easily programmable;
   data: history of bank operation
4. arguably no (or not enough) data yet

Daily Needs: Food, Clothing, Housing, Transportation

data ⟶ ML ⟶ skill

1. Food (Sadilek et al., 2013)
   - data: Twitter data (words + location)
   - skill: tell food poisoning likeliness of restaurant properly
2. Clothing (Abu-Mostafa, 2012)
   - data: sales figures + client surveys
   - skill: give good fashion recommendations to clients
3. Housing (Tsanas and Xifara, 2012)
   - data: characteristics of buildings and their energy load
   - skill: predict energy load of other buildings closely
4. Transportation (Stallkamp et al., 2012)
   - data: some traffic sign images and meanings
   - skill: recognize traffic signs accurately

ML is everywhere!

Education

data $\longrightarrow$ $\boxed{\text{ML}}$ $\longrightarrow$ skill

- data: students' records on quizzes on a Math tutoring system
- skill: predict whether a student can give a correct answer to another quiz question

A Possible ML Solution

answer correctly $\approx$ $[\![$recent strength of student $>$ difficulty of question$]\!]$

- give ML 9 million records from 3000 students
- ML determines (**reverse-engineers**) strength and difficulty automatically

key part of the **world-champion** system from National Taiwan Univ. in KDDCup 2010

Entertainment: Recommender System (1/2)

data $\longrightarrow$ ML $\longrightarrow$ skill

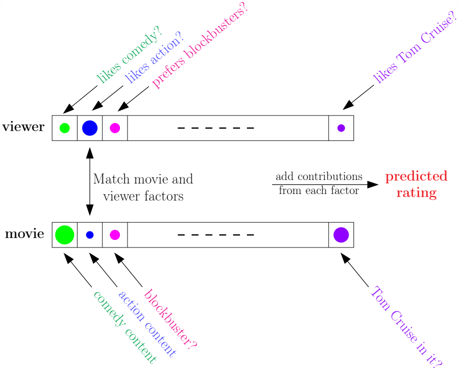
- data: how many users have rated some movies
- skill: predict how a user would rate an unrated movie

A Hot Problem

- competition held by Netflix in 2006
  - 100,480,507 ratings that 480,189 users gave to 17,770 movies
  - 10% improvement = **1 million dollar prize**
- similar competition (movies $\rightarrow$ songs) held by Yahoo! in KDDCup 2011
  - 252,800,275 ratings that 1,000,990 users gave to 624,961 songs

How can machines **learn our preferences**?

Entertainment: Recommender System (2/2)



A Possible ML Solution

- pattern:
  rating ← viewer/movie factors
- learning:
  known rating
  → learned factors
  → unknown rating prediction

key part of the **world-champion** (again!)
system from National Taiwan Univ.
in KDDCup 2011

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?'

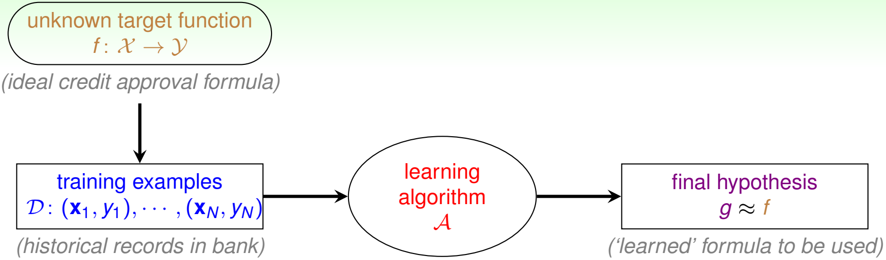
Formalize the Learning Problem

Basic Notations

- input: $\mathbf{x} \in \mathcal{X}$ (customer application)
- output: $y \in \mathcal{Y}$ (good/bad after approving credit card)
- unknown pattern to be learned $\Leftrightarrow$ target function: $f\colon \mathcal{X} \to \mathcal{Y}$ (ideal credit approval formula)
- data $\Leftrightarrow$ training examples: $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)\}$ (historical records in bank)
- hypothesis $\Leftrightarrow$ skill with hopefully good performance: $g\colon \mathcal{X} \to \mathcal{Y}$ ('learned' formula to be used)

$$\{(\mathbf{x}_n, y_n)\} \text{ from } f \longrightarrow \boxed{\text{ML}} \longrightarrow g$$

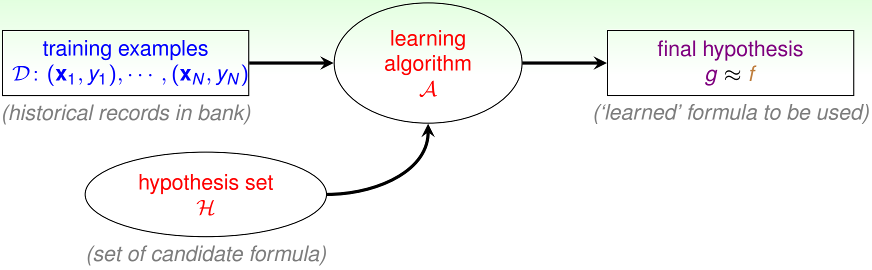
Learning Flow for Credit Approval



- 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?

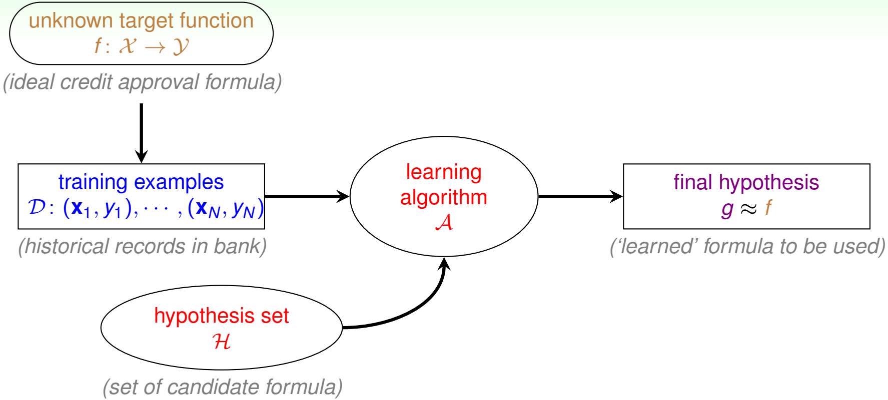
The Learning Model



- 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 $\leq$ 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



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

Machine Learning and Data Mining

Machine Learning

use data to compute hypothesis g that approximates target f

Data Mining

use **(huge)** data to **find property** that is interesting

- if 'interesting property' **same as** 'hypothesis that approximate target'
  —**ML = DM** (usually what KDDCup does)
- if 'interesting property' **related to** 'hypothesis that approximate target'
  —**DM can help ML, and vice versa** (often, but not always)
- traditional DM also focuses on **efficient computation in large database**

difficult to distinguish ML and DM in reality

Machine Learning and Artificial Intelligence

Machine Learning

use data to compute hypothesis g that approximates target f

Artificial Intelligence

compute **something that shows intelligent behavior**

- $g \approx f$ is something that shows intelligent behavior —**ML can realize AI**, among other routes
- e.g. chess playing
  - traditional AI: game tree
  - ML for AI: 'learning from board data'

ML is one possible route to realize AI

Machine Learning and Statistics

Machine Learning

use data to compute hypothesis g that approximates target f

Statistics

use data to **make inference about an unknown process**

- g is an inference outcome; f is something unknown —statistics **can be used to achieve ML**
- traditional statistics also focus on **provable results with math assumptions**, and care less about computation

statistics: many useful tools for ML

Summary

1. **When** Can Machines Learn?

Lecture 1: The Learning Problem

- Course Introduction
  foundation oriented and story-like
- What is Machine Learning
  use data to approximate target
- Applications of Machine Learning
  almost everywhere
- Components of Machine Learning
  $\mathcal{A}$ **takes** $\mathcal{D}$ **and** $\mathcal{H}$ **to get** g
- Machine Learning and Other Fields
  related to DM, AI and Stats

- **next: a simple and yet useful learning model (**$\mathcal{H}$ **and** $\mathcal{A}$**)**

2. Why Can Machines Learn?
3. How Can Machines Learn?
4. How Can Machines Learn Better?