Lecture 3: Types of Learning

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)
Types of Learning

Roadmap

1. **When Can Machines Learn?**

   **Lecture 2: Learning to Answer Yes/No**
   
   PLA $A$ takes **linear separable** $\mathcal{D}$ and perceptrons $\mathcal{H}$ to get hypothesis $g$

2. **Why Can Machines Learn?**

3. **How Can Machines Learn?**

4. **How Can Machines Learn Better?**

   **Lecture 3: Types of Learning**
   
   - Learning with Different Output Space $\mathcal{Y}$
   - Learning with Different Data Label $y_n$
   - Learning with Different Protocol $f \Rightarrow (x_n, y_n)$
   - Learning with Different Input Space $\mathcal{X}$
Credit Approval Problem Revisited

<table>
<thead>
<tr>
<th>age</th>
<th>23 years</th>
</tr>
</thead>
<tbody>
<tr>
<td>gender</td>
<td>female</td>
</tr>
<tr>
<td>annual salary</td>
<td>NTD 1,000,000</td>
</tr>
<tr>
<td>year in residence</td>
<td>1 year</td>
</tr>
<tr>
<td>year in job</td>
<td>0.5 year</td>
</tr>
<tr>
<td>current debt</td>
<td>200,000</td>
</tr>
</tbody>
</table>

credit? \{no(-1), yes(+1)\}

unknown target function \(f: \mathcal{X} \rightarrow \mathcal{Y}\)

(ideal credit approval formula)

training examples \(\mathcal{D}: (x_1, y_1), \ldots, (x_N, y_N)\)

(historical records in bank)

learning algorithm \(A\)

final hypothesis \(g \approx f\)

('learned' formula to be used)

hypothesis set \(\mathcal{H}\)

(set of candidate formula)

\(\mathcal{Y} = \{-1, +1\}: \text{binary classification}\)
More Binary Classification Problems

- credit approve/disapprove
- email spam/non-spam
- patient sick/not sick
- ad profitable/not profitable
- answer correct/incorrect (KDDCup 2010)

core and important problem with many tools as building block of other tools
### Multiclass Classification: Coin Recognition Problem

- classify US coins (1c, 5c, 10c, 25c) by (size, mass)
- \( \mathcal{Y} = \{1c, 5c, 10c, 25c\} \), or \( \mathcal{Y} = \{1, 2, \cdots, K\} \) (abstractly)
- binary classification: special case with \( K = 2 \)

### Other Multiclass Classification Problems

- written digits \( \Rightarrow 0, 1, \cdots, 9 \)
- pictures \( \Rightarrow \) apple, orange, strawberry
- emails \( \Rightarrow \) spam, primary, social, promotion, update (Google)

**many applications** in practice, especially for ‘recognition’
Regressio: Patient Recovery Prediction Problem

- **binary classification**: patient features $\Rightarrow$ sick or not
- **multiclass classification**: patient features $\Rightarrow$ which type of cancer
- **regression**: patient features $\Rightarrow$ how many days before recovery
- $\mathcal{Y} = \mathbb{R}$ or $\mathcal{Y} = \text{[lower, upper]} \subset \mathbb{R}$ (bounded regression)
  — deeply studied in statistics

**Other Regression Problems**

- company data $\Rightarrow$ stock price
- climate data $\Rightarrow$ temperature

also core and important with many ‘statistical’ tools as building block of other tools
Structured Learning: Sequence Tagging Problem

- multiclass classification: word $\Rightarrow$ word class
- structured learning: sentence $\Rightarrow$ structure (class of each word)
- $\mathcal{Y} = \{PVN, PVP, NVN, PV, \cdots\}$, not including $VVVVV$
- huge multiclass classification problem (structure $\equiv$ hyperclass) without ‘explicit’ class definition

Other Structured Learning Problems

- protein data $\Rightarrow$ protein folding
- speech data $\Rightarrow$ speech parse tree

a fancy but complicated learning problem
Learning with Different Output Space $\mathcal{Y}$

- **binary classification**: $\mathcal{Y} = \{-1, +1\}$
- **multiclass classification**: $\mathcal{Y} = \{1, 2, \ldots, K\}$
- **regression**: $\mathcal{Y} = \mathbb{R}$
- **structured learning**: $\mathcal{Y} = \text{structures}$
- . . . and a lot more!!

unknown target function $f : \mathcal{X} \rightarrow \mathcal{Y}$

training examples $\mathcal{D} : (x_1, y_1), \ldots, (x_N, y_N)$

learning algorithm $A$

final hypothesis $g \approx f$

core tools: binary classification and regression
Fun Time

What is this learning problem?

The entrance system of the school gym, which does automatic face recognition based on machine learning, is built to charge four different groups of users differently: Staff, Student, Professor, Other. What type of learning problem best fits the need of the system?

1. binary classification
2. multiclass classification
3. regression
4. structured learning

Reference Answer: 2

There is an ‘explicit’ \( \mathcal{Y} \) that contains four classes.
Supervised: Coin Recognition Revisited

unknown target function $f: \mathcal{X} \rightarrow \mathcal{Y}$

training examples $D: (x_1, y_1), \cdots, (x_N, y_N)$

learning algorithm $A$

final hypothesis $g \approx f$

supervised learning:

every $x_n$ comes with corresponding $y_n$
Unsupervised: Coin Recognition without $y_n$

supervised multiclass classification

unsupervised multiclass classification $\iff$ ‘clustering’

Other Clustering Problems

- articles $\Rightarrow$ topics
- consumer profiles $\Rightarrow$ consumer groups

clustering: a challenging but useful problem
Unsupervised: Coin Recognition without $y_n$

supervised multiclass classification

unsupervised multiclass classification ⇐⇒ ‘clustering’

Other Clustering Problems

- articles ⇒ topics
- consumer profiles ⇒ consumer groups

*clustering*: a challenging but useful problem
Unsupervised: Learning without $y_n$

Other Unsupervised Learning Problems

- **clustering**: $\{x_n\} \Rightarrow \text{cluster}(x)$
  ($\approx$ ‘unsupervised multiclass classification’)
  —i.e. articles $\Rightarrow$ topics

- **density estimation**: $\{x_n\} \Rightarrow \text{density}(x)$
  ($\approx$ ‘unsupervised bounded regression’)
  —i.e. traffic reports with location $\Rightarrow$ dangerous areas

- **outlier detection**: $\{x_n\} \Rightarrow \text{unusual}(x)$
  ($\approx$ extreme ‘unsupervised binary classification’)
  —i.e. Internet logs $\Rightarrow$ intrusion alert

- ... and a lot more!!

**unsupervised learning**: diverse, with possibly very different performance goals
Semi-supervised: Coin Recognition with Some $y_n$

Types of Learning

Learning with Different Data Label $y_n$

- Semi-supervised: Coin Recognition with Some $y_n$

Other Semi-supervised Learning Problems

- face images with a few labeled $\Rightarrow$ face identifier (Facebook)
- medicine data with a few labeled $\Rightarrow$ medicine effect predictor

**semi-supervised learning**: leverage unlabeled data to avoid ‘expensive’ labeling

Hsuan-Tien Lin (NTU CSIE)
Reinforcement Learning

a ‘very different’ but natural way of learning

Teach Your Dog: Say ‘Sit Down’

*The dog pees on the ground.*
BAD DOG. THAT’S A VERY WRONG ACTION.

- cannot easily show the dog that \( y_n = \text{sit} \) when \( x_n = \text{‘sit down’} \)
- but can ‘punish’ to say \( \tilde{y}_n = \text{pee is wrong} \)

Other Reinforcement Learning Problems Using \((x, \tilde{y}, \text{goodness})\)

- (customer, ad choice, ad click earning) \(\Rightarrow\) ad system
- (cards, strategy, winning amount) \(\Rightarrow\) black jack agent

reinforcement: learn with ‘partial/implicit information’ (often sequentially)
Reinforcement Learning

a ‘very different’ but natural way of learning

Teach Your Dog: Say ‘Sit Down’

The dog sits down.
Good Dog. Let me give you some cookies.

- still cannot show $y_n = \text{sit}$ when $x_n = \text{‘sit down’}$
- but can ‘reward’ to say $\tilde{y}_n = \text{sit}$ is good

Other Reinforcement Learning Problems Using $(x, \tilde{y}, \text{goodness})$

- (customer, ad choice, ad click earning) $\Rightarrow$ ad system
- (cards, strategy, winning amount) $\Rightarrow$ black jack agent

reinforcement: learn with ‘partial/implicit information’ (often sequentially)
Mini Summary

Learning with Different Data Label $y_n$

- **supervised**: all $y_n$
- unsupervised: no $y_n$
- semi-supervised: some $y_n$
- reinforcement: implicit $y_n$ by goodness($\tilde{y}_n$)
- ... and more!!

unknown target function $f: \mathcal{X} \rightarrow \mathcal{Y}$

training examples $\mathcal{D} : (x_1, y_1), \ldots, (x_N, y_N)$

learning algorithm $A$

final hypothesis $g \approx f$

hypothesis set $\mathcal{H}$

core tool: supervised learning
What is this learning problem?

To build a tree recognition system, a company decides to gather one million of pictures on the Internet. Then, it asks each of the 10 company members to view 100 pictures and record whether each picture contains a tree. The pictures and records are then fed to a learning algorithm to build the system. What type of learning problem does the algorithm need to solve?

1. supervised
2. unsupervised
3. semi-supervised
4. reinforcement

Reference Answer: 3

The 1,000 records are the labeled \((x_n, y_n)\); the other 999,000 pictures are the unlabeled \(x_n\).
batch supervised multiclass classification: learn from all known data
More Batch Learning Problems

- batch of (email, spam?) $\Rightarrow$ spam filter
- batch of (patient, cancer) $\Rightarrow$ cancer classifier
- batch of patient data $\Rightarrow$ group of patients

batch learning: a very common protocol
Online: Spam Filter that ‘Improves’

- **batch spam filter:** learn with known (email, spam?) pairs, and predict with fixed $g$
- **online spam filter, which sequentially:**
  1. observe an email $x_t$
  2. predict spam status with current $g_t(x_t)$
  3. receive ‘desired label’ $y_t$ from user, and then update $g_t$ with $(x_t, y_t)$

**Connection to What We Have Learned**
- PLA can be easily adapted to online protocol (how?)
- reinforcement learning is often done online (why?)

- online: hypothesis ‘improves’ through receiving data instances **sequentially**
Active Learning: Learning by ‘Asking’

- **Protocol ↔ Learning Philosophy**
  - batch: ‘duck feeding’
  - online: ‘passive sequential’
  - active: ‘question asking’ (sequentially)
    —query the $y_n$ of the chosen $x_n$

unknown target function $f : \mathcal{X} \to \mathcal{Y}$

- training examples $\mathcal{D} : (x_1, y_1), \ldots, (x_N, y_N)$

- hypothesis set $\mathcal{H}$

- active: improve hypothesis with fewer labels (hopefully) by asking questions *strategically*

final hypothesis $g \approx f$
Mini Summary

Learning with Different Protocol \( f \Rightarrow (x_n, y_n) \)

- **batch**: all known data
- **online**: sequential (passive) data
- **active**: strategically-observed data
- \ldots and more!!

unknown target function \( f : \mathcal{X} \rightarrow \mathcal{Y} \)

training examples \( \mathcal{D} : (x_1, y_1), \ldots, (x_N, y_N) \)

learning algorithm \( \mathcal{A} \)

final hypothesis \( g \approx f \)

hypothesis set \( \mathcal{H} \)

core protocol: batch
What is this learning problem?

A photographer has 100,000 pictures, each containing one baseball player. He wants to automatically categorize the pictures by its player inside. He starts by categorizing 1,000 pictures by himself, and then writes an algorithm that tries to categorize the other pictures if it is ‘confident’ on the category while pausing for (& learning from) human input if not. What protocol best describes the nature of the algorithm?

1. batch
2. online
3. active
4. random

Reference Answer: 3

The algorithm takes a active but naïve strategy: ask when ‘confused’. You should probably do the same when taking a class. :-(
Credit Approval Problem Revisited

unknown target function \( f : \mathcal{X} \rightarrow \mathcal{Y} \)  
(ideal credit approval formula)

training examples \( \mathcal{D} : (x_1, y_1), \cdots, (x_N, y_N) \)  
(historical records in bank)

learning algorithm \( \mathcal{A} \)

final hypothesis \( g \approx f \)  
(‘learned’ formula to be used)

hypothesis set \( \mathcal{H} \)  
(set of candidate formula)

concrete features: each dimension of \( \mathcal{X} \subseteq \mathbb{R}^d \) represents ‘sophisticated physical meaning’

<table>
<thead>
<tr>
<th>feature</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>23 years</td>
</tr>
<tr>
<td>gender</td>
<td>female</td>
</tr>
<tr>
<td>annual salary</td>
<td>NTD 1,000,000</td>
</tr>
<tr>
<td>year in residence</td>
<td>1 year</td>
</tr>
<tr>
<td>year in job</td>
<td>0.5 year</td>
</tr>
<tr>
<td>current debt</td>
<td>200,000</td>
</tr>
</tbody>
</table>
More on Concrete Features

- **(size, mass)** for coin classification
- **customer info** for credit approval
- **patient info** for cancer diagnosis
- often including ‘human intelligence’ on the learning task

concrete features: the ‘easy’ ones for ML
Raw Features: Digit Recognition Problem (1/2)

- digit recognition problem: features ⇒ meaning of digit
- a typical supervised multiclass classification problem
Raw Features: Digit Recognition Problem (2/2)

by Concrete Features

\[ \mathbf{x} = (\text{symmetry, density}) \]

by Raw Features

- 16 by 16 gray image \( \mathbf{x} \equiv (0, 0, 0.9, 0.6, \cdots) \in \mathbb{R}^{256} \)
- ‘simple physical meaning’; thus more difficult for ML than concrete features

Other Problems with Raw Features

- image pixels, speech signal, etc.

raw features: often need human or machines to convert to concrete ones
Abstract Features: Rating Prediction Problem

Rating Prediction Problem (KDDCup 2011)

- given previous (userid, itemid, rating) tuples, predict the rating that some userid would give to itemid?
- a regression problem with $\mathcal{Y} \subseteq \mathbb{R}$ as rating and $\mathcal{X} \subseteq \mathbb{N} \times \mathbb{N}$ as (userid, itemid)
- ‘no physical meaning’; thus even more difficult for ML

Other Problems with Abstract Features

- student ID in online tutoring system (KDDCup 2010)
- advertisement ID in online ad system

abstract: again need ‘feature conversion/extraction/construction’
Mini Summary

- **concrete**: sophisticated (and related) physical meaning
- **raw**: simple physical meaning
- **abstract**: no (or little) physical meaning
- ... and more!!

unknown target function

\[ f: \mathcal{X} \rightarrow \mathcal{Y} \]

training examples

\[ \mathcal{D}: (x_1, y_1), \ldots, (x_N, y_N) \]

learning algorithm

\[ \mathcal{A} \]

final hypothesis

\[ g \approx f \]

hypothesis set

\[ \mathcal{H} \]

‘easy’ input: concrete
What features can be used?

Consider a problem of building an online image advertisement system that shows the users the most relevant images. What features can you choose to use?

1. concrete
2. concrete, raw
3. concrete, abstract
4. concrete, raw, abstract

Reference Answer: 4

concrete user features, raw image features, and maybe abstract user/image IDs
Summary

1. When Can Machines Learn?

Lecture 2: Learning to Answer Yes/No

Lecture 3: Types of Learning

- Learning with Different Output Space $\mathcal{Y}$
  - [classification], [regression], [structured]
- Learning with Different Data Label $y_n$
  - [supervised], un/semi-supervised, reinforcement
- Learning with Different Protocol $f \Rightarrow (x_n, y_n)$
  - [batch], online, active
- Learning with Different Input Space $\mathcal{X}$
  - [concrete], raw, abstract

- next: learning is impossible?!

2. Why Can Machines Learn?

3. How Can Machines Learn?

4. How Can Machines Learn Better?