Machine Learning Overviews and Applications

Hsuan-Tien Lin (林軒田)
htlin@csie.ntu.edu.tw

Department of Computer Science & Information Engineering
National Taiwan University

NTU BIME Seminar Talk, 10/24/2019

materials mostly taken from my “Learning from Data” book, my “Machine Learning Foundations” free online course, and works from NTU CLLab and NTU KDDCup teams
About Me

Hsuan-Tien Lin

- Professor, Dept. of CSIE, National Taiwan University
- Leader of the Computational Learning Laboratory
- Co-author of the textbook “Learning from Data: A Short Course” (often ML best seller on Amazon)
- Instructor of the NTU-Coursera Mandarin-teaching ML Massive Open Online Courses
  - “Machine Learning Foundations”
  - “Machine Learning Techniques”
What is Machine Learning
What is Machine Learning

From Learning to Machine Learning

Learning: acquiring skill
with experience accumulated from observations

Observations → Learning → Skill

Machine Learning: acquiring skill
with experience accumulated/computed from data

Data → ML → Skill

What is skill?
What is Machine Learning

A More Concrete Definition

skill
⇔ improve some performance measure (e.g. prediction accuracy)

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

data → ML → improved performance measure

An Application in Computational Finance

stock data → ML → more investment gain

Why use machine learning?
• ‘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 (AI) systems
What is Machine Learning Skill ⇔ Artificial Intelligence

From Big Data to Artificial Intelligence

big data → ML → artificial intelligence

ingredient → tools/steps → dish

“cooking” needs many possible tools & procedures

(Hotos Licensed under CC BY 2.0 from Andrea Goh on Flickr)
What is Machine Learning

Key Essence of Machine Learning

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

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
Snapshot Applications of Machine Learning
Communication

for 4G LTE communication

- data:
  - channel information (the channel matrix representing mutual information)
  - configuration (precoding, modulation, etc.) that reaches the highest throughput
- skill: predict best configuration to the base station in a new environment

previous work of my student Yi-An Lin as intern @ MTK
for cross-screen ad placement

- **data:**
  - customer information
  - device information
  - ad information

- **skill:** predict best ad to show to the user across devices so that she/he clicks

ongoing work of my collaboration with Appier
Daily Needs: Food, Clothing, Housing, Transportation

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**: 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**: 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 → 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 Machine Learning
## Components of Learning: Metaphor Using Credit Approval

### Applicant Information

<table>
<thead>
<tr>
<th>Component</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>

**unknown** pattern to be learned:

‘approve credit card good for bank?’
Formalize the Learning Problem

Basic Notations

- **input**: $x \in \mathcal{X}$ (customer application)
- **output**: $y \in \mathcal{Y}$ (good/bad after approving credit card)
- **unknown pattern to be learned** $\iff$ **target function**: $f : \mathcal{X} \rightarrow \mathcal{Y}$ (ideal credit approval formula)
- **data** $\iff$ **training examples**: $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$ (historical records in bank)
- **hypothesis** $\iff$ **skill** with hopefully **good performance**: $g : \mathcal{X} \rightarrow \mathcal{Y}$ (‘learned’ formula to be used)
Components of Machine Learning

Learning Flow for Credit Approval

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

training examples
\[ D : (x_1, y_1), \ldots, (x_N, y_N) \]
(historical records in bank)

learning algorithm \( A \)

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

- 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?
Components of Machine Learning

The Learning Model

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

- 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 $\mathcal{H}$
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 \( \mathcal{A} \)

Final hypothesis \( g \approx f \)

('learned' formula to be used)

Hypothesis set \( \mathcal{H} \)

(set of candidate formula)

**Machine learning:**

Use data to compute hypothesis \( g \)

That approximates target \( f \)
Machine Learning Research in CLLab
Oracle: truth $f(x) + \text{noise } e(x)$

\[
\begin{align*}
\text{data (instance } x_n, \text{ label } y_n) & \quad \text{learning algorithm} \\
\text{good learning system } g(x) & \quad \text{learning model } \{h(x)\}
\end{align*}
\]

CLLab Works: Loosen the Limits of ML

1. cost-sensitive classification: limited protocol (classification) + auxiliary info. (cost)
2. multi-label classification: limited protocol (classification) + structure info. (label relation)
3. active learning: limited protocol (unlabeled data) + requested info. (query)
4. online learning: limited protocol (streaming data) + feedback info. (loss)

next: (1) cost-sensitive classification
Which Digit Did You Write?

1
one (1)

2
two (2)

3
three (3)

A **classification** problem—grouping "pictures" into different "categories"
Traditional Classification Problem

Oracle: truth \( f(x) + \text{noise } e(x) \)

- data (instance \( x_n \), label \( y_n \))

learning algorithm → good learning system \( g(x) \)

learning model \( \{g_\alpha(x)\} \)

1. input: a batch of examples (digit \( x_n \), intended label \( y_n \))
2. desired output: some \( g(x) \) such that \( g(x) \neq y \) seldom for future examples \( (x, y) \)
3. evaluation for some digit

\[
(x = 2, y = 2)
\]

\[
\neg g(x) = \begin{cases} 
1 & : \text{wrong} \\
2 & : \text{right} \\
3 & : \text{wrong}
\end{cases}
\]

Are all the \textbf{wrongs} equally bad?
What is the Status of the Patient?

H1N1-infected  cold-infected  healthy

another classification problem — grouping “patients” into different “status”
Patient Status Prediction

error measure = society cost

<table>
<thead>
<tr>
<th>actual</th>
<th>predicted</th>
<th>H1N1</th>
<th>cold</th>
<th>healthy</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1N1</td>
<td>0</td>
<td>1000</td>
<td>100000</td>
<td></td>
</tr>
<tr>
<td>cold</td>
<td>100</td>
<td>0</td>
<td>3000</td>
<td></td>
</tr>
<tr>
<td>healthy</td>
<td>100</td>
<td>30</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

- H1N1 mis-predicted as healthy: very high cost
- cold mis-predicted as healthy: high cost
- cold correctly predicted as cold: no cost

human doctors consider costs of decision; can computer-aided diagnosis do the same?
Our Contributions

<table>
<thead>
<tr>
<th></th>
<th>binary</th>
<th>multiclass</th>
</tr>
</thead>
<tbody>
<tr>
<td>regular</td>
<td>well-studied</td>
<td>well-studied</td>
</tr>
<tr>
<td>cost-sensitive</td>
<td>known (Zadrozny, 2003)</td>
<td>ongoing (our works)</td>
</tr>
</tbody>
</table>

theoretic, algorithmic and empirical studies of cost-sensitive classification

- ICML 2010: a theoretically-supported algorithm with superior experimental results
- BIBM 2011: application to real-world bacteria classification with promising experimental results
- etc.
More on KDDCup
What is KDDCup?

Background

- an annual competition on KDD (knowledge discovery and data mining)
- organized by ACM SIGKDD, starting from 1997, now the most prestigious data mining competition
- usually lasts 3-4 months
- participants include famous research labs (IBM, AT&T) and top universities (Stanford, Berkeley)

Aim

- bridge the gap between theory and practice, such as
  - scalability and efficiency
  - missing data and noise
  - heterogeneous data
  - unbalanced data
- define the state-of-the-art
More on KDDCup

KDDCups: 2008 to 2015 (1/4)

2008
- organizer: Siemens
- topic: breast cancer prediction (medical)
- data size: 0.2M
- teams: > 200
- NTU: **co-champion** with IBM

2009
- organizer: Orange
- topic: customer behavior prediction (business)
- data size: 0.1M
- teams: > 400
- NTU: **3rd place** of slow track
### KDDCups: 2008 to 2015 (2/4)

#### 2010
- **organizer:** PSLC Data Shop
- **topic:** student performance prediction (education)
- **data size:** 30M
- **teams:** > 100
- **NTU:** champion and student-team champion

#### 2011
- **organizer:** Yahoo!
- **topic:** music preference prediction (recommendation)
- **data size:** 300M
- **teams:** > 1000
- **NTU:** double champions
### 2012
- **organizer**: Tencent
- **topic**: web user behavior prediction (Internet)
- **data size**: 150M
- **teams**: > 800
- **NTU**: champion of track 2

### 2013
- **organizer**: Microsoft Research
- **topic**: paper-author relationship prediction (academia)
- **data size**: 600M
- **teams**: > 500
- **NTU**: double champions
KDDCups: 2008 to 2015 (4/4)

### 2014
- **organizer:** DonorsChoose
- **topic:** charity proposal recommendation (social work)
- **data size:** 850M
- **teams:** > 450
- **NTU:** top 20

### 2015
- **organizer:** XuetangX
- **topic:** dropout student prediction (online education)
- **data size:** 100M
- **teams:** > 800
- **NTU:** 4th place
Our Systematic Steps in KDDCups

1. **data analysis** (on part of data)
   - calculate **statistics** to identify outliers
   - visualize data to see trend/pattern

2. **feature extraction**
   - feature **design by human**: common encoding, domain knowledge, etc.
   - feature **learning by machines**: sparse coding, matrix factorization, deep learning, etc.

3. **model learning**
   - model **exploration** (trial-and-evaluate) to improve performance
   - model **selection** to avoid overfitting

4. **hypotheses blending** (towards **big ensemble**)
   - careful non-linear blending to be sophisticated
   - careful linear blending (voting/averaging) to be robust

---

can **follow those step for your applications**, except for maybe “big ensemble”!
That's about all. Thank you!