Active Learning by Learning

Hsuan-Tien Lin
htlin@csie.ntu.edu.tw

National Taiwan University

Institute of Information Systems and Applications,
National Tsing Hua University,
March 23, 2022

joint work with Wei-Ning Hsu (AAAI 2015)
About Me

Professor
National Taiwan University

Chief Data Science Consultant
(former Chief Data Scientist)
Appier Inc.

Co-author
Learning from Data

Instructor
NTU-Coursera Mandarin MOOCs
ML Foundations/Techniques
Active Learning

Apple Recognition Problem

Note: Slide Taken from my “ML Techniques” MOOC

- need **apple classifier**: is this a picture of an apple?
- gather photos under CC-BY-2.0 license on Flicker (thanks to the authors below!) and label them as apple/other for learning

(APAL stands for Apple and Pear Australia Ltd)

- Dan Foy
  - https://flic.kr/p/jNQ55
- APAL
  - https://flic.kr/p/jzP1VB
- adrianbartel
  - https://flic.kr/p/03h2hZ
- ANdrzej cH.
  - https://flic.kr/p/51DK8
- Stuart Webster
  - https://flic.kr/p/9C3Ybd
- nachans
  - https://flic.kr/p/9XD7Ag
- APAL
  - https://flic.kr/p/jzRe4u
- Jo Jakeman
  - https://flic.kr/p/7jwGp
- APAL
  - https://flic.kr/p/jzPYNn
- APAL
  - https://flic.kr/p/jzScif
Active Learning

Apple Recognition Problem

Note: Slide Taken from my “ML Techniques” MOOC

- need **apple classifier**: is this a picture of an apple?
- gather photos under CC-BY-2.0 license on Flicker (thanks to the authors below!) and **label them as apple/other** for learning

- Mr. Roboto
  - https://flic.kr/p/i5BN85
- Richard North
  - https://flic.kr/p/bHhPkB
- Richard North
  - https://flic.kr/p/d8tGou
- Emilian Vicol
  - https://flic.kr/p/bpmGXW
- Robert Mc-Queen
  - https://flic.kr/p/pZv1Mf
- Crystal
  - https://flic.kr/p/kaPYp
- jfh686
  - https://flic.kr/p/6vjRFH
- skyseeker
  - https://flic.kr/p/2MynV
- Janet Hudson
  - https://flic.kr/p/7QDBbm
- Rennett Stowe
  - https://flic.kr/p/agmnrk

Hsuan-Tien Lin (NTU)
Batch (Traditional) Machine Learning

Note: Flow Taken from my “ML Foundations” MOOC

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

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

(\( \text{apple}, +1 \)), (\( \text{apple}, +1 \)), (\( \text{banana}, +1 \))

(\( \text{banana}, -1 \)), (\( \text{pineapple}, -1 \)), (\( \text{orange}, -1 \))

learning algorithm \( \mathcal{A} \)

final hypothesis \( g \approx f \)

hypothesis set \( \mathcal{H} \)

batch supervised classification: learn from fully labeled data

Hsuan-Tien Lin (NTU)
Active Learning: Learning by ‘Asking’

but labeling is expensive

Protocol ⇔ Learning Philosophy

• batch: ‘duck feeding’
• active: ‘question asking’ (iteratively)
  —query $y_n$ of chosen $x_n$

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

labeled training examples

(\text{apple}, +1), (\text{apple}, +1), (\text{orange}, +1)

(banana, -1), (\text{peach}, -1), (\text{grapes}, -1)

unlabeled training examples

(a), (b), (c), (d)

active: improve hypothesis with fewer labels (hopefully) by asking questions strategically
Pool-Based Active Learning Problem

Given

- labeled pool $\mathcal{D}_l = \left\{ (\text{feature } \mathbf{x}_n, \text{label } y_n \text{ (e.g. IsApple?)}) \right\}_{n=1}^{N}$
- unlabeled pool $\mathcal{D}_u = \left\{ \tilde{\mathbf{x}}_s \right\}_{s=1}^{S}$

Goal

design an algorithm that iteratively

1. strategically query some $\tilde{\mathbf{x}}_s$ to get associated $\tilde{y}_s$
2. move $(\tilde{\mathbf{x}}_s, \tilde{y}_s)$ from $\mathcal{D}_u$ to $\mathcal{D}_l$
3. learn classifier $g^{(t)}$ from $\mathcal{D}_l$

and improve test accuracy of $g^{(t)}$ w.r.t #queries

how to query strategically?
How to Query Strategically?

by DFID - UK Department for International Development;
licensed under CC BY-SA 2.0 via Wikimedia Commons

**Strategy 1**
ask *most confused* question

**Strategy 2**
ask *most frequent* question

**Strategy 3**
ask *most helpful* question

do you use a **fixed strategy** in practice? 😊
Active Learning

Choice of Strategy

Strategy 1: uncertainty
ask most confused question

Strategy 2: representative
ask most frequent question

Strategy 3: exp.-err. reduction
ask most helpful question

• choosing one single strategy is non-trivial:

• human-designed strategy heuristic and confine machine’s ability

 can we free the machine 😊
by letting it learn to choose the strategies?

Hsuan-Tien Lin (NTU)
Our Contributions

a philosophical and algorithmic study of active learning, which ...

- allows machine to make intelligent choice of strategies, just like my cute kids
- studies sound feedback scheme to tell machine about goodness of choice, just like what I do
- results in promising active learning performance, just like (hopefully) bright future of my kids 😊

will describe key philosophical ideas behind our proposed approach
Online Choice of Strategy

Idea: Trial-and-Reward Like Human

by DFID - UK Department for International Development;
licensed under CC BY-SA 2.0 via Wikimedia Commons

$K$ strategies:
$A_1, A_2, \ldots, A_K$

- try one strategy
- “goodness” of strategy as reward

two issues: try and reward
Online Choice of Strategy

Reduction to Bandit

when do humans trial-and-reward?

-gambling 😊

$K$ strategies:
$A_1, A_2, \ldots, A_K$

try one strategy

“goodness” of strategy as reward

$K$ bandit machines:
$B_1, B_2, \ldots, B_K$

try one bandit machine

“luckiness” of machine as reward

—will take one well-known probabilistic bandit learner (EXP4.P)

intelligent choice of strategy

$\implies$ intelligent choice of bandit machine
Online Choice of Strategy

Active Learning by Learning

\[ K \text{ strategies: } \mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_K \]

- try one strategy
- “goodness” of strategy as reward

**Given: \( K \) existing active learning strategies**

for \( t = 1, 2, \ldots, T \)

1. let EXP4.P decide strategy \( \mathcal{A}_k \) to try
2. query the \( \tilde{x}_s \) suggested by \( \mathcal{A}_k \), and compute \( g^{(t)} \)
3. evaluate goodness of \( g^{(t)} \) as reward of trial to update EXP4.P

only remaining problem: what reward?
Ideal Reward

Ideal reward after updating classifier $g^{(t)}$ by the query $(x_{nt}, y_{nt})$:

$$\text{accuracy } \frac{1}{M} \sum_{m=1}^{M} \left[ y_m = g^{(t)}(x_m) \right] \text{ on test set } \{(x_m, y_m)\}_{m=1}^{M}$$

- Test accuracy as reward:
  area under query-accuracy curve $\equiv$ cumulative reward

- Test accuracy infeasible in practice
  —Labeling expensive, remember? 😊

difficulty: approximate test accuracy on the fly
Design of Reward

Training Accuracy as Reward

\[
\text{test accuracy } \frac{1}{M} \sum_{m=1}^{M} \left[ y_m = g^{(t)}(x_m) \right] \quad \text{infeasible, naïve replacement:} \\
\text{accuracy } \frac{1}{t} \sum_{\tau=1}^{t} \left[ y_{n_{\tau}} = g^{(t)}(x_{n_{\tau}}) \right] \quad \text{on labeled pool} \quad \{(x_{n_{\tau}}, y_{n_{\tau}})\}_{\tau=1}^{t}
\]

- training accuracy as reward: training accuracy \( \approx \) test accuracy?
- not necessarily!!
  —for active learning strategy that asks easiest questions:
    - training accuracy high: \( x_{n_{\tau}} \) easy to label
    - test accuracy low: not enough information about harder instances

training accuracy: too biased to approximate test accuracy
Design of Reward

**Weighted Training Accuracy as Reward**

- Training accuracy $\frac{1}{t} \sum_{\tau=1}^{t} \left[ y_{n_\tau} = g^{(t)}(x_{n_\tau}) \right]$ biased, want **less-biased estimator**

- **non-uniform sampling theorem**: if $(x_{n_\tau}, y_{n_\tau})$ sampled with probability $p_\tau > 0$ from data set $\{(x_n, y_n)\}_{n=1}^{N}$ in iteration $\tau$,

  $$\text{weighted training accuracy} \approx \frac{1}{t} \sum_{\tau=1}^{t} \frac{1}{p_\tau} \left[ y_{n_\tau} = g(x_{n_\tau}) \right]$$

  $$\approx \frac{1}{N} \sum_{n=1}^{N} \left[ y_n = g(x_n) \right] \text{ in expectation}$$

- **with probabilistic query like EXP4.P**: weighted training accuracy $\approx$ test accuracy

**weighted training accuracy**: less biased approx. of test accuracy on the fly
Design of Reward

Active Learning by Learning (Hsu and Lin, 2015)

**Given: K existing active learning strategies**

for $t = 1, 2, \ldots, T$

1. let EXP4.P decide strategy $A_k$ to try
2. query the $\tilde{x}_s$ suggested by $A_k$, and compute $g^{(t)}$
3. evaluate weighted training accuracy of $g^{(t)}$ as reward of trial to update EXP4.P

**Other possible rewards?**
Design of Reward

Human-Designed Criterion as Reward

(Baram et al., 2004) COMB approach:

bandit + **balancedness** of $g(t)$ on unlabeled data as reward

- **why?** human criterion that matches classifier to **domain assumption**
- **but many** active learning applications are on **unbalanced data!** —assumption may be **unrealistic**

existing strategies: active learning **by acting**;
COMB: active learning **by acting**;
**ours:** active learning **by learning**
Experiments

Comparison with Single Strategies

- no single best strategy for every data set
  —choosing needed
- ALBL consistently matches the best
  —similar findings across other data sets

**ALBL**: effective in making intelligent choices
Experiments

Comparison with Other Bandit-Driven Algorithms

ALBL \approx \text{COMB}

\begin{itemize}
  \item ALBL > ALBL-Train generally
    \hspace{1cm}—importance-weighted mechanism needed for correcting biased training accuracy
  \item ALBL consistently comparable to or better than COMB
    \hspace{1cm}—learning performance more useful than human-criterion
\end{itemize}

ALBL: effective in utilizing performance
Summary

Active Learning by Learning

- based on bandit learning + less biased performance estimator as reward
- effective in making intelligent choices —comparable or superior to the best of existing strategies
- effective in utilizing learning performance —superior to human-criterion-based approach

open-source tool developed

https://github.com/ntucllab/libact

Wait! Discussion and more!
Discussion for Theoreticians

**weighted training accuracy**\[ \frac{1}{t} \sum_{\tau=1}^{t} \frac{1}{\rho_{\tau}} \left[ y_{n_{\tau}} = g^{(t)}(x_{n_{\tau}}) \right] \] as reward

- **is reward unbiased estimator** of test performance?
  - no for learned $g^{(t)}$ (yes for fixed $g$)

- **is reward fixed before playing?**
  - no because $g^{(t)}$ learned from $(x_{n_{t}}, y_{n_{t}})$

- **is reward independent of each other?**
  - no because past history all in reward

**ALBL: tools from theoreticians**
+ wild/unintended usage
Have We Made Active Learning More Realistic? (1/2)

Yes!

open-source tool **libact** developed (Yang, 2017)

https://github.com/ntucllab/libact

- including uncertainty, QUIRE, PSDS, . . ., and ALBL
- received > 500 stars and continuous issues

“**libact** is a Python package designed to make active learning easier for real-world users”
No!

- single-most raised issue: hard to install on Windows/Mac—because several strategies requires some C packages
- performance in a recent industry project:

  - uncertainty sampling often suffices
  - ALBL dragged down by bad strategy

“libact is a Python package designed to make active learning easier for real-world users”
Other Attempts for Realistic Active Learning

“learn” a strategy beforehand rather than on-the-fly?

- transfer active learning experience (Chu and Lin, 2016)
  —not easy to realize in open-source package

Other active learning tasks?

- NLP (Yuan et al., 2020)
- reinforcement learning (Chen et al., 2020)
- annotation cost-sensitive (Tsou and Lin, 2019)
- classification cost-sensitive (Huang and Lin, 2016)

many more needs to be satisfied: mini-batch, multi-label query, weak-label query, etc.
1. **scalability bottleneck** of artificial intelligence: choice of methods/models/parameter/…

2. think outside of the **math** box: ‘non-rigorous’ usage may be **good enough**

3. important to be **brave yet patient**
   - idea: 2012
   - paper: (Hsu and Lin, AAAI 2015);
   - software: (Yang et al., 2017)

4. easy-to-use in design ≠ **easy-to-use in reality**

**Thank you! Questions?**