Active Learning by Learning

Hsuan-Tien Lin, National Taiwan University
based on joint work with Wei-Ning Hsu, AAAI ’15

This Webinar is provided to you by
IEEE Computational Intelligence Society
https://cis.ieee.org
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
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  
flic.kr/p/jNQ55

APAL  
flic.kr/p/jzP1VB

adrianbartel  
flic.kr/p/bdy2hZ

ANdrzej ch.  
flic.kr/p/51DKA8

Stuart Webster  
flic.kr/p/9C3Ybd

nachans  
flic.kr/p/9XD7Ag

APAL  
flic.kr/p/jzRe4u

Jo Jakeman  
flic.kr/p/7jwtGp

APAL  
flic.kr/p/jzPYNr

APAL  
flic.kr/p/jzScif
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)

Mr. Roboto. flic.kr/p/i5BN85

Richard North flic.kr/p/bHhPkB

Richard North flic.kr/p/d8tGou

Emilian Robert Vicol flic.kr/p/bpmGXW

Nathaniel McQueen flic.kr/p/pZv1Mf

Crystal flic.kr/p/kaPYp

jf686 flic.kr/p/6vjRFH

skyseeker flic.kr/p/2MynV

Janet Hudson flic.kr/p/7QDBbm

Rennett Stowe flic.kr/p/agmnrk
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**

- $(\cdot, +1)$
- $(\cdot, +1)$
- $(\cdot, +1)$
- $(\cdot, -1)$
- $(\cdot, -1)$
- $(\cdot, -1)$

**unlabeled training examples**

- $(\cdot)$
- $(\cdot)$
- $(\cdot)$
- $(\cdot)$

**learning algorithm** $\mathcal{A}$

**final hypothesis** $g \approx f$

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

**Given**

- labeled pool $\mathcal{D}_l = \{(\text{feature } x_n, \text{label } y_n (\text{e.g. IsApple?}))\}^N_{n=1}$
- unlabeled pool $\mathcal{D}_u = \{\tilde{x}_s\}^S_{s=1}$

**Goal:** design an algorithm that iteratively

1. **strategically query** some $\tilde{x}_s$ to get associated $\tilde{y}_s$
2. move $(\tilde{x}_s, \tilde{y}_s)$ 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?

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? 😊
Choice of Strategy

Strategy 1: uncertainty
ask most confused question

Strategy 2: representative
ask most frequent question

Strategy 3: exp. error 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?
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) the **bright future** of my kids 😊

will describe **key philosophical ideas** behind our proposed approach
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

Active Learning by Learning Hsuan-Tien Lin
Reduction to Bandit

when do humans trial-and-reward?

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

intelligent choice of strategy \[\Rightarrow\] intelligent choice of bandit machine
Active Learning by Learning

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

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 $A_k$ to try
2. query the $\tilde{x}_s$ suggested by $A_k$, 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 $(x_{n_t}, y_{n_t})$:

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

• **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
Training Accuracy as Reward

\[
\frac{1}{M} \sum_{m=1}^{M} \left[ y_m = g^{(t)}(x_m) \right]
\]

infeasible, naïve replacement:

\[
\frac{1}{t} \sum_{\tau=1}^{t} \left[ y_{n\tau} = g^{(t)}(x_{n\tau}) \right]
\]
on labeled pool \( \{(x_{n\tau}, y_{n\tau})\}_{\tau=1}^{t} \)

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

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

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

• with **probabilistic query** like EXP4.P: 
  **weighted training accuracy** \(\approx\) **test accuracy**

**weighted training accuracy**: 
**less biased** approx. of test accuracy on the fly
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?
Human-Designed Criterion as Reward

(Baram et al., 2004) COMB approach:

\[ \text{bandit + balancedness of } g^{(t)} \text{ on unlabeled data as reward} \]

- 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**
Comparison with Single Strategies

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**
Comparison with Other Bandit-Driven Algorithms

- **ALBL \approx COMB**
  - diabetes

- **ALBL > COMB**
  - sonar

- **ALBL** > **ALBL-Train** generally
  - importance-weighted mechanism needed for correcting biased training accuracy
- **ALBL** consistently comparable to or better than **COMB**
  - learning performance more useful than human-criterion

**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](https://github.com/ntucllab/libact)

Wait! Discussion and more!
Discussion for Theoreticians

**weighted training accuracy** \( \frac{1}{t} \sum_{\tau=1}^{t} \frac{1}{p_{\tau}} \left[ y_{n_{\tau}} = g^{(t)}(x_{n_{\tau}}) \right] \) 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**
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”
Have We Made Active Learning More Realistic? (2/2)

No! (Not Completely!)

- 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. Easy-to-use in design $\neq$ **easy-to-use in reality**

Thank you! Questions?