Active Learning by Bandit Learning

Hsuan-Tien Lin (林軒田)
htlin@csie.ntu.edu.tw
Appier/National Taiwan University
(沛星互動科技/國立台灣大學資訊工程系)

A Symposium on Complex Data Analysis, May 26, 2017
joint works with Wei-Ning Hsu (AAAI 2015)
and Hong-Min Chu (ICDM 2017)
About Me
Hsuan-Tien Lin

- Chief Data Scientist, Appier
- Associate Professor, Dept. of CSIE, National Taiwan University
- Co-author of textbook “Learning from Data: A Short Course”
- Instructor of the NTU-Coursera Mandarin-teaching ML Massive Open Online Courses
  - “Machine Learning Foundations”:
    www.coursera.org/course/ntumlone
  - “Machine Learning Techniques”:
    www.coursera.org/course/ntumlttwo
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/6y2hZ

ANdrzej cH.
https://flic.kr/p/51DKA8

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/7jytGp

APAL
https://flic.kr/p/jzPYNv

APAL
https://flic.kr/p/jzScif

Hsuan-Tien Lin (Appier/NTU)
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.

Richard North

Richard North

Emilian Vicol

Robert Vicol

Nathaniel McQueen

https://flic.kr/p/i5BN85

https://flic.kr/p/bHhPkB

https://flic.kr/p/d8tGou

https://flic.kr/p/bpmGXW

https://flic.kr/p/pZv1Mf

Crystal

jfh686

skyseeker

Janet Hudson

Rennett Stowe

https://flic.kr/p/kaPYp

https://flic.kr/p/bHhPkB

https://flic.kr/p/6vjRFH

https://flic.kr/p/2MynV

https://flic.kr/p/7QDBbm

https://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 \( f : \mathcal{X} \to \mathcal{Y} \)

labeled training examples

\[
(\text{apple}, +1), (\text{orange}, +1), (\text{apple}, +1) \\
(\text{banana}, -1), (\text{orange}, -1), (\text{apple}, -1)
\]

learning algorithm \( \mathcal{A} \)

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

final hypothesis \( g \approx f \)
Pool-Based Active Learning Problem

Given

- labeled pool \( D_l = \{ (\text{feature } x_n, \text{label } y_n (\text{e.g. IsApple?})) \}_{n=1}^{N} \)
- unlabeled pool \( D_u = \{ \tilde{x}_s \}_{s=1}^{S} \)

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 \( D_u \) to \( D_l \)
3. learn classifier \( g^{(t)} \) from \( D_l \)

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?
Our Contributions

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

- allows machine to make **intelligent choice of strategies**, just like my cute daughter & son
- 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 daughter & son 😊

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
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**
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 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

Trainign Accuracy as Reward

\[ \text{test accuracy} \quad \frac{1}{M} \sum_{m=1}^{M} [y_m = g(t)(x_m)] \quad \text{infeasible, naïve replacement:} \]

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

- **training accuracy** as reward:
  \[ \text{training accuracy} \approx \text{test accuracy} \quad ? \]

- 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 as Reward

- **training accuracy** $\frac{1}{t} \sum_{\tau=1}^{t} [y_{n_\tau} = g^{(t)}(x_{n_\tau})]$ 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 $\tau$,

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

  $\approx \frac{1}{N} \sum_{n=1}^{N} [y_n = g(x_n)]$ 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**
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

**vehicle**

**sonar**

**diabetes**
Comparison with Other Adaptive Blending Algorithms

Experiments

- **ALBL** \(\approx\) **COMB**
- **ALBL** \(>\) **COMB**

### Graphs

- **diabetes**
- **sonar**

#### Key Points

- **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
Discussion

Mini 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 blending

New Directions

- **open-source tool** developed
  (https://github.com/ntucllab/libact)
- extending to **more sophisticated active learning problems**

Wait! Discussion and another interesting work!
Discussion for Statisticians

**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 statistics  
+ wild/unintended usage
Beyond Trial and Reward

Task 1

\(K\) strategies: \(A_1, A_2, \cdots, A_K\)

- try one strategy

“goodness” of strategy as reward

Task 2

\(K\) strategies: \(A_1, A_2, \cdots, A_K\)

- try one strategy

“goodness” of strategy as reward

Task 3

\(K\) strategies: \(A_1, A_2, \cdots, A_K\)

- try one strategy

“goodness” of strategy as reward

- always start fresh when trying new tasks
- always try one strategy

—do you do so as a human? 😊
Integrate-Reward-and-Transfer Like Human

Task 1

$K$ strategies: $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_K$

try integrated strategy

“goodness” of strategy as reward

Task 2

$K$ strategies: $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_K$

try integrated strategy

“goodness” of strategy as reward

Task 3

$K$ strategies: $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_K$

try integrated strategy

“goodness” of strategy as reward

more human-like:

- try integrated strategy
- transfer trial experience to other tasks
Cross-Data-Set Active Learning Problem

Given

- unlabeled $D_u^{(1)}$ and labeled pool
  $$D_i^{(1)} = \left\{ (\text{feature } x_n, \text{label } y_n \ (\text{e.g. IsApple?})) \right\}_{n=1}^N$$
- unlabeled $D_u^{(2)}$ and labeled pool
  $$D_i^{(2)} = \left\{ (\text{feature } x'_m, \text{label } y'_m \ (\text{e.g. IsOrange?})) \right\}_{m=1}^M$$
- unlabeled $D_u^{(3)}$ and labeled pool
  $$D_i^{(3)} = \left\{ (\text{feature } x''_p, \text{label } y''_p \ (\text{e.g. IsVerb?})) \right\}_{p=1}^P$$

Goal

exploit active learning experience from $\{(D_i^{(v)}, D_u^{(v)})\}_{v=1}^{V-1}$ to improve the active learning performance on $(D_i^{(V)}, D_u^{(V)})$

how to define experience?
Transfer Active Learning

Experience and Integration

- philosophical idea: experience $\equiv$ how to integrate
- EXP4.P in ALBL as probabilistic integration:
  randomly chosen strategy w.r.t. strategy choice probability

naïve experience transfer in ALBL:
initialize choice probability from prior tasks
Transfer Active Learning

Deterministic Instead of Probabilistic Integration

Active Learning

\[ K \] strategies, \( N_u \) unlabeled instances, maximize test performance

**ALBL**
- **action**: one of \( K \)
- **bandit solver**: EXP4.P
- **reward**: weighted accuracy w.r.t. prob.

**Linear Strategy Aggregation (LSA)**
- **action**: one of \( N_u \)
- **bandit solver**: LinUCB (UCB with ridge regression on contexts)
- **goal**: weighted accuracy \( \text{w.r.t. UCB} \)
- **context** \( z_{\ell,t} \) for bandit solver
  \( \iff \) strategy scores on each instance
- **weights** \( w_t \) from bandit solver
  \( \iff \) experience

next: Transfer LSA
Transfer LSA with Biased Regularization

- LSA on \((D^{(1)}_\ell, D^{(1)}_u)\) — with zero experience

\[
\mathbf{w}_t = \operatorname{argmin}_\mathbf{w} \lambda \|\mathbf{w} - \mathbf{0}\|^2 + \|Z_t\mathbf{w} - \mathbf{r}_t\|^2
\]

- T-LSA on \((D^{(V)}_\ell, D^{(V)}_u)\) — with previous experience

\[
\mathbf{w}_t = \operatorname{argmin}_\mathbf{w} \lambda \|\mathbf{w} - \mathbf{w}^{(V-1)}\|^2 + \|Z_t\mathbf{w} - \mathbf{r}_t\|^2
\]

intuition: enforce \(\mathbf{w}\) to be close to \(\mathbf{w}^{(V-1)}\)
Experiments

Results

- **T-LSA** > **LSA**: transferring beneficial
- **T-LSA** > **T-ALBL**: deterministic transferring better
Active Learning by (Bandit) 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 blending

Transfer Active Learning by (Bandit) Learning

- based on **linear-UCB bandit learning** + **biased regularization** for experience transfer
- effective in **passing active learning experience**
  —better than cold-start

Thank you! Questions?