Active Learning by Bandit Learning

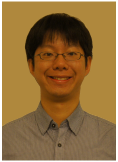
Hsuan-Tien Lin (林軒田)
htlin@csie.ntu.edu.tw

Appier/National Taiwan University
(沛星互動科技/國立台灣大學資訊工程系)

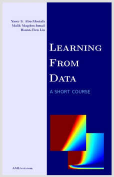
National Center for Theoretical Sciences, March 22, 2018
joint works with Wei-Ning Hsu (AAAI 2015)
and Hong-Min Chu (ICDM 2017)

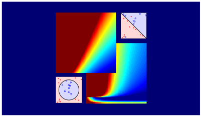
About Me

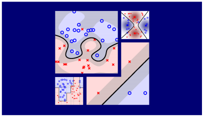
Hsuan-Tien Lin



- Chief Data Scientist, Appier
- 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/ntumltwo`







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/bdy2hZ`


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/7jwtGp`


APAL `https://flic.kr/p/jzPYNr`


APAL `https://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**


Mr. Roboto.
`https://flic.kr/p/i5BN85`


Richard North
`https://flic.kr/p/bHhPkB`


Richard North
`https://flic.kr/p/d8tGou`


Emilian Robert Vicol
`https://flic.kr/p/bpmGXW`


Nathaniel McQueen
`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`

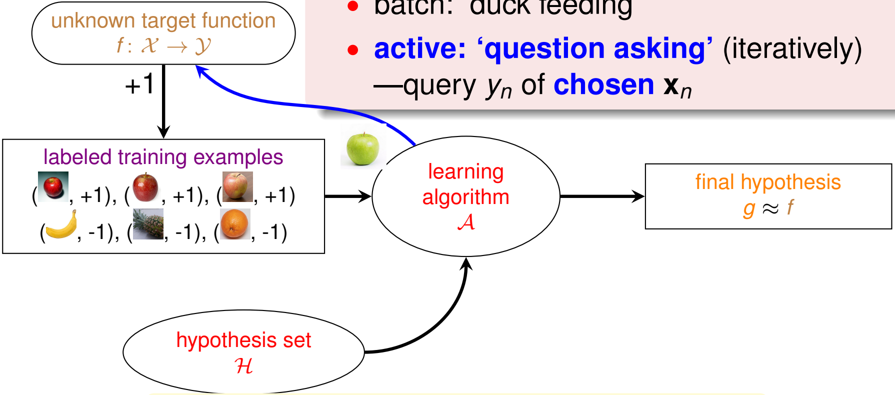

Rennett Stowe
`https://flic.kr/p/agmnrk`

Active Learning: Learning by ‘Asking’

but labeling is **expensive**

Protocol ⇔ Learning Philosophy

- batch: ‘duck feeding’
- **active: ‘question asking’** (iteratively) —query y_n of **chosen** $\mathbf{x}_n$



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

Pool-Based Active Learning Problem

Given

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

Choice of Strategy

Strategy 1:
uncertainty

ask **most confused** question

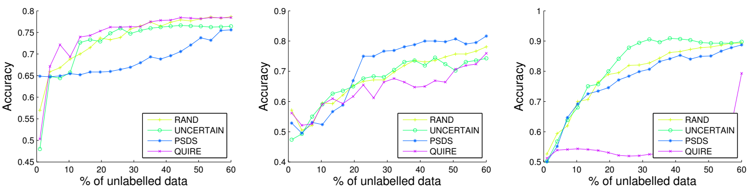
Strategy 2:
representative

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 ...

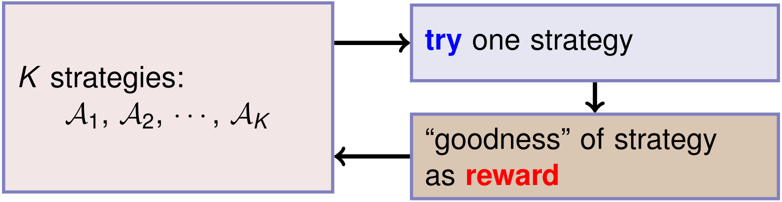
- allows machine to make **intelligent choice of strategies**, just like my cute daughter & son
- studies **sound feedback scheme** to tell machine about goodness of choice, just like what I do
- results in **promising active learning performance**, just like (hopefully) bright future of my daughter & 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:
$\mathcal{A}_1, \mathcal{A}_2, \cdots, \mathcal{A}_K$

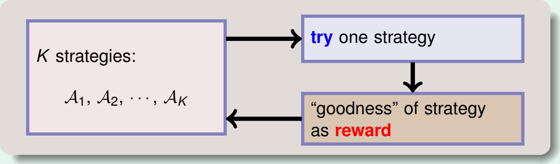
try one strategy

"goodness" of strategy as **reward**

two issues: **try** and **reward**

Reduction to Bandit

when do humans **trial**-and-**reward**?
gambling ☺

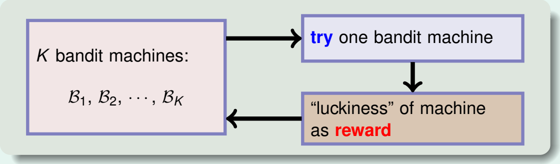


K strategies:

$\mathcal{A}_1, \mathcal{A}_2, \cdots, \mathcal{A}_K$

try one strategy

"goodness" of strategy as **reward**



K bandit machines:

$\mathcal{B}_1, \mathcal{B}_2, \cdots, \mathcal{B}_K$

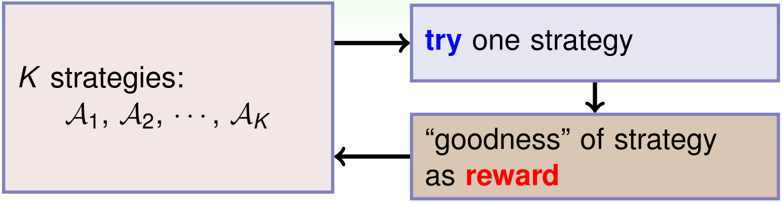
try one bandit machine

"luckiness" of machine as **reward**

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

intelligent choice of strategy
$\Longrightarrow$ intelligent choice of **bandit machine**

Active Learning by Learning



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

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

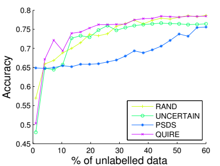
only remaining problem: **what reward**?

Ideal Reward

ideal reward after updating classifier $g^{(t)}$ by the query $(\mathbf{x}_{n_t}, y_{n_t})$:

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

- **test accuracy** as reward:
  area under query-accuracy curve ≡ cumulative reward



- test accuracy **infeasible** in practice
  —labeling **expensive**, remember? ☺

difficulty: approximate test accuracy **on the fly**

Training Accuracy as Reward

test accuracy $\cancel{\frac{1}{M}\sum_{m=1}^{M}[\![y_m = g^{(t)}(\mathbf{x}_m)]\!]}$ infeasible, naïve replacement:

accuracy $\frac{1}{t}\sum_{\tau=1}^{t}\left[\!\left[y_{n_\tau} = g^{(t)}(\mathbf{x}_{n_\tau})\right]\!\right]$ on **labeled pool** $\{(\mathbf{x}_{n_\tau}, y_{n_\tau})\}_{\tau=1}^{t}$

- **training accuracy** as reward:
  training accuracy $\approx$ **test accuracy**?
- not necessarily!!
  —for active learning strategy that asks **easiest** questions:
  - training accuracy **high**: $\mathbf{x}_{n_\tau}$ 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 ~~$\frac{1}{t}\sum_{\tau=1}^{t}[\![y_{n_\tau} = g^{(t)}(\mathbf{x}_{n_\tau})]\!]$ biased~~,
want **less-biased estimator**

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

$$\textbf{weighted training accuracy } \frac{1}{t}\sum_{\tau=1}^{t}\frac{1}{p_\tau}[\![y_{n_\tau} = g(\mathbf{x}_{n_\tau})]\!]$$

$$\approx \quad \frac{1}{N}\sum_{n=1}^{N}[\![y_n = g(\mathbf{x}_n)]\!] \text{ in } \textbf{expectation}$$

- 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:

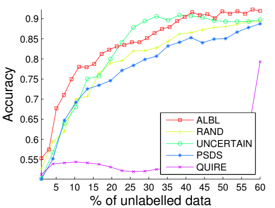
bandit + **balancedness** of $g^{(t)}$ on unlabeled data as reward

- 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**

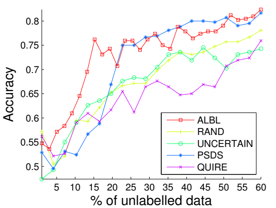
Comparison with Single Strategies

UNCERTAIN Best



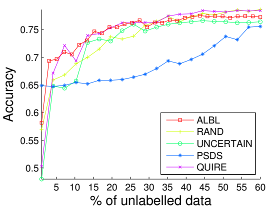
vehicle

PSDS Best



sonar

QUIRE Best



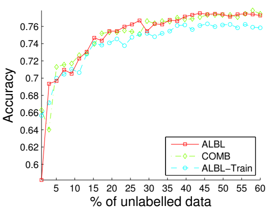
diabetes

- **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**

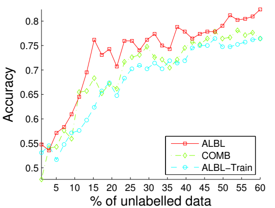
Comparison with Other Adaptive Blending Algorithms

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**

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!

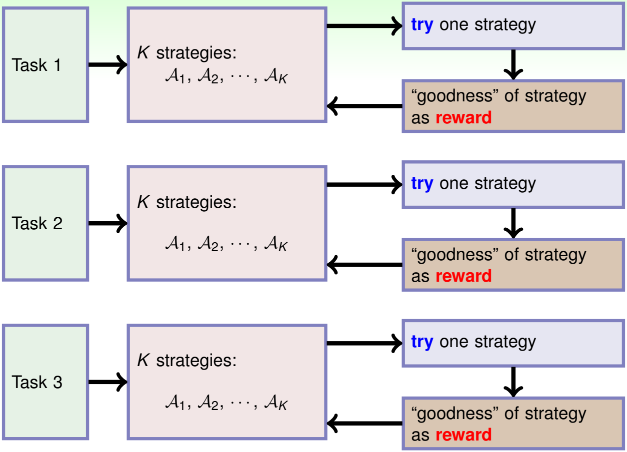
Discussion for Statisticians

weighted training accuracy $\frac{1}{t}\sum_{\tau=1}^{t}\frac{1}{p_\tau}[\![y_{n_\tau} = g^{(t)}(\mathbf{x}_{n_\tau})]\!]$ 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 $(\mathbf{x}_{n_t}, y_{n_t})$
- 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: $\mathcal{A}_1, \mathcal{A}_2, \cdots, \mathcal{A}_K$

try one strategy

"goodness" of strategy as **reward**

Task 2

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

try one strategy

"goodness" of strategy as **reward**

Task 3

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

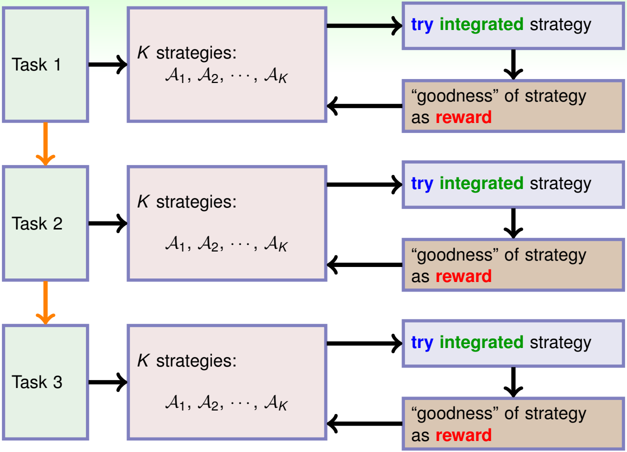
try one strategy

"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: $\mathcal{A}_1, \mathcal{A}_2, \cdots, \mathcal{A}_K$

try **integrated** strategy

"goodness" of strategy as **reward**

Task 2

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

try **integrated** strategy

"goodness" of strategy as **reward**

Task 3

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

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 $\mathcal{D}_u^{(1)}$ and labeled pool

  $\mathcal{D}_l^{(1)} = \Big\{\big($feature $\mathbf{x}_n$ , **label** y_n (e.g. IsApple?)$\big)\Big\}_{n=1}^{N}$

- unlabeled $\mathcal{D}_u^{(2)}$ and labeled pool

  $\mathcal{D}_l^{(2)} = \Big\{\big($feature $\mathbf{x}'_m$ , **label** y'_m (e.g. IsOrange?)$\big)\Big\}_{m=1}^{M}$

- unlabeled $\mathcal{D}_u^{(3)}$ and labeled pool

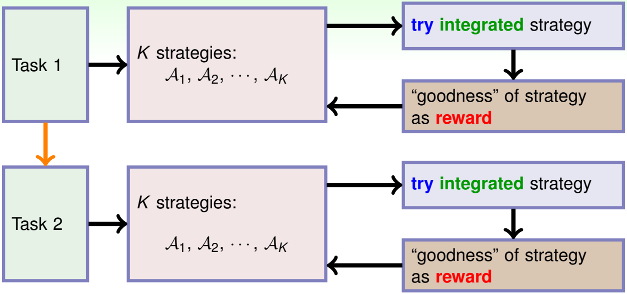
  $\mathcal{D}_l^{(3)} = \Big\{\big($feature $\mathbf{x}''_p$, **label** y''_p (e.g. IsVerb?)$\big)\Big\}_{p=1}^{P}, \ldots$

Goal

exploit **active learning experience** from $\{(\mathcal{D}_l^{(v)}, \mathcal{D}_u^{(v)}))\}_{v=1}^{V-1}$ to improve the active learning performance on $(\mathcal{D}_l^{(V)}, \mathcal{D}_u^{(V)})$

how to define **experience**?

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

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 **w.r.t. UCB**
- context $\mathbf{z}_{\ell,t}$ for bandit solver $\Leftrightarrow$ strategy scores on each instance
- weights $\mathbf{w}_t$ from bandit solver $\Leftrightarrow$ experience

next: Transfer LSA

Transfer LSA with Biased Regularization

Transfer LSA with *Biased Regularization*

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

$$\mathbf{w}_t = \operatorname*{argmin}_{\mathbf{w}} \lambda \|\mathbf{w} - \mathbf{0}\|^2 + \|\mathbf{Z}_t\mathbf{w} - \mathbf{r}_t\|^2$$

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

$$\mathbf{w}_t = \operatorname*{argmin}_{\mathbf{w}} \lambda \|\mathbf{w} - \mathbf{w}^{(V-1)}\|^2 + \|\mathbf{Z}_t\mathbf{w} - \mathbf{r}_t\|^2$$

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

Experiments

T-LSA versus others

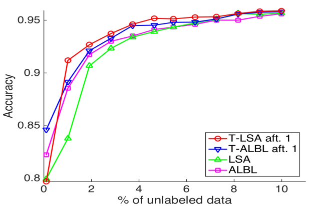


Figure: homogeneous

T-LSA versus others

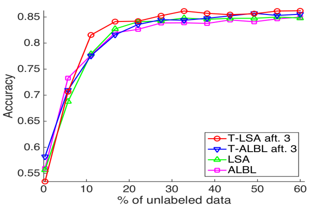


Figure: heterogeneous

Results

- **T-LSA** $>$ LSA: transferring beneficial
- **T-LSA** $>$ **T-ALBL**: deterministic transferring better

Final Summary

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?