

Active Learning via Reduction To Supervised Classification

John Langford (Yahoo! Research) with

Alina Beygelzimer (IBM Research)

Sanjoy Dasgupta (UCSD)

Daniel Hsu (Rutgers & Upenn)

Nikos Karampatziakis (Cornell)

Tong Zhang (Rutgers)

Can a learning algorithm effectively interactively choose which examples to label?

The Active Learning Setting

Repeatedly:

- 1 Observe unlabeled example x .
- 2 Make prediction \hat{y} .
- 3 Asking for label? Yes/no
- 4 If yes, observe label y .

Goal: Simultaneously minimize the number of mistakes and the number of labels requested.

Good solutions imply more efficient learning *and* a better understanding of how to deal with other forms of interactive learning.

Typical heuristics for active learning

Start with a pool of unlabeled data

Pick a few points at random and get their labels

Repeat

- Fit a classifier to the labels seen so far

- Query the unlabeled point that is closest to the boundary
(or most uncertain, or most likely to decrease overall uncertainty,...)

Typical heuristics for active learning

Start with a pool of unlabeled data

Pick a few points at random and get their labels

Repeat

- Fit a classifier to the labels seen so far

- Query the unlabeled point that is closest to the boundary
(or most uncertain, or most likely to decrease overall
uncertainty,...)

Biased sampling: labeled points are not representative of the underlying distribution!

Typical heuristics for active learning

Start with a pool of unlabeled data

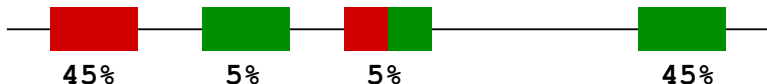
Pick a few points at random and get their labels

Repeat

- Fit a classifier to the labels seen so far

- Query the unlabeled point that is closest to the boundary (or most uncertain, or most likely to decrease overall uncertainty,...)

Biased sampling: labeled points are not representative of the underlying distribution!



Typical heuristics for active learning

Start with a pool of unlabeled data

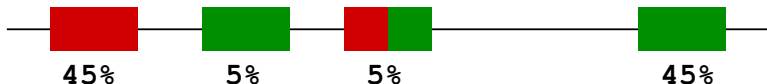
Pick a few points at random and get their labels

Repeat

Fit a classifier to the labels seen so far

Query the unlabeled point that is closest to the boundary
(or most uncertain, or most likely to decrease overall
uncertainty,...)

Biased sampling: labeled points are not representative of the underlying distribution!



Even with infinitely many labels, converges to a classifier with 5% error instead of the best achievable, 2.5%. *Not consistent!*

Is this fixable?

Is this fixable?

- 1 BBL 2006: Yes, ignoring all issues except label efficiency.

Is this fixable?

- 1 BBL 2006: Yes, ignoring all issues except label efficiency.
- 2 DHM 2007: Yes, with an online algorithm also achieving unlabeled data efficiency.

Is this fixable?

- 1 BBL 2006: Yes, ignoring all issues except label efficiency.
- 2 DHM 2007: Yes, with an online algorithm also achieving unlabeled data efficiency.
- 3 BDL 2009: The same for other loss functions.

Is this fixable?

- 1 BBL 2006: Yes, ignoring all issues except label efficiency.
- 2 DHM 2007: Yes, with an online algorithm also achieving unlabeled data efficiency.
- 3 BDL 2009: The same for other loss functions.
- 4 BHLZ 2010: Yes, given an efficient loss optimization algorithm. This talk.

Importance Weighted Active Learning via Reduction

$$S = \emptyset$$

While (unlabeled examples remain)

- 1 Receive unlabeled example x .
- 2 Set $p = \text{Rejection-Threshold}(x, S)$.
- 3 If $U(0, 1) \leq p$, get label y , and add $(x, y, \frac{1}{p})$ to S .
- 4 Let $h = \text{Learn}(S)$.

Consistency: (BDL2009) For all reasonable choices of Rejection-Threshold, the algorithm is consistent.

What should **Rejection-Threshold** be?

On the k th unlabeled point, let:

$\hat{e}(h, S) = \frac{1}{k} \sum_{(x,y,i) \in S} i \mathbb{1}(h(x) \neq y) =$ importance weighted error rate.

What should **Rejection-Threshold** be?

On the k th unlabeled point, let:

$\hat{e}(h, S) = \frac{1}{k} \sum_{(x,y,i) \in S} i \mathbb{1}(h(x) \neq y) =$ importance weighted error rate.

Let h' = minimum error rate hypothesis choosing other label.

What should **Rejection-Threshold** be?

On the k th unlabeled point, let:

$\hat{e}(h, S) = \frac{1}{k} \sum_{(x,y,i) \in S} i \mathbb{1}(h(x) \neq y)$ = importance weighted error rate.

Let h' = minimum error rate hypothesis choosing other label.

Let $\Delta = \hat{e}(h', S) - \hat{e}(h, S)$ = error rate difference.

What should **Rejection-Threshold** be?

On the k th unlabeled point, let:

$\hat{e}(h, S) = \frac{1}{k} \sum_{(x,y,i) \in S} i \mathbb{1}(h(x) \neq y)$ = importance weighted error rate.

Let h' = minimum error rate hypothesis choosing other label.

Let $\Delta = \hat{e}(h', S) - \hat{e}(h, S)$ = error rate difference.

Choose $p = 1$ if $\Delta \leq O\left(\sqrt{\frac{\log k}{k}}\right)$

Otherwise, let $p = O\left(\frac{\log k}{\Delta^2 k}\right)$

What should **Rejection-Threshold** be?

On the k th unlabeled point, let:

$\hat{e}(h, S) = \frac{1}{k} \sum_{(x,y,i) \in S} i \mathbb{1}(h(x) \neq y)$ = importance weighted error rate.

Let h' = minimum error rate hypothesis choosing other label.

Let $\Delta = \hat{e}(h', S) - \hat{e}(h, S)$ = error rate difference.

Choose $p = 1$ if $\Delta \leq O\left(\sqrt{\frac{\log k}{k}}\right)$

Otherwise, let $p = O\left(\frac{\log k}{\Delta^2 k}\right)$

Competition: (BHLZ2010) With high probability, the IWAL reduction has a similar error rate as supervised learning on k points.

What should **Rejection-Threshold** be?

On the k th unlabeled point, let:

$\hat{e}(h, S) = \frac{1}{k} \sum_{(x,y,i) \in S} i \mathbb{1}(h(x) \neq y)$ = importance weighted error rate.

Let h' = minimum error rate hypothesis choosing other label.

Let $\Delta = \hat{e}(h', S) - \hat{e}(h, S)$ = error rate difference.

Choose $p = 1$ if $\Delta \leq O\left(\sqrt{\frac{\log k}{k}}\right)$

Otherwise, let $p = O\left(\frac{\log k}{\Delta^2 k}\right)$

Competition: (BHLZ2010) With high probability, the IWAL reduction has a similar error rate as supervised learning on k points.

Success: (BHLZ2010) If there is a small **disagreement coefficient** θ , the algorithm requires only $O(\theta \sqrt{k \log k})$ + a minimum due to noise (K2006).

Disagreement Coefficient (Hanneke 2007)

Characterizes known examples where active learning can help.
Defined for any set of classifiers H and distribution D .

Disagreement Coefficient (Hanneke 2007)

Characterizes known examples where active learning can help.
Defined for any set of classifiers H and distribution D .

For any ϵ features x are of interest if there exists a hypothesis h :

- 1 With error rate less than ϵ larger than the best h^* .
- 2 That disagree with the best hypothesis, $h^*(x) \neq h(x)$.

Disagreement Coefficient (Hanneke 2007)

Characterizes known examples where active learning can help.
Defined for any set of classifiers H and distribution D .

For any ϵ features x are of interest if there exists a hypothesis h :

- 1 With error rate less than ϵ larger than the best h^* .
- 2 That disagree with the best hypothesis, $h^*(x) \neq h(x)$.

Disagreement coefficient is $\theta = \max_x \frac{\Pr(\text{interesting}_\epsilon x)}{\epsilon}$
(See ICML 2009 tutorial for examples)

The Martingale Barrier Problem

Proofs are complex, but rest on the solution to a Martingale Barrier Problem.

The Martingale Barrier Problem

Proofs are complex, but rest on the solution to a Martingale Barrier Problem.

Given a coin of bias < 0.5 , how can we choose the probability of p of a coin flip so that:

- 1 The average number of heads is small: $\frac{1}{k} \sum_{(h,p) \in S} \frac{h}{p} < 0.5$.
- 2 The number of coin flips is minimized: $\min \sum_{(h,p) \in S} p$.
- 3 The probability is nontrivial: $p > 0$.

The Martingale Barrier Problem

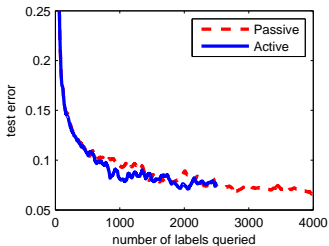
Proofs are complex, but rest on the solution to a Martingale Barrier Problem.

Given a coin of bias < 0.5 , how can we choose the probability of p of a coin flip so that:

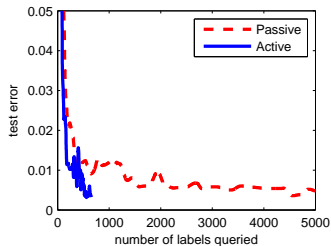
- 1 The average number of heads is small: $\frac{1}{k} \sum_{(h,p) \in S} \frac{h}{p} < 0.5$.
- 2 The number of coin flips is minimized: $\min \sum_{(h,p) \in S} p$.
- 3 The probability is nontrivial: $p > 0$.

p too small, implies that condition (1) is violated with a reasonable probability.

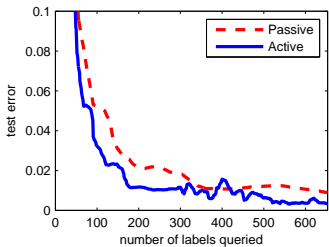
Decision Tree Experiments



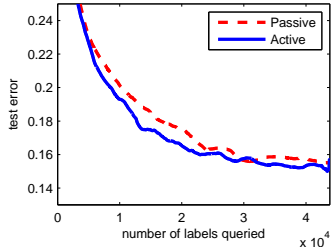
MNIST 3s vs 5s



KDDCUP99

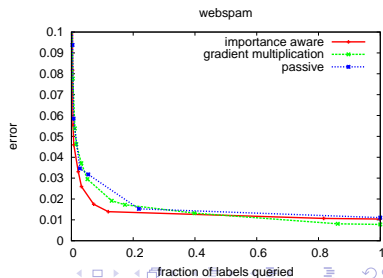
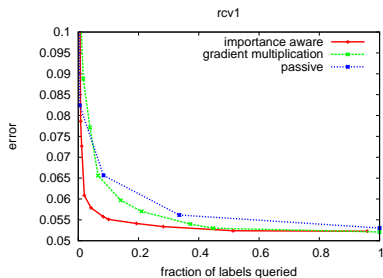
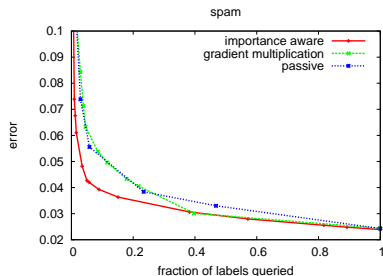
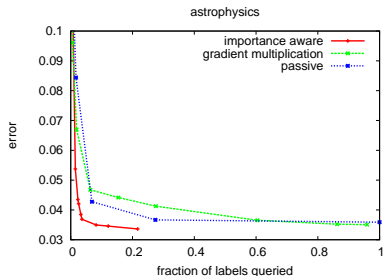


KDDCUP99 (close-up)



MNIST multi-class (close-up)

Online Linear Learning results (with Nikos)



Demonstration

Fringe Benefits

This approach has **many** nice properties.

Fringe Benefits

This approach has **many** nice properties.

- 1 Always consistent.

Fringe Benefits

This approach has **many** nice properties.

- ① Always consistent.
- ② Computationally efficient given any efficient optimization-style classification algorithm.

Fringe Benefits

This approach has **many** nice properties.

- 1 Always consistent.
- 2 Computationally efficient given any efficient optimization-style classification algorithm.
- 3 Unlabeled data efficient.

Fringe Benefits

This approach has **many** nice properties.

- 1 Always consistent.
- 2 Computationally efficient given any efficient optimization-style classification algorithm.
- 3 Unlabeled data efficient.
- 4 Online Compatible.

Fringe Benefits

This approach has **many** nice properties.

- 1 Always consistent.
- 2 Computationally efficient given any efficient optimization-style classification algorithm.
- 3 Unlabeled data efficient.
- 4 Online Compatible.
- 5 Label Efficient.

Fringe Benefits

This approach has **many** nice properties.

- 1 Always consistent.
- 2 Computationally efficient given any efficient optimization-style classification algorithm.
- 3 Unlabeled data efficient.
- 4 Online Compatible.
- 5 Label Efficient.
- 6 Compatible with any optimization-style classification algorithm.

Fringe Benefits

This approach has **many** nice properties.

- 1 Always consistent.
- 2 Computationally efficient given any efficient optimization-style classification algorithm.
- 3 Unlabeled data efficient.
- 4 Online Compatible.
- 5 Label Efficient.
- 6 Compatible with any optimization-style classification algorithm.
- 7 Works for other loss functions.

Fringe Benefits

This approach has **many** nice properties.

- 1 Always consistent.
- 2 Computationally efficient given any efficient optimization-style classification algorithm.
- 3 Unlabeled data efficient.
- 4 Online Compatible.
- 5 Label Efficient.
- 6 Compatible with any optimization-style classification algorithm.
- 7 Works for other loss functions.
- 8 Interpolates to supervised learning.

Fringe Benefits

This approach has **many** nice properties.

- 1 Always consistent.
- 2 Computationally efficient given any efficient optimization-style classification algorithm.
- 3 Unlabeled data efficient.
- 4 Online Compatible.
- 5 Label Efficient.
- 6 Compatible with any optimization-style classification algorithm.
- 7 Works for other loss functions.
- 8 Interpolates to supervised learning.
- 9 Allows you to switch learning algorithms later (!)

Fringe Benefits

This approach has **many** nice properties.

- 1 Always consistent.
- 2 Computationally efficient given any efficient optimization-style classification algorithm.
- 3 Unlabeled data efficient.
- 4 Online Compatible.
- 5 Label Efficient.
- 6 Compatible with any optimization-style classification algorithm.
- 7 Works for other loss functions.
- 8 Interpolates to supervised learning.
- 9 Allows you to switch learning algorithms later (!)
- 10 Empirically, yields substantial label savings.

Active Learning is only one kind of interactive learning. Does a similar strategy work with other forms of interactive learning?

Bibliography

- 1 Nina Balcan, Alina Beygelzimer, John Langford, Agnostic Active Learning. ICML 2006.
- 2 Alina Beygelzimer, Sanjoy Dasgupta, and John Langford, Importance Weighted Active Learning, ICML 2009.
- 3 Alina Beygelzimer, Daniel Hsu, John Langford, Tong Zhang, Agnostic Active Learning Without Constraints, NIPS 2010.
- 4 Sanjoy Dasgupta, Daniel J. Hsu, and Claire Monteleoni. A general agnostic active learning algorithm. NIPS 2007.
- 5 Hanneke, S. A Bound on the Label Complexity of Agnostic Active Learning. ICML 2007.
- 6 Matti Kaariainen, Active Learning in the Non-realizable Case, ALT 2006.
- 7 Nikos Karampatziakis and John Langford, Importance Weight Aware Gradient Updates, <http://arxiv.org/abs/1011.1576>