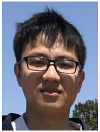
Practical Agnostic Active Learning

Alina Beygelzimer
Yahoo Research

based on joint work with Sanjoy Dasgupta, Daniel Hsu, John Langford, Francesco Orabona, Chicheng Zhang, and Tong Zhang

 *     

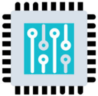
* introductory slide credit

Active learning

Labels are often much more expensive than inputs:

- documents, images, audio, video,
- drug compounds, ...

Can interaction help us learn more effectively?



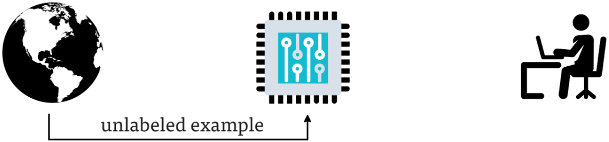
Learn an accurate classifier requesting as few labels as possible.

Active learning

Labels are often much more expensive than inputs:

- documents, images, audio, video,
- drug compounds, . . .

Can interaction help us learn more effectively?



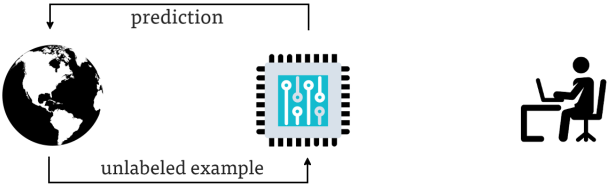
Learn an accurate classifier requesting as few labels as possible.

Active learning

Labels are often much more expensive than inputs:

- documents, images, audio, video,
- drug compounds, ...

Can interaction help us learn more effectively?



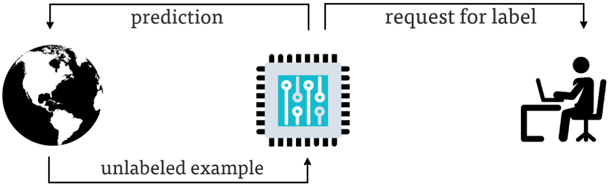
Learn an accurate classifier requesting as few labels as possible.

Active learning

Labels are often much more expensive than inputs:

- documents, images, audio, video,
- drug compounds, . . .

Can interaction help us learn more effectively?



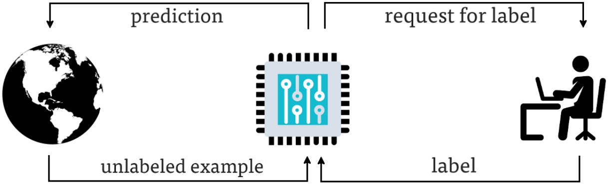
Learn an accurate classifier requesting as few labels as possible.

Active learning

Labels are often much more expensive than inputs:

- documents, images, audio, video,
- drug compounds, ...

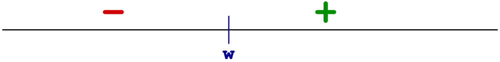
Can interaction help us learn more effectively?



Learn an accurate classifier requesting as few labels as possible.

Why should we hope for success?

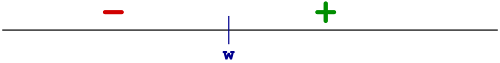
Threshold functions on the real line. Target is a threshold.



Supervised: Need $\approx 1/\epsilon$ labeled points. With high probability, any consistent threshold has $\leq \epsilon$ error.

Why should we hope for success?

Threshold functions on the real line. Target is a threshold.

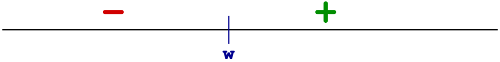


Supervised: Need $\approx 1/\epsilon$ labeled points. With high probability, any consistent threshold has $\leq \epsilon$ error.

Active learning: start with $1/\epsilon$ *unlabeled* points.

Why should we hope for success?

Threshold functions on the real line. Target is a threshold.



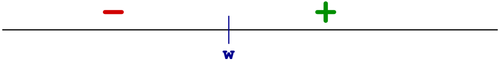
Supervised: Need $\approx 1/\epsilon$ labeled points. With high probability, any consistent threshold has $\leq \epsilon$ error.

Active learning: start with $1/\epsilon$ *unlabeled* points.

Binary search: need just $\log 1/\epsilon$ labels. *Exponential improvement in label complexity!*

Why should we hope for success?

Threshold functions on the real line. Target is a threshold.



Supervised: Need $\approx 1/\epsilon$ labeled points. With high probability, any consistent threshold has $\leq \epsilon$ error.

Active learning: start with $1/\epsilon$ *unlabeled* points.

Binary search: need just $\log 1/\epsilon$ labels. *Exponential improvement in label complexity!*

Nonseparable data? Other hypothesis classes?

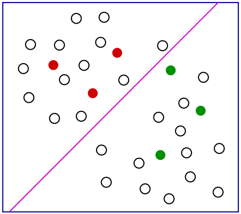
Typical heuristics for active learning

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, ...)



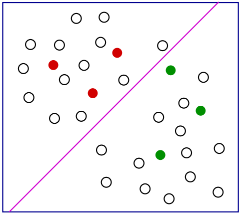
Typical heuristics for active learning

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, . . .)



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

Sampling bias

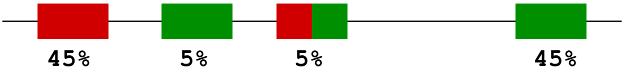
Start with a pool of unlabeled data

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,...)

Example:



Sampling bias

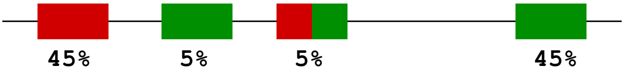
Start with a pool of unlabeled data

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,...)

Example:



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

"Missed cluster effect" (Schütze *et al*, 2006)

Setting:

- Hypothesis class H. For $h \in H$,

$$\mathrm{err}(h) = \Pr_{(x,y)}[h(x) \neq y]$$

- Minimum error rate $\nu = \min_{h \in H} \mathrm{err}(h)$
- Given ϵ, find $h \in H$ with $\mathrm{err}(h) \leq \nu + \epsilon$

Desiderata:

- General H
- Consistent: always converge
- Agnostic: deal with arbitrary noise, $\nu \geq 0$
- Efficient: statistically and computationally

Setting:

- Hypothesis class H. For $h \in H$,

$$\mathrm{err}(h) = \Pr_{(x,y)}[h(x) \neq y]$$

- Minimum error rate $\nu = \min_{h \in H} \mathrm{err}(h)$
- Given ϵ, find $h \in H$ with $\mathrm{err}(h) \leq \nu + \epsilon$

Desiderata:

- General H
- Consistent: always converge
- Agnostic: deal with arbitrary noise, $\nu \geq 0$
- Efficient: statistically and computationally

Is this achievable?

Setting:

- Hypothesis class H. For $h \in H$,

$$\operatorname{err}(h) = \Pr_{(x,y)}[h(x) \neq y]$$

- Minimum error rate $\nu = \min_{h \in H} \operatorname{err}(h)$
- Given ϵ, find $h \in H$ with $\operatorname{err}(h) \leq \nu + \epsilon$

Desiderata:

- General H
- Consistent: always converge
- Agnostic: deal with arbitrary noise, $\nu \geq 0$
- Efficient: statistically and computationally

Is this achievable? Yes (BBL-2006, DHM-2007, BDL-2009, BHLZ-2010, BHLZ-2016)

Importance Weighted Active Learning

$S_0 = \emptyset$

For $t = 1, 2, \ldots, n$

1. Receive unlabeled example x_t and set $S_t = S_{t-1}$.
2. Choose a probability of labeling p_t.
3. Flip a coin Q_t with $\mathbf{E}[Q_t] = p_t$. If $Q_t = 1$, request y_t and add $(x_t, y_t, \frac{1}{p_t})$ to S_t.
4. Let $h_{t+1} = \mathrm{LEARN}(S_t)$.

Empirical importance-weighted error

$$\mathrm{err}_n(h) = \frac{1}{n} \sum_{t=1}^{n} \frac{Q_t}{p_t} \mathbf{1}[h(x_t) \neq y_t]$$

Minimizer $\mathrm{LEARN}(S_t) = \arg\min_{h \in H} \mathrm{err}_t(h)$

Consistency: The algorithm is consistent as long as p_t are bounded away from 0.

How should p_t be chosen?

Let Δ_t = increase in empirical importance-weighted error rate if learner is forced to change its prediction on x_t.

Set $p_t = 1$ if $\Delta_t \leq O\left(\sqrt{\frac{\log t}{t}}\right)$; otherwise, $p_t = O\left(\frac{\log t}{\Delta_t^2 t}\right)$.

- $h_t = \arg\min\{\mathrm{err}_{t-1}(h) : h \in H\}$
- $h'_t = \arg\min\{\mathrm{err}_{t-1}(h) : h \in H \text{ and } h(x_t) \neq h_t(x_t)\}$
- $\Delta_t = \mathrm{err}_{t-1}(h'_t) - \mathrm{err}_{t-1}(h_t)$

How should p_t be chosen?

Let Δ_t = increase in empirical importance-weighted error rate if learner is forced to change its prediction on x_t.

Set $p_t = 1$ if $\Delta_t \leq O\left(\sqrt{\frac{\log t}{t}}\right)$; otherwise, $p_t = O\left(\frac{\log t}{\Delta_t^2 t}\right)$.

- $h_t = \arg\min\{\text{err}_{t-1}(h) : h \in H\}$
- $h'_t = \arg\min\{\text{err}_{t-1}(h) : h \in H$ and $h(x_t) \neq h_t(x_t)\}$
- $\Delta_t = \text{err}_{t-1}(h'_t) - \text{err}_{t-1}(h_t)$

How can we compute Δ_t in constant time?

How should p_t be chosen?

Let Δ_t = increase in empirical importance-weighted error rate if learner is forced to change its prediction on x_t.

Set $p_t = 1$ if $\Delta_t \leq O\left(\sqrt{\frac{\log t}{t}}\right)$; otherwise, $p_t = O\left(\frac{\log t}{\Delta_t^2 t}\right)$.

- $h_t = \arg\min\{\mathrm{err}_{t-1}(h) : h \in H\}$
- $h'_t = \arg\min\{\mathrm{err}_{t-1}(h) : h \in H$ and $h(x_t) \neq h_t(x_t)\}$
- $\Delta_t = \mathrm{err}_{t-1}(h'_t) - \mathrm{err}_{t-1}(h_t)$

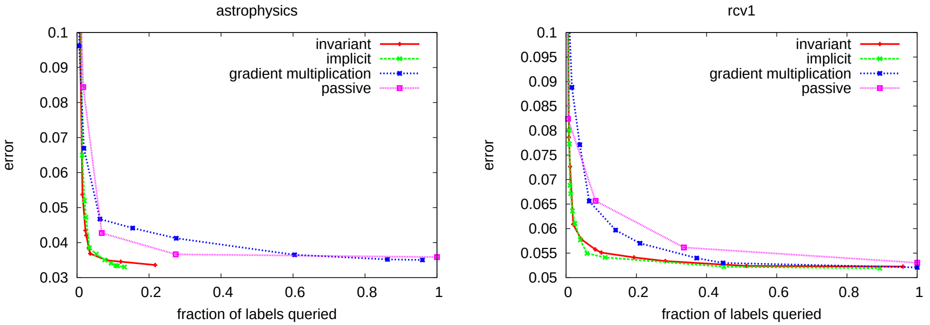
How can we compute Δ_t in constant time?

- Find the smallest i_t such that $h_{t+1} = h'_t$ after $(x_t, h'_t(x_t), i_t)$ update.
- We have $(t-1) \cdot \mathrm{err}_{t-1}(h'_t) \leq (t-1) \cdot \mathrm{err}_{t-1}(h_t) + i_t$. Thus $\Delta_t \leq i_t/(t-1)$.

Practical Considerations

Importance weight aware SGD updates [Karampatziakis and Langford]. Solve for i_t directly. E.g., for logistic,

$$i_t = \frac{2 w_t^T x_t}{\eta_t \,\mathrm{sign}(w_t^T x_t) x_t^T x_t}$$



Guarantees

IWAL achieves error similar to that of supervised learning on n points:

Accuracy Theorem: For all $n \geq 1$,

$$\mathrm{err}(h_n) \leq \mathrm{err}(h^*) + \sqrt{\frac{C \log n}{n-1}}$$

with high probability.

Guarantees

IWAL achieves error similar to that of supervised learning on n points:

Accuracy Theorem: For all $n \geq 1$,

$$\mathrm{err}(h_n) \leq \mathrm{err}(h^*) + \sqrt{\frac{C \log n}{n-1}}$$

with high probability.

Label Efficiency Theorem: With high probability, the expected number of labels queried after n iteractions is at most

$$\underbrace{O(\theta\, \mathrm{err}(h^*) n)}_{\text{minimum due to noise}} + O\left(\theta \sqrt{n \log n}\right)$$

where θ is the disagreement coefficient.

The crucial ratio: Disagreement Coefficient (Hanneke-2007)

r-ball around a minimum-error hypothesis h^*:

$$B(h^*, r) = \{h \in H : \Pr[h(x) \neq h^*(x)] \leq r\}$$

Disagreement region of $B(h^*, r)$:

$$\mathrm{DIS}(B(h^*, r)) = \{x \in X \mid \exists h, h' \in B(h^*, r) : h(x) \neq h'(x)\}$$

The disagreement coefficient measures the rate of

$$\theta = \sup_{r>0} \frac{\Pr[\mathrm{DIS}(B(h^*, r))]}{r}$$

Example:

The crucial ratio: Disagreement Coefficient (Hanneke-2007)

r-ball around a minimum-error hypothesis h^*:

$$B(h^*, r) = \{h \in H : \Pr[h(x) \neq h^*(x)] \leq r\}$$

Disagreement region of $B(h^*, r)$:

$$\mathrm{DIS}(B(h^*, r)) = \{x \in X \mid \exists h, h' \in B(h^*, r) : h(x) \neq h'(x)\}$$

The disagreement coefficient measures the rate of

$$\theta = \sup_{r>0} \frac{\Pr[\mathrm{DIS}(B(h^*, r))]}{r}$$

Example:

- Thresholds in $\mathbb{R}$, any data distribution. $\theta = 2$.

Benefits

1. Always consistent.

Benefits

1. Always consistent.
2. Efficient.
   2.1 Label efficient, unlabeled data efficient, computationally efficient.

Benefits

1. Always consistent.
2. Efficient.
   2.1 Label efficient, unlabeled data efficient, computationally efficient.
3. Compatible.
   3.1 With online algorithms
   3.2 With any optimization-style classification algorithms
   3.3 With any Loss function
   3.4 With supervised learning
   3.5 With switching learning algorithms (!)

Benefits

1. Always consistent.
2. Efficient.
   2.1 Label efficient, unlabeled data efficient, computationally efficient.
3. Compatible.
   3.1 With online algorithms
   3.2 With any optimization-style classification algorithms
   3.3 With any Loss function
   3.4 With supervised learning
   3.5 With switching learning algorithms (!)
4. Collected labeled set is reusable with a different algorithm or hypothesis class.

Benefits

1. Always consistent.
2. Efficient.
   2.1 Label efficient, unlabeled data efficient, computationally efficient.
3. Compatible.
   3.1 With online algorithms
   3.2 With any optimization-style classification algorithms
   3.3 With any Loss function
   3.4 With supervised learning
   3.5 With switching learning algorithms (!)
4. Collected labeled set is reusable with a different algorithm or hypothesis class.
5. It works, empirically.

Applications

- News article categorizer
- Image classification
- Hate-speech detection / comments sentiment
- NLP sentiment classification (satire, newsiness, gravitas)

Active learning in Vowpal Wabbit

Simulating active learning: (knob $c > 0$)

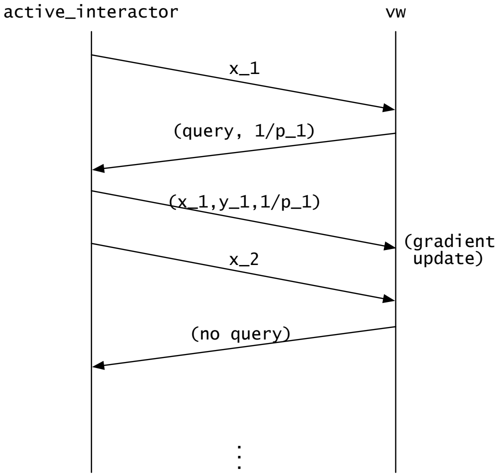
```
vw --active_simulation --active_mellowness c
```

Deploying active learning:

```
vw --active_learning --active_mellowness c --daemon
```

- `vw` interacts with an `active_interactor` (`ai`)
- receives labeled and unlabeled training examples from `ai` over network
- for each unlabeled data point, `vw` sends back a query decision (and an importance weight if label is requested)
- `ai` sends labeled importance-weighted examples as requested
- `vw` trains using labeled importance-weighted examples

Active learning in Vowpal Wabbit



`active_interactor.cc` (in git repository) demonstrates how to implement this protocol.

Needle in a haystack problem: Rare classes

Learning interval functions $h_{a,b}(x) = \mathbf{1}[a \leq x \leq b]$, for $0 \leq a \leq b \leq 1$.

Supervised learning: need $O(1/\epsilon)$ labeled data.

Active learning: need $O(1/W + \log 1/\epsilon)$ labels, where W is the width of the target interval. No improvement over passive learning.

Needle in a haystack problem: Rare classes

Learning interval functions $h_{a,b}(x) = \mathbf{1}[a \leq x \leq b]$, for $0 \leq a \leq b \leq 1$.

Supervised learning: need $O(1/\epsilon)$ labeled data.

Active learning: need $O(1/W + \log 1/\epsilon)$ labels, where W is the width of the target interval. No improvement over passive learning.

...but given *any* example of the rare class, the label complexity drops to $O(\log 1/\epsilon)$.

Dasgupta 2005; Attenberg & Provost 2010: Search and insertion of labeled rare class examples helps.

predicted label: Entertainment

predicted label: Entertainment

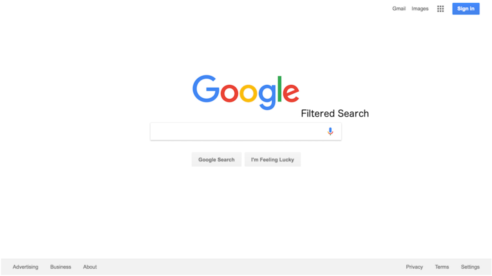
predicted label: Entertainment

How can editors feed observed mistakes into an active learning algorithm?

Is this fixable?

Beygelzimer-Hsu-Langford-Zhang (NIPS-16):

Define a Search oracle:



The active learner interactively restricts the searchable space guiding Search where it's most effective.

Search Oracle

Oracle Search

Require: Working set of candidate models V

Ensure: Labeled example (x, y) s.t. $h(x) \neq y$ for all $h \in V$ (systematic mistake), or $\perp$ if there is no such example.

Search Oracle

Oracle Search

Require: Working set of candidate models V

Ensure: Labeled example (x, y) s.t. $h(x) \neq y$ for all $h \in V$ (systematic mistake), or $\perp$ if there is no such example.

How can a counterexample to a version space be used?

Search Oracle

Oracle Search

Require: Working set of candidate models V

Ensure: Labeled example (x, y) s.t. $h(x) \neq y$ for all $h \in V$ (systematic mistake), or $\perp$ if there is no such example.

How can a counterexample to a version space be used?

Nested sequence of model classes of increasing complexity:

$$H_1 \subseteq H_2 \subseteq \dots H_{k^*} \dots$$

Advance to more complex classes as simple are proved inadequate.

Search + Label

Search + Label can provide exponentially large problem-dependent improvements over Label alone, with a general agnostic algorithm.

Union of intervals example:

- $\tilde{O}(k^* + \log(1/\epsilon))$ Search queries
- $\tilde{O}((\operatorname{poly}\log(1/\epsilon) + \log k^*)(1 + \frac{\nu^2}{\epsilon^2}))$ Label queries

Search + Label

Search + Label can provide exponentially large problem-dependent improvements over Label alone, with a general agnostic algorithm.

Union of intervals example:

- $\tilde{O}(k^* + \log(1/\epsilon))$ Search queries
- $\tilde{O}((\text{poly}\log(1/\epsilon) + \log k^*)(1 + \frac{\nu^2}{\epsilon^2}))$ Label queries

How can we make it practical?

- Drawbacks as with any version space approach
- Can we reformulate as a reduction to supervised learning?

Interactive Learning

Interactive settings:

- Active learning
- Contextual bandit learning
- Reinforcement learning

Bias is a pervasive issue:

- The learner creates the data it learns from / is evaluated on
- State of the world depends on the learner's decisions

Interactive Learning

Interactive settings:

- Active learning
- Contextual bandit learning
- Reinforcement learning

Bias is a pervasive issue:

- The learner creates the data it learns from / is evaluated on
- State of the world depends on the learner's decisions

How can we use supervised learning techonology in these new interactive settings?

Optimal Multiclass Bandit Learning (ICML-2017)

For $t = 1 \dots T$:

1. Observe x_t
2. Predict label $\hat{y}_t \in [1, \dots, K]$
3. Pay and observe $\mathbf{1}[\hat{y}_t \neq y_t]$ (ad not clicked)

No stochastic assumptions on the input sequence.

Compete with multiclass linear predictors $\{W \in R^{K \times d}\}$, where

$$W(x) = \arg\max_{k \in [K]} (W \cdot x)_k$$

Mistake Bounds

Banditron [Kakade-Shalev-Shwartz-Tewari, ICML-08]
SOBA [Beygelzimer-Orabona-Zhang, ICML-17]

Perceptron	Banditron	SOBA
$L + \sqrt{T}$	$L + T^{2/3}$	$L + \sqrt{T}$

where L is the competitor's total hinge loss.

Per-round hinge loss of W

$$l_t(W) = \max_{r \neq y_t}[1 - (Wx_t)_{y_t} + (Wx_t)_r]_+ \geq \mathbf{1}[y_t \neq \hat{y}_t]$$

Resolves a COLT open problem (Abernethy and Rakhlin'09)

The Multiclass Perceptron

A linear multiclass predictor is defined by a matrix $W \in R^{k \times d}$.
For $t = 1 \ldots T$:

- Receive $x_t \in \mathbb{R}^d$
- Predict $\hat{y}_t = \arg\max_r (W^t x_t)_r$
- Receive y_t
- Update $W^{t+1} = W^t + U^t$ where
  $U_t = \mathbf{1}[\hat{y}_t \neq y_t](e_{y_t} - e_{\hat{y}_t}) \otimes x_t$

Bandit Setting

- If $\hat{y}_t \neq y_t$, we are blind to the value of y_t
- Solution: Randomization!

SOBA:

- A second order perceptron with a novel unbiased estimator for the perceptron update and the second order update.
- Passive-aggressive update (sometimes updating when there is no mistake but the margin is small)


The End