Active Learning

John Langford © Microsoft Research

Machine Learning the Future Class, May 1
An instrument of mass machine learning

Mechanical Turk is a marketplace for work.
We give businesses and developers access to an on-demand, scalable workforce. Workers select from thousands of tasks and work whenever it’s convenient.

255,697 HITs available. View them now.

Make Money by working on HITs

HITs - Human Intelligence Tasks - are individual tasks that you work on. Find HITs now.

As a Mechanical Turk Worker you:
- Can work from home
- Choose your own work hours
- Get paid for doing good work

Get Results from Mechanical Turk Workers

Ask workers to complete HITs - Human Intelligence Tasks - and get results using Mechanical Turk. Register Now

As a Mechanical Turk Requester you:
- Have access to a global, on-demand, 24 x 7 workforce
- Get thousands of HITs completed in minutes
- Pay only when you’re satisfied with the results

How can we formalize it’s use?
A lot of unlabeled data is plentiful and cheap, eg.
  documents off the web
  speech samples
  images and video

*But labeling can be expensive.*
Exploiting unlabeled data

A lot of unlabeled data is plentiful and cheap, eg.
documents off the web
speech samples
images and video

*But labeling can be expensive.*
Exploiting unlabeled data

A lot of unlabeled data is plentiful and cheap, eg.
documents off the web
speech samples
images and video

*But labeling can be expensive.*

Unlabeled points  Supervised learning
Exploiting unlabeled data

A lot of unlabeled data is plentiful and cheap, eg.
- documents off the web
- speech samples
- images and video

*But labeling can be expensive.*

Unlabeled points  Supervised learning  Semisupervised and active learning
Can interaction help us learn effectively?

**The Active Learning Setting**

Repeatedly:

1. Observe unlabeled example $x$.
2. Asking for label? Yes/no
3. If yes, observe label $y$.

Goal: Simultaneously optimize quality of learned classifier and minimize the number of labels requested.
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: 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:

```
45%  5%  5%  45%
```

Even with infinitely many labels, converges to a classifier with 5%
error instead of the best achievable, 2.5%.
Not consistent!
This problem occurs in practice.
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!

This problem occurs in practice.
Outline

1. Importance weighting
2. Rare Classes
3. Cost Sensitive
$S = \emptyset$

While (unlabeled examples remain)

1. Receive unlabeled example $x$.
2. Choose a probability of labeling $p$.
3. With probability $p$ get label $y$, and add $(x, y, \frac{1}{p})$ to $S$.
4. Let $h = \text{Learn}(S)$.

Consistency Theorem: For all methods choosing $p > 0$, the algorithm is consistent.
How should $p$ be chosen?

On the $k$th unlabeled point

let: $\hat{e}(h, S) = \frac{1}{k} \sum_{(x,y,i) \in S} i \mathbb{1}(h(x) \neq y) = \text{importance weighted error rate.}$
How should \( p \) be chosen?

On the \( k \)th unlabeled point

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

Let \( h' = \text{minimum error rate hypothesis choosing other label.} \)
How should $p$ be chosen?

On the $k$th unlabeled point

let: $\hat{e}(h, S) = \frac{1}{k} \sum_{(x,y,i) \in S} i1(h(x) \neq y) = \text{importance weighted error rate.}$

Let $h' = \text{minimum error rate hypothesis choosing other label.}$

Let $\Delta = \hat{e}(h', S) - \hat{e}(h, S) = \text{error rate difference.}$
How should $p$ be chosen?

**On the $k$th unlabeled point**

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

Let $h' = \text{minimum error rate hypothesis choosing other label.}$

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

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

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

On the $k$th unlabeled point

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

Let $h' = \text{minimum error rate hypothesis choosing other label.}$

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

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

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

Accuracy Theorem: With high probability, the IWAL reduction has a similar error rate to supervised learning on $k$ points.
How should $p$ be chosen?

On the $k$th unlabeled point

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

Let $h' = \text{minimum error rate hypothesis choosing other label.}$

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

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

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

Accuracy Theorem: With high probability, the IWAL reduction has a similar error rate to supervised learning on $k$ points.

Efficiency Theorem: If there is a small disagreement coefficient $\theta$, the algorithm requires only $O\left(\theta \sqrt{k \log k}\right) + \text{a minimum due to noise.}$
Disagreement Coefficient

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

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

For any $\epsilon$ features $x$ are of interest if there exists a hypothesis $h$:

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

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

For any $\epsilon$ features $x$ are of interest if there exists a hypothesis $h$:

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

Disagreement coefficient is $\theta = \max_{\epsilon} \Pr(\text{interesting} \_ \_ \_ x)$
Disagreement coefficient: examples

Thresholds in $\mathbb{R}$, any data distribution. $\theta = 2$. Linear separators through the origin in $\mathbb{R}^d$, uniform data distribution. $\theta \leq \sqrt{d}$. Linear separators in $\mathbb{R}^d$, smooth data density bounded away from zero. $\theta \leq c(h^*)d$ where $c(h^*)$ is a constant depending on the target $h^*$. 
Disagreement coefficient: examples

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

\[ \theta = 2. \]
Disagreement coefficient: examples

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

  $$\theta = 2.$$ 

- Linear separators through the origin in $\mathbb{R}^d$, uniform data distribution.

  $$\theta \leq \sqrt{d}.$$
Disagreement coefficient: examples

- Thresholds in \( \mathbb{R} \), any data distribution.
  \[
  \theta = 2.
  \]

- Linear separators through the origin in \( \mathbb{R}^d \), uniform data distribution.
  \[
  \theta \leq \sqrt{d}.
  \]

- Linear separators in \( \mathbb{R}^d \), smooth data density bounded away from zero.
  \[
  \theta \leq c(h^*)d
  \]
  where \( c(h^*) \) is a constant depending on the target \( h^* \).
Decision Tree Experiments

**MNIST 3s vs 5s**

**KDDCUP99**

**KDDCUP99 (close-up)**

**MNIST multi-class (close-up)**
An Approximate IWAL

Let \( h(x) = \text{Learn}(S) \).
Let \( h'(x) = \text{Learn}_{h(x) \neq y}(S) \).
Claim: If \( \text{Learn} \) minimizes error rates, for all \( \epsilon > 0 \)

\[
\text{Learn}(S \cup (x, -h(x), t\Delta + \epsilon)) = h'(x)
\]

In other words \( t\Delta \) = importance weight required to change label for current \( x \).
An Approximate IWAL

Let \( h(x) = \text{Learn}(S) \).

Let \( h'(x) = \text{Learn}_{h(x) \neq y}(S) \).

Claim: If \text{Learn} minimizes error rates, for all \( \epsilon > 0 \)

\[
\text{Learn}(S \cup (x, -h(x), t\Delta + \epsilon)) = h'(x)
\]

In other words \( t\Delta = \text{importance weight required to change label for current } x \).

Using Vowpal Wabbit as base learner, estimate \( t \cdot \Delta \) as the number of gradient updates with \( x \) required for prediction to switch (from 0 to 1, or from 1 to 0).

\text{e.g.}, for importance weight-aware square-loss update:

\[
\Delta_t := \frac{1}{t \cdot \eta_t} \cdot \log \frac{\max\{h(x), 1 - h(x)\}}{0.5}
\]
Simulating active learning: (tuning parameter $C > 0$)

$vw$ --active_simulation --active_mellowness $C$

(increasing $C \rightarrow \infty =$ supervised learning)
Simulating active learning: (tuning parameter \( C > 0 \))
\[ \texttt{vw} --\texttt{active\_simulation} --\texttt{active\_mellowness} C \]
(increasing \( C \rightarrow \infty = \) supervised learning)

Deploying active learning:
\[ \texttt{vw} --\texttt{active\_learning} --\texttt{active\_mellowness} C --\texttt{daemon} \]
- \texttt{vw} interacts with an \texttt{active\_interactor} (ai)
- receives labeled and unlabeled training examples from ai over network
- for each unlabeled data point, \texttt{vw} sends back a query decision (and an importance weight if label is requested)
- ai sends labeled importance-weighted examples as requested
- \texttt{vw} trains using labeled importance-weighted examples
Active learning in Vowpal Wabbit

`active_interactor` implements the protocol by querying `vw` and receiving feedback. The diagram shows the interaction:

- `x_1` is queried with the probability of `1/p_1`.
- `(x_1, y_1, 1/p_1)` is sent to `vw`.
- `x_2` is queried and sent to `vw` with no further query.
- This process continues...

`active_interactor.cc` (in git repository) demonstrates how to implement this protocol.
Demonstration: RCV1

```
vw --active_simulation --active_mellowness 0.005 -b 22
--loss_function logistic --ngram 2 --skips 4 -c
rcv1.train.raw.txt
```
Demonstration: RCV1

```bash
vw --active_simulation --active_mellowness 0.005 -b 22
--loss_function logistic --ngram 2 --skips 4 -c
rcv1.train.raw.txt
```

1. 21K labels vs. 760K for supervised
2. 8s vs. 15s for supervised
3. Substantially better than uniform random sampling.
Fringe Benefits

This approach has many nice properties.
This approach has many nice properties.

1. Always consistent.
Fringe Benefits

This approach has many nice properties.

1. Always consistent.
2. Efficient.
   1. Label Efficient.
   2. Unlabeled data efficient.
   3. Computationally efficient.
Fringe Benefits

This approach has many nice properties.

1. Always consistent.
2. Efficient.
   1. Label Efficient.
   2. Unlabeled data efficient.
   3. Computationally efficient.
3. Compatible.
   1. With Online Algorithms
   2. With any optimization-style classification algorithm.
   3. With any Loss function
   4. With supervised learning
   5. With switching learning algorithms (!)

It works, empirically.
This approach has many nice properties.

1. Always consistent.
2. Efficient.
   1. Label Efficient.
   2. Unlabeled data efficient.
   3. Computationally efficient.
3. Compatible.
   1. With Online Algorithms
   2. With any optimization-style classification algorithm.
   3. With any Loss function
   4. With supervised learning
   5. With switching learning algorithms (!)
4. It works, empirically.
Outline

1. Importance weighting
2. Rare Classes
3. Cost Sensitive
How many labels for a rare class?
How many labels for a rare class?

Attenberg & Provost 2010: Search and insertion of labeled rare class examples helps.
Why does search help?

Potentially: exponential improvement in label complexity!
Why does search help?

Potentially: exponential improvement in label complexity!
Why does search help?

Potentially: exponential improvement in label complexity!
Why does search help?

Potentially: exponential improvement in label complexity!
Why does search help?

Potentially: exponential improvement in label complexity!
Why does search help?

Potentially: exponential improvement in label complexity!
Why does search help?

Potentially: exponential improvement in label complexity!
When does Search help?

Does searching for counterexamples help?
When does Search help?

Does searching for counterexamples help?
When does Search help?

Does searching for counterexamples help?

No!
Does searching for counterexamples help?

No!
When does Search help?

Does searching for counterexamples help?

No!
When does Search help?

Does searching for counterexamples help?

No!
Counterexample to version space instead!

Theorem: Search for version space counterexample can reduce Label calls exponentially by starting with simple set of classifiers and moving to more complex as they are proved inadequate.
Counterexample to version space instead!

Theorem: Search for version space counterexample can reduce Label calls exponentially by starting with simple set of classifiers and moving to more complex as they are proved inadequate.
Theorem: Search for version space counterexample can reduce Label calls exponentially by starting with simple set of classifiers and moving to more complex as they are proved inadequate.
Outline

1. Importance weighting
2. Rare Classes
3. Cost Sensitive
How to do Active Cost Sensitive Classification?

<table>
<thead>
<tr>
<th>Cost-sensitive multi-class classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution $D$ over $X \times [0, 1]^k$, where a vector in $[0, 1]^k$ specifies the cost of each of the $k$ choices.</td>
</tr>
<tr>
<td>Find a classifier $h : X \rightarrow {1, \ldots, k}$ minimizing the expected cost</td>
</tr>
<tr>
<td>$\text{cost}(h, D) = \mathbb{E}<em>{(x,c) \sim D} [c</em>{h(x)}]$.</td>
</tr>
</tbody>
</table>
How to do Active Cost Sensitive Classification?

Cost-sensitive multi-class classification

Distribution $D$ over $X \times [0, 1]^k$, where a vector in $[0, 1]^k$ specifies the cost of each of the $k$ choices.

Find a classifier $h : X \rightarrow \{1, \ldots, k\}$ minimizing the expected cost

$$\text{cost}(h, D) = \mathbb{E}_{(x,c) \sim D}[c_{h(x)}].$$

Should queries be per-example or per-cost?
Which class costs should be queried?

Minimum cost < smallest maximum cost

Cost difference matters
Which class costs should be queried?

- Minimum cost < smallest maximum cost
Which class costs should be queried?

1. Minimum cost < smallest maximum cost
2. Cost difference matters
Cost Overlapped Active Learning Results

Theorem: Works efficiently if (1) cost predictors factorize (2) squared loss optimizer is efficient (3) world is IID.
Theorem: Works efficiently if (1) cost predictors factorize (2) squared loss optimizer is efficient (3) world is IID.
Bibliography: Agnostic Active Learning history


IWAL I  A Beygelzimer, S Dasgupta, and J Langford, Importance Weighted Active Learning, ICML 2009.

IWAL II  A Beygelzimer, D Hsu, J Langford, T Zhang, Agnostic Active Learning Without Constraints, NIPS 2010.


Search II  A Beygelzimer, D Hsu, J Langford, C Zhang, Search Improves Label for Active Learning, NIPS 2016.