Active Learning

John Langford @ Microsoft Research

NYU Large Scale Learning Class, April 23

(Slides partially from Sanjoy Dasgupta, Daniel Hsu, Nikos Karamptziakis)

An instrument of mass machine learning

How can we formalize it's use?

Exploiting unlabeled data

A lot of unlabeled data is plentiful and cheap, eg.

- documents off the web
- speech samples
- images and video

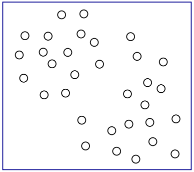
But labeling can be expensive.

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.



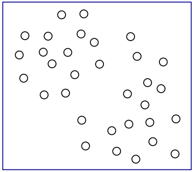
Unlabeled points

Exploiting unlabeled data

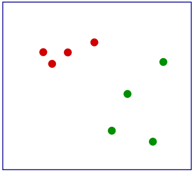
A lot of unlabeled data is plentiful and cheap, eg.

- documents off the web
- speech samples
- images and video

But labeling can be expensive.



Unlabeled points



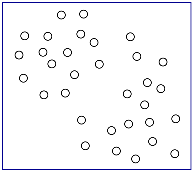
Supervised learning

Exploiting unlabeled data

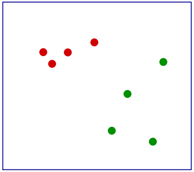
A lot of unlabeled data is plentiful and cheap, eg.

- documents off the web
- speech samples
- images and video

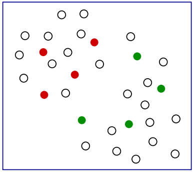
But labeling can be expensive.



Unlabeled points



Supervised learning



Semisupervised and active learning

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.

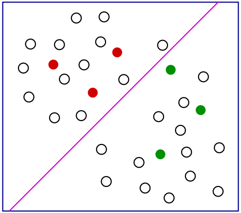
Typical heuristics for active learning

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,...)



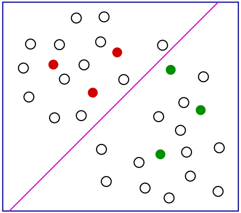
Typical heuristics for active learning

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

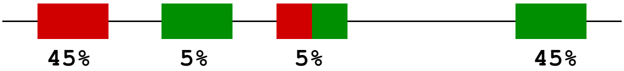
Start with a pool of unlabeled data

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:



Sampling bias

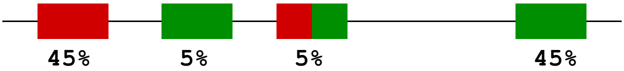
Start with a pool of unlabeled data

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.

Importance Weighted Active Learning via Reduction

$S = \emptyset$

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 kth 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.

How should p be chosen?

On the kth 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.

How should p be chosen?

On the kth 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.

How should p be chosen?

On the kth 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 $p = O\left(\frac{\log k}{\Delta^2 k}\right)$

How should p be chosen?

On the kth 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 $p = O\left(\frac{\log k}{\Delta^2 k}\right)$

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

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 θ, the algorithm requires only $O\left(\theta\sqrt{k \log k}\right)$ + 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 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 disagress 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 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 disagress with the best hypothesis, $h^*(x) \neq h(x)$.

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

Disagreement coefficient: examples

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^*.

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

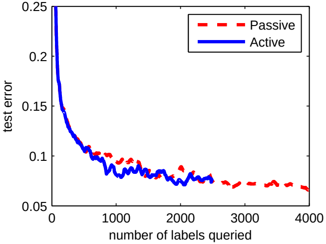
Proofs are complex, but rest on the solution to a Martingale Barrier Problem.

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

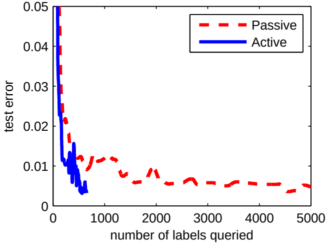
1. The average number of heads is small: $\frac{1}{k}\sum_{(h,p)\in S}\frac{h}{p} < 0.5$.
2. The number of coin flips is minimized: $\min \sum_{(h,p)\in S} p$.
3. The probability is nontrivial: $p > 0$.

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

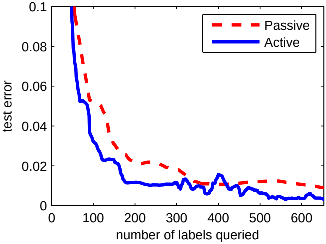
Decision Tree Experiments



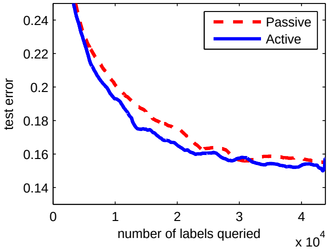
MNIST 3s vs 5s



KDDCUP99



KDDCUP99 (close-up)



MNIST multi-class (close-up)

An Approximate IWAL

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

$$\texttt{Learn}(S \cup (x, -h(x), t\Delta + \epsilon)) = h'(x)$$

In other words $t\Delta =$ importance weight required to change label for current x.

An Approximate IWAL

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

$$\texttt{Learn}(S \cup (x, -h(x), t\Delta + \epsilon)) = h'(x)$$

In other words $t\Delta =$ importance weight required to change label for current x.

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).

e.g., for importance weight-aware square-loss update:

$$\Delta_t := \frac{1}{t \cdot \eta_t} \cdot \log \frac{\max\{h(x),\ 1 - h(x)\}}{0.5}$$

Active learning in Vowpal Wabbit

Simulating active learning: (tuning paramter $C > 0$)

`vw --active_simulation --active_mellowness C`

(increasing $C \to \infty$ = supervised learning)

Active learning in Vowpal Wabbit

Simulating active learning: (tuning paramter $C > 0$)

`vw --active_simulation --active_mellowness C`

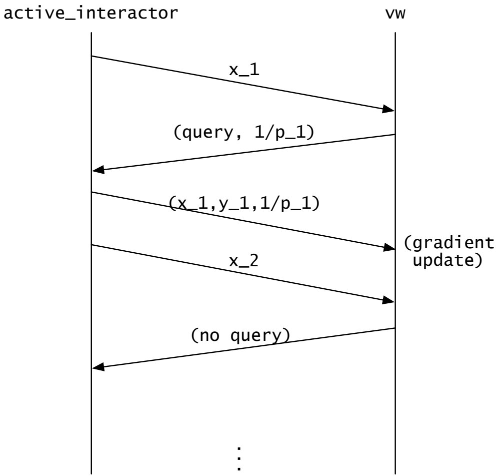
(increasing $C \to \infty$ = supervised learning)

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.

Demonstration: RCV1

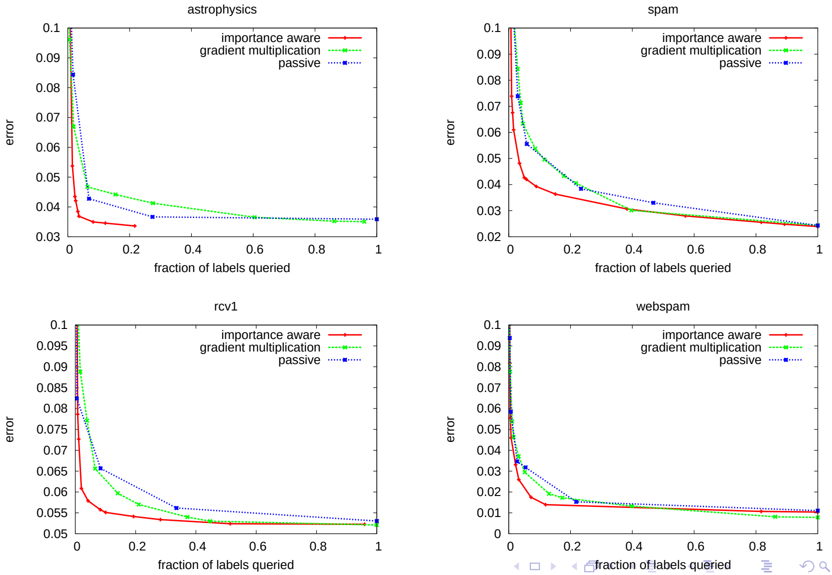
```
vw --active_simulation --active_mellowness 0.005 -b 22
--loss_function logistic --ngram 2 --skips 4 -c
rcv1.train.raw.txt
```

Demonstration: RCV1

```
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.

Online Linear Learning results



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. 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 (!)

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 (!)
4. It works, empirically.

Are we done?

Many other issues come up when trying to use human labelers. At NYU, there is some good work by people in Wharton on this.

Bibliography

Possibility N Balcan, A Beygelzimer, J Langford, Agnostic Active Learning. ICML 2006.

Noise M Kaariainen, Active Learning in the Non-realizable Case, ALT 2006.

Disagree S Hanneke. A Bound on the Label Complexity of Agnostic Active Learning. ICML 2007.

Online S Dasgupta, D Hsu, and C Monteleoni. A general agnostic active learning algorithm. NIPS 2007.

Weights F Bach. Active learning for misspecified generalized linear models. NIPS 2007.

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

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

Practical N Karampatziakis and J Langford, Importance Weight Aware Gradient Updates, UAI 2011.