Efficient Active Learning

Alina Beygelzimer Daniel Hsu
Nikos Karampatziakis John Langford
Tong Zhang

July 2nd 2011

Active Learning

Labeling can be expensive.

Can interaction help us learn effectively?

The Active Learning Setting

Repeatedly:

1. Observe unlabeled example x and predict.
2. Decide whether to query label.
3. If yes, observe label y.

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

Exploration vs Exploitation

Good active learning algorithm must balance between:

- Exploration: querying the label of new point
- Exploitation: predicting using current hypothesis

Importance Weighted Active Learning

$S = \emptyset$

While (unlabeled examples remain)

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

Theorem: (Consistency, BDL 2009) For all methods choosing $p > 0$, the algorithm is consistent.

- Consistency implies no brittleness.
- Importance weights allow sample reuse.

How should p be chosen?

On the kth unlabeled point

Let $\hat{e}(h, S) = \frac{1}{k} \sum_{(x,y,i) \in S} i\mathbb{I}(h(x) \neq y)$

How should p be chosen?

On the kth unlabeled point

Let $\hat{e}(h, S) = \frac{1}{k} \sum_{(x,y,i) \in S} i \mathbb{I}(h(x) \neq y)$

Let $h = \text{argmin}_{\bar{h} \in \mathcal{H}} \hat{e}(\bar{h}, S)$. ERM hypothesis

How should p be chosen?

On the kth unlabeled point

Let $\hat{e}(h, S) = \frac{1}{k} \sum_{(x,y,i) \in S} i \mathbb{I}(h(x) \neq y)$

Let $h = \text{argmin}_{\bar{h} \in \mathcal{H}} \hat{e}(\bar{h}, S)$. ERM hypothesis

Let $h' = \text{argmin}_{\bar{h} \in \mathcal{H} : \bar{h}(x) \neq h(x)} \hat{e}(\bar{h}, S)$. Alternative

How should p be chosen?

On the kth unlabeled point

Let $\hat{e}(h, S) = \frac{1}{k} \sum_{(x,y,i) \in S} i \mathbb{I}(h(x) \neq y)$

Let $h = \mathrm{argmin}_{\bar{h} \in \mathcal{H}} \hat{e}(\bar{h}, S)$. ERM hypothesis

Let $h' = \mathrm{argmin}_{\bar{h} \in \mathcal{H} : \bar{h}(x) \neq h(x)} \hat{e}(\bar{h}, S)$. Alternative

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

How should p be chosen?

On the kth unlabeled point

Let $\hat{e}(h, S) = \frac{1}{k} \sum_{(x,y,i) \in S} i \mathbb{I}(h(x) \neq y)$

Let $h = \text{argmin}_{\bar{h} \in \mathcal{H}} \hat{e}(\bar{h}, S)$. ERM hypothesis

Let $h' = \text{argmin}_{\bar{h} \in \mathcal{H}: \bar{h}(x) \neq h(x)} \hat{e}(\bar{h}, S)$. Alternative

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

Theorems

(Accuracy, BHLZ 2010) With high probability, the IWAL reduction has a similar error rate to supervised learning on k points.

(Efficiency, BHLZ 2010) If there is a small disagreement coefficient θ, the queried labels are only $O\left(\theta\sqrt{k \log k}\right) +$ a minimum due to noise (Kaariainen 2006).

Active Learning with Online Gradient

- ERM $\approx$ Supervised Learning $=$ Tractable.

Active Learning with Online Gradient

- ERM $\approx$ Supervised Learning $=$ Tractable.
- Assume ERM hypothesis $h =$ current iterate w_k

Active Learning with Online Gradient

- ERM $\approx$ Supervised Learning $=$ Tractable.
- Assume ERM hypothesis $h =$ current iterate w_k
- Alternative hypothesis $h' = w_k + s_k x_k$

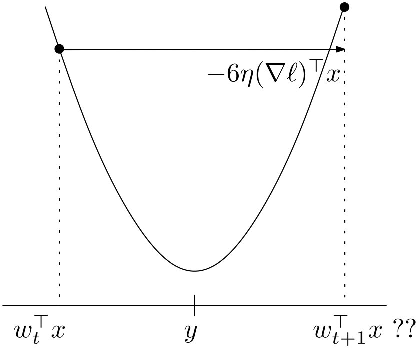
Active Learning with Online Gradient

- ERM $\approx$ Supervised Learning $=$ Tractable.
- Assume ERM hypothesis $h =$ current iterate w_k
- Alternative hypothesis $h' = w_k + s_k x_k$
- $s_k =?$ h' not necessary

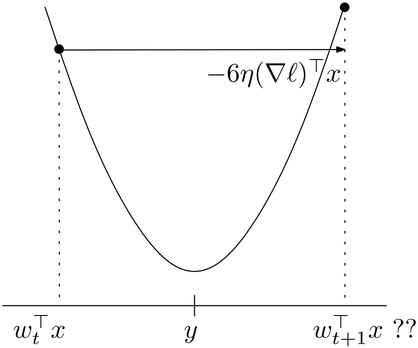
Active Learning with Online Gradient

- ERM $\approx$ Supervised Learning $=$ Tractable.
- Assume ERM hypothesis $h =$ current iterate w_k
- Alternative hypothesis $h' = w_k + s_k x_k$
- $s_k =?$ h' not necessary
- Only need: Δ difference in error rates

Learning with importance weights



Learning with importance weights

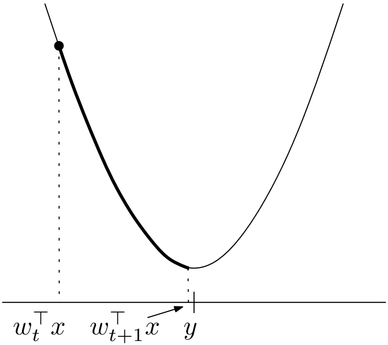


Large importance weights (small p's) tricky

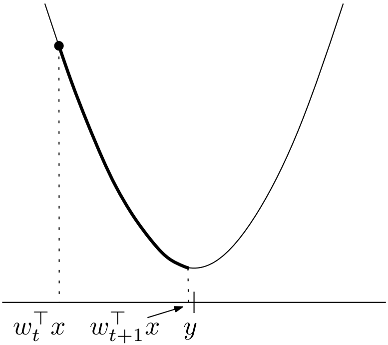
Principle

Having an example with importance weight *i* should be equivalent to having the example *i* times in the dataset.

Learning with importance weights



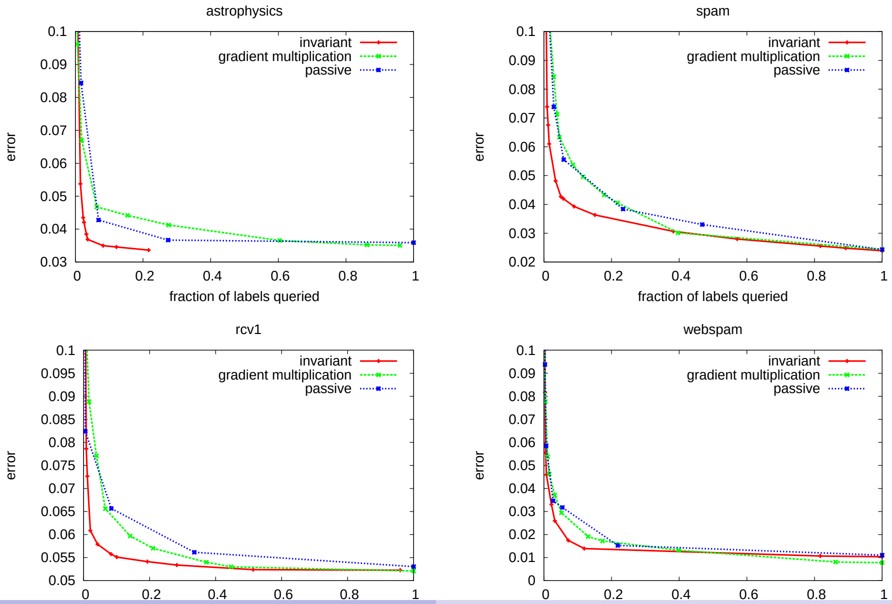
Learning with importance weights



$$w_{t+1} = w_t - \frac{w_t^\top x - y}{x^\top x}\left(1 - e^{-i\eta x^\top x}\right) x$$

Closed form for logistic, hinge, many other losses.

Online Gradient Descent Results



How fast is it?

- As fast as (passive) online gradient descent
  - Train on RCV1 ($\approx$780K docs 77 features/doc): 2.6 sec.
  - Passive online gradient descent takes 2.5 sec
  - 91K queries (11%)

How fast is it?

- As fast as (passive) online gradient descent
  - Train on RCV1 (≈780K docs 77 features/doc): 2.6 sec.
  - Passive online gradient descent takes 2.5 sec
  - 91K queries (11%)
- Faster!
  - On RCV1 with 3500 features/doc 2 min.
  - Passive gradient descent takes 3 min.
  - 89K queries

Conclusions

- A consistent active learning algorithm
- A reduction. Plug in your learner
- Online gradient descent (C4.5 poster)
- For more details

BHLZ 2010 Agnostic Active Learning without Constraints, NIPS.
KL 2011 Online Importance Weight Aware Updates, UAI.

  - Check implementation in Vowpal Wabbit
    http://github.com/JohnLangford/vowpal_wabbit