Importance Weighted Active Learning (IWAL) [BDL'09]

$S=\emptyset$

For $t = 1, 2, \ldots$ until no more unlabeled data

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

Learn = base supervised learner

$$\mathbb{E}\Big[\underbrace{\frac{1}{n}\sum_{t=1}^{n} \mathbb{1}(\text{got label } y_t) \cdot \frac{1}{p_t} \cdot \mathbb{1}(h(x_t) \neq y_t)}_{\text{importance weighted error estimate}}\Big] = \Pr(h(X) \neq Y).$$

New instantiation of IWAL

[BHLZ'10, this conference]: strong consistency / label efficiency guarantees by using

$$p_t = \min \left\{ 1,\ C \cdot \left(\frac{1}{\Delta_t^2} \cdot \frac{\log t}{t-1} \right) \right\}$$

where Δ_t = increase in training error rate if learner is forced to change its prediction on the new unlabeled point x_t.

New instantiation of IWAL

[BHLZ'10, this conference]: strong consistency / label efficiency guarantees by using

$$p_t = \min\left\{1,\ C \cdot \left(\frac{1}{\Delta_t^2} \cdot \frac{\log t}{t-1}\right)\right\}$$

where Δ_t = increase in training error rate if learner is forced to change its prediction on the new unlabeled point x_t.

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

New instantiation of IWAL

[BHLZ'10, this conference]: strong consistency / label efficiency guarantees by using

$$p_t = \min\left\{1,\ C \cdot \left(\frac{1}{\Delta_t^2} \cdot \frac{\log t}{t-1}\right)\right\}$$

where Δ_t = increase in training error rate if learner is forced to change its prediction on the new unlabeled point x_t.

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

e.g., for importance weight-aware square-loss update [KL'10, arxiv]:

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

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`

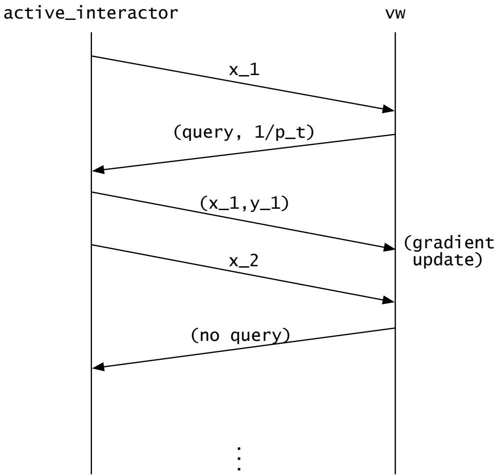
(increasing $C \to \infty$ = supervised learning)

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.

Active learning simulation results

RCV1 (text binary classification task):

training:

```
vw --active_simulation --active_mellowness 0.000001
   -d rcv1-train -f active.reg -l 10 --initial_t 10
```

number of examples = 781265

total queries = 98074 (*i.e.*, $< 13\%$ of the examples)

(caveat: progressive validation loss not reflective of test loss)

Active learning simulation results

RCV1 (text binary classification task):

training:

```
vw --active_simulation --active_mellowness 0.000001
   -d rcv1-train -f active.reg -l 10 --initial_t 10
```

number of examples = 781265
total queries = 98074 (*i.e.*, $< 13\%$ of the examples)
(caveat: progressive validation loss not reflective of test loss)

testing:

```
vw -t -d rcv1-test -i active.reg
```

average loss = 0.04872
(average loss of supervised learner: 0.055)

Active learning simulation results

More results from [KL'10, arxiv]: