Active Learning

Matt Clark

Daniel Gu

Matt Morgan

Keegan Ryan

Motivation

- Labeled training data for supervised learning is hard and expensive to obtain
- It is often the case that we have few labeled training examples, but many unlabeled training examples
- Given these constraints, we want to study how many labeled training examples we really need (under a possibly adaptive strategy) in order to get a “good enough” learner

Active Learning vs. Passive Learning

- In *passive learning*, the learner simply accepts labeled training examples and trains on them all at once
- By contrast, in *active learning*, the learner receives unlabeled training examples and can request labels for training examples it sees

The PAC Model

- Intuitively, our goal is to find active learners “as good” as passive learners
- We need a formal notion of “as good” so we can get rigorous guarantees
- We will work within the *probably approximately correct (PAC)* model

Definitions

- Let f: X → Y, the *target function*, be drawn from a family F known to the learning algorithm
- Our learner is given training examples $(x_1, f(x_1)), \ldots, (x_m, f(x_m))$ drawn from X × Y drawn i.i.d. from a probability distribution D
- Our learner produces a hypothesis h based on this data

Definition 1 ***Error in the PAC Model***

The error with respect to the produced hypothesis h is defined as

$$E(h) = \sum_{x \in h \Delta f} D(x)$$

where Δ is the symmetric difference; that is, the error is the probability that h and f will disagree on a sample randomly drawn from D.

Definition 2 (ϵ, δ)***-PAC Learning***

A learner achieves (ϵ, δ)-PAC Learning *if the produced hypothesis h if with probability $1 - \delta$ (with respect to D) h achieves an error $E(h) \leq \epsilon$.*

Sample and Label Complexity

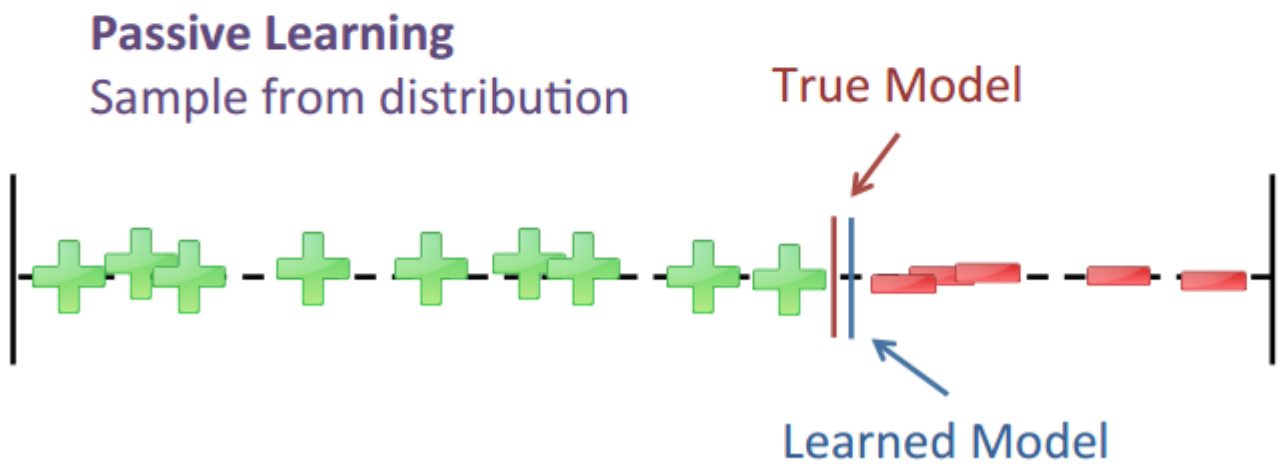
- We can then think of the *sample complexity* of a passive learning task as the number of training samples it takes to get a (ε, δ)-PAC learner
- Analogously, we can think of the *label complexity* of an active learning task to be the number of labels an active learning algorithm needs to request to get a (ε, δ) PAC-learner

Models of Active Learning

- In the *membership query model*, we are allowed to generate our own training examples are give them to the oracle to label
- In the *streaming selective sampling model*, we receive training examples one by one from a stream, and can choose whether to request a label or not
- In the *pool-based sampling model*, we have a small pool of labeled training examples and a large pool of unlabeled training examples, and we can choose unlabeled examples to label from the pool

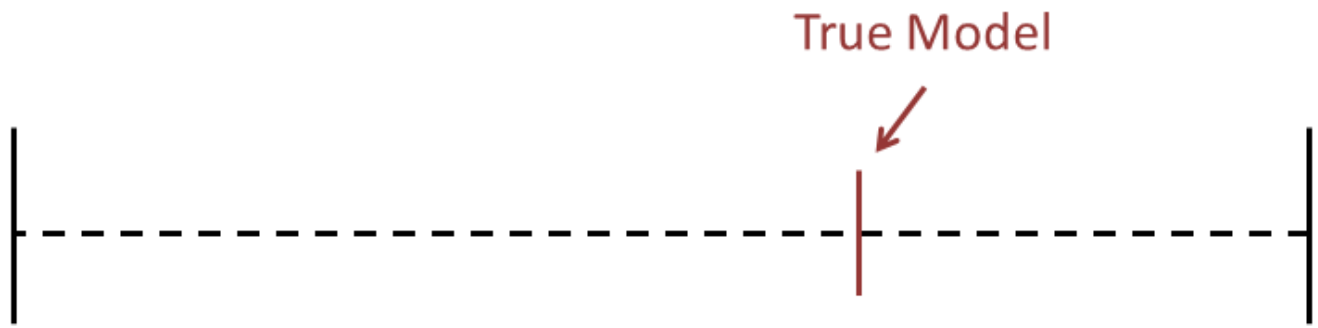
Simple Example

- 1 feature
- Learn threshold function



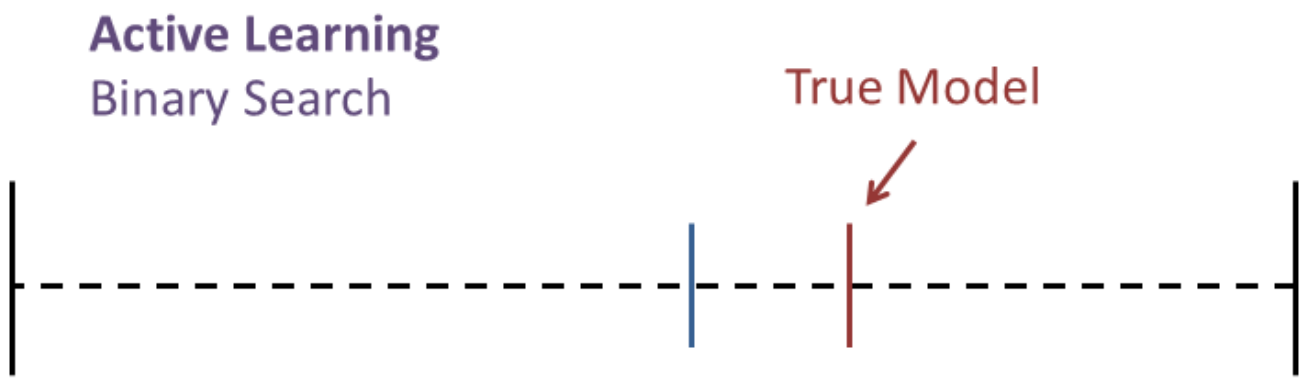
Simple Example

- 1 feature
- Learn threshold function



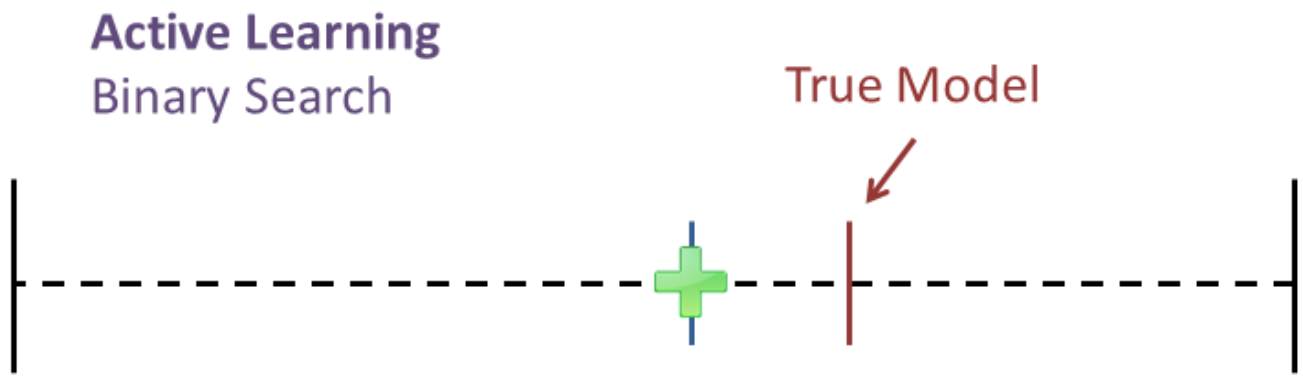
Simple Example

- 1 feature
- Learn threshold function



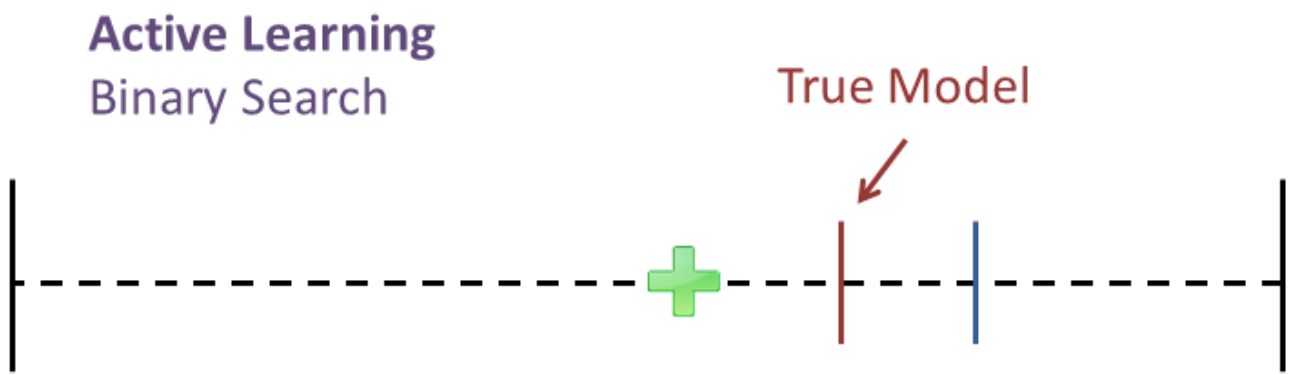
Simple Example

- 1 feature
- Learn threshold function



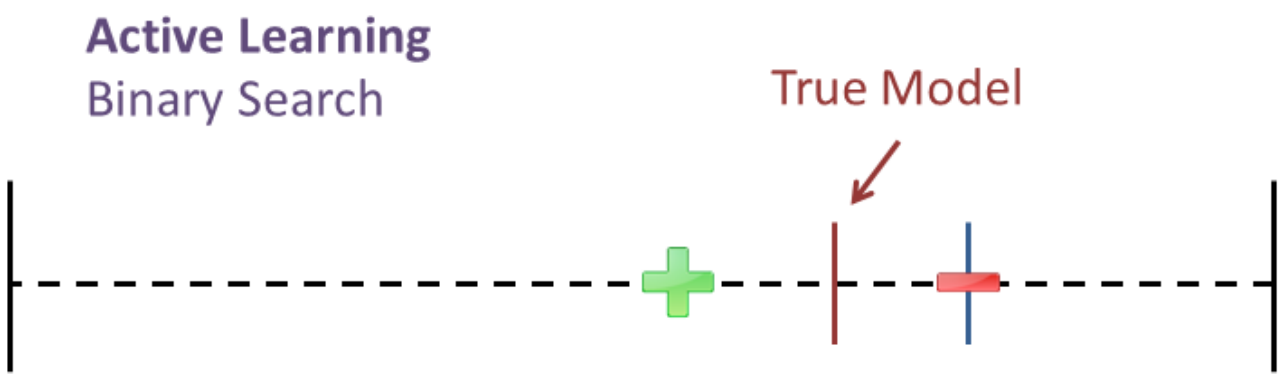
Simple Example

- 1 feature
- Learn threshold function



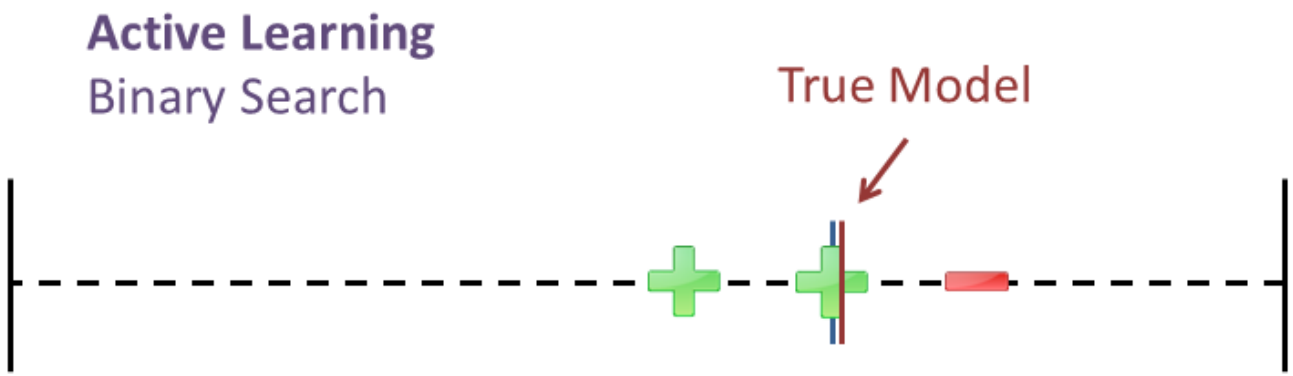
Simple Example

- 1 feature
- Learn threshold function



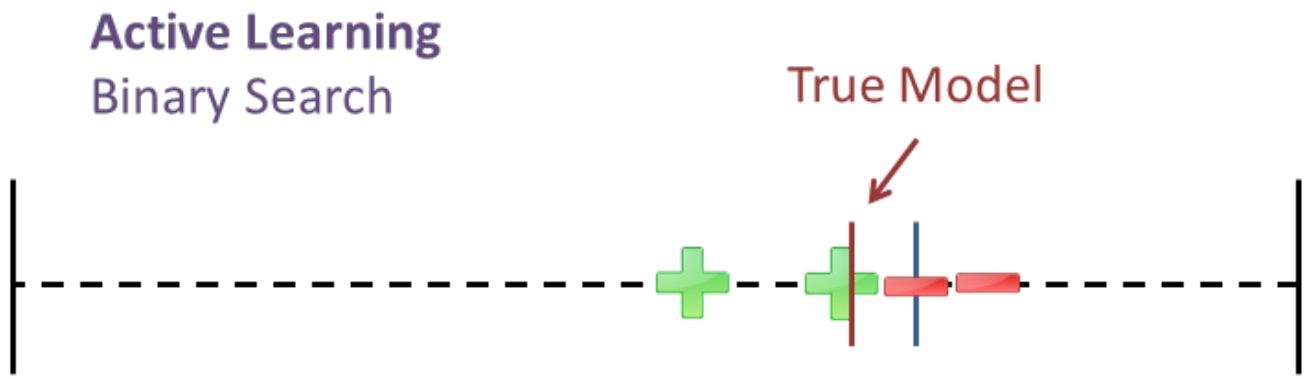
Simple Example

- 1 feature
- Learn threshold function



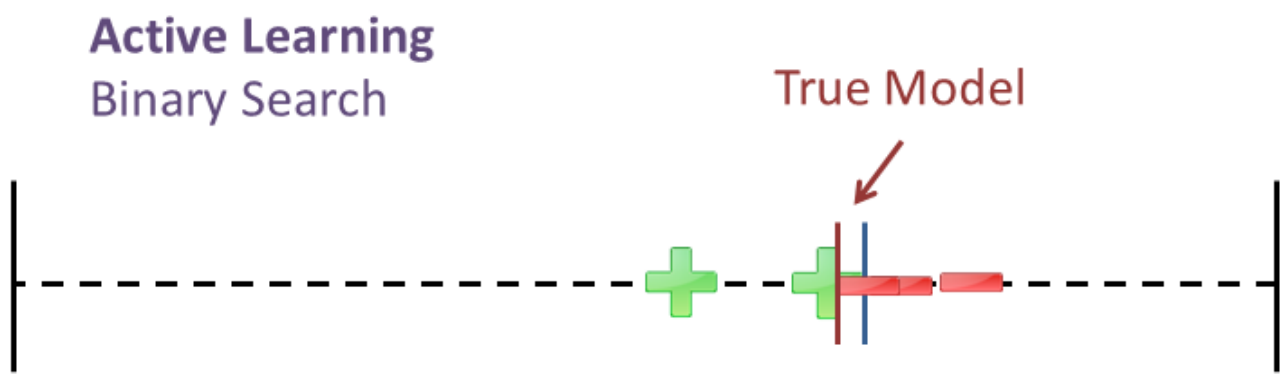
Simple Example

- 1 feature
- Learn threshold function



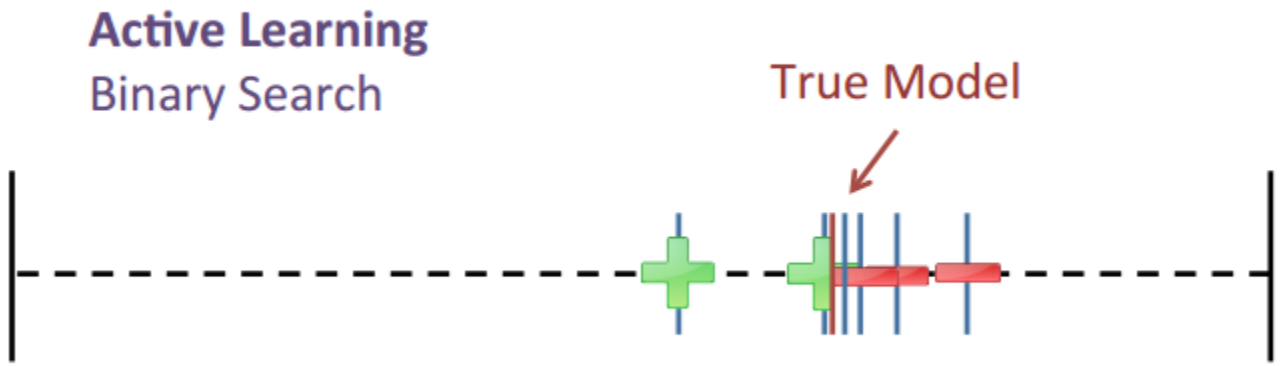
Simple Example

- 1 feature
- Learn threshold function



Simple Example

- 1 feature
- Learn threshold function



We get an improvement from O(1/e) to O(log(1/e)!

Realizable vs Agnostic

- We further need to distinguish between two settings: the *realizable* setting and the agnostic setting
- In the realizable setting, we assume our hypothesis contains a hypothesis which perfectly categorizes the data
- In the agnostic setting, we have no guarantee that our hypothesis has a no-loss predictor

Agnostic Active Learning

- Finding algorithms which are consistent in the agnostic case is a central difficulty in active learning
- Some results in the agnostic setting are known, for example Hannecke 2007 for the A^2 algorithm

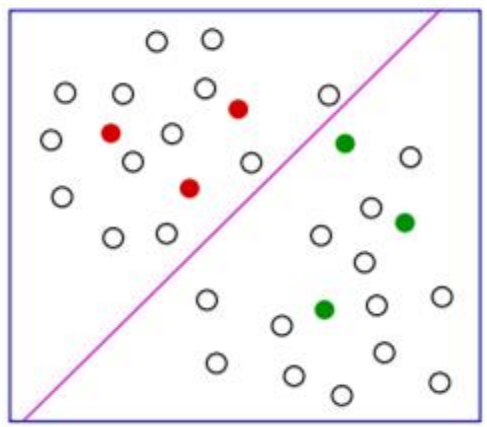
Typical heuristics for active learning

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

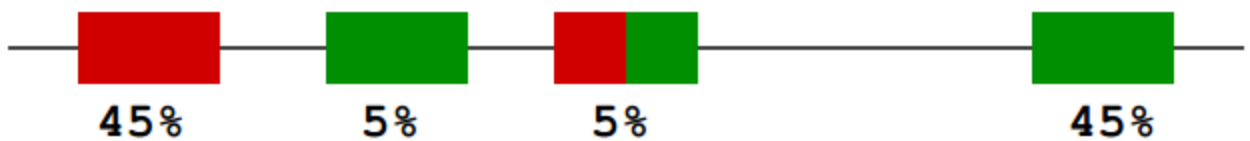
Start with a pool of unlabeled data

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

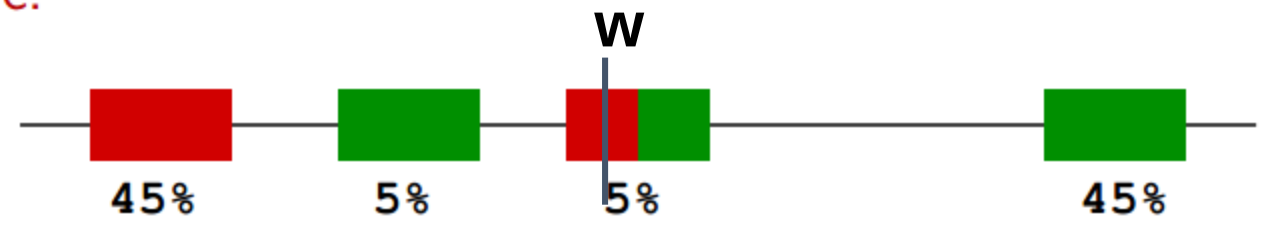
Start with a pool of unlabeled data

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:



Sampling bias

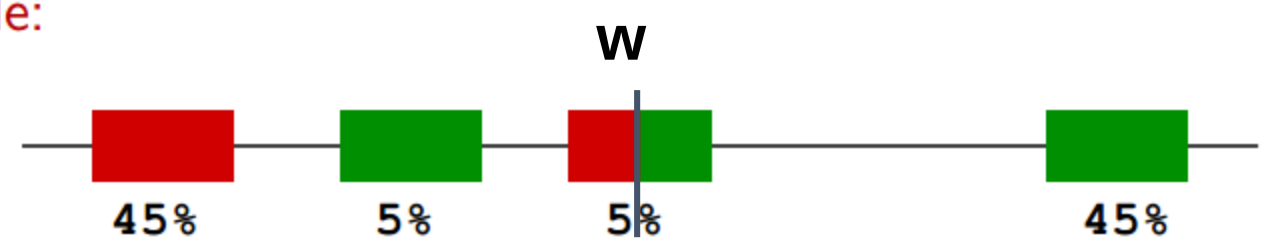
Start with a pool of unlabeled data

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

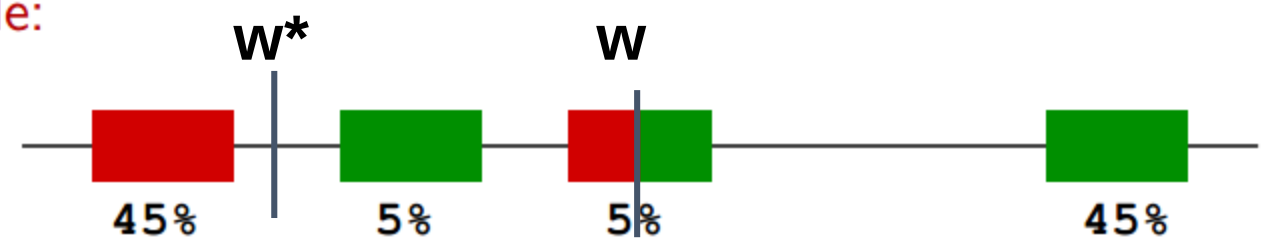
Start with a pool of unlabeled data

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:



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!*

Manifestation in practice, eg. Schutze et al 03.

Notation

- We will stick to the notation of Beygelzeimer et al 2009.
- Let X be an *input space*, Y be the *label space*, and Z be a *prediction space*
- Training examples are drawn i.i.d. from $X \times Y$ according to a probability distribution D
- Learning algorithm outputs a hypothesis from the hypothesis class $H = \{h : X \rightarrow Z\}$
- We have a loss function $l: Z \times Y \rightarrow R^{+}$

Algorithm 1 IWAL (subroutine rejection-threshold)

Set $S_0 = \emptyset$.

For t from $1, 2, ...$ until the data stream runs out:

1. Receive x_t .
2. Set p_t = rejection-threshold$(x_t, \{x_i, y_i, p_i, Q_i : 1 \leq i < t\})$.
3. Flip a coin $Q_t \in \{0, 1\}$ with $\mathbf{E}[Q_t] = p_t$. If $Q_t = 1$, request y_t and set $S_t = S_{t-1} \cup \{(x_t, y_t, 1/p_t)\}$, else $S_t = S_{t-1}$.
4. Let $h_t = \arg\min_{h \in H} \sum_{(x,y,c) \in S_t} c \cdot l(h(x), y)$.

Importance weighting

- Streaming setting
- Assume examples drawn $i.i.d$ from D

Definitions:

$$L(h) = E_{(x,y)\sim D}[l(h(x), y)]$$

$$L_T(h) = \frac{1}{T}\sum_{t=1}^{T}\frac{Q_t}{p_t}l(h(x_t, y_t)$$

Importance weighting

- Streaming setting
- Assume examples drawn $i.i.d$ from D

Definitions:

$$L(h) = E_{(x,y)\sim D}[l(h(x), y)]$$

$$L_T(h) = \frac{1}{T}\sum_{t=1}^{T}\frac{Q_t}{p_t}l(h(x_t, y_t)$$

Note that when $E[Q_t] = p_t$, $E[L_T(h)] = L(h)$

Extended class of loss functions $\phi : \mathbb{R} \to \mathbb{R}$

We are considering the subclass of loss functions that can be represented by $\phi(yz)$, where $Y = \{-1, 1\}$ and $Z \subset \mathbb{R}$.

Includes:

- 0 - 1 loss $\mathbb{1}(yz < 0)$
- Squared loss $(1 - yz)^2$
- Logistic loss $ln(1 + e^{-yz})$
- Hinge loss $(1 - yz)_+$

If Z is bounded, $\phi(yz)$ is bounded. Assume normalized: $\phi(yz) \in [0, 1]$.

Theorem 1

For all distributions D, for all finite hypothesis classes H, for any $\delta > 0$, if there is a constant $p_{min} > 0$ such that $p_t \geq p_{min}$ for all $1 \leq t \leq T$,

Then:

$$P\left[\arg\max_{h \in H} |L_T(h) - L(h)| > \frac{\sqrt{2}}{p_{min}} \sqrt{\frac{\ln |H| + \ln \frac{2}{\delta}}{T}}\right] < \delta$$

Comparison with other bounds

Supervised: Chernoff Occam's Razor Bound (Langford, 2005)

$$P\left[\arg\max_{h\in H} : |L_T(h) - L(h)| > \sqrt{\frac{\ln|H| + \ln\frac{1}{\delta}}{2T}}\right] < \delta$$

Theorem 1:

$$P\left[\arg\max_{h\in H} |L_T(h) - L(h)| > \frac{\sqrt{2}}{p_{min}}\sqrt{\frac{\ln|H| + \ln(2) + \ln\frac{1}{\delta}}{T}}\right] < \delta$$

Rejection-threshold

Recall from Algorithm 1:

$$\text{Set } p_t = \text{rejection-threshold}(x_t, \{x_i, y_i, p_i, Q_i : 1 \leq i < t\}).$$

Algorithm 2 loss-weighting $(x, \{x_i, y_i, p_i, Q_i : i < t\})$

1. Initialize $H_0 = H$.
2. Update $L^*_{t-1} = \min_{h \in H_{t-1}} \frac{1}{t-1} \sum_{i=1}^{t-1} \frac{Q_i}{p_i} l(h(x_i), y_i)$

   Set H_t to

$$\left\{ h \in H_{t-1} : \frac{1}{t-1} \sum_{i=1}^{t-1} \frac{Q_i}{p_i} l(h(x_i), y_i) \leq L^*_{t-1} + \Delta_{t-1} \right\}$$

3. Return $p_t = \max_{f,g \in H_t, y \in Y} l(f(x), y) - l(g(x), y)$.

Theorem 2

Pick any D, finite H

Let $h^* \in H$ be a minimizer of L, and let h_T be a minimizer of L_T
Define $L_T^* \equiv L_T(h_t)$

Then for any $\delta > 0$, w.p. at least $1 - \delta$, for any T:

i $h^* \in H_T$

ii $f, g \in H \implies L(f) - L(g) \leq 2\Delta_{T-1}$

Plugging in h_T and h^* gives $L(h_T) - L(h^*) \leq 2\Delta_{T-1}$

Theorem 2: Proof

Lemma 3

For all D, H, $\delta > 0$, w.p. at least $1 - \delta$, for all T, and all $f, g \in H_T$,

$$|L_T(f) - L_T(g) - L(f) + L(g)| \leq \Delta_T$$

w.p. at least $1 - \delta$, where $\Delta_t = \sqrt{\frac{8}{t} \ln(2t(t+1)|H|^2/\delta)}$

Part (i): $h^* \in H_T$

(**Lemma 3** : $|L_T(f) - L_T(g) - L(f) + L(g)| \leq \Delta_T$ w.p. $1 - \delta$)

- $H_1 = H_0 = H \implies T = 1$ holds
- $T \to T + 1$
  - $\star$ $L_T(h^*) - L_T(h_T) \leq L(h^*) - L(h_T) + \Delta_T$
    - $*$ Recall: $L(h) = E_{(x,y)\sim D}[l(h(x), y)]$

      $\implies L(h^*) - L(h_T) \leq 0$
  - $\star$ $L_T(h^*) - L_T(h_T) \leq \Delta_T$

Part (i): $h^* \in H_T$

(**Lemma 3** : $|L_T(f) - L_T(g) - L(f) + L(g)| \leq \Delta_T$ w.p. $1 - \delta$)

- $H_1 = H_0 = H \implies T = 1$ holds
- $T \to T + 1$
  - $\star$ $L_T(h^*) - L_T(h_T) \leq L(h^*) - L(h_T) + \Delta_T$
    - $*$ Recall: $L(h) = E_{(x,y)\sim D}[l(h(x), y)]$

      $\implies L(h^*) - L(h_T) \leq 0$
  - $\star$ $L_T(h^*) - L_T(h_T) \leq \Delta_T$
  - $\star$ $L_T(h^*) \leq L_T^* + \Delta_T$

Recall inclusion criterion: $H_t = \{h \in H_{t-1} : L_t(h) \leq L_{t-1}^* + \Delta_{t-1}\}$

Part (ii): $L(f) - L(g) \leq 2\Delta_{T-1}$

(**Lemma 3** : $|L_T(f) - L_T(g) - L(f) + L(g)| \leq \Delta_T$ w.p. $1 - \delta$)

Let h_T minimize L_T over H_T, and denote $L_T^* = L_T(h_T)$

- $L(f) - L(g) \leq L_{T-1}(f) - L_{T-1}(g) + \Delta_{T-1}$
  - $\star$ $f \in H_T \implies L_{T-1}(f) \leq L_{T-1}^* + \Delta_{T-1}$
  - $\star$ $-L_{T-1}(g) \leq -L_{T-1}^*$
- $L_{T-1}(f) - L_{T-1}(g) + \Delta_{T-1} \leq L_{T-1}^* + \Delta_{T-1} - L_{T-1}^* + \Delta_{T-1} = 2\Delta_{T-1}$
- $L(f) - L(g) \leq 2\Delta_{T-1}$

Part (ii): $L(f) - L(g) \leq 2\Delta_{T-1}$

(**Lemma 3** : $|L_T(f) - L_T(g) - L(f) + L(g)| \leq \Delta_T$ w.p. $1 - \delta$)

Let h_T minimize L_T over H_T, and denote $L_T^* = L_T(h_T)$

- $L(f) - L(g) \leq L_{T-1}(f) - L_{T-1}(g) + \Delta_{T-1}$
  - $\star$ $f \in H_T \implies L_{T-1}(f) \leq L_{T-1}^* + \Delta_{T-1}$
  - $\star$ $-L_{T-1}(g) \leq -L_{T-1}^*$
- $L_{T-1}(f) - L_{T-1}(g) + \Delta_{T-1} \leq L_{T-1}^* + \Delta_{T-1} - L_{T-1}^* + \Delta_{T-1} = 2\Delta_{T-1}$
- $L(f) - L(g) \leq 2\Delta_{T-1}$

  Since $h_T, h^* \in H_T$, $\implies L(h_T) \leq L(h^*) + 2\Delta_{T-1}$

Analysis

The theoretical results of the paper

What do all the theorems mean?

Goal:

- Talk about the lower bound
- Sketch the proof for the upper bound
- Revisit some of the steps for the upper bound

Lower Bound on Requested Labels

What's the best performance we can get?

Theorem 12 addresses this.
No matter the active learner, we can always create a dataset that:

- has $L^* > 0$ optimal error
- must make at least TL^* queries

The term that's linear in T must always be there.

Lemma 13 is used to help this proof by construction.

Upper Bound on Label Complexity (Thm 11)

Querying the labels may be costly. We want algorithms that query as infrequently as possible, while still performing as well as passive learning (Theorem 2).

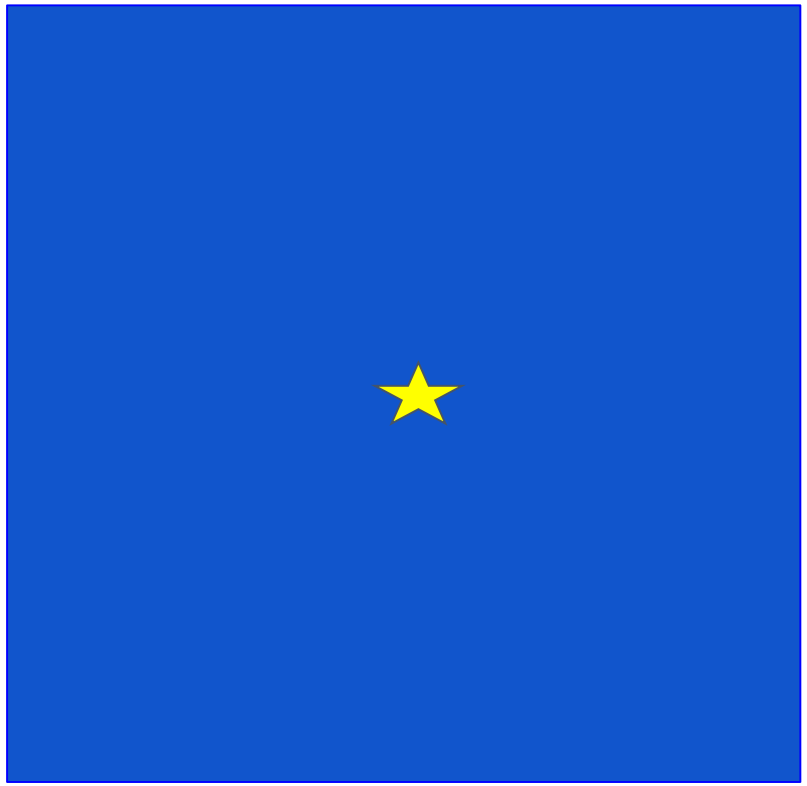
We see $E[\#\text{ requested}] \leq 4\theta K_l(TL^* + O(\sqrt{T}(\ln(|H|T/\delta))))$

Here we walk through a sketch of the proof of Theorem 11. This will give us a probable upper bound on the expected number of requested labels in the IWAL algorithm.

Recall

There is a call to Algorithm 2 at each time step, giving a probability p_t for every time step.

The expected number of requested labels is the sum of all p_t.



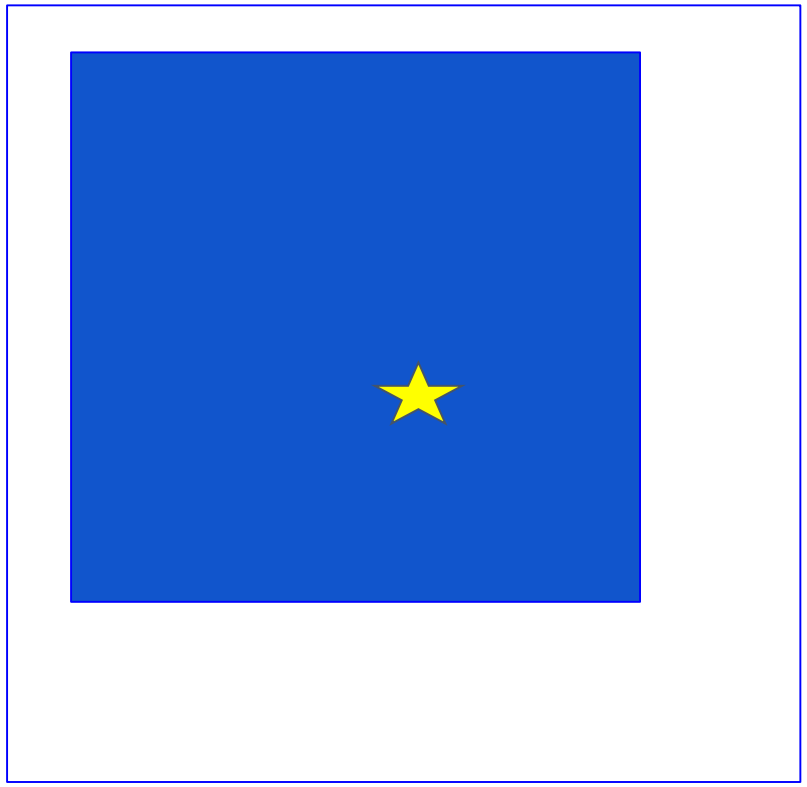
p_1 = ...

Set of hypotheses $\mathbf{H_t}$ and optimal hypothesis **h***

Recall

There is a call to Algorithm 2 at each time step, giving a probability p_t for every time step.

The expected number of requested labels is the sum of all p_t.



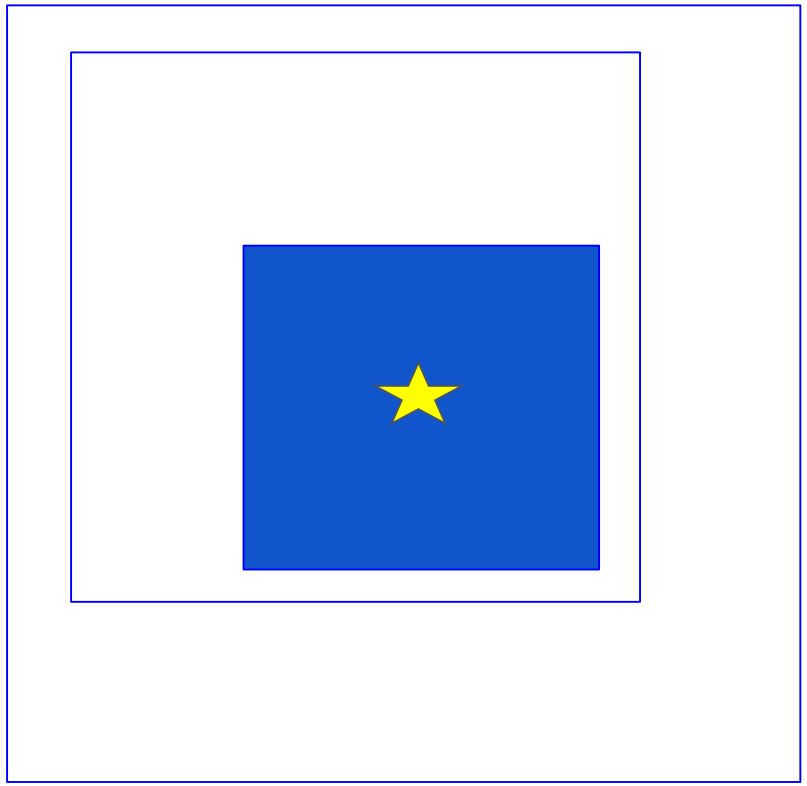
p_1 = …
p_2 = …

Set of hypotheses $\mathbf{H_t}$ and optimal hypothesis **h***

Recall

There is a call to Algorithm 2 at each time step, giving a probability p_t for every time step.

The expected number of requested labels is the sum of all p_t.

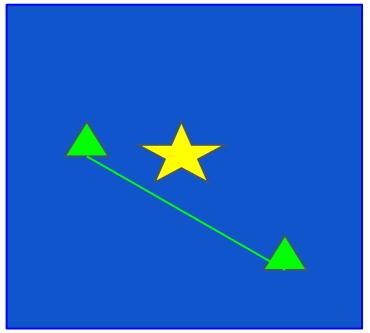


p_1 = …
p_2 = …
p_3 = …

Set of hypotheses $\mathbf{H_t}$ and optimal hypothesis **h***

Upper Bound on p_t

The value of p_t comes from the maximum difference in loss between two hypotheses in H_t.



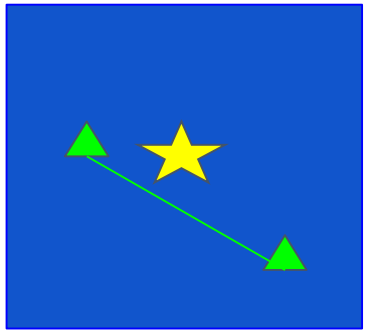
$$E_x[p_t] = E_x[\max_{f,g \in H\_t,\ y \in Y} l(f(x),y) - l(g(x),y)]$$

Upper Bound on p_t

$$E_x[p_t] = E_x[\max_{f,g \in H\_t,\ y \in Y} l(f(x),y) - l(g(x),y)]$$

The RHS looks a lot like the LHS of the definition of the disagreement coef.:

$$E[\sup_{h \in B(h^*,r)} \sup_y |l(h(x),y) - l(h^*(x),y)|]$$



Disagreement Coefficient

Define the metric $\rho(f,g) = E[\max_y |l(f(x),y) - l(g(x),y)|]$

This gives us the distance between two hypotheses.
If we pick an input at random, how bad can we expect the difference in loss to be?
Hypotheses that are close will usually have similar loss, no matter what the true label is.

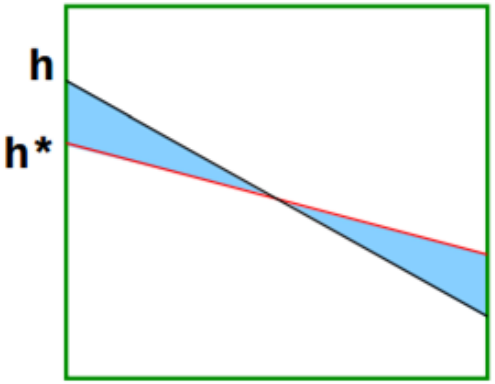
Image source:
http://hunch.net/~active_learning/active_learning_icml09.pdf

Disagreement Coefficient

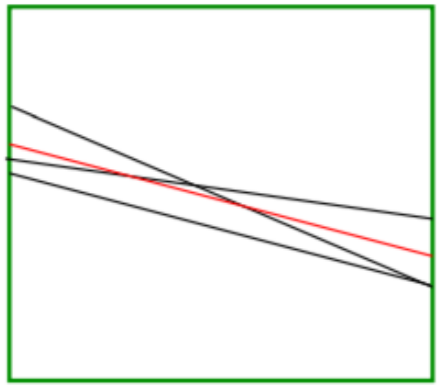
Smallest θ such that, for all r,

$$E[\sup_{h \in B(h^*,r)} \sup_y |l(h(x),y) - l(h^*(x),y)|] \leq \theta r$$

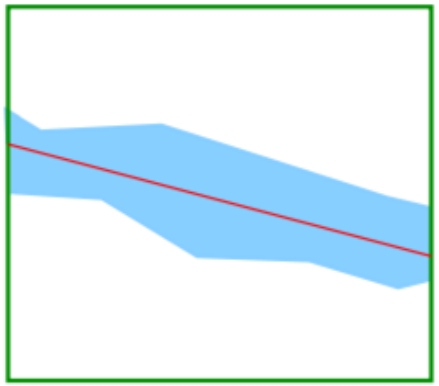
Note that it's *similar* to $\rho(h,h^*)$, except for the $\sup_{h \in B(h^*,r)}$ term.
Measure worst-case difference in loss over *all* hypotheses near h*.



$d(h^*, h) = \mathbb{P}$[shaded region]



Some elements of $B(h^*, r)$



DIS($B(h^*, r)$)

Image source:
http://hunch.net/~active_learning/active_learning_icml09.pdf

Disagreement Coefficient

Smallest θ such that, for all r,

$$E[\sup_{h \in B(h^*,r)} \sup_y |l(h(x),y) - l(h^*(x),y)|] \leq \theta r$$

Conveniently, the upper bound for the LHS scales linearly with the radius of the ball. This will be useful for proving Theorem 11.

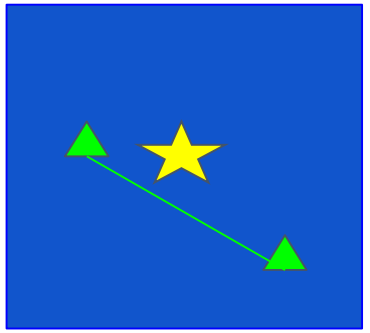
Lemma 10 bounds the disagreement coefficient for linear classifiers.

Upper Bound on p_t

$$E_x[p_t] = E_x[\max_{f,g \in H\_t,\ y \in Y} l(f(x),y) - l(g(x),y)]$$

The RHS looks a lot like the LHS of the definition of the disagreement coef.:

$$E[\sup_{h \in B(h^*,r)} \sup_y |l(h(x),y) - l(h^*(x),y)|]$$

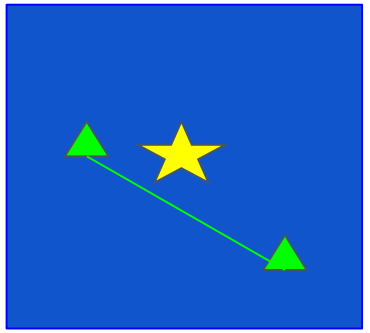


Upper Bound on p_t

$$E_x[p_t] = E_x[\max_{f,g \in H\_t,\ y \in Y} l(f(x),y) - l(g(x),y)]$$

The RHS looks a lot like the LHS of the definition of the disagreement coef.:

$$E[\sup_{h \in B(h^*,r)} \sup_y |l(h(x),y) - l(h^*(x),y)|]$$



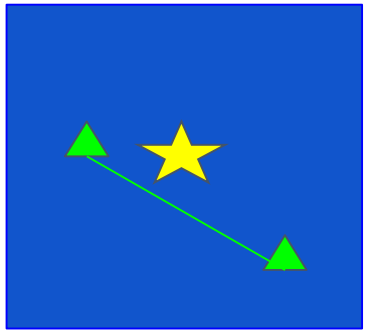
Need to bound in terms of fixed h* instead of g∈H_t

Upper Bound on p_t

$$E_x[p_t] = E_x[\max_{f,g \in H\_t,\ y \in Y} l(f(x),y) - l(g(x),y)]$$

The RHS looks a lot like the LHS of the definition of the disagreement coef.:

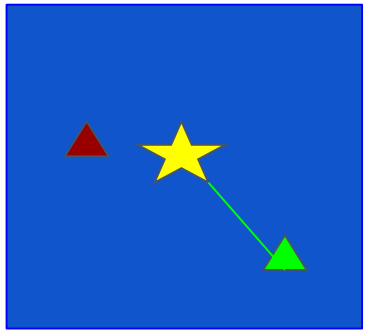
$$E[\sup_{h \in B(h^*,r)} \sup_y |l(h(x),y) - l(h^*(x),y)|]$$



Need to make a ball around h* that is a superset of H_t

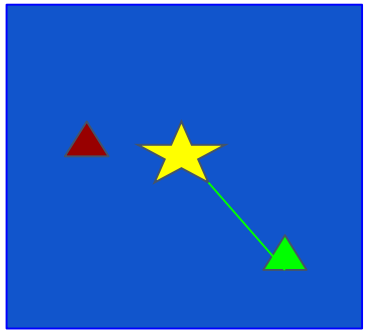
Upper Bound on p_t

Instead of considering pairs of hypotheses, just choose the one whose loss disagrees most from h*. By the triangle inequality, no two functions are separated by more than twice this difference.



Upper Bound on p_t

Instead of considering pairs of hypotheses, just choose the one whose loss disagrees most from h*. By the triangle inequality, no two functions are separated by more than twice this difference.



$$E_x[p_t] = E_x[\max_{f,g \in H\_t,\ y \in Y} l(f(x),y) - l(g(x),y)] =$$

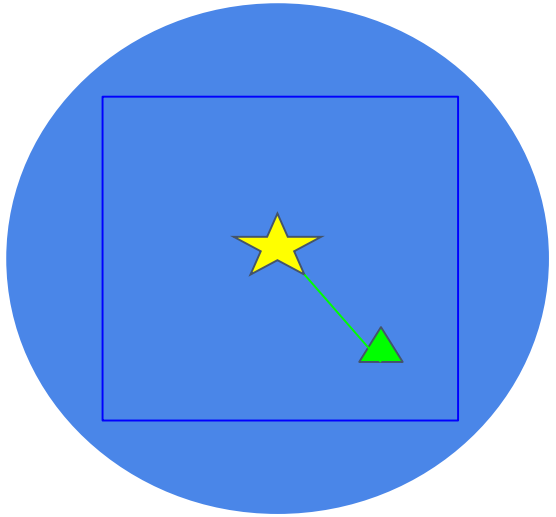
$$E_x[\sup_{f,g \in H\_t,\ y \in Y} | l(f(x),y) - l(g(x),y) |] \leq 2E_x[\sup_{f \in H\_t,\ y \in Y} | l(f(x),y) - l(h^*(x),y) |]$$

Upper Bound on p_t

All hypotheses in H_t are contained in some ball around h^*.

According to Lemma 8, which we will prove, this ball has radius

$$r = 2K_l(L^* + \Delta_{t-1})$$

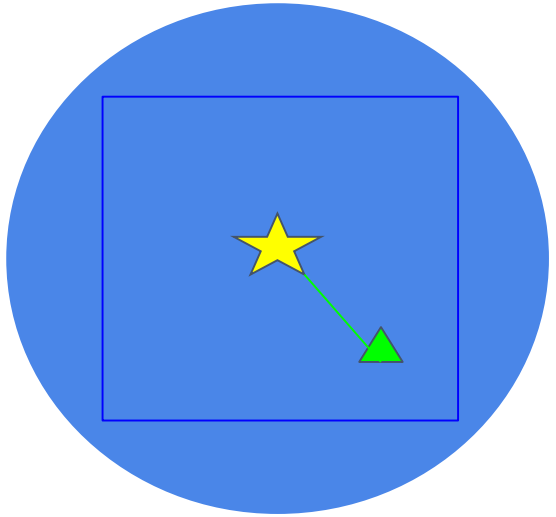


Upper Bound on p_t

All hypotheses in H_t are contained in some ball around h^*.

According to Lemma 8, which we will prove, this ball has radius

$$r = 2K_l(L^* + \Delta_{t-1})$$

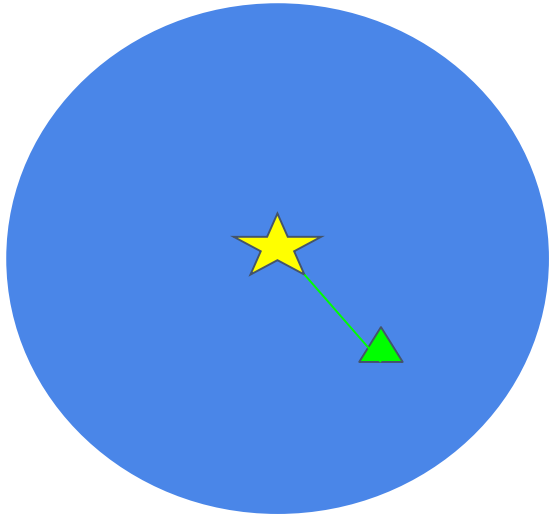


$$2E_x[\sup_{f \in H\_t,\, y \in Y} | l(f(x),y) - l(h^*(x),y) |]$$

$$\leq 2E_x[\sup_{f \in B(h^*,r),\, y \in Y} | l(f(x),y) - l(h^*(x),y) |]$$

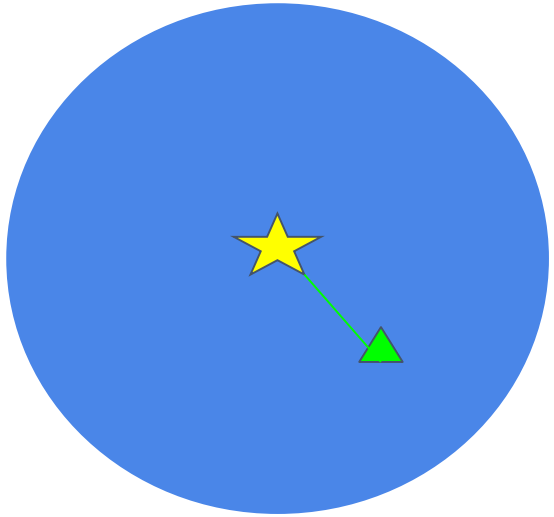
Upper Bound on p_t

We can now use the definition of the disagreement coefficient.



Upper Bound on p_t

We can now use the definition of the disagreement coefficient.



$$2E_x[\sup_{f \in B(h^*,r),\, y \in Y} | l(f(x),y) - l(h^*(x),y) |]$$

$$\leq 2\theta r = 4\theta K_l(L^* + \Delta_{t-1})$$

Upper Bound on Requested Labels

Combining this, we get

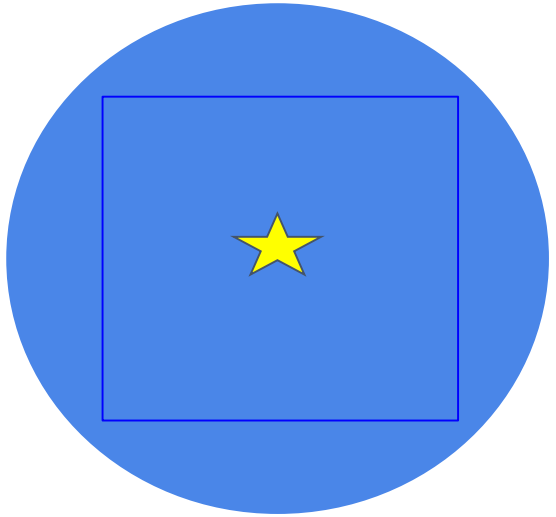
$$E_x[p_t] \leq 4\theta K_l(L^* + \Delta_{t-1})$$

We then sum over all t to get

$$E[\#\text{ requested}] \leq 4\theta K_l(TL^* + O(\sqrt{T}(\ln(|H|T/\delta))))$$
$$= O(TL^*) + O(\text{sublinear in } T)$$

Creating a ball of the correct size

Here we explain how we got the radius of the ball around H_t



Slope Asymmetry

Definition 4: The slope asymmetry of a loss function $l: Z \times Y \rightarrow [0,\infty)$ is

$$K_l = \sup_{z,z' \in Z} | \max_{y \in Y} l(z,y)-l(z',y) \,/\, \min_{y \in Y} l(z,y)-l(z',y) \,|$$

We can pick two possible responses (z, z'). Depending on the true label (e.g. +1 or -1), the difference between the losses of our responses may be large or small, positive or negative. If the most negative difference has about the same magnitude as the most positive difference for all responses, the loss function has low asymmetry.

Slope Asymmetry

Definition 4: The slope asymmetry of a loss function $l: Z \times Y \rightarrow [0,\infty)$ is

$$K_l = \sup_{z,z' \in Z} | \max_{y \in Y} l(z,y)-l(z',y) \,/\, \min_{y \in Y} l(z,y)-l(z',y) \,|$$

Example: 0-1 loss

We can pick any two responses. If they have the same sign, $l(z,y)-l(z',y)=0$ for all y. If they have a different sign, $l(z,y)-l(z'y)=\pm 1$. Thus $K_l=1$, the lowest possible value.

Slope Asymmetry

Definition 4: The slope asymmetry of a loss function $l: Z \times Y \to [0,\infty)$ is

$$K_l = \sup_{z,z' \in Z} | \max_{y \in Y} l(z,y)-l(z',y) \,/\, \min_{y \in Y} l(z,y)-l(z',y) \,|$$

Example: Hinge loss $l(z,y) = \max(0, 1-zy)$

Say we have $z \gg 0$ and $z'=0$.

$l(z,y)-l(z',y) = z \gg 0$ when $y = -1$.

$l(z,y)-l(z',y) = -1$ when $y = +1$.

Thus we can see that $K_l=\infty$

Slope Asymmetry

Definition 4: The slope asymmetry of a loss function $l: Z \times Y \rightarrow [0,\infty)$ is

$$K_l = \sup_{z,z' \in Z} |\max_{y \in Y} l(z,y)-l(z',y) \,/\, \min_{y \in Y} l(z,y)-l(z',y)\ |$$

Example: $l(z,y) = \phi(zy)$ for some differentiable ϕ

Assume the $z \in [-B,+B]$, $y \in \{+1, -1\}$, and $C_0 \leq |\phi'(zy)| \leq C_1$

Then $K_l \leq C_1/C_0$ (Lemma 5)

Intuition: loss functions whose slope varies a lot with the label have high slope asymmetry.

Slope Asymmetry

Corollary 6 gives a bound for logistic loss on a bounded response space.

This is the loss function the authors used in their experiments.

Lemma 8

We now can find the distance between two hypotheses. How does this distance relate to the expected loss of the hypotheses? We want to create a ball around h* that contains H_t, but H_t currently only has properties in terms of the expected loss.

Lemma 8 (Proof)

$\rho(h,h^*) = E_x[\max_y |l(h(x),y) - l(h^*(x),y)|]$

$\leq K_l E_{x,y}[|l(h(x),y) - l(h^*(x),y)|]$ since K_l bounds how large the difference can be for all y

$\leq K_l(E_{x,y}[l(h(x),y)] + E_{x,y}[l(h^*(x),y)])$ by Δ-ineq, def of loss function, and linearity of exp.

$= K_l(L(h) + L(h^*))$ by the definition of expected loss.

Lemma 8 (Application)

$$\rho(h,h^*) \le K_l(L(h) + L(h^*))$$
$$\le K_l(2L(h^*) + 2\Delta_{t-1})$$
$$= 2K_l(L^* + \Delta_{t-1})$$

for $h \in H_t$, $L(h) \le L(h^*) + 2\Delta_{t-1}$ (Lemma 2)

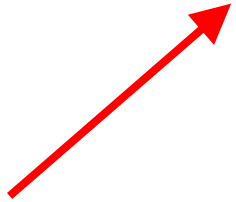
by definition.

Lemma 8 (Application)

$$\rho(h,h^*) \le K_l(L(h) + L(h^*))$$
$$\le K_l(2L(h^*) + 2\Delta_{t-1})$$
$$= 2K_l(L^* + \Delta_{t-1})$$

for $h \in H_t$, $L(h) \le L(h^*) + 2\Delta_{t-1}$ (Lemma 2)

by definition.



This term is why the upper bound has a term linear in T.

Implementation

Implementation and Experiments

Theory is useless without computational feasibility and results!

Importance of this paper is that it has all three.

We never actually implemented Algorithm 2.

Experimental Setup

For experiment, hypothesis set is bounded-length linear separators along with a convex loss function.

$$\{u \in \mathbb{R}^d : \|u\|^2 \leq B\}$$

Implementing Algorithm 2

Algorithm 2 features two optimization problems.

First, find optimal loss.
Then, find max loss-difference.

Algorithm 2 loss-weighting $(x, \{x_i, y_i, p_i, Q_i : i < t\})$

1. Initialize $H_0 = H$.
2. Update $L^*_{t-1} = \min_{h \in H_{t-1}} \frac{1}{t-1} \sum_{i=1}^{t-1} \frac{Q_i}{p_i} l(h(x_i), y_i)$

   Set H_t to

   $$\left\{h \in H_{t-1} : \frac{1}{t-1} \sum_{i=1}^{t-1} \frac{Q_i}{p_i} l(h(x_i), y_i) \leq L^*_{t-1} + \Delta_{t-1}\right\}$$
3. Return $p_t = \max_{f,g \in H_t, y \in Y} l(f(x), y) - l(g(x), y)$.

Implementing Algorithm 2

Both optimization problems are being solved over restricted hypothesis set.

$$H_t = \bigcap_{t'<t} \left\{ u \in \mathbb{R}^d : \|u\|^2 \leq B \text{ and } \frac{1}{t'} \sum_{i=1}^{t'} \frac{Q_i}{p_i} \phi(u \cdot (y_i x_i)) \leq L^*_{t'} + \Delta_{t'} \right\}$$

First Optimization Problem

$$L_T^* = \min_{u \in H_T} \sum_{i=1}^{T} \frac{Q_i}{p_i} \phi(u \cdot (y_i x_i))$$

First optimization is just a convex program, which can be solved using known computationally feasible methods.

Second Optimization Problem

$$\max_{u,v \in H_T} \phi(y(u \cdot x)) - \phi(y(v \cdot x)),\ y \in \{+1, -1\}$$

Second optimization problem is trickier. However, if ϕ is non-increasing, (as in 0-1 loss, hinge loss, or logistic loss), it is equivalent to:

$$\max\{\phi(A(x)) - \phi(-A(-x)), \phi(A(-x)) - \phi(-A(x))\}$$

This can be efficiently solved too!

$$A(x) \equiv \min_{u \in H_T} u \cdot x$$

Experimental Setup

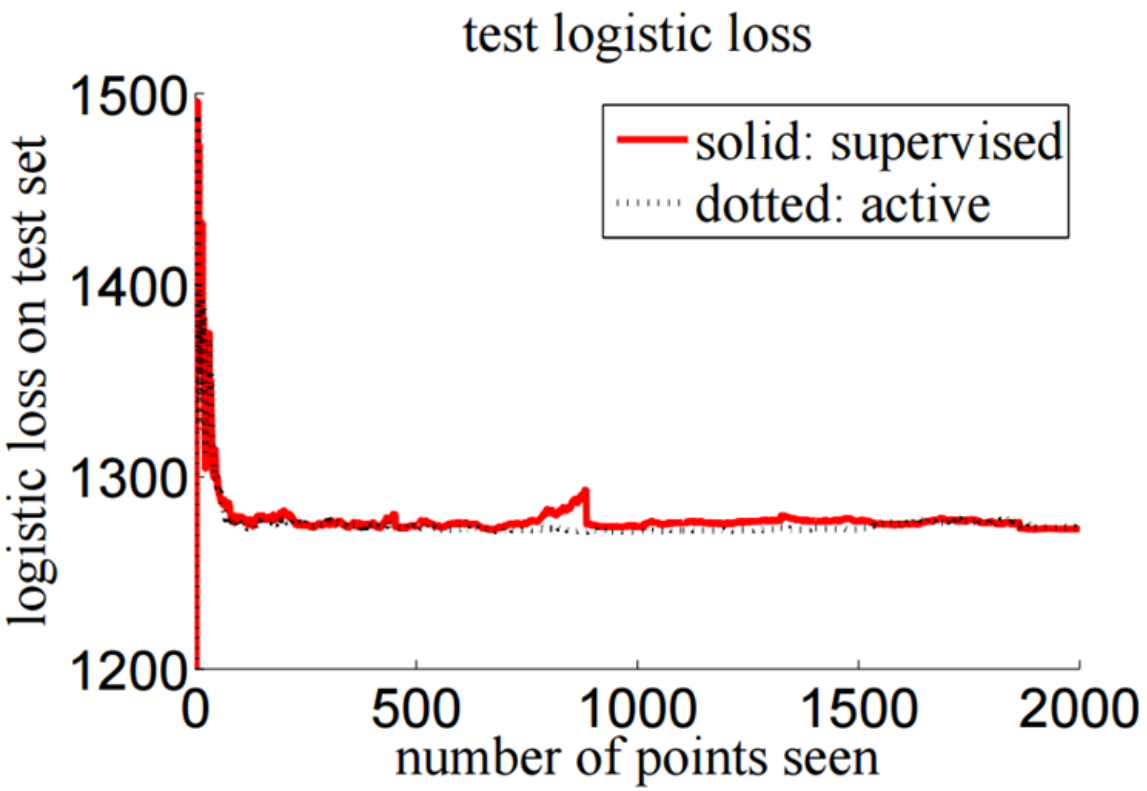
So this method is feasible, but not fast. For experiment, introduce some modifications for speed and simplicity.

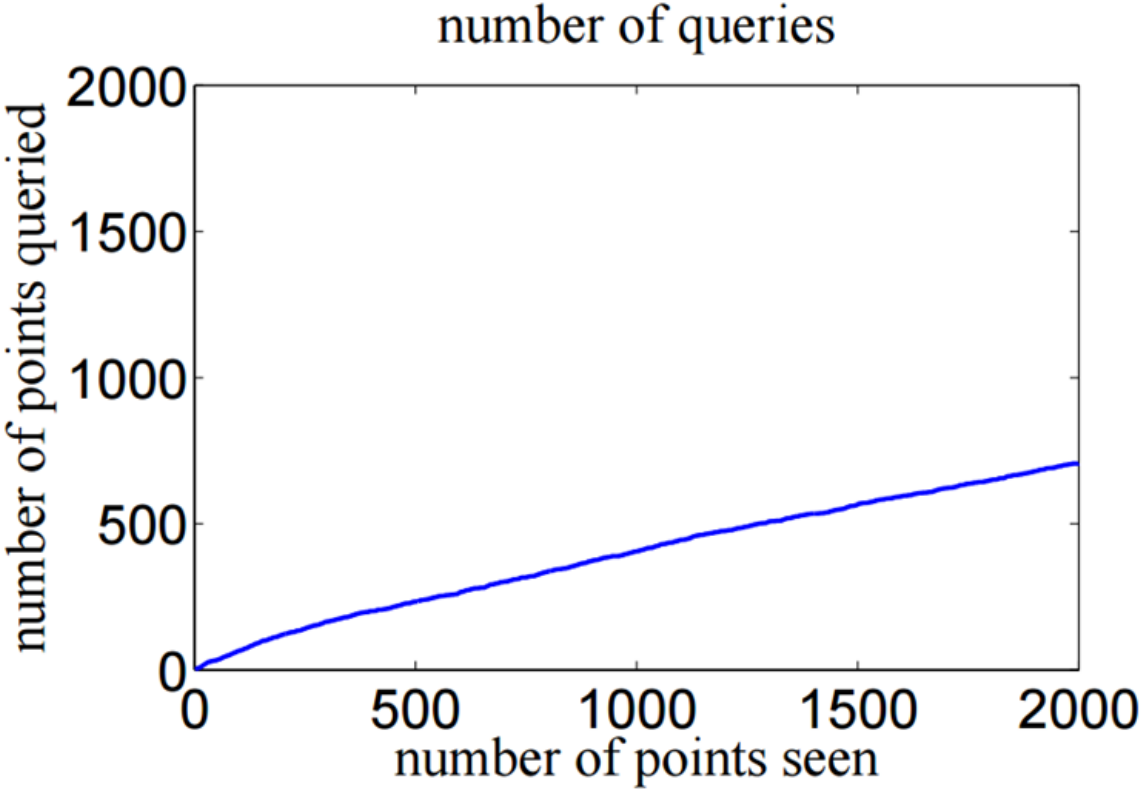
For first optimization, minimize over H rather than H_T.
For second optimization, instead of defining H_T by T-1 convex constraints, only enforce the last constraint. (Which corresponds to time T - 1)
May choose p_t conservatively, but still preserves consistency by Thm. 1!

MNIST Experiment

Produce a binary classifier for 3's and 5's from handwritten MNIST data.
Use PCA for dimensionality reduction.
1000 of each class for training, 1000 of each class for testing.

MNIST Experiment Results





Same accuracy as passive learning!
However, uses less than ⅓ of the labels!

Alternative Implementation (Bootstrap)

Results are promising, but algorithm only feasible for linear classifiers with convex loss functions. For other classifiers, what do we do?

Try an alternate rejection-threshold algorithm.

Will use a rough-and-tumble bootstrap method:

1. Ask for all labels in an initial batch of the training data.
2. Train a set of predictors on this bootstrap. This will serve as an approximation of the version space.
3. Given a new x_t, return

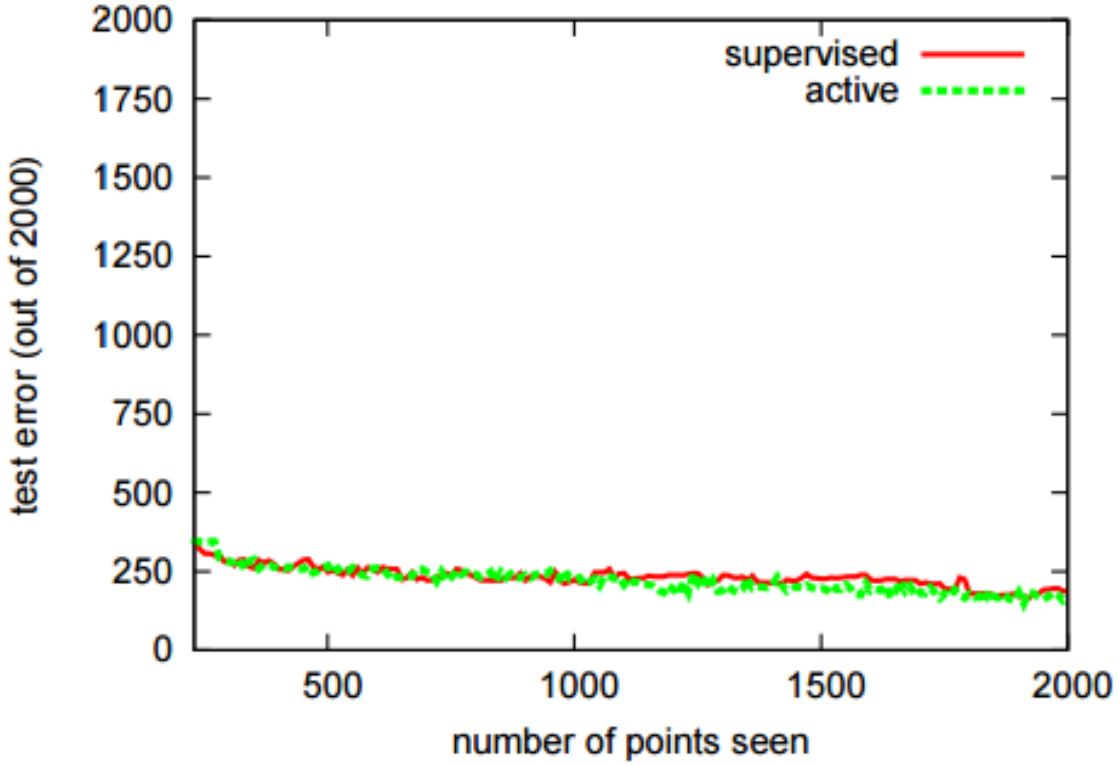
Note that this has been reduced to importance-weighted *batch* passive learning!

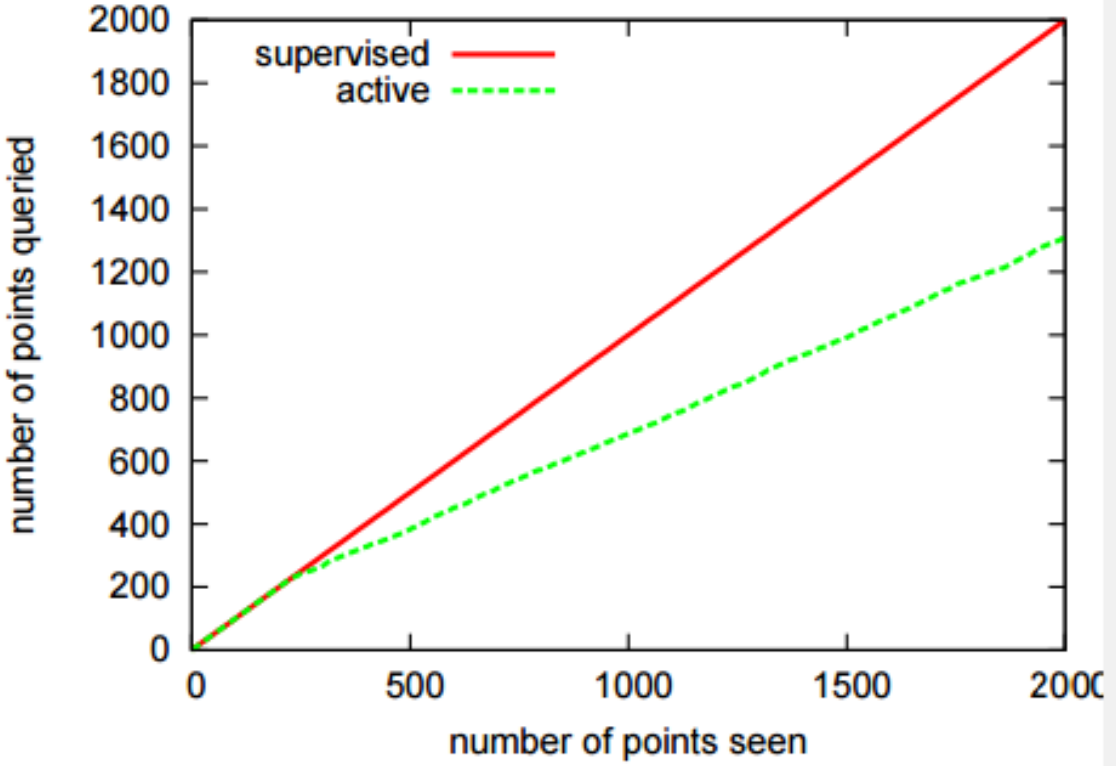
$$p_t = p_{\min} + (1 - p_{\min}) \left[\max_{y, h_i \in H, h_j \in H} L(h_i(x), y) - L(h_j(x), y) \right]$$

Bootstrap Experiments

Use 10 decision trees as H.
Bootstrap on first 10% of training set.
Use p_min = 0.1
Tested on some multiclass and binary classification problems.

Bootstrap Experiments Results





Bootstrap Experiments Results

Bootstrap results on other standard benchmark datasets.

Data set	IWAL error rate	Passive error rate	Queried	Train/test split
adult	14.1%	14.5%	40%	4000/2000
letter	13.8%	13.0%	75.0%	14000/6000
pima	23.3%	26.4%	67.6%	538/230
spambase	9.0%	8.9%	44.2%	3221/1380
yeast	28.8%	28.6%	82.2%	1000/500

ls!

Conclusion

IWAL is very exciting.
Good theoretical bounds tied to good empirical accuracy that is computationally feasible and often applicable.
Reduces labels needed -> saves money!

Questions

Sources

- Yisong Yue
- S. Dasgupta and J. Langford. A tutorial on active learning *Presentation at the 26th Conference on Machine Learning,* 2009.

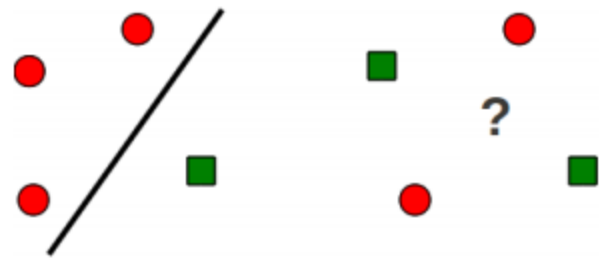
Extra Slides

Sample Complexity Results

- There are upper and lower bound results on the sample complexity of certain tasks
- For example, learning a half-space in n dimensions with respect to the uniform distribution has an upper bound of $O(1/\varepsilon(n + \log(1/\delta)))$ and a matching lower bound (Long 2003).
- In general, such bounds depend on ε, δ, and the VC dimension of the model class

Shattering

- **Definition:** A set of points is **shattered** by H if for all **possible binary labelings** of points, there exists some h that classifies perfectly.



In 2D, linear models cannot shatter 4 points!

VC Dimension

- VC(H) = most # points that can be shattered
  - If H is linear models in 2D feature space:
    - VC(H) = 3

Definition of a generalization bound (tells us whether we are overfitting or not)

With Prob. ≥ 1-δ :

$$E_{out}(h) \leq E_{in}(h) + O\left(\frac{\log\left(\frac{2N}{VC(H)}+1\right)+\log\left(\frac{1}{\delta}\right)}{\sqrt{N}}\right)$$

VC Dimension and Sample Complexity

- In the supervised learning case, if we want to achieve an ε-learner, we need at most d/ε^2 examples, where d is the VC dimension
- The VC dimension of the hypothesis class also affects label complexity bounds, along with another parameter called the *disagreement coefficient*

Disagreement Coefficient

- Define a metric on hypotheses which is the probability that they differ:

$$d(h, h') = \mathbb{P}[h(X) \neq h'(X)]$$

- We will call the subset of X on which some hypotheses in a version space V disagree the *disagreement region*

$$\mathrm{DIS}(V) = \{x \in \mathcal{X} : \text{there exist } h, h' \in V \text{ with } h(x) \neq h'(x)\}.$$

Disagreement Coefficient cont'd.

- The *disagreement coefficient* measures how the probability that a random point in the disagreement region in a ball around the optimal hypothesis scales with r:

$$\theta = \sup_r \frac{\mathbb{P}[\mathrm{DIS}(B(h^*, r))]}{r}$$

- Bounds or upper bounds for the disagreement coefficient are known in some cases; for example, for linear separators in R^d, $\theta \leq \sqrt{d}$, so the label complexity is O(d^(3/2)log(1/ε))

Definition: Martingale

A sequence of random variables $\{X_1, X_2, \ldots\}$ that for all n satisfies both

$$E(|X_n|) < \infty$$

and

$$E(X_{n+1}|X_0, \ldots, X_n) = X_n$$

(equivalently $E(X_{n+1} - X_n|X_0, \ldots, X_{n-1}) = 0$)

Azuma's inequality

For a Martingale $\{X_k : k = 0, 1, 2, 3, \ldots\}$ with $|X_k - X_{k-1}| < c_k$ a.s., $\forall N > 0$ and $\forall t \in \mathbb{R}$,

$$P(X_N - X_0 \geq t) \leq \exp\left(\frac{-t^2}{2\sum_{k=1}^{N} c_k^2}\right)$$

$$P(X_N - X_0 \leq -t) \leq \exp\left(\frac{-t^2}{2\sum_{k=1}^{N} c_k^2}\right)$$

$$P(|X_N - X_0| \geq t) \leq 2\exp\left(\frac{-t^2}{2\sum_{k=1}^{N} c_k^2}\right)$$

Proof of Lemma 3

- Begin by fixing T, $f, g \in H_T$
- Define $Z_t = \frac{Q_t}{p_t}[l(f(x_t), y_t) - l(g(x_t), y_t)] - (L(f) - L(g))$
- $E[Z_t | Z_1, \ldots, Z_{t-1}] =$

$$E_{x_t, y_t}[l(f(x_t), y_t) - l(g(x_t), y_t))|Z_1, \ldots, Z_{t-1}] - (L(f) - L(g))$$

Proof of Lemma 3

- Begin by fixing T, $f, g \in H_T$
- Define $Z_t = \frac{Q_t}{p_t}[l(f(x_t), y_t) - l(g(x_t), y_t)] - (L(f) - L(g))$
- $E[Z_t | Z_1, \ldots, Z_{t-1}] =$

$$E_{x_t, y_t}[l(f(x_t), y_t) - l(g(x_t), y_t))|Z_1, \ldots, Z_{t-1}] - (L(f) - L(g))$$

Recall

$$L(h) = E_{(x,y)\sim D}[l(h(x), y)]$$

Proof of Lemma 3

- Begin by fixing T, $f, g \in H_T$
- Define $Z_t = \frac{Q_t}{p_t}[l(f(x_t), y_t) - l(g(x_t), y_t)] - (L(f) - L(g))$
- $E[Z_t | Z_1, \ldots, Z_{t-1}] =$

$$E_{x_t, y_t}[l(f(x_t), y_t) - l(g(x_t), y_t))|Z_1, \ldots, Z_{t-1}] - (L(f) - L(g))$$

Recall

$$L(h) = E_{(x,y)\sim D}[l(h(x), y)]$$

- $E[Z_t | Z_1, \ldots, Z_{t-1}] = 0$

Proof of Lemma 3

(Define $Z_t = \frac{Q_t}{p_t}[l(f(x_t), y_t) - l(g(x_t), y_t)] - [L(f) - L(g)]$)

Now we want to show that Z_t is bounded.

- $f, g \in H_t \implies f, g \in H_1, H_2, \ldots, H_{t-1}$

  Recall $p_t := \max_{f,g \in H_t, y \in Y}[l(f(x), y) - l(g(x), y)]$

- $\implies \forall t \leq T,\ p_t \geq |l(f(x_t), y_t) - l(g(x_t), y_t)|$
- $|Z_t| \leq \frac{1}{p_t}|l(f(x_t), y_t) - l(g(x_t), y_t)| + |L(f) - L(g)| \leq 2$

Proof of Lemma 3

(Define $Z_t = \frac{Q_t}{p_t}[l(f(x_t), y_t) - l(g(x_t), y_t)] - [L(f) - L(g)]$)

Now that we have $\{Z_t\}_{t=1}^{T}$ is Martingale, we can apply Azuma's inequality

$$P(|X_N - X_0| \geq t) \leq 2\exp\left(\frac{-t^2}{2\sum_{k=1}^{N} c_k^2}\right)$$

with $c_k = 2$

Proof of Lemma 3

(Define $Z_t = \frac{Q_t}{p_t}[l(f(x_t), y_t) - l(g(x_t), y_t)] - [L(f) - L(g)]$)

Now that we have $\{Z_t\}_{t=1}^{T}$ is Martingale, we can apply Azuma's inequality

$$P(|X_N - X_0| \geq t) \leq 2\exp\left(\frac{-t^2}{2\sum_{k=1}^{N} c_k^2}\right)$$

with $c_k = 2$

- $P(|L_T(f) - L_T(g) - L(f) + L(g)| \geq \Delta_T) = P(|\sum_{t=1}^{T} Z_t| \geq T\Delta_T)$
- $P(|\sum_{t=1}^{T} Z_t| \geq T\Delta_T) \leq 2\exp\left(\frac{-T\Delta_T^2}{8}\right)$

Proof of Lemma 3

- $P(|L_T(f) - L_T(g) - L(f) + L(g)| \geq \Delta_T) \leq 2\exp\left(\frac{-T\Delta_T^2}{8}\right)$

  Recall $\Delta_t = \sqrt{\frac{8}{t}\ln(\frac{2t(t+1)|H|^2}{\delta})}$

- $2\exp\left(\frac{-T\Delta_T^2}{8}\right) = \frac{\delta}{T(T+1)|H|^2}$

Change fixed $f, g \in H$ to $\forall f, g \in H$ via union bound (factor of $|H|^2$), then union bound over T