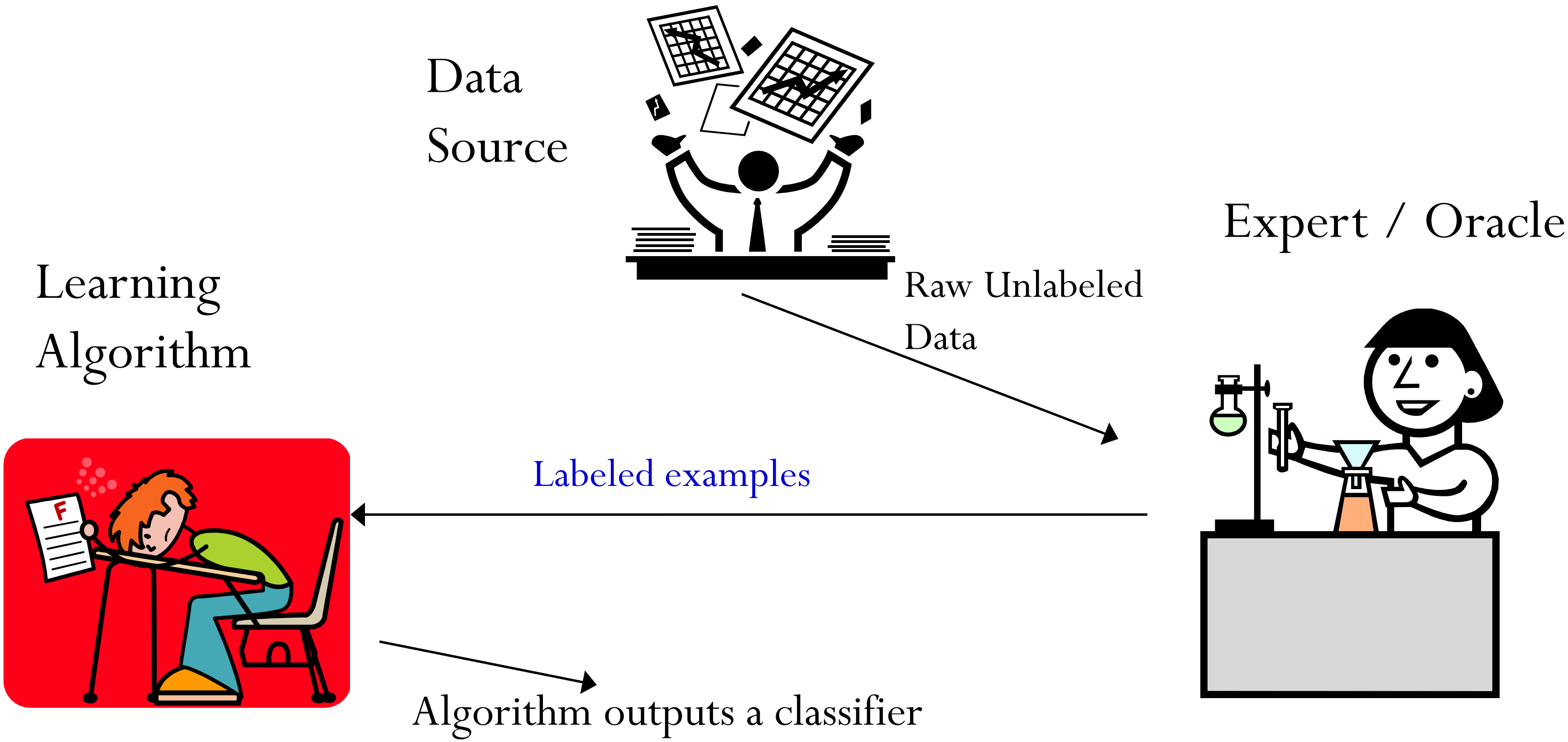
Sample Complexity of Active Learning

Maria Florina Balcan

School of Computer Science
Georgia Institute of Technology

Passive Supervised Learning



Standard Passive Supervised Learning


- X - *instance/feature space*
- S={(x, l)} - set of labeled examples
  - labeled examples - drawn i.i.d. from distr. D over X and labeled by some target concept c^*
    - labels $\in$ {-1,1} - binary classification

- Do optimization over S, find hypothesis $h \in C$.
- Goal: h has small error over D.

$$err(h)=Pr_{x \in D}(h(x) \neq c^*(x))$$

c* in C, realizable case
c* not in C, agnostic case

Sample Complexity: Uniform Convergence Bounds

- Infinite Hypothesis Case, Realizable Case

Theorem

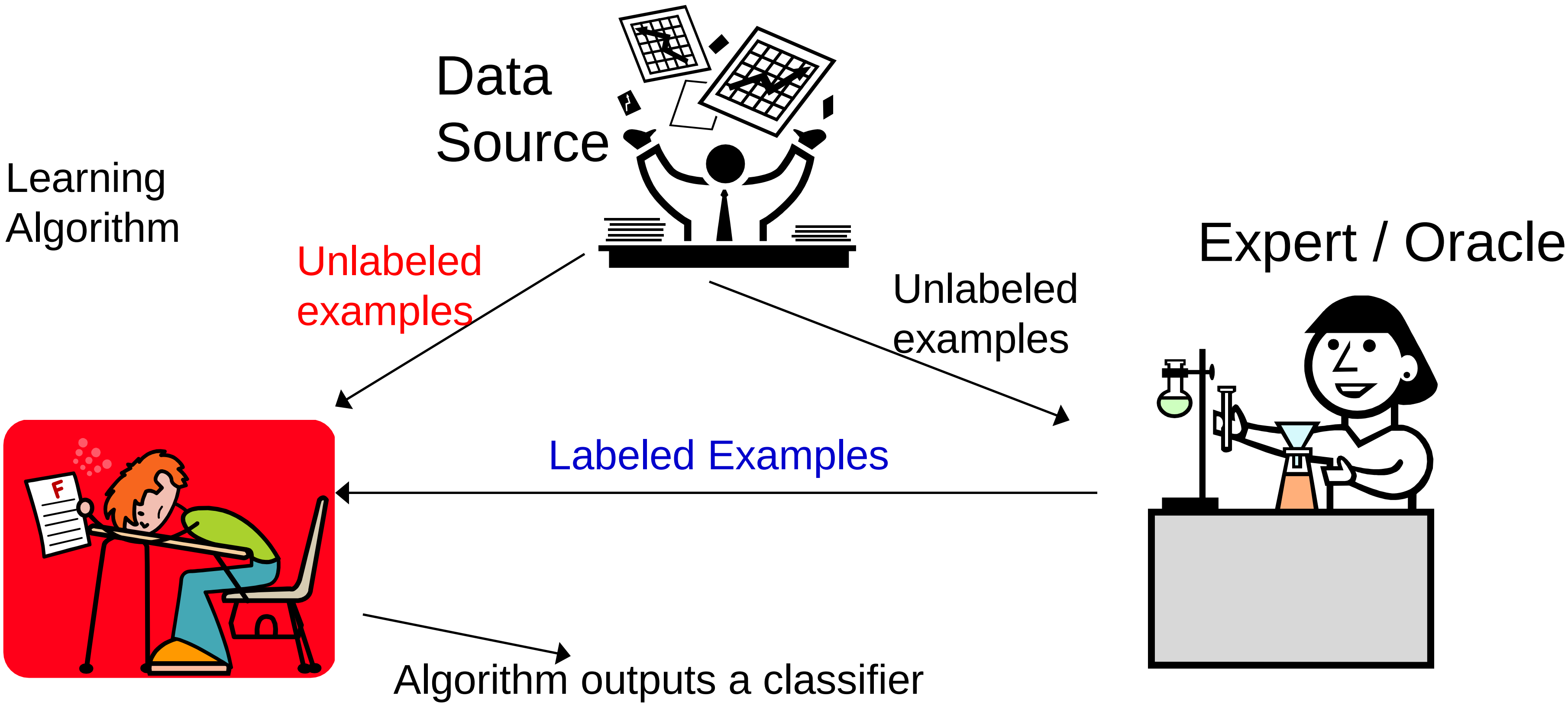
$$m = O\left(\frac{1}{\varepsilon}\left[VCdim(C)\log\left(\frac{1}{\varepsilon}\right) + \log\left(\frac{1}{\delta}\right)\right]\right)$$

labeled examples are sufficient so that with prob. $1-\delta$, all $h \in C$ with $err(h) \geq \varepsilon$ have $\widehat{err}(h) > 0$.

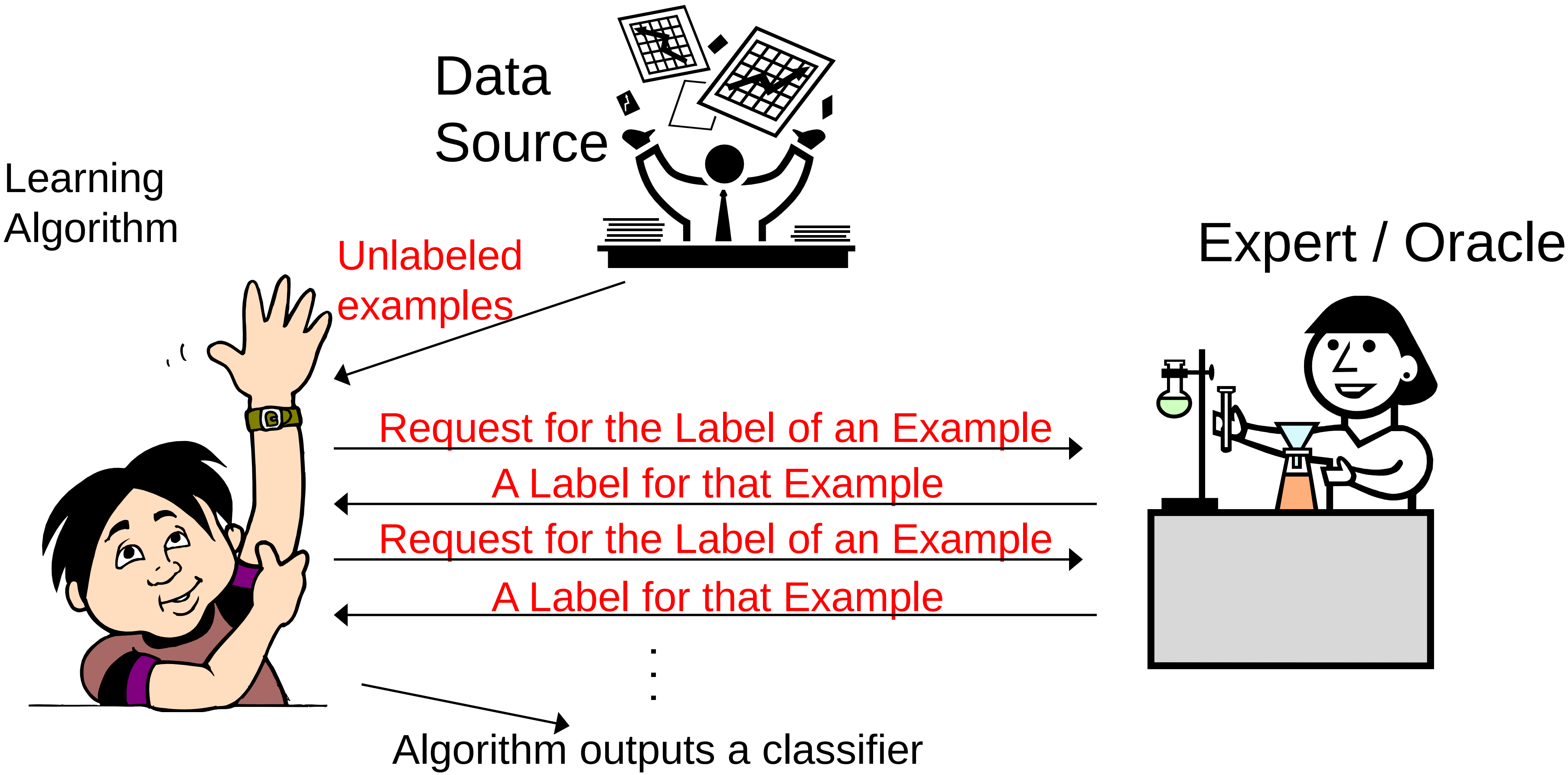
E.g., if C - class of linear separators in R^d, then need $O(d/\varepsilon)$ examples to achieve generalization error ε.

Non-realizable case – replace ε with ε^2.

Semi-Supervised Passive Learning



Active Learning



Active Learning

- We get to see unlabeled data first, and there is a charge for every label.
- The learner has the ability to choose specific examples to be labeled.
- The learner works harder, in order to use fewer labeled examples.

- Do we need fewer examples in this setting than in the passive learning setting?
- How many labels can we save by querying adaptively?

Outline

- Standard PAC-style active learning analysis

  e.g., Das04, Das05, DKM05, BBL06, Kaa06, Han07a&b, BBZ07, DHM07

- A new analysis framework

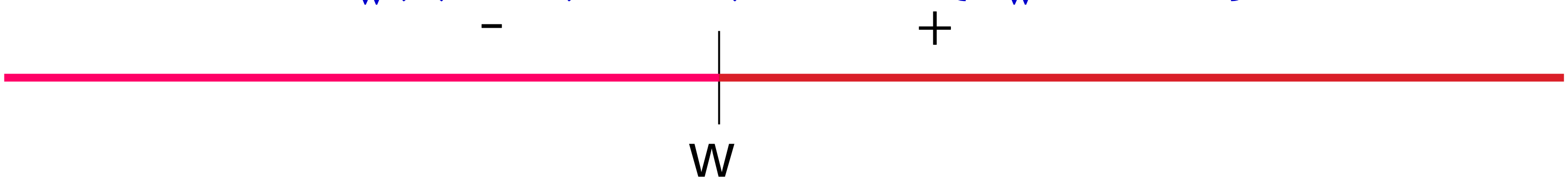
  Joint with Steve Hanneke and Jenn Wortman

- Conclusions & Open Problems

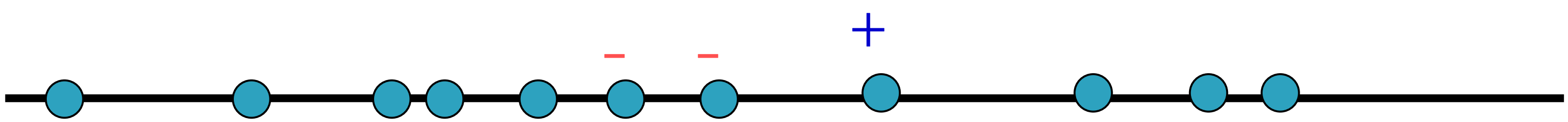
Can adaptive querying help? [CAL92, Dasgupta04]

- Consider threshold functions on the real line:

$$h_w(x) = 1(x \geq w), \quad C = \{h_w : w \in R\}$$



- Sample with $1/\varepsilon$ unlabeled examples.



- Binary search – need just $O(\log 1/\varepsilon)$ labels.

Active setting: $O(\log 1/\varepsilon)$ labels to find an ε-accurate threshold.

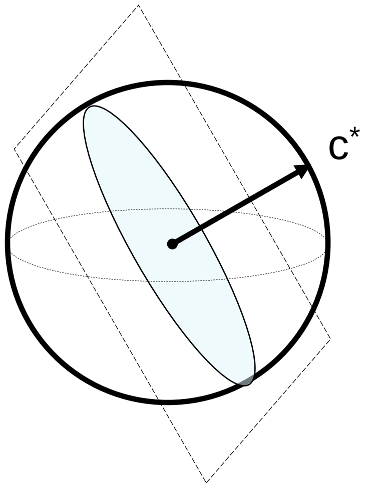
Supervised learning provably needs $\Omega(1/\varepsilon)$ labels. [Antos Lugosi, 96]

Exponential improvement in sample complexity ☺

Other Examples where Active Learning helps

- C - homogeneous linear separators in R^d, D - uniform distribution over unit sphere.

“Region of disagreement” ([CAL’92]):



Pick a few points at random from the current region of uncertainty, query their labels, throw out hypothesis if you are statistically confident they are suboptimal.

Realizable: need only $d^{3/2} \log (1/\varepsilon)$ labeled examples to learn a classifier of error ϵ .

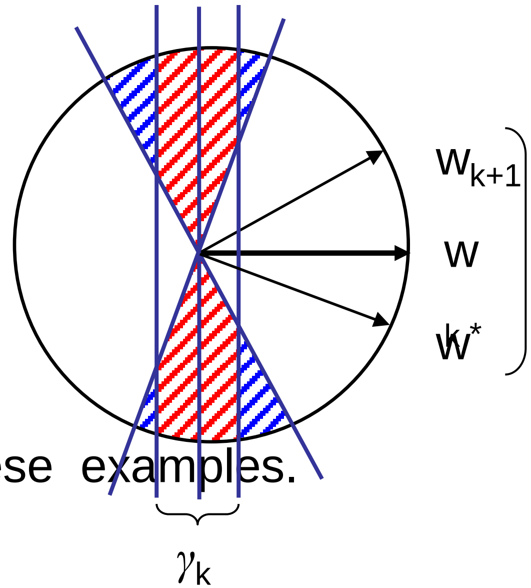
With $d^{3/2}$ labeled examples can halve the region of disagreement.

Other Examples where Active Learning helps

- C - homogeneous linear separators in R^d, D - uniform distribution over unit sphere.

Realizable: only $d \log (1/\varepsilon)$ labeled examples to learn a classifier of error ϵ [Dasgupta-Kalai-Monteleoni,COLT 2005]

[Balcan-Broder-Zhang, COLT 07]



Use O(d) examples to find w_1 of error 1/8.

iterate k=2, … , $\log(1/\varepsilon)$

- rejection sample m_k samples x from D satisfying $|w_{k-1}^T \cdot x| \leq \gamma_k$;
- label them;
- find $w_k \in B(w_{k-1}, 1/2^k)$ consistent with all these examples.

end iterate

[Balcan-Broder-Zhang, COLT 07]

Agnostic Active Learning Results

A^2 the first algorithm which is robust to noise.

[Balcan, Beygelzimer, Langford, ICML'06] [Balcan, Beygelzimer, Langford, JCSS'08]

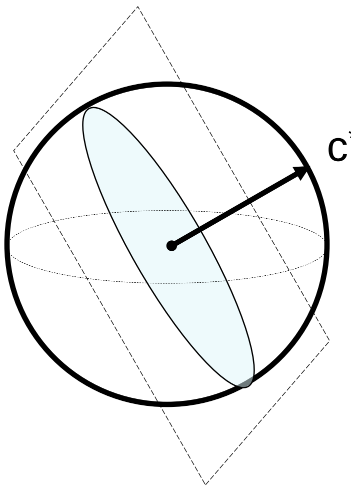
"Region of disagreement" style: Pick a few points at random from the current region of uncertainty, query their labels, throw out hypothesis if you are statistically confident they are suboptimal.

(similar to [CAL'92] realizable case)

Guarantees for A^2:

- Fall-back & exponential improvements.
- C – thresholds, low noise, exponential improvement.
- C - homogeneous linear separators in R^d,

D - uniform over unit sphere, low noise, only $d^2 \log(1/\varepsilon)$ labels to find h with error ε.



Interesting subsequent work. [Hanneke'07, DHM'07]

Active Learning might not help [Dasgupta04]

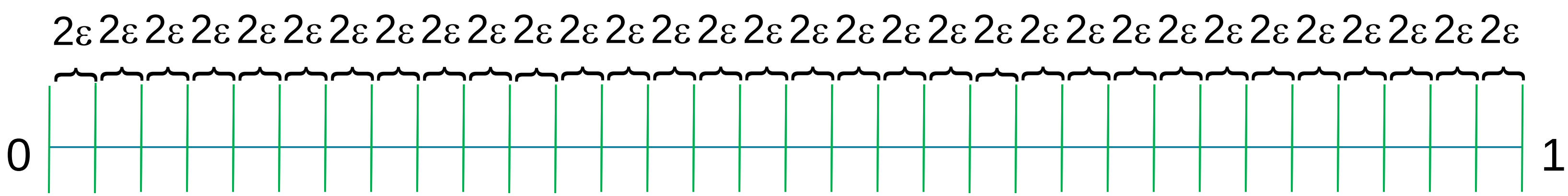
C = {intervals on the line}.

E.g., suppose D is uniform on [0,1]

In this case: learning to accuracy ε requires $1/\varepsilon$ labels…

Intervals on the line

Suppose D is uniform on [0,1]



Suppose the target labels everything “-1”

Need $\Omega(1/\varepsilon)$ label requests to guarantee the target isn’t one of these.

Active Learning does not help. 

Subtle Variation on the traditional model

Non-verifiable and Target Dependent Sample Complexity

target-dependent

budget

Definition: An algorithm $A(n, \delta)$ achieves *sample complexity* $S(\epsilon, \delta, f)$ for $(\mathbb{C}, \mathcal{D})$ if it outputs a classifier h_n after at most n label requests, and for any target function $f \in \mathbb{C}$, $\epsilon > 0$, $\delta > 0$, for any $n \geq S(\epsilon, \delta, f)$,

$$\mathbb{P}[er(h_n) \leq \epsilon] \geq 1 - \delta.$$

Intervals on the line

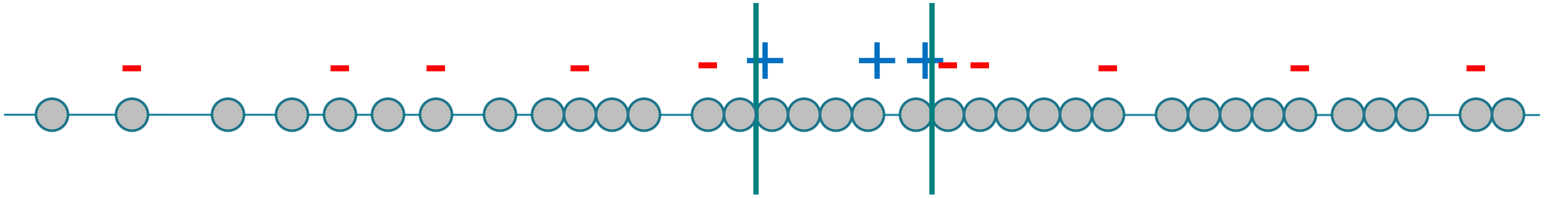
Algorithm

Take a large number of unlabeled examples.

Phase 1: Query random examples until we find a +1 example.
(if use all n label requests before finding a +1 example, return the empty interval)

Phase 2: Do binary searches to the left and right of the +1 point.

After n total label requests, return the smallest consistent $h \in C$.



Intervals on the line

Algorithm

Take a large number of unlabeled examples.
Phase 1: Query random examples until we find a +1 example.
(if use all n label requests before finding a +1 example, return the empty interval)
Phase 2: Do binary searches to the left and right of the +1 point.
After n total label requests, return any consistent h∈C.

Asymptotic analysis:
Case 1: If the target f has $\mathbb{P}[f(X) = +1] = w > 0$,
we find a +1 after $\propto \frac{1}{w} \log \frac{1}{\delta}$ requests.
The binary searches need only $O(\log \frac{1}{\epsilon})$ to approximate the boundaries.
Sample Complexity: $S(\epsilon, \delta, f) \propto \frac{1}{w} \log \frac{1}{\delta} + \log \frac{1}{\epsilon} = O(\log \frac{1}{\epsilon})$.

Case 2: If $\mathbb{P}[f(X) = +1] = 0$,
we will return an h with $er(h) = 0$ for any $n \geq 0$.

Sample Complexity: $S(\epsilon, \delta, f) = 0$

Subtle Variation on the traditional model

Non-verifiable and Target Dependent Sample Complexity

budget

target-dependent

Definition: An algorithm $A(n, \delta)$ achieves *sample complexity* $S(\epsilon, \delta, f)$ for $(\mathbb{C}, \mathcal{D})$ if it outputs a classifier h_n after at most n label requests, and for any target function $f \in \mathbb{C}$, $\epsilon > 0$, $\delta > 0$, for any $n \geq S(\epsilon, \delta, f)$,

$$\mathbb{P}[er(h_n) \leq \epsilon] \geq 1 - \delta.$$

Can Active Learning *Always* Help?

Active Learning Always Helps!

Theorem: For any pair $(\mathbb{C}, \mathcal{D})$, and any passive learning sample complexity $S_p(\epsilon, \delta, f)$ for $(\mathbb{C}, \mathcal{D})$, there exists an active learning algorithm achieving a sample complexity $S_a(\epsilon, \delta, f)$ s.t., for all targets $f \in \mathbb{C}$ for which $S_p(\epsilon, \delta, f) = \omega(1)$,

$$S_a(\epsilon, \delta, f) = o(S_p(\epsilon/4, \delta, f)).$$

Corollary: For any pair $(\mathbb{C}, \mathcal{D})$, there is an active learning algorithm that achieves a sample complexity $S_a(\epsilon, \delta, f)$ such that

$$\forall f \in \mathbb{C}, S_a(\epsilon, \delta, f) = o(1/\epsilon).$$

Proof Outline

- Claim 1: The result is certainly true for “threshold-esc” problems – where the problem gets easier the longer we work at it (based on [Hanneke07], “disagreement coefficient” analysis)
- Claim 2: Any C can be partitioned into $C_1, C_2, C_3, \ldots$ with this property.
- Claim 3: There is an aggregation algorithm that uses all of $C_1, C_2, C_3, \ldots$ but is never much worse than using just the C_i that contains the target f.

Exponential Improvements

It is often possible to achieve *polylogarithmic* sample complexity for all targets.

$$S(\epsilon, \delta, f) = \gamma_f \cdot polylog(1/(\epsilon\delta)),$$

For example:

- linear separators, under uniform distributions on an r-sphere
- Axis-aligned rectangles, under uniform distributions on $[0,1]^r$
- Finite unions of intervals on the real line (arbitrary distributions)

Can also preserve polylog sample complexities under some transformations:

- Unions, “close” distributions, mixtures of distributions

Conclusions

- Lots of exciting work recently.
- [BHW]: Active learning can always achieve a strictly superior asymptotic sample complexity compared to passive learning.

Big Open Directions

- Efficient and practical active learning algorithms that work in the presence of certain types of noise.

- Incorporate other type of interaction in the learning process.

Thank You