Part II: Active Learning in the PAC Setting

Chicheng Zhang

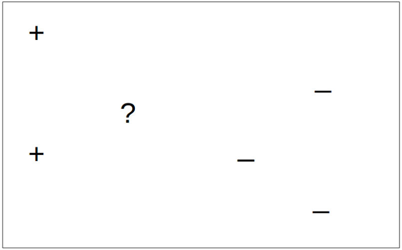
University of California San Diego

June 21, 2017

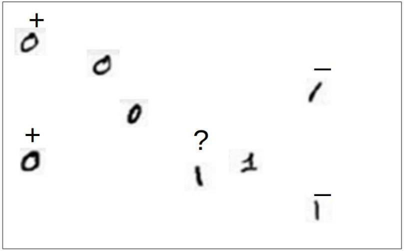
Outline

Membership Query vs PAC Model

Membership Query Model



PAC Model



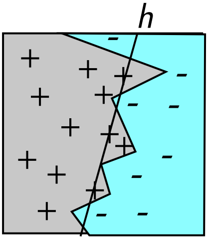
Probably Approximately Correct (PAC) active learning:

- Query labels only of given unlabeled examples
- Evaluation metric: classification error wrt distribution

Outline

PAC Model Setup

- Data distribution D over $\mathcal{X} \times \{-1, 1\}$
  unlabeled distribution $D_{\mathcal{X}}$
- Classifier $h : \mathcal{X} \to \{-1, 1\}$
- Hypothesis class $\mathcal{H}$

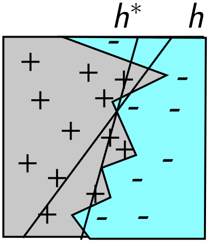


PAC Model: Evaluation

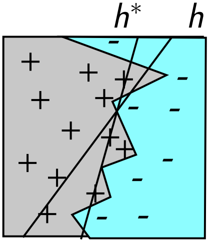
- Error: $\mathrm{err}(h) = \mathbb{P}_D[h(x) \neq y]$
- Optimal classifier

$$h^* = \mathrm{argmin}_{h \in \mathcal{H}} \, \mathrm{err}(h)$$

- Excess error: $\mathrm{err}(h) - \mathrm{err}(h^*)$



PAC Model: Evaluation



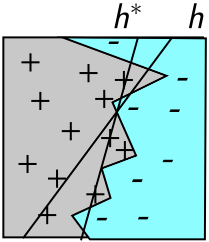
- Error: $\mathrm{err}(h) = \mathbb{P}_D[h(x) \neq y]$
- Optimal classifier

$$h^* = \mathrm{argmin}_{h \in \mathcal{H}} \, \mathrm{err}(h)$$

- Excess error: $\mathrm{err}(h) - \mathrm{err}(h^*)$

- PAC learning goal: get a classifier $\hat{h}$ with excess error ϵ

PAC Model: Evaluation



- Error: $\mathrm{err}(h) = \mathbb{P}_D[h(x) \neq y]$
- Optimal classifier

$$h^* = \mathrm{argmin}_{h \in \mathcal{H}} \mathrm{err}(h)$$

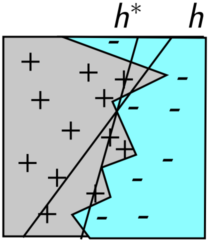
- Excess error: $\mathrm{err}(h) - \mathrm{err}(h^*)$
- PAC learning goal: get a classifier $\hat{h}$ with excess error ϵ with probability $1 - \delta$ over the draw of random sample S

PAC Model: Evaluation

- Error: $\mathrm{err}(h) = \mathbb{P}_D[h(x) \neq y]$
- Optimal classifier

$$h^* = \mathrm{argmin}_{h \in \mathcal{H}} \mathrm{err}(h)$$

- Excess error: $\mathrm{err}(h) - \mathrm{err}(h^*)$



- PAC learning goal: get a classifier $\hat{h}$ with excess error ϵ with probability $1 - \delta$ over the draw of random sample S
- Empirical error in sample S:

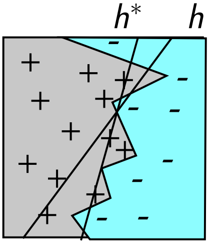
$$\mathrm{err}(h, S) = \frac{1}{|S|} \sum_{(x,y) \in S} \mathbb{1}\{h(x) \neq y\}$$

PAC Model: Evaluation

- Error: $\mathrm{err}(h) = \mathbb{P}_D[h(x) \neq y]$
- Optimal classifier

$$h^* = \mathrm{argmin}_{h \in \mathcal{H}} \mathrm{err}(h)$$

- Excess error: $\mathrm{err}(h) - \mathrm{err}(h^*)$

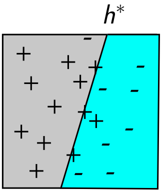


- PAC learning goal: get a classifier $\hat{h}$ with excess error ϵ with probability $1 - \delta$ over the draw of random sample S
- Empirical error in sample S:

$$\mathrm{err}(h, S) = \frac{1}{|S|} \sum_{(x,y) \in S} \mathbb{1}\{h(x) \neq y\}$$

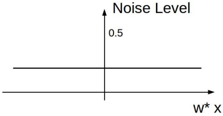
- Sample complexity $n(\epsilon, \delta)$: sample size needed to achieve goal

PAC Learning: Noise Models

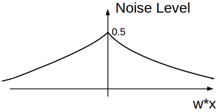


- Realizable: $\mathrm{err}(h^*) = 0$

Flipping Probability $\eta(x) := \mathbb{P}[Y \neq h^*(x)|x]$



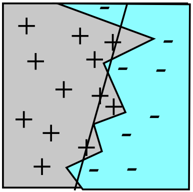
- η-Random classification noise (RCN): $\eta(x) = \eta \leq \frac{1}{2}$



- β-Tsybakov noise condition (TNC): $\mathbb{P}[\eta(x) \geq \frac{1}{2} - t] \leq O(t^{\frac{1}{\beta}})$

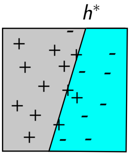
Agnostic Noise Model

- No assumption on label generation process
- Optimal error rate $\mathrm{err}(h^*) = \nu$

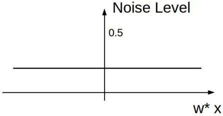


PAC Learning: Noise Models

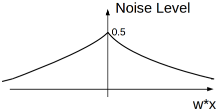
- Realizable: $\text{err}(h^*) = 0$



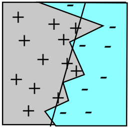
- η-Random classification noise (RCN): $\eta(x) = \eta \leq \frac{1}{2}$



- β-Tsybakov noise condition (TNC): $\mathbb{P}[\eta(x) \geq \frac{1}{2} - t] \leq O(t^{\frac{1}{\beta}})$



- ν-Agnostic: optimal error $\text{err}(h^*) = \nu$



Sample Complexity in PAC Passive Learning

- "Difficulty" of noise models:
  Realizable $<$ RCN $<$ TNC $<$ Agnostic
- d: VC dimension of $\mathcal{H}$

Noise Model	$n(\epsilon, \delta)$
Realizable	$\tilde{O}(d \cdot \frac{1}{\epsilon})$
η-RCN	$\tilde{O}(\frac{d}{1-2\eta} \cdot \frac{1}{\epsilon})$
β-TNC	$\tilde{O}(d \cdot \epsilon^{\frac{1}{1+\beta}-2})$
ν-Agnostic	$\tilde{O}(d \cdot \frac{\nu+\epsilon}{\epsilon^2})$

PAC Active Learning

Given:

- Access to **unlabeled examples** drawn from $D_{\mathcal{X}}$
- **Abilities to query label oracle $\mathcal{O}$**

Goal:

- Get a classifier $\hat{h}$ with excess error ϵ with probability $1 - \delta$

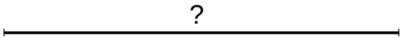
Label Complexity $m(\epsilon, \delta)$:

- How many **label queries** are needed to achieve this goal?

Special Challenges in PAC Active Learning

PAC active learning algorithms need to adapt to distribution since:

- Labels queries outside the support is not allowed
- Evaluation metric is classification error



Membership Query Model

PAC Model

PAC Active Learning Algorithms

- Disagreement-based Active Learning(DBAL) [CAL94, BBL09, DHM07, Han07, Han09, Kol10, HY12, Han14]..
- Confidence-based Active Learning(CBAL) [ZC14, BL13]
- Cluster-based Active Learning [DH08, UWBD13]

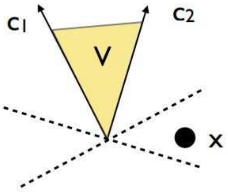
Outline

Outline

DBAL: Realizable Case [CAL94]

Main Idea:

- Maintain a set of candidate classifiers $V \subseteq \mathcal{H}$
- Query the label of an example x if x is in the disagreement region of V



Definition

Given a set of classifiers V, the disagreement region of V,

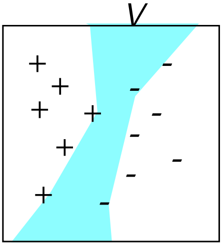
$$\mathrm{DIS}(V) := \{x : \text{there exist } h_1, h_2 \text{ in } V, h_1(x) \neq h_2(x)\}$$

Candidate Sets

- Realizable case: use version spaces as candidate sets

Definition

A version space V is the set of all classifiers h in hypothesis class $\mathcal{H}$ that agree with labeled examples seen so far.



DBAL: Algorithm

Input: target excess error ϵ, failure probability δ. Initialize candidate set $V_0 = \mathcal{H}$

For phases $k = 1, 2, \ldots, k_0 = \lceil \ln \frac{1}{\epsilon} \rceil$:

- Candidate set V_{k-1}, target excess error $\epsilon_k = 2^{-k}$

DBAL: Algorithm

Input: target excess error ϵ, failure probability δ. Initialize candidate set $V_0 = \mathcal{H}$

For phases $k = 1, 2, \ldots, k_0 = \lceil \ln \frac{1}{\epsilon} \rceil$:

- Candidate set V_{k-1}, target excess error $\epsilon_k = 2^{-k}$
- **Label Query:**

DBAL: Algorithm

Input: target excess error ϵ, failure probability δ. Initialize candidate set $V_0 = \mathcal{H}$

For phases $k = 1, 2, \ldots, k_0 = \lceil \ln \frac{1}{\epsilon} \rceil$:

- Candidate set V_{k-1}, target excess error $\epsilon_k = 2^{-k}$
- **Label Query:**

Where to query?

Labels of all x outside $\mathrm{DIS}(V_{k-1})$ are predictable

Query on the examples in $\mathrm{DIS}(V_{k-1})$

DBAL: Algorithm

Input: target excess error ϵ, failure probability δ. Initialize candidate set $V_0 = \mathcal{H}$

For phases $k = 1, 2, \ldots, k_0 = \lceil \ln \frac{1}{\epsilon} \rceil$:

- Candidate set V_{k-1}, target excess error $\epsilon_k = 2^{-k}$
- **Label Query:**

How many labels to query?

Enough s.t. excess error of each h in V_k is at most ϵ_k

Need $\approx \tilde{O}(\frac{d\mathbb{P}[\mathrm{DIS}(V_{k-1})]}{\epsilon_k})$ labels from $\mathrm{DIS}(V_{k-1})$

DBAL: Algorithm

Input: target excess error ϵ, failure probability δ. Initialize candidate set $V_0 = \mathcal{H}$

For phases $k = 1, 2, \ldots, k_0 = \lceil \ln \frac{1}{\epsilon} \rceil$:

- Candidate set V_{k-1}, target excess error $\epsilon_k = 2^{-k}$
- **Label Query:** $S_k \leftarrow$ Sample $\tilde{O}(d \frac{\mathbb{P}[\mathsf{DIS}(V_{k-1})]}{\epsilon_k})$ examples in $\mathsf{DIS}(V_{k-1})$ and query for labels

DBAL: Algorithm

Input: target excess error ϵ, failure probability δ. Initialize candidate set $V_0 = \mathcal{H}$
For phases $k = 1, 2, \ldots, k_0 = \lceil \ln \frac{1}{\epsilon} \rceil$:

- Candidate set V_{k-1}, target excess error $\epsilon_k = 2^{-k}$
- **Label Query:** $S_k \leftarrow$ Sample $\tilde{O}(d \frac{\mathbb{P}[\mathrm{DIS}(V_{k-1})]}{\epsilon_k})$ examples in $\mathrm{DIS}(V_{k-1})$ and query for labels
- **Prune Candidate Set:**

DBAL: Algorithm

Input: target excess error ϵ, failure probability δ. Initialize candidate set $V_0 = \mathcal{H}$

For phases $k = 1, 2, \ldots, k_0 = \lceil \ln \frac{1}{\epsilon} \rceil$:

- Candidate set V_{k-1}, target excess error $\epsilon_k = 2^{-k}$
- **Label Query:** $S_k \leftarrow$ Sample $\tilde{O}(d \frac{\mathbb{P}[\mathrm{DIS}(V_{k-1})]}{\epsilon_k})$ examples in $\mathrm{DIS}(V_{k-1})$ and query for labels
- **Prune Candidate Set:**

How to do the pruning?

Remove from V_{k-1} the classifiers that does not agree with S_k

DBAL: Algorithm

Input: target excess error ϵ, failure probability δ. Initialize candidate set $V_0 = \mathcal{H}$

For phases $k = 1, 2, \ldots, k_0 = \lceil \ln \frac{1}{\epsilon} \rceil$:

- Candidate set V_{k-1}, target excess error $\epsilon_k = 2^{-k}$
- **Label Query:** $S_k \leftarrow$ Sample $\tilde{O}(d \frac{\mathbb{P}[\mathrm{DIS}(V_{k-1})]}{\epsilon_k})$ examples in $\mathrm{DIS}(V_{k-1})$ and query for labels
- **Prune Candidate Set:**
  $V_k \leftarrow \{h \in V_{k-1} : h \text{ agrees with all } (x, y) \in S_k\}$

DBAL: Algorithm

Input: target excess error ϵ, failure probability δ. Initialize candidate set $V_0 = \mathcal{H}$

For phases $k = 1, 2, \ldots, k_0 = \lceil \ln \frac{1}{\epsilon} \rceil$:

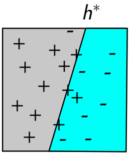
- Candidate set V_{k-1}, target excess error $\epsilon_k = 2^{-k}$
- **Label Query:** $S_k \leftarrow$ Sample $\tilde{O}(d \frac{\mathbb{P}[\mathrm{DIS}(V_{k-1})]}{\epsilon_k})$ examples in $\mathrm{DIS}(V_{k-1})$ and query for labels
- **Prune Candidate Set:**
  $V_k \leftarrow \left\{ h \in V_{k-1} : h \text{ agrees with all } (x, y) \in S_k \right\}$

Return $\hat{h} \leftarrow$ an arbitrary classifier from V_{k_0}.

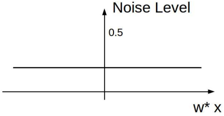
Outline

PAC Learning: Noise Models

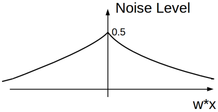
- Realizable: $\mathrm{err}(h^*) = 0$



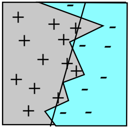
- η-Random classification noise (RCN): $\eta(x) = \eta \leq \frac{1}{2}$



- β-Tsybakov noise condition (TNC): $\mathbb{P}[\eta(x) \geq \frac{1}{2} - t] \leq O(t^{\frac{1}{\beta}})$

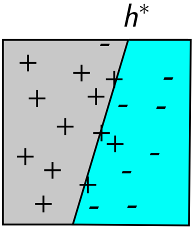


- Agnostic: optimal error $\mathrm{err}(h^*) = \nu$



DBAL: Non-Realizable Case

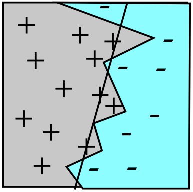
Realizable Case:



There is some h^* in $\mathcal{H}$ such that $h^*(x) = y$, for all $(x, y) \sim D$

Use version space as set of candidate classifiers

Non-Realizable Case:



h^* is the classifier in $\mathcal{H}$ with min error

Use $(1 - \delta)$ confidence set for h^* as candidate classifiers

Construction of Confidence Sets

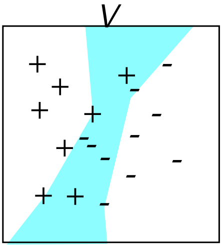
- Generalization bounds [VC71]: w.p. $1-\delta$ over the draw of a sample S of size m iid from D, for all h in $\mathcal{H}$,

$$|\operatorname{err}(h, S) - \operatorname{err}(h)| \leq \tilde{O}\left(\sqrt{\frac{d}{m}}\right)$$

- Choose: all h with

$$\operatorname{err}(h, S) \leq \min_{h' \in \mathcal{H}} \operatorname{err}(h', S) + \tilde{O}\left(\sqrt{\frac{d}{m}}\right)$$

- More careful construction needed in active learning



DBAL: Non-Realizable Case

Realizable Case:

There is some h^* in $\mathcal{H}$ such that $h^*(x) = y$, for all $(x, y) \sim D$

Use version space as set of candidate classifiers

At phase k, draw $\tilde{O}(d\frac{\mathbb{P}[\mathrm{DIS}(V_{k-1})]}{\epsilon_k})$ examples

Non-Realizable Case:

h^* is the classifier in $\mathcal{H}$ with min error

Use $(1 - \delta)$ confidence set for h^* as candidate classifiers

At phase k, adaptively draw enough examples for excess error $\frac{\epsilon_k}{\mathbb{P}[\mathrm{DIS}(V_{k-1})]}$ in disagreement region

DBAL: Algorithm in Non-Realizable Case

Input: target excess error ϵ, failure probability δ.
Initialize candidate set $V_0 = \mathcal{H}$
For phases $k = 1, 2, \ldots, k_0 = \lceil \ln \frac{1}{\epsilon} \rceil$:

- Candidate set V_{k-1}, target excess error $\epsilon_k = 2^{-k}$.

DBAL: Algorithm in Non-Realizable Case

Input: target excess error ϵ, failure probability δ.
Initialize candidate set $V_0 = \mathcal{H}$
For phases $k = 1, 2, \ldots, k_0 = \lceil \ln \frac{1}{\epsilon} \rceil$:

- Candidate set V_{k-1}, target excess error $\epsilon_k = 2^{-k}$.
- **Label Query:**

DBAL: Algorithm in Non-Realizable Case

Input: target excess error ϵ, failure probability δ.
Initialize candidate set $V_0 = \mathcal{H}$
For phases $k = 1, 2, \ldots, k_0 = \lceil \ln \frac{1}{\epsilon} \rceil$:

- Candidate set V_{k-1}, target excess error $\epsilon_k = 2^{-k}$.
- **Label Query:**

Where to query?
Query on the examples in $\mathrm{DIS}(V_{k-1})$

DBAL: Algorithm in Non-Realizable Case

Input: target excess error ϵ, failure probability δ.
Initialize candidate set $V_0 = \mathcal{H}$
For phases $k = 1, 2, \ldots, k_0 = \lceil \ln \frac{1}{\epsilon} \rceil$:

- Candidate set V_{k-1}, target excess error $\epsilon_k = 2^{-k}$.
- **Label Query:**

How many labels to query?
Enough s.t. excess error of each h in V_k is at most ϵ_k
Adaptively draw enough examples to achieve error at most $\frac{\epsilon_k}{\mathbb{P}[\mathrm{DIS}(V_{k-1})]}$ on $\mathrm{DIS}(V_{k-1})$

DBAL: Algorithm in Non-Realizable Case

Input: target excess error ϵ, failure probability δ.
Initialize candidate set $V_0 = \mathcal{H}$
For phases $k = 1, 2, \ldots, k_0 = \lceil \ln \frac{1}{\epsilon} \rceil$:

- Candidate set V_{k-1}, target excess error $\epsilon_k = 2^{-k}$.
- **Label Query:**
  $S_k \leftarrow$ Adaptively sample just enough examples on $\mathrm{DIS}(V_{k-1})$ and query for their labels to get target excess error $\frac{\epsilon_k}{\mathbb{P}[\mathrm{DIS}(V_{k-1})]}$

DBAL: Algorithm in Non-Realizable Case

Input: target excess error ϵ, failure probability δ.
Initialize candidate set $V_0 = \mathcal{H}$
For phases $k = 1, 2, \ldots, k_0 = \lceil \ln \frac{1}{\epsilon} \rceil$:

- Candidate set V_{k-1}, target excess error $\epsilon_k = 2^{-k}$.
- **Label Query:**
  $S_k \leftarrow$ Adaptively sample just enough examples on $\text{DIS}(V_{k-1})$ and query for their labels to get target excess error $\frac{\epsilon_k}{\mathbb{P}[\text{DIS}(V_{k-1})]}$
- **Prune Candidate Set:**

DBAL: Algorithm in Non-Realizable Case

Input: target excess error ϵ, failure probability δ.
Initialize candidate set $V_0 = \mathcal{H}$
For phases $k = 1, 2, \ldots, k_0 = \lceil \ln \frac{1}{\epsilon} \rceil$:

- Candidate set V_{k-1}, target excess error $\epsilon_k = 2^{-k}$.
- **Label Query:**
  $S_k \leftarrow$ Adaptively sample just enough examples on $\text{DIS}(V_{k-1})$ and query for their labels to get target excess error $\frac{\epsilon_k}{\mathbb{P}[\text{DIS}(V_{k-1})]}$
- **Prune Candidate Set:**

How to do the pruning?
Remove from V_{k-1} the classifiers that have a large empirical error on S_k

DBAL: Algorithm in Non-Realizable Case

Input: target excess error ϵ, failure probability δ.
Initialize candidate set $V_0 = \mathcal{H}$
For phases $k = 1, 2, \ldots, k_0 = \lceil \ln \frac{1}{\epsilon} \rceil$:

- Candidate set V_{k-1}, target excess error $\epsilon_k = 2^{-k}$.
- **Label Query:**
  $S_k \leftarrow$ Adaptively sample just enough examples on $\mathrm{DIS}(V_{k-1})$ and query for their labels to get target excess error $\frac{\epsilon_k}{\mathbb{P}[\mathrm{DIS}(V_{k-1})]}$
- **Prune Candidate Set:**

$$V_k \leftarrow \left\{ h \in V_{k-1} : \mathrm{err}(h, S_k) - \min_{h \in V_{k-1}} \mathrm{err}(h, S_k) \leq O\left(\frac{\epsilon_k}{\mathbb{P}[\mathrm{DIS}(V_{k-1})]} \right) \right\}$$

DBAL: Algorithm in Non-Realizable Case

Input: target excess error ϵ, failure probability δ.
Initialize candidate set $V_0 = \mathcal{H}$
For phases $k = 1, 2, \ldots, k_0 = \lceil \ln \frac{1}{\epsilon} \rceil$:

- Candidate set V_{k-1}, target excess error $\epsilon_k = 2^{-k}$.
- **Label Query:**
  $S_k \leftarrow$ Adaptively sample just enough examples on $\text{DIS}(V_{k-1})$ and query for their labels to get target excess error $\frac{\epsilon_k}{\mathbb{P}[\text{DIS}(V_{k-1})]}$
- **Prune Candidate Set:**

$$V_k \leftarrow \left\{ h \in V_{k-1} : \text{err}(h, S_k) - \min_{h \in V_{k-1}} \text{err}(h, S_k) \leq O\left(\frac{\epsilon_k}{\mathbb{P}[\text{DIS}(V_{k-1})]} \right) \right\}$$

Return $\hat{h} \leftarrow$ an arbitrary classifier from V_{k_0}.

DBAL: Algorithm in Non-Realizable Case

Input: target excess error ϵ, failure probability δ.
Initialize candidate set $V_0 = \mathcal{H}$
For phases $k = 1, 2, \ldots, k_0 = \lceil \ln \frac{1}{\epsilon} \rceil$:

- Candidate set V_{k-1}, target excess error $\epsilon_k = 2^{-k}$.
- **Label Query:**
  $S_k \leftarrow$ Adaptively sample just enough examples on $\text{DIS}(V_{k-1})$ and query for their labels to get target excess error $\frac{\epsilon_k}{\mathbb{P}[\text{DIS}(V_{k-1})]}$
- **Prune Candidate Set:**

$$V_k \leftarrow \left\{ h \in V_{k-1} : \text{err}(h, S_k) - \min_{h \in V_{k-1}} \text{err}(h, S_k) \leq O\left(\frac{\epsilon_k}{\mathbb{P}[\text{DIS}(V_{k-1})]} \right) \right\}$$

Return $\hat{h} \leftarrow$ an arbitrary classifier from V_{k_0}.
Computationally efficient implementation in [DHM07, BDL09, Han09, BHLZ10, HAH^{+}15]...

Outline

Statistical Consistency

Theorem

Suppose DBAL is run with parameters ϵ and δ. Then with probability $1-\delta$, the output $\hat{h}$ satisfies that

$$\mathrm{err}(\hat{h}) - \mathrm{err}(h^*) \leq \epsilon.$$

Main Idea:

- After phase k, all classifier in V_k have excess error $\leq \epsilon_k$
- Specifically, after phase k_0, all classifiers in V_{k_0} have excess error $\leq \epsilon_{k_0} \leq \epsilon$

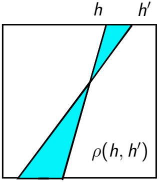
Label Complexity

Key factor: Shrinkage of $\mathbb{P}[\mathrm{DIS}(V_k)]$
Depends on:

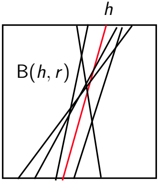
- Shrinkage of the V_k's radius
- Ratio of $\mathbb{P}[\mathrm{DIS}(V_k)]$ to the radius of V_k

Label Complexity: Definitions

- Disagreement metric:
  $\rho(h, h') = \mathbb{P}_D[h(x) \neq h'(x)]$



- Disagreement ball:
  $\mathrm{B}(h, r) = \{h' \in \mathcal{H} : \rho(h, h') \leq r\}$



Factor 1: Shrinkage of the V_k's Radius

Harder noise condition $\Rightarrow$ Slower shrinkage

Noise Model	radius(V_k)
Realizable	$\tilde{O}(\epsilon_k)$
η-RCN	$\tilde{O}(\frac{\epsilon_k}{1-2\eta})$
β-TNC	$\tilde{O}(\epsilon_k^{\frac{1}{1+\beta}})$
ν-Agnostic	$\tilde{O}(\nu + \epsilon_k)$

Version Space Radius Shrinkage under Noise Models

Factor 2: Disagreement Coefficient

Relating $\mathbb{P}[\mathrm{DIS}(V_k))]$ to V_k's radius

Definition ([Han07, Ale87, RR11])

Given a concept class $\mathcal{H}$, data distribution D, the disagreement coefficient with respect to $\mathcal{H}$ and D is defined as:

$$\theta = \sup_{h \in \mathcal{H}, r > 0} \frac{\mathbb{P}[\mathrm{DIS}(\mathrm{B}(h, r))]}{r}$$

Corollary

$$\mathbb{P}[\mathrm{DIS}(V)] \leq \theta \cdot \mathrm{radius}(V).$$

Shrinkage of Disagreement Region

Relationship $\mathbb{P}[\mathrm{DIS}(V_k)] \leq \theta \cdot \mathrm{radius}(V_k)$ implies:

Noise Model	$\mathbb{P}[\mathrm{DIS}(V_k)]$
Realizable	$\tilde{O}(\theta \cdot \epsilon_k)$
η-RCN	$\tilde{O}(\theta \cdot \frac{\epsilon_k}{1-2\eta})$
β-TNC	$\tilde{O}(\theta \cdot \epsilon^{\frac{1}{1+\beta}})$
ν-Agnostic	$\tilde{O}(\theta \cdot (\nu + \epsilon_k))$

Disagreement Region Shrinkage under Noise Models

Label Complexity Analysis: Main Idea

Realizable Case

- Label complexity in phase k: $m_k = \tilde{O}(d\frac{\mathbb{P}[\mathsf{DIS}(V_{k-1})]}{\epsilon_k}) = \tilde{O}(d\theta)$
- Total label complexity: $\sum_{k=1}^{k_0} m_k = \tilde{O}(d\theta \ln \frac{1}{\epsilon})$

The analysis can be extended to non-realizable cases straightforwardly.

Label Complexity

Theorem

Suppose DBAL is run with parameters ϵ and δ. Then with probability $1 - \delta$, the number of label requests is:

Noise Model	Label Complexity
Realizable	$\tilde{O}(\theta \cdot d \cdot \ln \frac{1}{\epsilon})$
η-RCN	$\tilde{O}(\theta \cdot \frac{d}{(1-2\eta)^2} \cdot \ln \frac{1}{\epsilon})$
β-TNC	$\tilde{O}(\theta \cdot d \cdot \epsilon^{\frac{2}{1+\beta}-2})$
ν-Agnostic	$\tilde{O}(\theta \cdot d \cdot \frac{(\nu+\epsilon)^2}{\epsilon^2})$

Comparison to Passive Learning

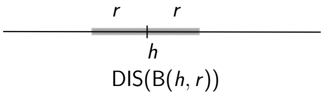
DBAL improves over passive learning if θ is finite

Noise Model	Improvement Factor
Realizable	$\tilde{O}(\theta\epsilon)$
η-RCN	$\tilde{O}(\frac{\theta\epsilon}{1-2\eta})$
β-TNC	$\tilde{O}(\theta\epsilon^{\frac{1}{1+\beta}})$
ν-Agnostic	$\tilde{O}(\theta(\nu+\epsilon))$

Disagreement Coefficient: Examples

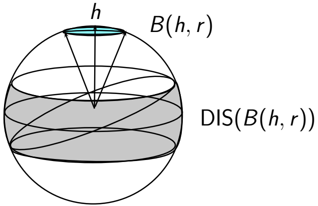
Thresholds

- $D_{\mathcal{X}}$: Uniform([0, 1])
- $\mathcal{H}$: threshold classifiers $h_t(x) = I(x \geq t)$, $t \in [0, 1]$
- $\theta \leq 2$



Linear Classification

- $D_{\mathcal{X}}$: uniform over unit sphere
- $\mathcal{H}$: linear classifiers through the origin $h_w = \text{sign}(w \cdot x)$, $w \in \mathbb{R}^d$
- $\theta = O(\sqrt{d})$



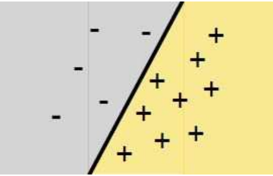
Outline

Confidence-based Active Learning(CBAL)

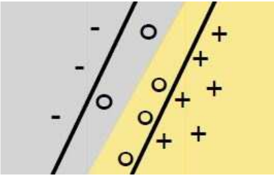
- The label query policy of DBAL is too conservative
  - perform label query as long as an example is in disagreement region
- Idea of CBAL: select a subset of disagreement region using confidence-rated predictors

Confidence-based Active Learning(CBAL)

- Confidence-rated predictor(CRP): classifiers that can say "Don't know" ($\perp$)



Output of a binary classifier



Output of a CRP

- Main idea of CBAL:
  - Maintain a confidence-rated predictor $\mathcal{P}$
  - Use $\mathcal{P}$ to make label query decision: Query the label of x if $\mathcal{P}$ says "Don't know" on x

CBAL: Algorithmic Framework

- Inputs: target excess error ϵ, failure probability δ.
- Initialization: $V_0 \leftarrow \mathcal{H}$.
- For phase $k = 1$ to $k_0 = \lceil \log \frac{1}{\epsilon} \rceil$:
  - Candidate set V_{k-1}, target excess error $\epsilon_k = 2^{-k}$.
  - **Transduction**: Draw a set of $\tilde{O}(\frac{d}{\epsilon_k^2})$ unlabeled examples U_k iid from $D_{\mathcal{X}}$.
  - **Selection**: Run Algorithm **CRP** on U_k with error guarantee $O(\epsilon_k)$ with uncertainty set V_{k-1}, get abstention probability $\{\gamma_i\}_{i=1}^{n_k}$, normalize it to a distribution Γ_k. Let $\phi_k = \text{abs}_{U_k}(\mathcal{P}) = \frac{1}{n_k} \sum_{i=1}^{n_k} \gamma_i$.
  - **Label Query**: $S_k \leftarrow$ Adaptively sample just enough examples to get target excess error $O(\frac{\epsilon_k}{\phi_k})$ on Γ_k and query their labels.
  - **Prune Candidate Set**: Update candidate set

$$V_k \leftarrow \left\{ h \in V_{k-1} : \text{err}(h, S_k) - \min_{h \in V_{k-1}} \text{err}(h, S_k) \le O\left(\frac{\epsilon_k}{\phi_k}\right) \right\}$$

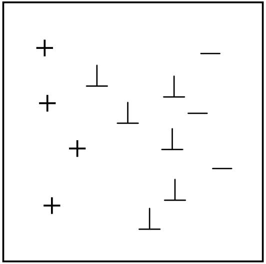
- Return $\hat{h} \leftarrow$ an arbitrary classifier in V_{k_0}.

Confidence-rated Prediction [EYW10]

- Given x, $\mathcal{P}(x) \in \{-1, +1, \perp\}$

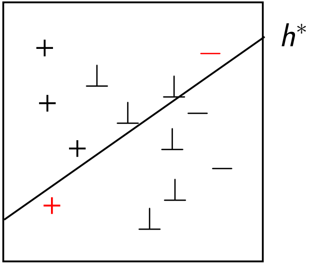
Confidence-rated Prediction [EYW10]

- Given x, $\mathcal{P}(x) \in \{-1, +1, \perp\}$



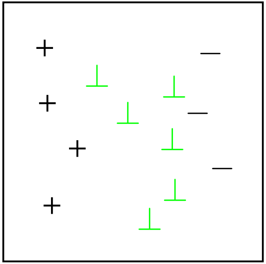
Confidence-rated Prediction [EYW10]

- Given x, $\mathcal{P}(x) \in \{-1, +1, \perp\}$
- Error: $\text{err}_D(\mathcal{P}) = \mathbb{P}_D[\mathcal{P}(x) \neq h^*(x), \mathcal{P}(x) \neq \perp]$



Confidence-rated Prediction [EYW10]

- Given x, $\mathcal{P}(x) \in \{-1, +1, \perp\}$
- Error: $\mathrm{err}_D(\mathcal{P}) = \mathbb{P}_D[\mathcal{P}(x) \neq h^*(x), \mathcal{P}(x) \neq \perp]$
- Abstention: $\mathrm{abs}_D(\mathcal{P}) = \mathbb{P}_D[\mathcal{P}(x) = \perp]$

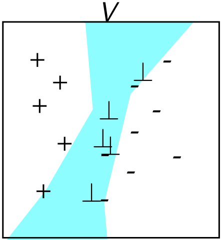


Confidence-rated Predictor in Transductive Setting

- Transductive Setting: given unlabeled examples $U = \{x_1, \ldots, x_n\}$ drawn from $D_{\mathcal{X}}$, make predictions on U
- "Soft" prediction: $\mathcal{P}(x_i) = \begin{cases} +1 & \text{w.p. } \xi_i \\ -1 & \text{w.p. } \zeta_i \\ \perp & \text{w.p. } \gamma_i \end{cases}$
- A confidence-rated predictor $\mathcal{P}$ on U is described as n 3-tuples: $\{(\xi_i, \zeta_i, \gamma_i)\}_{i=1}^n$
- Error: $\mathrm{err}_U(\mathcal{P}) = \frac{1}{n}\sum_{i=1}^n \mathbb{1}[h^*(x_i) = -1]\xi_i + \mathbb{1}[h^*(x_i) = +1]\zeta_i$
- Abstention: $\mathrm{abs}_U(\mathcal{P}) = \frac{1}{n}\sum_{i=1}^n \gamma_i$

A Confidence-rated Predictor with Guaranteed Error

- Given set of unlabeled examples U
- Given uncertainty set of classifiers V, h^* is known to be in V
- Error guarantee η: $\text{err}_U(\mathcal{P}) \leq \eta$



A Confidence-rated Predictor with Guaranteed Error

- Algorithm **CRP**
- Input: uncertainty set V, unlabeled set U, error guarantee η
- Construct a linear program:

$$\min \frac{1}{n} \sum_{i=1}^{n} \gamma_i$$

subject to:

$$\forall i,\ \ \xi_i + \zeta_i + \gamma_i = 1$$

$$\forall i,\ \ \xi_i, \zeta_i, \gamma_i \geq 0$$

$$\forall h \in V,\ \ \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}[h(x_i) = -1]\xi_i + \mathbb{1}[h(x_i) = +1]\zeta_i \leq \eta$$

- Confidence-rated predictor $\mathcal{P}$ returned is described as the optimal solution of the LP $\{(\xi_i^*, \zeta_i^*, \gamma_i^*)\}_{i=1}^{n}$

CBAL: Algorithm

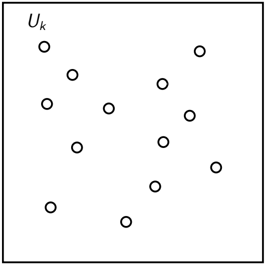
- Inputs: target excess error ϵ, failure probability δ.
- Initialize candidate set $V_0 = \mathcal{H}$.

CBAL: Algorithm

- For phase $k = 1$ to $k_0 = \lceil \log \frac{1}{\epsilon} \rceil$:
  - Candidate set V_{k-1}, target excess error $\epsilon_k = 2^{-k}$

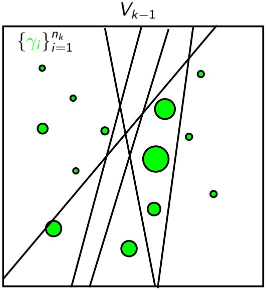
CBAL: Algorithm

- For phase $k = 1$ to $k_0 = \lceil \log \frac{1}{\epsilon} \rceil$:
  - Candidate set V_{k-1}, target excess error $\epsilon_k = 2^{-k}$
  - **Transduction**: Draw a set of $\tilde{O}(\frac{d}{\epsilon_k^2})$ unlabeled examples U_k iid from $D_{\mathcal{X}}$.



CBAL: Algorithm

- For phase $k = 1$ to $k_0 = \lceil \log \frac{1}{\epsilon} \rceil$:
  - Candidate set V_{k-1}, target excess error $\epsilon_k = 2^{-k}$
  - **Transduction**: Draw a set of $\tilde{O}(\frac{d}{\epsilon_k^2})$ unlabeled examples U_k iid from $D_{\mathcal{X}}$.
  - **Selection**: Run Algorithm **CRP** on U_k with error guarantee $O(\epsilon_k)$ with uncertainty set V_{k-1}, get abstention probability $\{\gamma_i\}_{i=1}^{n_k}$, normalize it to a distribution Γ_k. Let $\phi_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \gamma_i$.



CBAL: Algorithm

- For phase $k = 1$ to $k_0 = \lceil \log \frac{1}{\epsilon} \rceil$:
  - Candidate set V_{k-1}, target excess error $\epsilon_k = 2^{-k}$
  - **Transduction**: Draw a set of $\tilde{O}(\frac{d}{\epsilon_k^2})$ unlabeled examples U_k iid from $D_{\mathcal{X}}$.
  - **Selection**: Run Algorithm **CRP** on U_k with error guarantee $O(\epsilon_k)$ with uncertainty set V_{k-1}, get abstention probability $\{\gamma_i\}_{i=1}^{n_k}$, normalize it to a distribution Γ_k. Let $\phi_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \gamma_i$.
  - **Label Query**:

Where to query?
Query on the examples drawn from distribution Γ_k

CBAL: Algorithm

- For phase $k = 1$ to $k_0 = \lceil \log \frac{1}{\epsilon} \rceil$:
  - Candidate set V_{k-1}, target excess error $\epsilon_k = 2^{-k}$
  - **Transduction**: Draw a set of $\tilde{O}(\frac{d}{\epsilon_k^2})$ unlabeled examples U_k iid from $D_{\mathcal{X}}$.
  - **Selection**: Run Algorithm **CRP** on U_k with error guarantee $O(\epsilon_k)$ with uncertainty set V_{k-1}, get abstention probability $\{\gamma_i\}_{i=1}^{n_k}$, normalize it to a distribution Γ_k. Let $\phi_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \gamma_i$.
  - **Label Query**:

How many labels to query?
Enough s.t. excess error of each h in V_k is at most ϵ_k
Adaptively draw enough examples to achieve error at most $O(\frac{\epsilon_k}{\phi_k})$ on Γ_k

CBAL: Algorithm

- For phase $k = 1$ to $k_0 = \lceil \log \frac{1}{\epsilon} \rceil$:
  - Candidate set V_{k-1}, target excess error $\epsilon_k = 2^{-k}$
  - **Transduction**: Draw a set of $\tilde{O}(\frac{d}{\epsilon_k^2})$ unlabeled examples U_k iid from $D_{\mathcal{X}}$.
  - **Selection**: Run Algorithm **CRP** on U_k with error guarantee $O(\epsilon_k)$ with uncertainty set V_{k-1}, get abstention probability $\{\gamma_i\}_{i=1}^{n_k}$, normalize it to a distribution Γ_k. Let $\phi_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \gamma_i$.
  - **Label Query**: $S_k \leftarrow$ Adaptively sample just enough examples to get target excess error $O(\frac{\epsilon_k}{\phi_k})$ on Γ_k and query their labels.

CBAL: Algorithm

- For phase $k = 1$ to $k_0 = \lceil \log \frac{1}{\epsilon} \rceil$:
  - Candidate set V_{k-1}, target excess error $\epsilon_k = 2^{-k}$
  - **Transduction**: Draw a set of $\tilde{O}(\frac{d}{\epsilon_k^2})$ unlabeled examples U_k iid from $D_{\mathcal{X}}$.
  - **Selection**: Run Algorithm **CRP** on U_k with error guarantee $O(\epsilon_k)$ with uncertainty set V_{k-1}, get abstention probability $\{\gamma_i\}_{i=1}^{n_k}$, normalize it to a distribution Γ_k. Let $\phi_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \gamma_i$.
  - **Label Query**:
    $S_k \leftarrow$ Adaptively sample just enough examples to get target excess error $O(\frac{\epsilon_k}{\phi_k})$ on Γ_k and query their labels.
  - **Prune Candidate Set**:

CBAL: Algorithm

- For phase $k = 1$ to $k_0 = \lceil \log \frac{1}{\epsilon} \rceil$:
  - Candidate set V_{k-1}, target excess error $\epsilon_k = 2^{-k}$
  - **Transduction**: Draw a set of $\tilde{O}(\frac{d}{\epsilon_k^2})$ unlabeled examples U_k iid from $D_{\mathcal{X}}$.
  - **Selection**: Run Algorithm **CRP** on U_k with error guarantee $O(\epsilon_k)$ with uncertainty set V_{k-1}, get abstention probability $\{\gamma_i\}_{i=1}^{n_k}$, normalize it to a distribution Γ_k. Let $\phi_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \gamma_i$.
  - **Label Query**:
    $S_k \leftarrow$ Adaptively sample just enough examples to get target excess error $O(\frac{\epsilon_k}{\phi_k})$ on Γ_k and query their labels.
  - **Prune Candidate Set**:

How to do the pruning?
Remove from V_{k-1} the classifiers that have a large empirical error on S_k

CBAL: Algorithm

- For phase $k = 1$ to $k_0 = \lceil \log \frac{1}{\epsilon} \rceil$:
  - Candidate set V_{k-1}, target excess error $\epsilon_k = 2^{-k}$
  - **Transduction**: Draw a set of $\tilde{O}(\frac{d}{\epsilon_k^2})$ unlabeled examples U_k iid from $D_{\mathcal{X}}$.
  - **Selection**: Run Algorithm **CRP** on U_k with error guarantee $O(\epsilon_k)$ with uncertainty set V_{k-1}, get abstention probability $\{\gamma_i\}_{i=1}^{n_k}$, normalize it to a distribution Γ_k. Let $\phi_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \gamma_i$.
  - **Label Query**:
    $S_k \leftarrow$ Adaptively sample just enough examples to get target excess error $O(\frac{\epsilon_k}{\phi_k})$ on Γ_k and query their labels.
  - **Prune Candidate Set**:
    Update candidate set

$$V_k \leftarrow \left\{ h \in V_{k-1} : \mathrm{err}(h, S_k) - \min_{h \in V_{k-1}} \mathrm{err}(h, S_k) \leq O\left(\frac{\epsilon_k}{\phi_k}\right) \right\}$$

CBAL: Algorithm

- For phase $k = 1$ to $k_0 = \lceil \log \frac{1}{\epsilon} \rceil$:
  - Candidate set V_{k-1}, target excess error $\epsilon_k = 2^{-k}$
  - **Transduction**: Draw a set of $\tilde{O}(\frac{d}{\epsilon_k^2})$ unlabeled examples U_k iid from $D_{\mathcal{X}}$.
  - **Selection**: Run Algorithm **CRP** on U_k with error guarantee $O(\epsilon_k)$ with uncertainty set V_{k-1}, get abstention probability $\{\gamma_i\}_{i=1}^{n_k}$, normalize it to a distribution Γ_k. Let $\phi_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \gamma_i$.
  - **Label Query**:
    $S_k \leftarrow$ Adaptively sample just enough examples to get target excess error $O(\frac{\epsilon_k}{\phi_k})$ on Γ_k and query their labels.
  - **Prune Candidate Set**:
    Update candidate set

$$V_k \leftarrow \left\{ h \in V_{k-1} : \mathrm{err}(h, S_k) - \min_{h \in V_{k-1}} \mathrm{err}(h, S_k) \leq O\left(\frac{\epsilon_k}{\phi_k}\right) \right\}$$

- Return $\hat{h} \leftarrow$ an arbitrary classifier in V_{k_0}.

CBAL: Statistical Consistency

Theorem
Suppose CBAL is run with parameters ϵ and δ. Then with probability $1 - \delta$, the output $\hat{h}$ satisfies that

$$\mathrm{err}(\hat{h}) - \mathrm{err}(h^*) \leq \epsilon.$$

CBAL: Label Complexity

- $\Phi(V, \eta)$: the minimum abstention probability of a confidence-rated predictor with uncertainty set V with error guarantee η under distribution $D_{\mathcal{X}}$
- $\Phi(V, \eta) \leq \Phi(V, 0) \leq \mathbb{P}_D[\mathrm{DIS}(V)]$
- Define confidence coefficient $\sigma(\eta) := \sup_{r>0} \frac{\Phi(\mathrm{B}(h^*, r), \eta)}{r}$
- $\sigma(\eta) \leq \theta$ and can sometimes be much smaller

CBAL: Shrinkage of Uncertainty Region

The size of the sampling region again depends on:

- radius of confidence set V_k
- confidence coefficient σ

Noise Model	Size of Uncertainty Region
Realizable	$\tilde{O}(\sigma(\epsilon_k) \cdot \epsilon_k)$
η-RCN	$\tilde{O}(\sigma(\epsilon_k) \cdot \frac{\epsilon_k}{1-2\eta})$
β-TNC	$\tilde{O}(\sigma(\epsilon_k) \cdot \epsilon_k^{\frac{1}{1+\beta}})$
ν-Agnostic	$\tilde{O}(\sigma(\epsilon_k) \cdot (\nu + \epsilon_k))$

Uncertainty Region Shrinkage in CBAL

CBAL: Label Complexity

Theorem

Suppose CBAL is run with parameters ϵ and δ. With probability $1-\delta$, the number of label requests is

Noise Model	Label Complexity
Realizable	$\tilde{O}(\sigma(\epsilon) \cdot d \cdot \ln \frac{1}{\epsilon})$
η-RCN	$\tilde{O}(\sigma(\epsilon) \cdot \frac{d}{(1-2\eta)^2} \cdot \ln \frac{1}{\epsilon})$
β-TNC	$\tilde{O}(\sigma(\epsilon) \cdot d \cdot \epsilon^{\frac{2}{1+\beta}-2})$
ν-Agnostic	$\tilde{O}(\sigma(\epsilon) \cdot d \cdot \frac{(\nu+\epsilon)^2}{\epsilon^2})$

Comparison

CBAL improves over DBAL by replacing θ with $\sigma(\epsilon)$ in label complexity

Noise Model	DBAL	CBAL
Realizable	$\tilde{O}(\theta \cdot d \cdot \ln \frac{1}{\epsilon})$	$\tilde{O}(\sigma(\epsilon) \cdot d \cdot \ln \frac{1}{\epsilon})$
η-RCN	$\tilde{O}(\theta \cdot \frac{d}{(1-2\eta)^2} \cdot \ln \frac{1}{\epsilon})$	$\tilde{O}(\sigma(\epsilon) \cdot \frac{d}{(1-2\eta)^2} \cdot \ln \frac{1}{\epsilon})$
β-TNC	$\tilde{O}(\theta \cdot d \cdot \epsilon^{\frac{2}{1+\beta}-2})$	$\tilde{O}(\sigma(\epsilon) \cdot d \cdot \epsilon^{\frac{2}{\kappa}-2})$
ν-Agnostic	$\tilde{O}(\theta \cdot d \cdot \frac{(\nu+\epsilon)^2}{\epsilon^2})$	$\tilde{O}(\sigma(\epsilon) \cdot d \cdot \frac{(\nu+\epsilon)^2}{\epsilon^2})$

Example: linear classification under uniform distribution

- $\sigma(\epsilon) = O(\min(\sqrt{d}, \ln \frac{1}{\epsilon}))$ [BBZ07, BL13], whereas $\theta = O(\sqrt{d})$
- CBAL improves over DBAL by a factor of $\tilde{O}(\sqrt{d})$ in label complexity

Outline

Conclusions and Open Problems

- DBAL: general, statistically consistent, relatively high label complexity
- CBAL: general, statistically consistent, lower label complexity
- Open Problems:
  - Better algorithms for statistically consistent active learning
  - Computationaly efficiency
  - New notion of soft confidence in active learning

Thank you!
Questions?

References I

Kenneth S Alexander.
Rates of growth and sample moduli for weighted empirical processes indexed by sets.
Probability Theory and Related Fields, 75(3):379–423, 1987.

M.-F. Balcan, A. Beygelzimer, and J. Langford.
Agnostic active learning.
J. Comput. Syst. Sci., 75(1):78–89, 2009.

M.-F. Balcan, A. Z. Broder, and T. Zhang.
Margin based active learning.
In *COLT*, 2007.

A. Beygelzimer, S. Dasgupta, and J. Langford.
Importance weighted active learning.
In *ICML*, 2009.

References II

A. Beygelzimer, D. Hsu, J. Langford, and T. Zhang.
Agnostic active learning without constraints.
In *NIPS*, 2010.

M.-F. Balcan and P. M. Long.
Active and passive learning of linear separators under log-concave distributions.
In *COLT*, 2013.

D. A. Cohn, L. E. Atlas, and R. E. Ladner.
Improving generalization with active learning.
Machine Learning, 15(2), 1994.

S. Dasgupta and D. Hsu.
Hierarchical sampling for active learning.
In *ICML*, 2008.

References III

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

R. El-Yaniv and Y. Wiener.
On the foundations of noise-free selective classification.
JMLR, 2010.

Tzu-Kuo Huang, Alekh Agarwal, Daniel J Hsu, John Langford, and Robert E Schapire.
Efficient and parsimonious agnostic active learning.
In *Advances in Neural Information Processing Systems*, pages 2755–2763, 2015.

S. Hanneke.
A bound on the label complexity of agnostic active learning.
In *ICML*, 2007.

References IV

S. Hanneke.
Theoretical Foundations of Active Learning.
PhD thesis, Carnegie Mellon University, 2009.

Steve Hanneke.
Theory of disagreement-based active learning.
Foundations and Trends® in Machine Learning,
7(2-3):131–309, 2014.

S. Hanneke and L. Yang.
Surrogate losses in passive and active learning.
CoRR, abs/1207.3772, 2012.

V. Koltchinskii.
Rademacher complexities and bounding the excess risk in active learning.
JMLR, 2010.

References V

Maxim Raginsky and Alexander Rakhlin.
Lower bounds for passive and active learning.
In *Advances in Neural Information Processing Systems*, pages 1026–1034, 2011.

R. Urner, S. Wulff, and S. Ben-David.
Plal: Cluster-based active learning.
In *COLT*, 2013.

V. N. Vapnik and A. Ya. Chervonenkis.
On the uniform convergence of relative frequencies of events to their probabilities.
Theory of Probability and its Applications, 16(2):264–280, 1971.

C. Zhang and K. Chaudhuri.
Beyond disagreement-based agnostic active learning.
In *NIPS*, 2014.