Active learning: The classics

Christopher Tosh

Columbia University

TRIPODS Bootcamp
Supervised learning pipeline
Supervised learning pipeline

Cheap!

Expensive!

Active learning
Active learning
Active learning
A quick example: linear thresholds

Linear threshold:

\[ h^*(x) = \begin{cases} + & \text{if } x > v^* \\ - & \text{if } x \leq v^* \end{cases} \]
A quick example: linear thresholds

Supervised approach:

- Draw $O(1/\epsilon)$ labeled data points
- Any consistent threshold $h$ has error $\text{err}(h) \leq \epsilon$
Active learning

A quick example: linear thresholds

Supervised approach:

- Draw $O(1/\epsilon)$ labeled data points
- Any *consistent* threshold $h$ has error $\text{err}(h) \leq \epsilon$
A quick example: linear thresholds

Active learning approach:

- Draw $O(1/\epsilon)$ unlabeled data points
- Repeatedly query median unlabeled point and infer labels for some unlabeled points
- Stop when there are two adjacent points of different labels
A quick example: linear thresholds

Active learning approach:
- Draw $O(1/\epsilon)$ unlabeled data points
- Repeatedly query median unlabeled point and infer labels for some unlabeled points
- Stop when there are two adjacent points of different labels
Active learning approach:
- Draw $O(1/\epsilon)$ unlabeled data points
- Repeatedly query median unlabeled point and infer labels for some unlabeled points
- Stop when there are two adjacent points of different labels

A quick example: linear thresholds
A quick example: linear thresholds

Active learning approach:
- Draw $O(1/\epsilon)$ unlabeled data points
- Repeatedly query median unlabeled point and infer labels for some unlabeled points
- Stop when there are two adjacent points of different labels
A quick example: linear thresholds

Active learning approach:

- Draw $O(1/\epsilon)$ unlabeled data points
- Repeatedly query median unlabeled point and infer labels for some unlabeled points
- Stop when there are two adjacent points of different labels
Active learning approach:

- Draw $O(1/\epsilon)$ unlabeled data points
- Repeatedly query median unlabeled point and infer labels for some unlabeled points
- Stop when there are two adjacent points of different labels
A quick example: linear thresholds

Active learning approach:

- Draw $O(1/\epsilon)$ unlabeled data points
- Repeatedly query median unlabeled point and infer labels for some unlabeled points
- Stop when there are two adjacent points of different labels
A quick example: linear thresholds

Active learning approach:
- Draw $O(1/\epsilon)$ unlabeled data points
- Repeatedly query median unlabeled point and infer labels for some unlabeled points
- Stop when there are two adjacent points of different labels

Number of labels requested: $O(\log 1/\epsilon)$
Overview

- **Today**: General hypothesis classes
  - Mellow
  - Aggressive

- **Tomorrow**: Interactive learning
  - Nonparametric active learning
  - Interactive clustering
A partition of (some) active learning work

<table>
<thead>
<tr>
<th>Aggressive</th>
<th>Separable data</th>
<th>General (nonseparable) data</th>
</tr>
</thead>
<tbody>
<tr>
<td>QBC [FSST97]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Splitting index</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[D05]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GBS [D04, N09]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A^2 algorithm [BBL06, H07]</td>
<td>Reduction to supervised [DHM07]</td>
<td>Importance weighted [BDL09]</td>
</tr>
<tr>
<td></td>
<td>Confidence rated prediction [ZC14]</td>
<td></td>
</tr>
<tr>
<td>Mellow</td>
<td>CAL [CAL94]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A partition of (some) active learning work

<table>
<thead>
<tr>
<th></th>
<th>Separable data</th>
<th>General (nonseparable) data</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Aggressive</strong></td>
<td>QBC [FSST97]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Splitting index [D05]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GBS [D04, N09]</td>
<td></td>
</tr>
<tr>
<td><strong>Mellow</strong></td>
<td>CAL [CAL94]</td>
<td>A² algorithm [BBL06, H07]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reduction to supervised [DHM07]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Importance weighted [BDL09]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Confidence rated prediction [ZC14]</td>
</tr>
</tbody>
</table>
Noiseless realizable setting

- Fixed binary hypothesis class $\mathcal{H}$
- Realizable: some true hypothesis $h^* \in \mathcal{H}$
- Noiseless: query $x$ and observe $h^*(x)$
- Pool of unlabeled data drawn from $\mathcal{D}$ (essentially unlimited)
- **Goal**: learn low error hypothesis $h \in \mathcal{H}$ –
  $$\text{err}(h) = \Pr_{x \sim \mathcal{D}}(h(x) \neq h^*(x))$$
Active learning: Version spaces

**Version space**: set of hypotheses consistent with all the labels seen so far.

- Start with version space $V_0 = \mathcal{H}$.
- For $t = 1, 2, \ldots$
  - Query $x_t$ and observe label $y_t = h^*(x_t)$.
  - Set $V_t = \{ h \in V_{t-1} : h(x_t) = y_t \}$. 
Active learning: Version spaces

**Version space**: set of hypotheses consistent with all the labels seen so far.

- Start with version space $V_0 = \mathcal{H}$.
- For $t = 1, 2, \ldots$
  - Query $x_t$ and observe label $y_t = h^*(x_t)$.
  - Set $V_t = \{ h \in V_{t-1} : h(x_t) = y_t \}$. 

![Diagram of version space with examples](image.png)
Active learning: Version spaces

**Version space**: set of hypotheses consistent with all the labels seen so far.

- Start with version space $V_0 = \mathcal{H}$.
- For $t = 1, 2, \ldots$
  - Query $x_t$ and observe label $y_t = h^*(x_t)$.
  - Set $V_t = \{h \in V_{t-1} : h(x_t) = y_t\}$.
Active learning: Version spaces

**Version space**: set of hypotheses consistent with all the labels seen so far.

- Start with version space $V_0 = \mathcal{H}$.
- For $t = 1, 2, \ldots$
  - Query $x_t$ and observe label $y_t = h^*(x_t)$.
  - Set $V_t = \{h \in V_{t-1} : h(x_t) = y_t\}$.
Active learning: Version spaces

**Version space**: set of hypotheses consistent with all the labels seen so far.

- Start with version space \( V_0 = \mathcal{H} \).
- For \( t = 1, 2, \ldots \)
  - Query \( x_t \) and observe label \( y_t = h^*(x_t) \).
  - Set \( V_t = \{ h \in V_{t-1} : h(x_t) = y_t \} \).
Active learning: Version spaces

**Version space**: set of hypotheses consistent with all the labels seen so far.

- **Start with version space** $V_0 = \mathcal{H}$.
- For $t = 1, 2, \ldots$
  - Query $x_t$ and observe label $y_t = h^*(x_t)$.
  - Set $V_t = \{ h \in V_{t-1} : h(x_t) = y_t \}$.

**Observation**: $h^* \in V_t$ for $t = 0, 1, 2, \ldots$
A mellow strategy: CAL

Strategy:

- Randomly sample \( x \sim \mathcal{D} \)
- Query \( x \) if there are two hypotheses \( h, h' \in V_t \) satisfying

\[
h(x) \neq h'(x)
\]
A mellow strategy: CAL

Strategy:

- Randomly sample $x \sim \mathcal{D}$
- Query $x$ if there are two hypotheses $h, h' \in V_t$ satisfying

$$h(x) \neq h'(x)$$

Properties:

- Simple
- Consistent
- Label complexity of CAL $\leq$ Label complexity of random strategy
- Efficient to implement*
CAL: Label complexity

For two hypotheses $h, h' \in \mathcal{H}$, define

$$d(h, h') = \Pr_{x \sim \mathcal{D}}(h(x) \neq h'(x)).$$

Define a ball of radius $r$ as

$$B(h, r) = \{h' \in \mathcal{H} : d(h, h') \leq r\}$$

Define the disagreement region of radius $r$ around $h$ as

$$\text{DIS}(h, r) = \{x : \exists h_1, h_2 \in B(h, r) \text{ s.t. } h_1(x) \neq h_2(x)\}.$$ 

Then for target hypothesis $h^*$, disagreement coefficient is

$$\theta = \sup_{r \in (0,1)} \frac{\Pr_{x \sim \mathcal{D}}(x \in \text{DIS}(h^*, r))}{r}.$$
Disagreement coefficient: Example

Linear thresholds:

\[ h^*(x) = \begin{cases} 
+ & \text{if } x > v^* \\
- & \text{if } x \leq v^* 
\end{cases} \]
Disagreement coefficient: Example

\[ h(x) \neq h'(x) \text{ iff } x \in \text{green region} \implies d(h, h') = \Pr(x \in \text{green region}) \]
Disagreement coefficient: Example

\[ d(h^*, h_L) = r = d(h^*, h_R) \]

\[ B(h^*, r) = \text{blue region} = \text{DIS}(h^*, r) \]
Disagreement coefficient: Example

\[ d(h^*, h_L) = r = d(h^*, h_R) \]

\[ \Pr(x \in \text{DIS}(h^*, r)) = \Pr(x \in I_L) + \Pr(x \in I_R) = d(h^*, h_L) + d(h^*, h_R) = 2r \]

\[ \theta = \sup_{r \in (0,1)} \frac{\Pr_{x \sim \mathcal{D}}(x \in \text{DIS}(h^*, r))}{r} = 2. \]
Other cases:

- Thresholds: $\theta = 2$

- Homogeneous linear separators under uniform distribution: $\theta \leq \sqrt{d}$

- Intervals of width $w$ under uniform distribution: $\theta = \max \left\{ \frac{1}{w}, 4 \right\}$

- Finite hypothesis classes: $\theta \leq |\mathcal{H}|$. 
CAL: Label complexity

**Theorem**

If VC-dimension of $\mathcal{H}$ is $d$ and disagreement coefficient is $\theta$, then

$$\# \text{ of labels requested by CAL} \leq \tilde{O}\left(d \theta \log \frac{1}{\epsilon}\right)$$
Theorem

If VC-dimension of $\mathcal{H}$ is $d$ and disagreement coefficient is $\theta$, then

$$\# \text{ of labels requested by CAL} \leq \tilde{O} \left( d\theta \log \frac{1}{\epsilon} \right)$$

Compare to passive learning:

$$\# \text{ of labels needed for passive learning} \geq \Omega \left( \frac{d}{\epsilon} \right)$$
CAL: Label complexity proof

Start with $V_0 = \mathcal{H}$

For $t = 1, 2, \ldots$:

- Draw unlabeled point $x_t \sim \mathcal{D}$
- If $\exists h, h' \in V_{t-1}$ s.t. $h(x_t) \neq h'(x_t)$, query for label $y_t$
- Otherwise, create pseudo-label $\tilde{y}_t$
- Update $V_t = \{ h \in V_{t-1} : h(x_t) = y_t \text{ (or } \tilde{y}_t) \}$
CAL: Label complexity proof

Start with $V_0 = \mathcal{H}$

For $t = 1, 2, \ldots$:

- Draw unlabeled point $x_t \sim \mathcal{D}$
- If $\exists h, h' \in V_{t-1}$ s.t. $h(x_t) \neq h'(x_t)$, query for label $y_t$
- Otherwise, create pseudo-label $\tilde{y}_t$
- Update $V_t = \{ h \in V_{t-1} : h(x_t) = y_t \text{ (or } \tilde{y}_t) \}$

Observation 1: We always have $h^*(x_t) = y_t$ (or $\tilde{y}_t$).

Observation 2: The (pseudo)-labeled dataset $(x_1, y_1/\tilde{y}_1), \ldots, (x_n, y_n/\tilde{y}_n)$ is an i.i.d. labeled dataset.
CAL: Label complexity proof

Start with $V_0 = \mathcal{H}$

For $t = 1, 2, \ldots$:

- Draw unlabeled point $x_t \sim \mathcal{D}$
- If $\exists h, h' \in V_{t-1}$ s.t. $h(x_t) \neq h'(x_t)$, query for label $y_t$
- Otherwise, create pseudo-label $\tilde{y}_t$
- Update $V_t = \{ h \in V_{t-1} : h(x_t) = y_t \text{ (or } \tilde{y}_t) \}$

Observation 1: We always have $h^*(x_t) = y_t$ (or $\tilde{y}_t$).

Observation 2: The (pseudo)-labeled dataset $(x_1, y_1/\tilde{y}_1), \ldots, (x_n, y_n/\tilde{y}_n)$ is an i.i.d. labeled dataset.

Conclusion: With probability $1 - \delta$, for every $t \geq 1$ and every $h \in V_t$,

$$\text{err}(h) \leq O\left(\frac{1}{t} \left( d \log t + \log \frac{t(t + 1)}{\delta} \right) \right) =: r_t.$$
CAL: Label complexity proof (continued)

With probability $1 - \delta$, for every $t \geq 1$ and every $h \in V_t$,

$$
\text{err}(h) \leq O \left( \frac{1}{t} \left( d \log t + \log \frac{t(t+1)}{\delta} \right) \right) =: r_t.
$$

At round $t$, CAL queries $x_t$ if and only if there is a hypothesis $h \in V_{t-1}$ such that $h(x_t) \neq h^*(x_t)$. 
CAL: Label complexity proof (continued)

With probability $1 - \delta$, for every $t \geq 1$ and every $h \in V_t$,

$$\text{err}(h) \leq O\left(\frac{1}{t} \left(d \log t + \log \frac{t(t+1)}{\delta}\right)\right) =: r_t.$$  

At round $t$, CAL queries $x_t$ if and only if there is a hypothesis $h \in V_{t-1}$ such that $h(x_t) \neq h^*(x_t)$.

$h \in V_{t-1}$ implies $h \in B(h^*, r_{t-1})$. $\implies$ query $x_t$ only if $x_t \in \text{DIS}(h^*, r_{t-1})$. 
CAL: Label complexity proof (continued)

\[ \mathbb{E}[\# \text{ of queries up to time } n] = \sum_{t=1}^{n} \mathbb{E}[\mathbb{E}[\mathbb{1}(\text{query } x_t) | V_{t-1}]] \leq \sum_{t=1}^{n} \mathbb{P}(x_t \in \text{DIS}(h^*, r_{t-1})) \leq \sum_{t=1}^{n} \theta \cdot r_{t-1} \leq O\left(\theta \left(d \log n + \log \frac{1}{\delta}\right) \log n\right) \]

Choosing \( n \) such that \( r_n \leq \epsilon \) makes the above \( \tilde{O}(d\theta \log \frac{1}{\epsilon}) \).
CAL: Label complexity proof (continued)

\[
\mathbb{E}[\# \text{ of queries up to time } n] = \sum_{t=1}^{n} \mathbb{E}[\mathbb{E}[\mathbb{1}(\text{query } x_t) \mid V_{t-1}]] \\
\leq \sum_{t=1}^{n} \Pr(x_t \in \text{DIS}(h^*, r^{t-1})) \\
\leq \sum_{t=1}^{n} \theta \cdot r_{t-1} \\
\leq O \left( \theta \left( d \log n + \log \frac{1}{\delta} \right) \log n \right)
\]

Choosing \( n \) such that \( r_n \leq \epsilon \) makes the above \( \tilde{O}(d\theta \log \frac{1}{\epsilon}) \).

Can turn from expectation bound to high probability bound using martingale deviation inequalities.
A partition of (some) active learning work

<table>
<thead>
<tr>
<th></th>
<th>Separable data</th>
<th>General (nonseparable) data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggressive</td>
<td>QBC [FSST97]</td>
<td>A^2 algorithm [BBL06, H07]</td>
</tr>
<tr>
<td></td>
<td>Splitting index [D05]</td>
<td>Reduction to supervised [DHM07]</td>
</tr>
<tr>
<td></td>
<td>GBS [D04, N09]</td>
<td>Importance weighted [BDL09]</td>
</tr>
<tr>
<td>Mellow</td>
<td>CAL [CAL94]</td>
<td>Confidence rated prediction [ZC14]</td>
</tr>
</tbody>
</table>
General (nonseparable) data setting

- Fixed binary hypothesis class $\mathcal{H}$
- Possibly not realizable: Query data point $x$ and receive
  \[ y \sim \Pr_{(X,Y)\sim\mathcal{D}}(Y \mid X = x) \]
- Target hypothesis: $h^* \in \mathcal{H}$ that minimizes error
  \[ \text{err}(h) = \Pr_{(X,Y)\sim\mathcal{D}}(h(X) \neq Y) \]
- Pool of unlabeled data drawn from $\mathcal{D}$ (essentially unlimited)
- **Goal**: learn low error hypothesis $h \in \mathcal{H}$
An agnostic mellow strategy: $A^2$ algorithm

**Issue:** Can no longer use version spaces.

**Solution:** Define effective ‘version space’ based on generalization bounds.
An agnostic mellow strategy: $A^2$ algorithm

**Issue:** Can no longer use version spaces.

**Solution:** Define effective ‘version space’ based on generalization bounds.

**Standard learning theory result:** For labeled dataset $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ drawn from distribution $D$,

$$|\text{err}_D(h) - \text{err}_S(h)| \leq \frac{1}{n} + \sqrt{\frac{\ln \frac{4}{\delta} + d \ln \frac{2en}{d}}{n}} =: G(n, \delta)$$

for every $h \in \mathcal{H}$ with probability $1 - \delta$. 
An agnostic mellow strategy: A² algorithm

**Issue:** Can no longer use version spaces.

**Solution:** Define effective ‘version space’ based on generalization bounds.

**Standard learning theory result:** For labeled dataset $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ drawn from distribution $\mathcal{D}$,

$$|\text{err}_\mathcal{D}(h) - \hat{\text{err}}_S(h)| \leq \frac{1}{n} + \sqrt{\frac{\ln \frac{4}{\delta} + d \ln \frac{2en}{d}}{n}} =: G(n, \delta)$$

for every $h \in \mathcal{H}$ with probability $1 - \delta$.

**Key idea:** With probability $1 - \delta$, any $h \in \mathcal{H}$ satisfying

$$\hat{\text{err}}_S(h) \geq \inf_{h' \in \mathcal{H}} \hat{\text{err}}_S(h) + 2G(n, \delta)$$

must have $\text{err}_\mathcal{D}(h) > \inf_{h' \in \mathcal{H}} \text{err}_\mathcal{D}(h)$. 
An agnostic mellow strategy: $A^2$ algorithm

Start with $V_0 = \mathcal{H}, S_0 = \emptyset$

For $t = 1, 2, \ldots, T$:

- Repeat until we have $n_t$ samples $S_t$:
  - Draw $x \sim \mathcal{D}$.
  - If $\exists h, h' \in V_{t-1}$ s.t. $h(x) \neq h'(x)$, query its label.
  - Otherwise, discard $x$.

- Set $V_t = \{ h \in V_{t-1} : \hat{\text{err}}_{S_t}(h) \leq \inf_{h' \in \mathcal{H}} \hat{\text{err}}_{S_t}(h') + 2G(n_t, \delta) \}$

$\hat{h} = \arg\min_{h \in V_T} \hat{\text{err}}_{S_T}(h)$
An agnostic mellow strategy: $A^2$ algorithm

Start with $V_0 = \mathcal{H}$, $S_0 = \emptyset$

For $t = 1, 2, \ldots, T$:

- Repeat until we have $n_t$ samples $S_t$:
  - Draw $x \sim \mathcal{D}$.
  - If $\exists h, h' \in V_{t-1}$ s.t. $h(x) \neq h'(x)$, query its label.
  - Otherwise, discard $x$.

- Set $V_t = \{ h \in V_{t-1} : \hat{\text{err}}_{S_t}(h) \leq \inf_{h' \in \mathcal{H}} \hat{\text{err}}_{S_t}(h') + 2G(n_t, \delta) \}$

$\hat{h} = \arg\min_{h \in V_T} \hat{\text{err}}_{S_T}(h)$

Theorem (Hanneke 2007)

Let $\nu = \inf_{h \in \mathcal{H}} \hat{\text{err}}_{S_t}(h)$. With probability $1 - \delta$, $\text{err}(\hat{h}) \leq \nu + \epsilon$ and

$$\# \text{ queries} \leq O \left( \theta^2 \left( 1 + \frac{\nu^2}{\epsilon^2} \right) \left( d \log \frac{1}{\epsilon} + \log \frac{1}{\delta} \right) \log \frac{1}{\epsilon} \right)$$
An agnostic mellow strategy: $A^2$ algorithm

Start with $V_0 = \mathcal{H}$, $S_0 = \emptyset$

For $t = 1, 2, \ldots, T$:

- Repeat until we have $n_t$ samples $S_t$:
  - Draw $x \sim \mathcal{D}$.
  - If $\exists h, h' \in V_{t-1}$ s.t. $h(x) \neq h'(x)$, query its label.
  - Otherwise, discard $x$.

- Set $V_t = \{h \in V_{t-1} : \hat{\text{err}}_{S_t}(h) \leq \inf_{h'\in \mathcal{H}} \hat{\text{err}}_{S_t}(h') + 2G(n_t, \delta)\}$

$\hat{h} = \arg\min_{h \in V_T} \hat{\text{err}}_{S_T}(h)$

Theorem (Hanneke 2007)

Let $\nu = \inf_{h \in \mathcal{H}} \hat{\text{err}}_{S_t}(h)$. With probability $1 - \delta$, $\text{err}(\hat{h}) \leq \nu + \epsilon$ and

$$\# \text{ queries} \leq O \left( \theta^2 \left( 1 + \frac{\nu^2}{\epsilon^2} \right) \left( d \log \frac{1}{\epsilon} + \log \frac{1}{\delta} \right) \log \frac{1}{\epsilon} \right)$$
An agnostic mellow strategy: $A^2$ algorithm

**Theorem (Beygelzimer et al. 2007)**

For any $\nu, \epsilon > 0$ such that $2\epsilon \leq \nu \leq 1/4$, any input space, and any hypothesis class $\mathcal{H}$ of VC-dimension $d$, there is a distribution such that

(a) the best achievable error rate of a hypothesis in $\mathcal{H}$ is $\nu$ and

(b) any active learner seeking a hypothesis with error $\nu + \epsilon$ must make $\frac{d\nu^2}{\epsilon^2}$ queries to succeed with probability at least $1/2$. 
An agnostic mellow strategy: $A^2$ algorithm

**Theorem (Beygelzimer et al. 2007)**

For any $\nu, \epsilon > 0$ such that $2\epsilon \leq \nu \leq 1/4$, any input space, and any hypothesis class $\mathcal{H}$ of VC-dimension $d$, there is a distribution such that

(a) the best achievable error rate of a hypothesis in $\mathcal{H}$ is $\nu$ and

(b) any active learner seeking a hypothesis with error $\nu + \epsilon$ must make $rac{d\nu^2}{\epsilon^2}$ queries to succeed with probability at least $1/2$.

...BUT the distribution from Beygelzimer et al. is not very ‘natural.’
When are these algorithms efficient?

Computational challenges:

- **CAL/A²**: Maintaining a version space can be computationally challenging...
  - Don’t always need to do so explicitly.
Efficient CAL

To run CAL, we need to be able to determine if $x$ falls in the disagreement region of $V$: 

$$\exists h, h' \in V \text{ s.t. } h(x) \neq h'(x)$$

**Assumption**: We have an ERM oracle $\text{learn}((x_1, y_1), \ldots, (x_n, y_n))$: 
- Returns $h \in \mathcal{H}$ s.t. $h(x_i) = y_i$ for $i = 1, \ldots, n$ if it exists 
- Returns $\bot$ otherwise
Efficient CAL

To run CAL, we need to be able to determine if \( x \) falls in the disagreement region of \( V \):

\[
\exists h, h' \in V \text{ s.t. } h(x) \neq h'(x)
\]

**Assumption:** We have an ERM oracle \( \text{learn}((x_1, y_1), \ldots, (x_n, y_n)) \):
- Returns \( h \in \mathcal{H} \) s.t. \( h(x_i) = y_i \) for \( i = 1, \ldots, n \) if it exists
- Returns \( \bot \) otherwise

To run CAL at round \( t \):
- Have data \( (x_1, y_1), \ldots, (x_{t-1}, y_{t-1}) \).
- Query \( x \) if

\[
\text{learn}((x_1, y_1), \ldots, (x_{t-1}, y_{t-1}), (x, +)) \neq \bot
\]
\[
\text{learn}((x_1, y_1), \ldots, (x_{t-1}, y_{t-1}), (x, -)) \neq \bot
\]
Active research directions

- Aggressive strategies for general data
- Active learning without a fixed hypothesis class
  - Nested hypothesis classes
- Circumventing lower bounds
  - Tsybakov noise, Massart noise
- Specialized algorithms for special cases
  - Linear functions, neural nets, ...
## A partition of (some) active learning work

<table>
<thead>
<tr>
<th></th>
<th>Separable data</th>
<th>General (nonseparable) data</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Aggressive</strong></td>
<td>QBC [FSST97]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Splitting index [D05]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GBS [D04, N09]</td>
<td></td>
</tr>
<tr>
<td><strong>Mellow</strong></td>
<td>CAL [CAL94]</td>
<td>A(^2) algorithm [BBL06, H07]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reduction to supervised [DHM07]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Importance weighted [BDL09]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Confidence rated prediction [ZC14]</td>
</tr>
</tbody>
</table>
Mellow v.s. aggressive

Mellow active learning strategies:
- Query any data point whose label cannot be confidently inferred.

Aggressive active learning strategies:
- Query **informative** data points.
Generalized binary search

Introduce a prior probability measure $\pi$ over $\mathcal{H}$.
- Assigns preferences over hypotheses.

Examples:
- **Finite classes**: Uniform distribution over $\mathcal{H}$.
- **Homogeneous linear separators**: Log-concave distributions, e.g. normal distribution.
- **General classes**: $e^{-R(h)}$ where $R(\cdot)$ is some regularizer.
Generalized binary search

Introduce a prior probability measure $\pi$ over $\mathcal{H}$.
- Assigns preferences over hypotheses.

Generalized binary search criterion:
- Query data point that is guaranteed to lead to most probability mass of version space being eliminated:

$$\arg\min_x \max \{ \pi(V_x^+), \pi(V_x^-) \}$$

where $V_x^+ = \{ h \in V : h(x) = + \}$ and $V_x^- = V \setminus V_x^+$. 
Generalized binary search: A change in objective

Given a finite pool of unlabeled data, a deterministic active learning strategy induces a decision tree $T$ whose leaves are the elements of $\mathcal{H}$. 

![Decision Tree Diagram]
Generalized binary search: A change in objective

Given a finite pool of unlabeled data, a deterministic active learning strategy induces a decision tree $T$ whose leaves are the elements of $\mathcal{H}$.

Possible objectives:

- Worst case cost: $\max_{h \in \mathcal{H}}$ length of path in $T$ to get to $h$
- Average case cost: $\sum_{h \in \mathcal{H}} (\text{length of path in } T \text{ to get to } h) \cdot \pi(h)$
Generalized binary search: A change in objective

Given a finite pool of unlabeled data, a deterministic active learning strategy induces a decision tree $T$ whose leaves are the elements of $\mathcal{H}$.

Possible objectives:

- **Worst case cost**: $\max_{h \in \mathcal{H}}$ length of path in $T$ to get to $h$
- **Average case cost**: $\sum_{h \in \mathcal{H}} (\text{length of path in } T \text{ to get to } h) \cdot \pi(h)$
Extra time  
GBS

Generalized binary search: Theorem

**Theorem (Dasgupta 2004)**

Let $\pi$ be any prior over $\mathcal{H}$. Suppose the optimal search tree has average cost $Q^*$. Then the average cost of the GBS search tree is at most

$$4Q^* \ln \frac{1}{\min_h \pi(h)}.$$
Generalized binary search: Theorem

**Theorem (Dasgupta 2004)**

Let \( \pi \) be any prior over \( \mathcal{H} \). Suppose the optimal search tree has average cost \( Q^* \). Then the average cost of the GBS search tree is at most

\[
4Q^* \ln \frac{1}{\min_h \pi(h)}.
\]

If instead only query \( \alpha \)-approximately greedy points, i.e. points \( x \) which satisfy

\[
\pi(V_x^+) \pi(V_x^-) \geq \frac{1}{\alpha} \max_{x^*} \pi(V_{x^+}) \pi(V_{x^-})
\]

then cost becomes \( O \left( \alpha Q^* \ln \frac{1}{\min_h \pi(h)} \right) \) (Golovin and Krause 2010).
Efficient GBS

To run GBS, we need to be able to approximately determine the split \( \pi(V_x^+), \pi(V_x^-) \).

**Assumption:** We have a sampling oracle \( \text{sample}(V) \):

- Returns a sample from \( \pi|_V \) (\( \pi \) conditioned on \( V \))
Efficient GBS

To run GBS, we need to be able to approximately determine the split $\pi(V_x^+), \pi(V_x^-)$

**Assumption:** We have a sampling oracle \text{sample}(V):

- Returns a sample from $\pi|_V$ ($\pi$ conditioned on $V$)

To run GBS at round $t$:

- Have version space $V$.
- Sample hypotheses $h_1, \ldots, h_n$ using \text{sample}(V).
- Query $x$ that minimizes

$$\frac{1}{n} \max \left\{ \sum_{i=1}^{n} \mathbb{1}[h_i(x) = +], \sum_{i=1}^{n} \mathbb{1}[h_i(x) = -] \right\} \approx \max \{ \pi(V_x^+), \pi(V_x^-) \}$$