Active Learning

sample /sense

analyze and infer

adapt sensing or experiment
Training examples come in pairs, feature X and label Y.

**Goal**: Design a rule for predicting Y given X
Training examples come in pairs, feature X and label Y.

**Goal:** Design a rule for predicting Y given X
Training examples come in pairs, feature X and label Y.

**Goal**: Design a rule for predicting Y given X.
Training examples come in pairs, feature X and label Y.

**Goal:** Design a rule for predicting Y given X
Training examples come in pairs, feature X and label Y.

**Goal**: Design a rule for predicting Y given X
Machine Learning (Passive)

Raw unlabeled data

\[ X_1, X_2, X_3, \ldots \]

passive learner

expert/oracle analyzes/experiments to determine labels
Machine Learning (Passive)

Raw unlabeled data

$X_1, X_2, X_3, \ldots$

passive learner

expert/oracle analyzes/experiments to determine labels
Machine Learning (Passive)

- Raw unlabeled data: $X_1, X_2, X_3, \ldots$
- Labeled data: $(X_1, Y_1), (X_2, Y_2), (X_3, Y_3), \ldots$

- passive learner
- expert/oracle analyzes/experiments to determine labels
Machine Learning (Passive)

Raw unlabeled data

$X_1, X_2, X_3, \ldots$

Labeled data

$(X_1, Y_1), (X_2, Y_2), (X_3, Y_3), \ldots$

passive learner

automatic classifier

expert/oracle
analyzes/experiments
to determine labels
Active Learning

Raw unlabeled data

$x_1, x_2, x_3, \ldots$

active learner

expert/oracle
analyzes/experiments
to determine labels
Active Learning

Raw unlabeled data

$X_1, X_2, X_3, \ldots$

active learner

expert/oracle
analyzes/experiments
to determine labels
Active Learning

Raw unlabeled data

$X_1, X_2, X_3, \ldots$

Learner requests labels for **selected** data

$(X_1, ?)$

**active learner**

**expert/oracle** analyzes/experiments to determine labels
Active Learning

Raw unlabeled data

\[ X_1, X_2, X_3, \ldots \]

Learner requests labels for **selected** data

\[ (X_1, \text{?}) \]

\[ (X_1, Y_1) \]

**active learner**

**expert/oracle** analyzes/experiments to determine labels

Friday, May 20, 2011
Active Learning

Raw unlabeled data

\[ X_1, X_2, X_3, \ldots \]

Learner requests labels for **selected** data

\[ (X_1, ?) \]

\[ (X_1, Y_1) \]

\[ (X_3, ?) \]

**active learner**

**expert/oracle**

analyzes/experiments to determine labels

Friday, May 20, 2011
Active Learning

Raw unlabeled data

\[ X_1, X_2, X_3, \ldots \]

Learner requests labels for selected data

- \((X_1, ?)\)
- \((X_1, Y_1)\)
- \((X_3, ?)\)
- \((X_3, Y_3)\)

active learner

expert/oracle analyzes/experiments to determine labels
Active Learning

Raw unlabeled data

\[ X_1, X_2, X_3, \ldots \]

Learner requests labels for **selected** data

- \((X_1, \, ?)\)
- \((X_1, Y_1)\)
- \((X_3, \, ?)\)
- \((X_3, Y_3)\)

**active learner**

**expert/oracle** analyzes/experiments to determine labels

**automatic classifier**

Friday, May 20, 2011
Applications of Active Learning

Hand-written character recognition

Document classification

Systems biology

Sensor networks

In many applications, obtaining labels or running experiments is costly!
A Stylized Environmental Sensing Task

Where is it shady vs. sunny?
A Stylized Environmental Sensing Task

Where is it shady vs. sunny?
A Stylized Environmental Sensing Task

Where is it shady vs. sunny?

Suppose we have N wireless sensors. Do we need to query them all?
Classic Binary Search

Where is it shady vs. sunny?
Classic Binary Search

Where is it shady vs. sunny?
Classic Binary Search

**adaptive sensing**: sequentially select points for labeling
**Classic Binary Search**

**adaptive sensing**: sequentially select points for labeling

\[ \mathcal{X} = [0, 1] \]

\[ \mathcal{H} = \{ \text{thresholds at } \frac{1}{n}, \frac{2}{n}, \ldots, 1 \} \]
Classic Binary Search

**adaptive sensing**: sequentially select points for labeling

\[ \mathcal{X} = [0, 1] \]

\[ \mathcal{H} = \{ \text{thresholds at } \frac{1}{n}, \frac{2}{n}, \ldots, 1 \} \]

\[ \frac{1}{3} = 0… \]
**Classic Binary Search**

- **Adaptive Sensing**: sequentially select points for labeling

\[ \mathcal{X} = [0, 1] \]

\[ \mathcal{H} = \{ \text{thresholds at } \frac{1}{n}, \frac{2}{n}, \ldots, 1 \} \]

\[ 1/3 = 01... \]
Classic Binary Search

adaptive sensing: sequentially select points for labeling

\[ \mathcal{X} = [0, 1] \]
\[ \mathcal{H} = \{ \text{thresholds at } \frac{1}{n}, \frac{2}{n}, \ldots, 1 \} \]

\[ \frac{1}{3} = 010\ldots \]
Classic Binary Search

**adaptive sensing**: sequentially select points for labeling

\[ \mathcal{X} = [0, 1] \]

\[ \mathcal{H} = \{ \text{thresholds at } \frac{1}{n}, \frac{2}{n}, \ldots, 1 \} \]

\[ \frac{1}{3} = 0.10101\ldots \]
Classic Binary Search

**adaptive sensing**: sequentially select points for labeling

\[ X = [0, 1] \]

\[ \mathcal{H} = \{ \text{thresholds at } \frac{1}{n}, \frac{2}{n}, \ldots, 1 \} \]

\[ \frac{1}{3} = 0101\ldots \text{ requires } \log_2 N \text{ queries} \]
Classic Binary Search

**adaptive sensing**: sequentially select points for labeling

\[ X = [0, 1] \]

\[ \mathcal{H} = \{ \text{thresholds at } \frac{1}{n}, \frac{2}{n}, \ldots, 1 \} \]

\[ \frac{1}{3} = 0101\ldots \text{ requires } \log_2 N \text{ queries} \]

**non-adaptive**: query points uniformly (possibly random)

Friday, May 20, 2011
**Classic Binary Search**

**adaptive sensing**: sequentially select points for labeling

\[ X = [0, 1] \]
\[ H = \{ \text{thresholds at } \frac{1}{n}, \frac{2}{n}, \ldots, 1 \} \]

\[ 1/3 = 0101 \ldots \]
requires \( \log_2 N \) queries

**non-adaptive**: query points uniformly (possibly random)
Classic Binary Search

**adaptive sensing**: sequentially select points for labeling

\[ \mathcal{X} = [0, 1] \]
\[ \mathcal{H} = \{ \text{thresholds at } \frac{1}{n}, \frac{2}{n}, \ldots, 1 \} \]

\[ \frac{1}{3} = 0101\ldots \]
requires \( \log_2 N \) queries

**non-adaptive**: query points uniformly (possibly random)

\[ \frac{1}{3} = 0101\ldots \]
requires \( O(N) \) queries
Classic Binary Search

adaptive sensing: sequentially select points for labeling

\[ \mathcal{X} = [0, 1] \]

\[ \mathcal{H} = \{ \text{thresholds at } \frac{1}{n}, \frac{2}{n}, \ldots, 1 \} \]

1/3 = 0101... requires \( \log_2 N \) queries

non-adaptive: query points uniformly (possibly random)

requires \( O(N) \) queries

adaptive sensing is dramatically more efficient
Environmental Sensing

Lake Wingra, Madison WI

acoustic doppler sensing of water current in Lake Wingra

water current velocity map (darker = high velocity)

classification into high- and low-velocity regions

Chin Wu, Civil & Environmental Engr.
http://limnology.wisc.edu/

Non-adaptive Survey: 48 hrs
Adaptive Survey: 14 hrs

Outline of Part 3
Noisy Binary Search: What if the expert/oracle responses are not completely reliable?
Outline of Part 3

**Noisy Binary Search**: What if the expert/oracle responses are not completely reliable?

**Minimax Analysis of Active Learning**: What are the fundamental capabilities and limits of active learning?
Outline of Part 3

**Noisy Binary Search:** What if the expert/oracle responses are not completely reliable?

**Minimax Analysis of Active Learning:** What are the fundamental capabilities and limits of active learning?

**Generalized Binary Search:** Can binary search be generalized in order to learn more complex decision rules?
Outline of Part 3

**Noisy Binary Search**: What if the expert/oracle responses are not completely reliable?

**Minimax Analysis of Active Learning**: What are the fundamental capabilities and limits of active learning?

**Generalized Binary Search**: Can binary search be generalized in order to learn more complex decision rules?

**Unsupervised Active Learning**: Can active learning help in unsupervised learning problems such as clustering?
Binary Search and Noise

At what income level is a person more likely to be Republican vs. Democrat?

Probability of voting for Obama

income
Binary Search and Noise

At what income level is a person more likely to be Republican vs. Democrat?

Probability of voting for Obama

Friday, May 20, 2011
At what income level is a person more likely to be Republican vs. Democrat?
Binary Search and Noise

At what income level is a person more likely to be Republican vs. Democrat?

\[ \theta^* = \$250K \]

\[ \text{probably Democrat} \quad \text{probably Republican} \]
Bounded and Unbounded Noise

\begin{figure}
\centering
\includegraphics[width=\textwidth]{bounded_noise}
\end{figure}
Bounded and Unbounded Noise

$P(Y = 1 | X = x)$

Diagram showing the probability $P(Y = 1 | X = x)$ on a graph with $y$-axis and $x$-axis.
Bounded and Unbounded Noise

\[ P(Y = 1 | X = x) \]

more probably 1  \quad more probably 0
Bounded and Unbounded Noise

"bounded noise": strictly more/less probably 1 at all locations
Bounded and Unbounded Noise

"unbounded noise": like the toss of a fair coin at threshold

\[ P(Y = 1|X = x) \]
Horstein’s Multiplicative Weighting Method
Horstein’s Multiplicative Weighting Method
Horstein’s Multiplicative Weighting Method

![Diagram](image)

- $h^*$
- $b$
- $p_0(\theta)$
Horstein’s Multiplicative Weighting Method

$h^*$

$p_0(\theta)$
Horstein’s Multiplicative Weighting Method

Update ‘posterior’ density based on noise bound $b$
Horstein’s Multiplicative Weighting Method

Sequentially take samples at posterior median

$p_1(\theta)$

$h^*$
Horstein’s Multiplicative Weighting Method

sequentially take samples at posterior median
Horstein’s Multiplicative Weighting Method

Sequentially take samples at posterior median

Friday, May 20, 2011
Channel Coding with Noiseless Feedback

sender

1,0,1,1,0,1...

...?

receiver

1,0,0,1,0,1.

Friday, May 20, 2011
Channel Coding with Noiseless Feedback

sender

receiver

noise bound
= BSC crossover prob

threshold location
= n bit message

1,0,1,1,0,1...

1,0,0,1,0,1...

1,0,0,1,0,1...

Friday, May 20, 2011
Channel Coding with Noiseless Feedback

sender

receiver

noiseless feedback

noise bound
= BSC crossover prob

threshold location
= n bit message

1,0,1,1,0,1...

1,0,0,1,0,1...

1,0,0,1,0,1...

1,0,0,1,0,1...

1,0,0,1,0,1...

Friday, May 20, 2011
Channel Coding with Noiseless Feedback

Both sender and receiver implement Horstein's algorithm.

Sender deduces which binary symbol to send next in order to yield the greatest possible reduction in the receiver's uncertainty about the n-bit message.

- Noise bound = BSC crossover prob
- Threshold location = n bit message

Friday, May 20, 2011
Active Learning in Unbounded Noise

Classical Binary Search

Noisy Binary Search
Active Learning in Unbounded Noise

Classic Binary Search

Noisy Binary Search

unbounded noise

Friday, May 20, 2011
Active Learning in Unbounded Noise

Classic Binary Search

Noisy Binary Search

No strong cue about the location of the boundary

unbounded noise
Active Learning in Unbounded Noise

Classic Binary Search

Noisy Binary Search

unbounded noise

No strong cue about the location of the boundary

Rui Castro (Columbia): “How much does active learning help in this case?”
Unbounded Noise Effects

Near $\frac{1}{2}$-level,

\[ c|x - \theta^*|^{\kappa-1} \leq |\eta(x) - 1/2| \leq C|x - \theta^*|^{\kappa-1}, \quad \kappa \geq 1 \]

\[ \eta(x) := P(Y = 1 | X = x) \]

similar conditions are commonly employed in nonparametric statistics, Tsybakov (2004)
Unbounded Noise Effects

Near $\frac{1}{2}$-level, $c|x - \theta^*|^{\kappa-1} \leq |\eta(x) - 1/2| \leq C|x - \theta^*|^{\kappa-1}$, $\kappa \geq 1$

$\eta(x) := \Pr(Y = 1 | X = x)$

$\kappa = 2$

similar conditions are commonly employed in nonparametric statistics, Tsybakov (2004)
Unbounded Noise Effects

Near $\frac{1}{2}$-level,

$$c|x - \theta^*|^{\kappa-1} \leq |\eta(x) - 1/2| \leq C|x - \theta^*|^{\kappa-1}, \quad \kappa \geq 1$$

similar conditions are commonly employed in nonparametric statistics, Tsybakov (2004)
Unbounded Noise Effects

Near $\frac{1}{2}$-level, \[ c|x - \theta^*|^\kappa - 1 \leq |\eta(x) - 1/2| \leq C|x - \theta^*|^\kappa - 1, \quad \kappa \geq 1 \]

\[ \eta(x) := P(Y = 1|X = x) \]

$\kappa = 2$

$\kappa \rightarrow 1$

$\kappa \rightarrow \infty$

similar conditions are commonly employed in nonparametric statistics, Tsybakov (2004)
Horstein’s Algorithm in Unbounded Noise
Horstein’s Algorithm in Unbounded Noise

Consider discrete set of thresholds and discretized version of $P(Y=1|X=x)$
Consider discrete set of thresholds and discretized version of $P(Y=1|X=x)$.

If $1/2$ level is not aligned with discrete thresholds, then noise of discretized problem is bounded, but depends on resolution of discretization $t$ and the behavior of $P(Y=1|X=x)$ at the $1/2$ level.
Horstein’s Algorithm in Unbounded Noise

Consider discrete set of thresholds and discretized version of $P(Y=1|X=x)$

If $1/2$ level is not aligned with discrete thresholds, then noise of discretized problem is bounded, but depends on resolution of discretization $\delta$ and the behavior of $P(Y=1|X=x)$ at the $1/2$ level

$$P[h_n(X) \neq Y] - P[h^*(X) \neq Y] \leq t^\kappa + t^{-1} \exp(-nc^2t^{2\kappa-2})$$
Horstein’s Algorithm in Unbounded Noise

Consider discrete set of thresholds and discretized version of $P(Y=1|X=x)$

If $\frac{1}{2}$ level is not aligned with discrete thresholds, then noise of discretized problem is bounded, but depends on resolution of discretization $t$ and the behavior of $P(Y=1|X=x)$ at the $\frac{1}{2}$ level

$$P[h_n(X) \neq Y] - P[h^*(X) \neq Y] \leq t^{\kappa} + t^{-1} \exp(-nc^2t^{2\kappa-2})$$

$$= O\left(\left[\frac{\log n}{n}\right]^{\frac{\kappa}{2\kappa-2}}\right)$$
Rates of Convergence

passive: \( n^{-2/3} \)

active: \( n^{-1} \)

passive: \( \kappa \rightarrow 1 \rightarrow n^{-1} \)

active: \( e^{-cn} \)

passive: \( \kappa \rightarrow \infty \rightarrow n^{-1/2} \)

active: \( n^{-1/2} \)
Are you a good active learner?

Castro, Kalish, Nowak, Qian, Rogers & Zhu (NIPS 2008)

Investigate human active learning in task analogous to 1-d threshold problem
Are you a good active learner?

Castro, Kalish, Nowak, Qian, Rogers & Zhu (NIPS 2008)

Investigate human active learning in task analogous to 1-d threshold problem

alien eggs
Are you a good active learner?

Castro, Kalish, Nowak, Qian, Rogers & Zhu (NIPS 2008)

Investigate human active learning in task analogous to 1-d threshold problem

alien eggs

\[ \theta \]

\begin{align*}
0 & & 0.125 & & 0.25 & & 0.375 & & 0.5 & & 0.625 & & 0.75 & & 0.875 & & 1 \\
\end{align*}
Are you a good active learner?

Castro, Kalish, Nowak, Qian, Rogers & Zhu (NIPS 2008)

Investigate human active learning in task analogous to 1-d threshold problem

alien eggs

more probably birds
Are you a good active learner?

Investigate human active learning in task analogous to 1-d threshold problem

Castro, Kalish, Nowak, Qian, Rogers & Zhu (NIPS 2008)
Are you a good active learner?

Castro, Kalish, Nowak, Qian, Rogers & Zhu (NIPS 2008)

Investigate human active learning in task analogous to 1-d threshold problem

Subjects observe random egg hatchings (passive learning) or they can select eggs to hatch (active learning).

They are asked to determine the egg shape where snakes become more probable than birds.
Are you a good active learner?

Castro, Kalish, Nowak, Qian, Rogers & Zhu (NIPS 2008)

Investigate human active learning in task analogous to 1-d threshold problem

Results: Human learning rates agree with theory, $1/n$ in passive mode and $\exp(-cn)$ in active mode.
Learning Multidimensional Threshold Functions

\[ \eta(x) \]

\[ d = 1 \]
Learning Multidimensional Threshold Functions

\[ \eta(x) \]

\[ d = 1 \]

\[ d > 1 \]
Learning Multidimensional Threshold Functions

\[ y \]

\[ \eta(x) \]

\[ d = 1 \]

\[ d > 1 \]
Learning Rates for Multidimensional Thresholds

sharp transition \( \kappa = 1 \)

smooth transition \( \kappa > 1 \)
Learning Rates for Multidimensional Thresholds

sharp transition \( \kappa = 1 \)

Hölder-\( \alpha \) smooth decision boundary

smooth transition \( \kappa > 1 \)
Learning Rates for Multidimensional Thresholds

Active Learning: Theorem (R. Castro and RN ’07)

\[
\left( \frac{1}{n} \right)^{\frac{\kappa}{2\kappa + \rho - 2}} \leq \inf_{h_n, S_n} \sup_{P_{XY} \in BF(\alpha, \kappa)} \mathcal{E}(h_n) \leq \left( \frac{\log n}{n} \right)^{\frac{\kappa}{2\kappa + \rho - 2}}
\]

\((\rho = (d - 1)/\alpha)\)
Learning Rates for Multidimensional Thresholds

Active Learning: Theorem (R. Castro and RN ’07)

\[
\left( \frac{1}{n} \right)^{\frac{\kappa}{2\kappa + \rho - 2}} \leq \inf_{h_n, S_n} \sup_{P_{XY} \in BF(\alpha, \kappa)} \mathcal{E}(h_n) \leq \left( \frac{\log n}{n} \right)^{\frac{\kappa}{2\kappa + \rho - 2}}
\]

\( \rho = (d - 1)/\alpha \)

Compare with passive learning

\[
\inf_{h_n} \sup_{P_{XY} \in BF(\alpha, \kappa)} \mathcal{E}(h_n) \asymp \left( \frac{1}{n} \right)^{\frac{\kappa}{2\kappa + \rho - 1}} \quad \text{as } \rho \to 0
\]

\text{and } \kappa \to 1
Learning Rates for Multidimensional Thresholds

**Active Learning: Theorem (R. Castro and RN ’07)**

\[
\left(\frac{1}{n}\right)^{\frac{\kappa}{2\kappa + \rho - 2}} \leq \inf_{h_n, S_n} \sup_{P_{XY} \in BF(\alpha, \kappa)} \mathcal{E}(h_n) \leq \left(\frac{\log n}{n}\right)^{\frac{\kappa}{2\kappa + \rho - 2}}
\]

\((\rho = (d - 1)/\alpha)\)

Compare with passive learning

\[
\inf_{h_n} \sup_{P_{XY} \in BF(\alpha, \kappa)} \mathcal{E}(h_n) \asymp \left(\frac{1}{n}\right)^{\frac{\kappa}{2\kappa + \rho - 1}} \quad \text{as } \rho \to 0
\]

and \(\kappa \to 1\)
Learning Rates for Multidimensional Thresholds

**Main idea**: reduce multidimensional problem to a sequence of 1-dim problems

Active Learning: Theorem (R. Castro and RN ’07)

\[
\left( \frac{1}{n} \right)^{\frac{\kappa}{2\kappa + \rho - 2}} \leq \inf_{h_n, S_n} \sup_{P_{XY} \in BF(\alpha, \kappa)} \mathcal{E}(h_n) \leq \left( \frac{\log n}{n} \right)^{\frac{\kappa}{2\kappa + \rho - 2}}
\]

\[(\rho = (d - 1)/\alpha)\]

Compare with passive learning

\[
\inf_{h_n} \sup_{P_{XY} \in BF(\alpha, \kappa)} \mathcal{E}(h_n) \asymp \left( \frac{1}{n} \right)^{\frac{\kappa}{2\kappa + \rho - 1}}
\]

as \(\rho \to 0\) and \(\kappa \to 1\)
Algorithms for Active Learning

\[ \mathcal{X} := \text{domain or query space} \]
\[ \mathcal{Y} := \{-1, +1\} \]
\[ \mathcal{H} := \text{hypothesis space} \quad \forall h \in H, \ h : \mathcal{X} \to \mathcal{Y} \]
Algorithms for Active Learning

\[ \mathcal{X} := \text{domain or query space} \]

\[ \mathcal{Y} := \{-1, +1\} \]

\[ \mathcal{H} := \text{hypothesis space} \quad \forall h \in H, \ h : \mathcal{X} \to \mathcal{Y} \]
Algorithms for Active Learning

\( \mathcal{X} := \text{domain or query space} \)

\( \mathcal{Y} := \{ -1, +1 \} \)

\( \mathcal{H} := \text{hypothesis space} \quad \forall h \in H, \ h : \mathcal{X} \rightarrow \mathcal{Y} \)

Question: How many queries are required to determine \( h^* \)?
Algorithms for Active Learning

\[ \mathcal{X} := \text{domain or query space} \]
\[ \mathcal{Y} := \{-1, +1\} \]
\[ \mathcal{H} := \text{hypothesis space} \quad \forall h \in H, \ h : \mathcal{X} \to \mathcal{Y} \]

Question: How many queries are required to determine \( h^* \)?

If \( \mathcal{H} \) is finite with \( N := |\mathcal{H}| \), then identification of \( h^* \) requires at least \( \log_2 N \) bits/queries.
initialize: \( n = 0, \mathcal{H}_0 = \mathcal{H} \)

while \( |\mathcal{H}_n| > 1 \)
initialize: $n = 0$, $\mathcal{H}_0 = \mathcal{H}$

while $|\mathcal{H}_n| > 1$

1) Select $x_n = \arg \min_{x \in \mathcal{X}} |\sum_{h \in \mathcal{H}_n} h(x)|$. 

**Generalized Binary Search (aka Splitting Algorithm)**

$h$ hypothesis space

$q$ query space

Friday, May 20, 2011
Generalized Binary Search (aka Splitting Algorithm)

initialize: $n = 0$, $\mathcal{H}_0 = \mathcal{H}$

while $|\mathcal{H}_n| > 1$

1) Select $x_n = \arg\min_{x \in \mathcal{X}} |\sum_{h \in \mathcal{H}_n} h(x)|$.

Selects a query for which disagreement among hypotheses is maximal
Generalized Binary Search (aka Splitting Algorithm)

initialize: \( n = 0, \mathcal{H}_0 = \mathcal{H} \)

while \( |\mathcal{H}_n| > 1 \)

1) Select \( x_n = \arg \min_{x \in \mathcal{X}} |\sum_{h \in \mathcal{H}_n} h(x)| \).

2) Query with \( x_n \) to obtain response \( y_n = h^*(x_n) \).
Generalized Binary Search (aka Splitting Algorithm)

initialize: \( n = 0, \mathcal{H}_0 = \mathcal{H} \)

while \( |\mathcal{H}_n| > 1 \)

1) Select \( x_n = \arg \min_{x \in \mathcal{X}} |\sum_{h \in \mathcal{H}_n} h(x)|. \)

2) Query with \( x_n \) to obtain response \( y_n = h^*(x_n) \).

3) Set \( \mathcal{H}_{n+1} = \{ h \in \mathcal{H}_n : h(x_n) = y_n \} \), \( n = n + 1 \).
Bisection in Higher Dimensions

Consider the decision boundaries of a collection of classifiers in a multidimensional feature space
Consider the decision boundaries of a collection of classifiers in a multidimensional feature space.

Unlike the situation in 1-d, there is no natural ordering of classifiers/boundaries and therefore no immediately obvious approach to binary search.
Bisection in Higher Dimensions

Consider the decision boundaries of a collection of classifiers in a multidimensional feature space.

Unlike the situation in 1-d, there is no natural ordering of classifiers/boundaries and therefore no immediately obvious approach to binary search.

... but if we are given one positive and negative example, then we perform a bisection along the path between these points.
Consider the decision boundaries of a collection of classifiers in a multidimensional feature space. Unlike the situation in 1-d, there is no natural ordering of classifiers/boundaries and therefore no immediately obvious approach to binary search. But if we are given one positive and negative example, then we perform a bisection along the path between these points.
Bisection in Higher Dimensions

Consider the decision boundaries of a collection of classifiers in a multidimensional feature space.

Unlike the situation in 1-d, there is no natural ordering of classifiers/boundaries and therefore no immediately obvious approach to binary search.

... but if we are given one positive and negative example, then we perform a bisection along the path between these points.
Bisection in Higher Dimensions

Consider the decision boundaries of a collection of classifiers in a multidimensional feature space.

Unlike the situation in 1-d, there is no natural ordering of classifiers/boundaries and therefore no immediately obvious approach to binary search.

... but if we are given one positive and negative example, then we perform a bisection along the path between these points.

Bisecting paths of this sort exist under a mild and verifiable property we call the "neighborly condition".
Learning Halfspaces in $\mathbb{R}^d$
Learning Halfspaces in $\mathbb{R}^d$

\[
\begin{pmatrix}
\begin{array}{cccccccccccc}
A_1 & A_2 & A_3 & A_4 & A_5 & A_6 & A_7 & A_8 & A_9 & A_{10} & A_{11} \\
+ & - & - & - & - & + & + & + & + & + & + \\
+ & + & - & - & - & + & + & + & + & + & - \\
+ & + & + & - & - & - & + & - & - & + & + \\
+ & + & + & + & - & - & - & - & + & + & + \\
\end{array}
\end{pmatrix}
\]

Diagram with four halfspaces $h_1, h_2, h_3, h_4$ and their corresponding regions.
Learning Halfspaces in $\mathbb{R}^d$

bisecting queries

\[
\begin{bmatrix}
A_1 & A_2 & A_3 & A_4 & A_5 & A_6 & A_7 & A_8 & A_9 & A_{10} & A_{11}
\end{bmatrix}
\]

\[
\begin{bmatrix}
h_1 \\
h_2 \\
h_3 \\
h_4
\end{bmatrix}
\begin{bmatrix}
+ & - & - & - & - & + & + & + & + & + & + \\
+ & + & + & - & - & - & + & - & + & + & + \\
\end{bmatrix}
\]
Learning Halfspaces in $\mathbb{R}^d$

queries generate only $O(N^d)$ of the possible $2^N$ binary patterns!
Learning Halfspaces in $\mathbb{R}^d$

Can GBS find near-bisecting queries in general?

queries generate only $O(N^d)$ of the possible $2^N$ binary patterns!

bisecting queries
Learning Halfspaces in $\mathbb{R}^d$

Can GBS find near-bisecting queries in general?

If $\mathcal{H}$ is a collection of $N$ halfspaces on $\mathcal{X} = \mathbb{R}^d$, then GBS terminates with the correct halfspace after $O(\log N)$ queries.

Queries generate only $O(N^d)$ of the possible $2^N$ binary patterns!
Example

Suppose we have a sensor network observing a binary activation pattern with a linear boundary. How many sensors must be queried to determine the pattern?

100 sensors, 9900 possible linear boundaries
Example

Suppose we have a sensor network observing a binary activation pattern with a linear boundary. How many sensors must be queried to determine the pattern?

100 sensors, 9900 possible linear boundaries
Example

Suppose we have a sensor network observing a binary activation pattern with a linear boundary. How many sensors must be queried to determine the pattern?

Correct boundary determined after querying 12 sensors
“Is the person wearing a hat?”
“Is the person wearing a hat?”

“Does the person have blue eyes?”
“Is the person wearing a hat?”

“Does the person have blue eyes?”

GBS is quite effective if responses are reliable
“Is the person wearing a hat?”

“Does the person have blue eyes?”

GBS is quite effective if responses are reliable
Generalized Binary Search with Noise

Generalized Binary Search (GBS)
initialize: $n = 0, \mathcal{H}_0 = \mathcal{H}$
while $|\mathcal{H}_n| > 1$
1) Select $x_n = \arg\min_{x \in X} |\sum_{h \in \mathcal{H}_n} h(x)|$
2) Query with $x_n$ to obtain response $y_n = h^*(x_n)$
3) Set $\mathcal{H}_{n+1} = \{h \in \mathcal{H}_n : h(x_n) = y_n\}$, $n = n + 1$
Generalized Binary Search with Noise

Generalized Binary Search (GBS)

initialize: \( n = 0, \mathcal{H}_0 = \mathcal{H} \)

while \(|\mathcal{H}_n| > 1\)

1) Select \( x_n = \arg \min_{x \in \mathcal{X}} |\sum_{h \in \mathcal{H}_n} h(x)| \)

2) Query with \( x_n \) to obtain response \( y_n = h^*(x_n) \)

3) Set \( \mathcal{H}_{n+1} = \{ h \in \mathcal{H}_n : h(x_n) = y_n \} \), \( n = n + 1 \)

Suppose that the binary response \( y \in \{-1, 1\} \) to query \( x \in \mathcal{X} \) is an independent realization of the random variable \( Y \) satisfying \( \mathbb{P}(Y = h^*(x)) > \mathbb{P}(Y = -h^*(x)) \), where \( h^* \in \mathcal{H} \) is fixed but unknown (i.e., the response is only probably correct)
Suppose that the binary response $y \in \{-1, 1\}$ to query $x \in X$ is an independent realization of the random variable $Y$ satisfying $\Pr(Y = h^*(x)) > \Pr(Y = -h^*(x))$, where $h^* \in H$ is fixed but unknown (i.e., the response is only probably correct).

The noise bound is defined as $\alpha := \sup_{x \in X} \Pr(Y \neq h^*(x))$. 

---

**Generalized Binary Search (GBS)**

initialize: $n = 0$, $H_0 = H$

while $|H_n| > 1$

1) Select $x_n = \arg \min_{x \in X} |\sum_{h \in H_n} h(x)|$

2) Query with $x_n$ to obtain response $y_n = h^*(x_n)$

3) Set $H_{n+1} = \{h \in H_n : h(x_n) = y_n\}$, $n = n + 1$

---
Suppose that the binary response \( y \in \{-1, 1\} \) to query \( x \in \mathcal{X} \) is an independent realization of the random variable \( Y \) satisfying \( \mathbb{P}(Y = h^*(x)) > \mathbb{P}(Y = -h^*(x)) \), where \( h^* \in \mathcal{H} \) is fixed but unknown (i.e., the response is only probably correct).

The \textit{noise bound} is defined as \( \alpha := \sup_{x \in \mathcal{X}} \mathbb{P}(Y \neq h^*(x)) \)
Theory of Generalized Binary Search

GBS with N hypotheses/classifiers
Theory of Generalized Binary Search

GBS with $N$ hypotheses/classifiers

**Noiseless Search**

**Theorem 1**  If the neighborly condition holds, then GBS terminates with the correct hypothesis after at most $c \log N$ queries, where $c > 0$ is a small constant.
Theory of Generalized Binary Search

GBS with N hypotheses/classifiers

### Noiseless Search

**Theorem 1** If the neighborly condition holds, then GBS terminates with the correct hypothesis after at most $c \log N$ queries, where $c > 0$ is a small constant.

### Noisy Search

**Theorem 2** Let $\mathbb{P}$ denotes the underlying probability measure (governing noises and algorithm randomization). If $\beta > \alpha$ and the neighborly condition holds, then the noisy GBS algorithm generates a sequence of hypotheses satisfying

$$
\mathbb{P}(\hat{h}_n \neq h^*) \leq N (1 - \lambda)^n \leq Ne^{-cn}, n = 0, 1, \ldots
$$

with exponential constant $c > 0$. 

Friday, May 20, 2011
If we desire $P(\hat{h}_n \neq h^*) < \delta$, then we require only $n = \frac{1}{\lambda} \log \frac{N}{\delta}$ queries.
Active Clustering
Difficult or impossible to measure/observe everything in large systems
Difficult or impossible to measure/observe everything in large systems
Difficult or impossible to measure/observe everything in large systems
Network Structure and Clustering

Complex systems are not defined by the independent functions of individual components, rather they depend on the orchestrated interactions of these elements.
Network Structure and Clustering

Complex systems are not defined by the independent functions of individual components, rather they depend on the orchestrated interactions of these elements.

Network(s) of interactions can be revealed via clustering based on measured features.
Network Structure and Clustering

Complex systems are not defined by the independent functions of individual components, rather they depend on the orchestrated interactions of these elements.

Network(s) of interactions can be revealed via clustering based on measured features

- genes and expression/interaction profiles
- network routers and traffic/distance profiles

Gautam Dasarathy  Brian Eriksson

Friday, May 20, 2011
Network Structure and Clustering

Complex systems are not defined by the independent functions of individual components, rather they depend on the orchestrated interactions of these elements.

Network(s) of interactions can be revealed via clustering based on measured features

**Similarity-Based Clustering:** Each component (gene/router) has an associated feature (measurement profile). Components can be clustered based on feature similarities.
Internet Topology Inference
Internet Topology Inference
Internet Topology Inference
Internet Topology Inference

Correlation between traffic patterns at two points can indicate the similarity between nodes (e.g., number of shared links in paths)
Internet Topology Inference

Correlation between traffic patterns at two points can indicate the **similarity** between nodes (e.g., number of shared links in paths)
Network Mapping

\[ s_{1,2} > s_{1,3}, s_{2,3} \]

RTT\(_1\) & RTT\(_2\) more correlated than RTT\(_1\) & RTT\(_3\) or RTT\(_2\) & RTT\(_3\)
Network Mapping

\[ s_{1,2} > s_{1,3}, s_{2,3} \]

RTT\(_1\) & RTT\(_2\) more correlated than RTT\(_1\) & RTT\(_3\) or RTT\(_2\) & RTT\(_3\)
Active Clustering
Active Clustering

Questions:
1. Can we cluster from a subsample similarities?
2. Does random subsampling suffice?
Active Clustering

**Questions:**
1. Can we cluster from a subsample similarities?

2. Does random subsampling suffice?

**Redundancy**
Active Clustering

**Questions**
1. Can we cluster from a subsample similarities?
2. Does random subsampling suffice?

**Redundancy**
Active Clustering

Questions:
1. Can we cluster from a subsample similarities?

2. Does random subsampling suffice?

Redundancy
Active Clustering

**Questions**:
1. Can we cluster from a subsample similarities?
   
   A: Maybe unnecessary to obtain all pairwise similarities

2. Does random subsampling suffice?

**Redundancy**

Need to ask?

Very Similar

Very Dis-similar

Very Similar
Active Clustering

Questions:
1. Can we cluster from a subsample similarities?
   
   A: Maybe unnecessary to obtain all pairwise similarities

2. Does random subsampling suffice?

Redundancy

Passive (Random) Subsampling

Random subsampling will miss small clusters

Actually, we can show that at least $O(n^2/m)$ pairwise similarities are required to recover clusters of size $m$. 
Questions:
1. Can we cluster from a subsample similarities?
   A: Maybe unnecessary to obtain all pairwise similarities
2. Does random subsampling suffice?
   A: No! We will require $O(n^2)$ random similarities

Redundancy

Passive (Random) Subsampling

Random subsampling will miss small clusters

Actually, we can show that at least $O(n^2/m)$ pairwise similarities are required to recover clusters of size $m$. 
Active Clustering: Efficient Hierarchical Clustering
The proposed method **adaptively** selects the most informative pairwise similarities to recover the hierarchical clustering.

Under mild assumptions, we can discern the “outlier” of three items using only 3 pairwise similarities. i.e.,
The proposed method **adaptively** selects the most informative pairwise similarities to recover the hierarchical clustering.

Under mild assumptions, we can discern the “outlier” of three items using only 3 pairwise similarities. i.e.,

\[
\text{intra-cluster similarities} > \text{inter-cluster similarities}
\]
The proposed method **adaptively** selects the most informative pairwise similarities to recover the hierarchical clustering.

Under mild assumptions, we can discern the “outlier” of three items using only 3 pairwise similarities. i.e.,

\[ S(\bullet, \bullet) > \max \{S(\bullet, \bullet), S(\circ, \circ)\} \]

intra-cluster similarities > inter-cluster similarities
Active Clustering: Efficient Hierarchical Clustering
Active Clustering: Efficient Hierarchical Clustering

This is a sequential procedure ...
Active Clustering: Efficient Hierarchical Clustering

This is a sequential procedure ...

**Inserting a new object into a tree with \( i \) leaves**

- Pick an internal node \( v \) with \( \approx \frac{i}{2} \) objects as descendants
- Find two leaves \( x_k \) and \( x_j \) whose common ancestor is \( v \)
- Find outlier\((x_k, x_j, v)\) and discard a portion of the tree
- Proceed till there are only two leaves left and insert using a final outlier test.
Active Clustering: Efficient Hierarchical Clustering

This is a sequential procedure ...

**Inserting a new object into a tree with \( i \) leaves**
- Pick an internal node \( v \) with \( \approx i/2 \) objects as descendants
- Find two leaves \( x_k \) and \( x_j \) whose common ancestor is \( v \)
- Find outlier\((x_k, x_j, v)\) and discard a portion of the tree
- Proceed till there are only two leaves left and insert using a final outlier test.
Active Clustering: Efficient Hierarchical Clustering

This is a sequential procedure ...

**Inserting a new object into a tree with \( i \) leaves**

- Pick an internal node \( v \) with \( \approx \frac{i}{2} \) objects as descendants
- Find two leaves \( x_k \) and \( x_j \) whose common ancestor is \( v \)
- Find outlier(\( x_k, x_j, v \)) and discard a portion of the tree
- Proceed till there are only two leaves left and insert using a final outlier test.
Active Clustering: Efficient Hierarchical Clustering

This is a sequential procedure ...

**Inserting a new object into a tree with $i$ leaves**

- Pick an internal node $v$ with $\approx i/2$ objects as descendants
- Find two leaves $x_k$ and $x_j$ whose common ancestor is $v$
- Find outlier($x_k$, $x_j$, $v$) and discard a portion of the tree
- Proceed till there are only two leaves left and insert using a final outlier test.

![Diagram showing the process of inserting a new object into a tree.](image-url)
Active Clustering: Efficient Hierarchical Clustering

This is a sequential procedure ...

Inserting a new object into a tree with $i$ leaves
- Pick an internal node $v$ with $\approx i/2$ objects as descendants
- Find two leaves $x_k$ and $x_j$ whose common ancestor is $v$
- Find outlier($x_k$, $x_j$, $v$) and discard a portion of the tree
- Proceed till there are only two leaves left and insert using a final outlier test.

---

**Step 1**

**Step 2**

**Step 3**
This is a sequential procedure ...

**Inserting a new object into a tree with \( i \) leaves**
- Pick an internal node \( v \) with \( i/2 \) objects as descendants
- Find two leaves \( x_k \) and \( x_j \) whose common ancestor is \( v \)
- Find outlier\((x_k, x_j, v)\) and discard a portion of the tree
- Proceed till there are only two leaves left and insert using a final outlier test.

Active Clustering: Efficient Hierarchical Clustering
Active Clustering: Efficient Hierarchical Clustering

This is a sequential procedure ...

Inserting a new object into a tree with \( i \) leaves

1. Pick an internal node \( v \) with \( \approx i/2 \) objects as descendants
2. Find two leaves \( x_k \) and \( x_j \) whose common ancestor is \( v \)
3. Find outlier \((x_k, x_j, v)\) and discard a portion of the tree
4. Proceed till there are only two leaves left and insert using a final outlier test.

Theorem:
Under certain assumptions, the hierarchical clustering of \( n \) objects can be recovered using no more than \( 3n \log n \) sequentially and adaptively selected pairwise similarities.

within a constant factor of the information theoretic lower bound
Robust Active Clustering
Robust Active Clustering

The previous technique is very sensitive to noise/errors and violations of the assumptions.

$$S(\bullet, \circ) > \max \{S(\bullet, \circ), S(\circ, \circ)\}$$

outlier
Robust Active Clustering

The previous technique is very sensitive to noise/errors and violations of the assumptions.

\[ S(\bigcirc, \bigcirc) > \max \{S(\bigcirc, \bigcirc), S(\bigcirc, \bigcirc)\} \]

To overcome this, we design a **top-down recursive splitting approach** and use **voting** to boost our confidence about each decision we make.
Robust Active Clustering

The previous technique is very sensitive to noise/errors and violations of the assumptions.

$$S(\bullet, \circ) > \max \{S(\bullet, \circ), S(\circ, \circ)\}$$

To overcome this, we design a top-down recursive splitting approach and use voting to boost our confidence about each decision we make.

Goal: In each step, split a single cluster into 2 sub-clusters efficiently
Robust Active Clustering Procedure

**Strategy**: Sequentially decide which of the two sub-clusters each point goes into.
1. Pick a random object and call it the “seed”
Robust Active Clustering Procedure

**Strategy:** Sequentially decide which of the two sub-clusters each goes into.
1. Pick a random object and call it the “seed”
2. For the other objects, decide if they are similar to or not.
**Strategy:** Sequentially decide which of the two sub-clusters each green dot goes into.

1. Pick a random object and call it the “seed”.
2. For the other objects, decide if they are similar to the green dot or not.
3. Towards this, randomly pick $m$ “reinforcement” objects from $C$. 

**Robust Active Clustering Procedure**

Friday, May 20, 2011
**Strategy:** Sequentially decide which of the two sub-clusters each object goes into.

1. Pick a random object and call it the “seed”.
2. For the other objects, decide if they are similar to the seed or not.
3. Towards this, randomly pick $m$ “reinforcement” objects from $C$. Count the number of times $\text{outlier}(\text{seed}, \cdot, \cdot) = \cdot$. 
Strategy: Sequentially decide which of the two sub-clusters each • goes into.
1. Pick a random object and call it the “seed”.
2. For the other objects, decide if they are similar to • or not.
3. Towards this, randomly pick \( m \) “reinforcement” objects from \( C \). Count the number of times outlier(•, •, •) is •.
4. If roughly \( m/2 \) times, • is similar to •.
**Strategy:** Sequentially decide which of the two sub-clusters each \( \bullet \) goes into.

1. Pick a random object and call it the “seed”.
2. For the other objects, decide if they are similar to \( \bigcirc \) or not.
3. Towards this, randomly pick \( m \) “reinforcement” objects from \( C \). Count the number of times outlier(\( \bullet \), \( \bigcirc \), \( \bigcirc \)) is \( \bullet \).
4. If roughly \( m/2 \) times, \( \bullet \) is **similar** to \( \bigcirc \). If almost never, \( \bullet \) goes in the other cluster.
**Strategy:** Sequentially decide which of the two sub-clusters each ● goes into.

1. Pick a random object and call it the “seed”.
2. For the other objects, decide if they are similar to ● or not.
3. Towards this, randomly pick $m$ “reinforcement” objects from $C$. Count the number of times outlier(●, ●, ○) is ○.
4. If roughly $m/2$ times, ● is similar to ○. If almost never, ● goes in the other cluster.

**Theorem:** This procedure correctly clusters $n$ objects using $O(n \log^2 n)$ similarities and is robust to a significant fraction of errors.
Active Learning Summary

Classification:
NA ⇒ sample complexity $n \sim d/\epsilon$
A ⇒ sample complexity $n \sim d \log \epsilon^{-1}$

Remote Sensing:
NA ⇒ error $\sim O(n^{-1/2})$
A ⇒ error $\sim O(n^{-2})$

Network Mapping:
NA ⇒ $O(n^2)$ probes
A ⇒ $O(n \log n)$ probes
Related Work (an incomplete list)

**Active learning**
P. Hall & I. Molchanov (2003), Willett, Castro & Nowak (2005), Dasgupta
Beygelzimer, Dasgupta & Langford (2009), Hanneke (2011)

**Minimax Analysis of Statistical Learning**
Marron (1983), Yatrocos (1985), Barron (1991), Korostelev & Tsybakov

**Binary Search and Learning by Queries**

**Channel Coding with Feedback** (just the classics)
Horstein (1963), Schalkwijk & Kailath (1966), Burnashev & Zigangirov (1974),
Burnashev (1976)