Hypothesis selection with computational constraints

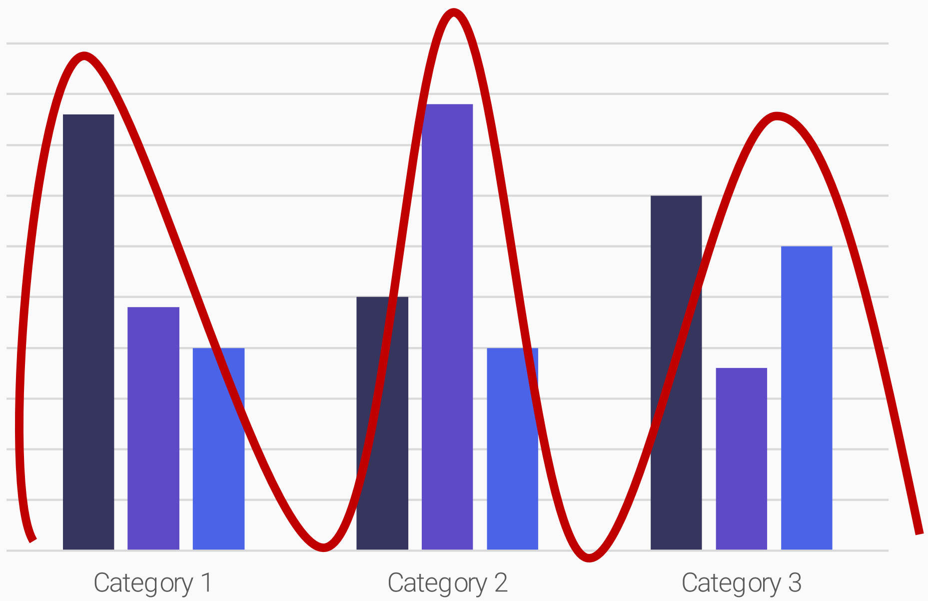
Maryam Aliakbarpour (Rice University + Simons)

Joint work with Mark Bun (BU) and Adam Smith (BU)

Extroverted sublinear algorithms (Summer 2024)

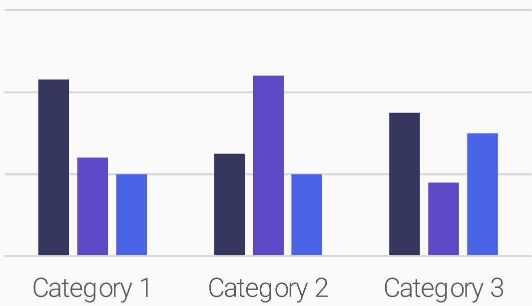
Density estimation

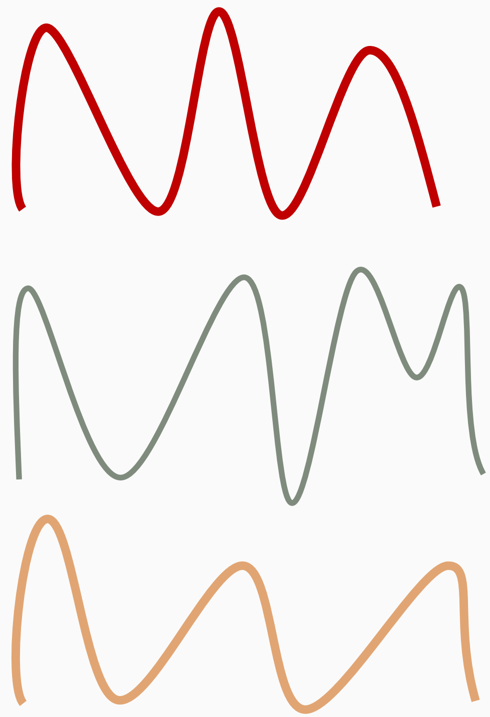
Dataset of samples from an unknown distribution
Goal: find density



Hypothesis selection

Several candidate distributions are given to us:





Applications: Cover method

Denoising

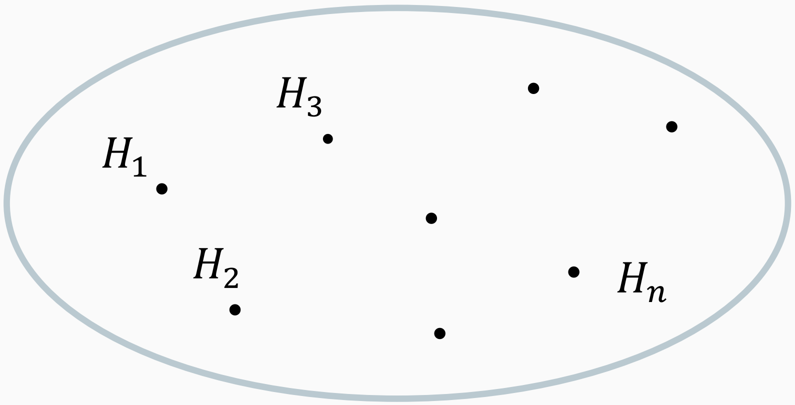
Interpretability

Determines strategies

1

Definitions

Hypothesis selection: learn P in $\mathcal{H}$



$\mathcal{H} = n$ known distributions

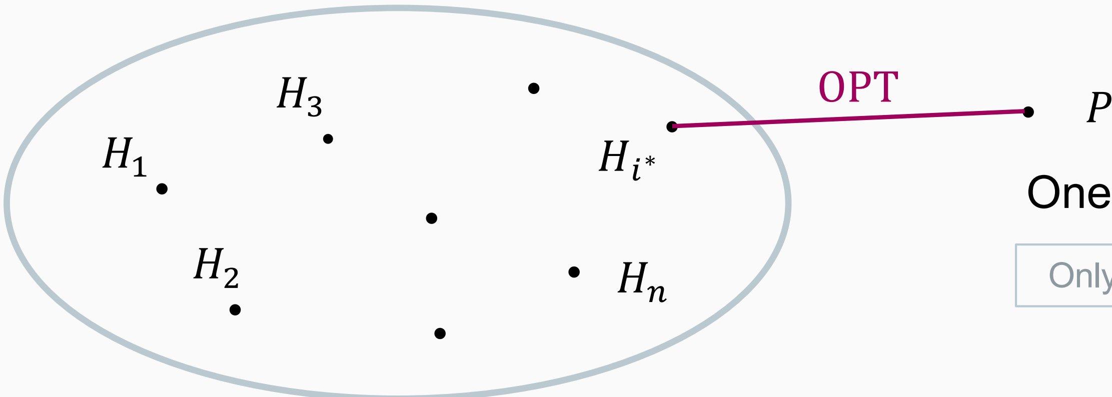
Known PDF

One unknown distribution

Only sample access

Goal: Select $\widehat{H} \in \mathcal{H}$, such that $\widehat{H}$ is close to P.

Hypothesis selection: learn P in $\mathcal{H}$



One unknown distribution

Only sample access

$\mathcal{H} = n$ known distributions

Known PDF

OPT := distance to the closest distribution in $\mathcal{H}$

Goal: Select $\hat{H} \in \mathcal{H}$, such that $\hat{H}$ is close to P.

$$|\hat{H} - P|_{TV} \le \alpha \cdot \mathrm{OPT} + \epsilon$$

Hypothesis selectin

$|\widehat{H} - P|_{TV} \leq \alpha \cdot \mathrm{OPT} + \epsilon$

Proper: $\widehat{H} \in \mathcal{H}$

Sample complexity: $s := \Theta\left(\frac{\log n}{\epsilon^2}\right)$

Yatracos '85, Devroye Lugosi '96 '97

Distribution learning

$|\hat{P} - P|_{TV} \leq \epsilon$

Arbitrary $\hat{P}$

Sample complexity: $\Theta\left(\frac{\text{doman size}}{\epsilon^2}\right)$

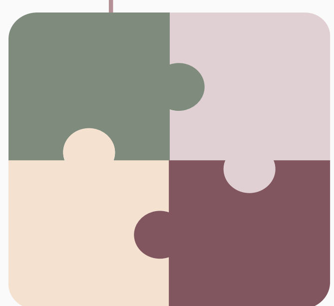
2 Our results

Classic goal: data efficiency

Use as few data points as possible

Accuracy

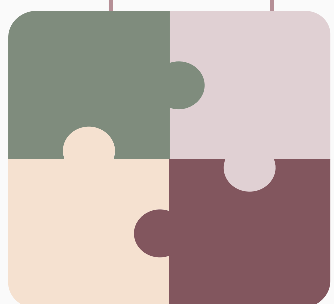
Dependencies on the error parameter



Sample complexity

data points

New goal: understanding tradeoffs between resources



Memory complexity

Accuracy

Dependencies on the error parameter

Sample complexity

data points

Time complexity

Our result

Theorem

There exists an algorithm that solves the hypothesis selection problem that runs in almost linear time in n, and obtains the following guarantee:

$$|\widehat{H} - P|_{TV} \leq 3 \cdot \mathrm{OPT} + \epsilon.$$

- Bousquet, Kane, and Moran'19 $\alpha \geq 3$ is necessary.

Table 1: Summary of Past Results in Hypothesis Selection. All algorithms use $s = \Theta(\log n/\epsilon^2)$ samples.

Result	α	Time Complexity	Additional requirement
Min distance estimate [DL01]	3	$O(n^3 \cdot s)$	
Scheffé tournament [DL01]	9	$O(n^2 \cdot s)$	
Min distance estimate [MS08]	3	$O(n^2 \cdot s)$	
[AJOS14, AFJ+18]	9	$\tilde{O}(n \cdot s)$	
→ [ABS23]	5	$\tilde{O}(n \cdot s)$	
[MS08]	3	$O(n \cdot s)$	Exponential time preprocessing
[DK14, ABS23]	≥ 3	$\tilde{O}(n \cdot s)$	Assume knowledge of OPT
Lower bound [BKM19]	Achieving $\alpha < 3$ requires poly$(\lvert\mathcal{X}\rvert)$ samples		
→ This work:Algorithm 1	3	$\tilde{O}(n \cdot s \ / \ \epsilon)$	
This work: Algorithm 4	4	$\tilde{O}\ (n \cdot s)$	

3 Backgrounds

Scheffe' sets

Minimum distance estimate

Scheffe´ set of two distributions

Scheffe´ set: $S(H_1, H_2) \coloneqq \{x \mid H_1(x) < H_2(x)\}$

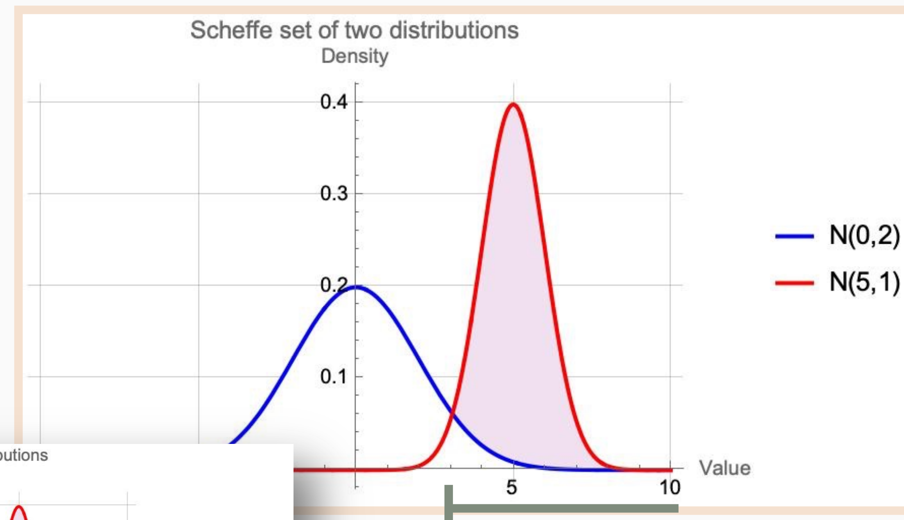
Maximizes the discrepancy:

$$H_2\big(S(H_1, H_2)\big) - H_1\big(S(H_1, H_2)\big)\ \{x \mid H_1(x) < H_2(x)\}$$

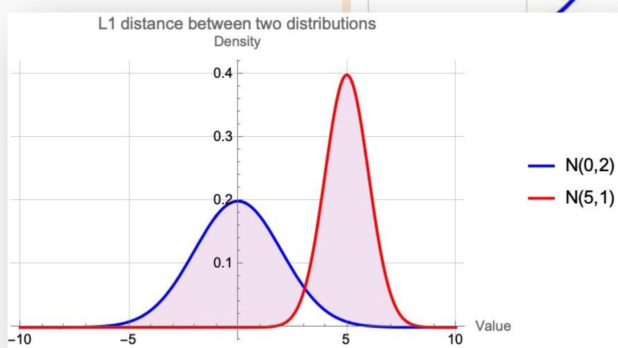
$$= \max_A |H_1(A) - H_2(A)|$$

$$=: TV(H_1, H_2)$$

$$= \frac{1}{2} L_1(H_1, H_2)$$



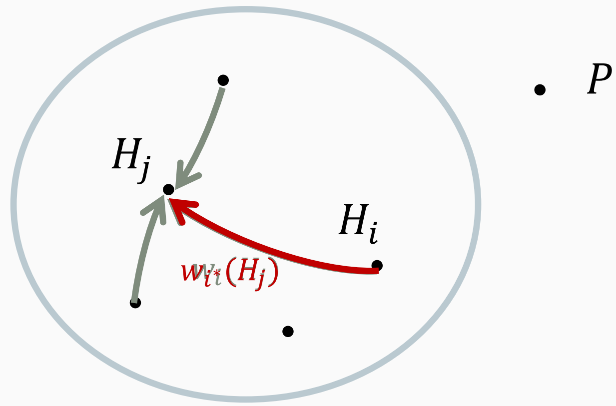
$S(H_1, H_2)$



Semi-distances: a proxy for distance to P

$$TV(H_j, P) \coloneqq \max_A |H_j(A) - P(A)|$$

$$\geq \underbrace{\left| H_j\left(S(H_i, H_j)\right) - P\left(S(H_i, H_j)\right) \right|}_{w_i(H_j) \coloneqq}$$



Observation: if $|H_{i^*} - P| = \mathrm{OPT}$, then

$$TV(H_j, P) \leq 2\ \mathrm{OPT} + w_{i^*}(H_j)$$

Find H_j such that $w_{i^*}(H_j) \leq \mathrm{OPT}$ $\Rightarrow$ $\mathrm{TV}(H_j, P) \leq 3\ \mathrm{OPT}$

Minimum distance estimate (MDE)

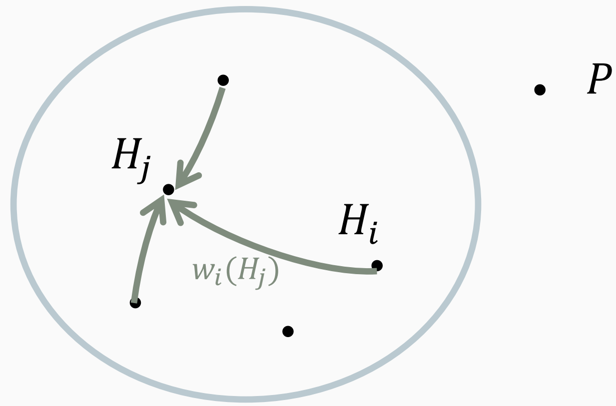
[Devroye, Lugosi 01]

Goal: Find H_j such that $TV(H_j, P) \leq 3\ \mathrm{OPT} + \epsilon$

Output: H_j with minimum $\underbrace{W(H_j) \coloneqq \max_i w_i(H_j)}_{\text{Score of } H_j}$

Proof: We show $w_{i^*}(H_j) \leq \mathrm{OPT}$

$$w_{i^*}(H_j) \leq W(H_j) \leq W(H_{i^*}) \leq TV(H_{i^*}, P) = \mathrm{OPT}$$



Sample complexity: $s = O\left(\frac{\log(n)}{\epsilon^2}\right)$ ✓

Time complexity: $O(n^2 \cdot s)$ ✗

4

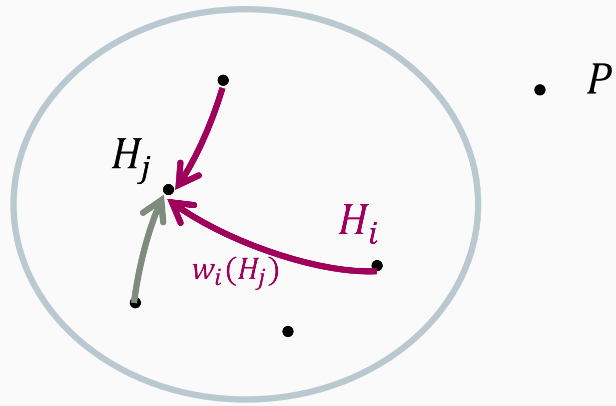
Our algorithm

The beauty of good enough

- Not enough time to compute $W(H_j) \coloneqq \max_i w_i(H_j)$
- current scores $\widetilde{W}(H_j) \coloneqq \max_{\text{some } i's} w_i(H_j)$,
- Initially all zero
- Update H_j via H_i:

$$\widetilde{W}(H_j) \leftarrow \max\left(w_i(H_j), \widetilde{W}(H_j)\right)$$

- Output: H_j with small $\widetilde{W}(H_j)$

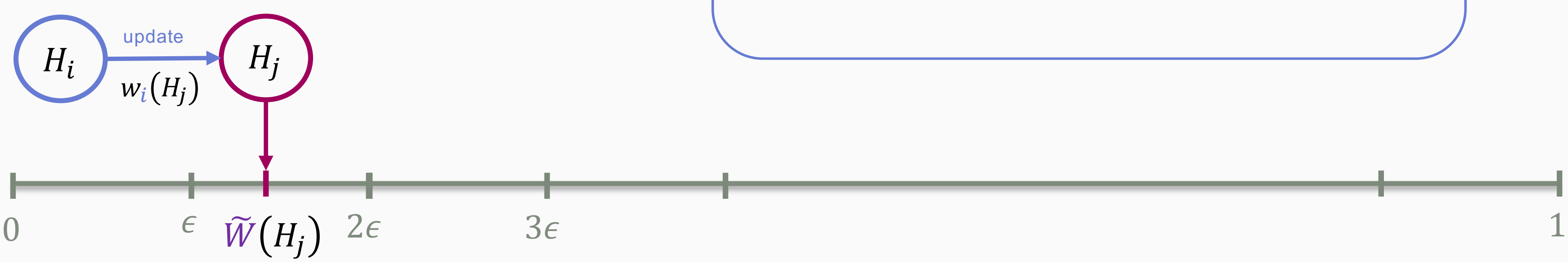


Bucketing

- $\mathrm{L} = \Theta\left(\frac{1}{\epsilon}\right)$ buckets
- H_j belongs to B_ℓ iff $\widetilde{W}\left(H_j\right) \in [(\ell - 1)\,\epsilon,\, \ell\epsilon)$

Bucketing

- $\mathrm{L} = \Theta\left(\frac{1}{\epsilon}\right)$ buckets
- H_j belongs to B_ℓ iff $\widetilde{W}(H_j) \in [(\ell-1)\,\epsilon, \ell\epsilon)$



Update via H_i

$$\widetilde{W}(H_j) \leftarrow \max\left(w_i(H_j), \widetilde{W}(H_j)\right)$$

Good H_i cause substantial updates:

Moves many H_j 's out of their buckets.

Algorithm

For $\ell = 1, 2, \dots, L$

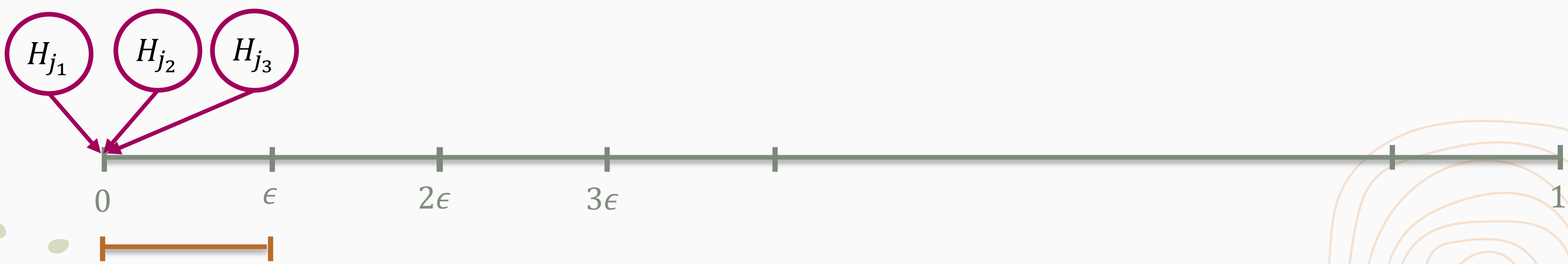
For $\mathrm{i} = 1, 2, \dots, n$

Check if H_i can cause substantial updates in B_ℓ

If yes, performs all updates by H_i

If no substantial update found, and B_ℓ is not empty yet

output a random hypothesis in B_ℓ and halt



Algorithm

For $\ell = 1, 2, \ldots, L$

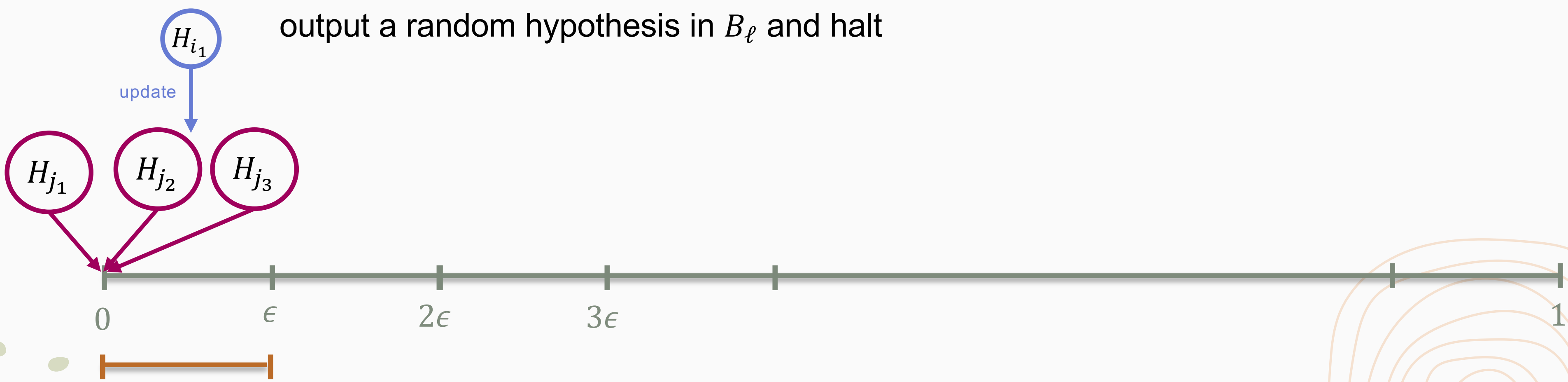
For i $= 1, 2, \ldots, n$

Check if H_i can cause substantial updates in B_ℓ

If yes, performs all updates by H_i

If no substantial update found, and B_ℓ is not empty yet

output a random hypothesis in B_ℓ and halt



Algorithm

For $\ell = 1, 2, \ldots, L$

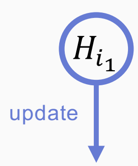
For $\mathrm{i} = 1, 2, \ldots, n$

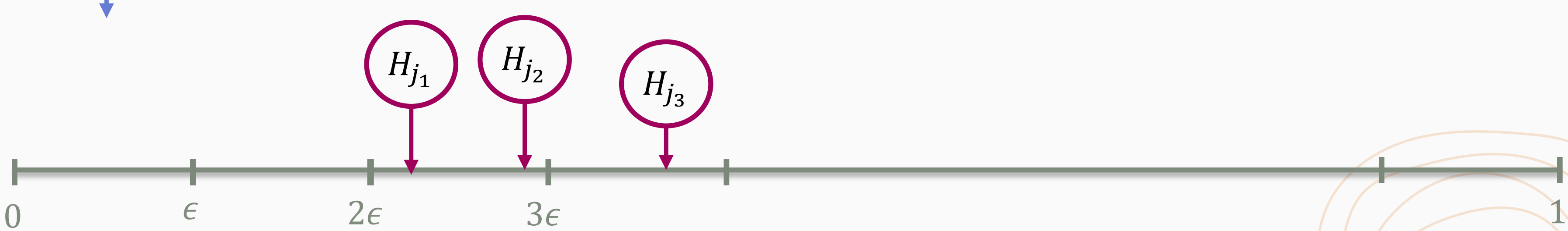
Check if H_i can cause substantial updates in B_ℓ

If yes, performs all updates by H_i

If no substantial update found, and B_ℓ is not empty yet

output a random hypothesis in B_ℓ and halt





Algorithm

For $\ell = 1, 2, \ldots, L$

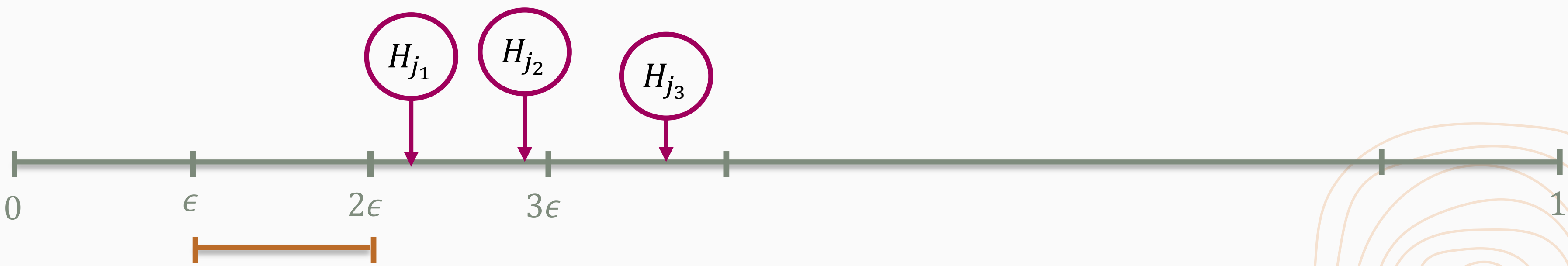
For $\mathrm{i} = 1, 2, \ldots, n$

Check if H_i can cause substantial updates in B_ℓ

If yes, performs all updates by H_i

If no substantial update found, and B_ℓ is not empty yet

output a random hypothesis in B_ℓ and halt



Algorithm

For $\ell = 1, 2, \dots, L$

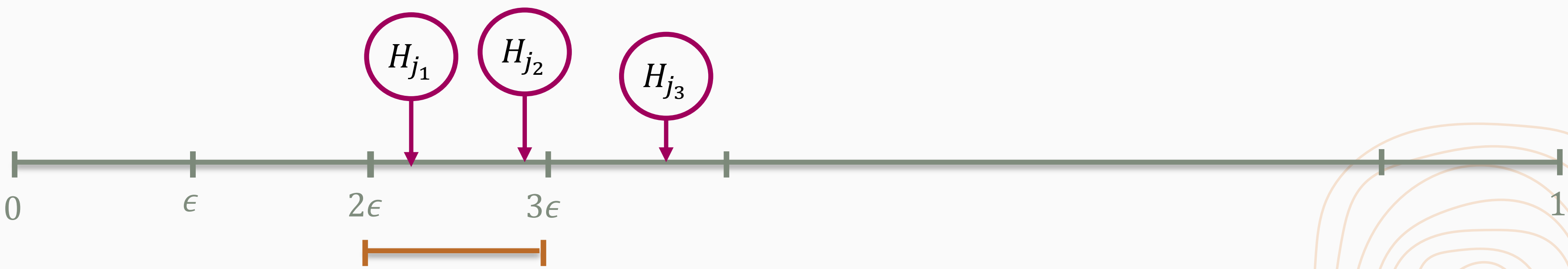
For $\mathrm{i} = 1, 2, \dots, n$

Check if H_i can cause substantial updates in B_ℓ

If yes, performs all updates by H_i

If no substantial update found, and B_ℓ is not empty yet

output a random hypothesis in B_ℓ and halt



Algorithm

For $\ell = 1, 2, \ldots, L$

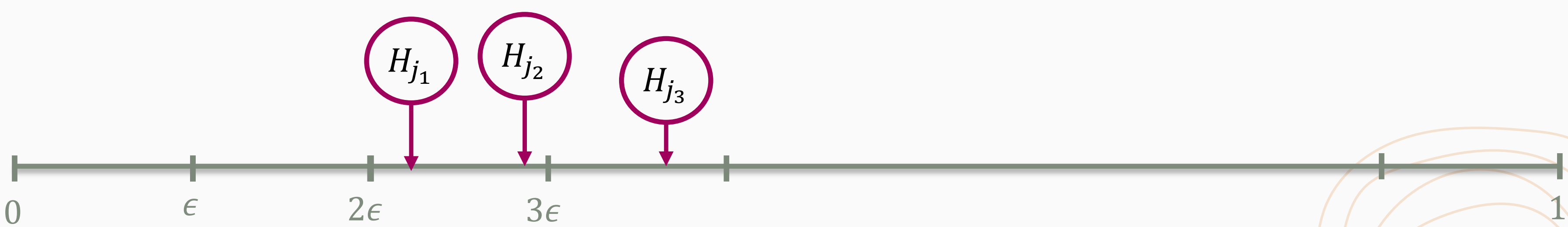
For $\mathrm{i} = 1, 2, \ldots, n$

Check if H_i can cause substantial updates in B_ℓ

If yes, performs all updates by H_i

If no substantial update found, and B_ℓ is not empty yet

output a random hypothesis in B_ℓ and halt



No good H_{i_1} found.

Algorithm

For $\ell = 1, 2, \dots, L$

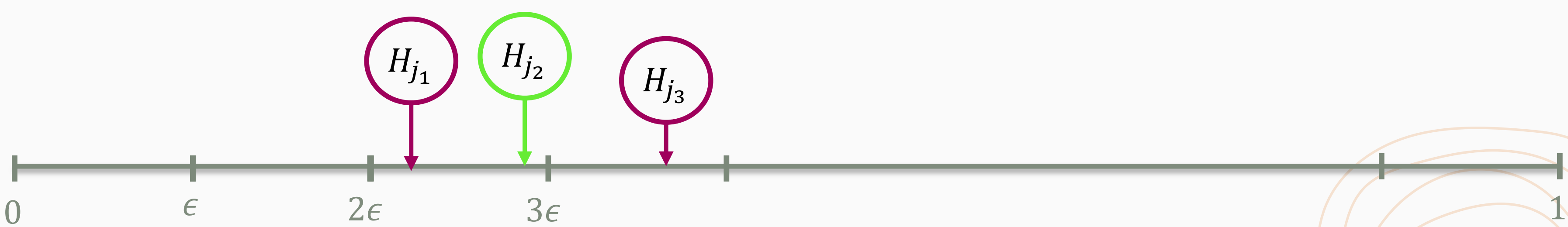
For $\mathrm{i} = 1, 2, \dots, n$

Check if H_i can cause substantial updates in B_ℓ

If yes, performs all updates by H_i

If no substantial update found, and B_ℓ is not empty yet

output a random hypothesis in B_ℓ and halt



No good H_{i_1} found.

Time complexity

For $\ell = 1, 2, \ldots, L$

For $\mathrm{i} = 1, 2, \ldots, n$

Check if H_i can cause substantial updates in B_ℓ

If yes, performs all updates by H_i

If no substantial update found, and B_ℓ is not empty yet

output a random hypothesis in B_ℓ and halt

$\tilde{\Theta}(1)$

$\tilde{\theta}\ (n)$

Time: $\Theta(n \cdot L \cdot s) = \Theta(n \cdot s/\epsilon)$

Why it works

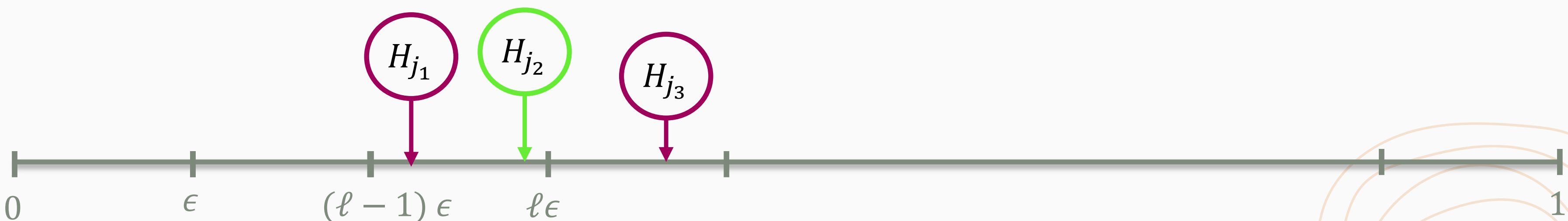
When we get stuck, H_{i^*} could not substantially update the bucket

Most H_j in the bucket are good!

Proof: We show $w_{i^*}(H_j) \leq \mathrm{OPT} + 2\epsilon$

$(\ell - 1)\epsilon \leq \widetilde{W}(H_{i^*}) \leq TV(H_{i^*}, P) = \mathrm{OPT}$

For constant fraction of H_j's $w_{i^*}(H_j) \leq \widetilde{W}(H_j) + \epsilon \leq \ell\epsilon + \epsilon$



No good H_{i_1} found.

Thanks!