Active Learning a Convex Body in Low Dimensions

Sariel Har-Peled, Mitchell Jones and Saladi Rahul

SoCG '19 (YRF), June 18-21, 2019

University of Illinois at Urbana-Champaign

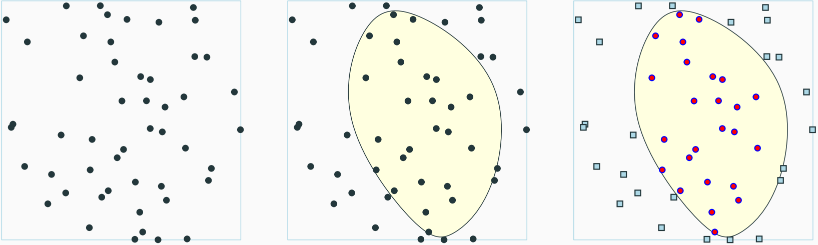
An innocent problem

Problem

Input: $P \subset \mathbb{R}^2$, oracle for unknown convex body C.

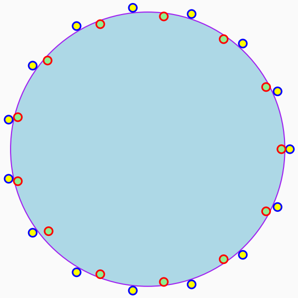
Oracle: Query $q \in \mathbb{R}^2$, returns true $\iff q \in C$.

Goal: Compute $P \cap C$ using fewest number of oracle queries.



Remarks

- Active learning
- Worst case: query all points
- **Question:** In what model can we do better?

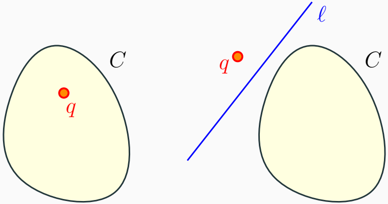


Modified problem

Problem

Input: $P \subset \mathbb{R}^2$, oracle for unknown convex body C.

Oracle: Separation oracle



Goal: Compute $P \cap C$ using fewest number of oracle queries.

Motivation

- Slighter stronger model

Motivation

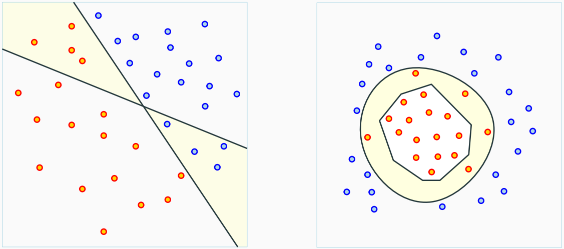
- Slighter stronger model
- Separation oracles are well-known (OR)

Motivation

- Slighter stronger model
- Separation oracles are well-known (OR)
- Other models previously studied [Angluin, 1987] [Panahi, Adler, et al., 2013] [Har-Peled, Kumar, et al., 2016]

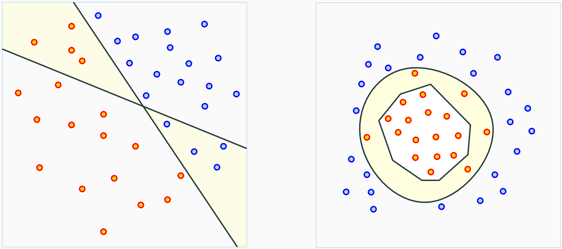
PAC learning

- Allow error in classification
- Random sampling



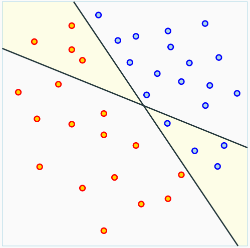
PAC learning

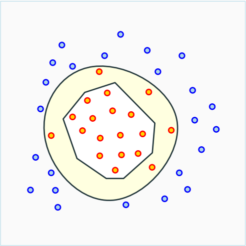
- Allow error in classification
- Random sampling
- C has bounded complexity $\implies$ finite VC dimension $\implies$ random sample of size $\approx O(\varepsilon^{-1} \log \varepsilon^{-1}) \implies \varepsilon n$ error



PAC learning

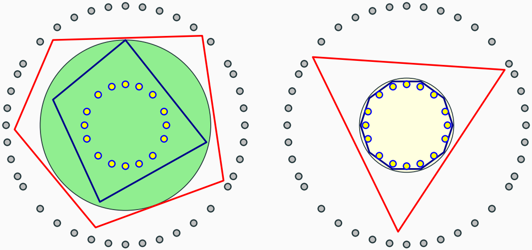
- Allow error in classification
- Random sampling
- C has bounded complexity $\implies$ finite VC dimension $\implies$ random sample of size $\approx O(\varepsilon^{-1} \log \varepsilon^{-1}) \implies \varepsilon n$ error
- Scheme fails for arbitrary convex regions





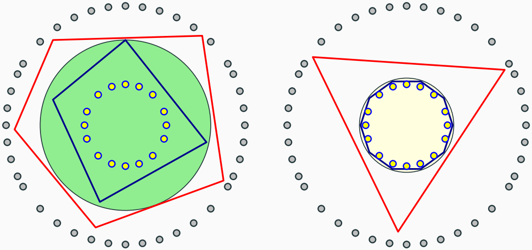
Hard vs. easy instances

- Worst case: query all points

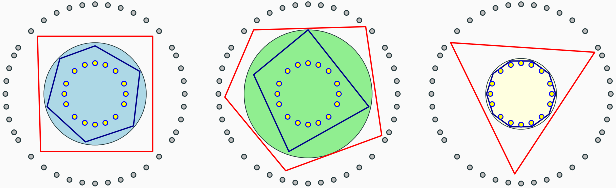


Hard vs. easy instances

- Worst case: query all points
- **Goal:** design instance sensitive algorithms

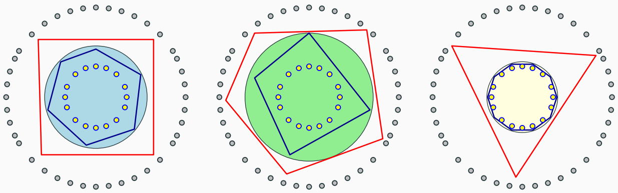


A lower bound



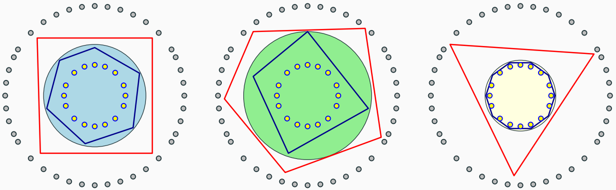
- F_{in} = convex polygon with fewest vertices s.t. $F_{\text{in}} \subseteq C$ and $C \cap P = F_{\text{in}} \cap P$.

A lower bound

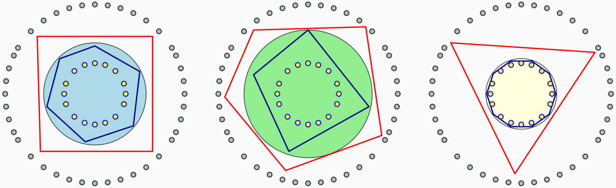


- F_{in} = convex polygon with fewest vertices s.t. $F_{\text{in}} \subseteq C$ and $C \cap P = F_{\text{in}} \cap P$.
- F_{out} = convex polygon with fewest vertices s.t. $C \subseteq F_{\text{out}}$ and $C \cap P = F_{\text{out}} \cap P$.

A lower bound



- F_{in} = convex polygon with fewest vertices s.t. $F_{\text{in}} \subseteq C$ and $C \cap P = F_{\text{in}} \cap P$.
- F_{out} = convex polygon with fewest vertices s.t. $C \subseteq F_{\text{out}}$ and $C \cap P = F_{\text{out}} \cap P$.
- Separation price $\sigma(P, C) = |F_{\text{in}}| + |F_{\text{out}}|$.



- F_{in} = convex polygon with fewest vertices s.t. $F_{\text{in}} \subseteq C$ and $C \cap P = F_{\text{in}} \cap P$.
- F_{out} = convex polygon with fewest vertices s.t. $C \subseteq F_{\text{out}}$ and $C \cap P = F_{\text{out}} \cap P$.
- Separation price $\sigma(P, C) = |F_{\text{in}}| + |F_{\text{out}}|$.

Lemma

Any algorithm must make at least $\sigma(P, C)$ oracle queries.

Results

Problem	Lowerbound	Upperbound
Classify (2D)	$\sigma(P, C)$	$O(k(P) \log n)$ (†)

(†) $k(P)$ = largest # of pts of P in convex position

Results

Problem	Lowerbound	Upperbound
Classify (2D)	$\sigma(P, C)$	$O(k(P) \log n)$ (†)
Classify (2D)	$\sigma(P, C)$	$O(\sigma(P, C) \log^2 n)$

(†) $k(P) =$ largest # of pts of P in convex position

Results

Problem	Lowerbound	Upperbound
Classify (2D)	$\sigma(P, C)$	$O(k(P) \log n)$ (†)
Classify (2D)	$\sigma(P, C)$	$O(\sigma(P, C) \log^2 n)$
Classify (3D)	—	$O(k(P) \log n)$ (†)

(†) $k(P) =$ largest # of pts of P in convex position

Results

Problem	Lowerbound	Upperbound
Classify (2D)	$\sigma(P, C)$	$O(k(P) \log n)$ (†)
Classify (2D)	$\sigma(P, C)$	$O(\sigma(P, C) \log^2 n)$
Classify (3D)	—	$O(k(P) \log n)$ (†)
Verify in (2D)	$\lvert F_{\mathrm{in}} \rvert$	$O(\lvert F_{\mathrm{in}} \rvert \log n)$

(†) $k(P)$ = largest # of pts of P in convex position

Results

Problem	Lowerbound	Upperbound
Classify (2D)	$\sigma(P, C)$	$O(k(P) \log n)$ (†)
Classify (2D)	$\sigma(P, C)$	$O(\sigma(P, C) \log^2 n)$
Classify (3D)	—	$O(k(P) \log n)$ (†)
Verify in (2D)	$\lvert F_{\mathrm{in}} \rvert$	$O(\lvert F_{\mathrm{in}} \rvert \log n)$
Verify out (2D)	$\lvert F_{\mathrm{out}} \rvert$	$O(\lvert F_{\mathrm{out}} \rvert \log n)$ (‡)

(†) $k(P) =$ largest # of pts of P in convex position

(‡) Randomized, w.h.p

Results

Problem	Lowerbound	Upperbound
Classify (2D)	$\sigma(P, C)$	$O(k(P) \log n)$ (†)
Classify (2D)	$\sigma(P, C)$	$O(\sigma(P, C) \log^2 n)$
Classify (3D)	—	$O(k(P) \log n)$ (†)
Verify in (2D)	$\lvert F_{\mathrm{in}} \rvert$	$O(\lvert F_{\mathrm{in}} \rvert \log n)$
Verify out (2D)	$\lvert F_{\mathrm{out}} \rvert$	$O(\lvert F_{\mathrm{out}} \rvert \log n)$ (‡)

(†) $k(P) =$ largest # of pts of P in convex position

(‡) Randomized, w.h.p

The greedy algorithm: preliminaries

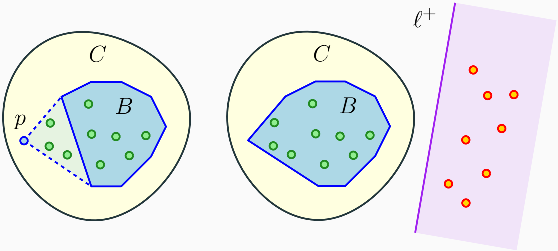
- Maintain approximation $B \subseteq C$

The greedy algorithm: preliminaries

- Maintain approximation $B \subseteq C$
- Operations:

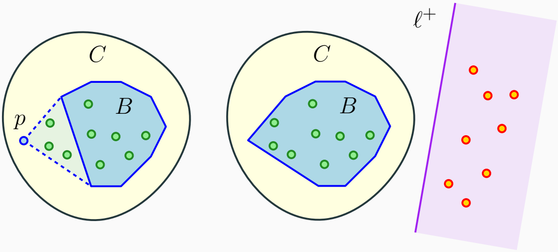
The greedy algorithm: preliminaries

- Maintain approximation $B \subseteq C$
- Operations:
  1. **expand**(p): Update $B = \mathcal{CH}(B + p)$
  2. **remove**(ℓ^+): Classify points $P \cap \ell^+$ as outside C



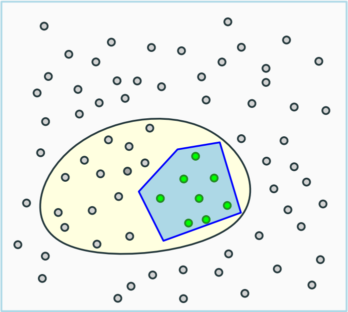
The greedy algorithm: preliminaries

- Maintain approximation $B \subseteq C$
- Operations:
  1. **expand**(p): Update $B = \mathcal{CH}(B + p)$
  2. **remove**(ℓ^+): Classify points $P \cap \ell^+$ as outside C
- $c \in \mathbb{R}^2$ is a centerpoint for P if for all halfspaces ℓ^+: $c \in \ell^+ \implies |P \cap \ell^+| \geqslant |P|/3$.



The greedy algorithm

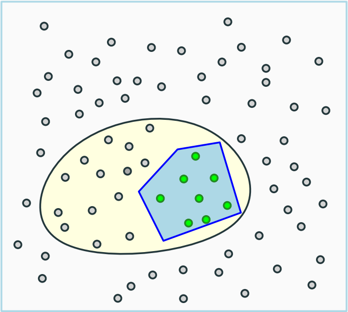
$U \subseteq P$ unclassified points. While $U \neq \varnothing$:



The greedy algorithm

$U \subseteq P$ unclassified points. While $U \neq \varnothing$:

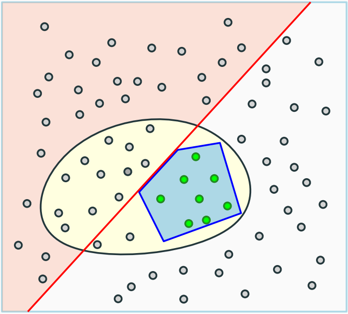
1. $\ell^+ =$ halfspace tangent to B maximizing $|\ell^+ \cap U|$



The greedy algorithm

$U \subseteq P$ unclassified points. While $U \neq \varnothing$:

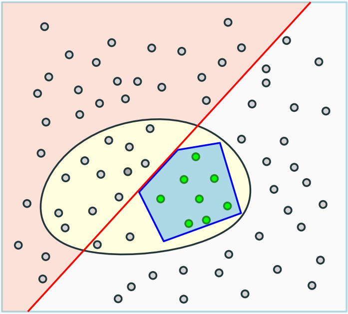
1. $\ell^+ =$ halfspace tangent to B maximizing $|\ell^+ \cap U|$



The greedy algorithm

$U \subseteq P$ unclassified points. While $U \neq \varnothing$:

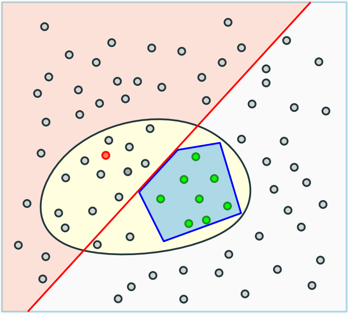
1. $\ell^+ =$ halfspace tangent to B maximizing $|\ell^+ \cap U|$
2. $c =$ centerpoint of $\ell^+ \cap U$



The greedy algorithm

$U \subseteq P$ unclassified points. While $U \neq \varnothing$:

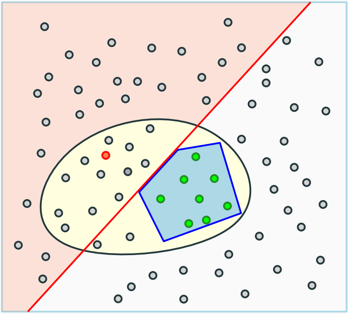
1. $\ell^+ =$ halfspace tangent to B maximizing $|\ell^+ \cap U|$
2. $c =$ centerpoint of $\ell^+ \cap U$



The greedy algorithm

$U \subseteq P$ unclassified points. While $U \neq \varnothing$:

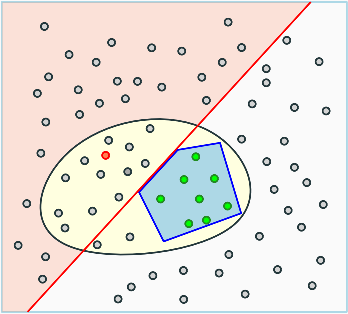
1. $\ell^+ =$ halfspace tangent to B maximizing $|\ell^+ \cap U|$
2. $c =$ centerpoint of $\ell^+ \cap U$
3. Query oracle using c:



The greedy algorithm

$U \subseteq P$ unclassified points. While $U \neq \varnothing$:

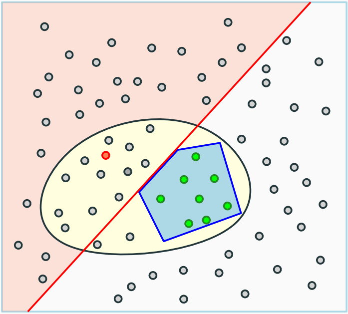
1. $\ell^+ =$ halfspace tangent to B maximizing $|\ell^+ \cap U|$
2. $c =$ centerpoint of $\ell^+ \cap U$
3. Query oracle using c:
   (A) $c \in C \implies$ **expand**(c)



The greedy algorithm

$U \subseteq P$ unclassified points. While $U \neq \varnothing$:

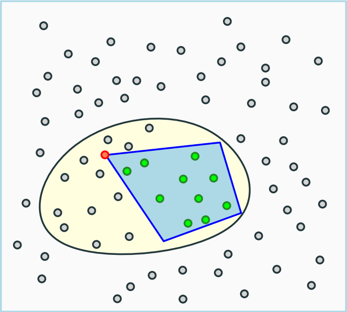
1. $\ell^+ =$ halfspace tangent to B maximizing $|\ell^+ \cap U|$
2. $c =$ centerpoint of $\ell^+ \cap U$
3. Query oracle using c:
   (A) $c \in C \implies$ **expand**(c)
   (B) $c \notin C$, h is a separating line $\implies$ **remove**(h)



The greedy algorithm

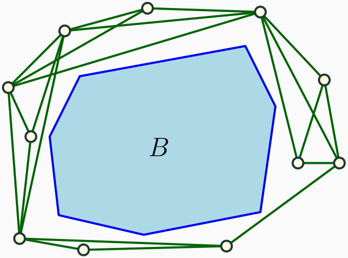
$U \subseteq P$ unclassified points. While $U \neq \varnothing$:

1. ℓ^+ = halfspace tangent to B maximizing $|\ell^+ \cap U|$
2. c = centerpoint of $\ell^+ \cap U$
3. Query oracle using c:
   (A) $c \in C \implies$ **expand**(c)
   (B) $c \notin C$, h is a separating line $\implies$ **remove**(h)



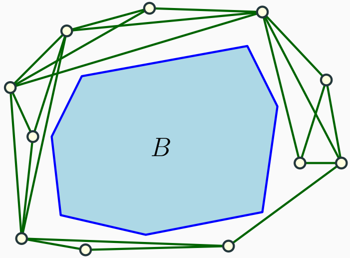
Analysis sketch

- Count visible pairs of points



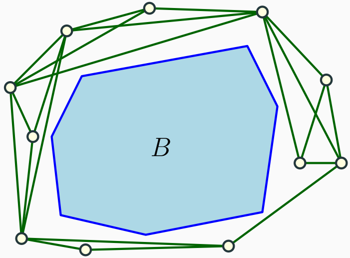
Analysis sketch

- Count visible pairs of points
- In each iteration:



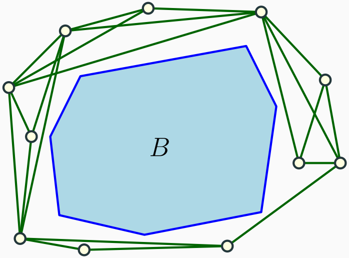
Analysis sketch

- Count visible pairs of points
- In each iteration:
  (A) Pairs lose visibility



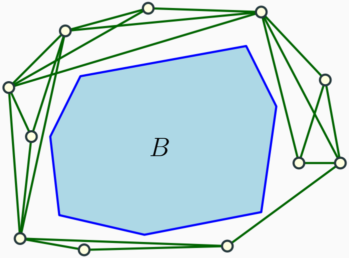
Analysis sketch

- Count visible pairs of points
- In each iteration:
  - (A) Pairs lose visibility
  - (B) Classify points



Analysis sketch

- Count visible pairs of points
- In each iteration:
  - (A) Pairs lose visibility
  - (B) Classify points



Our result

The greedy algorithm uses $O(k(P) \log n)$ queries.

($k(P) =$ largest # of pts of P in convex position.)

Conclusion & open problems

Problem	Lowerbound	Upperbound
Classify (2D)	$\sigma(P, C)$	$O(k(P) \log n)$ $O(\sigma(P, C) \log^2 n)$
Classify (3D)	—	$O(k(P) \log n)$
Verify in	$\lvert F_{\text{in}} \rvert$	$O(\lvert F_{\text{in}} \rvert \log n)$
Verify out	$\lvert F_{\text{out}} \rvert$	$O(\lvert F_{\text{out}} \rvert \log n)$

Conclusion & open problems

Problem	Lowerbound	Upperbound
Classify (2D)	$\sigma(P, C)$	$O(k(P) \log n)$ $O(\sigma(P, C) \log^2 n)$
Classify (3D)	—	$O(k(P) \log n)$
Verify in	$\lvert F_{\text{in}} \rvert$	$O(\lvert F_{\text{in}} \rvert \log n)$
Verify out	$\lvert F_{\text{out}} \rvert$	$O(\lvert F_{\text{out}} \rvert \log n)$

- Shaving log factors?

Conclusion & open problems

Problem	Lowerbound	Upperbound
Classify (2D)	$\sigma(P, C)$	$O(k(P) \log n)$ $O(\sigma(P, C) \log^2 n)$
Classify (3D)	—	$O(k(P) \log n)$
Verify in	$\lvert F_{\mathrm{in}} \rvert$	$O(\lvert F_{\mathrm{in}} \rvert \log n)$
Verify out	$\lvert F_{\mathrm{out}} \rvert$	$O(\lvert F_{\mathrm{out}} \rvert \log n)$

- Shaving log factors?
- Near-optimal solution in 3D?

Conclusion & open problems

Problem	Lowerbound	Upperbound
Classify (2D)	$\sigma(P, C)$	$O(k(P) \log n)$ $O(\sigma(P, C) \log^2 n)$
Classify (3D)	—	$O(k(P) \log n)$
Verify in	$\lvert F_{\mathrm{in}} \rvert$	$O(\lvert F_{\mathrm{in}} \rvert \log n)$
Verify out	$\lvert F_{\mathrm{out}} \rvert$	$O(\lvert F_{\mathrm{out}} \rvert \log n)$

- Shaving log factors?
- Near-optimal solution in 3D?
- Higher dimensions?

Conclusion & open problems

Problem	**Lowerbound**	**Upperbound**
Classify (2D)	$\sigma(P, C)$	$O(k(P) \log n)$ $O(\sigma(P, C) \log^2 n)$
Classify (3D)	—	$O(k(P) \log n)$
Verify in	$\lvert F_{\text{in}} \rvert$	$O(\lvert F_{\text{in}} \rvert \log n)$
Verify out	$\lvert F_{\text{out}} \rvert$	$O(\lvert F_{\text{out}} \rvert \log n)$

- Shaving log factors?
- Near-optimal solution in 3D?
- Higher dimensions?
- Conjecture: Greedy extends to $\mathbb{R}^d$ using $O(k(P)^{\lfloor d/2 \rfloor} \log n)$ queries (only interesting when $k(P) \ll (\frac{n}{\log n})^{2/d}$)

References i

D. Angluin. *Queries and concept learning*. *Machine Learning*, 2(4): 319–342, 1987.

F. Panahi, A. Adler, A. F. van der Stappen, and K. Goldberg. *An efficient proximity probing algorithm for metrology*. *Int. Conf. on Automation Science and Engineering, CASE 2013*, 342–349, 2013.

S. Har-Peled, N. Kumar, D. M. Mount, and B. Raichel. *Space exploration via proximity search*. *Discrete Comput. Geom.*, 56(2): 357–376, 2016.