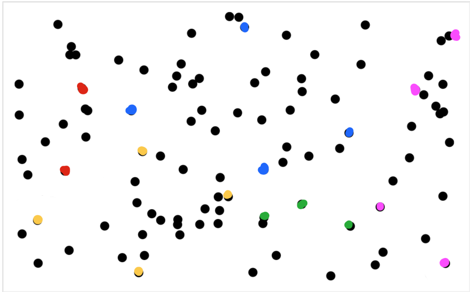
Active Learning of Classifiers with Label and Seed Queries

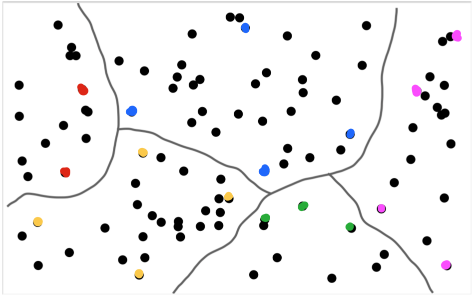
Marco Bressan, Nicolò Cesa-Bianchi, Silvio Lattanzi, Andrea Paudice, Maximilian Thiessen

October 21, 2022

Problem Definition

Learn a k-clustering $\mathcal{C} = (C_1, \ldots, C_k)$ of a finite set $X \subset \mathbb{R}^m$ with the help of a label oracle

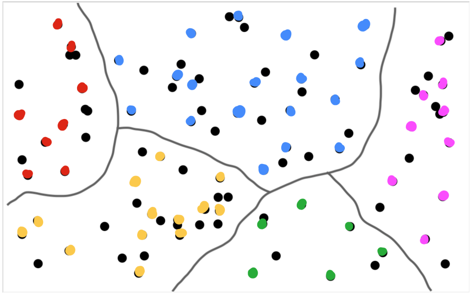




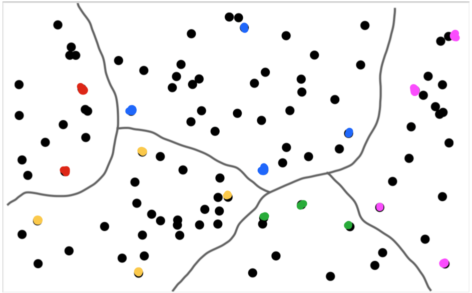
Goal 1: query efficiency, i.e., $\mathcal{O}(\log n)$ where $n = |X|$

Goal 2: computational efficiency, i.e., poly$(n + m)$ time

Problem Definition



passive learning: poly $\frac{1}{\varepsilon}$ labeled samples

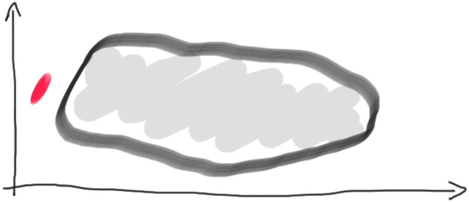


active learning: $\log \frac{1}{\varepsilon}$ queries (goal)

Problem Definition

We consider *convex* clusters in $\mathbb{R}^m$. In general, one still needs $\Omega(n)$ queries.

$\mathbb{R}$: $\mathcal{O}(\log n)$ queries via binary search

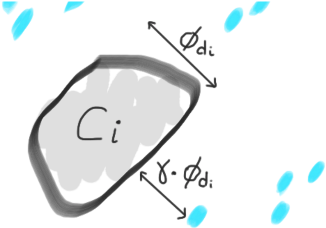


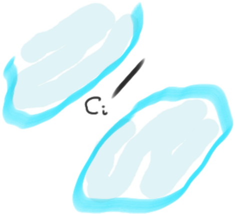
$\mathbb{R}^m$?

Convex Hull Margin

A clustering $\mathcal{C} = (C_1, \ldots, C_k)$ of X has (strong) **convex hull margin** $\gamma > 0$ if for every $i = 1, \ldots, k$ there is a **pseudometric** d_i[1] s.t. for all $j \neq i$

$$d_i(\mathrm{conv}(C_i), \mathrm{conv}(C_j)) > \gamma \cdot \phi_{d_i}(C_i)$$



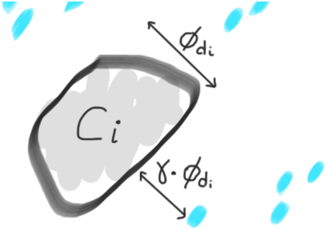


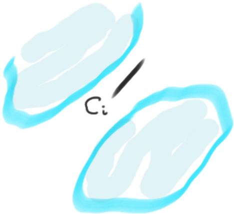
[1]induced by a seminorm

Convex Hull Margin

A clustering $\mathcal{C} = (C_1, \ldots, C_k)$ of X has (strong) **convex hull margin** $\gamma > 0$ if for every $i = 1, \ldots, k$ there is a **pseudometric** d_i[1] s.t. for all $j \neq i$

$$d_i(\mathrm{conv}(C_i), \mathrm{conv}(C_j)) > \gamma \cdot \phi_{d_i}(C_i)$$





Every cluster has its own "personalised" idea of distance d_i (unknown to the algorithm!)

Allows for SVM margin, norms induced by PSD matrices, projections on subspaces, ...

[1] induced by a seminorm

Previous Results

Theorem [BCLP'21]. k-clusterings with (strong) convex hull margin γ are learnable in time poly$(n+m)$ using a number of label queries that w.h.p. is in

$$\text{poly}(k, m, 1/\gamma)\left(1+\frac{1}{\gamma}\right)^{m} \cdot \log n$$

Moreover $\Omega\left(\left(1+\frac{1}{\gamma}\right)^{\frac{m-1}{2}}\right)$ label queries are needed in the worst case.

Can we avoid the curse of dimensionality using more powerful queries?

Seed Queries

We consider a query seed that tells whether a subset $U \subseteq X$ intersects the i-th cluster.

Hey oracle, any point from cluster 3 *here?*

Seed Queries

We consider a query seed that tells whether a subset $U \subseteq X$ intersects the i-th cluster.

Hey oracle, any point from cluster 3 *here?*

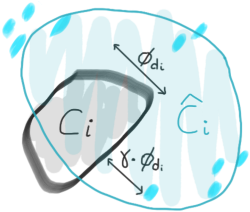
Using just seed queries, we can simulate the Halving algorithm to e.g., learn 2-clusterings with convex hull margin with $\mathcal{O}(m \log n)$ queries.

However, not clear if this is possible in time $\mathrm{poly}(n+m)$.

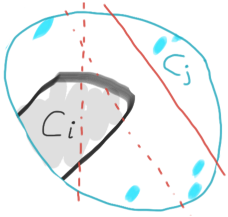
Learning Clusters with Label Queries and Seed Queries

Theorem. k-clusterings with (strong) convex hull margin γ are learnable in time $f(k)\,\mathrm{poly}(n+m)$ using in expectation $f(k)\,m^2 \log n$ label queries and $f(k)\,m \log \frac{m}{\gamma}$ seed queries.

Two phases:



Rounding

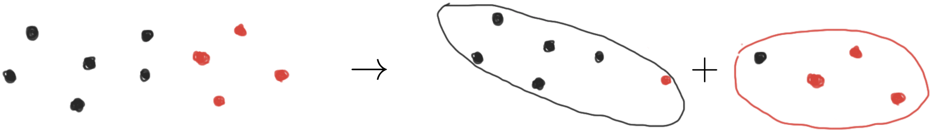


Separating

Rounding the Clusters

Definition. An α-rounding of X (w.r.t. $\mathcal{C}$) is a k-tuple $((X_i, E_i))_{i\in[k]}$ where

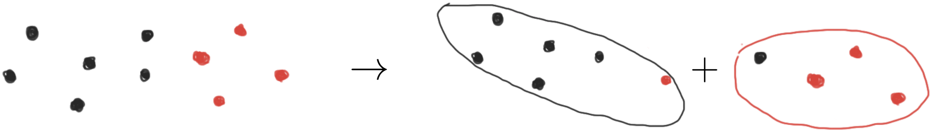
- $(X_i)_{i\in[k]}$ is a partition of X
- each E_i is an ellipsoid such that $X_i \subseteq E_i \subseteq \alpha\,\mathrm{conv}(C_i)$



Rounding the Clusters

Definition. An α-rounding of X (w.r.t. $\mathcal{C}$) is a k-tuple $((X_i, E_i))_{i\in[k]}$ where

- $(X_i)_{i\in[k]}$ is a partition of X
- each E_i is an ellipsoid such that $X_i \subseteq E_i \subseteq \alpha\,\mathrm{conv}(C_i)$

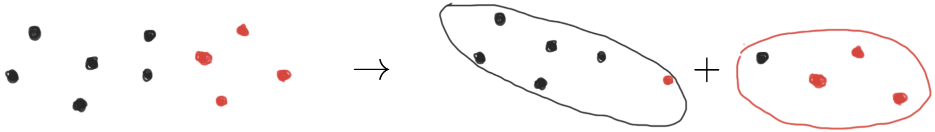


Lemma. Let p_i be the PSD metric of E_i. Then $p_i(\mathrm{conv}(X_i \cap C_i), \mathrm{conv}(X_i \cap C_j)) \geq \frac{\gamma}{\alpha}$.

Rounding the Clusters

Definition. An α-rounding of X (w.r.t. $\mathcal{C}$) is a k-tuple $((X_i, E_i))_{i\in[k]}$ where

- $(X_i)_{i\in[k]}$ is a partition of X
- each E_i is an ellipsoid such that $X_i \subseteq E_i \subseteq \alpha\,\mathrm{conv}(C_i)$



Lemma. Let p_i be the PSD metric of E_i. Then $p_i(\mathrm{conv}(X_i \cap C_i), \mathrm{conv}(X_i \cap C_j)) \geq \frac{\gamma}{\alpha}$.

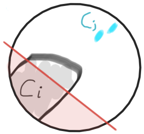
Lemma. An α-rounding of X with $\alpha = \mathcal{O}(m^3)$ can be computed in poly$(n + m)$ time and $\mathcal{O}(m^2 \log n)$ label queries.

Cutting the Version Space

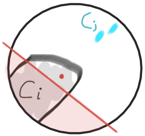
Initialize the version space to $V = B^m(0,1)$.

Compute the center of mass μ of the version space V.

Check if the halfspace corresponding to μ is correct on X_i using two seed queries.



If not, we can cut V by at least $(1 - 1/e)$ by Grünbaum's theorem.

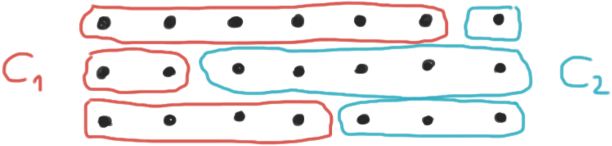


Lemma: this algorithm terminates in $\mathcal{O}(\log \frac{m}{\gamma})$ rounds (and queries).

Polynomial runtime in expectation through carefully approximating μ using hit-and-run

Lower Bounds

Can force m independent binary searches to learn $\mathcal{C}$ resulting in $\Omega(km \log \frac{1}{\gamma})$ lower bound.



Open Questions

What can we do with a **noisy** oracle?

Can we make this **parallel**, **distributed**, ...?

Thanks!