Paper - Exact recovery of Mangled Clusters

* Relaxes the γ-margin property of Ashtiani et al ~~to~~ include ellipsoid clusterings & still uses only $O(\log n)$ queries & polynomial time to find the clustering.

↳ The problem is defined by a triple (X, k, γ) where $X \subset R^d$ is a set of n points, $k \geq 2$ is the number of clusters & $\gamma \in R > 0$ is the margin. We assume $\exists$ a latent clustering $C = \{C_1, C_2 \ldots C_k\}$ over input set X. We have access to an oracle $scq(x, x')$ that answers 1 if x, x' belong to the same cluster & -1 otherwise. The goal is to recover C using as few queries as possible.

* No algorithm can take less than n queries if clustering is arbitrary. So, we must assume some structure which is a margin condition.

Let $W \in R^{d \times d}$ be a positive semidefinite matrix. Hence W induces the seminorm $\|x\|_W = \sqrt{x^T W x}$ & pseudo-metric

$$d_W(x, y) = \|x - y\|_W$$

Definition → Clustering Margin → A cluster C has margin $\gamma > 0$ if $\exists$ a PSD matrix $W = W(C)$ & a point $c \in R^d$ such that $\forall y \notin C$ & $\forall x \in C$, we have $d_W(y, c) > \sqrt{1+\gamma}\, d_W(x, c)$. If this holds for all clusters then the clustering C has margin γ.

Define $\Delta(\hat{C}, C) = \min_{\sigma \in S_k} \frac{1}{2n} \sum_{i=1}^{k} |C_i \Delta \hat{C}_{\sigma(i)}|$ where S_k is set of all permutations of $[k]$. The goal is to minimize $\Delta(\hat{C}, C)$, & use minimum queries to get $\Delta(\hat{C}, C) = 0$. The rank of a cluster

C, denoted by rank(C), is the rank of the subspace spanned by its points.

Theorem → Any instance (X, k, γ), whose latent clustering C has margin γ. Let $n = |X|$, $r \leq d$ be max. rank of a cluster in C & let $f(r, \gamma) = \max\{2^r, O(\frac{r}{\gamma} \ln(\frac{r}{\gamma}))^r\}$. RECUR outputs C with probability 1 & runs in time $O((k \ln n)(n + k^2 \ln k))$ with high probability using $O((k \ln n)(k^2 d^2 \ln k + f(r, \gamma)))$ same-cluster queries.

RECUR →

1. sampling → Draw points uniformly at random till, for some cluster C, we have a sample ~~size~~ S_C of size $\approx d^2$. Then, with good probability $|C| \simeq \frac{1}{k}|X|$ & by standard PAC bounds, any ellipsoid $\mathcal{E}$ containing S_C contains atleast half of C.
2. Computing the Minimum Volume Enclosing Ellipsoid (MVEE) → $\mathcal{E} = \mathcal{E}_J(S_C)$. $\mathcal{E}$ contains atleast half of C & some more points from X/C. Now, we find & remove these points.
3. Tesselating the MVEE → To recover $C \cap \mathcal{E}$, partition $\mathcal{E}$ into $\left(\frac{d}{\gamma}\right)^d$ hyperrectangles, each being monochromatic. So, they are seperated in $\left(\frac{d}{\gamma}\right)^d$ queries.

Theorem → Suppose we are given a subset $S_C \subseteq C$ where C is any unknown cluster. Then we can learn $C \cap \mathcal{E}_J(S_C)$ using $\max\{2^r, O(\frac{r}{\gamma} \ln \frac{r}{\gamma})^r\}$ same-cluster queries where $r = \text{rank}(C)$ & $\mathcal{E}_J(S_C)$ is MVEE of S_C

The MVEE → compute an ellipsoid close to $conv(S_C)$.

A d-rounding of S_C is any ellipsoid satisfying $\frac{1}{d}\mathcal{E} \subseteq conv(S_C) \subseteq \mathcal{E}$. $\mathcal{E} = \mathcal{E}_J(S_C)$ is a d-rounding of S_C; The d can be lowered to γ as well.

Monochromatic Tessellation → Definition → Monochromatic Subset →

A set $B \subset \mathbb{R}^d$ is monochromatic w.r.t cluster C if it doesnt contain 2 points x, y with $x \in C$ & $y \notin C$.

If $\mathcal{E}$ can be divided into m monochromatic subsets, then we can find $C \cap \mathcal{E}$ in m queries. ~~Constructi~~

Construction for $m \simeq \left(\frac{d}{\gamma} \ln \frac{d}{\gamma}\right)^d$.

Lemma → $\frac{\gamma}{C} < d_W(x,y)$ if $x \in C$ & $y \notin C$.

Let z be the point w.r.t. which margin of C holds.

$d_W(y,z) > \sqrt{1+\gamma}\, d_W(x,z)$

By triangle inequality, $d_W(y,x) \geq d_W(y,z) - d_W(x,z)$

$> (\sqrt{1+\gamma} - 1)(d_W(x,z))$

So, for $\gamma \leq c^2 - 2c \Rightarrow 1+\gamma \geq \left(1+\frac{\gamma}{c}\right)^2$

$\Rightarrow d_W(y,x) > \sqrt{\left(1+\frac{\gamma}{c}\right)^2} - 1 = \frac{\gamma}{c} d_W(x,z)$

∴ So, if $x, y \in X$ such that $x \in C$ & $y \notin C$, then $|x_i - y_i| \gtrsim \frac{\gamma}{d}$ for some i. (This is scaled down by $d_W(x,z)$)

⇒ For $\rho \simeq 1 + \frac{\gamma}{d}$, Hyperrectangle with sides $[\beta_i, \beta_i \rho]$ is monochromatic.

Consider only the positive orthant of the ellipsoid. Let the semiaxes of $\mathcal{E}$ be the canonical basis of R^d & its center μ be the origin. Let L_i be the length of i-th semi-axis of $\mathcal{E}$. We cover it with $\log_e\left(\frac{L_i}{\beta_i}\right)$ intervals of length ~~increasing~~ increasing geometrically with e.

$$T_i = \{[0,\beta_i], (\beta_i, \beta_i e] \dots (\beta_i e^{b-1}, \beta_i e^b]\} \text{ where } \beta_i > 0, e > 1$$

& $b \geq 0$ are f^ns of γ & d.

Definition $\rightarrow$ Let R_+^d be the positive orthant of R^d. The ~~tessell~~ tessellation $\mathcal{R}$ of $\mathcal{E} \cap R_+^d$ is the set of $(b+1)^d$ hyperrectangles expressed in canonical basis $\{u_1, \dots u_d\}$ of $\mathcal{E}$: $\mathcal{R} = T_1 \times \dots \times T_d$

If $\beta_i \simeq \frac{\gamma}{d} L_i$ then the point $(\beta_1, \beta_2 \dots \beta_d)$ lies well inside $conv(S_c)$. By setting β_i, e, b appropriately, $\mathcal{R}$ satisfies the following

1. $|\mathcal{R}| \leq \max\left\{1, O\left(\frac{d}{\gamma} \ln \frac{d}{\gamma}\right)^d\right\}$
2. $\mathcal{E} \cap R_+^d \subseteq \bigcup_{R \in \mathcal{R}} R$
3. For every $R \in \mathcal{R}$, the set $R \cap \mathcal{E}$ is monochromatic w.r.t C

Algorithm - TessellationLearn(X, S_c, γ)

1 compute $\mathcal{E} \leftarrow \mathcal{E}_J(S_c)$ or any other r-rounding of S_c
2 compute $\mathcal{E}_X \leftarrow X \cap \mathcal{E}$
3 compute β_i, e, b as a f^n of r, γ
4 for every $y \in \mathcal{E}_X$ do
    map y to $R(y)$
5 $x_c \leftarrow$ any point in S_c
6 while there is some unlabeled R do

label(R) ← SCQ(x_c, y), where y is any point s.t. R(y) = R
return all y mapped to R such that label(R) = +1

R(y) represents the f^n that maps y to the hyperrectangle it belongs to.

Exact recovery of all clusters →

Algorithm RECUR(X, k, γ, ϵ)

$\hat{C}_1, \hat{C}_2 \ldots \hat{C}_k \leftarrow \phi$

while $|X| > \epsilon n$ do

draw samples with replacement from $|X|$ until $|S_C| \geq bd^2 \ln k$ for some C.

$C_\epsilon \leftarrow$ TessellationLearn(X, S_c, γ)

add C_ϵ to the corresponding $\hat{C}_i$

$X \leftarrow X \setminus C_\epsilon$

return $\hat{C} = \{\hat{C}_1, \ldots \hat{C}_k\}$

Lemma 1 → The clustering $\hat{C}$ returned by RECUR deterministically satisfy $\Delta(\hat{C}, C) \leq \epsilon$. & for $\epsilon < \frac{1}{n}$ $\Delta(\hat{C}, C) = 0$

Lemma 2 → RECUR(X, k, γ, ϵ) makes $O(k^3 \ln(\frac{1}{\epsilon}))$ ~~same~~ queries in expectation & for all fixed $a \geq 1$, RECUR($X, k, \gamma, 0$) with probability at least $1 - n^{-a}$ makes $O(k^3 \ln k \ln n)$ queries & runs in time $O((k \ln n)(n + k^2 \ln k)) = \tilde{O}(kn + k^3)$

Proof → Lemma 3 → RECUR(X, k, γ, ϵ) makes atmost $8k \ln(\frac{1}{\epsilon})$ rounds in expectation & for all fixed $a \geq 1$, with probability atleast $1 - n^{-a}$ performs at most $(8k + 6a\sqrt{k}) \ln n$ rounds

At each round, with probability $\frac{1}{2}$, a fraction of $\frac{1}{4k}$ of points are labeled & removed. So at each round size of X drops by $\left(1-\frac{1}{8k}\right)$ in expectation. Hence roughly $8k\ln\left(\frac{1}{\epsilon}\right)$ rounds occur to drop $|X|$ below ϵn.

<u>Query cost of RECUR</u> $\rightarrow$ RECUR draws at most $6kd^2\ln k$ $= O(k\ln k)$ samples. Each cluster assignment requires atmost k queries $\Rightarrow O(k^2\ln k)$ same cluster queries

Tesellation Learn requires $f(d,\gamma)$ queries $= O(1)$ queries

So total expected queries $= \boxed{O(k^3\ln k\ln n)}$

<u>Running time of RECUR</u> $\rightarrow$ Drawing samples needs $O(k^2\ln k)$

For tessellation learn $\rightarrow$ $O(|S_c|^{3.5}\ln|S_c|)$ for MVEE construction

Since $|S_c| = O(d^2\ln k)$ by construction, $= \tilde{O}(1)$

Computing $\mathcal{E}_X = X \cap \mathcal{E}$ takes time $O(|X|\,\mathrm{poly}(d)) = O(n)$

For the classification, in time $O(|X\cap\mathcal{E}|)$ we can build a dictionary mapping every $R \in \mathcal{R}$ to set $R\cap\mathcal{E}_X$. Then each classification takes $O(1)$ time. To enumerate all positive R queries $O(|X\cap\mathcal{E}|\,\mathrm{poly}(d))$ time. So combining with no. of rounds bound, with probability atleast $1-n^{-1}$, runs in time $O((k\ln n)(n+k^2\ln k))$