EXPLAINABLE CLUSTERING

ANDREY SHAPIRO

Abstract. We review the topic of explainable clustering for the k-medians problem. An assignment of data points to k cluster centers is an explainable clustering if it is determined by a decision tree where each node separates data points along a threshold in a single dimension. In particular, we review the first algorithm presented by Dagupta et al.[1] when they first formalized the concept of explainable clustering. We also formalize several reductions which can be used to improve the analysis of data-oblivious algorithms for explainable clustering (an algorithm is data-oblivious if it operates without looking at data points). Finally, we review two data-oblivious algorithms designed by Esfandiari et al.[2] and Gamlath et al.[3], and in particular, focus on the methods they used.

Contents

1. Introduction

Clustering is a field of Theoretical Computer Science closely tied to machine learning and exploratory data analysis. In machine learning it is used primarily in unsupervised learning where it is used to label naturally occurring groups in the data set. Shalev-Shwartz and Ben-David provide a great introduction of the topic in chapter 22 of their book: *Understanding Machine Learning* [5].

The standard clustering problem gives us a high-dimensional data set D and asks us to find k centers $\mu^1, ..., \mu^k$ minimizing the cost: $\sum_{x \in D} \min_{i \in [k]} ||x - \mu^i||$. Note that the problem is different depending on the norm used. For this paper, we will focus on the $k-$medians problem which uses the l_1 norm. The $k-$means

Date: September 2022.

problem (which uses the l_2 norm) is another common type of clustering problem and several results in this paper have their equivalent for the $k-$means problem.

However, in many high-stakes decisions we are not satisfied by what is essentially a black box algorithm that produces groupings of data points. Often, we wish to be able to *explain* why a data point is put into one group or another. This served as the primary motivation for Dasgupta et al.[1] when they first formalised the problem of **explainable clustering**.

2. Preliminaries

We will now introduce some notation that will be used throughout the paper.

Definition 2.1. Let us fix a data set D over d dimensions and a $k \in \mathbb{N}$. The *cluster center set* $C = \{\mu^1, ..., \mu^k\}$ is a set of k points that minimises the cost:

$$\sum_{x \in D} \min_{i \in [k]} ||x - \mu_i||$$

We will define:

(1) $C(x) := \text{argmin}_{\mu \in C} ||x - \mu||$
(2) $a_i(C) := \min_{\mu \in C} \mu_i$
(3) $b_i(C) := \max_{\mu \in C} \mu_i$
(4) $I_i(C) := [a_i, b_i]$
(5) $B(C) := I_1 \times I_2 \times ... \times I_d$
(6) $L := \sum_{i=1}^{k} |I_i|$

In words, $C(x)$ is x's closest center, $I_i(C)$ is the smallest interval in the i'th dimension containing all the centers, and $B(C)$ is the bounding box that is generated by these intervals. We will drop the (C) notation when it is clear which cluster set is being used.

Definition 2.2. For $i \in [d]$, $\theta \in I_i$, the *cut* (i, θ) splits the data set into $D_l = \{x \in D : x_i \leq \theta\}$ and $D_r = \{x \in D : x_i > \theta\}$.

An *explainable clustering* $\mathcal{T}$ is a decision tree such that:

(1) Each node is a cut (i, θ) which splits the given data set into two branches.
(2) Each leaf l of $\mathcal{T}$ contains exactly one element in C.

By property (2), we can then define $\mathcal{T}(x)$ as the $\mu \in C$ in the same leaf.
Further, for each node u in $\mathcal{T}$, we can define the region $cell(u) = \{x : x \text{ reaches node } u\}$.

We can now define the notion of cost for explainable clusterings $\mathcal{T}$.

Definition 2.3. Given an algorithm that (possibly randomly) generates an explainable clustering $\mathcal{T}$ for a cluster center set C, define for a point x:

$$cost(x) = \mathbb{E}[||x - \mathcal{T}(x)||]$$

$$opt(x) = ||x - C(x)||$$

We can naturally extend these definitions to a data set D:

$$cost(D) = \sum_{x \in D} cost(x)$$

$$OPT(D) = \sum_{x \in D} opt(x)$$

In short, the goal of explainable clustering algorithms is to minimise the value $cost(D)/OPT(D)$.

Remark 2.1. It is worth noting that finding the optimum clustering is NP-hard and in practice, polynomial-time approximation algorithms are used [5]. However, we will assume throughout the paper that the clustering given to us is optimum.

3. First Formulation of Explainable Clustering

Dasgupta et al.[1] presented **Iterative Mistake Minimization** - the first algorithm that generated an explainable clustering with

$$cost(D)/OPT(D) = O(k)$$

We give a brief overview of how the algorithm works.

The algorithm builds the decision tree $\mathcal{T}$ top down. Starting at the root, it takes a node such that $|cell(u) \cap C| > 1$ and it cuts it with (i, θ), sending the data points x into corresponding branches L, R, while discarding the x's which are separated from their closest center $C(x)$. (i, θ) is chosen as the cut that minimises the number discarded x's (i.e. the cut which misplaces the least amount of points).

This algorithm has a $O(d|D|log(|D|))$ run-time since it looks at every data point in order to make decisions. As a result, the run-time depends on the size of the data set D, which can be (and usually is) very large compared to k or d.

Further, Dasgupta et al.[1] also show that this algorithm has

$$cost(\mathcal{T})/OPT(C, D) = O(k)$$

This is done by bounding the cost increase incurred by each level of the tree, and since the height of the tree is at most $k - 1$, we get the above upper bound. Several future papers, such as Esfandiari et al.[2] and Gamlath et al.[3] which we will cover later in the paper, would target this particular step by trying to bound the cost incurred by each level inversely proportional to the depth of the level, thereby ensuring better-balanced trees and therefore a lower upper bound on the cost increase.

For a lower bound, Dasgupta et al.[1] proved that a k-medians algorithm can do no better than $\Omega(log(k))$ in the worst case. They do this by arguing the existence of a configuration $C \subset \{\pm 1\}^d$ that will incur a cost of at least $log(k)$ under any sequence of cuts. The data set D of the configuration is constructed by setting exactly one coordinate of each center to 0 (thereby generating a data set of size kd). Using a probabilistic argument they are able to show that there is a C such that the centers are far enough away so that any mistake incurs an $\Omega(d)$ cost and so that there will be $\Omega(klog(k))$ mistakes.

4. Data-Oblivious Algorithms

As we saw with the IMM algorithm, looking directly at the data points can be costly for run-time, especially since the number of data points in the data sets tends to be incredibly large. As a result, data-oblivious algorithms arose which generate a threshold tree by only looking at the given centers.

Before presenting these algorithms we will mention several reductions that can be made when dealing with any oblivious algorithm. These reductions will allow us to focus our cost analysis on a smaller subset of possible configurations, thereby simplifying the proofs.

4.1. **Single Point Reduction.**

We will begin by showing that for any oblivious algorithm, it is sufficient to only consider configurations with one data point.

Theorem 4.1. *For an oblivious algorithm and a data set D with the corresponding cluster set C, there is $x \in D$ such that*

$$\frac{cost(D)}{OPT(D)} \leq \frac{cost(\{x\})}{OPT(\{x\})}$$

Proof. Let us take the $x \in D$ that maximizes the value $cost(x)/opt(x)$ (we will immediately discard all points such that $opt(x) = 0$ because then $cost(x) = 0$ since these points lie on a center).

We have for all $y \in D$,

$$\frac{cost(x)}{opt(x)} \geq \frac{cost(y)}{opt(y)}$$

so

$$cost(x)opt(y) \geq opt(x)cost(y)$$

It then follows that

$$\sum_{y \in D} cost(x)opt(y) \geq \sum_{y \in D} opt(x)cost(y)$$

and thus,

$$\frac{cost(\{x\})}{OPT(\{x\})} = \frac{cost(x)}{opt(x)} \geq \frac{\sum_{y \in D} cost(y)}{\sum_{y \in D} opt(y)} = \frac{cost(D)}{OPT(D)}$$

□

Remark 4.2. It should be pointed out that C does not correspond to the optimum clustering of $\{x\}$. However, this is of no concern for two reasons: first, if we prove any upper bound for this configuration, it still proves the same upper bound for all valid configurations (configurations where C does correspond to the optimum clustering of the given data set). Secondly, we can define a data set D' where we take x and also add two data points at the exact locations of each $\mu \in C$. Then the optimal clustering centers of D' will be precisely C, while $OPT(C, D') = OPT(C, \{x\})$ and $cost(C, D') = cost(C, \{x\})$ since the newly added points contribute nothing to the optimal cost and can never be separated from their optimal centers.

Now that we have reduced to the one-point case, we can introduce some handy notation. We will refer to the data point as x and can set it to be the origin by translation while also translating all centers. We know this preserves the expected cost since all distances are preserved. Further, we can number the centers $\mu^1, ..., \mu^k$ in order of their distance to x (for our purposes this will be l_1 distance) and we

can set $||\mu^1|| = 1$ by scaling. This immediately gives us that $OPT(\{x\}) = 1$. Since we know the point is always the origin, we can simplify our notation $cost(C) := cost(\{x\})$ and so $cost(\{x\})/OPT(\{x\}) = cost(C)$.

4.2. **Bounding Box Reductions.** We can further reduce the problem to configurations such that the bounding box B contains x.

Theorem 4.3. *For a clustering center set C and a point $x \notin B$, there is some $x' \in B$ such that $cost(x)/opt(x) \leq cost(x')/opt(x')$.*

Proof. For each $i \in [d]$ define

$$c_i = \begin{cases} x_i & \text{if } x_i \in I_i \\ a_i & \text{if } x_i < a_i \\ b_i & \text{if } x_i > b_i \end{cases}$$

Then let $x' := (c_1, ..., c_d)$ and call $v := ||x - x'||$.

Now consider any $y \in B$. We have that $y_i \in I_i$ for all $i \in [d]$ so if $x_i \notin I_i$ we have either $y_i < x_i$ in which case $y_i \leq b_i = c_i < x_i$ or $y_i > x_i$ in which case $y_i \geq a_i = c_i > x_i$. If $x_i \in I_i$ then $x_i = c_i$. Thus, we get that $|y_i - x'_i| + |x'_i - x_i| = |y_i - x_i|$. From this we have:

$$\begin{aligned} ||y - x|| &= \sum_{i \in [d]} |y_i - x_i| \\ &= \sum_{i \in [d]} |y_i - x'_i| + |x'_i - x_i| \\ &= ||y - x'|| + ||x' - x|| \\ &= ||y - x'|| + v \end{aligned}$$

Thus, we have that $opt(x) = opt(x') + v$ and likewise, $cost(x) = cost(x') + v$.

Now, since we know that $cost(y) \geq opt(y)$ and therefore $cost(y) \cdot opt(y) + v \cdot opt(y) \leq cost(y) \cdot opt(y) + v \cdot cost(y)$ for any point, we can deduce:

$$cost(x)/opt(x) = (cost(x') + v)/(opt(x') + v) \leq cost(x')/opt(x')$$

As desired. □

When the algorithm is such that the cuts are chosen uniformly over the edges of the bounding box, we can go even further by restricting the region in which the centers can be located.

We proceed by introducing a transformation that allows us to modify the given configuration while keeping the expected cost unchanged. From here on out, we will assume $x = 0$.

Definition 4.1. For a given configuration $C = \{\mu^1, ...\mu^k\}$ over d dimensions, an interval (a, b) with $a > 0$ (we will allow b to equal ∞), and a dimension p, we define:

$$T_p(a, b, C) := \{\phi^1, ..., \phi^k\}$$

where for $j \in [k]$

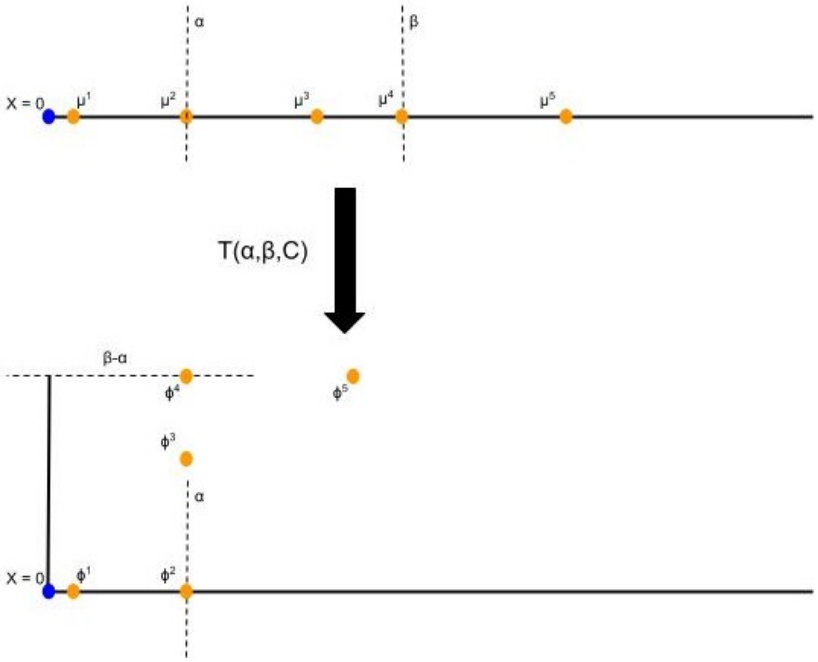

FIGURE 1. The application of the transformation T to one dimension of C

$$\phi_i^j = \begin{cases} a & \text{if } \mu_i^j \in [a,b) \\ \mu_i^j & \text{if } \mu_i^j < a \\ \mu_i^j - b + a & \text{if } \mu_i^j \geq b \end{cases}$$

$$\phi_{d+1}^j = \begin{cases} \mu_i^j - a & \text{if } \mu_i^j \in [a,b) \\ 0 & \text{if } \mu_i^j < a \\ b - a & \text{if } \mu_i^j \geq b \end{cases}$$

and for all $q \in [d]$ with $q \neq i$

$$\phi_q^j = \mu_q^j$$

The definition can also be naturally extended to accommodate intervals (a, b) with $b < 0$.

In words, transformation T takes an interval, cuts it out, and pastes it into a new dimension while preserving lengths. An example of the application of transformation T is presented in Figure 1.

Remark 4.4. We point out that $||\phi^j|| = ||\mu^j||$ and that $\mu_i^j = \phi_i^j + \phi_{d+1}^j$.

Now let us show that this transformation indeed preserves expected cost.

Lemma 4.5. $cost(T_i(a, b, C)) = cost(C)$

Proof. Let $T_i(a, b, C) = C'$ and let B, B' be the bounding boxes of C and C' respectively. Further, let us denote B_p as the length of the bounding box in the p'th dimension.

We can see that $B_p = B'_p$ for all $p \neq i, d+1$ since those dimensions remained unchanged. Furthermore, if we let μ^+ be the center with the largest i'th coordinate and μ^- as the center with the smallest, we have

$$B_i = \mu_i^+ - \mu_i^- = \phi_i^+ + \phi_{d+1}^+ - \phi_i^- - \phi_{d+1}^- = B'_i + B'_{d+1}$$

Hence, the probability that dimension $p \neq i, d+1$ is chosen for a cut is the same in C as it is in C', and further, for any chosen θ, the same subset of centers will be cut. Likewise, the probability that a cut is chosen from dimension i in configuration C is the same as the probability that a cut is chosen from dimensions i or $d+1$ in configuration C'.

Now, we can partition the i'th edge of B into three parts: $Q = [\mu_i^-, a]$, $R = [a, b]$, $S = [b, \mu_i^+]$ (if $\mu_i^+ < b$ then S is empty and $R = [a, \mu_i^+]$). Likewise, we can partition the i'th edge of B' into $Q' = [\phi_i^-, a]$, $S' = [a, \phi_i^+]$ and the $d+1$'th edge can be labeled as R'.

Note that $\phi_i^- < a$, and so $\phi_i^- = \mu_i^-$. Thus, $Q = Q'$. Further, we can see that any cut θ in Q will cut away all μ with $\mu_i > \theta + \mu_i^-$ and likewise, it will cut away exactly the ϕ with $\phi_i > \theta + \mu_i^-$. Now, there are several cases:

(1) If $\mu_i \leq \theta + \mu_i^- \leq a$, then $\phi_i = \mu_i \leq \theta + \mu_i^-$ so both ϕ and μ are not cut.
(2) If $\mu_i, a > \theta + \mu_i^-$, we have that either $\phi_i \geq a$ and is therefore cut, or
(3) $\phi_i < a$ and so $\phi_i = \mu_i > \theta + \mu_i^-$ and is therefore cut.

Hence, ϕ is cut by θ if and only if μ is.

Now, $R' = [0, \min\{b, \mu_i^+\} - a]$ while $R = [a, \min\{b, \mu_i^+\}]$. Hence, we can also look at any cut θ over the length of R. θ will cut any μ with $\mu_i > a + \theta \leq b$ and any ϕ with $\phi_{d+1} > \theta \leq b - a$. We have two cases:

(1) If $\mu_i \leq a + \theta \leq b$ then $\phi_{d+1} \leq \mu_i - a < \theta$ and so neither is cut.
(2) If $\mu_i > a + \theta$ then $\phi_i \geq \mu_i - a > \theta$ and so both are cut.

Thus, ϕ is cut if and only if μ is cut.

Finally, if S is empty then $\phi_i^+ = a$ and so S' is also empty. Otherwise, we have $\mu_i^+ > b$ and so $\phi_i^+ = \mu_i^+ - b + a$. Thus, $S = [b, \mu_i^+]$ while $S' = [a, \mu_i^+ - b + a]$ and so they have the same length. Now, taking any cut θ on S, it cuts μ such that $\mu_i > \theta + b$ and the corresponding cut cuts ϕ with $\phi_i > \theta + a$. We again have several cases:

(1) If $\mu_i > \theta + b$ then $\phi_i = \mu_i - b + a > a + \theta$ and so both are cut.
(2) If $\mu_i \leq \theta + b$ either $\mu_i < b$ in which case $\phi_i < a + \theta$ and so neither is cut or,
(3) $\mu_i \leq \theta + b$ and $\mu_i \geq b$, in which case $\phi_i = \mu_i - b + a < a + \theta$ and so neither is cut.

Thus, ϕ is cut if and only if μ is cut.

We have shown that we can partition B and B' into corresponding intervals of equal length such that the corresponding cuts in these intervals remove the same subset of centers. From this it follows that $cost(C) = cost(C')$

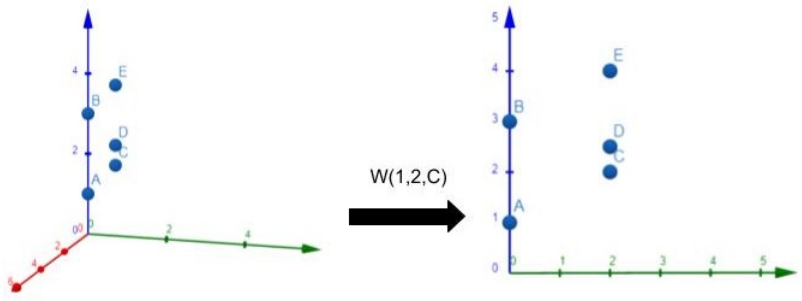

FIGURE 2. The application of the transformation $W(1, 2, C_1)$ for green being dimension 1, red dimension 2, and blue dimension 3. The points are: $A = (0, 0, 1)$, $B = (0, 0, 3)$, $C = (1, 1, 2)$, $D = (1, 1, 2.5)$, and $E = (1, 1, 4)$

□

We can also introduce a corresponding transformation W that decreases the dimension of the configuration under certain conditions.

Definition 4.2. For two dimensions i, q such that $\{\mu_i^j : j \in [k]\} = \{0, \alpha\}$ and $\{\mu_q^j : j \in [k]\} = \{0, \beta\}$ for any $\alpha, \beta \geq 0$, and such that $\{j : \mu_i^j \neq 0\} = \{j : \mu_q^j \neq 0\}$, we can define:

$$W(i, q, C) = \{\psi^1, ..., \psi^k\}$$

Where

$$\psi_i^j = \mu_i^j + \mu_q^j$$

$$\psi_p^j = \mu_p^j$$

for $p \neq i, q$. And we remove the q'th dimension by setting:

$$\phi_q^j = 0.$$

Essentially, W takes two dimensions on which all cuts behave identically and combines them. An example is presented in Figure 2.

Again, we will show that this transformation preserves the expected value.

Lemma 4.6. $cost(C) = cost(W(i, q, C))$

Proof. Let us denote $W(i, q, C) = C'$ and B, B' as before. We know that cuts remain unchanged in unaffected dimensions since $B_i' + B_q' = B_i' = \alpha + \beta = B_i + B_q$. Now, let $A = \{j : \mu_i^j \neq 0\}$. We can naturally break up $[0, B_i']$ into $[0, B_i], (B_i, B_i + B_q]$ with lengths α and β respectively. Hence, since any cut in either B_i or B_q will result in all μ^j with $j \in A$ being cut, and since any cut in B_i' will result in all ψ^j with $j \in A$ being cut. We have that ψ^j is cut if and only if μ^j is cut. Thus, as before, we have $cost(C) = cost(C')$

□

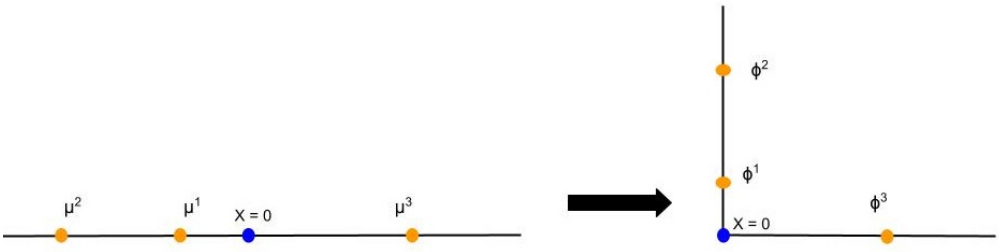

FIGURE 3. The application of the transformation $T(-\infty, 0, C)$ to one dimension of C

To summarize, transformation T takes an interval, cuts it out, and pastes it into a new dimension while preserving lengths. On the other hand, W takes two dimensions on which all cuts behave identically and combines them.

These two transformations are tools for manipulating our given configurations into simpler forms while keeping the expected cost unchanged. Using these transformations we get two corollaries.

Corollary 4.7. *For any set of centers $C = \{\mu^1, ...\mu^k\}$ over d dimensions there is a set of centers $C' = \{\phi^1, ..., \phi^k\}$ over $2d$ dimensions such that $cost(C) = cost(C')$ while also satisfying $\phi^i_j \geq 0$ for all $i \in [k], j \in [2d]$.*

Lemma 4.6 is useful because it restricts our worst-case analysis to configurations with all centers in the same section. Furthermore, since we know $x = 0$ and that it is in the bounding box, we know that x will be in the "bottom" corner of the bounding box. This greatly simplifies the object we need to analyse.

Proof. We can arrive at our desired C' by defining:

$$C_1 := T_1(-\infty, 0, C)$$
$$C_y := T_y(-\infty, 0, C_{y-1})$$
$$C' := C_k.$$

For a visualisation of the effect of applying $T_i(-\infty, 0, C)$ refer to Figure 3.

As we have shown, at each step we preserve the expected cost. Further, at step y we remove all negative values in the y'th dimension and add a dimension. Hence, we are left with $2d$ dimensions and no negative-valued coordinates.

□

We have shown that for a data-point oblivious algorithm which samples cuts uniformly, its cost analysis can be restricted to configurations with the following:

(1) One data point at $x = 0$
(2) Centers $\mu^1, ..., \mu^k$ with non-negative coordinates
(3) The bounding box contains $x = 0$

Using the transformations T and W we can also introduce the notion of a *standard form* - a simple but equivalent form of any configuration.

Definition 4.3. Fix a configuration C over d dimensions such that:

(1) For all $i \in [d]$, $\{\mu^j_i : j \in [k]\} = \{0, \alpha_i\}$ for some $\alpha_i \geq 0$

(2) For any $i \neq q$, $\{j : \mu_i^j = \alpha_i\} \neq \{j : \mu_q^j = \alpha_q\}$

Then we say that C is in **standard form**.

The consequence of the two conditions is that each dimension can be labeled with some $S \subset [k]$ and a value α_S.

With the notation of a standard form defined we can show that using the transformations T and W we can turn any C into an equivalent C' in the standard form.

Corollary 4.8. *For any configuration C there is a C' in the standard form such that $cost(C) = cost(C')$.*

Proof. First, we will apply Corollary 4.7 to ensure that C has no negative coordinates. Then, since there is a finite number of centers, we can partition each dimension i into intervals $I_1^i, I_2^i, ..., I_{l_i}^i$ such that each I_p^i does not contain 0 or any centers in its interior. We then apply T to each of these intervals (starting with I_l and working backwards to I_1). We wish to show that the resulting configuration C_1 satisfies condition (1) of Definition 4.3.

First, let us take any i from the original d dimensions of C. For ease of notation we will denote the intervals on this dimension as $J_1, ..., J_l$ and say that $J_p = [a_{p-1}, a_p]$ with $a_0 = 0$. Before any transformation is applied, we can observe that for all μ, $\mu_i \leq a_{l-1}$ or $\mu_i = a_l$. Now, after T is applied to J_l, all μ with $\mu_i = a_l$ will have instead $\mu_i = a_{l-1}$ and all other centers will remain unchanged. Thus, we have that for all μ, either $\mu_i = a_{l-1}$ or $\mu_i \leq a_{l-2}$ since there are no μ with $\mu_i \in (a_{l-2}, a_{l-1})$. Thus, by induction, we have that after all $J_l, ..., J_1$ are applied, we have that all μ have $\mu_i = 0$. Thus, we have that after all transformations are applied, all d of the original dimensions vanish.

Further, each dimension created by an application of T to one of our intervals I_i^j will satisfy condition (1) since there are no centers on the interior of I_i^j and so all centers will take a value of 0 or the length of I_i^j.

Thus, C_1 satisfies condition (1).

Next, in order to satisfy condition (2), we will take any dimensions i, q such that $\{j : \mu_i^j = \alpha_i\} = \{j : \mu_q^j = \alpha_q\}$. Then we know that we can apply $W(i, q, C_1)$ in order to combine the two dimensions into just i. Furthermore, the new dimension created will obey condition 1 since the new α value will simply be the sum $\alpha_i + \alpha_q$. Thus, we can freely apply W as necessary in order to insure that condition (2) is satisfied, while preserving condition (1). This will result in one dimension i such that $\mu_i = 0$ for all μ which can be removed entirely since it has no impact on the configuration.

Thus, we are left with a configuration C' in standard form such that $cost(C) = cost(C')$.

It can be further noted that the dimension of C' is at most $2^k - 2$ since there can only be one dimension for each subset of $[k]$ and also since there is no dimension corresponding to the empty set (all μ_i are zero) or for the full set (since then 0 would not be in the bounding box).

□

Thus, we have shown that when analyzing the worst-case expected cost of an algorithm, is suffices to focus exclusively on configurations in standard form.

5. Almost Tight Approximation

In their paper Esfandiari et al.[2] present an algorithm which generates an explainable clustering $\mathcal{T}$ independently (with respect to run-time) from the size of the data set. It achieves this by not looking at the data points at all - hence the term "data oblivious". As a result, it runs much faster than IMM.

Aside from improving run-time, it also has a better expected cost: $O(\log\, k\, \log\log\, k)$ compared to IMM's $O(k)$ [1].

Algorithm 1 Esfandiari et al.

Create tree $\mathcal{T}$ with a single node $u_0 \leftarrow \emptyset$ with $\mathcal{M}(u_0) = \{\mu^1, ..., \mu^k\}$
while $\exists$ leaf u$\in \mathcal{T}$ with $|\mathcal{M}(u)| \geq 2$ **do**
  for $r = 1$ to d **do**
    $a_r = \min_{\mu^i \in \mathcal{M}(u)}\{\mu_r^i\}$
    $b_r = \max_{\mu^i \in \mathcal{M}(u)}\{\mu_r^i\}$
  end for
  Sample (r, t) for $r \in [d]$ and $t \in (a_r, b_r)$ uniformly with respect to the total length $\sum_{r=1}^{d} b_r - a_r$.
  Add left child $\mathcal{L}(u) \leftarrow \{x_r < t\}$
  $\mathcal{M}(\mathcal{L}(u)) = \mathcal{M}(u) \cap \{\mu^i : \mu_r^i < t\}$
  Add right child $\mathcal{R}(u) \leftarrow \{x_r \geq t\}$
  $\mathcal{M}(\mathcal{R}(u)) = \mathcal{M}(u) \cap \{\mu^i : \mu_r^i \geq t\}$
end while
  return $\mathcal{T}$

We will now go through how Esfandiari et al.[2] showed their algorithm achieves an $O(log\ k\ log\ log\ k)$ cost increase.

The main method employed by the paper is the partition of the centers into groups that contain centers exponentially further from the origin. The second step is to bound the probability that the center x is assigned to a center in each of these groups.

To begin, we will introduce some notation that will serve as the frame work for the methods used by Esfandiari et al.[2]. We should note that the notation presented here is a simplified version of the notation used in the original paper since due to the reductions in Section 4 we can assume that $\mu_j^i \geq 0$ for all $i \in [k]$, $j \in [d]$.

Definition 5.1. First, we remind the reader that $x = 0$ and that $\mu^1, \mu^2, ...$ are ordered by increasing distance from the origin. We can then partition $[k]$ into $S_0, S_1, ...$ by letting $i \in S_h$ if $2^h \leq ||\mu^i|| < 2^{h+1}$. Then, we can define

$$P(H) := \max\left(\bigcup_{h \leq H} S_H\right)$$

The goal of the proof is to bound the probability that x will be assigned to some index in S_H for each $H \geq 2$.

Then we can define the event $\omega(p, H)$ for some $H \geq 2$ as follows:

Definition 5.2. (1) $\omega(P(H-2), H)$ is the event that there is some $s \in S_H$ such that the first sampled line that splits μ^1 from μ^s, splits $\mu^1, ..., \mu^{P(H-2)}$ from $x = 0$ and μ^s.

(2) $\omega(p, H)$ for $p < P(H-2)$ is the event that there is some $s \in S_H$ such that the first sampled line that splits μ^1 from μ^s also splits $\mu^1, ..., \mu^p$ from $x = 0$ and μ^s, and that the first sampled line that splits μ^{p+1} from μ^s also splits μ^{p+1} from μ^1.

The idea here is that for x to be assigned to some cluster with its index in S_H, there must be some $s \in S_H$ such that conditions (1) or (2) are satisfied.

Hence, by the union bound over p, we have that the probability of x being assigned to a cluster in S_H is at most $\sum_{p \leq P(H-2)} Pr[\omega(p, H)]$.

It then follows:

$$cost(C) \leq O\left(4 + \sum_{H \geq 2} 2^H \cdot \sum_{p \leq P(H-2)} Pr[\omega(p, H)]\right) \tag{5.1}$$

We have the $O(4)$ term from the fact that for $H < 2$ any assignment will have cost at most $2^2 = 4$. Now the task becomes to bound each of the probabilities $Pr[\omega(p, H)]$ in a way that scales inversely with 2^H.

Definition 5.3. To do this, we define for any $p > 0$:

$$\alpha_p := \sum_{j=1}^{d} \min_{i \in [p]} \{\mu_j^i\}$$

In words, α_p is the length of the subset of the bounding box edges on which a cut will result in all $\mu^1, ..., \mu^p$ being cut.

We also have

$$\beta_J := \alpha_{P(J)}$$

In words, β_J is the length of the subset of the bounding box edges on which a cut will eliminate all centers in $S_0, ..., S_J$.

Also, we define

$$\beta_{-1} := 1$$

We now present the results of the second step of the proof, i.e. the bounding of $Pr[\omega(p, H)]$. However, we omit the technical proofs and refer the reader to the original paper [2].

Lemma 5.2. *For $p = P(H-2)$, we can bound:*

$$Pr[\omega(p, H)] \leq log(k) \cdot \frac{\beta_{H-2} - \beta_H}{2^H}$$

Lemma 5.3. *For $p < P(H-2)$, we can bound:*

$$Pr[\omega(p, H)] \leq log(k) \cdot \frac{\alpha_p - \alpha_{p+1}}{2^H} \cdot min(log(k) \cdot \frac{||\mu^{p+1}||}{2^H}, 1)$$

With these, we can prove the following theorem.

Theorem 5.4. *For Algorithm 1, given any configuration C, we have $cost(C) \leq O(log\ k\ log\ log\ k)$*

Sketch of proof in [2].

By substituting Lemmas 5.2 and 5.3 into Equation 5.1 we get:

$$
\begin{aligned}
cost(C) &\leq O\left(4 + \sum_{H\geq 2} 2^H \cdot \sum_{p\leq P(H-2)} Pr[\omega(p,H)]\right) \\
&\leq O\left(\sum_{H\geq 2} 2^H \cdot \left[Pr[\omega(P(H-2),H) + \sum_{p<P(H-2)} Pr[\omega(p,H)]\right]\right) \\
&\leq O\left(\sum_{H\geq 2} 2^H \cdot \left[\log(k)\cdot\frac{\beta_{H-2}-\beta_H}{2^H} + \sum_{p<P(H-2)} \log(k)\cdot\frac{\alpha_p-\alpha_{p+1}}{2^H}\cdot\min(\log(k)\cdot\frac{||\mu^{p+1}||}{2^H},1)\right]\right) \\
&\leq O\left(\log(k)\cdot\sum_{H\geq 2} 2^H \cdot \left[\frac{\beta_{H-2}-\beta_H}{2^H} + \sum_{J=0}^{H-2}\sum_{p+1\in S_J}\frac{\alpha_p-\alpha_{p+1}}{2^H}\cdot\min(\log(k)\cdot\frac{||\mu^{p+1}||}{2^H},1)\right]\right)
\end{aligned}
$$

The last step works because since $p < P(H-2)$, $p+1 \in S_{H-2}$. Then continuing:

$$
= O\left(\log(k)\cdot\sum_{H\geq 2} 2^H \cdot \left[\sum_{J=0}^{H}\frac{\beta_{J-1}-\beta_J}{2^H}\cdot\min(\log(k)\cdot\frac{||\mu^{p+1}||}{2^H},1)\right]\right)
$$

This step comes from the fact that $\sum_{p+1\in S_J}(\alpha_p - \alpha_{p+1}) = \beta_{J-1} - \beta_J$. Finally:

$$
\begin{aligned}
&\leq O\left(\log(k)\cdot\sum_{H\geq 2}\left[\sum_{J=0}^{H}(\beta_{J-1}-\beta_J)\cdot\min(\log(k)\cdot\frac{2^J}{2^H},1)\right]\right) \\
&\leq O\left(\log(k)\cdot\sum_{J\geq 0}(\beta_{J-1}-\beta_J)\cdot\sum_{H\geq J}\min(\log(k)\cdot\frac{2^J}{2^H},1)\right) \\
&\leq O\left(\log(k)\cdot\sum_{J\geq 0}(\beta_{J-1}-\beta_J)\cdot\log(\log(k))\right) \\
&\leq O\left(\log(k)\cdot\log(\log(k))\cdot\beta_{-1}\right) = O(\log k \log\log k)
\end{aligned}
$$

The penultimate step follows from the fact that for sufficiently large J, $\beta_J = \alpha_k$, and $\alpha_k = 0$ since by the reduction achieved by Theorem 4.3 there is no cut that removes all k centers. Hence, we have shown the desired upper bound on cost. □

Remark 5.5. It should be noted that in the original paper [2], they also proved the upper bound with respect to dimension, arriving at $cost(C) \leq O(d\cdot\log^2 d)$. We did not cover this part of the proof, but it is worth pointing out that although the reductions we used from Section 4 increase the number of dimensions d, the one we used (Corollary 4.7) only doubled the number of dimensions, which keeps the $O(d\cdot\log^2 d)$ unchanged.

6. A Different Approach

In their paper, Gamlath et al.[3] also present a data-oblivious algorithm which generates an explainable clustering $\mathcal{T}$ in $O(dk\log k)$ time.

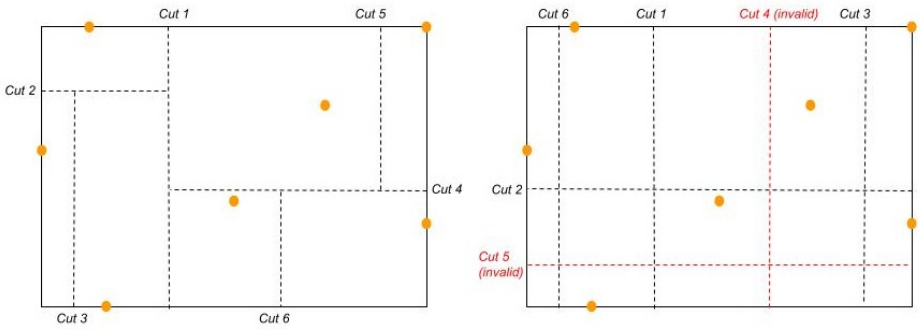

FIGURE 4. The cuts chosen by Algorithm 1 on the left vs the cuts chosen by Algorithm 2 on the right (notice that some cuts are redundant because they do not split any centers in any leaf)

Algorithm 2 Gamlath et al.

Create tree $\mathcal{T}$ with a single node $u_0 \leftarrow \emptyset$ with $\mathcal{M}(u_0) = \{\mu^1, ..., \mu^k\}$
for $r = 1$ to d **do**
  $a_r = \min_{\mu^i \in \mathcal{M}(u_0)}\{\mu^i_r\}$
  $b_r = \max_{\mu^i \in \mathcal{M}(u_0)}\{\mu^i_r\}$
end for
while $\exists$ leaf u$\in \mathcal{T}$ with $|\mathcal{M}(u)| \geq 2$ **do**
  Sample (r, t) for $r \in [d]$ and $t \in (a_r, b_r)$ uniformly with respect to the total length $\sum_{r=1}^{d} b_r - a_r$.
  if (r, t) separates some μ^i, μ^j in any leaf $u \in \mathcal{T}$ **then** for all such u
    Add left child $\mathcal{L}(u) \leftarrow \{x_r < t\}$
    $\mathcal{M}(\mathcal{L}(u)) = \mathcal{M}(u) \cap \{\mu^i : \mu^i_r < t\}$
    Add right child $\mathcal{R}(u) \leftarrow \{x_r \geq t\}$
    $\mathcal{M}(\mathcal{R}(u)) = \mathcal{M}(u) \cap \{\mu^i : \mu^i_r \geq t\}$
  end if
end while
  return $\mathcal{T}$

The main difference between Algorithm 2 and Algorithm 1 is that the space we select the cuts from is fixed at the start, and cuts are applied to all nodes that they are relevant to while those that are not relevant to any nodes, i.e. don't separate any centers, are tossed away. This difference is illustrated in Figure 4. However, under the single-point reduction from Section 4, these two algorithms act identically with respect to cost analysis. This is because we are only considering configurations with one data point and so no distinction is made between applying a cut to all nodes (all but one of which have no data points) and applying the cut only to the node containing our chosen x. Hence, the two algorithms have identical upper bounds on the cost increase of explainable clustering.

Although Algorithm 1 has a better proven upper-bound than Algorithm 2 ($O(log\ k \cdot log\ log\ k)$ vs $O(log^2 k)$), and despite the fact that the two algorithms function identically with respect to upper-bound analysis, the analysis of Algorithm 2 is still

worthwhile to investigate because it uses a different technique in its proof and because the same paper that presented Algorithm 2 also conjectured that the cost (asymptotic in k) actually has an upper bound of $O(log\ k)$ [3].

Whereas in the previous section we focused our analysis on partitioning the centers into groups and bounding the probability of them being chosen, in this analysis we partition the steps of the algorithm into groups, and bound the cost increase incurred by each group. However, the grouping is similar in the sense that groups are determined by the steps of the algorithm that halve the maximum distance between two active centers (whereas in the previous section each group represented a doubling in distance to the origin).

We will now go over the proof for the upper bound of Algorithm 2. However, much like [2], the original paper [3] analysed all possible configurations while we will use the reductions in Section 4 in order to shorten the proof.

Let $f_i(\theta)$ be the indicator that tells us if the cut (i, θ) separates $x = 0$ and μ^1 (1 if it does, 0 if it doesn't). Also recall that $L := \sum_{i=1}^{k} |I_i|$ where $I_1 \times ... \times I_d = B$

Then we have the following lemma:

Lemma 6.1. $\mathbb{E}[f_i(\theta)] = 1/L$.

In the original paper [3], the proof for this lemma is substantial and the lemma itself also has $\leq$ instead of $=$. However, due to the reductions in Section 4 we get equality and the lemma follows from the fact that the chance that x is separated from μ_1 by any cut is $||\mu_1 - x||/L = 1/L$.

We now present the main part of the proof by Gamlath et al.[3]. The central method in their paper is to track the value of the maximum distance between any two centers still not cut and to use this value to bound the cost increase as the algorithm proceeds.

We first define:

Definition 6.1. For a step in the algorithm t,

$$c_{max}(t) = \max_{u \in \text{Leaves}(t)} \max_{\mu^i, \mu^j \in u} ||\mu^i - \mu^j||.$$

We define c_{min} in an analogous way.

Lemma 6.2. *Fix the cuts selected during the first $t-1$ iterations. Let $M := 3ln(k) \cdot 2L/d_{max}(t)$. Then*

$$Pr[c_{max}(t + M) \leq c_{max}(t)/2] \geq 1 - 1/k$$

where the probability is over the random cuts selected in iterations $t, t+1, ..., t+M-1$.

Proof. Let us take two μ^i, μ^j in the same leaf. Then the probability that a cut separates them is $||\mu^i - \mu^j||/L$. Thus, if the centers are at least $c_{max}(t)/2$ away from each other, the probability that they are not separated is:

$$\left(1 - \frac{c_{max}/2}{L}\right)^M = \left(1 - \frac{c_{max}(t)}{2L}\right)^{3ln(k) \cdot 2L/c_{max}(t)} \leq (1/e)^{3ln(k)} = 1/k^3.$$

Then since there are at most $\binom{k}{2}$ pairs of centers, by the union bound, we have a probability of at least $1 - 1/k$ that each pair is separated by the sequence of cuts in

iterations $t, ..., t+M-1$, in which case all pairs of centers in the same leaf are at most a distance of $c_{max}(t)/2$ away from each other and so $c_{max}(t+M) \leq c_{max}(t)/2$. □

Lemma 6.2 gives us a way to break the algorithm process into $1+log_2(c_{max}(0)/c_{min}(0))$ *steps* (each step is a sequence of cuts that halves the value of c_{max}). We then only need to bound the expected cost increase at each of these steps by $O(log(k))$ to prove the result.

Lemma 6.3. *For each step r corresponding to the sequence*

$$r_s = \{t : c_{max}(t) \in (c_{max}/2^{r+1}, c_{max}/2^r]\}$$

the expected cost increase over this step is bounded by $12\ ln(k)$.

Sketch of proof in [3].

$$\begin{aligned}\mathbb{E}\left[\text{cost-increase}(r)\right] &\leq \mathbb{E}\left[\sum_{t\in r_s} c_{max}(t) f_{i_t}(\theta_t)\right] \\ &\leq \mathbb{E}\left[\sum_{t\in r_s} c_{max}(t)\right]/L \\ &\leq M \cdot c_{max}(t)/L = 6\ ln(k)\end{aligned}$$

The third step comes from the fact that $c_{max}(t)$ decreases as t increases, and that we expect $|r_s| \leq M$ with high probability.

It should be noted that, as is, there are two issues with the proof. First, the probability that M cuts halve the value of $c_{max}(t)$ is high but not equal to 1. Second, $c_{max}(0)/c_{min}(0)$ can be arbitrarily large. Both of these issues are addressed in detail in the original paper [3], but in the interest of preserving the focus of this paper, we will omit the detailed proofs. We will only mention that the first issue is addressed by increasing the bound by a factor of 2 (hence the $12\ ln(k)$ final result), and the second is addressed by modifying the original algorithm to throw away cuts that separate centers that are too close with respect to c_{max}.

□

Then by combining Lemma 6.2 and Lemma 6.3 we get the final $O(log^2(k))$ upper bound on the cost increase.

7. Conjecture

In their paper, Gamlath et al.[3] conjectured that their algorithm achieves

$$\frac{cost(D)}{OPT(D)} = O(1 + H_k - 1) = O(ln(k))$$

where H_k is the k'th harmonic number. This is in contrast to the $O(log^2(k))$ performance that they proved in the same paper. The intuition for this conjecture arises from the analysis of the following special case presented by Gamlath et al.[3].

Lemma 7.1. *Let $D = \{0\}$ and $C = \{\mu^1, .., \mu^k\}$ such that $\mu^i = (0, ..., M_i, .., 0)$ for $i > 1$, $M_i \geq 1$, and $\mu^1 = (1, 0, ..., 0)$. Then $cost(\{0\})/OPT(\{0\}) \leq 1 + H_{k-1}$.*

Proof. We will prove this by induction over k. For $k = 1$, we get an expected value of 1 automatically.

Now suppose this is true for $k < N$ for some N. Then given a set $C = \{\mu^1, ..., \mu^N\}$ with corresponding $M_1, ..., M_N$ (we can assume they are in increasing order and that $M_1 = 1$) let us define $L := \sum_{i=1}^N M_i$. Then on the first cut, the probability that μ^i is removed is M_i/L (notice that it is impossible to cut away multiple centers at once). We can write

$$\begin{aligned} cost(C) &= \sum_{i=1}^{N} Pr[\mu^i \text{ is cut}] \cdot cost(C \setminus \{\mu^i\}) \\ &= \sum_{i=1}^{N} \frac{M_i}{L} \cdot cost(C \setminus \{\mu^i\}) \\ &= \frac{1}{L} \cdot cost(C \setminus \{\mu^1\}) + \sum_{i=2}^{N} \frac{M_i}{L} \cdot cost(C \setminus \{\mu^i\}) \\ &\leq \frac{1}{L} \cdot cost(C \setminus \{\mu^1\}) + \frac{L-1}{L}(1 + H_{N-2}) \end{aligned}$$

Now, since it is impossible to cut more than one center at a time, and since the further away a center is from 0 the more likely it is to be cut, we know that the expected cost will be smaller than the average length M_I:

$$cost(C \setminus \{\mu^1\}) \leq \frac{\sum_{i=2}^{N} M_i}{N-1} = \frac{L-1}{N-1}.$$

Then we have:

$$cost(C) \leq \frac{L-1}{L}(1+H_{N-2}) + \frac{L-1}{L(N-1)} = \frac{L-1}{L}\left(1 + H_{N-2} + \frac{1}{(N-1)}\right) < 1 + H_{N-1}.$$

Thus, by induction, we have $cost(C) \leq 1 + H_{k-1}$ for $|C| = k$.

□

Consider in particular configurations denoted by C_M (for $M \geq 1$) which are defined by setting $M_1 = 1$ and $M_i = M$ for all $i > 1$. Then we can show that for large enough M, we can asymptotically approach $1 + H_{N-1}$. First, observe that as $M \to \infty$, $M/L \to 1/(N-1)$.

We have

$$cost(C) = \sum_{i=1}^{N} \frac{M_i}{L} \cdot cost(C \setminus \{\mu^i\}) = \frac{M}{L} \cdot cost(C \setminus \{\mu^2\}) \cdot (N-1) + \frac{1}{L} \cdot M$$

And then letting M tend to infinity we approach:

$$cost(C \setminus \{\mu^2\}) \cdot 1 + \frac{1}{N-1} = 1 + H_{N-2} + \frac{1}{N-1} = 1 + H_{N-1}$$

Hence, if the conjecture were correct, the bound would be tight.

Remark 7.2. It is also worth noting that in standard form, C_M has the form $\alpha_S = M$ for $S = \{i\}$ with $i > 1$, $\alpha_S = 1$ for $S = \{1\}$, and $\alpha_S = 0$ for all other $S \subset [k]$.

Due to the reductions in Section 4, much like C_M, all configurations we consider will also have one data point at the origin and cluster centers with non-negative coordinates. Then since $cost(C_M)$ tends to $1 + H_{k-1}$, one possible approach to proving the conjecture from [3] is to show that we can take any given configuration C and modify it into C_M for some M while at each step increasing the expected cost.

8. Acknowledgements

I would like to thank my research mentor Olga Medrano for her guidance and for introducing me to this topic. I would also like to thank Professor Peter May for organizing the REU program that made this paper possible.

References

[1] Sanjoy Dasgupta, Nave Frost, Michal Moshkovitz, and Cyrus Rashtchian. Explainable k-means and k-medians clustering. In *Proceedings of the 37th International Conference on Machine Learning*, Vienna, Austria, pages 12–18, 2020.

[2] Hossein Esfandiari, Vahab Mirrokni, and Shyam Narayanan. Almost tight approximation algorithms for explainable clustering. In *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA* 2022. SIAM, 2022.

[3] Buddhima Gamlath, Xinrui Jia, Adam Polak, and Ola Svensson. Nearly-Tight and Oblivious Algorithms for Explainable Clustering. In *Proceedings of the 35th Conference on Neural Information Processing Systems*, NeurIPS 2021, volume 34, pages 28929-28939, 2021.

[4] Konstantin Makarychev and Liren Shan. Near-optimal algorithms for explainable k-medians and k-means. *In Proceedings of the 38th International Conference on Machine Learning, ICML 2021*, volume 139 of *Proceedings of Machine Learning Research*, pages 7358–7367. PMLR, 2021.

[5] Shai Shalev-Shwartz and Shai Ben-David. Chapter 22. In Understanding Machine Learning: From Theory to Algorithms, 307–20. Cambridge: Cambridge University Press, 2014.