QUERY K-MEANS CLUSTERING FOR CROWDSOURCING

BY

CHAO PAN

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2019

Urbana, Illinois

Adviser:

Professor Olgica Milenkovic

ABSTRACT

This thesis focuses on solving the K-means clustering problem approximately with side information provided by crowdsourcing. Both binary same-cluster oracle and general crowdsourcing framework are considered. It can be shown that, under some mild assumptions on the smallest cluster size, one can obtain a $(1+\epsilon)$-approximation for the optimal potential with probability at least $1-\delta$, where $\epsilon > 0$ and $\delta \in (0,1)$, using an expected number of $O(\frac{K^3}{\epsilon\delta})$ noiseless same-cluster queries and comparison-based clustering of complexity $O(ndK + \frac{K^3}{\epsilon\delta})$; here, n denotes the number of points and d the dimension of space. Compared to a handful of other known approaches that perform importance sampling to account for small cluster sizes, the proposed query technique reduces the number of queries by a factor of roughly $O(\frac{K^6}{\epsilon^3})$, at the cost of possibly missing very small clusters. This setting is extended to the case where some queries to the oracle produce erroneous information, and where certain points, termed outliers, do not belong to any clusters. Incorporating state-of-the-art results in crowdsourcing can further improve the performance of the algorithm. Note that the proof techniques used in this thesis differ from previous methods used for K-means clustering analysis, as they rely on estimating the sizes of the clusters and the number of points needed for accurate centroid estimation and subsequent nontrivial generalizations of the double Dixie cup problem. The performances of proposed algorithms are illustrated on both synthetic and real datasets, including MNIST and CIFAR 10.

To my parents, Shunhua Chen and Ming Pan, and my girlfriend, Selina Zhang, for their love and support.

ACKNOWLEDGMENTS

I would like express my gratitude to everybody who has, directly or indirectly, had a part in helping this thesis come to be. First and foremost, I would like to thank my wonderful adviser, Olgica Milenkovic, who has supported me, helped me and guided me during these two years. She encouraged me to explore in several different research areas, and provided me with valuable advice both in research and life. Thanks to our entire machine learning group, especially Eli Chien, Jianhao Peng, Pan Li, Abhishek Agarwal, and Srilakshmi Pattabiraman. We came up with so many great ideas in those meetings and brainstorming sessions. Finally, I would like to thank all staff members in the Coordinated Science Laboratory and the Department of Electrical and Computer Engineering for the clean working environment and the endless delicious snacks.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Background and Related Work

K-means clustering is one of the most studied unsupervised learning problems [1, 2, 3], with a rich application domain spanning areas as diverse as lossy source coding and quantization [4], image segmentation [5] and community detection [3]. The core question in K-means clustering is to find a set of K centroids that minimizes the K-means potential function, equal to the sum of the squared distances of the points from their closest centroids. An optimal set of centroids can be used to partition the points into clusters by simply assigning each point to its closest centroid.

It is well established that the K-means clustering problem is NP-hard even for the case when $K = 2$, or when the points lie in a two-dimensional Euclidean space [6]. Moreover, finding a $(1 + \epsilon)$-approximation for $0 < \epsilon < 1$ remains NP-hard, unless further assumptions are made on the point and cluster structures [7, 8]. Among state-of-the-art K-means approximation methods are the algorithms of Kanungo et al. [9] and Ahmadian et al. [10]. There also exist many heuristic algorithms for solving the problem, including Lloyd's algorithm [2] and Hartigan's method [1].

An interesting new direction in K-means clustering was recently initiated by Ashtiani et al. [11] who proposed to examine the effects of side-information on the complexity of the K-means algorithm. In their semi-supervised active clustering framework, one is allowed to query an oracle whether two points from the dataset belong to the same optimal cluster. The oracle answer to queries involving any pair of points is assumed to be consistent with a unique optimal solution, and it takes the form "same (cluster)" and "different (cluster)". The method of Ashtiani et al. [11] operates on special cluster structures which satisfy the so-called γ-margin assumption with $\gamma > 1$, which asserts that every point is at least a γ-factor closer to its corresponding centroid than any other centroid. The oracle queries are noiseless and $O(K \log n + K^2 \frac{\log K + \log(\frac{1}{\delta})}{(\gamma-1)^4})$ same-cluster queries on n points are needed to ensure that with probability at least $1-\delta$, the obtained partition is the sought optimal solution. Ailon et al. [12] proposed to dispose of the γ-margin assumption and exact clustering requirements, and addressed the issue of noisy same-cluster queries in the context of the K-means++ algorithm. In their framework, each pairwise query may return the wrong answer with some prescribed probability, but repeated queries on the same pair of points always produce the same answers. Given that no constraints on the cluster sizes and distances of points are made, one is required to perform elaborate nonuniform probabilistic sampling and subsequent selection of points that represent uniform samples in the preselected pool. This two-layer sampling procedure results in a large number of noiseless and noisy queries - in the former case, with running time of the order of $O(\frac{ndK^9}{\epsilon^4})$ - and may hence be impractical whenever the number of clusters is large, the smallest cluster size is bounded away from one, and the queries are costly and available only for a small set of pairs of points. Further extensions of the problem include the work of Gamlath et al. [13] that provides a framework for ensuring small clustering error probabilities via PAC (probably approximately correct) learning, and the weak-oracle analysis of Kim and Ghosh which allows for “do not know” answers [14].

In practice this required side information can come from online crowdsourcing services like Amazon Mechanical Turk. However, all the oracles mentioned above are binary, which involve only two samples in each event of query (denoted as same-cluster queries in later context). A more natural setting in crowdsourcing assumes that there are w workers, s samples and K classes (denoted as crowdsourcing queries in later context). Each worker labels a sample as one of K categories directly, and the query complexity is measured by $w \times s$. Note that the labels collected by crowdsourcing can also be of low quality because workers are often non-experts and sometimes unreliable. Dawid and Skene [15] developed a maximum likelihood approach based on the EM algorithm to infer true labels from noisy but redundant worker labels. Furthermore, Zhang et al. [16] proposed a provably optimal spectral method for initialization of the EM algorithm. They showed that true labels for each sample can be exactly recovered with high probability when w and s satisfy some mild conditions.

1.2 Contributions

Unlike other semi-supervised approaches proposed for K-means clustering, we address the problem in the natural setting where the size of the smallest cluster is bounded from below by a small value dependent on the number of clusters K and the approximation constant ϵ, and where the points contain outliers. Hence, we do not require that the clusters satisfy the γ-margin property, nor do we insist on being able to deal with very small clusters that seldom appear in practice. Bounding the smallest cluster size is a prevalent analytical practice in clustering, community detection and learning on graphs [17, 18, 19]. Often, K-means clustering methods are actually constrained to avoid solutions that produce empty or small clusters as these are considered to be artifacts or consequences of poor local minima solutions [20]. Let $\alpha = (\frac{n}{K s_{\min}})$, $1 \leq \alpha \leq \frac{n}{K}$, denote the cluster size imbalance, where $s_{\min}$ equals the size of the smallest cluster in the optimal clustering; when $\alpha = 1$, all clusters are of the same size $\frac{n}{K}$. Furthermore, when the upper bound is met, the size of the smallest cluster equals one.

As for outliers, in same-cluster queries they are defined as points at "large" distance from all clusters, for which all queries return negative answers and hence add additional uncertainty regarding point placements; in the setting of generalized crowdsourcing they are defined as samples that are not labelled by workers, which may happen frequently in practice.

Our main results are summarized below.

Theorem 1.1 (Query complexity with noiseless same-cluster queries)**.** Assume that one is given parameters $\epsilon \in (0,1)$, $\delta \in (0,1)$ and K, and n points in $\mathbb{R}^d$. Furthermore, assume that the unique optimal clustering has imbalance α, where $\alpha \in [1, \frac{n}{K}]$. Then, there exists a same-cluster query algorithm with an expected number of queries $O\left(\frac{\alpha K^3}{\epsilon\delta}\right)$ that with probability at least $1-\delta$ outputs a set of cluster centers whose corresponding clustering potential function is within a multiplicative factor $(1+\epsilon)$ of the optimal. The expected running time of the query-based clustering algorithm equals $O(Kdn + \alpha\frac{K^3}{\epsilon\delta})$.

Theorem 1.2 (Query complexity with noisy same-cluster queries and outliers)**.** Assume that one is given parameters $\epsilon \in (0,1)$, $\delta \in (0,1)$ and K, and n points in $\mathbb{R}^d$. Let p_o be the fraction of outliers in the dataset. Furthermore, assume that the unique optimal clustering without outliers has imbalance α, where $\alpha \in [1, \frac{n}{K}]$, and that the oracle may return an erroneous answer with probability $p_e < 1/2$. When presented with a query involving at least one outlier point, the oracle always produces the answer "different (cluster)". Then, there exists a noisy same-cluster query algorithm that requires

$$O\left(\frac{\alpha K^4}{\delta\epsilon\,(1-p_o)(1-2p_e)^8}\log^2\frac{\alpha K^2}{\delta\,(2p_e-1)^4\,(1-p_o)}\right)$$

queries and with probability at least $1-\delta$ outputs clusters whose corresponding clustering potential function is within $(1+\epsilon)$ of the optimal. The expected complete running time of the noisy clustering algorithm is bounded from above by $O\left(Kdn + \frac{\alpha K^6}{\delta\epsilon(1-p_o)(2p_e-1)^{10}}\log^3\frac{\alpha K^2}{\delta\epsilon(2p_e-1)^4(1-p_o)}\right)$, provided that the outliers satisfy a mild separability constraint (see Chapter 2 for more details).

Theorem 1.3 (Query complexity with noisy crowdsourcing queries)**.** Assume that one is given parameters $\epsilon \in (0,1)$, $\delta \in (0,1)$ and K, and n points in $\mathbb{R}^d$. Let $p_e \in [\rho, 1-\rho]$ be the probability that a worker mislabels a sample. Furthermore, assume that the unique optimal clustering has imbalance α, where $\alpha \in [1, \frac{n}{K}]$. Let $\kappa = \left|1 - p_e - \frac{1}{K}\right|$, $\bar{D} = \frac{K-1-Kp_e}{K-1}\log\frac{(K-1)(1-p_e)}{p_e}$. Then, there exists a crowdsourcing algorithm with w workers and s items sampled uniformly at random (without replacement) from whole dataset satisfying

$$\begin{aligned} w &= \Omega\left(\frac{\log(1/\rho)\log(Ks/\delta) + \log ws}{\bar{D}}\right) \\ s &= \Omega\left(\frac{\log w/\sqrt{\delta}}{\kappa^6\min\{\kappa^2, \rho^2, (\rho\bar{D})^2\}} + \frac{\alpha K^2}{\delta\epsilon} + \alpha K\log\frac{K}{\delta}\right), \end{aligned}$$

which can output a set of cluster centers whose corresponding clustering potential function is within a multiplicative factor $(1+\epsilon)$ of the optimal with probability at least $1-\delta$. The required query complexity is $w \times s$.

Note that Theorem 1.1 gives performance guarantees in expectation. Nevertheless, a straightforward application of Markov's inequality and the union bound allow us to also bound, with high probability, the query complexity.

Also Theorem 1.3 considers the case without outliers for simplicity. However, the algorithm can be easily adapted to cases with outliers by proper normalization [16].

In noiseless binary oracle setting, we conclude that using $O\left(\frac{\alpha K^3}{\delta\epsilon}\right)$ queries, with probability at least $1-\delta$ our clustering produces a $(1+\epsilon)$-approximation. For example, by choosing $\delta = 0.01$, we guarantee that, with probability at least 0.99, the query complexity of our noiseless method equals $O(\frac{\alpha K^3}{\epsilon})$. Compared to the result of Ailon et al. [12], as long as $s_{\min} \geq \frac{n\epsilon^3}{K^7}$, our method is more efficient than the two-level sampling procedure of [12]. The efficiency gap increases with $s_{\min}$. As an illustrative example, let $n = 10^6$, $K = 10$ and $\epsilon = 0.1$. Then, the minimum cluster size constraint only requires the smallest cluster to contain at least one point (since $\frac{n\epsilon^3}{K^7} = 10^{-4} < 1$).

Our proof techniques rely on novel generalizations of the double Dixie cup problem [21, 22]. Similarly to Ailon et al. [12], we make use of Lemma 2 from [23] described in Chapter 2. But unlike the former approach, which first performs K-means++ sampling and then subsampling that meets the conditions of Lemma 2, we perform a one-pass sampling. Given the smallest cluster size constraint, it is possible to estimate during the query phase the number of points one needs to collect from each cluster so as to ensure a $(1+\epsilon)$-approximation for all the estimated centroid. With this information at hand, queries are performed until each cluster (representing a coupon type) contains sufficiently many points (coupons). The double Dixie cup problem pertains to the same setting, and asks for the smallest number of coupons one has to purchase in order to collect s complete sets of coupons. The main technical difficulty arises from the fact that the number of coupons required is represented by the expected value of the maximum order statistics of random variables distributed according to the Erlang distribution [22], for which asymptotic analysis is hard when the number of types of coupons is not a constant. In our setting, the number of types depends on K, and the number of coupons purchased cannot exceed n. To address this issue, we use Poissonization methods [24] and concentration inequalities. Detailed proofs are relegated to Appendix A.

For the case of noisy same-cluster queries and outliers, our solution consists of two steps. In the first step, we invoke the results of Mazumdar and Saha [25, 26] that describe how to reconstruct all clusters of sufficiently large sizes when using similarity matrices of stochastic block model [27] along with same-cluster queries. The underlying modeling assumption is that every query can be wrong independently from all other queries with probability p, and that we cannot repeatedly ask the same query and apply majority voting to decrease the error probability, as each query response is fixed. In the second step, we simply compute the cluster centers via averaging.

In the given context, we only need to retrieve a fraction of the cluster points correctly. Note that the minimum cluster size our algorithm can handle is constrained both in terms of sampling complexity of the double Dixie cup as well as in terms of the cluster sizes that [26] can handle. Additional issues arise when considering outliers, in which case we assume the oracle always returns a negative answer ("different clusters"). Note that if the first point queried is an outlier, the seeding procedure may fail as an answer of the form "different clusters" may cause outliers to be placed into valid clusters. To mitigate this problem, we propose a simple search and comparison scheme which ensures that the first point assigned to any cluster is not an outlier.

For crowdsourcing setting, our algorithm can also be divided into two parts. Firstly, we use labels inferred from crowdsourcing data to fill in each cluster with the least number we need. Then we can output centers of already classified samples as estimates of true centers. Since the results in [16] shows that one can recover all true labels for sampled data with high probability under some mild assumptions, we use it as an initialization step of our algorithm.

We experimentally tested the proposed algorithms on synthetic and real datasets in terms of the approximation accuracy for the potential function, query complexity and the misclassification ratio, equal to the ratio of the number of misclassified data points and the total number of points. Note that misclassification errors arise as the centroids are only estimates of the true centroids, and placements of points according to closest centroids may be wrong. Synthetic datasets are generated via Gaussian mixture models, while the real world datasets pertain to image classification with crowdsourced query answers, including the MNIST [28] and CIFAR-10 [29] datasets. The results show order-of-magnitude performance improvements compared to other known techniques.

A few comments are in order. The models studied in [11, 26] are related to our work through the use of query models for improving clustering. Nevertheless, Ashtiani et al. [11] only consider ground truth clusters satisfying the γ-margin assumption, and K-means clustering with perfect (noiseless) queries. The focus of the work by Mazumdar et al. [26] is on the stochastic block model, and although it allows for noisy queries it does not address the K-means problem directly. The two models most closely related to ours are Ailon et al. [12] and Kim and Ghosh [14]. Ailon et al. [12] focus on developing approximate K-means algorithms with noisy same-cluster queries. The three main differences between this line of work and ours are that we impose mild smallest cluster size constraints which significantly reduce the query complexity in both noiseless and noisy regimes, that we introduce outliers into our analysis, and that our proofs are based on a variation of the double Dixie cup problem rather than standard theoretical computer science analyses that use notions of covered and uncovered clusters. The work of Kim and Ghosh [14] is related to ours only insofar as it allows for query responses of the form “do not know” which can also be used for dealing with outliers.

This work about Query K-means clustering was previously published in NeurIPS 2018 [30] and is adapted here with permission.

CHAPTER 2

PROBLEM FORMULATION

We start with a formal definition of the K-means problem. Given a set of n points $\mathcal{X} \subset \mathbb{R}^d$, and the number of clusters K, the K-means problem asks for finding a set of points $\mathbf{C} = \{c_1, ..., c_K\} \subset \mathbb{R}^d$ that minimizes the following objective function

$$\phi(\mathcal{X}; \mathbf{C}) = \sum_{x \in \mathcal{X}} \min_{c \in \mathbf{C}} ||x - c||^2,$$

where $|| \cdot ||$ denotes the L_2 norm. Throughout the thesis, it is assumed that the optimal solution is unique, and we denote it by $\mathbf{C}^* = \{c_1^*, ..., c_K^*\}$. The set of centroids $\mathbf{C}^*$ induces an optimal partition $\mathcal{X} = \bigcup_{i=1}^K \mathcal{C}_i^*$, where $\forall i \in [K], \mathcal{C}_i^* = \{x \in \mathcal{X} : ||x - c_i^*|| \leq ||x - c_j^*|| \; \forall j \neq i\}$. We use $\phi_K^*(\mathcal{X})$ to denote the optimal value of the objective function.

As already stated, the K-means clustering problem is NP-hard, and hard to approximate within a $(1+\epsilon)$ factor, for $0 < \epsilon < 1$. An important question in the approximate clustering setting was addressed by Inaba et al. [23], who showed how many points from a set have to be sampled uniformly at random to guarantee that, for any $\epsilon > 0$ and with high probability, the centroid of the set can be estimated within a multiplicative $(1+\epsilon)$-term. This result was used by Ailon et al. [12] in the second (sub)sampling procedure. In our work, we make use of the same result in order to determine the smallest number of points (coupons) one needs to collect for each cluster (coupon type). For completeness, the result is stated below.

Lemma 2.1 (Centroid lemma, Lemma 2 of [23])**.** Let $\mathcal{A}$ be a set of points obtained by sampling with replacement m points independently from each other, uniformly at random, from a point set $\mathcal{S}$. Then, for any $\delta > 0$, one has

$$P(\phi(\mathcal{S}; c(\mathcal{A})) \leq (1 + \frac{1}{\delta m})\phi^*(\mathcal{S})) \geq 1 - \delta,$$

where $c(\mathcal{A})$ stands for the centroid of $\mathcal{A}$.

In our proof, the Centroid lemma is used in conjunction with a generalization of the double Dixie cup problem to establish the stated query complexity results in the noiseless and noisy setting. The double Dixie cup problem is an extension of the classical coupon collector problem in which the collector is required to collect $m \geq 2$ sets of coupons. While the classical coupon collector problem may be analyzed using elementary probabilistic tools, the double Dixie cup problem solution requires using generating functions and complex analysis techniques. For the most basic incarnation of the problem where each coupon type is equally likely and each coupon needs to be collected at least m times, where m is a constant, Newman and Shepp [21] showed that one needs to purchase an average of $O(K(\log K + (m-1)\log\log K))$ coupons. This setting is inadequate for our analysis, as our coupons represent points from different clusters that have different sizes, and hence give rise to different coupon (cluster point) probabilities. Furthermore, in our analysis we require $m = \frac{K}{\delta\epsilon}$, which scales with K and hence is harder to analyze. The starting point of our generalization of the nonuniform probability double Dixie cup problem is the work of Doumas and Papanicolaou [22]. We extend the Poissonization argument and perform a careful analysis of the expectation of the maximum order statistics of independent random variables distributed according to the Erlang distribution. All technical details are delegated to Appendix A.

Often, one seeks the K-means solutions in a setting where the cluster points $\mathcal{X}$ satisfy certain separability and cluster size constraints, such as the γ-margin and the bounded minimum cluster size constraint, respectively. Both are formally defined below.

Definition 2.1 (The γ-margin property [11])**.** Let $\gamma > 1$ be a real number. We say that $\mathcal{X}$ satisfies the γ-margin property if $\forall\, i \neq j \in [K]$, $x \in \mathcal{C}_i^*$, $y \in \mathcal{C}_j^*$, one has

$$\gamma||x - c_i^*|| < ||y - c_i^*||.$$

To describe the cluster size constraint, we now formally introduce the previously mentioned notion of α-imbalance.

Definition 2.2 (The α-imbalance property)**.** Let $\alpha \in [1, n/K]$ be a real number. We say that the point set $\mathcal{X}$ satisfies the α-imbalance property if $\alpha = \frac{n}{K\, s_{\min}}$.

To avoid complicated and costly two-level queries, we impose an α-imbalance constraint on the optimal clustering, excluding outliers.

For the set of outliers, we use a milder version of the γ-margin constraint, described as follows. Assume that $\mathcal{X} = \mathcal{X}_t \cup \mathcal{X}_o$, where $\mathcal{X}_t$ and $\mathcal{X}_o$ are the nonintersecting sets of true cluster points and outliers, respectively. Outliers are formally defined as follows.

Definition 2.3. The set $\mathcal{X}_o$ consists of points that satisfy the $\Gamma(\xi)$-separation property, defined as

$$\forall x \in \mathcal{X}_o,\ \forall\ i \in [K],\ ||x - c_i^*|| > \max_{y \in \mathcal{C}_i^*} ||y - c_i^*|| + \sqrt{\frac{\xi\, \phi^*(\mathcal{C}_i^*)}{|\mathcal{C}_i^*|}} \geq \Gamma(\xi).$$

Here, $\Gamma(\xi)$ stands for the minimum of the lower bounds obtained for all values of $i \in [K]$.

This is a reasonable modeling assumption, as outliers are commonly defined as points that lie in "outlier clusters" that are well-separated from all "regular" clusters. The definition is reminiscent of the γ-margin assumption, but adapted to outliers. Note that the second term serves as a scaled proxy for the empirical standard deviation of the average distance between cluster points and their centroids. In this extended setting, the objective is to minimize the function $\phi(\mathcal{X}_t, \mathbf{C})$. Furthermore, with a slight abuse of notation, we use $\mathcal{C}_1^*, ..., \mathcal{C}_K^*$ to denote the optimal partition for both $\mathcal{X}_t$ and $\mathcal{X}$. It should be clear from the context which clusters are referred to.

Side information for the K-means problem is provided by a query oracle $\mathcal{O}$ such that

$$\forall x_1, x_2 \in \mathcal{X},\ \mathcal{O}(x_1, x_2) = \begin{cases} 0, & \text{if } \exists i\ \in [K] \text{ s.t. } x_1 \in \mathcal{C}_i^*, x_2 \in \mathcal{C}_i^*; \\ 1, & \text{otherwise.} \end{cases} \tag{2.1}$$

Query complexity is measured in terms of the number of times that an algorithm requests access to the oracle. The goal is to devise query algorithms with query complexity as small as possible. The noisy oracle $\mathcal{O}_n$ may be viewed as the response of a binary symmetry channel with parameter p_e to an input produced by a noiseless oracle $\mathcal{O}$. Equivalently, $\forall x_1, x_2 \in \mathcal{X}$, $P(\mathcal{O}_n(x_1, x_2) = \mathcal{O}(x_1, x_2)) = 1 - p_e$, and $P(\mathcal{O}_n(x_1, x_2) \neq \mathcal{O}(x_1, x_2)) = p_e$, independently from other queries. Each pair (x_1, x_2) is queried only once, and the noisy oracle $\mathcal{O}_n$ always produces the same answer for the same query. When presented with at least one outlier point in the pair (x_1, x_2), the noiseless oracle always returns $\mathcal{O}(x_1, x_2) = 1$, while the noisy oracle $\mathcal{O}_n$ may flip the answer with probability p_e. The problem of identifying outliers placed in regular clusters is resolved by invoking the algorithm of [26], which places outliers into small clusters that are expurgated from the list of valid clusters.

In the setting of crowdsourcing, side information for K-means problem comes from workers instead of a given oracle. Suppose that there are w workers, s items and K classes. The true label y_j of item j follows some underlying distribution which is unknown. Denote $z_{ij} \in \mathbb{R}^K$ as the label that worker i assigns to item j, which is a one-hot vector. Let π_i be the probability that worker i is able to classify an item into one of those K categories. If not, $z_{ij} = \mathbf{0}$ is considered as an outlier. Workers may also make mistakes during the process. It is assumed that the probability μ_{ilc} that worker i labels an item in class l as class c is independent of any particular samples. Define $C_i = [\mu_{ilc}]_{l\in[K],c\in[K]} \in \mathbb{R}^{K\times K}$ as the confusion matrix of worker i. For fair comparison with the query oracle mentioned above, we considered a special case, where μ_{ilc} is a constant for any $l \neq c$ and C_i's are the same for all $i \in [w]$. In this case, if we define p_e as the probability that a worker mislabel an item, all the diagonal elements of C_i are $1 - p_e$ and the other elements are $\frac{1-p_e}{K-1}$. Note that this is a simplified version of the *one-coin* model in crowdsourcing, which can be solved nearly optimally by the algorithm of [16].

CHAPTER 3

ALGORITHMIC SOLUTIONS

In this chapter, we present three algorithms for different settings. In the process, we sketch some of the proofs establishing the theoretical performance guarantees of our methods.

3.1 Approximate Noiseless Same-Cluster Query K-means Clustering

Algorithm 1: Approximate Noiseless Same-Cluster Query K-means Clustering

```
Input: A set of $n$ points $\mathcal{X}$, number of clusters $K$, an oracle $\mathcal{O}$
Output: Estimates of the centers $\mathcal{C}$
Initialization: $t = 1,\ \mathcal{C}_i = \emptyset,\ R_i = \emptyset,\ \forall i \in [K]$.
Uniformly at random sample a point $x$ from $\mathcal{X}$, $\mathcal{C}_1 \leftarrow \mathcal{C}_1 \cup \{x\}$,
 $R_1 \leftarrow x$.
while $\min_{i\in[K]} |\mathcal{C}_i| < \frac{K}{\delta\epsilon}$ do
    Uniformly at random sample with replacement a point $x$ from $\mathcal{X}$.
    if $\forall i \in [t],\ \mathcal{O}(R_i, x) = 0$ then
        $\mathcal{C}_i \leftarrow \mathcal{C}_i \cup \{x\}$.
    else
        $t \leftarrow t + 1$, $\mathcal{C}_t \leftarrow \{x\}$, $R_t \leftarrow x$.
    end
end
for $k = 1$ to $K$ do
    Let $c_{k,i}$ denote the $i^{th}$ element added to $\mathcal{C}_k$, $\mu_k = \frac{1}{|\mathcal{C}_k|}\sum_{i=1}^{S_k} c_{k,i}$,
     $\mathbf{C} \leftarrow \mathbf{C} \cup \{\mu_k\}$.
end
```

The noiseless same-cluster Query K-means algorithm is conceptually simple and it consists of two steps. In the first step, we sample and query pairs of points until we collect at least $\frac{K}{\delta\epsilon}$ points for each of the K clusters. In the second step, we compute the centroids of clusters by using the queried and classified points. The number of points to be collected is dictated by the size of the smallest cluster and the double Dixie cup coupon collector's requirements derived in Appendix A, and summarized below.

Lemma 3.1. Assume that there are K types of coupons and that the smallest probability of a coupon type $p_{\min}$ is lower bounded by $\frac{1}{\alpha K}$, with $\alpha \in [1, \frac{n}{K}]$. Then, on average, one needs to sample at most

$$2\alpha K(\log K + m \log 2)$$

coupons in order to guarantee the presence of at least m complete sets, where $m = O(K)$.

Note that in our analysis, we require that $m = \frac{K}{\delta\epsilon}$, for some $\epsilon, \delta > 0$, while classical coupon collection and Dixie cup results are restricted to using constant m [22, 21]. In the latter case, the number of samples equals $O(K(\log K + (m-1)\log\log K))$, which significantly differs from our bound.

Two remarks are in order. First, one may modify Algorithm 1 to enforce a stopping criterion for the sampling procedure. Furthermore, when performing pairwise oracle queries, we assumed that in the worst case, one needs to perform K queries, one for each cluster. Clearly, one may significantly reduce the query complexity by choosing, at each query time, to first probe the clusters with estimated centroids closest to the queried point.

3.2 Approximate Noisy Same-Cluster Query K-means Clustering with Outliers

The steps of the algorithm for approximate same-cluster query-based clustering with noisy responses and outliers are listed in Algorithm 2. The gist of the approach is to assume that outliers create separate clusters that are filtered out using the noisy-query clustering method of [26]. Unfortunately, the aforementioned method assumes that sampling is performed without replacement, which in our setting requires that we modify the Centroid lemma to account for sampling points uniformly at random without replacement. This modification is described in Lemma 3.2.

Lemma 3.2 (The Modified Centroid Lemma)**.** Let $\mathcal{S}$ be a set of points obtained by sampling m points uniformly at random *without replacement* from a point set $\mathcal{A}$. Then, for any $\delta > 0$, with probability at least $1-\delta$, one has

$$\phi(\mathcal{A}; c(\mathcal{S})) \leq \left(1 + \frac{1 - \frac{m-1}{|\mathcal{A}|-1}}{\delta m}\right) \phi_1^*(\mathcal{A}) \leq \left(1 + \frac{1}{\delta m}\right) \phi_1^*(\mathcal{A}).$$

Here, $c(\mathcal{S})$ denotes the center of mass center of $\mathcal{S}$, and $m \leq |\mathcal{A}|$.

Furthermore, the requirement that sampling is performed without replacement gives rise to a new version of the double Dixie cup coupon collection paradigm in which one is given only a limited supply of coupons of each type, with the total number of coupons being equal to n. As a result, the number of points sampled from each cluster without replacement can be captured by an i.i.d. multivariate hypergeometric random vector with parameters $(n, np_1, ..., np_K, m)$. To establish the query complexity results in this case, we need not to estimate the expected number of points sampled, but instead to ensure concentration results for hypergeometric random vectors. This is straightforward to accomplish, as it is well known that a hypergeometric random variable may be written as a sum of independent but nonidentically distributed Bernoulli random variables [31]. Along with tight bounds on the Kullback–Leibler divergence and Hoeffding's inequality [32], this leads to the following bound on the probability of sampling a sufficiently large number of points from the smallest cluster.

Theorem 3.1. Without loss of generality, assume that $p_1 \leq p_2 \leq \ldots p_K$, where $p_i \in (0,1)$ for all i, and $\sum_i p_i = 1$. Furthermore, assume that during the query procedure, M points from K nonuniformly sized clusters of sizes $(np_1, ..., np_K)$ are sampled uniformly at random, without replacement. Denote $S_i^\dagger$ as the number of samples filled in cluster i after this process. Then, the probability that each cluster is filled with at least $\frac{Mp_1}{2}$ is bounded as

$$P\{\min S_i^\dagger \geq \frac{Mp_1}{2}\} \geq 1 - K \exp\left(-\frac{Mp_1}{8}\right). \tag{3.1}$$

Recall that the oracle treats outliers as points that do not belong to the optimal clusters, so that in Algorithm 3, outliers are treated as singleton clusters. In this case, the minimum cluster size requirement from [26] automatically filters out all outliers. Nevertheless, nontrivial changes compared

```
Algorithm 2: Approximate Noisy Same-Cluster Query $K$-means
Clustering with Outliers
    Input: A set of $n$ points $\mathcal{X}$, the number of clusters $K$, a noisy oracle
           $\mathcal{O}_n$ with output error probability $p_e$, a precomputed value
           $M$, and probability $p_o$ of outliers
    Output: Centroids set $\mathbf{C}$
    Phase 1: Seed the clusters by running Algorithm 3 for noisy
             query-based clustering
    Uniformly at random sample $M$ points from $\mathcal{X}$ without replacement.
    The sampled set equals $\mathcal{A}$.
    Run Algorithm 3 on $\mathcal{A}$ to obtain a $K$-partition of $\mathcal{A} = \bigcup_{i=1}^{K} \mathcal{A}_i$.
    Phase 2: Estimate the centroids
    For all $i \in [K]$, $c_i \leftarrow c(\mathcal{A}_i)$ where $c(\mathcal{A}_i)$ is the center of mass of the
    set $\mathcal{A}_i$. $\mathbf{C} \leftarrow \{c_1, ..., c_K\}$.
```

to the noisy query algorithm derived from [26] are needed, as the presence of outliers changes the effective number of clusters. How to deal with this issue is described in Appendix A.

For completeness, we describe Algorithm 3 used in our main routine, and first proposed in [26]. The parameters and routines used in the algorithm are as follows: $N = \frac{64k^2 \log(n)}{(1-2p_e)^4}, c = \frac{16}{(1-2p_e)^2}$, and $T(a) = p_e a + \frac{6\sqrt{N \log(n)}}{(1-2p_e)}, \theta(a) = 2p_e(1-p_e)a+2\sqrt{N\log(n)}$, where K is the number of clusters, n is the number of data points and p_e is the error probability. For a weighted graph $G(V, E)$, we let $N^+(u)$ denote all the neighbors of u in V' that are connected with u by a $+1$ weight edge.

3.3 Approximate Noisy Crowdsourcing Query K-means Clustering

The steps of the algorithm for approximate crowdsourcing-based Query K-means clustering are listed in Algorithm 4 which, similarly to Algorithm 2, uses Algorithm 5 proposed in [16] as the initialization strategy. The performance guarantee for Algorithm 4 is shown in Theorem 1.3, and is explained in Appendix A in more detail.

The parameters and routines used in Algorithm 5 are as follows: for every two workers a and b, denote $N_{ab} = \frac{K-1}{K}\left(\frac{\sum_{j=1}^{s} \mathbf{1}(z_{aj}=z_{bj})}{s} - \frac{1}{K}\right)$ as their

```
Algorithm 3: Clustering with a Noisy Same-Cluster Oracle $\mathcal{O}_n$
Input   : A set of $n$ points $V$, the number of clusters $K$, a noisy
          oracle $\mathcal{O}_n$ and the error probability parameter $p_e$
Output: All clusters in the set active, i.e., clusters of size at least
          $\Omega(\frac{k\log(n)}{(1-2p_e)^4})$
The Main Algorithm:
Initialization: Start with an empty graph $G' = (V', E')$, with all
vertices in $V$ unassigned. The cluster set active is empty.
Phase 1: Selection of a small subgraph
Add vertices uniformly at random chosen from the unassigned
vertices in $V \backslash V'$ to $V'$, ensuring that the size of $V'$ is $N$. If there are
not sufficiently many vertices left in $V \backslash V'$ to add to $V'$, add all of
$V \backslash V'$.
Update the weights for $G(V', E')$ by querying the oracle. For each
pair of vertices $(u, v)$, set $w(u, v) = +1$ if the answer is "yes" and $-1$
otherwise.
Phase 2: Active cluster identification
for each pair $(u, v)$ in $V'$ and $u \neq v$ do
    if $|N^+(u)| \geq T(|V'|)$ and $|N^+(v)| \geq T(|V'|)$ and
    $|N^+(u) \triangle N^+(v)| \leq \theta(|V'|)$ then
        Place $u, v$ into the same cluster.
end
Include all clusters formed in this step that have size at least $N/k$.
Remove all vertices in such clusters from $V'$ and any edge incident
on them from $E'$.
Phase 3: Growth of the active cluster set
for every unassigned vertex $v \in V \backslash V'$ do
    for every cluster $\mathcal{C} \in$ active do
        Randomly pick $c\log(n)$ distinct vertices from $\mathcal{C}$ and query $v$
        with them.
        if the majority answers are yes then
            include $v$ in $\mathcal{C}$.
            Break the loop and continue to another unassigned vertex.
    end
end
If there are still points in $V \backslash V'$, move to Phase 1 to obtain the
remaining clusters.
```

```
Algorithm 4: Approximate Noisy Crowdsourcing Query $K$-means Clustering
Input: A set of $n$ points $\mathcal{X}$, the number of clusters $K$, a noisy
       crowdsourcing service with error probability of workers $p_e$,
       and precomputed values $w$ and $s$
Output: Centroids set $\mathbf{C}$
Phase 1: Seed the clusters by running Algorithm 5 for noisy
         crowdsourcing label inference
Uniformly at random sample $s$ points from $\mathcal{X}$ without replacement.
The sampled set equals $\mathcal{A}$.
Run Algorithm 5 on $\mathcal{A}$ with $w$ workers to obtain a $K$-partition of
$\mathcal{A} = \bigcup_{i=1}^{K} \mathcal{A}_i$.
Phase 2: Estimate the centroids
For all $i \in [K]$, $c_i \leftarrow c(\mathcal{A}_i)$ where $c(\mathcal{A}_i)$ is the center of mass of the
set $\mathcal{A}_i$. $\mathbf{C} \leftarrow \{c_1, ..., c_K\}$.
```

similarity. And for every worker i, a pair of other workers (a_i, b_i) is defined as $(a_i, b_i) = \arg\max_{(a,b)} \{|N_{ab}| : a \neq b \neq i\}$. Note that original algorithm for the *one-coin* model in [16] assumes different error probabilities for different workers, while in our case the estimate of error probabilities are constrained to be the same by averaging at each iteration. For consistency with the notation in [16], we denote $p_c = 1 - p_e$ as the accuracy of workers.

Algorithm 5: Inferring True Labels from Noisy Crowdsourcing Queries

```
Input: The number of clusters $K$, the number of workers $w$, the
       number of samples $s$, and observed labels $z_{ij} \in \mathbb{R}^K$ for
       $i \in [w]$ and $j \in [s]$
Output: Predicted labels $\widehat{y}_j$ for $j \in [s]$
Phase 1: Initialize estimate of $p_c$
Initialize $\widehat{p}_c$ by $\widehat{p}_c \leftarrow \frac{1}{w}\sum_{i=1}^{w}\left(\frac{1}{K} + \operatorname{sign}\left(N_{ia_1}\right)\sqrt{\frac{N_{ia_i}N_{ib_i}}{N_{a_ib_i}}}\right)$.
If $\widehat{p}_c < \frac{1}{K}$, then set $\widehat{p}_c \leftarrow \frac{2}{K} - \widehat{p}_c$.
Phase 2: Infer sample labels using EM algorithm
Iterate the following two steps until convergence, where $\widehat{q}_{jl}$ is the
estimate probability for sample $j \in [s]$ belonging to class $l \in [K]$:

    $\widehat{q}_{jl} \propto \exp\left(\sum_{i=1}^{w} \mathbf{1}\left(z_{ij} = e_l\right)\log\left(\widehat{p}_c\right) + \mathbf{1}\left(z_{ij} \neq e_l\right)\log\left(\frac{1-\widehat{p}_c}{K-1}\right)\right)$,
    $\widehat{p}_c \leftarrow \frac{1}{w}\sum_{i=1}^{w}\left(\leftarrow \frac{1}{s}\sum_{j=1}^{s}\sum_{l=1}^{K}\widehat{q}_{jl}\mathbf{1}\left(z_{ij} = e_l\right)\right)$.

Output $\widehat{y}_j = \arg\max_{l\in[K]}\left\{\widehat{q}_{jl}\right\}$ for $j \in [s]$.
```

CHAPTER 4

EXPERIMENTS

For fair comparison with state-of-the-art Query K-means algorithms, only results related with same-cluster oracles (Algorithm 1 and 2) are shown in this chapter.

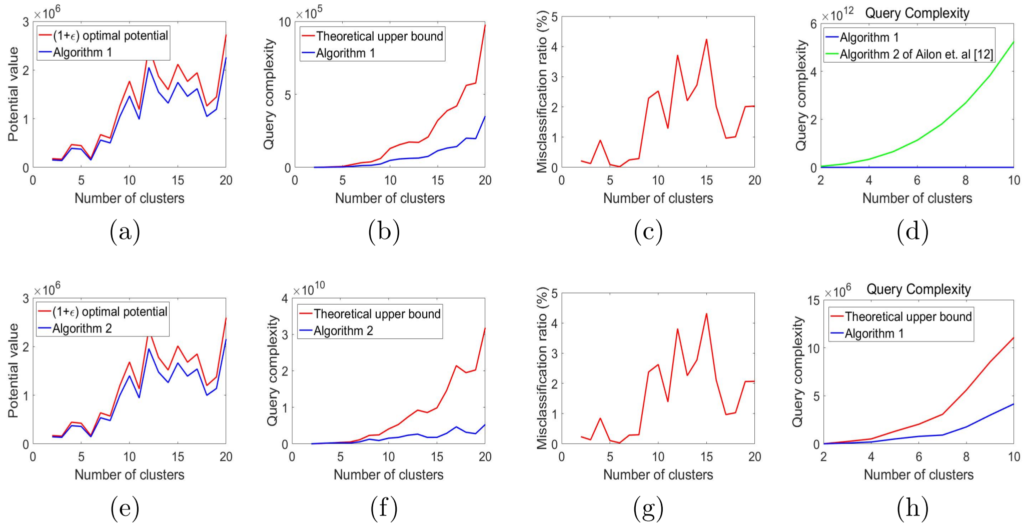
4.1 Synthetic Data

For our synthetic data experiments, we start by selecting all relevant problem parameters, the number of clusters K, the cluster imbalance α, the dimension of the point dataset d, the approximation factor ϵ and the error tolerance level δ. We uniformly at random sample K cluster centroids in the hypercube $[0,5]^d$ – this choice of the centroids allows one to easily control the overlap between clusters. Then, we generate n_i points for each cluster $i = 1, \ldots, K$, where the values $\{n_i\}_{i=1}^K$ are chosen so as to satisfy the α-imbalance property and so that $n_i \in [1000, 6000]$. The points in the cluster indexed by i are obtained by sampling d-dimensional vectors from a Gaussian distribution $\mathcal{N}(0, \sigma_i^2\, I)$, with I representing the $d \times d$ identity matrix, and adding these Gaussian samples to the corresponding cluster centroid. When generating outliers, we uniformly at random choose a subset of points of size $p_o \times n$, where n is the total number of points to be clustered. Then we adjust the positions of the points to make sure that they satisfy the $\Gamma(2)$-separation property, described in the previous sections. In the noisy oracle setting, we assume that the oracle produces the correct answer with probability $1 - p_e$, for $p_e \in \left(0, \frac{1}{2}\right)$.

We evaluated our algorithms with respect to three performance measures. The first measure is the value of the potential function. As all our algorithms are guaranteed to produce a $(1+\epsilon)$-approximation for the optimal potential, it is of interest to compare the theoretically guaranteed and actually obtained potential values. The second performance measure is the query complexity, for which we once again have analytic upper bounds. The third performance measure is the overall misclassification ratio, defined as the fraction of misclassified data points. We also compared our Algorithm 1 with the state-of-the art Algorithm 2 of [12] for the case when there exists one cluster containing one point only. Recall that [12] does not require the smallest cluster size to be bounded away from one, and may in principle operate more efficiently in settings where clusters of smallest possible size (one) exist. As will be seen from our simulation studies, even in this case, our method significantly outperforms [12].

The results of our experiments for the noiseless setting are shown in Figure 4.1. As may be seen, our analytic approximation results for the potential closely match the results obtained via simulations. In contrast, the actual query complexity is significantly lower in practice than predicted through our analysis, due to the fact that we assumed a worst case scenario for pairwise queries, and set the number of comparisons to K. For the misclassification ratio, we observe that the general trend is as expected – the larger the number of clusters K, the larger the misclassification ratio. Still, the misclassification error in all tested examples did not exceed 2.9%. From Figure 4.1-(d) we can see clearly that our method performs significantly better than Algorithm 2 in [12] even when α is fairly large. We did not compare our noisy query method with outliers with the noisy sampling method of [12] as the latter cannot deal with outliers.

Figure 4.1-(d) reveals that there exists a substantial gap between the query complexity of our method and that of [12] in the noiseless setting. For example, when $K = 5$ and $K = 10$, we require $510,932$ and 4.16×10^6 queries. In comparison, Algorithm [12] requires 6.55×10^{11} and 5.24×10^{12} queries, which in the latter case is roughly a five-order larger number of queries. As a matter of fact, the algorithm in [12] involves a very large constant in its complexity bound, equal to $\frac{2^{23} K^3}{\epsilon^4}$, which for practical clustering settings dominates the complexity expression.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4.1: Panels (a) to (c) and (e) to (g) list the results for synthetic data and the noiseless oracle Algorithm 1 and noisy oracle with outliers Algorithm 2, respectively. The parameters are $d = 20, K = [2 : 20]$, and $\alpha = [1, 6], \sigma_i = [0, 2], \delta = \epsilon = 0.2, p_o = p_e = 0.05$. Panels (a) and (e) plot the potential, (b) and (f) the query complexity, and (c) and (g) the misclassification ratio. Panels (d) and (h) provide comparisons with the noiseless Algorithm 2 of Ailon et al. [12] for a clustering problem with one cluster of size equal to one, with all cluster sizes in the range $[100, 600]$.

4.2 Real Data

Since the query complexity of our methods is independent of the size of the dataset, we can provide efficient solutions to large-scale crowdsourcing problems that can be formulated as K-means problems, such as is the case with image classification. We use the following two image classification datasets for which the ground-truth clusters are known and can hence be used to generate the outputs of both the noiseless and noisy oracle:

1) The well-known MNIST dataset [28] comprises $60,000$ training and $10,000$ test images of handwritten digits. Each image is normalized to fit into a 28×28 pixel bounding box and is anti-aliased, which results in grayscale levels.

2) The CIFAR-10 dataset [29] contains $60,000$ color images with 32×32 pixels, grouped into 10 different clusters of equal size, representing 10 different objects. The clusters are nonintersecting and we sampled $10,000$ cluster points.

Table 4.1: Real Datasets Results

	Actual query complexity	Theoretical query complexity
MNIST-Algorithm 1	12,195	38,868
MNIST-Algorithm 2	3,628,193,647	6,439,271,969
CIFAR 10-Algorithm 1	12,490	37,479
CIFAR 10-Algorithm 2	128,458,964	898,432,836

Here, we set $p_o = 0$ and $p_e = 0.05$, hence asserting that there are no outliers, but that 5% of the data points are mislabelled. Note that all the query complexities reported are needed to achieve a $(1 + \epsilon)$-approximation of the potential. The results are shown in Table 4.1.

CHAPTER 5

CONCLUSION

This thesis considers the problem of Query K-means clustering with both same-cluster and crowdsourcing queries, where same-cluster queries mean that one is given access to potentially noisy side information regarding pairs of points belonging to the same cluster or not. Our solution significantly improves upon the state-of-the-art results, showing that one can obtain a $(1+\epsilon)$-approximation for the optimal solution with probability at least $1-\delta$, where $\epsilon > 0, \delta \in (0,1)$, using $O(\frac{K^3}{\epsilon\delta})$ noiseless same-cluster queries in expectation and a comparison-based clustering algorithm of complexity $O(ndK+\frac{K^3}{\epsilon\delta})$ in expectation, where n is the number of points and d is the dimension of space. In contrast, previously reported results required $O(\frac{K^9}{\epsilon^4})$ same-cluster queries and with time complexity $O(\frac{ndK^9}{\epsilon^4})$, where they fix $\delta = 0.01$. Unlike previous approaches, we also focus on a new error model which more realistically captures missing information or outliers in datasets frequently used in crowdsourcing and biological network analysis. Also, our proof techniques differ from previous methods used for K-means clustering analysis, as they rely on estimating the sizes of the clusters and the number of points needed for accurate centroid estimation and subsequent nontrivial generalizations of the double Dixie cup problem.

APPENDIX A

PROOFS AND EXTENSIONS

A.1 Proof of Theorem 1.1

The bases of our proof are the Centroid lemma and a new problem in the area of double Dixie cup problems.

The Centroid lemma asserts that in order to obtain a $(1+\epsilon)$-approximation of the potential with probability at least $1-\delta$, one only needs to sample (with replacement) $m = \frac{1}{\delta\epsilon}$ points, which is a value independent of the size of set $\mathcal{A}$. This fact can be directly observed from the following equality:

$$\sum_{x \in \mathcal{A}} ||x - c(S)||^2 = \sum_{x \in \mathcal{A}} ||x - c(\mathcal{A})||^2 + |\mathcal{A}| \cdot ||c(S) - c(\mathcal{A})||^2. \tag{A.1}$$

The first term on the right-hand side is the optimal potential $\phi_1^*(\mathcal{A})$. The second term corresponds to the centroid estimation error. In order to obtain a $(1+\epsilon)$-approximation, we hence need $\epsilon\phi_1^*(\mathcal{A}) \geq |\mathcal{A}|||c(S) - c(\mathcal{A})||^2$. At first glance, it appears that the existence of a small set of points far removed from large clusters of points in $\mathcal{A}$ may cause the estimate of $c(\mathcal{A})$ to be highly imprecise as the sampling strategy is uniformly at random, and this small subset may never be sampled. However, whenever these assumptions are true, $\phi_1^*(\mathcal{A})$ itself is large and the error is within the required ϵ-margin.

Based on the above discussion, we need to sample (with replacement) points uniformly at random until each query cluster contains at least $\frac{K}{\delta\epsilon}$ points. Hence, by the Centroid lemma 2.1, the centroids estimated according to the collected points guarantee that for all $\mathcal{C}_i^*$, $i = 1, \ldots, K$, one has

$$P\{\phi(\mathcal{C}_i^*; \mathbf{C}) \leq (1 + \frac{1}{\delta m})\phi_1^*(\mathcal{C}_i^*)\} \geq 1 - \frac{\delta}{K}.$$

Invoking the union bound, we obtain

$$P\{\sum_{i=1}^{K} \phi(\mathcal{C}_i^*; \mathbf{C}) \leq \sum_{i=1}^{K}(1+\epsilon)\phi_1^*(\mathcal{C}_i^*)\} = P\{\phi(\mathcal{X}; \mathbf{C}) \leq (1+\epsilon)\phi^*(\mathcal{X})\} \geq 1-\delta.$$

Thus, Algorithm 1 ensures a $(1+\epsilon)$-approximation of the potential with probability at least $1-\delta$.

In the next step, we establish the number of required iterations of the query procedure. Note that in each iteration within the while loop, with probability $p_i = \frac{|\mathcal{C}_i^*|}{n}$ we sample a point from optimal cluster $\mathcal{C}_i^*$. The while loop terminates if we have at least $\frac{K}{\delta\epsilon}$ points from all K clusters. Clearly, this is an instance of the double Dixie cup coupon collector problem [21, 33, 22].

Let the random variable $T_K(m, \mathbf{p})$, where $m = \frac{K}{\delta\epsilon}$, equal the number of executed iterations of the algorithm. In the double Dixie cup setting, it equals the number of coupons purchased until each type of coupon is observed at least m times. The probability of sampling a coupon of type i equals p_i. From a slight modification of the analysis in [22] involving Poissonization techniques, we arrive at the following result:

$$\mathbb{E}[T_K(m,\mathbf{p})] = \mathbb{E}[\max_{i\in[K]}\{X_i\}], \; X_i\text{'s are independent,} \tag{A.2}$$

and distributed according to the Erlang distribution, $X_i \sim$ Erlang (m, λ_i), where $\lambda_i = \frac{1}{p_i}$. Recall that the Erlang$(m, \lambda_i)$ distribution makes probability mass assignments according to

$$P\{X_i = x\} = \frac{x^{m-1}\lambda_i^m}{(m-1)!}\exp(-\lambda_i x) = \frac{x^{m-1}}{p_i^m(m-1)!}\exp\left(-\frac{x}{p_i}\right). \tag{A.3}$$

An naive approach to upper bounding (A.2) is to replace the max value by the sum of all terms involved. However, this bound is very loose and we hence resort to a different approach.

For any $t \in (0, p^*)$, where $p^* = \frac{s_{min}}{n}$, we have

$$\begin{aligned}
\mathbb{E}[\max X_i] &= \mathbb{E}[\frac{1}{t} \log \exp(t \max X_i)] \leq \frac{1}{t} \log \mathbb{E}[\exp(t \max X_i)] \\
&= \frac{1}{t} \log \mathbb{E}[\max \exp(tX_i)] (\text{monotonicity of the exponential}) \\
&\leq \frac{1}{t} \log \sum_{i=1}^{K} \mathbb{E}[\exp(tX_i)] \\
&= \frac{1}{t} \log \sum_{i=1}^{K} \left(\frac{p_i}{p_i - t}\right)^m \\
&\leq \frac{2}{p^*} \log(K 2^m) \text{ (choosing } t = \frac{p^*}{2}) \\
&\leq 2\alpha K(\log K + m \log 2) \text{ (invoking the } \alpha\text{-imbalance property).}
\end{aligned} \tag{A.4}$$

Plugging $m = \frac{K}{\delta\epsilon}$ into the above expression and noting that we require at most $K\mathbb{E}[T_K(m, \mathbf{p})]$ queries establishes the result.

A.2 Extensions

A.2.1 Clustering with outliers

In what follows, we focus on analyzing the query algorithm with outlier points and a noiseless oracle. We first present an algorithm that addresses this problem, Algorithm 6.

This algorithm has theoretical performance guarantees established by Theorem A.1.

Theorem A.1. Let $p_o = \frac{|\mathcal{X}_o|}{n}$. For all $\mathcal{X}$ for which the subsets $\mathcal{X}_t$ satisfy the α-imbalance property, Algorithm 6 outputs a set of centroids $\mathbf{C}$ such that with probability at least $1-\delta$, $\phi(\mathcal{X}_t; \mathbf{C}) \leq (1+\epsilon)\phi^*(\mathcal{X}_t)$. The expected query complexity of the algorithm is bounded from above by

$$\begin{aligned}
&\frac{2\alpha K^2}{1-p_o}(\log K + 2\log 2) + 2(\frac{\alpha K p_o}{1-p_o}(\log(2K) + 2\log 2))^2 \\
&+ \frac{2\alpha K}{1-p_o}(p_o + K(1-p_o))(\log K + (\frac{K}{\delta\epsilon} - 2)\log 2).
\end{aligned}$$

Algorithm 6: Query K-means Clustering with Outliers and a Noiseless Oracle $\mathcal{O}$

```
Input: A set of $n$ points $\mathcal{X}$, the number of clusters $K$, a noiseless
       oracle $\mathcal{O}$, two parameters $\delta \in (0,1), \epsilon \in (0,1)$
Output: Set of centroids $\mathcal{C}$
Phase 1: Find $K$ pairs of non-outlier points
Initialization: $\mathcal{S}_1 = \emptyset$, $R = 1$, Count= 0.
Uniformly at random sample (with replacement) a point $x$ from $\mathcal{X}$.
$\mathcal{S}_1 \leftarrow \mathcal{S}_1 \cup \{x\}$.
while $Count \leq K$ do
    Uniformly at random sample (with replacement) a point $x$ from
    $\mathcal{X}$.
    Query one point from each cluster $\mathcal{S}_r, r \in [R]$ in pair with $x$.
    if $\exists r \in [R],\ a \in \mathcal{S}_r\ s.t.\ \mathcal{O}_2(a, x) = 0$ then
        $\mathcal{S}_r \leftarrow \mathcal{S}_r \cup \{x\}$.
        Count ← Count+1.
    else
        $R \leftarrow R + 1$.
        Create a new cluster $\mathcal{S}_R = \{x\}$.
    end
end
Dispose of all clusters containing a single point only. Let $\mathcal{S}$ be the
  resulting clusters.
Phase 2: Run Algorithm 1 with clusters seeds $\mathcal{S} = \{S_1, ..., S_K\}$
while Until Algorithm 1 terminates do
    Uniformly at random sample (with replacement) a point $y$ from
    $\mathcal{X}$.
    if $\exists i \in [K]\ s.t.\ x \in S_i, \mathcal{O}(y, x) = 0$ then
        proceed with Algorithm 1.
    else
        Remove $y$.
    end
end
```

Once the clusters are seeded with sufficiently many points so that the centroids may be estimated with sufficiently high precision, all the remaining points are placed based on the $\Gamma(\beta)$-margin between outlier and non-outlier points. Clearly, if $\beta > 0$ one can distinguish all outliers from non-outliers provided that we computed the exact centroids. It is impossible to distinguish outliers from non-outliers if $\beta \leq 0$ by using distance information only. Thus, we assume that $\beta > 0$ in all our subsequent derivations. With this assumption, we arrive at the following corollary.

Corollary A.1. Assume that optimal clusters satisfy the $\Gamma(\beta)$-separation property with $\epsilon \leq \beta^2$. Let $\mathbf{C}$ be the output of Algorithm 6. For all $x \in \mathcal{X}$, let $d(x) = \min_{c \in \mathbf{C}} ||x - c||$. Assign all points $x \in \mathcal{X}$ that have not been queried to their closest centers as long as $d(x) \leq \Gamma(\beta)$. Otherwise, declare the point to be an outlier. By Theorem A.1, the resulting clustering provides a $(1+\epsilon)$-approximation of the optimal potential with probability at least $1 - \delta$.

We are now ready to present the proof of our main result in this section. First, we argue that the described algorithm indeed provides a $(1+\epsilon)$-approximation of the potential with probability at least $(1-\delta)$. Note that based on Phase 1 of Algorithm 6, we can ensure that each of the clusters $\mathcal{S}$ contains one pair of points that does not include outliers. Upon executing Phase 2 of the algorithm, by Theorem 1.1, we can immediately establish the claimed approximation guarantees.

Next, we focus on bounding the expected query complexity of the algorithm. We decompose the random variable Q capturing the number of pairwise queries made into Q_1, the query complexity of Phase 1, and Q_2, the query complexity of Phase 2.

Consider Q_1 first. Note that the process in Phase 1 will terminate if and only if we sample at least two points from each $\mathcal{C}_i^*$. Since we are sampling with replacement, this can be solved exactly by the double Dixie cup problem. Again using Poissonization arguments, we can establish that the number of points sampled in Phase 1 at step t is a random variable $Z(t) \sim$ Poisson (t). Let $Z_j(t) \sim$ Poisson $(p_j t), j \in \{o, 1, ..., K\}$, where the $Z_j(t)$ variables are independent. Moreover, let X_j denote the number of queries until we sample two points from the optimal cluster $\mathcal{C}_j^*$ and let $X = \max_{i \in [K]} X_i$. Then, we have

$$\begin{aligned}&\mathbb{E}[Q_1] \leq \mathbb{E}\left[\mathbb{E}\left[K\sum_{i=1}^{K} Z_i(X) + \sum_{j=1}^{Z_o(X)} (K+j-1)|X\right]\right] = \\ &K\,\mathbb{E}[X] + \mathbb{E}\left[\mathbb{E}\left[\frac{Z_o(X)(Z_o(X)-1)}{2}|X\right]\right] = K\,\mathbb{E}[X] + \frac{p_o^2}{2}\,\mathbb{E}[X^2].\end{aligned} \tag{A.5}$$

This first term $K\sum_{i=1}^{K} Z_i(X)$ arises due to the fact that when we sample a point from $\mathcal{X}_t$, we use at most K queries to place it. When we sample an outlier point from $\mathcal{X}_o$, assuming we have already sampled ℓ outliers, we will require most $K+\ell$ queries. This gives rise to the second term $\sum_{j=1}^{Z_o(X)}(K+j-1)$.

Next, we derive bounds for $\mathbb{E}[X]$ and $\mathbb{E}[X^2]$. For $\mathbb{E}[X]$, noting that in this case we have $m=2$ and setting $\lambda = \frac{p^*}{2} \in (0, p^* = \min_{i\in\{1,\ldots,K\}} p_i)$, we obtain

$$\begin{aligned}&\mathbb{E}[X] = \mathbb{E}\left[\max_{i\in[K]} X_i\right] \leq \frac{1}{\lambda}\log\sum_{i=1}^{K}\mathbb{E}\left[\exp(\lambda X_i)\right] = \\ &\frac{1}{\lambda}\log\sum_{i=1}^{K}\left(\frac{p_i}{p_i-\lambda}\right)^m \leq \frac{2}{p^*}\log(K2^m) \leq \frac{2\alpha K}{1-p_o}(\log K + 2\log 2).\end{aligned} \tag{A.6}$$

The last equality follows from the α-imbalance assumption, and the fact that $m=2$ by the design of the algorithm. To bound the second moment, we cannot use $\mathbb{E}\left[\exp(X_j^2)\right]$ as this expectation does not exist (since X_j is not sub-Gaussian, but sub-exponential instead).

For all $t \in (0, p^*)$, we have

$$\begin{aligned}&\mathbb{E}[X^2] = \mathbb{E}\left[(\max\, X_i)^2\right] = \mathbb{E}\left[\frac{1}{t^2}(\log(\exp(t\max X_i^2)))^2\right] = \\ &\mathbb{E}\left[\frac{1}{t^2}(\log(\max\exp(tX_i^2)))^2\right].\end{aligned}$$

Next, we note that $(\log x)^2$ is concave over $x \in [e, \infty)$ and nondecreasing, so that

$$
\begin{aligned}
&= \mathbb{E}\left[\frac{1}{t^2}(\log(\max \exp(tX_i)))^2\right] \\
&\leq \mathbb{E}\left[\frac{1}{t^2}(\log((\max \exp(tX_i))\mathbf{1}(\max tX_i \geq 1) + e\mathbf{1}(\max tX_i < 1)))^2\right] \\
&\leq \mathbb{E}\left[\frac{1}{t^2}(\log((\max \exp(tX_i)) + e))^2\right] \\
&\leq \frac{1}{t^2}(\log(\mathbb{E}\left[\max \exp(tX_i) + e\right])^2 \text{ (Jensen's inequality)} \\
&\leq \frac{1}{t^2}(\log(\sum_{i=1}^{K} \mathbb{E}[\exp(tX_i) + e])^2 \\
&= \frac{1}{t^2}(\log(\sum_{i=1}^{K}(\frac{p_i}{p_i - t})^m + e))^2 \qquad \text{(A.7)} \\
&\leq \frac{4}{(p^*)^2}(\log(K2^m + e))^2 \text{ (setting } t = \frac{p^*}{2}) \\
&\leq \left(\frac{2\alpha K}{1-p_o}\log(K2^m + e)\right)^2 \\
&\leq \left(\frac{2\alpha K}{1-p_o}\log(2K2^m)\right)^2 \text{ (assuming } K2^m \geq e) \\
&= \left(\frac{2\alpha K}{1-p_o}(\log(2K) + m\log 2)\right)^2.
\end{aligned}
$$

Note that since $m = 2$, obviously $K2^m = 4K \geq 4 \geq e$. Setting $m = 2$ in (A.7) and plugging (A.7) and (A.6) into (A.5), we have

$$
\begin{aligned}
\mathbb{E}[Q_1] &\leq K\,\mathbb{E}[X] + \frac{p_o^2}{2}\mathbb{E}[X^2] \qquad \text{(A.8)} \\
&\leq \frac{2\alpha K^2}{1-p_o}(\log K + 2\log 2) + 2(\frac{\alpha K p_o}{1-p_o}(\log(2K) + 2\log 2))^2. \qquad \text{(A.9)}
\end{aligned}
$$

To bound Q_2, we use an analysis similar to that described in the proof of Theorem 1.1 and in the previous derivations. By the same Poissonization argument as in Theorem 1.1 and above, the number of points sampled in Phase 2 at time t is $Z(t) \sim \text{Poisson}(t)$ and let $Z_j(t) \sim \text{Poisson}(p_j t)$, $j \in \{o, 1, ..., K\}$; the variables $Z_j(t)$ are independent. Let X_j be the time by which we have sampled m points from $\mathcal{C}_j^*$, for $j \in \{1, ..., K\}$ and let $X = \max_{i\in[K]} X_i$. Note that the variables $X_j \sim \text{Erlang}(m, \lambda_j)$, where $\lambda_j = \frac{1}{p_j}$. X_j

are independent. Then,

$$\mathbb{E}[Q_2] \leq \mathbb{E}\left[\mathbb{E}\left[Z_o(X) + K\sum_{i=1}^{K} Z_i(X)|X\right]\right] = (p_o + K(1-p_o))\,\mathbb{E}[X]. \quad \text{(A.10)}$$

Since $X = \max_{i\in[K]} X_i$ is independent of Z_o and Z_o is Poisson distributed, the first term equals $p_o\,\mathbb{E}[X]$. The second term is obtained as follows.

Let $Y = \sum_{i=1}^{K} Z_i(X)$, so that $\mathbb{E}[X] = \mathbb{E}\left[\sum_{i=1}^{Y} U_i\right] = \frac{1}{1-p_o}\,\mathbb{E}[Y]$, where the variables U_i are iid exponential with rate $1-p_o$. Plugging in equation (A.4) with $m = \frac{K}{\delta\epsilon}$, we obtain

$$\mathbb{E}[Q_2] \leq \frac{2\alpha K}{1-p_o}(p_o + K(1-p_o))(\log K + (\frac{K}{\delta\epsilon} - 2)\log 2). \quad \text{(A.11)}$$

Consequently,

$$\begin{aligned}\mathbb{E}[Q] \leq \frac{2\alpha K^2}{1-p_o}(\log K + 2\log 2) + 2(\frac{\alpha K p_o}{1-p_o}(\log(2K) + 2\log 2))^2 \\ + \frac{2\alpha K}{1-p_o}(p_o + K(1-p_o))(\log K + (\frac{K}{\delta\epsilon} - 2)\log 2),\end{aligned} \quad \text{(A.12)}$$

which completes the proof.

A.2.2 The case of finding centers of top r largest clusters

We denote the ratio of the size of cluster $|S_i^*|$ to n as $p_i = \frac{|S_i^*|}{n}$. Without loss of generality we assume p_i are also sorted. Then we define $\bar{p} = \frac{p_r + p_{r+1}}{2}$ and $\Delta = p_r - \bar{p}$. Moreover, let $KL(a||b)$ be the KL-divergence for Bernoulli(a) and Bernoulli(b). We denote $D = \min\{KL(\bar{p}||\bar{p}+\Delta), KL(\bar{p}||\bar{p}-\Delta)\}$.

Theorem A.2. Assume that $\bar{p} \in (0,1)$ and $\Delta > 0$. Let us denote $\tilde{p} = \sum_{i=1}^{r} p_i$. There exists an algorithm which will output a set of centers $\mathcal{C}$ such that with probability at least $1-\delta$, $\phi(\mathcal{X}^{(r)};\mathcal{C}) \leq (1+\epsilon)\phi^*(\mathcal{X}^{(r)})$. The expected query complexity $\mathbb{E}Q$ can be bounded as

$$\mathbb{E}Q \leq \frac{K}{D}\log\frac{2K}{\delta} + \frac{2}{p_r}(1 - \tilde{p} + K\tilde{p})(\log K + \frac{2K}{\delta\epsilon}\log 2).$$

Moreover, the expected time complexity for outputting $\mathcal{C}$ is $O(\mathbb{E}Q)$.

Remark A.1. The first term in the query complexity upper bound is the query complexity for Phase 1, which is clearly $KM = \frac{K}{D}\log\frac{2K}{\delta}$ by our choice of M. The second term will be similar to the case of Theorem A.1. Note that our bound here should be further refinable since the bound we use for Phase 2 here is to collect $\frac{2K}{\delta\epsilon}$ for all top r largest clusters starting with one point for each cluster. However for simplicity we present the looser bound here.

A.2.3 Clustering with a noisy oracle

Next, we analyze the query algorithm with a noisy oracle. Recall that the noisy oracle $\mathcal{O}_n$ gives a correct answer with probability $1-p_e$, where $p_e < \frac{1}{2}$. For the same query, we always get the same answer, which prevents us from repeatedly asking the same query to increase the probability of success [26]. This assumption is motivated by crowdsourcing applications in which non-experts often provide answers based on the same source (i.e., the first result obtained by searching Google). Nevertheless, the assumption that the answers are provided independently is unrealistic but still used in order to make the analysis tractable [26].

Before describing the underlying algorithm, let M be the smallest positive integer that satisfies two inequalities,

$$\frac{M}{\log M} \geq \frac{128\alpha K^2}{(2p_e-1)^4} \tag{A.13}$$

and

$$M \geq \tilde{M} = \max\{\frac{6\alpha K}{\delta\epsilon}, 8\alpha K\log\frac{3K}{\delta}\}.$$

The noisy query algorithm is described below.

Theorem A.3 (Theoretical guarantees for Algorithm 7)**.** Assume that one is given a set of n points $\mathcal{X}$ with an underlying optimal set of K clusters $\mathcal{X} = \bigcup_{i=1}^{K}\mathcal{C}_i^*$. Suppose that $\mathcal{X}$ satisfies the α-imbalance property. Let

$$\tilde{M} = \max\{\frac{6\alpha K}{\delta\epsilon}, 8\alpha K\log\frac{3K}{\delta}\} \tag{A.14}$$

and $M \geq \tilde{M}$. Let $M \in \mathbb{N}$ be the smallest positive integer simultaneously satisfying (A.13) and $\tilde{M} \leq M$. Algorithm 7 returns a set of centers $\mathbf{C}$ such that with probability at least $1-\delta$, $\phi(\mathcal{X};\mathbf{C}) \leq \phi_K^*(\mathcal{X})$, provided that all

Algorithm 7: Query K-means Clustering with a Noisy Oracle $\mathcal{O}_n$

```
Input: A set of $n$ points $\mathcal{X}$, the number of clusters $K$, a noisy oracle
       $\mathcal{O}_n$ and a precomputed value $M$
Output: Set of centers $\mathbf{C}$
Phase 1: Seed the clusters by running Algorithm 3
Uniformly at random sample (without replacement) $M$ point $x$
independently from $\mathcal{X}$. Denote the obtained subset by $\mathcal{A}$.
Run Algorithm 3 on the set $\mathcal{A}$. Generate a $K$-partition of
$\mathcal{A} = \bigcup_{i=1}^{K} S_i$.
Phase 2: Estimate the centroids
For all $i \in [K]$, set $c_i \leftarrow c(S_i)$ where $c(S_i)$ is the average of set $S_i$.
$\mathbf{C} \leftarrow \{c_1, ..., c_K\}$.
```

points are assigned to their closest centers in $\mathbf{C}$. The query complexity of the algorithm is $O(\frac{MK^2 \log M}{(1-2p_e)^4})$, while the overall running time of the algorithm is $O(Kn + \frac{MK \log M}{(1-2p_e)^2} + KN^{\omega})$, with $N = \frac{64K^2 \log M}{(1-2p_e)^4}$ and $\omega \leq 2.373$ (the complexity exponent in fast matrix multiplication).

Remark A.2. First, observe that given $\tilde{M}' = \frac{8\alpha K}{\epsilon\delta} \log(\frac{3K}{\delta})$, one has $\tilde{M}' \geq \tilde{M}$. Hence, for any M satisfying $M \geq \tilde{M}'$ we automatically have $M \geq \tilde{M}$. To handle the condition (A.13), we use a bootstrapping approximation for the log term, ignoring all log-log and smaller terms. This procedure leads to the following bound:

$$M \geq \frac{128\alpha K^2}{(2p_e - 1)^4} \log \frac{128\alpha K^2}{(2p_e - 1)^4}.$$

For fixed constants p_e, δ, ϵ, we have $M = O(\alpha K^2 \log(\alpha K))$. This implies that the resulting query complexity of Algorithm 7 is $O(\alpha K^4 \log(\alpha K) \times \log(\alpha K^2 \log(\alpha K)))$, or $O(\alpha K^4 (\log(\alpha K))^2)$.

Next, we prove Theorem A.3. Our proof will rely on the theoretical guarantee of Algorithm 2 in [26], restated below.

Theorem A.4 (Theorem 3 of [26])**.** Assume that one is given a set of M points $\mathcal{X}$ partitioned into K clusters, $\mathcal{X} = \bigcup_{i=1}^{K} \mathcal{C}_i$. Let $N = \frac{64K^2 \log M}{(1-2p_e)^4}$. Then Algorithm 2 in [26] returns all clusters of size at least $\frac{64K \log M}{(1-2p_e)^4}$ with probability at least $1 - \frac{2}{M}$. The query complexity of the method is $O(\frac{MK^2 \log M}{(1-2p_e)^4})$ and the total running is $O(\frac{MK \log M}{(1-2p_e)^2} + KN^{\omega})$, where $\omega \leq 2.373$ is the complexity exponent of fast matrix multiplication.

Remark A.3. Note that in [26] they do not assume that the underlying partition $\bigcup_{i=1}^{K} \mathcal{C}_i$ is the optimal solution of k-means problem. Hence in the statement of theorem we use $\mathcal{C}$ to denote the underlying partition instead of $\mathcal{C}^*$. The key is that the partition $\mathcal{C}$ should be consistent with the answers given by the (noiseless) oracle.

We start by modifying Lemma 2.1 for the case that sampling is performed without replacement.

Lemma A.1 (The Centroid lemma for sampling without replacement)**.** Let S be a set of points obtained by sampling m points without replacement and uniformly at random from A. Then, for any $\delta > 0$, with probability at least $1 - \delta$,

$$\phi(\mathcal{A}; c(S)) \leq (1 + \frac{1 - \frac{m-1}{|A|-1}}{\delta m})\phi_1^*(\mathcal{A}) \leq (1 + \frac{1}{\delta m})\phi_1^*(\mathcal{A}).$$

Here $c(S)$ denotes the centroid of the set S. Clearly, we require $m \leq |A|$.

Proof. Let $S = \{y_1, ..., y_m\}$ be the set of m points we sampled. Let $\mathbb{E}_{\bar{y}}$ denote the expectation with respect to $y_1, ..., y_m$. By using a bias variance decomposition, we have

$$\sum_{x \in \mathcal{A}} ||x - c(S)||^2 = \sum_{x \in \mathcal{A}} ||x - c(\mathcal{A})||^2 + |\mathcal{A}| \cdot ||c(S) - c(\mathcal{A})||^2.$$

We start by analyzing the term $\mathbb{E}_{\bar{y}}||c(S) - c(\mathcal{A})||^2$. By definition, we have

$$\begin{aligned}
&\mathbb{E}_{\bar{y}}||c(S) - c(\mathcal{A})||^2 = \mathbb{E}_{\bar{y}}||\frac{1}{m}\sum_{i=1}^{m}(y_i - c(\mathcal{A}))||^2 = \frac{1}{m^2}\mathbb{E}_{\bar{y}}||\sum_{i=1}^{m}(y_i - c(\mathcal{A}))||^2 \\
&= \frac{1}{m^2}\mathbb{E}_{\bar{y}}(\sum_{i=1}^{m}||(y_i - c(\mathcal{A}))||^2 + \sum_{i \neq j}\langle y_i - c(\mathcal{A}), y_j - c(\mathcal{A})\rangle) \\
&= \frac{1}{m^2}(\sum_{i=1}^{m}\mathbb{E}_{\bar{y}}||(y_i - c(\mathcal{A}))||^2 + \sum_{i \neq j}\mathbb{E}_{\bar{y}}\langle y_i - c(\mathcal{A}), y_j - c(\mathcal{A})\rangle) \\
&= \frac{1}{m^2}(m\phi^*(\mathcal{A}) + \sum_{i \neq j}\mathbb{E}_{y_i}\langle y_i - c(\mathcal{A}), \mathbb{E}_{y_j|y_i}(y_j - c(\mathcal{A}))\rangle).
\end{aligned}$$

Furthermore, note that

$$\begin{aligned}
&\mathbb{E}_{y_j|y_i}(y_j - c(\mathcal{A})) = \frac{1}{|A| - 1}\sum_{y_j \in \mathcal{A}/\{y_i\}}((y_j - c(\mathcal{A})) \\
&= \frac{1}{|\mathcal{A}| - 1}(|\mathcal{A}|c(\mathcal{A}) - y_i - (|\mathcal{A}| - 1)c(\mathcal{A})) = \frac{-1}{|\mathcal{A}| - 1}(y_i - c(\mathcal{A})).
\end{aligned}$$

Hence, we have

$$\begin{aligned}
&\frac{1}{m^2}(m\phi^*(\mathcal{A}) + \sum_{i\neq j} \mathbb{E}_{y_i} \left\langle y_i - c(\mathcal{A}), \mathbb{E}_{y_j|y_i}(y_j - c(\mathcal{A}))\right\rangle) \\
&= \frac{1}{m^2}\left(m\phi^*(\mathcal{A}) + \sum_{i\neq j} \mathbb{E}_{y_i} \left\langle y_i - c(\mathcal{A}), \frac{-1}{|\mathcal{A}|-1}(y_i - c(\mathcal{A}))\right\rangle\right) \\
&= \frac{1}{m^2}(m\phi^*(\mathcal{A}) - \frac{1}{|\mathcal{A}|-1}\sum_{i\neq j} \mathbb{E}_{y_i}||y_i - c(\mathcal{A})||^2) \\
&= \frac{1}{m^2}(m\phi^*(\mathcal{A}) - \frac{1}{|\mathcal{A}|-1}m(m-1)\phi^*(\mathcal{A})) = \frac{\phi^*(\mathcal{A})}{m}(1 - \frac{(m-1)}{|\mathcal{A}|-1}).
\end{aligned}$$

Combining the above equations and by invoking Markov's inequality, we obtain the desired result. □

We also make use of the following result.

Lemma A.2 ([34])**.** Let $D(p_x||p_y)$ denote the KL divergence between two Bernoulli distribution with parameters $p_x \leq p_y \in [0,1]$. Then,

$$D(p_x||p_y) \leq \frac{(p_y - p_x)^2}{2p_y}. \tag{A.15}$$

Remark A.4. Note that this bound is tighter than the one obtained directly from Pinsker's inequality whenever $p_y \leq 1/8$.

We are now ready to prove Theorem A.3.

Proof. Assume that we sample (without replacement) uniformly at random M points from $\mathcal{X}$, and denote the subsampled set of points by $\mathcal{X}'$. Note that $\mathcal{X}'$ can be partitioned into at most K clusters so that for all $i \in [K], S_i^\dagger = \mathcal{X}' \bigcap \mathcal{C}_i^*$. Clearly, the vector $(S_1^\dagger, ..., S_K^\dagger)$ is a multivariate hypergeometric random vector with parameters $(n, np_1, ..., np_K, M)$, where $p_i = \frac{|S_i^*|}{n}$, $\forall i \in [K]$. As before, we write $p^* = \min_i p_i = \frac{1}{\alpha K}$, where the second equality follows from the α-imbalance property. In particular, $S_i^\dagger$ is a hypergeometric random variable with parameters (n, np_i, M). Using Hoeffding's inequality [32, 35], we obtain

$$
\begin{aligned}
&P\{S_i^\dagger < M(p_i - \frac{p_i}{2})\} \leq \exp\left(-MD(\frac{p_i}{2}||p_i)\right) \leq \exp\left(-\frac{Mp_i}{8}\right) \\
&\Rightarrow P\{S_i^\dagger < \frac{Mp_i}{2}\} \leq \exp\left(-\frac{Mp_i}{8}\right) \Rightarrow P\{S_i^\dagger < \frac{Mp^*}{2}\} \leq \exp\left(-\frac{Mp^*}{8}\right).
\end{aligned}
\tag{A.16}
$$

Here, we used the bound $D((1-a)p||p) \geq \frac{1}{2}a^2p,\ a \in [0, \frac{1}{2}]$, which is a direct consequence of Lemma A.2. By using the union bound, we have

$$
P\{\min S_i^\dagger \geq \frac{Mp^*}{2}\} \geq 1 - K\exp(-\frac{Mp^*}{8}), \tag{A.17}
$$

which completes the proof of Theorem 3.1. Min $S_i^\dagger \geq \max\{\frac{64K\log M}{(2p_e-1)^4}, \frac{3K}{\delta\epsilon}\}$ is required, which corresponds to (A.13) and gives rise to the first term in (A.14). The first term under the maximum is needed in order to satisfy the requirements of Theorem A.4. The second term under the maximum is needed because we want to apply Lemma 3.2. By properly choosing M we can ensure that these two conditions are met. Also, we want the statement to hold with probability at least $1 - \frac{\delta}{3}$, for which we need $M \geq 8\alpha K \log\frac{3K}{\delta}$. This gives rise to the second term in (A.14). Again, for our given choice of M this constraint is also satisfied. As a result, from Theorem A.4, we know that upon completion of Phase 1, we will have generated the desired partition $S_1^\dagger, ..., S_K^\dagger$ with probability at least $1 - \frac{2}{M}$. Due to our choice of M, the former probability is at least $1 - \frac{\delta}{3}$.

Finally, since every point in $S_1^\dagger, ..., S_K^\dagger$ is obtained by sampling uniformly at random from $\mathcal{X}$, Lemma 3.2, the union bound and the choice of M guarantee that, with probability at least $1 - \frac{\delta}{3}$, the resulting set of centers $\mathcal{C}$ provides a $(1+\epsilon)$-approximation of the optimal potential ϕ^*. The query complexity and the time complexity follow directly from Theorem A.4. This completes the proof. □

A.3 Proof of Theorem 1.2

Let us restate Theorem 1.2 as follows.

Theorem A.5 (Theoretical guarantees for Algorithm 2)**.** Assume that one is given a set of n points $\mathcal{X}$ with an underlying optimal K-clustering $\mathcal{X} = \bigcup_{i=1}^{K} \mathcal{C}_i^*$ and that the clusters satisfy the α-imbalance property. Let

$$\begin{aligned}
\tilde{M} &= \max\left\{\frac{128\alpha K^2}{(2p_e-1)^4}\log\frac{128\alpha K^2}{(2p_e-1)^4}, \frac{8\alpha K}{\delta\epsilon}, 8\alpha K\log\frac{4K}{\delta}\right\},\\
M &= \frac{2}{1-p_o}\tilde{M} + \frac{1}{2(1-p_o)^2}\log\frac{4}{\delta},\\
N &= \frac{64K^2\log M}{(1-2p_e)^4} + M - \tilde{M}.
\end{aligned}$$

Algorithm 2 returns a set of centers $\mathbf{C}$ such that, with probability at least $1-\delta$, $\phi(\mathcal{X};\mathbf{C}) \leq \phi_K^*(\mathcal{X})$. The query complexity of the algorithm is $O(\frac{MK^2\log M}{(1-2p_e)^4})$. Moreover, if we assign all points to their closest centers in $\mathbf{C}$, we can complete the clustering in time $O(Knd + \frac{MK\log M}{(1-2p_e)^2} + KN^\omega)$, where $N \sim O(\frac{\alpha K^2\log M}{(1-2p_e)^4})$ and $\omega \leq 2.373$ is the complexity exponent of fast matrix multiplication.

Proof. In order to use Theorem A.3, we first need to make sure that the M points selected from $\mathcal{X}$ will contain at least $\tilde{M}$ non-outlier points with high probability, where $\tilde{M}$ satisfies the conditions required by Theorem A.3. We also need to adapt the value of the parameter N, as N is used to lower bound the size of the largest cluster as N/K, and in our setting outliers need to be taken into consideration. There are two approaches to deal with this issue.

The first approach is to select M points uniformly at random from $\mathcal{X}$, containing at least $\tilde{M}$ non-outliers with probability at least $1-\frac{\delta}{4}$. Clearly, in this case, the number of outliers is upper bounded by $M-\tilde{M}$. If $\frac{N-M+\tilde{M}}{k} \geq \frac{8\sqrt{N\log M}}{(1-2p_e)^2}$, we can then directly use the result of [26]. We can simplify the problem as one in which there are M independent Bernoulli random variables $\{X_i\}_{i=1}^M$, that take the value 0 with probability p_o (outliers), standing for outlier, and 1 with probability $1-p_o$ (non-outliers). Then the number of non-outliers among these M points is the sum of the independent Bernoulli random variables described above. By Hoeffding's inequality, we have

$$P\{\sum_{i=1}^{M} X_i \leq \mathbb{E}\left[\sum_{i=1}^{M} X_i\right] - t\} \leq \exp\left(-\frac{2t^2}{M}\right).$$

Let $t = \mathbb{E}\left[\sum_{i=1}^{M} X_i\right] - \tilde{M} = (1-p_o)M - \tilde{M}$, and $\exp(-2t^2/M) \leq \frac{\delta}{4}$. Then the selected M points will contain more than $\tilde{M}$ non-outliers with probability at least $1-\frac{\delta}{4}$. Combining the above results we obtain the following inequality:

$$((1-p_o)M-\tilde{M})^2 \geq \frac{M}{2}\log\frac{4}{\delta}.$$

By solving this inequality we get $M \geq \frac{2\tilde{M}}{1-p_o}+\frac{1}{2(1-p_o)^2}\log\frac{4}{\delta}$. Based on Theorem A.3, we also need $\tilde{M}$ to satisfy

$$\tilde{M} = \max\left\{\frac{128\alpha K^2}{(2p_e-1)^4}\log\frac{128\alpha K^2}{(2p_e-1)^4}, \frac{8\alpha K}{\delta\epsilon}, 8\alpha K\log\frac{4K}{\delta}\right\}.$$

For the second part of analysis which ensures each cluster in subset $\mathcal{A}$ has enough points with probability 1, we need

$$\frac{N-M+\tilde{M}}{k} \geq \frac{8\sqrt{N\log M}}{(1-2p_e)^2}.$$

By solving this inequality for N, we get

$$N = \frac{64K^2\log M}{(1-2p_e)^4}+M-\tilde{M}.$$

In this case, we know that with probability at least $1-\delta$, Algorithm 2 provides a $(1+\epsilon)$-approximation of the true potential for the case of queries involving non-outliers.

The second approach is to select M points uniformly at random from $\mathcal{X}$, containing at least $\tilde{M}$ non-outliers with probability at least $1-\frac{\delta}{5}$. Following the same procedure as described above, we get

$$M = \frac{2\tilde{M}}{1-p_o}+\frac{1}{2(1-p_o)^2}\log\frac{5}{\delta}.$$

For the second part of the analysis which ensures that each cluster in subset $\mathcal{A}$ has enough points with high probability, we require the N chosen points in each round to contain at least N' non-outliers, where $\frac{N'}{K} \geq \frac{8\sqrt{N\log M}}{(1-2p_e)^2}$ with probability at least $\frac{\delta}{5K}$. Then,

$$N = \frac{2N'}{1-p_o}+\frac{1}{2(1-p_o)^2}\log\frac{5K}{\delta},$$

and

$$N' = \frac{128K^2 \log M + 4\sqrt{2}(1-2p_e)^2 K \sqrt{\log M \log \frac{5K}{\delta}}}{(1-2p_e)^4(1-p_o)}.$$

By using the union bound for all error events, we conclude that with probability at least $1-\delta$, Algorithm 2 offers a $(1+\epsilon)$-approximation guarantee for the optimal potential for non-outlier points.

Note that although in both methods we had to change the value of N, the value remained $O(\frac{\alpha K^2 \log M}{(1-2p_e)^4})$. Therefore, the overall query complexity equals $O(\frac{MK^2 \log M}{(1-2p_e)^4})$. Furthermore, if all points are assigned to their closest centers in $\mathbf{C}$, the clustering can be completed with overall running time $O(Knd + \frac{MK \log M}{(1-2p_e)^2} + KN^{\omega})$, where $\omega \leq 2.373$ is the complexity exponent of fast matrix multiplication. □

A.4 Proof of Theorem 1.3

Our proof will rely on the theoretical guarantee of Algorithm 2 in [16], restated below.

Theorem A.6 (Theorem 4 of [16])**.** Assume that one is given parameters $\delta \in (0,1)$ and the number of clusters K. Define $p_e \in [\rho, 1-\rho]$ as the probability that a worker mislabels a sample. Let $\kappa = \left|1 - p_e - \frac{1}{K}\right|, \bar{D} = \frac{K-1-Kp_e}{K-1} \log \frac{(K-1)(1-p_e)}{p_e}$. Then, Algorithm 5 can exactly recover true labels of all samples with probability at least $1-\delta$ if the number of workers w and the number of samples s satisfy

$$w = \Omega\left(\frac{\log(1/\rho)\log(Ks/\delta) + \log ws}{\bar{D}}\right),\ s = \Omega\left(\frac{\log w/\sqrt{\delta}}{\kappa^6 \min\{\kappa^2, \rho^2, (\rho\bar{D})^2\}}\right).$$

Combining this result with Lemma 3.2 and Theorem 3.1 finishes the proof of Theorem 1.3.

REFERENCES

[1] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[2] S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[3] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[4] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, 2002.

[5] S. Ray and R. H. Turi, "Determination of number of clusters in k-means clustering and application in colour image segmentation," in *Proceedings of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques.* Calcutta, India, 1999, pp. 137–143.

[6] M. Mahajan, P. Nimbhorkar, and K. Varadarajan, "The planar k-means problem is NP-hard," in *International Workshop on Algorithms and Computation.* Springer, 2009, pp. 274–285.

[7] P. Awasthi, M. Charikar, R. Krishnaswamy, and A. K. Sinop, "The hardness of approximation of Euclidean k-means," *arXiv preprint arXiv:1502.03316*, 2015.

[8] E. Lee, M. Schmidt, and J. Wright, "Improved and simplified inapproximability for k-means," *Information Processing Letters*, vol. 120, pp. 40–43, 2017.

[9] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "A local search approximation algorithm for k-means clustering," *Computational Geometry*, vol. 28, no. 2-3, pp. 89–112, 2004.

[10] S. Ahmadian, A. Norouzi-Fard, O. Svensson, and J. Ward, "Better guarantees for k-means and euclidean k-median by primal-dual algorithms," in *Foundations of Computer Science (FOCS), 2017 IEEE 58th Annual Symposium on.* IEEE, 2017, pp. 61–72.

[11] H. Ashtiani, S. Kushagra, and S. Ben-David, "Clustering with same-cluster queries," in *Advances in Neural Information Processing Systems*, 2016, pp. 3216–3224.

[12] N. Ailon, A. Bhattacharya, R. Jaiswal, and A. Kumar, "Approximate clustering with same-cluster queries," *arXiv preprint arXiv:1704.01862*, 2017.

[13] B. Gamlath, S. Huang, and O. Svensson, "Semi-supervised algorithms for approximately optimal and accurate clustering," *arXiv preprint arXiv:1803.00926*, 2018.

[14] T. Kim and J. Ghosh, "Semi-supervised active clustering with weak oracles," *arXiv preprint arXiv:1709.03202*, 2017.

[15] A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the EM algorithm," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 28, no. 1, pp. 20–28, 1979.

[16] Y. Zhang, X. Chen, D. Zhou, and M. I. Jordan, "Spectral methods meet EM: A provably optimal algorithm for crowdsourcing," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 3537–3580, 2016.

[17] S.-Y. Yun and A. Proutiere, "Community detection via random and adaptive sampling," in *Conference on Learning Theory*, 2014, pp. 138–175.

[18] K.-G. Woo, J.-H. Lee, M.-H. Kim, and Y.-J. Lee, "FINDIT: A fast and intelligent subspace clustering algorithm using dimension voting," *Information and Software Technology*, vol. 46, no. 4, pp. 255–271, 2004.

[19] G. Dasarathy, R. Nowak, and X. Zhu, "S2: An efficient graph based active learning algorithm with application to nonparametric classification," in *Conference on Learning Theory*, 2015, pp. 503–522.

[20] P. Bradley, K. Bennett, and A. Demiriz, "Constrained k-means clustering," *Microsoft Research, Redmond*, pp. 1–8, 2000.

[21] D. J. Newman and L. Shepp, "The double Dixie cup problem," *The American Mathematical Monthly*, vol. 67, no. 1, pp. 58–61, 1960.

[22] A. V. Doumas and V. G. Papanicolaou, "The coupon collector's problem revisited: generalizing the double Dixie cup problem of Newman and Shepp," *ESAIM: Probability and Statistics*, vol. 20, pp. 367–399, 2016.

[23] M. Inaba, N. Katoh, and H. Imai, "Applications of weighted Voronoi diagrams and randomization to variance-based k-clustering," in *Proceedings of the Tenth Annual Symposium on Computational geometry.* ACM, 1994, pp. 332–339.

[24] W. Szpankowski, "Analytic poissonization and depoissonization," *Average Case Analysis of Algorithms on Sequences*, pp. 442–519, 2001.

[25] A. Mazumdar and B. Saha, "Clustering with an oracle," in *Communication, Control, and Computing (Allerton), 2016 54th Annual Allerton Conference on.* IEEE, 2016, pp. 738–739.

[26] A. Mazumdar and B. Saha, "Clustering with noisy queries," in *Advances in Neural Information Processing Systems*, 2017, pp. 5790–5801.

[27] E. Abbe, A. S. Bandeira, and G. Hall, "Exact recovery in the stochastic block model," *IEEE Transactions on Information Theory*, vol. 62, no. 1, pp. 471–487, 2016.

[28] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner et al., "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[29] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Technical report, University of Toronto*, vol. 1, no. 4, 2009.

[30] I. Chien, C. Pan, and O. Milenkovic, "Query k-means clustering and the double Dixie cup problem," in *Advances in Neural Information Processing Systems*, 2018, pp. 6649–6658.

[31] S. Hui and C. Park, "The representation of hypergeometric random variables using independent Bernoulli random variables," *Communications in Statistics-Theory and Methods*, vol. 43, no. 19, pp. 4103–4108, 2014.

[32] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 13–30, 1963.

[33] N. B. Shank and H. Yang, "Coupon collector problem for non-uniform coupons and random quotas," *The Electronic Journal of Combinatorics*, vol. 20, no. 2, p. 33, 2013.

[34] Wikipedia contributors, "Chernoff bound — Wikipedia, the free encyclopedia," 2018. [Online]. Available: https://goo.gl/CFJsvT

[35] M. Skala, “Hypergeometric tail inequalities: ending the insanity,” *arXiv preprint arXiv:1311.5939*, 2013.