Optimal Learning of Joint Alignments with a Faulty Oracle

Kasper Green Larsen[*] Michael Mitzenmacher [†] Charalampos E. Tsourakakis[‡]

September 21, 2019

Abstract

We consider the following problem, which is useful in applications such as joint image and shape alignment. The goal is to recover n discrete variables $g_i \in \{0, \ldots, k-1\}$ (up to some global offset) given noisy observations of a set of their pairwise differences $\{(g_i - g_j) \bmod k\}$; specifically, with probability $\frac{1}{k} + \delta$ for some $\delta > 0$ one obtains the correct answer, and with the remaining probability one obtains a uniformly random incorrect answer. We consider a learning-based formulation where one can perform a query to observe a pairwise difference, and the goal is to perform as few queries as possible while obtaining the exact joint alignment. We provide an easy-to-implement, time efficient algorithm that performs $O\big(\frac{n \lg n}{k\delta^2}\big)$ queries, and recovers the joint alignment with high probability. We also show that our algorithm is optimal by proving a general lower bound that holds for all non-adaptive algorithms. Our work improves significantly recent work by Chen and Candés [CC16], who view the problem as a constrained principal components analysis problem that can be solved using the power method. Specifically, our approach is simpler both in the algorithm and the analysis, and provides additional insights into the problem structure.

[*] Aarhus University, larsen@cs.au.dk.

[†] Harvard University, michaelm@eecs.harvard.edu.

[‡] Boston University, ctsourak@bu.edu.

1 Introduction

Learning a joint alignment from pairwise differences is a problem with various important applications ranging from shape matching [HSG13], to spectroscopy imaging [WS13]. In this work we adopt the following established mathematical formalization of this problem. There exists a set $V = [n]$ of n discrete items, and an assignment $g : V \to [k]$ according to which each item is assigned one out of k possible labels. To give an example, imagine a set of n images of the same object in k possible orientations/angles, where each $g(i)$ is one of k possible orientations (angles) of the camera when taking the i-th image. Recovering g would allow a better understanding of the 3-dimensional structure of the object. The assignment function g is unknown, but we may obtain a set of pairwise noisy difference samples $\{\tilde{f}(i,j) \stackrel{\text{def}}{=} (g(i) - g(j) + \text{noise}) \bmod k\}_{(i,j)\in\Omega}$ where $\Omega \subseteq \binom{[n]}{2}$ is a symmetric index set, i.e., a set of pairs $\{i,j\}$ with $i < j$. In this work, we consider the setting where each pair can be queried at most once (e.g., the measurement will not change on repeated queries), and the noisy measurement $\tilde{f}(x,y)$ is equal to

$$\tilde{f}(x,y) = \big(g(x) - g(y) + \eta_{xy}\big) \bmod k \tag{1}$$

where the additive noise values η_{xy} are i.i.d. random variables supported on $\{0, 1, \cdots, k-1\}$, with the following probability distribution that is slightly biased towards zero for some parameter $\delta > 0$:

$$\mathbf{Pr}\left[\eta_{xy} = i\right] = \begin{cases} \frac{1}{k} + \delta, & \text{if } i = 0; \\ \frac{1}{k} - \frac{\delta}{k-1}, & \text{for each } i \neq 0. \end{cases} \tag{2}$$

In this work we study the problem of recovering g up to some global offset by choosing the set of queries Ω.

Related Work. Learning joint alignments is a major problem that appears in numerous settings under different guises. In cryo-electron microscopy, the problem corresponds to recovering the angles from which 2d pictures of a 3d object were taken. This allows for the construction of a 3d model of the objective [SS12]. In shape matching, a key problem is assembling fractured surfaces [HFG$^+$06] and fusing scans to model reality [HH02], jointly optimizing the maps between shapes improves the performance compared to matching shapes in isolation [HG13].

Closest to our work lies the work of Chen and Candès [CC16], who study the same model (Equation (1)[1]). They provide an algorithm that is non-adaptive, and the underlying queries form a random binomial graph, i.e. each edge as queried independently with a fixed probability. They show that, in the setting where queries form a random binomial graph, the minimax probability of error tends to 1 if the number of queries is less than $\Omega\left(\frac{n \log n}{k\delta^2}\right)$ [CC16, Theorem 2,p. 7]. Their algorithm, based on the projected power method, has a required number of queries that matches the lower bound. Inferior results have been obtained in the past as well. Notably, a simpler non-adaptive algorithm with somewhat inferior query complexity that relies on simple breadth-first search was proposed by Mitzenmacher and Tsourakakis [MT18]. Chen et al. provide an SDP-based algorithm [CGH14] that is slower and with more stringent recovery conditions than [CC16]. A closely related but different approach with respect to the mathematical formulation is the phase/angular synchronization problem [Sin11, ZB18]. It is worth remarking that the special case $k = 2$ reduces to an active learning problem related to graph partitioning problem that is

[1] The parameter π_0 in their random corruption model, and our bias δ are connected with the following equation $\delta = \pi_0 \frac{k-1}{k}$.

well-studied, e.g. [MS17, TML$^+$17], with close connections to the classic planted partition problem [AS15, HWX16, McS01, Tso15].

Our Results. In this paper we provide a simpler non-adaptive algorithm that we prove also succeeds with high probability with $O\left(\frac{n\log n}{k\delta^2}\right)$ queries. Our algorithm is based on selecting a small seed set and using queries to obtain and reconcile all edge measurements for edges adjacent to these vertices; this approach itself appears of interest. We also provide a simpler lower bound argument showing our result is tight in terms of the number of queries required in this more general setting where queries are arbitrary.

2 Proposed Method

2.1 Preliminaries

Both our algorithm and our lower bound proof need tight concentration inequalities on the probability that the majority of a collection of biases η_{xy} is equal to 0. We state the two concentration inequalities here. The proofs are in Section 2.4. The first lemma considers the case of small δ:

Lemma 2.1. *Let $k \geq 2$ be an integer, let $0 \leq \delta \leq 1/2k$ and let $X_1, \ldots, X_n$ be i.i.d. random variables such that each X_i takes the value 1 with probability $1/k + \delta$, the value -1 with probability $1/k - \delta/(k-1)$ and the value 0 otherwise. There exists constants $c_1, c_2 > 0$ such that:*

$$\Pr[\sum_i X_i \leq 0] \leq c_1 \exp(-\delta^2 nk/c_1)$$

and

$$\Pr[\sum_i X_i \leq 0] \geq c_2^{-1} \exp(-\delta^2 nkc_2).$$

And the second considers the case of large δ:

Lemma 2.2. *Let $k \geq 2$ be an integer, let $1/2k < \delta \leq 1/4$ and let $X_1, \ldots, X_n$ be i.i.d. random variables such that each X_i takes the value 1 with probability $1/k + \delta$, the value -1 with probability $1/k - \delta/(k-1)$ and the value 0 otherwise. There exists constants $c_1, c_2 > 0$ such that:*

$$\Pr[\sum_i X_i \leq 0] \leq c_1 \exp(-\delta n/c_1)$$

and

$$\Pr[\sum_i X_i \leq 0] \geq c_2^{-1} \exp(-\delta nc_2).$$

2.2 Upper bound - Proposed Algorithm

Our algorithm is a simple and efficient non-adaptive algorithm. The basic idea is to choose a set of nodes S as a *seed* set of nodes. We then make all queries between S and the full node set V. Based on these queries, we first determine the label $g(s)$ of all nodes $s \in S$ (up to a cyclic shift). Once these have been determined, we can determine the labels of all remaining nodes v by using a plurality vote on $\{g(s) + \tilde{f}(v, s) \bmod k \mid s \in S\}$. We proceed to give the details.

Lemma 2.3 (Plurality vote)**.** *Let $S \subseteq V$ be an arbitrary seed set of nodes and assume $k \leq n^{o(1)}$. For any node $v \in V \setminus S$, the plurality vote among $\{g(s) + \tilde{f}(v,s) \bmod k \mid s \in S\}$ is equal to $g(v)$ with probability at least $1 - \frac{1}{n^2}$ if either:*

- *$0 \leq \delta \leq 1/2k$ and $|S| = \Omega(\frac{\lg n}{\delta^2 k})$, or*
- *$1/2k \leq \delta \leq 1/4$ and $|S| = \Omega(\frac{\lg n}{\delta})$.*

By taking a union bound over all nodes $v \notin S$, we obtain the following straight-forward corollary.

Corollary 2.4. *Assume we have a seed of nodes S, such that for all $s \in S$, we know $g(s) + \alpha \bmod k$ for some (shared) cyclic shift $\alpha \in \{0, \ldots, k-1\}$. Then it is possible to recover $g(v) + \alpha \bmod k$ for all $v \in V$ in $O(n|S|)$ time whp. provided that $k \leq n^{o(1)}$ and either:*

- *$0 \leq \delta \leq 1/2k$ and $|S| = \Omega(\frac{\lg n}{\delta^2 k})$, or*
- *$1/2k \leq \delta \leq 1/4$ and $|S| = \Omega(\frac{\lg n}{\delta})$.*

Proof of Lemma 2.3. Each query $\tilde{f}(v,s)$ returns $(g(v) - g(s) + \eta_{vs}) \bmod k$. We thus have $(g(s) + \tilde{f}(v,s)) \bmod k = (g(v) + \eta_{vs}) \bmod k$. Therefore $(g(s) + \tilde{f}(v,s)) \bmod k = g(v)$ with probability $1/k + \delta$, and for every $i \in \{1, \ldots, k-1\}$, we have $(g(s) + \tilde{f}(v,s)) \bmod k = (g(v) + i) \bmod k$ with probability $1/k - \delta/(k-1)$. Using Lemma 2.1 and a union bound over all $i \in \{1, \ldots, k-1\}$, we thus conclude for $0 \leq \delta \leq 1/2k$, that the plurality vote equals $g(v)$ with probability at least $1 - kc_1 \exp(-\delta^2 |S| k / c_1)$ for a constant $c_1 > 0$. For $|S| = \Omega(\frac{\lg n}{\delta^2 k})$ and $k \leq n^{o(1)}$, this is at least $1 - 1/n^2$. Similarly we use Lemma 2.2 and a union bound over all $i \in \{1, \ldots, k-1\}$ to conclude for $1/2k \leq \delta \leq 1/4$, that the plurality vote equals $g(v)$ with probability at least $1 - kc_1 \exp(-\delta |S| / c_1)$ for a constant $c_1 > 0$. For $|S| = \Omega(\frac{\lg n}{\delta})$ and $k \leq n^{o(1)}$, this is at least $1 - 1/n^2$. ■

Given Corollary 2.4 it suffices to find a seed set $S \subseteq V$ and determine the labels of the nodes in S up to the same cyclic shift α. Our next lemma shows how to do so via queries $\tilde{f}(s,v)$ for nodes $s \in S$ and $v \in V \setminus S$. Our key idea is to determine the difference $(g(s) - g(s')) \bmod k$ for pairs $s, s' \in S$ via queries $\tilde{f}(s,b) - \tilde{f}(s',b)$ for nodes $b \in V \setminus S$.

Lemma 2.5 (Learning Pairwise Differences)**.** *Let $S \subseteq V$ be an arbitrary set of nodes and assume $k \leq n^{o(1)}$. Let $s, s' \in S$ be two distinct nodes. Define $Z_{s,s'}$ as the plurality vote among the answers $\{(\tilde{f}(s,b) - \tilde{f}(s',b)) \bmod k\}_{b \in V \setminus S}$. If $|V \setminus S| = \Omega(\frac{\lg n}{k\delta^4} + \frac{\lg n}{\delta^2})$, then $Z_{a,a'} = (g(a) - g(a')) \bmod k$ with probability at least $1 - \frac{1}{n^2}$.*

Proof. Since

$$\tilde{f}(s,b) - \tilde{f}(s',b) = (g(s) - g(s') \bmod k) + (\eta_{s,b} - \eta_{s',b} \bmod k),$$

we need to understand the probability distribution of $Z_b = \eta_{s,b} - \eta_{s',b} \bmod k$. Intuitively, we wish that the probability $\Pr[Z_b = 0]$ is greater enough than each $\Pr[Z_b = i]$ where $i \neq 0$ so that the plurality vote gives the correct estimate for $g(s) - g(s')$. Indeed,

$$\Pr[Z_b = 0] = \sum_{j=0}^{k-1} \Pr[\eta_{s,b} = \eta_{s',b} = j] = \Pr[\eta_{s,b} = \eta_{s',b} = 0] + \sum_{j=1}^{k-1} \Pr[\eta_{s,b} = \eta_{s',b} = j] =$$

$$= \Big(\frac{1}{k} + \delta\Big)^2 + (k-1)\Big(\frac{1}{k} - \frac{\delta}{k-1}\Big)^2 = \frac{1}{k} + \frac{k\delta^2}{k-1}.$$

Also Z_b is uniform over $1, \dots, k-1$ with the remaining probability, i.e. $\Pr[Z_b = i] = \frac{1}{k} - \frac{k\delta^2}{(k-1)^2}$ for $i \neq 0$. We thus obtain the exact same guarantees as in Lemma 2.3 with δ replaced by $\delta' = \frac{k\delta^2}{k-1}$. That is, if either

- $0 \leq \frac{k\delta^2}{k-1} \leq 1/2k$ and $|V \setminus S| = \Omega(\frac{(k-1)^2 \lg n}{k^3 \delta^4})$, or
- $1/2k \leq \frac{k\delta^2}{k-1} \leq 1/4$ and $|V \setminus S| = \Omega(\frac{(k-1) \lg n}{k\delta^2})$.

then the plurality vote among $\{(\tilde{f}(s,b) - \tilde{f}(s',b)) \bmod k\}_{b \in B}$ equals $(g(s) - g(s')) \bmod k$ with probability at least $1 - 1/n^2$. Combining the two, we conclude from the above that the plurality vote is correct with probability at least $1 - 1/n^2$ provided that $|V \setminus S| = \Omega(\frac{(k-1)^2 \lg n}{k^3\delta^4} + \frac{(k-1)\lg n}{k\delta^2}) = \Omega(\frac{\lg n}{k\delta^4} + \frac{\lg n}{\delta^2})$. ■

In light of the above, our proposed algorithm is thus to pick a set S and perform all queries between S and $V \setminus S$. Based on Lemma 2.3 and Lemma 2.5, we set $|S| = O(\frac{\lg n}{k\delta^2})$ when $0 \leq \delta \leq 1/2k$ and $|S| = O(\frac{\lg n}{\delta})$ when $1/2k \leq \delta \leq 1/4$. We then fix a node $s \in S$ and assign it the label $\hat{g}(s) = 0$. We thus have $\hat{g}(s) = (g(s) + (0 - g(s))) \bmod k$, i.e. $g(s)$ has been recovered up to a cyclic shift of $(0 - g(s))$. Our goal is to recover all other labels up to the same cyclic shift.

We now compute an estimate $\mu_{s'}$ of $(g(s) - g(s')) \bmod k$ for every $s' \in S \setminus \{s\}$ using a plurality vote on $\{(\tilde{f}(s,b) - \tilde{f}(s',b)) \bmod k\}_{b \in V \setminus S}$. A union bound over all nodes in S together with Lemma 2.5 shows that all these estimates are correct *whp*. We then assign the label $\hat{g}(s') = \mu_{s'}$ to all remaining nodes $s' \in S$. If all plurality votes were correct, then $\hat{g}(s') = \mu_{s'} = (g(s') - g(s)) \bmod k = (g(s') + (0 - g(s))) \bmod k$ for all s'. That is, we have recovered each $g(s')$ up to the same cyclic shift $(0 - g(s)) \bmod k$.

To recover the labels of all remaining nodes $v \in V \setminus S$ in the graph (up to the shift $(0 - g(s)) \bmod k$), we use a plurality vote on $\{\hat{g}(s') + \tilde{f}(v,s') \bmod k\}_{s' \in S} = \{g(s') + (0 - g(s)) + \tilde{f}(v,s') \bmod k\}_{s' \in S}$. Corollary 2.4 and a union bound over all nodes in $V \setminus S$ gives us that our algorithm recovers all labels *whp*. Our proposed algorithm is also shown in pseudocode, see Algorithm 1.

Algorithm 1 Learning Joint Alignment with a Faulty Oracle

```
Choose $S \subseteq V$ such that $|S| = O(\frac{\log n}{k\delta^2})$ if $0 \leq \delta \leq 1/2k$ and $|S| = O(\frac{\lg n}{\delta})$ if $1/2k \leq \delta \leq 1/4$.
Perform all queries between $S$ and $V \setminus S$.
Fix a node $s \in S$ and assign it the label $\hat{g}(s) = 0$.
For each $s' \in S \setminus \{s\}$, compute an estimate $\mu_{s'}$ of $(g(s') - g(s)) \bmod k$ using the plurality vote among the queries $\{\tilde{f}(s',b) - \tilde{f}(s,b)\}_{b \in V \setminus S}$ and assign $s'$ the label $\hat{g}(s') = \mu_{s'}$.
For each $v \notin V \setminus S$, assign it a label corresponding to the result of the plurality vote among $\{\hat{g}(s) + \tilde{f}(v,s)\}_{s \in S}$.
```

As a last remark, notice that we can only choose $|S| = O(\frac{\lg n}{k\delta^2})$ or $|S| = O(\frac{\lg n}{\delta})$ provided that $\frac{\lg n}{k\delta^2} = O(n)$ in the first case and $\frac{\lg n}{\delta} = O(n)$ in the second case. Assume first that indeed $|S| \leq n/2$. Then Lemma 2.5 further requires that $|V \setminus S| = \Omega(\frac{\lg n}{k\delta^4} + \frac{\lg n}{\delta^2})$. Since $|V \setminus S| \geq n/2$ when $|S| \leq n/2$, this translates into $\frac{\lg n}{k\delta^4} + \frac{\lg n}{\delta^2} = O(n)$. This is a more strict requirement than $\frac{\lg n}{k\delta^2} = O(n)$ and $\frac{\lg n}{\delta} = O(n)$. We can thus invoke our algorithm as long as $\delta = \Omega((\lg n/nk)^{1/4})$ and $\delta = \Omega(\sqrt{1/n})$. We assume $k \leq n^{o(1)}$, hence the dominating requirement is $\delta = \Omega((\lg n/nk)^{1/4})$.

The algorithm is completely non-adaptive, correct *whp.* and each plurality vote can be computed in linear time in the number of estimates involved. The total running time of the algorithm is thus $O(|V||S|)$ and so is the number of queries. When $0 \leq \delta \leq 1/2k$, this is $O(\frac{n \lg n}{\delta})$ and when $1/2k \leq \delta \leq 1/4$, this is $O(\frac{n \lg n}{k\delta^2})$.

Theorem 2.6. *If $(\lg n/nk)^{1/4} \leq \delta \leq 1/2k$ and $k \leq n^{o(1)}$, then there is a non-adaptive and deterministic query algorithm that makes $O(\frac{n \log n}{\delta^2 k})$ queries, runs in $O(\frac{n \log n}{\delta^2 k})$ time and is correct whp.*

If $1/2k \leq \delta \leq 1/4$ and $k \leq n^{o(1)}$, then there is a non-adaptive and deterministic query algorithm that makes $O(\frac{n \log n}{\delta})$ queries, runs in $O(\frac{n \log n}{\delta})$ time and is correct whp.

2.3 Lower bound

In this section, we complement our algorithm with a matching lower bound:

Theorem 2.7. *If $1/n^{1/4} \leq \delta \leq 1/2k$ and $k \leq n^{o(1)}$, then any non-adaptive and possibly randomized query algorithm making $o(\frac{n \log n}{\delta^2 k})$ queries has success probability at most* $\exp(-n^{\Omega(1)})$.

If $1/2k \leq \delta \leq 1/4$ and $k \leq n^{o(1)}$, then any non-adaptive and possibly randomized query algorithm making $o(\frac{n \log n}{\delta})$ queries has success probability at most $\exp(-n^{\Omega(1)})$.

Let n be the number of vertices and consider a (possibly randomized) non-adaptive query algorithm $\mathcal{A}$, i.e. an algorithm that chooses the set of queries to make before seeing the results of the queries. Let ε be the success probability of $\mathcal{A}$, that is, for any latent function g, $\mathcal{A}$ recovers g (up to a cyclic rotation of the labels) with probability at least ε. Let t be the number of queries made by $\mathcal{A}$. The choice of queries is allowed to be randomized. Our goal is to show that ε is small if t is small.

Hard Distribution. We start by defining a hard distribution. Let $\mathbf{g}$ be a random latent function that assigns label 0 to the first vertex and a uniform random and independently chosen label in $\{0, \ldots, k-1\}$ to the remaining vertices.

Simplifying $\mathcal{A}$. Our first step is to simplify $\mathcal{A}$ for a cleaner analysis. Recall that a correct algorithm is allowed to return any cyclic rotation of the latent function g, i.e. any labeling that is equal to g up to adding the same constant mod k to all labels. Under our hard distribution $\mathbf{g}$, we always have that the first vertex has label 0. Therefore, we can define a new algorithm $\mathcal{A}^1$ which makes the same queries as $\mathcal{A}$, but when returning an assignment of labels, $\mathcal{A}^1$ takes the output of $\mathcal{A}$ and subtracts the label assigned by $\mathcal{A}$ to the first vertex from every single output label, mod k. In this way, for every $g \in \text{supp}(\mathbf{g})$, we get that $\mathcal{A}^1$ returns g whenever $\mathcal{A}$ is correct up to a cyclic rotation. That is, we now have an algorithm $\mathcal{A}^1$ that makes t queries and has success probability ε for any $g \in \text{supp}(\mathbf{g})$, even if we define success as returning the exact labeling (i.e. no cyclic shifts allowed). Our next simplifying step is to derandomize $\mathcal{A}^1$. By fixing the random coins of $\mathcal{A}^1$ (easy direction of Yao's principle), we obtain a deterministic algorithm $\mathcal{A}^2$ that makes t non-adaptive queries and is correct with probability ε *over the random choice of* $\mathbf{g}$. Since $\mathcal{A}^2$ is deterministic and non-adaptive, we can let E be the set of edges queried by $\mathcal{A}^2$ and let $\mathbf{f} \in E \to \{0, \ldots, k-1\}$ give the (random) results of the queries E.

We wish to simplify $\mathcal{A}^2$ even further by making assumptions about the labeling it returns when seeing a set of answers $f \in \text{supp}(\mathbf{f})$ to queries. Let S denote the event that $\mathcal{A}^2$ is correct. Then

$$\Pr[S] = \sum_{f \in \text{supp}(\mathbf{f})} \Pr[\mathbf{f} = f] \Pr[S \mid \mathbf{f} = f].$$

Since $\mathcal{A}^2$ is deterministic, it outputs a concrete labeling $\mathcal{A}^2(f)$ for any $f \in \mathbf{f}$. Thus

$$\Pr[S \mid \mathbf{f} = f] = \Pr[\mathbf{g} = \mathcal{A}^2(f) \mid \mathbf{f} = f].$$

Let $G(f)$ be the collection of all maximum likelihood labelings $g \in \mathbf{g}$ conditioned on $\mathbf{f} = f$, i.e. $G(f)$ contains all $g \in \operatorname{supp}(\mathbf{g})$ such that $\Pr[\mathbf{g} = g \mid \mathbf{f} = f] \geq \Pr[\mathbf{g} = g' \mid \mathbf{f} = f]$ for all $g' \in \operatorname{supp}(\mathbf{g})$. The above allows us to conclude that if we define the algorithm $\mathcal{A}^*$ which makes the same queries as $\mathcal{A}^2$, but always returns a uniform random $g \in G(\mathbf{f})$, then $\mathcal{A}^*$'s success probability is at least ε. This completes our simplifying steps and we will show that $\mathcal{A}^*$ has small success probability if t is small.

Performance of $\mathcal{A}^*$. To prove that $\mathcal{A}^*$ has low success probability if it makes few queries, we will show that there is a good chance that the correct labeling $\mathbf{g}$ is not the maximum likelihood estimate after seeing $\mathbf{f}$. For this, consider a vertex v different from the first vertex and let E_v be the subset of edges from E that have v as an end point. Since each edge in E has two end points, there must be a set W of at least $n/2$ vertices that have $|E_v| \leq 4t/n$. We form an independent set I from W by repeatedly selecting one vertex v from W and adding it to an initially empty I. We then remove all vertices incident to v from W. Since each v removes at most $4t/n$ other vertices from W, we are left with an I of size at least $(n-1)/(4t/n+1)$. The reason why we choose I as an independent set is that it implies that the queries corresponding to edges incident to a node $v \in I$ are independent of the queries incident to any other node $w \in I$.

Now let $f \in \operatorname{supp}(\mathbf{f})$ be an assignment to the edges and let $g \in \operatorname{supp}(\mathbf{g})$ be a classification of the vertices. Define from f and g the noise on edge $(u,v) \in E_v$ as $\eta_{uv}^{fg} = (g(u) - g(v) - f(u,v)) \bmod k$. For each $i \in \{0, \ldots, k-1\}$, define $c_v^{fg}(i)$ as the number of edges (u,v) incident to v for which $\eta_{uv}^{fg} = i$. Define the subset $I_{fg}^* \subseteq I$ containing all vertices $v \in I$ such that $c_v^{fg}(1) \geq c_v^{fg}(0)$. We claim that there are at least $2^{|I_{fg}^*|}$ distinct labelings $g' \in \operatorname{supp}(\mathbf{g})$ that all have $\Pr[\mathbf{g} = g' \mid \mathbf{f} = f] \geq \Pr[\mathbf{g} = g \mid \mathbf{f} = f]$. To see this, consider any labeling g' where $g'(v) = g(v)$ for $v \notin I_{fg}^*$ and either $g'(v) = g(v) - 1$ or $g'(v) = g(v)$ for $v \in I_{fg}^*$. There are $2^{|I_{fg}^*|}$ such g'. We will prove that $\Pr[\mathbf{g} = g' \mid \mathbf{f} = f] \geq \Pr[\mathbf{g} = g \mid \mathbf{f} = f]$. For a classification $g \in \operatorname{supp}(\mathbf{g})$ and assignment to the edges $f \in \operatorname{supp}(\mathbf{f})$, define E_{fg}^+ as the subset of edges for which $(g(u) - g(v) - f(u,v)) \bmod k = 0$ and let $E_{fg}^- = E \setminus E_{fg}^+$. Since the noises on the edges are independent, it follows that

$$\Pr[\mathbf{f} = f \mid \mathbf{g} = g] = \left(\frac{1}{k} + \delta\right)^{|E_{fg}^+|} \left(\frac{1}{k} - \frac{\delta}{k-1}\right)^{|E_{fg}^-|}$$

Comparing g' and g, we notice that all edges (u,w) with $v \notin \{u,w\}$ contribute the same to $\Pr[\mathbf{f} = f \mid \mathbf{g} = g]$ and $\Pr[\mathbf{f} = f \mid \mathbf{g} = g']$. However, for g', it holds that any edge where $g(v) - g(u) \bmod k = 1$ we now have $g'(v) - g'(u) \bmod k = g(v) - 1 - g(u) \bmod k = 0$. Hence $c_v^{fg'}(0) = c_v^{fg}(1) \geq c_v^{fg}(0)$. It follows that $\Pr[\mathbf{f} = f \mid \mathbf{g} = g'] \geq \Pr[\mathbf{f} = f \mid \mathbf{g} = g]$. Using Bayes' theorem, we get

$$\Pr[\mathbf{g} = g \mid \mathbf{f} = f] = \frac{\Pr[\mathbf{f} = f \mid \mathbf{g} = g] \Pr[\mathbf{g} = g]}{\Pr[\mathbf{f} = f]}.$$

and

$$\Pr[\mathbf{g} = g' \mid \mathbf{f} = f] = \frac{\Pr[\mathbf{f} = f \mid \mathbf{g} = g'] \Pr[\mathbf{g} = g']}{\Pr[\mathbf{f} = f]}.$$

Since $\mathbf{g}$ is uniform over its support, we have $\Pr[\mathbf{g} = g] = \Pr[\mathbf{g} = g']$. Hence we conclude that

$$\Pr[\mathbf{g} = g' \mid \mathbf{f} = f] \geq \Pr[\mathbf{g} = g \mid \mathbf{f} = f]$$

as claimed.

The above implies that $\mathcal{A}^*$ outputs g with probability at most $2^{-|I^*_{fg}|}$ when it sees the query answers f. Indeed, if there is even a single g' with $\Pr[\mathbf{g} = g' \mid \mathbf{f} = f] > \Pr[\mathbf{g} = g \mid \mathbf{f} = f]$, then $\mathcal{A}^*$ never outputs g, and otherwise, $\mathcal{A}^*$ outputs a uniform random labeling among the $2^{|I^*_{fg}|}$ candidates. To upper bound the succes probability of $\mathcal{A}^*$, we thus argue that $I^*_{\mathbf{fg}}$ is large with high probability when t is small.

Assume first that $1/n^{1/4} \leq \delta \leq 1/2k$ and $k \leq n^{o(1)}$. Using Lemma 2.1, each $v \in I$ is included in $I^*_{\mathbf{fg}}$ with probability at least $c_2^{-1} \exp(-\delta^2 |E_v| k c_2)$ for a constant $c_2 > 0$. Since $|E_v| \leq 4t/n$, it follows that for $t = o((n \lg n)/(k\delta^2))$, v will appear in $I^*_{\mathbf{fg}}$ with probability at least $n^{-o(1)}$. Furthermore, $|I| \geq (n-1)/(4t/n+1) = \Omega(\delta^2 nk/\lg n) = \Omega(n^{1/3})$. Moreover, these events are independent for different $v \in I$ since I forms an independent set. A Chernoff bound implies that $|I^*_{\mathbf{fg}}| \geq c_2^{-1} \exp(-\delta^2 |E_v| k c_2)|I|/2 \geq n^{1/3-o(1)}$ with probability at least $1 - \exp(-n^{1/3-o(1)})$. When this event B happens, the conditional success probability is no more than $\exp(-n^{1/3-o(1)})$. Hence the overall success probability is at most $\exp(-n^{1/3-o(1)}) \Pr[B] + (1 - \Pr[B]) = \exp(-n^{\Omega(1)})$.

Assume next that $1/2k \leq \delta \leq 1/4$ and $k \leq n^{o(1)}$. Using Lemma 2.2, each $v \in I$ occurs in $I^*_{\mathbf{fg}}$ with probability at least $c_2^{-1} \exp(-\delta |E_v| c_2)$ for a constant $c_2 > 0$. Since $|E_v| \leq 4t/n$, it follows that for $t = o((n \lg n)/\delta)$, v will appear in $I^*_{\mathbf{fg}}$ with probability at least $n^{-o(1)}$. We also have $|I| \geq (n-1)/(4t/n+1) = \Omega(\delta n/\lg n) = \Omega(n^{1/3})$. A Chernoff bound like above concludes that the success probability is no more than $\exp(-n^{\Omega(1)})$.

2.4 Concentration Inequalities

In this section, we prove the two concentration inequalities stated in Section 2.1. Our proofs use the standard Chernoff bounds as well as the following "reverse" Chernoff bound:

Theorem 2.8 ([Mou10]). *Let $C_1, \dots, C_m$ be i.i.d. 0/1 random variables with $\Pr[C_i = 1] = p$. For $p \leq 1/2$ and for any $0 \leq t \leq m(1-2p)$ it holds that:*

$$\Pr[\sum_{i=1}^{m} C_i \geq t + pm] \geq \frac{1}{4} \exp(-2t^2/pm).$$

We start by proving Lemma 2.1:

Proof of Lemma 2.1. For $0 \leq \delta \leq 1/2k$ and $n \geq k/2$, we first upper bound $\Pr[\sum_i X_i \leq 0]$. Let Y_i take the value 1 if X_i takes the value 1 and 0 otherwise. Let Z_i take the value 1 if $X_i = -1$ and 0 otherwise. By a Chernoff bound with $(1-\varepsilon)(1/k+\delta) = (1/k + \delta/2) \Rightarrow \varepsilon = \delta/(2(1/k+\delta))$, we get $\Pr[\sum_i Y_i \leq (1/k + \delta/2)n] \leq \exp(-\varepsilon^2(1/k+\delta)n/2)$. This is at most $\exp(-\delta^2 n/(8(1/k+\delta))) \leq \exp(-\delta^2 nk/8)$. Similarly, a Chernoff bound with $(1+\varepsilon)(1/k - \delta/(k-1)) = (1/k + \delta/2) \Rightarrow \varepsilon = (\delta/2 + \delta/(k-1))/(1/k - \delta/(k-1)) \geq \delta/(2(1/k - \delta/(k-1)))$, we have $\Pr[\sum_i Z_i \geq (1/k+\delta/2)n] \leq \exp(-\varepsilon^2(1/k - \delta/(k-1))n/3) \leq \exp(-\delta^2 n/(12(1/k - \delta/(k-1)))) \leq \exp(-\delta^2 nk/12)$. A union bound gives $\Pr[\sum_i X_i \leq 0] \leq 2\exp(-\delta^2 nk/12)$. If $n \leq k/2$, then $\delta^2 nk < 1$ and the statement follows trivially since there is a constant c_1 making $c_1 \exp(-\delta^2 nk/c_1)$ greater than or equal to 1.

To lower bound $\Pr[\sum_i X_i \leq 0]$, let $W = \sum_i (Y_i + Z_i)$. Conditioned on $W = m$, we have that $\sum_i X_i$ is distributed as the sum of m i.i.d. random variables taking the value 1 with probability $(1/k + \delta)/(2/k + \delta k/(k-1)) \leq 1/2 + \delta k/2$ and the value -1 with probability at least $1/2 - \delta k/2$. We will use the following "reverse" Chernoff bound:

Conditioned on $W = m > 0$, we ask what is the probability that we see at least $\lceil m/2 \rceil$ -1's, i.e. $\sum_i X_i \leq 0$. Fixing $t = \lceil m/2 \rceil - (1/2 - \delta k/2)m \leq \delta k m/2 + 1$ we see that

$$\Pr[\sum_i X_i \leq 0 \mid W = m] \geq \frac{1}{4}\exp(-2(\delta k m/2 + 1)^2/(1/2 - \delta k/2)m) \geq$$
$$\frac{1}{4}\exp(-8(\delta k m/2 + 1)^2/m) \geq \frac{1}{4}\exp(-16(\delta^2 k^2 m/4 + 1)).$$

Using that $\mathbb{E}[W] = (2/k + \delta k/(k-1))n$, Markov's inequality gives us that $W \leq (4/k + 2\delta k/(k-1))n \leq 8n/k$ with probability at least $1/2$. We also have $\sum_i X_i = 0$ when $\sum_i W_i = 0$. Hence

$$\Pr[\sum_i X_i \leq 0] \geq \frac{1}{8}\exp(-32(\delta^2 k n + 1)) \geq \frac{1}{8 \cdot e^{-32}}\exp(-32\delta^2 n k).$$

■

Next we prove Lemma 2.2:

Proof of Lemma 2.2. We start by upper bounding $\Pr[\sum_i X_i \leq 0]$. Let Y_i take the value 1 if X_i takes the value 1 and 0 otherwise. Let Z_i take the value 1 if $X_i = -1$ and 0 otherwise. A Chernoff bound gives

$$\Pr[\sum_i Y_i \leq (1/k + \delta/2)n] \leq \exp(-\delta^2 n/(8(1/k + \delta))) \leq \exp(-\delta n/8).$$

Similarly, we have

$$\Pr[\sum_i Z_i \geq (1/k + \delta/2)n] \leq \exp(-\delta^2 n/(12(1/k - \delta/(k-1)))) \leq \exp(-\delta^2 n/(12(1/k - 1/3(k-1))))$$
$$\leq \exp(-\delta^2 n/(12(1/k - 2/3k))) = \exp(-\delta^2 k n/4) \leq \exp(-\delta n/8)$$

A union bound gives $\Pr[\sum_i X_i \leq 0] \leq 2\exp(-\delta n/8)$. To lower bound $\Pr[\sum_i X_i \leq 0]$, first notice that

$$\Pr[\sum_i Y_i = 0] = (1 - 1/k - \delta)^n \geq (1 - 3\delta)^n = \exp(-n\sum_{j=1}^{\infty}(3\delta)^j/j) \geq$$
$$\geq \exp(-n(3\delta)\sum_{j=0}^{\infty}(3/4)^j) = \exp(-12\delta n).$$

We conclude that $\Pr[\sum_i X_i \leq 0] \geq \exp(-12\delta n)$. ■

3 Conclusion

In this work we provide an optimal algorithm both in terms of running time and query complexity for the problem of learning joint alignments with a faulty oracle. The algorithm is simple and performs well in practice compared to previous work. An interesting open problem is to explore whether there exists an adaptive algorithm with better query complexity. Finally, a remaining open question from Chen and Candés is whether we can characterize the performance of existing joint alignment algorithms if one is satisfied with approximate solutions.

References

[AS15] Emmanuel Abbe and Colin Sandon. Community detection in general stochastic block models: Fundamental limits and efficient algorithms for recovery. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 670–688. IEEE, 2015.

[CC16] Yuxin Chen and Emmanuel Candes. The projected power method: An efficient algorithm for joint alignment from pairwise differences. *arXiv preprint arXiv:1609.05820*, 2016.

[CGH14] Yuxin Chen, Leonidas J Guibas, and Qi-Xing Huang. Near-optimal joint object matching via convex relaxation. *arXiv preprint arXiv:1402.1473*, 2014.

[HFG$^+$06] Qi-Xing Huang, Simon Flöry, Natasha Gelfand, Michael Hofer, and Helmut Pottmann. Reassembling fractured objects by geometric matching. *ACM Transactions on Graphics (TOG)*, 25(3):569–578, 2006.

[HG13] Qi-Xing Huang and Leonidas Guibas. Consistent shape maps via semidefinite programming. In *Computer Graphics Forum*, volume 32, pages 177–186. Wiley Online Library, 2013.

[HH02] Daniel F Huber and Martial Hebert. *Automatic three-dimensional modeling from reality.* PhD thesis, Citeseer, 2002.

[HSG13] Qi-Xing Huang, Hao Su, and Leonidas Guibas. Fine-grained semi-supervised labeling of large shape collections. *ACM Transactions on Graphics (TOG)*, 32(6):190, 2013.

[HWX16] Bruce Hajek, Yihong Wu, and Jiaming Xu. Achieving exact cluster recovery threshold via semidefinite programming. *IEEE Transactions on Information Theory*, 62(5):2788–2797, 2016.

[McS01] Frank McSherry. Spectral partitioning of random graphs. In *42nd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 529–537. IEEE, 2001.

[Mou10] Nima Mousavi. How tight is the chernoff bound? `https://ece.uwaterloo.ca/~nmousavi/Papers/Chernoff-Tightness.pdf`, 2010.

[MS17] Arya Mazumdar and Barna Saha. Clustering with noisy queries. *arXiv preprint arXiv:1706.07510*, 2017.

[MT18] Michael Mitzenmacher and Charalampos E Tsourakakis. Joint alignment from pairwise differences with a noisy oracle. In *International Workshop on Algorithms and Models for the Web-Graph*, pages 59–69. Springer, 2018.

[Sin11] Amit Singer. Angular synchronization by eigenvectors and semidefinite programming. *Applied and computational harmonic analysis*, 30(1):20–36, 2011.

[SS12] Yoel Shkolnisky and Amit Singer. Viewing direction estimation in cryo-em using synchronization. *SIAM journal on imaging sciences*, 5(3):1088–1110, 2012.

[TML$^+$17] Charalampos E Tsourakakis, Michael Mitzenmacher, Kasper Green Larsen, Jarosław Błasiok, Ben Lawson, Preetum Nakkiran, and Vasileios Nakos. Predicting positive and negative links with noisy queries: Theory & practice. *arXiv preprint arXiv:1709.07308*, 2017.

[Tso15] Charalampos Tsourakakis. Streaming graph partitioning in the planted partition model. In *Proceedings of the 2015 ACM on Conference on Online Social Networks*, pages 27–35. 2015.

[WS13] Lanhui Wang and Amit Singer. Exact and stable recovery of rotations for robust synchronization. *Information and Inference: A Journal of the IMA*, 2(2):145–193, 2013.

[ZB18] Yiqiao Zhong and Nicolas Boumal. Near-optimal bounds for phase synchronization. *SIAM Journal on Optimization*, 28(2):989–1016, 2018.