Learning the Truth Vector in High Dimensions[☆]

Hu Ding[a,*], Jinhui Xu[b]

[a] *School of Computer Science and Technology*
University of Science and Technology of China
He Fei, 230027, China
[b] *Department of Computer Science and Engineering*
State University of New York at Buffalo
Buffalo, NY 14260, USA

Abstract

Truth Discovery is an important learning problem arising in data analytics related fields. It concerns about finding the most trustworthy information from a dataset acquired from a number of unreliable sources. The problem has been extensively studied and a number of techniques have already been proposed. However, all of them are of heuristic nature and do not have any quality guarantee. In this paper, we formulate the problem as a high dimensional geometric optimization problem, called *Entropy based Geometric Variance*. Relying on a number of novel geometric techniques, we further discover new insights to this problem. We show, for the first time, that the truth discovery problem can be solved with guaranteed quality of solution. Particularly, it is possible to achieve a $(1+\epsilon)$-approximation within nearly linear time under some reasonable assumptions. We expect that our algorithm will be useful for other data related applications.

Keywords: Truth Discovery, Entropy, High Dimension, Approximation Algorithm

[☆] This work was partially supported by NSF through grants CCF-1422324, IIS-1422591, CNS-1547167, CCF-1716400, IIS-1910492, and CCF-1656905. Part of the work was done when the first author was in IIIS, Tsinghua University. The first author was also supported by a start-up fund from Michigan State University. A preliminary version of this paper has appeared in the 32nd International Symposium on Computational Geometry (SoCG 2016)[1].

[*] Corresponding author

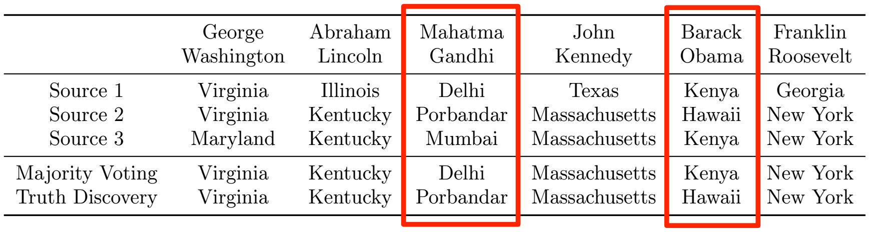
Email addresses: huding@ustc.edu.cn (Hu Ding), jinhui@buffalo.edu (Jinhui Xu)

1. Introduction

Truth discovery is an emerging topic in data analytics which has received a great deal of attentions in recent years [2]. Despite its extensive studies in the fields of data mining, database, machine learning, and big data [2, 3, 4, 5, 6, 7, 8], it has yet to be seriously considered by the theory community (to our best knowledge). The problem arises in scenarios where data are acquired from multiple sources which may contain false or inconsistent information, and the truth discovery problem is to find the most trustworthy information from these sources. The problem finds many applications in different areas, such as healthcare [9], crowd/social sensing [10, 11], and knowledge bases aggregation [12]. For example, in online social networks, a user's information can be recorded by multiple websites which may not be always consistent; thus it is desirable to find the most trustworthy information for each user. Similar problem also occurs in healthcare where medical records of a patient may be acquired by multiple hospitals or laboratories.

The main challenge of truth discovery comes from its unsupervised nature, *i.e.,* the level of reliability of each source is unknown in advance. A straightforward way for solving the problem is to take the average if the data are continuous or conduct majority voting if the data are categorical. Such approaches are implicitly based on the assumption that all sources are equally reliable. However, in many applications the level of reliability of each source could be quite different which may make the yielded solution significantly different from the truth, due to the neglect of "the wisdom of minority" [2, 3]. See the example in Figure 1 from [2]. Thus, estimating the reliability of each source should be taken into account when building the optimization model for truth discovery. In general, the two components, **reliability estimation** and **truth finding**, are tightly coupled and thus are expected to be solved simultaneously, where the truth should be closer to the source with higher reliability, and as a feedback, the source providing closer information should have a higher reliability. Another challenge of truth discovery is how to handle large-scale datasets when the number of sources and the data size of each source are both large.

A closely related research topic is *crowdsourcing aggregation*. A well known crowdsourcing platform is Amazon Mechanical Turk which provides a cost-efficient way to solicit labels from crowd workers [13]. Many mechanisms and inference algorithms have been developed for inferring true labels and workers' quality [14, 15, 16, 17, 18, 19]. As mentioned in [2], the main difference between crowdsourcing aggregation and truth discovery is that the former is an active procedure (one can control what and how much data to be generated by workers) while the latter is a passive procedure (one can only choose from available data sources).



	George Washington	Abraham Lincoln	Mahatma Gandhi	John Kennedy	Barack Obama	Franklin Roosevelt
Source 1	Virginia	Illinois	Delhi	Texas	Kenya	Georgia
Source 2	Virginia	Kentucky	Porbandar	Massachusetts	Hawaii	New York
Source 3	Maryland	Kentucky	Mumbai	Massachusetts	Kenya	New York
Majority Voting	Virginia	Kentucky	Delhi	Massachusetts	Kenya	New York
Truth Discovery	Virginia	Kentucky	Porbandar	Massachusetts	Hawaii	New York

Figure 1: Three sources are providing the birthplaces of 6 politicians. For *Mahatma Gandhi*, each source has an individual answer (*i.e.,* a tie case), and majority voting can only randomly pick one. More importantly, for *Barack Obama*, voting provides a totally wrong answer. However, truth discovery tries to distinguish reliable and unreliable sources and thus provide the right answer. In this example, the algorithm in [2] finds that source 2 has a higher reliability than the other two.

1.1. Problem Formulation and Existing Approaches

We first introduce the problem formulation of truth discovery used in the data mining community, and then convert it to a new geometric optimization problem, called **entropy based geometric variance**.

To model the truth discovery problem, the data from each source can be represented as a (possibly high dimensional) vector, where each dimension corresponds to one attribute/property (*e.g.,* age, income, or temperature). For categorical data, we can reduce them to continuous data as follows [2]. Suppose that one attribute has t categories; then it can be represented as a t-dimensional binary sub-vector, where each dimension indicates the membership of one category. We can finally embed all these sub-vectors (corresponding to the categorical attributes) into one unified vector in higher dimensional space. Note that this representation for categorical data may cause fractional memberships in the final solution, which is often acceptable in practice (*e.g.,* we may claim that one object belongs to class 1, 2, and 3 with probabilities of 70%, 20%, and 10%, respectively). Furthermore, we need a variable to represent the reliability of each source.

Definition 1 (Truth Discovery[4, 2])**.** *Let $P = \{p_1, p_2, \cdots, p_n\}$ be a set of vectors in $\mathbb{R}^d$ space with each p_i representing the data from the i-th source (among a set of n sources). The truth discovery problem is to find the truth vector $p^* \in \mathbb{R}^d$ and the reliability (weight) w_i for each i-th source, such that the following objective function is minimized,*

$$\sum_{i=1}^{n} w_i ||p^* - p_i||^2, \quad s.t. \quad \sum_{i=1}^{n} e^{-w_i} = 1. \tag{1}$$

In the above optimization problem (1), both p^* and the weights are variables. It is easy to see that when each w_i is fixed, p^* is simply the weighted mean, *i.e.,* $\frac{1}{\sum_{i=1}^{n} w_i} \sum_{i=1}^{n} w_i p_i$. This means that the higher the weight of p_i, the closer it is to p^*, which is consistent with the principle of truth discovery.

Weight normalization function. In the above optimization problem, equation $\sum_{i=1}^{n} e^{-w_i} = 1$ is used to normalize the source weights. This way of normalization was initially introduced in [4] (with no justification) and has demonstrated experimentally its superior performance. To understand the rationale behind this, below we give a theoretical justification. Firstly, we notice that some straightforward ways, such as $\sum_{i=1}^{n} w_i^p = 1$ for some $p > 0$, are inappropriate for weight normalization [4], since otherwise, p^* can trivially choose any p_l as its solution and set $w_l = 1$ and $w_i = 0$ for all $i \neq l$ (in this way the objective value will always be equal to the smallest possible value 0). By using equation $\sum_{i=1}^{n} e^{-w_i} = 1$, we can easily avoid this issue. Secondly, this exponential normalization function ensures that the resulting solution minimizes the **entropy**, which implies that the solution contains more information from the input according to Shannon's information theory [20]. To see this, we first borrow the following lemma from [4], which can be shown by using the *Lagrange multipliers* method.

Lemma 1 ([4]). *If the truth vector p^* is fixed, the following value for each weight w_l minimizes the the objective function (1),*

$$w_l = \log(\frac{\sum_{i=1}^{n} ||p^* - p_i||^2}{||p^* - p_l||^2}). \tag{2}$$

Let S denote the total squared distance to p^* (*i.e.*, $S = \sum_{i=1}^{n} ||p^* - p_i||^2$), and f_l denote the contribution of each p_l to S (*i.e.*, $f_l = \frac{||p^* - p_l||^2}{S}$). Then the induced entropy is

$$\begin{aligned} H &= -\sum_{l=1}^{n} f_l \log f_l \\ &= -\sum_{l=1}^{n} \frac{||p^* - p_l||^2}{S} \log \frac{||p^* - p_l||^2}{S} \\ &= \frac{1}{S} \sum_{l=1}^{n} ||p^* - p_l||^2 \log \frac{S}{||p^* - p_l||^2}. \end{aligned} \tag{3}$$

Below, we define the *Entropy based Geometric Variance.*

Definition 2. *Given a set of points P and a point p^* in $\mathbb{R}^d$, the entropy base geometric variance induced by p^* is $H \times S$, where H and S are respectively the entropy and variance defined in the above discussion.*

From Lemma 1 and the formula (3), we know that the objective function (1) is equal to the multiplication of S and H, *i.e.,* the entropy based geometric variance.

Theorem 1. *The optimization problem (1) is equivalent to finding a point p^* to minimize the entropy based geometric variance.*

Generally speaking, S represents the total variance from the sources to the truth vector, and the entropy H indicates how disorder the system is, where the higher the entropy, the greater disorder the system is. Since we minimize both of them, this implicitly explains the better performance of using the exponential normalization function in Definition 1.

Non-convexity. As shown in [4], when the truth vector or the weights are fixed, the optimization problem (1) is convex. However, when both of them are variables, the problem is non-convex in general. To see this, consider the following simple example. Suppose $n = 2$. Then the objective value is 0 when p^* coincides with either p_1 or p_2, according to Lemma 1 (note $\lim_{x \to 0} x \log(1/x) = 0$). This means that it is possible to have multiple isolated local or global optimal solutions for truth discovery, implying that truth discovery is non-convex.

Existing Approaches. To the best of our knowledge, all existing methods for truth discovery are based on some heuristic ideas, which achieve only a local optimal solution and have no quality guarantee on global optimality. A commonly used strategy is alternating minimization (or expectation-maximization) [4, 5, 10], which alternatively fixes either the weights or the truth vector, and optimizes the other. The optimization problem becomes convex when one of the two types of variables is fixed. This means that such approaches are guaranteed to converge to some local optima. Other approaches follow similar ideas and the reader is referred to a recent survey [2] for a comprehensive introduction to these approaches. Recently, Xiao *et al.* [7] showed an expectation-maximization based algorithm with quality guarantee, but their algorithm needs some strict probabilistic assumption on the input and requires the number of sources to be large enough. For general case of the truth discovery problem, it is still an open problem for bounding the errors of the alternating minimization and expectation-maximization methods [2].

The methods of alternating minimization and expectation-maximization are very common optimization techniques that have been extensively studied in the past. For example, Jain *et al.* [21] and Hardt [22] considered using alternating minimization to solve the matrix completion problem; Jain and Tewari [23] studied the performance of alternating minimization for regression problems. We also refer the reader to the survey [24] for more details of the expectation-maximization algorithms.

1.2. Preliminaries and Our Main Results

Different from existing approaches, our goal is to achieve a quality guaranteed solution for the truth discovery problem. In general, we assume that the number of sources n and the size of the data in each source d are both large. As a starting point, the following theorem suggests that it is easy to generate a 2-approximation in quadratic time (see Appendix for the proof).

Theorem 2. *If one tries every point in $\{p_i \mid 1 \leq i \leq n\}$ as a candidate for the truth vector, at least one yields a 2-approximation for the objective function in (1), and the total running time is $O(n^2 d)$.*

Theorem 2 implies that any further improvement needs to decrease either the approximation ratio or the running time. In this paper, we aim to achieve a $(1+\epsilon)$-approximation for truth discovery and also keep the time complexity as low as possible.

For ease of discussion, we use the following notations throughout the rest of this paper. Let $L_{\min} = \min\{||p_i - p_{i'}|| \mid 1 \leq i \neq i' \leq n\}$, $L_{\max} = \max\{||p_i - p_{i'}|| \mid 1 \leq i \neq i' \leq n\}$, and the spread ratio $\Delta = \frac{L_{\max}}{L_{\min}}$. To achieve a $(1+\epsilon)$-approximation for the truth discovery problem for any given small value $1 > \epsilon > 0$, we consider the following two cases.

- **Case 1.** $\min_{1 \leq i \leq n} ||p^* - p_i|| \leq \frac{\epsilon\sqrt{S}}{4\sqrt{n}\Delta}$, *i.e.,* some p_i locates very close to p^*.

- **Case 2.** $\min_{1 \leq i \leq n} ||p^* - p_i|| > \frac{\epsilon\sqrt{S}}{4\sqrt{n}\Delta}$, *i.e.,* no p_i locates very close to p^*.

In following sections, we will present efficient algorithms to solve the two cases separately. For case 1, we show that the nearest point p_i to p^* is actually a $(1+\epsilon)$-approximation (Section 2). For case 2, we first give a simple linear time algorithm with large approximation ratio (Section 3), based on an analysis on the distribution of the weights; then in Section 4, we reveal several new insights to the weights by using a novel *Log-Partition* technique, and perform a sequence of geometric operations to obtain a $(1+\epsilon)$-approximation. The time complexity depends on Δ. Note that spread ratio is commonly used as a parameter in many geometric algorithms and appears in the time complexity (such as [25]). Finally, in Section 5 we show that when Δ is not too large, the time complexity for both cases can be improved to nearly linear ($O(nd \times poly(\log n))$); also through dimension reduction, the complexity can be further improved to linear ($O(nd)$).

We introduce the following two folklore lemmas [26, 27] which are repeatedly used in our analysis. For the completeness we show their proofs in Appendix. Let $Q = \{q_i \mid 1 \leq i \leq n\}$ be a set of n points in $\mathbb{R}^d$ with each q_i associated with a weight $w_i \geq 0$, $W = \sum_{i=1}^{n} w_i$, and $m(Q)$ be the weighted mean of Q *i.e.*, $m(Q) = \sum_{i=1}^{n} w_i q_i / W$.

Lemma 2. *For an arbitrary point q, $\sum_{i=1}^{n} w_i ||q - q_i||^2 = W||q - m(Q)||^2 + \sum_{i=1}^{n} w_i ||m(Q) - q_i||^2$.*

Lemma 3. *Let Q_1 be a subset of Q with a total weight of αW for some $0 < \alpha \leq 1$, and $m(Q_1)$ be the weighted mean point of Q_1. Then $||m(Q_1) - m(Q)|| \leq \sqrt{\frac{1-\alpha}{\alpha}}\delta$, where $\delta^2 = \frac{1}{W}\sum_{i=1}^{n} w_i ||q_i - m(Q)||^2$.*

2. A $(1+\epsilon)$-Approximation for Case 1

In this section, we consider case 1. Without loss of generality, we assume that $||p^* - p_{i_0}|| \leq \frac{\epsilon\sqrt{S}}{4\sqrt{n}\Delta}$, *i.e.*, p_{i_0} is the point very close to p^*. Then, we have:

Lemma 4. *For any $i \neq i_0$, $||p^* - p_i|| \geq (1 - \frac{\epsilon}{4})||p_{i_0} - p_i||$.*

Proof. Since p^* is the weighted mean $\frac{\sum_{i=1}^{n} w_i p_i}{\sum_{i=1}^{n} w_i}$, we know that for any $1 \leq l \leq n$,

$$||p^* - p_l|| = ||\frac{\sum_{i=1}^{n} w_i p_i}{\sum_{i=1}^{n} w_i} - p_l|| \leq \sum_{i=1}^{n} (\frac{w_i}{\sum_{i=1}^{n} w_i})||p_i - p_l|| \leq L_{\max}. \tag{4}$$

Thus, we have $S \leq nL_{\max}^2$, and consequently

$$||p^* - p_{i_0}|| \leq \frac{\epsilon\sqrt{S}}{4\sqrt{n}\Delta} \leq \frac{\epsilon}{4}\frac{L_{\max}}{\Delta} = \frac{\epsilon}{4}L_{\min}. \tag{5}$$

Furthermore, due to triangle inequality, we have

$$\begin{aligned} ||p^* - p_i|| &\geq ||p_{i_0} - p_i|| - ||p^* - p_{i_0}|| \\ &\geq ||p_{i_0} - p_i|| - \frac{\epsilon}{4} L_{\min} \\ &\geq (1 - \frac{\epsilon}{4}) ||p_{i_0} - p_i|| \qquad (6) \end{aligned}$$

for any $i \neq i_0$. □

Now we can obtain a $(1+\epsilon)$-approximation for case 1.

Theorem 3. *For case 1, if one tries every point in $\{p_i \mid 1 \leq i \leq n\}$ as a candidate for the truth vector, at least one yields a $(1+\epsilon)$-approximation for the objective function in (1), and the total time complexity is $O(n^2 d)$.*

Proof. We prove this theorem by showing how large the objective value will increase if p^* is moved to p_{i_0}. Firstly, we suppose that the weights are fixed temporarily. Then, by Lemma 4 and the fact that $0 < \epsilon < 1$, we have

$$\frac{||p_{i_0} - p_i||^2}{||p^* - p_i||^2} \leq \frac{1}{(1 - \epsilon/4)^2} \leq 1 + \epsilon \qquad (7)$$

for any i. This means that the objective value is increased by a factor no more than $1 + \epsilon$. Once p^* is moved to p_{i_0}, we can further update the weights according to Lemma 1, and the objective value will not increase. Note that the contribution of p_{i_0} to the objective value will become 0 since $\lim_{x \to 0} x \log \frac{S}{x} = 0$.

Since we need to try every point to find out p_{i_0} (as the candidate for the truth vector) and each point takes $O(nd)$ time to evaluate the objective function, the total time complexity is thus $O(n^2 d)$. □

3. A Simple Linear Time Algorithm for Case 2

In this section, we present a simple linear time approximation algorithm for case 2. Although the approximation ratio is relatively large ($O(\log \frac{n\Delta}{\epsilon})$), the idea used in the algorithm sheds some lights on how to find a more refined solution, *e.g.,* $(1+\epsilon)$-approximation in Section 4. Additionally, we believe that this simple linear time algorithm is also of some independent interest.

We first estimate the range for the weights. Lemmas 5 and 6 provides the upper and lower bounds for each w_i, and Lemma 7 shows a lower bound on their summation.

Lemma 5. *For case 2, each weight* $w_i \leq 2\log\frac{n\Delta}{\epsilon}$.

Lemma 5 can be easily obtained from the assumption $\min_{1\leq i\leq n}||p^* - p_i|| > \frac{\epsilon\sqrt{S}}{4\sqrt{n}\Delta}$ and Lemma 1. Note that we assume $n \geq 16$ here; otherwise, we just need to increase the front constant "2" (of $2\log\frac{n\Delta}{\epsilon}$) a little bit.

Lemma 6. *For any constant* $2 > c > 1$*, at least one of the following two events happens:*

1. $\min_{1\leq i\leq n} w_i \geq \log c$*;*
2. *all weights except one are at least* $\log\frac{c}{c-1}$.

Proof. Suppose that the first event does not happen, *i.e.,* $\min_{1\leq i\leq n} w_i < \log c$. Then, by Lemma 1 we know that there exists a p_l such that $\frac{||p^*-p_l||^2}{\sum_{i=1}^n ||p^*-p_i||^2} > \frac{1}{c}$. Since $\frac{1}{c} > \frac{1}{2}$, there is at most one such p_l, and each of the other points should have a weight at least $\log\frac{1}{1-1/c} = \log\frac{c}{c-1}$, *i.e.,* the second event happens. Thus the lemma is true. □

Lemma 7. *The sum of the weights* $\sum_{i=1}^n w_i \geq n\log n$.

Proof. From Lemma 1, we know that $\sum_{i=1}^n w_i = \sum_{i=1}^n \log\frac{S}{||p^*-p_i||^2}$. It is easy to see that the function $f(x) = \log\frac{S}{x}$ is convex, since $f''(x) = \frac{1}{x^2} > 0$. By *Jensen's inequality*, we have

$$\sum_{i=1}^{n}\log\frac{S}{||p^*-p_i||^2} \geq n\log\frac{S}{\sum_{i=1}^n ||p^*-p_i||^2/n} = n\log n. \tag{8}$$

This completes the proof. □

Before introducing the algorithm, we first find a proper value for c in Lemma 6. For this purpose, we consider the second event in Lemma 6. If this event happens, we know that there exists one point, say p_l, with weight less than $\log c$, and all other weights are at least $\log\frac{c}{c-1}$. This implies that

$$||p^* - p_l||^2 > \frac{1}{c}S; \quad \textit{and} \quad ||p^* - p_i||^2 < (1-\frac{1}{c})S \ \ \forall i \neq l. \tag{9}$$

This means that p_l is farther away from p^* than all other points, and the smaller c, the larger difference is. To differentiate p_l from other p_is, we consider **the ratio of the largest over the second largest distances from other points to** p_l (see Figure 2), which is smaller than

$$(\sqrt{\frac{1}{c}}+\sqrt{1-\frac{1}{c}})/(\sqrt{\frac{1}{c}}-\sqrt{1-\frac{1}{c}}) \tag{10}$$

due to triangle inequality and the fact that when $x > y > 0$, the function $f(x, y) = \frac{x+y}{x-y}$ is decreasing on x and increasing on y. Similarly, the ratio for any other p_i for $i \neq l$ is bigger than

$$(\sqrt{\frac{1}{c}}-\sqrt{1-\frac{1}{c}})/(2\sqrt{1-\frac{1}{c}}). \tag{11}$$

To make (11) larger than (10), we need to have

$$c < 10 - 4\sqrt{5} \approx 1.056, \tag{12}$$

and (11) = (10) ≈ 1.618 if $c = 10 - 4\sqrt{5}$. Consequently, we have the following lemma.

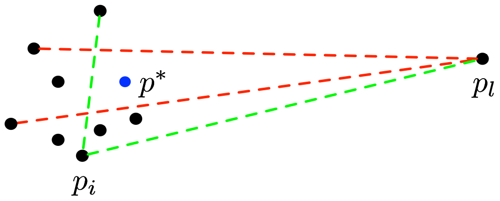


Figure 2: p_l and p_i are connected by dashed lines to their respective farthest and second farthest points; the blue point is p^*.

Lemma 8. *It is possible to find the point p_l with the smallest weight in $O(nd)$ time, if the second event in Lemma 6 happens with $1 < c < 10 - 4\sqrt{5}$.*

Proof. To prove this lemma, we can arbitrarily pick one point from the input and compute the ratio of the largest over the second largest distances from other points to it. From the above analysis, we know that if the ratio is smaller than 1.618, this point is p_l; otherwise the farthest point to it is p_l. Obviously, the total time of the above procedure is linear, *i.e.,* $O(nd)$. □

Algorithm 1 Linear time algorithm for case 2

```
Input: $P = \{p_i, | 1 \leq i \leq n\} \subset \mathbb{R}^d$
    1. Compute the (unit weighted) mean of $P$ , and denote it as $p_1^*$.
    2. Arbitrarily pick one point from $P$ and compute the ratio of the largest
       and second largest distances from other points to it,
        (a) remove the selected point from $P$ if the ratio is smaller than 1.618;
        (b) or remove the farthest point to it otherwise.
    3. Compute the mean of the remaining points, and denote it as $p_2^*$.
    4. Take the one from $\{p_1^*, p_2^*\}$ with a smaller objective value as the truth
       vector.
```

Now we are ready to present our algorithm (see Algorithm 1).

Theorem 4. *Algorithm 1 yields a* $(2\log\frac{n\Delta}{\epsilon}/\log c)$*-approximation for case 2, where* $c \approx 1.056$ *and the time complexity is* $O(nd)$*. In short, the approximation ratio is* $O(\log\frac{n}{\epsilon})$ *if* Δ *is a polynomial of* n*, or* $O(\log\frac{\Delta}{\epsilon})$ *otherwise.*

Proof. To prove this theorem, we consider the two events in Lemma 6 separately.

If the first event happens, we have $w_i \geq \log c$ for any i. Since p_1^* is the mean of P, we have

$$\sum_{i=1}^{n} ||p_1^* - p_i||^2 \leq \sum_{i=1}^{n} ||p^* - p_i||^2. \tag{13}$$

Consequently, if we fix the weights and move p^* to p_1^* (note that we can use Lemma 1 to update the weights and further reduce the objective value), the objective value of (1) will be

$$\begin{aligned}\sum_{i=1}^{n} w_i||p_1^* - p_i||^2 &\leq 2\log\frac{n\Delta}{\epsilon}\sum_{i=1}^{n}||p_1^* - p_i||^2\\ &\leq 2\log\frac{n\Delta}{\epsilon}\sum_{i=1}^{n}||p^* - p_i||^2\\ &\leq 2\log\frac{n\Delta}{\epsilon}\sum_{i=1}^{n}\frac{w_i}{\log c}||p^* - p_i||^2\\ &= (2\log\frac{n\Delta}{\epsilon}/\log c)\sum_{i=1}^{n} w_i||p^* - p_i||^2 \qquad (14)\end{aligned}$$

based on Lemma 5 & 6, which implies that p_1^* is a $(2\log\frac{n\Delta}{\epsilon}/\log c)$-approximation.

Now we consider the second event. Let p_l denote the point removed in step 2(a) or 2(b). From the proof of Lemma 8, we know that p_l has the smallest weight. Let $\tilde{p}^*$ be the weighted mean of $P\setminus\{p_l\}$. Suppose that the total weight of $P\setminus\{p_l\}$ is $\alpha\sum_{i=1}^{n} w_i$, then from Lemma 7 and the fact that $w_l \leq \log c < 1$, we have $\alpha > \frac{n\log n - 1}{n\log n}$. As a consequence, by Lemma 3 we have

$$||\tilde{p}^* - p^*||^2 < \frac{1}{n\log n - 1}\frac{\sum_{i=1}^{n} w_i||p^* - p_i||^2}{\sum_{i=1}^{n} w_i}. \qquad (15)$$

Then applying Lemma 2 in Section 1.2, we get

$$\sum_{i=1}^{n} w_i||\tilde{p}^* - p_i||^2 \leq \frac{n\log n}{n\log n - 1}\sum_{i=1}^{n} w_i||p^* - p_i||^2. \qquad (16)$$

(16) indicates that $\tilde{p}^*$ can replace p^* without causing much increase on the objective value. If we continue to move $\tilde{p}^*$ to p_2^*, the objective value becomes

$$\sum_{i=1}^{n} w_i||p_2^* - p_i||^2 = \sum_{i\neq l} w_i||p_2^* - p_i||^2 + w_l||p_2^* - p_l||^2. \qquad (17)$$

For the first term in the right hand side of (17), by a similar calculation to (14), we know that $\sum_{i\neq l} w_i||p_2^* - p_i||^2 < (2\log\frac{n\Delta}{\epsilon}/\log\frac{c}{c-1})\sum_{i\neq l} w_i||\tilde{p}^* - p_i||^2$. For the second term in the right hand side of (17), by an estimation similar to (10), we have $\frac{||p_2^* - p_l||}{||\tilde{p}^* - p_l||} \leq (\sqrt{\frac{1}{c}} + \sqrt{1 - \frac{1}{c}})/(\sqrt{\frac{1}{c}} - \sqrt{1 - \frac{1}{c}}) \approx 1.618$. (Note that both p_2^* and $\tilde{p}^*$ are a convex combination of $P\setminus p_l$). Putting (16) and (17) together, we know that p_2^* is a solution with approximation ratio

$$\frac{n\log n}{n\log n-1}\times\max\{2\log\frac{n\Delta}{\epsilon}/\log\frac{c}{c-1},1.618\}\le 2\log\frac{n\Delta}{\epsilon}/\log c. \tag{18}$$

Finally, it is easy to know that the time complexity is $O(nd)$. □

4. A $(1+\epsilon)$-Approximation for Case 2

In this section, we present a $(1+\epsilon)$-approximation for case 2. In Theorem 4, we consider only two groups of the points, *i.e.,* the point with the smallest weight and all others. In this section we show that by further partitioning the input points into more groups, it is possible to obtain a much better solution.

Definition 3 (Log-Partition)**.** *In case 2, let p_l and $p_{l'}$ be the points with the smallest and the second smallest weights, respectively. Then the log-partition is to divide the points in $\{p_i \mid 1\le i\le n\}$ into $k=\lceil\log_{1+\beta}\frac{2\log(n\Delta/\epsilon)}{w_{l'}}\rceil+1$ (where β is a small positive number that will be determined later) groups as follows:*

- $\mathcal{G}_1=\{p_l\}$.
- $\mathcal{G}_j=\{p_i \mid (1+\beta)^{j-2}w_{l'}\le w_i<(1+\beta)^{j-1}w_{l'}\}$ *for* $j\ge 2$.

Note that we cannot explicitly obtain the log-partition since we do not know the weights in advance. We can only assume that such a partition exists, which will be useful in the following analysis.

From Lemmas 5 and 6 and the fact that $\log(1+\beta)\approx\beta$ when β is a small positive number, we can easily have the following lemma.

Lemma 9. *In the log-partition, $k=O(\frac{1}{\beta}\log\log\frac{n\Delta}{\epsilon})$.*

In each $\mathcal{G}_j$, their weight difference is no more than a factor of $(1+\beta)$; as a consequence, their weighted mean and weighted standard deviation are very close to their mean and standard deviation respectively. In the remaining parts, we denote the mean and weighted mean of each $\mathcal{G}_j$ by $\hat{m}_j$ and m_j, the standard deviation $\sqrt{\frac{1}{|\mathcal{G}_j|}\sum_{p_i\in\mathcal{G}_j}||p_i-\hat{m}_j||^2}$ and weighted standard deviation $\sqrt{\frac{1}{\sum_{p_i\in\mathcal{G}_j}w_i}\sum_{p_i\in\mathcal{G}_j}w_i||p_i-m_j||^2}$ by $\hat{\delta}_j$ and δ_j, respectively.

Lemma 10. *For each $\mathcal{G}_j$ in the log-partition, $||\hat{m}_j - m_j|| \leq \beta\sqrt{1+\beta}\delta_j$, and $\delta_j \in [\frac{1}{\sqrt{1+\beta}}\hat{\delta}_j, \sqrt{1+\beta}\hat{\delta}_j]$.*

Proof. We first prove $||m_j - \hat{m}_j|| \leq \beta\sqrt{1+\beta}\delta_j$. Without loss of generality, we can assume that $\hat{m}_j$ is the origin, *i.e.*, $\sum_{p_i \in \mathcal{G}_j} p_i = 0$, which does not change the distance between m_j and $\hat{m}_j$. For simplicity, we denote $\alpha = (1+\beta)^{j-2} w_{l'}$. Then all the points in $\mathcal{G}_j$ have weights between α and $(1+\beta)\alpha$.

Firstly, $m_j = \frac{1}{\sum_{p_i \in \mathcal{G}_j} w_i} \sum_{p_i \in \mathcal{G}_j} w_i p_i = \frac{1}{\sum_{p_i \in \mathcal{G}_j} w_i} \sum_{p_i \in \mathcal{G}_j} (w_i - \alpha) p_i$ since $\hat{m}_j = 0$. Thus we have

$$\begin{aligned} ||m_j||^2 &= \frac{1}{(\sum_{p_i \in \mathcal{G}_j} w_i)^2} || \sum_{p_i \in \mathcal{G}_j} (w_i - \alpha) p_i ||^2 \\ &\leq \frac{1}{(\sum_{p_i \in \mathcal{G}_j} w_i)^2} |\mathcal{G}_j| \sum_{p_i \in \mathcal{G}_j} (w_i - \alpha)^2 ||p_i||^2 \\ &\leq \frac{1}{(|\mathcal{G}_j|\alpha)^2} |\mathcal{G}_j| \sum_{p_i \in \mathcal{G}_j} (\beta\alpha)^2 ||p_i||^2 \\ &= \frac{\beta^2}{|\mathcal{G}_j|} \sum_{p_i \in \mathcal{G}_j} ||p_i||^2. \end{aligned} \tag{19}$$

In addition, we have

$$\begin{aligned} \delta_j^2 &= \frac{1}{\sum_{p_i \in \mathcal{G}_j} w_i} \sum_{p_i \in \mathcal{G}_j} w_i ||p_i - m_j||^2 \\ &\geq \frac{1}{\sum_{p_i \in \mathcal{G}_j} w_i} \sum_{p_i \in \mathcal{G}_j} \alpha ||p_i - m_j||^2 \\ &\geq \frac{1}{\sum_{p_i \in \mathcal{G}_j} w_i} \sum_{p_i \in \mathcal{G}_j} \alpha ||p_i - \hat{m}_j||^2 \\ &= \frac{1}{\sum_{p_i \in \mathcal{G}_j} w_i} \sum_{p_i \in \mathcal{G}_j} \alpha ||p_i||^2 \\ &\geq \frac{1}{(1+\beta)\alpha|\mathcal{G}_j|} \sum_{p_i \in \mathcal{G}_j} \alpha ||p_i||^2 \\ &= \frac{1}{(1+\beta)|\mathcal{G}_j|} \sum_{p_i \in \mathcal{G}_j} ||p_i||^2. \end{aligned} \tag{20}$$

Combining (19) and (20), we have $||m_j - \hat{m}_j|| = ||m_j|| \leq \beta\sqrt{1+\beta}\delta_j$ (note that $\hat{m}_j$ is already translated to the origin).

Next, we show the relationship between δ_j and $\hat{\delta}_j$. From (20), we directly have that $\delta_j^2 \geq \frac{1}{1+\beta}\hat{\delta}_j^2$. Further, we have

$$\begin{aligned}
\delta_j^2 &= \frac{1}{\sum_{p_i \in \mathcal{G}_j} w_i} \sum_{p_i \in \mathcal{G}_j} w_i ||p_i - m_j||^2 \\
&\leq \frac{1}{\sum_{p_i \in \mathcal{G}_j} w_i} \sum_{p_i \in \mathcal{G}_j} w_i ||p_i - \hat{m}_j||^2 \\
&= \frac{1}{\sum_{p_i \in \mathcal{G}_j} w_i} \sum_{p_i \in \mathcal{G}_j} w_i ||p_i||^2 \\
&\leq \frac{1}{\alpha|\mathcal{G}_j|} \sum_{p_i \in \mathcal{G}_j} (1+\beta)\alpha ||p_i||^2 \\
&= \frac{1+\beta}{|\mathcal{G}_j|} \sum_{p_i \in \mathcal{G}_j} ||p_i||^2 = (1+\beta)\hat{\delta}_j^2. \qquad (21)
\end{aligned}$$

Thus, we have $\delta_j \in [\frac{1}{\sqrt{1+\beta}}\hat{\delta}_j, \sqrt{1+\beta}\hat{\delta}_j]$. □

Using Lemma 10, we can obtain a $(1+\epsilon)$-approximation algorithm for case 2. Below is the sketch of our idea.

Synopsis. The essential task of truth discovery is to find the weighted mean without knowing the weights in advance. Using log-partition, we can first divide the input points implicitly into k groups, and Lemma 10 enables us to ignore the weights inside each group. Then by applying random sampling techniques, we can estimate the weighed mean of each group, and find the weighted mean of the whole input using *simplex lemma*. We elaborate our ideas in the following subsections.

4.1. Modified Simplex Lemma

In [26], Ding and Xu introduced a simplex lemma for solving a large class of constrained clustering problems in high dimensional space. In this subsection, we show that despite developed for a different purpose, the simplex lemma is still applicable to our truth discovery problem.

Lemma 11 (Simplex Lemma [26])**.** *Given an unknown weighted point-set $Q \subset \mathbb{R}^d$, which is implicitly divided into k mutually exclusive groups $\{Q_j \mid 1 \leq j \leq k\}$,*

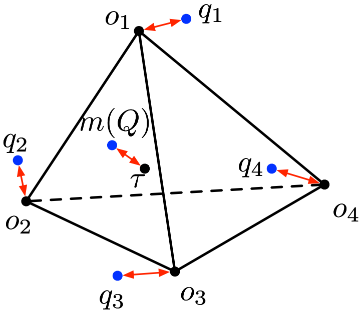


Figure 3: An illustration for Lemma 11 with $k = 4$; each o_j has a bounded distance to q_j, the corresponding exact weighted mean of Q_j, and the distance between τ and $m(Q)$ is also bounded.

and k points $\{o_j \mid 1 \leq j \leq k\}$ satisfying the condition that for each j, the distance between o_j and the weighted mean of the unknown Q_j is no more than a fixed value $L \geq 0$, it is possible to construct a grid of size $(8k/\epsilon)^k$ inside the simplex determined by $\{o_j \mid 1 \leq j \leq k\}$ such that at least one grid point τ satisfies the following inequality

$$||\tau - m(Q)|| \leq \sqrt{\epsilon}\delta(Q) + (1+\epsilon)L, \tag{22}$$

where $m(Q)$ and $\delta(Q)$ are the weighted mean and weighted standard deviation of Q, respectively.

Simplex lemma shows that it is possible to find an approximate weighted mean of an unknown point-set. The only known information is the approximate weighted mean of each unknown subset. L is a slack parameter to control the error bound in (22). See Figure 3. Also, a nice feature of the simplex lemma is that it needs to consider only a low dimensional subspace determined by the simplex ($k \ll d$), and thus can be applied to problems in high dimensional space.

It is easy to see that the simplex lemma is immediately applicable to the truth discovery problem for finding the weighted mean, if we are able to obtain the weighted mean (or only the mean due to Lemma 10) of each $\mathcal{G}_j$. The difficulty is that since some $\mathcal{G}_j$ could be quite small in its cardinality, it is extremely challenging to estimate the mean by using random sampling techniques. The following modified simplex lemma shows that it is actually possible to ignore such small-size groups.

Lemma 12 (Modified Simplex Lemma)**.** *Let Q, Q_j, ϵ, δ, and k be defined as in Lemma 11, and $\Gamma = \{j \mid \frac{w(Q_j)}{w(Q)} \geq \frac{\epsilon}{k}\}$, where $w(\cdot)$ is the total weight of a point-set. Then it is possible to construct a grid of size $(8k/\epsilon)^k$ inside the simplex determined by $\{o_j \mid j \in \Gamma\}$ such that at least one grid point τ satisfies the following*

$$||\tau - m(Q)|| \leq 2\sqrt{\frac{\epsilon}{1-\epsilon}}\delta(Q) + (1+\epsilon)L. \tag{23}$$

Proof. Let $Q_\Gamma = \cup_{j\in\Gamma} Q_j$. Then by Lemma 11, we immediately have the following inequality.

$$||\tau - m(Q_\Gamma)|| \leq \sqrt{\epsilon}\delta(Q_\Gamma) + (1+\epsilon)L, \tag{24}$$

where Q is simply replaced by Q_Γ. Now, we consider the differences between $m(Q)$, $\delta(Q)$ and $m(Q_\Gamma)$, $\delta(Q_\Gamma)$, respectively. Similar to (15) in Theorem 4 for proving the distance between $\tilde{p}^*$ and p^*, based on Lemma 3 we have

$$||m(Q_\Gamma) - m(Q)||^2 \leq \frac{w(Q \setminus Q_\Gamma)}{w(Q_\Gamma)}\delta^2(Q) \leq \frac{\epsilon}{1-\epsilon}\delta^2(Q), \tag{25}$$

where the last inequality comes from the facts that $w(Q \setminus Q_\Gamma) \leq k \times \frac{\epsilon}{k} w(Q)$ and $w(Q_\Gamma) \geq (1-\epsilon)w(Q)$. Furthermore, since $w(Q)\delta^2(Q) \geq w(Q_\Gamma)\delta^2(Q_\Gamma)$, we have

$$\delta^2(Q_\Gamma) \leq \frac{w(Q)}{w(Q_\Gamma)}\delta^2(Q) \leq \frac{1}{1-\epsilon}\delta^2(Q). \tag{26}$$

Plugging (25) and (26) into (24), we have

$$\begin{aligned} ||\tau - m(Q)|| &\leq ||\tau - m(Q_\Gamma)|| + ||m(Q_\Gamma) - m(Q)|| \\ &\leq \sqrt{\epsilon}\delta(Q_\Gamma) + (1+\epsilon)L + \sqrt{\frac{\epsilon}{1-\epsilon}}\delta(Q) \\ &\leq \sqrt{\epsilon}\frac{1}{\sqrt{1-\epsilon}}\delta(Q) + (1+\epsilon)L + \sqrt{\frac{\epsilon}{1-\epsilon}}\delta(Q) \\ &= 2\sqrt{\frac{\epsilon}{1-\epsilon}}\delta(Q) + (1+\epsilon)L. \end{aligned} \tag{27}$$

This completes the proof. □

4.2. The Algorithm Using Modified Simplex Lemma

The following two lemmas are commonly used random sampling techniques in Euclidean space. Lemma 13 shows that in order to estimate the mean of a point-set, one just needs to take the mean of a randomly selected sample. Lemma 14 further shows how to sample points in order to ensure that there are enough number of points in the sample from a hidden subset.

Lemma 13 ([28]). *Let T be a set of n points in $\mathbb{R}^d$ space, T' be a randomly selected subset of size t from T, and $\hat{m}(T)$, $\hat{m}(T')$ be the mean points of T and T' respectively. With probability $1-\eta$, $||\hat{m}(T)-\hat{m}(T')||^2 < \frac{1}{\eta t}\hat{\delta}^2(T)$, where $\hat{\delta}^2(T) = \frac{1}{n}\sum_{s\in T}||s-m(T)||^2$ and $0<\eta<1$.*

Lemma 14 ([26]). *Let Ω be a set of elements, and T be a subset of Ω with $\frac{|T|}{|\Omega|} = \alpha$ for some $\alpha \in (0,1)$. If randomly select $\frac{t \log \frac{t}{\eta}}{\log(1+\alpha)} = O(\frac{t}{\alpha}\log\frac{t}{\eta})$ elements from Ω, then with probability at least $1-\eta$, the sample contains t or more elements from T for $0<\eta<1$ and $t \in \mathbb{Z}^+$.*

By Lemma 12, we know that only those groups with large enough weight need to be considered. The following lemma further shows that each of such groups contains a significant fraction of the input points. This means that we can directly apply Lemmas 13 and 14 to estimate their means.

Lemma 15. *In the log-partition for case 2, if a group $\mathcal{G}_j$ has a total weight no less than $\frac{\epsilon}{k}\sum_{i=1}^n w_i$, it contains at least $\frac{\epsilon \log n}{2k\log(n\Delta/\epsilon)}n$ points,* i.e., $|\mathcal{G}_j|/|P| \geq \frac{\epsilon \log n}{2k\log(n\Delta/\epsilon)}$.

Proof. Let $|\mathcal{G}_j|$ and $w(\mathcal{G}_j)$ denote the number of points and the total weight in $\mathcal{G}_j$, respectively. From previous discussion (*i.e.,* Lemmas 5 and 7), we know that

$$w(\mathcal{G}_j) \leq 2\log\frac{n\Delta}{\epsilon}|\mathcal{G}_j|; \qquad \sum_{i=1}^n w_i \geq n\log n. \tag{28}$$

With the assumption $w(\mathcal{G}_j) \geq \frac{\epsilon}{k}\sum_{i=1}^n w_i$, we have $|\mathcal{G}_j| \geq (\frac{\epsilon}{k}\sum_{i=1}^n w_i)/(2\log\frac{n\Delta}{\epsilon})$ $\geq \frac{\epsilon\log n}{2k\log(n\Delta/\epsilon)}n$. □

Algorithm 2 $(1+\epsilon)$-algorithm for case 2

```
Input: $P = \{p_i, \mid 1 \leq i \leq n\} \subset \mathbb{R}^d$, and $\alpha = \frac{\epsilon^3 \log n}{2(\log\log(n\Delta/\epsilon))^2 \log(n\Delta/\epsilon)}$.
    1. Randomly take a sample $N$ from the input with size $\frac{4k}{\alpha\beta^2} \log \frac{16k^2}{\beta^2}$.
    2. Enumerate all the subsets having $4k/\beta^2$ points from $N$, compute their means, and put them into a set $U$.
    3. For any $k'$-tuple from $U$, where $k'$ is enumerated from $\{1, 2, \cdots, k\}$, apply Lemma 12 to build a grid inside the simplex determined by the $k'$-tuple.
    4. Try all the grid points, and output the one with the smallest objective value of (1) in Definition 1.
```

Now we are ready to present our refined algorithm for truth discovery. Firstly, we use Lemmas 14 and 15 to sample an enough number of points from each group with large enough weight. Then, we apply Lemma 13 to obtain their approximate means. Finally, we use the modified simplex lemma (*i.e.,* Lemma 12) to obtain the desired $(1+\epsilon)$-approximation. See Algorithm 2. Below, we analyze the correctness of the algorithm. For convenience, we denote the weighted standard deviation induced by p^*, *i.e.,* $\sqrt{\frac{1}{\sum_{i=1}^n w_i} \sum_{i=1}^n w_i ||p^* - p_i||^2}$, by $\delta(P)$.

A key step for analyzing the correctness of the algorithm is to determine the value of β for log-partition. When applying the modified simplex lemma, we have to keep the value of L to be roughly $O(\sqrt{\epsilon}\delta(P))$, such that the obtained grid point τ can result in a $(1 + O(1)\epsilon)$-approximation solution by Lemma 2. Note that the value of L depends on two parts, the distance between m_j and $\hat{m}_j$ (Lemma 10), and the error for estimating the position of $\hat{m}_j$ (Lemma 13), respectively. For simplicity, we only consider the first part temporarily, and actually the following analysis will show that the first part dominates the value of L. First, when $j \in \Gamma$ (see Lemma 12), we have the upper bound of $||m_j - \hat{m}_j||$,

$$\beta\sqrt{1+\beta}\delta_j < 2\beta\delta_j \leq 2\beta\sqrt{\frac{k}{\epsilon}}\delta(P) \tag{29}$$

by Lemma 10. Meanwhile, we know that $k = O(\frac{1}{\beta}\log\log\frac{n\Delta}{\epsilon})$ by Lemma 9. Thus, we need to set

$$\beta = \frac{\epsilon^2}{\log\log\frac{n\Delta}{\epsilon}} \tag{30}$$

to guarantee that $L = O(\sqrt{\epsilon}\delta(P))$. And as a consequence,

$$k = \frac{1}{\epsilon^2}(\log\log\frac{n\Delta}{\epsilon})^2. \tag{31}$$

Also, (31) together with Lemma 15 implies that $|\mathcal{G}_j|/|P| \geq \frac{\epsilon^3\log n}{2(\log\log(n\Delta/\epsilon))^2\log(n\Delta/\epsilon)}$ for each $j \in \Gamma$. By simple calculations and Lemmas 14, we know that with probability $(1-\frac{1}{4k})^k \geq 1 - 1/4 = 3/4$ the sample N contains at least $4k/\beta^2$ points from each of such group $\mathcal{G}_j$. From Lemma 13, we know that with probability $(1-\frac{1}{4k})^k \geq 3/4$, for each of such $\mathcal{G}_j$ the mean of the corresponding $4k/\beta^2$ points has a distance no more than $\beta\hat{\delta}_j \leq \beta\sqrt{1+\beta}\delta_j = O(\sqrt{\epsilon}\delta(P))$ to its mean (the inequality comes from Lemma 10). In total, L is bounded by $O(\sqrt{\epsilon}\delta(P))$, and we have a $(1+O(1)\epsilon)$-approximation (by Lemma 2).

As for the running time, we note that $k = \frac{1}{\epsilon^2}(\log\log\frac{n\Delta}{\epsilon})^2$. In Step 2 of Algorithm 2, we enumerate all the subsets having $4k/\beta^2$ points from N and put their means into the set U, and thus $|U| = O\big(|N|^{4k/\beta^2}\big)$; in Step 3, we enumerate all the k'-tuples from U for $k' = \{1, 2, \cdots, k\}$, and apply Lemma 12 to build a grid inside the simplex determined by each of the k'-tuples. So there are $O\big((|N|^{4k/\beta^2})^k\big)$ simplexes in total, and the grid size of each simplex is $(8k/\epsilon)^k$ (from Lemm 12). Consequently, the total number of grid points is

$$O\big((|N|^{4k/\beta^2})^k(8k/\epsilon)^k\big) = 2^{O(\frac{1}{\epsilon^8}(\log\log n\Delta)^7)}. \tag{32}$$

Since $\frac{1}{\epsilon^8}(\log\log n\Delta)^7 < \sigma\log n\Delta$ for any small positive σ if ϵ is fixed and $n\Delta$ is large enough, the time complexity is $O(2^{\sigma\log n\Delta}nd) = O((n\Delta)^\sigma nd)$.

Through the above analysis, we have the following theorem.

Theorem 5. *With probability* $9/16$*, Algorithm 2 outputs a* $(1+\epsilon)$*-approximation for case 2, and the time complexity is* $O((n\Delta)^\sigma nd)$*, where* σ *could be any small positive number. In short, the time complexity is* $O(n^{1+\sigma}d)$ *if* Δ *is a polynomial of* n*, or* $O(\Delta^\sigma nd)$ *otherwise.*

5. Improving the Time Complexity

A common strategy adopted by the $(1+\epsilon)$-approximation algorithms in Section 2 and 4 for the two cases is to first identify a set of candidates for the truth vector, then compute the objective value for each candidate, and finally output the candidate with the smallest objective value. Since computing the objective value for each candidate costs $O(nd)$ time, the total time is thus $O(z \times nd)$, where z is the number of candidates (*i.e.,* $z = n$ for case 1 and $z = (n\Delta)^{\sigma}$ for case 2). In this section, we show that when the spread ratio Δ is not too large, the **amortized time complexity** for computing all the objective values of the candidates can be reduced to sub-linear, and consequently the overall time complexity is nearly linear.

Recall that Theorem 1 tells us that the objective value is equal to the entropy based geometric variance $S \times H$ induced by p^*. In order to reduce the time complexity for computing the objective value, below we show how to efficiently compute S and H, respectively.

Lemma 16. *The value of S for all the z candidates can be computed in a total of $O((n+z)d)$ time,* i.e., *$O(\frac{nd}{z}+d)$ amortized time complexity for each candidate.*

Proof. Let $\hat{m}(P)$ be the (unit weighted) mean of the point-set P, *i.e.,* $\hat{m}(P) = \frac{1}{n}\sum_{i=1}^{n} p_i$. Then in $O(nd)$ time, we can compute the value of $\hat{S} = \sum_{i=1}^{n} ||\hat{m}(P) - p_i||^2$. For each candidate p^*, we know that its total variance $S = \sum_{i=1}^{n} ||p^* - p_i||^2 = n||p^* - \hat{m}(P)||^2 + \hat{S}$ (by Lemma 2 in Section 1.2). Clearly, the variance of p^* can be computed in $O(d)$ time by using the value of $\hat{S}$. This implies that the total time for computing the value of S for all z candidates is $O(nd + zd)$. □

From Lemma 16, we know that it is possible to compute the total variance S in an amortized sub-linear time. Below we discuss how to efficiently compute the entropy H. The following lemma comes from [29] for entropy estimation.

Lemma 17 ([29])**.** *Let $F = \{f_i \mid 1 \le i \le n\}$ be a discrete probabilistic distribution with the entropy $H = \sum_{i=1}^{n} -f_i \log f_i$, and two parameters $\epsilon, \delta \in (0,1)$.*

There exists an algorithm outputting a value $\tilde{H} \in [(1-\epsilon)H, (1+\epsilon)H]$ with probability $1-\delta$, which makes at most $O(\frac{1}{\epsilon^2 H}\log n \log(\frac{1}{\delta}))$ queries on F.

To estimate H, the algorithm presented in [29] does not read all the values in F. Instead, it takes only a subset of $O(\frac{1}{\epsilon^2 H}\log n \log(\frac{1}{\delta}))$ samples (*i.e.,* queries) from F. From the above lemma, we know that if H is small, the number of needed queries could be quite large, and consequently the time complexity could be high. To avoid this issue, we show in the following lemma that H can actually be lower bounded in our problem if Δ is not too large. Also note that in our problem each query costs only $O(d)$ time, since it can be computed by equation $f_i = \frac{||p^* - p_i||^2}{S}$, where S is the total variance already obtained in Lemma 16.

Lemma 18. *If $\Delta = \tilde{O}(\sqrt{n})$ $(= O(\sqrt{n} \times poly(\log n)))$, $H \geq \frac{1}{poly(\log n)}$.*

For simplicity, we let $l_i^2 = ||p^* - p_i||^2$. Then $H = \sum_{i=1}^{n} \frac{l_i^2}{S} \log \frac{S}{l_i^2}$. Since we only need to care about the ratio $\frac{l_i^2}{S}$, without loss of generality we can assume that $\min_{i<j} ||p_i - p_j||^2 = 1$ and $\max_{i<j} ||p_i - p_j||^2 = \Delta^2$. Before proving Lemma 18, we first have the following lemma.

Lemma 19. *Except for the smallest value in $\{l_i^2 \mid 1 \leq i \leq n\}$, all other values are between $\frac{1}{4}$ and Δ^2. Furthermore, $S > \frac{1}{4}\Delta^2$.*

Proof. Firstly, if there exist $i_1 \neq i_2$ such that both $l_{1_1}^2$ and $l_{i_2}^2$ are smaller than $1/4$, then from triangle inequality we know that

$$||p_{i_1} - p_{i_2}|| \leq ||p_{i_1} - p^*|| + ||p_{i_2} - p^*|| < 1, \tag{33}$$

which contradicts our assumption that $\min_{i<j} ||p_i - p_j||^2 = 1$. Secondly, from the construction of the set of candidates for p^* in both cases (see Sections 2 and 4), we know that p^* is always inside the convex hull of P. Thus, we have $\max\{l_i^2 \mid 1 \leq i \leq n\} \leq \Delta^2$ due to triangle inequality as well.

Assume that $||p_{i_1} - p_{i_2}|| = \max_{i<j} ||p_i - p_j|| = \Delta$. Again, from triangle inequality we know that either $||p_{i_1} - p^*||$ or $||p_{i_2} - p^*||$ is at least $\Delta/2$. Thus we can easily know that $S > \frac{1}{4}\Delta^2$. □

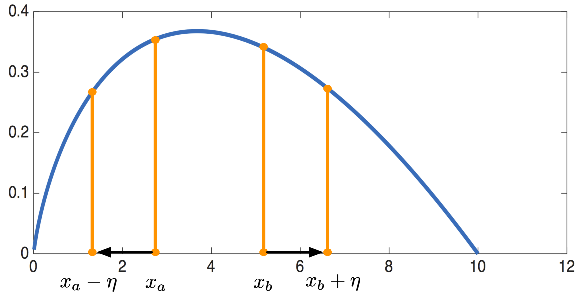


Figure 4: The curve of $g(x)$ with $S = 10$.

Proof. (**of Lemma 18**) Let $h = \min\{S, \Delta^2\}$, and $g(x) = \frac{x}{S} \log \frac{S}{x}$ for any $x \in [1/4, h]$, which is concave (as $g''(x) = -\frac{1}{Sx} < 0$). Considering two values $1/4 < x_a \le x_b < h$, we know that

$$g(x_a) + g(x_b) > g(x_a - \eta) + g(x_b + \eta), \tag{34}$$

where $\eta = \min\{x_a - 1/4, h - x_b\}$ (see Figure 4), and the sum of $x_a - \eta$ and $x_b + \eta$ is always $x_a + x_b$. This suggests that to find a lower bound of H for a fixed S, we can first identify two values $1/4 < l_{i_1}^2 \le l_{i_2}^2 < h$, and then decrease $l_{i_1}^2$ and increase $l_{i_2}^2$ in the same speed until either $l_{i_1}^2 = 1/4$ or $l_{i_2}^2 = h$. After repeating the above operation at most $n - 1$ times, we have at most one $l_i^2 \in (1/4, h)$, one smaller than $1/4$ (recall Lemma 19), and all the others are either $1/4$ or h. Suppose that t of them have a value of $1/4$ and $n - 2 - t$ of them have a value of h in $\{l_i^2 \mid 1 \le i \le n\}$, where $0 \le t \le n - 2$. Then we have:

$$\begin{aligned} H &\ge \frac{t}{4S} \log(4S) + \frac{(n-2-t)h}{S} \log \frac{S}{h}; && (35) \\ (t+1)\frac{1}{4} + (n-2-t)h \le S &\le (t+1)\frac{1}{4} + (n-1-t)h. && (36) \end{aligned}$$

If $S \le 2\Delta^2$, we know that at most two items equal to h (*i.e.*, $n - 2 - t \le 2$). Consequently, we know that the right hand side of (35) is at least $\frac{t}{4S} \log(4S) \ge \frac{n-4}{4S} \log(4S)$. Also notice that $S \in (\Delta^2/4, 2\Delta^2]$ in this case (where $S > \Delta^2/4$ comes from Lemma 19). Thus, we have

$$H \ge \frac{n-4}{4S} \log(4S) \ge \frac{n-4}{8\Delta^2} \log \Delta^2. \tag{37}$$

To bound the right hand side of the above inequality, we can actually assume that $\Delta^2 \ge 2$. Otherwise, from Lemma 19 and some simple calculations, we know that $H = \sum_{i=1}^{n} \frac{l_i^2}{S} \log \frac{S}{l_i^2} = \Theta(\log n)$. Thus, (37) becomes

$$H \geq \frac{n-4}{8\Delta^2} \log 2 \geq \frac{1}{poly(\log n)}, \tag{38}$$

and the lemma is true.

Now, we consider the case of $S > 2\Delta^2$, which immediately implies that $h = \Delta^2$. Note that $t \in [0, n-2]$. From (35) and (36), we have

$$\begin{aligned} H &\geq \frac{t}{(t+1) + 4(n-1-t)\Delta^2} \log(4S) + \frac{4(n-2-t)\Delta^2}{(t+1) + 4(n-1-t)\Delta^2} \log \frac{S}{\Delta^2} \\ &\geq \left(\frac{t}{(t+1) + 4(n-1-t)\Delta^2} + \frac{4(n-2-t)\Delta^2}{(t+1) + 4(n-1-t)\Delta^2}\right) \log 2 \\ &= \frac{t + 4(n-2-t)\Delta^2}{(t+1) + 4(n-1-t)\Delta^2} \log 2 \\ &= \frac{t + 4(n-2-t)\Delta^2}{t + 4(n-2-t)\Delta^2 + 1 + 4\Delta^2} \log 2 \\ &\geq \frac{n-2}{n-2+1+4\Delta^2} \log 2 \geq \frac{1}{poly(\log n)}. \end{aligned} \tag{39}$$

The second inequality follows from $S \geq 2\Delta^2$.

This completes the proof of Lemma 18. □

From previous discussion, we know that the time complexity can be improved to nearly linear if $\Delta = \tilde{O}(\sqrt{n})$. We can take the union of the candidates in both case 1 and case 2 (*i.e.*, $\min_{1 \leq i \leq n} ||p^* - p_i|| \leq \frac{\epsilon\sqrt{S}}{4\sqrt{n}\Delta}$ or $\min_{1 \leq i \leq n} ||p^* - p_i|| > \frac{\epsilon\sqrt{S}}{4\sqrt{n}\Delta}$), and denote it as $\mathcal{Z}$, where $|\mathcal{Z}| = n + (n\Delta)^\sigma = O(n)$ since $\Delta = \tilde{O}(\sqrt{n})$. Note that in case 2, the candidates are obtained by using random sampling which takes sub-linear time (see Section 4.2). Consequently, finding such a set $\mathcal{Z}$ needs only $O(|\mathcal{Z}|d) = O(nd)$ time. By Lemmas 16, 17 , and 18, we have the following theorem (when applying Lemma 17, we should replace the parameter δ by $O(\frac{\delta}{n})$ since we have to guarantee the success for all the $O(n)$ candidates; the number of queries increases only by a factor of $\log n$).

Theorem 6. *Given an instance P of the truth discovery problem with $\Delta = \tilde{O}(\sqrt{n})$ and two parameters $\epsilon, \delta \in (0,1)$, there exists an algorithm yielding a $(1+\epsilon)$-approximation with success probability $\frac{9}{16}(1-\delta)$. The time complexity is $\tilde{O}(nd)$, where the hiding constant in the big-O notation depends on ϵ and δ.*

In some real world applications, the dimensionality d (which is the maximum size of the data from each source) could be much larger than n. For this case, we can first apply the well known *JL-Lemma* [30] to reduce the dimensionality from d to $O(\frac{\log n}{\epsilon^2})$ and then apply our algorithm. Note that this can only slightly increase the objective value, since p^* is the weighted mean and consequently we have

$$\sum_{i=1}^{n} w_i ||p_i - p^*||^2 = \frac{1}{2\sum_{i=1}^{n} w_i} \sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j ||p_i - p_j||^2, \tag{40}$$

where JL-Lemma based dimension reduction approximately preserves the pairwise distances (see Section 6.4 in Appendix for the details). As for the running time, we know that a straightforward approach is just multiplying the data matrix $A \in \mathbb{R}^{d\times n}$ with the random projection matrix $R \in \mathbb{R}^{O(\frac{\log n}{\epsilon^2})\times d}$, which costs $O(nd \log n/\epsilon^2)$ time in total. We can also let R be a rescaled random sign matrix [31] and use the technique in [32] to further reduce the time complexity to $O(nd\frac{\log n}{\epsilon^2 \log d})$.

Corollary 1. *When $d = \Omega(n^c)$ for some constant $c > 0$, the time complexity in Theorem 6 can be further improved to $O(nd\frac{\log n}{\epsilon^2 \log d} + n \times poly(\log n)) = O(nd)$, where the hiding constant depends on c, ϵ and δ.*

Remark 1. *Following our work published in [1], Huang, together with the two authors of this paper, developed a new algorithm [33] which removes the dependence of Δ in the time complexity, but has a worse running time than the ones in Theorem 6 and Corollary 1. Their result is based on a novel range cover technique, which is interesting in its own right.*

References

[1] H. Ding, J. Gao, J. Xu, Finding global optimum for truth discovery: Entropy based geometric variance, in: 32nd International Symposium on Computational Geometry, SoCG 2016, June 14-18, 2016, Boston, MA, USA, 2016, pp. 34:1–34:16.

[2] Y. Li, J. Gao, C. Meng, Q. Li, L. Su, B. Zhao, W. Fan, J. Han, A survey on truth discovery, Acm Sigkdd Explorations Newsletter 17 (2) (2016) 1–16.

[3] H. Li, B. Zhao, A. Fuxman, The wisdom of minority: Discovering and targeting the right group of workers for crowdsourcing, in: Proceedings of the 23rd international conference on World wide web, ACM, 2014, pp. 165–176.

[4] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, J. Han, Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation, in: Proceedings of the 2014 ACM SIGMOD international conference on Management of data, ACM, 2014, pp. 1187–1198.

[5] Q. Li, Y. Li, J. Gao, L. Su, B. Zhao, M. Demirbas, W. Fan, J. Han, A confidence-aware approach for truth discovery on long-tail data, Proceedings of the VLDB Endowment 8 (4) (2014) 425–436.

[6] J. Pasternack, D. Roth, Knowing what to believe (when you already know something), in: Proceedings of the 23rd International Conference on Computational Linguistics, Association for Computational Linguistics, 2010, pp. 877–885.

[7] H. Xiao, J. Gao, Z. Wang, S. Wang, L. Su, H. Liu, A truth discovery approach with theoretical guarantee., in: KDD, 2016, pp. 1925–1934.

[8] X. Yin, J. Han, S. Y. Philip, Truth discovery with multiple conflicting information providers on the web, IEEE Transactions on Knowledge and Data Engineering 20 (6) (2008) 796–808.

[9] S. Mukherjee, G. Weikum, C. Danescu-Niculescu-Mizil, People on drugs: credibility of user statements in health communities, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2014, pp. 65–74.

[10] L. Su, Q. Li, S. Hu, S. Wang, J. Gao, H. Liu, T. F. Abdelzaher, J. Han, X. Liu, Y. Gao, et al., Generalized decision aggregation in distributed sens-

ing systems, in: Real-Time Systems Symposium (RTSS), IEEE, 2014, pp. 1–10.

[11] D. Wang, L. Kaplan, T. F. Abdelzaher, Maximum likelihood analysis of conflicting observations in social sensing, ACM Transactions on Sensor Networks (ToSN) 10 (2) (2014) 30.

[12] X. L. Dong, L. Berti-Equille, D. Srivastava, Integrating conflicting data: the role of source dependence, Proceedings of the VLDB Endowment 2 (1) (2009) 550–561.

[13] Amazon mechanical turk (accessed 08/21/2017).
URL `https://www.mturk.com/mturk/welcome`

[14] P. Smyth, U. M. Fayyad, M. C. Burl, P. Perona, P. Baldi, Inferring ground truth from subjective labelling of venus images, in: Advances in neural information processing systems, 1995, pp. 1085–1092.

[15] Y. Zhang, X. Chen, D. Zhou, M. I. Jordan, Spectral methods meet em: A provably optimal algorithm for crowdsourcing, The Journal of Machine Learning Research 17 (1) (2016) 3537–3580.

[16] P. Welinder, S. Branson, P. Perona, S. J. Belongie, The multidimensional wisdom of crowds, in: Advances in neural information processing systems, 2010, pp. 2424–2432.

[17] D. Zhou, S. Basu, Y. Mao, J. C. Platt, Learning from the wisdom of crowds by minimax entropy, in: Advances in Neural Information Processing Systems, 2012, pp. 2195–2203.

[18] L. Wang, Z.-H. Zhou, Cost-saving effect of crowdsourcing learning., in: IJCAI, 2016, pp. 2111–2117.

[19] J. Whitehill, T.-f. Wu, J. Bergsma, J. R. Movellan, P. L. Ruvolo, Whose vote should count more: Optimal integration of labels from labelers of

unknown expertise, in: Advances in neural information processing systems, 2009, pp. 2035–2043.

[20] C. E. Shannon, A mathematical theory of communication, ACM SIGMOBILE Mobile Computing and Communications Review 5 (1) (2001) 3–55.

[21] P. Jain, P. Netrapalli, S. Sanghavi, Low-rank matrix completion using alternating minimization, in: Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013, 2013, pp. 665–674.

[22] M. Hardt, Understanding alternating minimization for matrix completion, in: 55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014, 2014, pp. 651–660.

[23] P. Jain, A. Tewari, Alternating minimization for regression problems with vector-valued outputs, in: Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, 2015, pp. 1126–1134.

[24] M. R. Gupta, Y. Chen, Theory and use of the EM algorithm, Foundations and Trends in Signal Processing 4 (3) (2010) 223–296.

[25] P. Indyk, R. Motwani, S. Venkatasubramanian, Geometric matching under noise: Combinatorial bounds and algorithms., in: SODA, 1999, pp. 457–465.

[26] H. Ding, J. Xu, A unified framework for clustering constrained data without locality property, in: Proceedings of the Twenty-sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '15, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2015, pp. 1471–1490.

[27] A. Kumar, Y. Sabharwal, S. Sen, Linear-time approximation schemes for clustering problems in any dimensions, Journal of the ACM (JACM) 57 (2) (2010) 5.

[28] M. Inaba, N. Katoh, 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.

[29] S. Guha, A. McGregor, S. Venkatasubramanian, Sublinear estimation of entropy and information distances, ACM Transactions on Algorithms (TALG) 5 (4) (2009) 35.

[30] W. B. Johnson, J. Lindenstrauss, Extensions of lipschitz mappings into a hilbert space, Contemporary mathematics 26 (189-206) (1984) 1.

[31] D. Achlioptas, Database-friendly random projections: Johnson-lindenstrauss with binary coins, Journal of computer and System Sciences 66 (4) (2003) 671–687.

[32] E. Liberty, S. W. Zucker, The mailman algorithm: A note on matrix–vector multiplication, Information Processing Letters 109 (3) (2009) 179–182.

[33] Z. Huang, H. Ding, J. Xu, Faster algorithm for truth discovery via range cover, in: Algorithms and Data Structures - 15th International Symposium, WADS 2017, St. John's, NL, Canada, July 31 - August 2, 2017, Proceedings, 2017, pp. 461–472.

6. Appendix

6.1. Proof of Lemma 2

We use $< x, y >$ to denote the inner product of x and y in $\mathbb{R}^d$. The left hand side of the equation is

$$\begin{aligned}
\sum_{i=1}^{n} w_i||q - q_i||^2 &= \sum_{i=1}^{n} w_i||q - m(Q) + m(Q) - q_i||^2 \\
&= \sum_{i=1}^{n} w_i(||q - m(Q)||^2 + 2 < q - m(Q), m(Q) - q_i > + ||m(Q) - q_i||^2) \\
&= \sum_{i=1}^{n} w_i||q - m(Q)||^2 + 2 < q - m(Q), \sum_{i=1}^{n} w_i(m(Q) - q_i) > \\
&\quad + \sum_{i=1}^{n} w_i||m(Q) - q_i||^2 \\
&= W||q - m(Q)||^2 + \sum_{i=1}^{n} w_i||m(Q) - q_i||^2,
\end{aligned}$$

where the final equality follows from the fact that $\sum_{i=1}^{n} w_i(m(Q) - q_i) = 0$.

6.2. Proof of Lemma 3

Let $Q_2 = Q \setminus Q_1$, and $m(Q_2)$ be its weighted mean point. By Lemma 2, we have the following two equalities.

$$\sum_{q_i \in Q_1} w_i||q_i - m(Q)||^2 = \sum_{q_i \in Q_1} w_i||q_i - m(Q_1)||^2 + \alpha W \times ||m(Q_1) - m(Q)||^2, \quad (41)$$

and

$$\sum_{q_i \in Q_2} w_i||q_i - m(Q)||^2 = \sum_{q_i \in Q_2} w_i||q_i - m(Q_2)||^2 + (1 - \alpha)W \times ||m(Q_2) - m(Q)||^2. \quad (42)$$

Let $L = ||m(Q_1) - m(Q_2)||$. By the definition of weighted mean point, we have

$$\begin{aligned}
m(Q) &= \frac{1}{W} \sum_{q_i \in Q} w_i q_i \\
&= \frac{1}{W}(\sum_{q_i \in Q_1} w_i q_i + \sum_{q_i \in Q_2} w_i q_i) \\
&= \frac{1}{W}(\alpha W m(Q_1) + (1 - \alpha)W m(Q_2)). \quad (43)
\end{aligned}$$

Thus the three points $\{m(Q), m(Q_1), m(Q_2)\}$ are collinear, while $||m(Q_1) - m(Q)|| = (1-\alpha)L$ and $||m(Q_2) - m(Q)|| = \alpha L$. Meanwhile, by the definition of δ, we have

$$\delta^2 = \frac{1}{W}(\sum_{q_i \in Q_1} w_i||q_i - m(Q)||^2 + \sum_{q_i \in Q_2} w_i||q_i - m(Q)||^2). \tag{44}$$

Combining (41) and (42), we have

$$\begin{aligned} \delta^2 &\geq \frac{1}{W}(\alpha W \times ||m(Q_1) - m(Q)||^2 + (1-\alpha)W \times ||m(Q_2) - m(Q)||^2) \\ &= \alpha((1-\alpha)L)^2 + (1-\alpha)(\alpha L)^2 \\ &= \alpha(1-\alpha)L^2. \end{aligned} \tag{45}$$

Thus, we have $L \leq \frac{\delta}{\sqrt{\alpha(1-\alpha)}}$, which means that $||m(Q_1) - m(Q)|| = (1-\alpha)L \leq \sqrt{\frac{1-\alpha}{\alpha}}\delta$.

6.3. Proof of Theorem 2

Let $Opt = \sum_{i=1}^{n} w_i||p^* - p_i||^2$ be the optimal objective value. Then we know that at least one point, say p_l, has its squared distance to p^* no bigger than the average, *i.e.,* $||p^* - p_l||^2 \leq \frac{Opt}{\sum_{i=1}^{n} w_i}$. By applying Lemma 2, we have

$$\sum_{i=1}^{n} w_i||p_l - p_i||^2 \quad = \quad (\sum_{i=1}^{n} w_i)||p_l - p^*||^2 + \sum_{i=1}^{n} w_i||p^* - p_i||^2 \leq 2Opt. \tag{46}$$

Once p^* is replaced by p_l, we can further update the weights according to Lemma 1, and the objective value will not increase. Note that the contribution of p_l to the objective value will become 0 since $\lim_{x \to 0} x \log \frac{1}{x} = 0$.

Thus, p_l is a 2-approximation. Furthermore, finding such a p_l needs to try n times with each time costing $O(nd)$ time, which means that the total time complexity is $O(n^2 d)$.

6.4. Quality Preserving After Applying JL-Lemma

We first prove the formula (40). Let $W = \sum_{i=1}^{n} w_i$. From Lemma 2 we have

$$\sum_{j=1}^{n} w_j||p_i - p_j||^2 = W||p_i - p^*||^2 + \sum_{j=1}^{n} w_j||p_j - p^*||^2. \tag{47}$$

Consequently, we get

$$\begin{aligned}\sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j ||p_i - p_j||^2 &= \sum_{i=1}^{n} w_i \sum_{j=1}^{n} w_j ||p_i - p_j||^2 \\ &= \sum_{i=1}^{n} w_i (W||p_i - p^*||^2 + \sum_{j=1}^{n} w_j ||p_j - p^*||^2) \\ &= W\sum_{i=1}^{n} w_i ||p_i - p^*||^2 + W\sum_{j=1}^{n} w_j ||p_j - p^*||^2 \\ &= 2W\sum_{i=1}^{n} w_i ||p_i - p^*||^2, \qquad (48)\end{aligned}$$

which implies formula (40). Suppose that $\hat{p}^*$ is a $(1+\epsilon)$-approximation in the $O(\frac{\log n}{\epsilon^2})$-dimensional subspace after the random projection, and the corresponding weights are $\{\hat{w}_1, \cdots, \hat{w}_n\}$. Also let the projection of each p_i be $\hat{p}_i$. From (40) we have

$$\begin{aligned}&\frac{1}{2\sum_{i=1}^{n}\hat{w}_i}\sum_{i=1}^{n}\sum_{j=1}^{n}\hat{w}_i\hat{w}_j||\hat{p}_i - \hat{p}_j||^2 \\ \leq\ &(1+\epsilon)\frac{1}{2\sum_{i=1}^{n} w_i}\sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j ||\hat{p}_i - \hat{p}_j||^2. \qquad (49)\end{aligned}$$

If we apply JL-Lemma on both sides of (49), we have

$$\begin{aligned}&\frac{1}{2\sum_{i=1}^{n}\hat{w}_i}\sum_{i=1}^{n}\sum_{j=1}^{n}\hat{w}_i\hat{w}_j||p_i - p_j||^2 \\ \leq\ &\frac{(1+\epsilon)^2}{1-\epsilon}\frac{1}{2\sum_{i=1}^{n} w_i}\sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j ||p_i - p_j||^2. \qquad (50)\end{aligned}$$

Let $q = \frac{1}{\sum_{i=1}^{n}\hat{w}_i}\sum_{i=1}^{n}\hat{w}_i p_i$. Then we know that it is a $\frac{(1+\epsilon)^2}{1-\epsilon} \approx (1+3\epsilon)$-approximation in the original $\mathbb{R}^d$ space.