Certified Computation from Unreliable Datasets

Themis Gouleakis TGOULE@MIT.EDU
EECS and CSAIL, MIT

Christos Tzamos TZAMOS@MIT.EDU
Microsoft Research

Manolis Zampetakis MZAMPET@MIT.EDU
EECS and CSAIL, MIT

Editors: Sebastien Bubeck, Vianney Perchet and Philippe Rigollet

Abstract

A wide range of learning tasks require human input in labeling massive data. The collected data though are usually low quality and contain inaccuracies and errors. As a result, modern science and business face the problem of learning from unreliable data sets.

In this work, we provide a generic approach that is based on *verification* of only few records of the data set to guarantee high quality learning outcomes for various optimization objectives. Our method, identifies small sets of critical records and verifies their validity. We show that many problems only need $\mathrm{poly}(1/\varepsilon)$ verifications, to ensure that the output of the computation is at most a factor of $(1 \pm \varepsilon)$ away from the truth. For any given instance, we provide an *instance optimal* solution that verifies the minimum possible number of records to approximately certify correctness. Then using this instance optimal formulation of the problem we prove our main result: "every function that satisfies some Lipschitz continuity condition can be certified with a small number of verifications". We show that the required Lipschitz continuity condition is satisfied even by some NP-complete problems, which illustrates the generality and importance of this theorem.

In case this certification step fails, an invalid record will be identified. Removing these records and repeating until success, guarantees that the result will be accurate and will depend only on the verified records. Surprisingly, as we show, for several computation tasks more efficient methods are possible. These methods always guarantee that the produced result is not affected by the invalid records, since any invalid record that affects the output will be detected and verified.

Keywords: unreliable data set, verification, Lipschitz continuity

1. Introduction

Modern science and business involve using large amounts of data to perform various computational or learning tasks. The data required by a particular research group or enterprise usually contain *errors and inaccuracies* because of the following reasons:

- the validity of the data changes dynamically. For example data involving home locations of customers or employees are not constant over time. Hence a set of data collected in a particular time frame probably is not going to remain valid for the future,
- the data might be provided by other entities or collected online from a source that has no certification for their validity. For example data collected from crowdsourcing environments.

We call a set of data with the property that only a subset of them is valid an *unreliable data set*. The goal of this paper is to develop theoretically established methods that lead to *certified computation* over such data sets. Towards this goal we assume that we have the ability to *verify* the validity of a record in our data set. Usually this verification process is costly and hence it doesn't make sense to verify all the records in our data set every time we want to compute a function on them. On the other hand if the majority or the most important part of the data are invalid then trying to find a valid subset could lead to essentially verifying the entire data set. In our work we introduce the concept of *learning with certification*, in which we can distinguish between the following scenarios

1. the value of the function that we computed on the unreliable data set is close to the value of the function computed on the valid subset of our data set,
2. there exists at least one invalid record that could dramatically charge the value of the function that we want to compute.

by verifying only a small number of records.

Computations in Crowdsourcing. Crowdsourcing Doan et al. (2011) is a popular instantiation of an unreliable data set, where records are provided by a very large number of workers. These workers may need to put significant effort to extract high quality data and without the right incentives they might choose not to do so giving, as a result, very noisy and unreliable reports. Experimental evidence Kazai et al. (2011); Vuurens et al. (2011); Wais et al. (2010) suggests that There are a large number of examples where crowdsourcing fails in practice because of the unreliability of the data that it produces. An anecdotal failure of crowdsourcing is the example of Walmart's mechanism that made the famous rapper Pitbull travel to the remote island of Kodiak, Alaska, see e.g., ABCNews (2012). In 2012, Walmart asked their customers to vote, through Facebook, their favorite local store. The store with the most votes would host a promotional performance by Pitbull. Probably as a mean joke, a handful of people organized an #ExilePitbull campaign, inviting Facebook users to vote for the most remote Walmart store, at Kodiak. The campaign went viral and Pitbull performed at Kodiak, in July 2012. While the objective of Walmart was to learn the location that would maximize attendance to the concert, the resulting outcome was terribly off because the incentives of the workers were misaligned.

Our work is motivated by these observations and aims through the use of verification to provide a generic approach that guarantees high quality learning outcomes. Verification can be implemented either directly, in tasks such as peer grading, by having an expert regrade the assignment, or indirectly, e.g. in the Walmart example by verifying the locations of the voters. The main challenge in our framework it to minimize such verifications since they can be very costly.

1.1. Our Model and Results

A data set is a set of records $\mathcal{N}$. The set $\mathcal{N}$ may contain, apart from the valid subset of records $\mathcal{T}$, a set of records $\mathcal{N} \setminus \mathcal{T}$ that are invalid due to reasons that we described earlier. But how much does the presence of these invalid records affect the output of the computation? The answer to this question depends on the number but also on the *importance* of the invalid records, where the importance depends on the specific computation task that we want to run.

In order to assess whether the computed output is accurate, we can verify the validity of some of the records. Our goal is to verify as few reports as possible and eventually be confident that the output of the computation is accurate. At this point we have to define a measure to evaluate the accuracy of an output. Ideally, an accurate output is the output that we would get if all the records were valid. Such a benchmark, however, is impossible to achieve as the correct values of the invalid records are unobservable. We instead focus on a simpler benchmark. We want to decide whether given an unreliable data set the output of the computation based on $\mathcal{N}$ is close to the output of the computation based only on $\mathcal{T}$. That is, if we could see which records are invalid and perform the computation task after discarding them, would the output of the computation be close to the current value?

Certification Schemes A positive answer to the question above is called *certification* of the computation task based on the unreliable data set $\mathcal{N}$. A negative answer is a witness that at least one record in $\mathcal{N}$ is invalid. Our first goal of this paper is to provide certification schemes for general computation tasks that verify only a small number of records and can distinguish between these two cases.

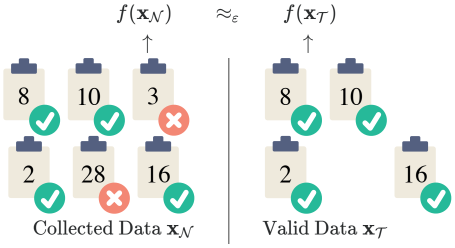
$f(\mathbf{x}_\mathcal{N})$
$\approx_\varepsilon$
$f(\mathbf{x}_\mathcal{T})$
8
10
3
2
28
16
Collected Data $\mathbf{x}_\mathcal{N}$
8
10
2
16
Valid Data $\mathbf{x}_\mathcal{T}$

Figure 1: The property that a certification scheme certifies.

As a toy example of these models let us consider the simple function $f(\mathbf{x}_\mathcal{N}) = \max_{i\in\mathcal{N}} x_i$, where we assume that each record is a real number $x_i \in \mathbb{R}$. For the certification task we want to check whether $f(\mathbf{x}_\mathcal{N}) = f(\mathbf{x}_\mathcal{T})$ or not. This can be easily done by checking whether the record $i^* = \arg\max_{i\in\mathcal{N}} x_i$ is valid or not.

Not all functions though have such efficient deterministic and exact correction schemes. For several functions, we can obtain randomized certification schemes that succeed with high probability and certify that the output is close up to a multiplicative factor. Moreover, for some functions it might not even be possible to efficiently certify them without certifying almost everything. One extreme such example is a threshold function that is 1 if all records are valid and 0 otherwise, $\mathbb{I}_{\mathcal{N}=\mathcal{T}}$ where we cannot obtain any meaningful approximation without verifying $\Omega(n)$ records of $\mathcal{N}$.

In Section 3, we provide efficient certification schemes for many different functions. Our results are the following:

- **sum function.** We start by presenting a randomized scheme for certifying the sum of all records that uses only $O(\frac{1}{\varepsilon})$ verifications to certify correctness up to a multiplicative factor $1 \pm \varepsilon$. This is a very useful primitive that can be used in several different tasks: For example for computing the average, we can compute and certify the total sum of records and divide by the total number of records which we can also certify as another summation task. Another example is the *max-of-sums* function, where as in the Walmart example we presented earlier, agents vote on different categories and the goal is to compute the total category that has the maximum number (or sum) of valid records. This can be easily certified by computing the max of all sums of records and certifying that this sum is approximately correct.
- **functions given by linear programs.** We then study the set of functions expressible as LPs where the input data corresponds to either variables or constraints of the LP. We show that for functions expressible as packing or covering LPs, only $O(\frac{1}{\varepsilon})$ verifications to certify

correctness up to a multiplicative factor $1 \pm \varepsilon$ while for more general LPs we provide a deterministic scheme that depends on the dimensionality (number of variables or constraints).

- **instance-optimal schemes.** To study more general functions, we devise a linear program that characterizes (up to a constant factor) for any given instance the minimum number of verifications needed for approximate certification. We show that even though optimal certification schemes may be arbitrarily complex, there are simple schemes that verify records independently that are almost-optimal.
- **Main theorem for certification.** Our main most general result is that we can provide explicit solutions to the instance optimal linear program for a large class of different objectives that satisfy a w-Lipschitz property. We illustrate the flexibility of this constraint by showing that even very complex functions that correspond to NP-hard problems satisfy the w-Lipschitz property. Specifically, using our general theorem, we prove this for the TSP problem and the Steiner tree problems where we show that the certification complexity is only $O(\frac{1}{\varepsilon})$. These capture settings where agents report their locations in a metric space and the goal is to design an optimal tour that visits all of them (TSP) or connecting them in a network by minimizing total cost (Steiner tree).

Correction Schemes Although very useful, the certification process fails when at least one invalid record is found. Naturally the next question to ask is how we can proceed in order to actually compute the value of the function that we are interested in by throwing away the invalid records. In a worst case example where all records are invalid, we would need to verify all the agents to complete the correction task. To get a more meaningful and realistic measure of the verification complexity of a correction task, we carefully define it in terms of a budget B. For verification complexity V, we assume that the designer has an initial budget for verifications $B = V$ which decreases as he performs verifications but might increase every time he finds an invalid record. The rationale behind the increase is that verifications of incorrect records lead to their removal from the data set, which makes it more accurate. Therefore, in this model we measure the number of verifications needed to correct one incorrect record. We distinguish correction schemes into two models depending on the budget increase.

In the *weak correction* model, the budget increases by V every time an invalid record is found. This means that finding an invalid record allows us to restart the process from the beginning.

In the *strong correction* model, the budget does not increase but does not decrease either. This means that verification of invalid records is costless.

Of course, as we said, in the worst case a correction scheme has to verify all the data in the data set which is not realistic. However, during the correction procedure for specific tasks it would be reasonable to have an upper bound on the number of invalid data that we are willing to verify before dismissing the entire data set for being too corrupted. In particular, if the correction scheme succeeds within the verification budgets we have an accurate output to our computation task. Otherwise, we can conclude that our data set is too corrupted and hence we need to collect the data from the beginning.

Notice that in the example of the $\max$ function, if the certification fails then we can continue by checking the second largest record and so on until we find a valid record which will give us the value $f(\mathbf{x}_\mathcal{T})$ precisely. However, strong correction schemes are much harder to obtain than weak ones in general.

If our computation or learning task has a deterministic certification scheme, e.g. the $\max$ function, it is easy to obtain weak correction schemes by repeating the certification scheme until success.

For randomized schemes though, one needs to be more careful as it is possible that errors can accumulate. This is easy to fix by requiring that the certification scheme fails with probability at most $1/n$. However, this increases the total weak-verification complexity by a logarithmic factor.

In Section 5, we prove our main result for weak correction schemes which implies that such an increase is not necessary and it is possible to obtain weak correction schemes with the same complexity as the underlying certification scheme (up to constant factors). To do this, we run the certification scheme many times and do not stop the first time it succeeds but continue until the total number of successes is more than the number of failures. A random walk argument guarantees that this produces the correct answer with constant probability. If the objective function is not monotone, additional care is needed to get the same guarantee.

While weak correction schemes with good verification complexity exist for all tasks that we can efficiently certify, strong correction schemes are more rare. In Section 6, we show that it is possible to obtain strong correction schemes for the sum function using only $O(\frac{1}{\varepsilon^2})$ verifications of valid records. Since that many verifications are necessary to get a $1 \pm \varepsilon$ multiplicative approximation for the sum, this implies a gap between the weak and strong correction models. The gap between them can be arbitrarily large though. As an example, the max-of-sums function we discussed earlier has certification and weak-correction complexity $O(\frac{1}{\varepsilon})$, although it is impossible to get a constant factor approximation in the strong correction model without verifying $\Omega(n)$ valid records.

Despite the impossibility of obtaining strong correction schemes even for simple functions such as the max-of-sums, we can show that efficient certification schemes exist for quite general optimization objectives. We prove (Theorem 12) a very interesting and tight connection of strong correction schemes with *sublinear algorithms that use conditional sampling* Gouleakis et al. (2017). We can exploit this connection to directly obtain efficient strong correction schemes. This gives efficient schemes for general optimization tasks such as clustering, minimum spanning tree, TSP and Steiner tree that capture settings where agent reports lie on some metric space.

1.2. Related Work

Our certification task resembles the task of property testing, as formalized in Goldreich et al. (1996), where one has to decide whether the data has a particular property versus being ε-far from it in some distance metric. In our case, the property we want to test is whether the evaluation of the function on all the collected data is equal to its evaluation on the subset of the valid data only.

Our correction task is related to a large body of work in statistics on how to deal with noisy or incomplete datasets. Several methods have been proposed for dealing with missing data. The popular method of imputation Rubin (1987); Little and Rubin (2002); Schafer (1997) corrects the dataset by filling in the missing values using the maximum likelihood estimates.

In addition, the field of robust statistics Hampel et al. (1980); Huber (2011) deals with the problem of designing estimators when the dataset contains random or adversarially corrupted datapoints. Several efficient algorithmic results have recently appeared in the context of robust parameter estimation and distribution learning Diakonikolas et al. (2016, 2017a, 2018, 2017b); Lai et al. (2016). The goal of these works is to learn the parameters of a multidimensional distribution, that belongs to a known parametric family, while a constant fraction of the samples have been adversarially corrupted. Charikar et al. (2017) deal with parameter estimation in cases where more than half of the dataset is corrupted and identification is impossible by providing a list of candidate estimates. They show that the correct estimate can be chosen as long as a small "verified" set of data is provided. In contrast to Charikar et al. (2017), the verification oracle in our model allows us to verify any subset of datapoints but verification is costly. Steinhardt et al. (2016) consider similar verification access to the dataset in crowdsourced peer grading settings.

Selective verification of datapoints has also been explored in the context of mechanism design. Fotakis et al. (2016) study mechanisms with verification and achieve truthfulness by solving a task similar to certification in social choice problems.

Finally, another related branch of literature considers the task of correcting datasets through local queries (Jha and Raskhodnikova (2011); Blum et al. (1990); Bhattacharyya et al. (2012); Saks and Seshadhri (2010); Ailon et al. (2008); Canonne et al. (2016)). For example, using local queries, Ailon et al. (2008) correct datasets to ensure monotonicity and other structural properties. Canonne et al. (2016) solve similar local correction tasks for noisy probability distributions.

2. Model and Preliminaries

Notation For $m \in \mathbb{N}$ we denote the set $\{1, \cdots, m\}$ by $[m]$. Let $\mathcal{N} = [n]$ be the set of all records of the data set and $\mathcal{T} \subseteq \mathcal{N}$ be the subset of records that are valid. The set $\mathcal{T}$ is unknown to the algorithm. Suppose we are given an input $\mathbf{x}_{\mathcal{N}} = (x_1, x_2, \cdots, x_n)$ of length n, where every x_i belongs to some set $\mathcal{D}$. Let $\mathbf{x}_{\mathcal{T}} = (x_j)_{j \in \mathcal{T}}$ be a vector consisting only of the coordinates of $\mathbf{x}$ that are in $\mathcal{T}$. Our general goal is to approximate the value of a symmetric function $f : \mathcal{D}^* \to \mathbb{R}_+$ on input $\mathbf{x}_{\mathcal{T}} \in \mathcal{D}^*$. Finally, Every input x_j with $j \in \mathcal{T}$ is called valid and the rest $\mathcal{N} \setminus \mathcal{T}$ are called invalid. We consider two different tasks; *certification* and *correction*.

In the **certification task**, we count the total number of verifications of records needed to test between the following two hypotheses:

(H1)

$$f(\mathbf{x}_{\mathcal{N}}) \in \left[1 - \varepsilon, \frac{1}{1 - \varepsilon}\right] \cdot f(\mathbf{x}_{\mathcal{T}})$$

(H2) there exists a record i such that $i \notin \mathcal{T}$.

We allow a small probability δ that the algorithm fails to find a witness, i.e.

$$\mathbb{P}\left(f(\mathbf{x}_{\mathcal{N}}) \notin \left[1 - \varepsilon, \frac{1}{1 - \varepsilon}\right] \cdot f(\mathbf{x}_{\mathcal{T}}) \wedge \text{ no invalid record found}\right) \leq \delta$$

In the **correction task**, the goal is to always compute an approximation to the correct answer, even when the certification task fails. We consider two models for correction the **weak correction** and the **strong correction**.

In the **weak correction** model after catching an invalid record we are allowed to restart the task and therefore we do not count the number of verifications that we already used before catching the invalid record. So if we have the guarantee that a weak correction scheme uses $v(n, \varepsilon)$ verifications and during the execution of the scheme we find k invalid records, then the total number of verifications used is at most $(k + 1) \cdot v(n, \varepsilon)$.

In the **strong correction** model instead of restarting every time we find an invalid record, we just ignore the data of this record and we also don't count them in the number of verifications. So if we have the guarantee that a strong correction scheme uses $v(n, \varepsilon)$ verifications and during the execution of the scheme we find k invalid records, then the total number of samples used is at most $k + v(n, \varepsilon)$.

3. Certification Schemes for Linear Programs

In this section, we present examples of certification schemes for frequently arising problems such as computing the sum of values or functions that can be expressed as a linear programs. In the next section we will see the more general statement about certification schemes for functions that satisfy a general Lipschitz continuity condition.

3.1. Computing the Sum of Records

One of the most basic certification tasks is computing the sum of the values of the records. For this task, we are given n positive real numbers $x_1, x_2, \ldots, x_n$ each one comming from a record in our data set. Our goal is to certify whether the sum of all the records is closed to the sum of the subset of records that are valid, i.e. belong to $\mathcal{T}$.

More formally, we want to check with probability of failure at most $\delta > 0$ whether $\sum_{i \in \mathcal{N}} x_i \in \left[1 - \varepsilon, \frac{1}{1-\varepsilon}\right] \cdot \sum_{i \in \mathcal{T}} x_i$ or there is at least one record i such that $i \notin \mathcal{T}$. We show that there exists an efficient certification scheme for this task:

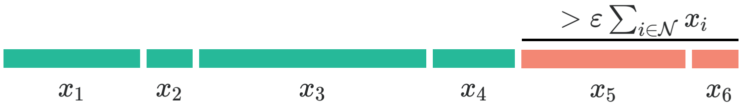

Figure 2: If the invalid records make up more than ε fraction of the total sum, there is at least ε probability that an invalid record is found with a single verification.

Lemma 1 *Let $x_1, x_2, \ldots, x_n \geq 0$ be the values of the records in $\mathcal{N}$ and $f(\mathbf{x}_{\mathcal{N}}) = \sum_{i \in \mathcal{N}} x_i$. Consider the probability distribution $p_i = \frac{x_i}{\sum_j x_j}$ which assigns to each record a probability proportional to its value x_i. Verifying $k = \Theta(\frac{1}{\varepsilon} \log(1/\delta))$ records sampled independently from p, guarantees that the certification task succeeds with probability at least $1 - \delta$.*

Proof Since $\mathcal{T} \subseteq \mathcal{N}$ and x_i's are positive numbers, the inequality $\sum_{i \in \mathcal{N}} x_i \geq (1 - \varepsilon) \sum_{i \in \mathcal{T}} x_i$ holds trivially. If the inequality $\sum_{i \in \mathcal{N}} x_i \leq \frac{1}{1-\varepsilon} \sum_{i \in \mathcal{T}} x_i$ does not hold, we can bound the probability that all of the k verifications fail to find an invalid record, as follows:

The probability that a single verification fails to find an invalid record is $\sum_{i \in \mathcal{T}} p_i = \frac{\sum_{i \in \mathcal{T}} x_i}{\sum_{i \in \mathcal{N}} x_i} < 1 - \varepsilon$.

Therefore the probability that all k verifications fail is at most $(1 - \varepsilon)^k$. Setting $k = \Theta(\frac{1}{\varepsilon} \log(1/\delta))$, we guarantee that an invalid record is found with probability at least $1 - \delta$. ■

3.2. Functions given by Linear Programs

We now extend the previous results for the sum function to more general objective functions that can be represented as linear programs. We first consider the special case of packing and covering LPs while later we present a result for general linear programs.

Packing LP

$$\begin{array}{lll} \max_y & \sum_{i \in \mathcal{N}} c_i & y_i \\ \text{s.t.} & \sum_{i \in \mathcal{N}} a_{ij} & y_i \le b_j, \quad j = 1, \dots, m \\ & & y_i \ge 0, \quad i \in \mathcal{N} \end{array}$$

Covering LP

$$\begin{array}{lll} \min_x & \sum_{j=1}^{m} b_j & x_j \\ \text{s.t.} & \sum_{j=1}^{m} a_{ij} & x_j \ge c_i, \quad i \in \mathcal{N} \\ & & x_j \ge 0, \quad j = 1, \dots, m \end{array}$$

Packing and Covering LP's are parameterized by the *non-negative* parameters a_{ij}, b_j, c_i. We assume that each record i contains all parameters under his control, i.e. the value c_i and a_{ij} for all j, while the parameters b_j are accurately known in advance.

Packing LPs capture settings where several resources (each available in a quantity b_j) are to be divided among a set of agents in the system and agents report how much of each resource they need (given by a_{ij}) and how much value they can generate if they are given the resources they ask for (given by c_{ij}). Our goal is to compute an efficient allocation to agents that maximizes the total value generated. For the certification task, we want to certify that the total value generated by the true agents in an optimal allocation is close to the value computed under the possibly incorrect reports. We show that efficient certification schemes exist by extending the certification scheme presented for the sum function:

Lemma 2 *Let $a_{ij}, c_i \ge 0$ be values contained in the records $\mathcal{N}$ and y^* be the optimal solution to the packing LP. Consider the probability distribution $p_i = \frac{y_i^* c_i}{\sum_j y_j^* z_j}$ which assigns records a probability proportional to their computed value $y_i^* c_i$. Verifying $k = \Theta(\frac{1}{\varepsilon} \log(1/\delta))$ records sampled independently from p, guarantees that the certification task for the packing LP succeeds with probability at least $1 - \delta$.*

To see why this lemma holds, notice that the value $\sum_{i \in \mathcal{N}} c_i y_i^*$ computed using all the records $\mathcal{N}$ is higher than the value $\sum_{i \in \mathcal{T}} c_i \bar{y}_i$ computed using only the valid records $\mathcal{T}$. Moreover, if $\sum_{i \in \mathcal{T}} c_i y_i^* \ge (1 - \varepsilon) \sum_{i \in \mathcal{N}} c_i y_i^*$, then it must be that $\sum_{i \in \mathcal{T}} c_i \bar{y}_i \ge (1 - \varepsilon) \sum_{i \in \mathcal{N}} c_i y_i^*$ as well, since setting $y_i = y_i^*$ for $i \in \mathcal{T}$ and $y_i = 0$ otherwise is a feasible solution to the packing LP under the valid records. Finally, if $\sum_{i \in \mathcal{T}} c_i y_i^* < (1 - \varepsilon) \sum_{i \in \mathcal{N}} c_i y_i^*$, it means that invalid records contribute more than an ε fraction of the total value and thus an invalid record can be easily found as in the previous case of the sum function.

Covering LPs naturally capture various settings with public goods where the designer wants to introduce new goods to satisfy all the demands coming from the records of our data set, but minimizing the total cost at the same time. In facility location problems, the designer wants to open facilities so that all a set of agents have access to at least one facility and the agents report which locations are accessible to them.

Certification schemes for covering LPs are less direct than previous examples, but can be easily obtained through LP duality. As the dual of a covering LP is a packing LP which has the *exact same* *value*, we can use the certification scheme of Lemma 2 to certify that value. We directly get the following:

Lemma 3 *Let $a_{ij}, c_i \geq 0$ be values in the records $\mathcal{N}$. Verifying $k = \Theta(\frac{1}{\varepsilon}\log(1/\delta))$ records sampled independently according to a distribution p given by the solution to the dual packing LP, guarantees that the certification task for the covering LP succeeds with probability at least $1-\delta$.*

General LPs can be written in the form of a packing or a covering LP but have arbitrary (possibly negative) parameters a_{ij}, b_j, c_i. The value of such LPs is harder to certify in general as a lot more verifications than before might be needed. However, we can show that m verifications suffice to certify their value exactly.

Lemma 4 *Let a_{ij}, c_i be (possibly negative) values contained in the records $\mathcal{N}$. The certification complexity for general LPs (written in the form of packing or covering LPs above) is at most m.*

To see why this is true, notice that in the covering LP formulation, the optimal value is given by at most m tight constraints as there are only m variables. Verifying the m records relevant to those constraints guarantees that the optimal value of the LP under only the value of the records is equal to the computed one. This is because only those m constraints determine the optimal value and even if every other constraint i was dropped (i.e. because $i \notin \mathcal{T}$) the value would remain the same. The result also holds for general LPs under the packing LP formulation by LP duality.

4. Certification Schemes for w-Lipschitz Functions and Applications

In this section, we present a unified way of finding *almost-optimal certification schemes*. For a given a function f, a desired approximation parameter ε and an instance $\mathbf{x}_{\mathcal{N}}$, we want to compute the "instance-optimal" number of verifications in order to certify that $f(\mathbf{x}_{\mathcal{N}}) \in \left[1-\varepsilon, \frac{1}{1-\varepsilon}\right] f(\mathbf{x}_{\mathcal{T}})$ with probability of failure at most $1/3$. The first result of this section is a structural result. We show that even though optimal schemes may be arbitrarily complex, there are simpler schemes, that verify records independently, which are almost-optimal.

To show this we define for every set $S \subseteq \mathcal{N}$ the probability p_S that the instance-optimal certification scheme $\mathcal{C}^*$ verifies at least one record in S, i.e. $p_S = \mathbb{P}\left(\bigcup_{i\in S} \{\mathcal{C}^* \text{ verifies record } i\}\right)$. For such an event, we say that the *certification scheme verifies* S and for simplicity we denote p_i, the probability that $\mathcal{C}^*$ verifies record i, i.e. $p_i = p_{\{i\}}$.

For the instance $\mathbf{x}_{\mathcal{N}}$, the set of invalid records could be any $S \subseteq \mathcal{N}$. For the certification scheme to work with failure probability at most $2/3$, we must have that $p_S \geq 2/3$ for any subset $S \subseteq \mathcal{N}$ such that $f(\mathbf{x}_{\mathcal{N}})/f(\mathbf{x}_{\mathcal{N}\setminus S}) \notin [1-\varepsilon, 1/(1-\varepsilon)]$. If this doesn't hold for some S, an adversary could choose the set of invalid records to be S and the certification scheme $\mathcal{C}^*$ would fail with probability more than $1/3$. Moreover, even though the optimal certification scheme $\mathcal{C}^*$ may verify records in a very correlated way, we have that $\sum_{i\in S} p_i \geq p_S \geq 2/3$ from a simple union bound. Therefore, the certification scheme $\mathcal{C}^*$ must satisfy the following set of necessary conditions:

$$\sum_{i\in S} p_i \geq 2/3 \quad \forall S \subseteq \mathcal{N} \text{ such that } \frac{f(\mathbf{x}_{\mathcal{N}})}{f(\mathbf{x}_{\mathcal{N}\setminus S})} \notin \left[1-\varepsilon, \frac{1}{1-\varepsilon}\right]$$

By linearity of expectation, the expected total number of verifications that $\mathcal{C}^*$ performs is,

$$\mathbb{E}[\text{total number of verifications}] = \mathbb{E}\left[\sum_{i\in\mathcal{N}} \mathbf{1}\{\mathcal{C}^* \text{ verifies record } i\}\right] = \sum_{i\in\mathcal{N}} p_i$$

The above imply that the value of the following linear program is a lower bound on the total number of verifications needed by the optimal scheme $\mathcal{C}^*$ for this specific instance $\mathbf{x}_\mathcal{N}$.

$$\begin{aligned} \min \quad & \sum_{i\in\mathcal{N}} p_i \\ \text{s.t.} \quad & \sum_{i\in S} p_i \geq 2/3, \quad \forall S \subseteq \mathcal{N}, \tfrac{f(\mathbf{x}_\mathcal{N})}{f(\mathbf{x}_{\mathcal{N}\setminus S})} \notin \left[1-\varepsilon, \tfrac{1}{1-\varepsilon}\right] \\ & 0 \leq p_i \leq 1, \qquad \forall i \in \mathcal{N} \end{aligned} \tag{1}$$

Notice that the solutions to LP (1), do not directly correspond to certification schemes with success probability $2/3$. However, as we show, any solution to LP (1) can be converted to a certification scheme with number of verifications at most twice as many as the optimal value of LP (1) and success probability $2/3$. Since the optimal value of LP (1) lower bounds the instance optimal number of verifications, our derived certification scheme will be a 2-approximation to the instance optimal scheme.

Definition 5 *For a solution $\bar{p}$ of LP (1), we define the certification scheme $\mathcal{C}_{\bar{p}}$ that verifies each record i independently with probability $q_i = \min\{2\bar{p}_i, 1\}$.*

It is clear that the certification scheme $\mathcal{C}_{\bar{p}}$ uses in expectation at most twice as many verifications as the optimal value of LP (1) and the instance optimal scheme. We now show that it also achieves, success probability of $2/3$ as required.

Assume that the subset of valid records is $\mathcal{T} = \mathcal{N} \setminus S$. The probability that the scheme $\mathcal{C}_{\bar{p}}$ does not verify anyone in the set $S = \{s_1, \ldots, s_m\}$ is

$$\mathbb{P}(\mathcal{C}_{\bar{p}} \text{ doesn't verify } \mathcal{T}) = \mathbb{P}((\mathcal{C}_{\bar{p}} \text{ doesn't verify } s_1) \wedge \cdots \wedge (\mathcal{C}_{\bar{p}} \text{ doesn't verify } s_m)) = \prod_{s\in S}(1-q_s)$$

Since $\bar{p}$ is a feasible solution to LP (1), the probability that some record from S is verified is

$$\mathbb{P}(\mathcal{C}_{\bar{p}} \text{ verifies } S) = 1 - \prod_{s\in S}(1-q_s) \geq 1 - \exp\left(-2\sum_{i\in S}\bar{p}_s\right) \geq 1 - \exp(-4/3) \geq 2/3.$$

This means that our certification scheme succeeds with probability $2/3$ using at most twice the optimal number of verifications in expectation. We can amplify the probability of $2/3$, making it arbitrarily close to one by repeating the certification scheme. Since the repetitions are independent and each of them fails with probability at most $1/3$, after r repetitions the total probability of failure is 3^{-r}. Repeating $r = \log(1/\delta)$ times, guarantees that for any subset S, the probability that it will be verified is at least $1-\delta$. This result is summarized in the following theorem.

Theorem 6 *For any given function $f : \mathcal{D}^* \to \mathbb{R}$ and any set of valus if records $\mathbf{x}_\mathcal{N}$, a solution to LP (1) corresponds to a certification scheme that verifies records of the data set independently using at most twice as many verifications as the optimal scheme for this instance and succeeds with probability $2/3$. Repeating the scheme $\log(1/\delta)$ times increases the success probability to $1-\delta$.*

Remark We note that the LP (1) has exponentially many constraints and it may be computationally intractable to solve depending on the function. It is very useful though as a tool to uncover the structure of approximately optimal certification schemes. For example, Theorem 6 implies that even though optimal schemes may be arbitrarily complex, there are simpler schemes, that verify records independently, which are almost-optimal.

In the following section, we derive a general methodology to obtain solutions to LP (1) for the very general class of $\mathbf{w}$-Lipschitz functions.

4.1. Certification Schemes for $\mathbf{w}$-Lipschitz Functions

In this section we show how we can use Theorem 6 to get sufficient smoothness conditions on the function f that can be used to provide certification schemes with small number of verifications.

For any record $i \in \mathcal{N}$ we define w_i to be the *weight* of the record i. The weight of record i will be the quantity that will determine the probability that we will verify record i according to the verification scheme that we want to define. We state now the property that we want f to satisfy in order to find a good verification scheme.

Definition 7 *We say that a function $f : \mathcal{D}^* \to \mathbb{R}$ is* $\mathbf{w}$-Lipschitz*, with $\mathbf{w} \in \mathbb{R}_+^n$, if for any $S \subseteq \mathcal{N}$ it holds that $|f(\mathbf{x}_\mathcal{N}) - f(\mathbf{x}_{\mathcal{N} \setminus S})| \leq \sum_{i \in S} w_i$.*

For any function that satisfies this Lipschitz property we can get a good verification scheme that depends on the weight vector $\mathbf{w}$.

Theorem 8 *For any non-negative $\mathbf{w}$-Lipschitz function $f : \mathcal{D}^* \to \mathbb{R}_+$, set of records $\mathcal{N}$ with value $\mathbf{x}_\mathcal{N}$, and real numbers $\varepsilon, \delta > 0$, there exists a certification scheme that uses at most $\frac{4 \sum_{i \in \mathcal{N}} w_i}{3 f(\mathbf{x}_\mathcal{N}) \varepsilon} \log(1/\delta)$ verifications and has probability of success at least $1 - \delta$.*

In Appendix D we present the proof of Theorem 8. Also, in Appendix A.1 and A.2, we present two applications of Theorem 8 to get certification schemes for the *Traveling Salesman (TSP)* and the *Steiner Tree* problems. In both applications, we show that the optimal solution is $\mathbf{w}$-Lipschitz with $\frac{\sum_{i \in \mathcal{N}} w_i}{f(\mathbf{x}_\mathcal{N})} \leq 2$. Hence, the total number of verifications by Theorem 8 is $O((1/\varepsilon) \log(1/\delta))$.

5. Weak Correction Model

We show how starting from a certification scheme, we can obtain a weak-correction scheme with the same verification complexity (up to constants).

Theorem 9 *Suppose that there exists a certification scheme for a function f that uses $q(n, \varepsilon)$ verifications and fails with probability $1/3$. Then, there exists a weak-correction scheme with verification complexity $O(q(n, \varepsilon) \log(1/\delta))$ that outputs an accurate estimate of the function f and fails with probability δ.*

Theorem 9 shows that the certification task we defined in section 3 is already strong enough to perform this seemingly more challenging task. Intuitively, this is because we can run many rounds of certification until we have enough confidence that we have an accurate result while we remove from the dataset any invalid record we might find during these rounds. Indeed, a simple way to make the conversion is to start from a certification scheme with error probability 1/3 and reduce its probability of error to δ/n, by repeating it $\log(n/\delta)$ times. Then use this scheme repeatedly until no more invalid records are detected. By a union bound, the probability of error is at most δ since the process takes at most n steps. Theorem 9 shows a stronger result than the above result showing that the logarithmic dependence on the number of records can be avoided if the stopping time is more carefully chosen. The full proof of Theorem 9 is presented in Appendix B.

6. Strong Correction Model

In section, we show that it is possible to obtain more efficient correction schemes for several problems. We show that this is true for the sum function which implies strong correction schemes for other functions such as the average. In addition, we show strong correction schemes for more general combinatorial tasks through a connection with conditional sampling. However, as we show there are simple functions, e.g. the composition of the max and the sum function, for which good weak correction schemes exist that do not admit strong correction schemes.

6.1. Computing the Sum of Values of Records

Using the formulation in Section 3.1, we get the following result. Its proof appears in Appendix C and shows that $\Theta(1/\varepsilon^2)$ verifications are both necessary and sufficient.

Lemma 10 *Let $x_1, x_2, \ldots, x_n \geq 0$ be the values of the records $\mathcal{N}$ and $f(\mathbf{x}_\mathcal{N}) = \sum_{i\in\mathcal{N}} x_i$. Consider the probability distribution $p_i = \frac{x_i}{\sum_j x_j}$ which selects a record x_i with probability proportional to its value. If we sample M times independently from p until $k = \Theta\left(\frac{1}{\varepsilon^2}\log(1/\delta)\right)$ valid records found, then the estimator $\hat{s} = \frac{k}{M}\sum_{i\in\mathcal{N}} x_i$ lies in $\left[1-\varepsilon, \frac{1}{1-\varepsilon}\right] \cdot \sum_{i\in\mathcal{T}} x_i$ w.p. at least $1-\delta$.*

6.2. Lower Bound for the Maximum of Sums Function

In this section we show that no efficient strong correction scheme exists for the composition of the max and the sum function. More precisely we assume we have a partition $\mathcal{J} = \{\mathcal{N}_1, \ldots, \mathcal{N}_\ell\}$ of the set $\mathcal{N}$ and we want a strong correction scheme for the function $f(\mathbf{x}_\mathcal{N}) = \max_{A\in\mathcal{J}} \sum_{i\in A} x_i$. Lemma 11 shows that any strong correction scheme for f that achieves constant approximation has to verify at least a constant fraction of the records. Its proof can be found in Appendix C.

Lemma 11 *Let $c > 0$. There exists a partition $\mathcal{J} = \{\mathcal{N}_1, \ldots, \mathcal{N}_\ell\}$ of $\mathcal{N}$ and a vector $\mathbf{x}_\mathcal{N} \in \mathbb{R}^n$ such that any strong correction scheme for the function $f(\mathbf{x}_\mathcal{N}) = \max_{A\in\mathcal{J}} \sum_{i\in A} x_i$, that returns an estimate $\hat{s} \in \left[\frac{1}{c}, c\right] \cdot f(\mathbf{x}_\mathcal{T})$ with probability at least $3/4$, must verify $|\mathcal{N}|/4c^2$ records.*

6.3. From Algorithms using Conditional Sampling to Strong Correction Schemes

The design of a strong correction scheme can be challenging since the guarantee is very strong. Our main theorem in this section shows that there is a correspondence of strong correction schemes with *sublinear algorithms using conditional sampling*, introduced recently in Gouleakis et al. (2017). We state here the main theorem for this section and we defer its proof and applications for Appendix E.

Theorem 12 *Any function that can be approximated using k conditional samples admits a strong correction scheme with cost k.*

Acknowledgements

The authors were supported by NSF CCF-1551875, CCF-1617730, CCF-1733808, and IIS-1741137.

References

ABCNews. Rapper Pitbull visits Kodiak http://abcnews.go.com/blogs/entertainment/2012/08/rapper-pitbull-visits-kodiak-alaska-and-gets-bear-repell 2012.

Nir Ailon, Bernard Chazelle, Seshadhri Comandur, and Ding Liu. Property-preserving data reconstruction. *Algorithmica*, 51(2):160–182, April 2008. ISSN 0178-4617. doi: 10.1007/s00453-007-9075-9. URL http://dx.doi.org/10.1007/s00453-007-9075-9.

Arnab Bhattacharyya, Elena Grigorescu, Madhav Jha, Kyomin Jung, Sofya Raskhodnikova, and David P. Woodruff. Lower bounds for local monotonicity reconstruction from transitive-closure spanners. *SIAM Journal on Discrete Mathematics*, 26(2):618–646, 2012. doi: 10.1137/100808186. URL http://dx.doi.org/10.1137/100808186.

M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. In *Proceedings of the Twenty-second Annual ACM Symposium on Theory of Computing*, STOC '90, pages 73–83, New York, NY, USA, 1990. ACM. ISBN 0-89791-361-2. doi: 10.1145/100216.100225. URL http://doi.acm.org/10.1145/100216.100225.

Clément L. Canonne, Dana Ron, and Rocco A. Servedio. Testing equivalence between distributions using conditional samples. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014*, pages 1174–1192, 2014.

Clement L. Canonne, Themis Gouleakis, and Ronitt Rubinfeld. Sampling correctors. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, ITCS '16, pages 93–102, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4057-1. doi: 10.1145/2840728.2840729. URL http://doi.acm.org/10.1145/2840728.2840729.

Moses Charikar, Jacob Steinhardt, and Gregory Valiant. Learning from untrusted data. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 47–60. ACM, 2017.

Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high dimensions without the computational intractability. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 655–664, 2016. doi: 10.1109/FOCS.2016.85. URL https://doi.org/10.1109/FOCS.2016.85.

Ilias Diakonikolas, Gautam Kamath, Daniel M Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Being robust (in high dimensions) can be practical. *arXiv preprint arXiv:1703.00893*, 2017a.

Ilias Diakonikolas, Daniel M Kane, and Alistair Stewart. Statistical query lower bounds for robust estimation of high-dimensional gaussians and gaussian mixtures. In *Foundations of Computer Science (FOCS), 2017 IEEE 58th Annual Symposium on*, pages 73–84. IEEE, 2017b.

Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robustly learning a gaussian: Getting optimal error, efficiently. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 2683–2702, 2018. doi: 10.1137/1.9781611975031.171. URL https://doi.org/10.1137/1.9781611975031.171.

Anhai Doan, Raghu Ramakrishnan, and Alon Y Halevy. Crowdsourcing systems on the world-wide web. *Communications of the ACM*, 54(4):86–96, 2011.

Dimitris Fotakis, Christos Tzamos, and Manolis Zampetakis. Mechanism design with selective verification. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 771–788. ACM, 2016.

Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996*, pages 339–348, 1996. doi: 10.1109/SFCS.1996.548493. URL https://doi.org/10.1109/SFCS.1996.548493.

Themistoklis Gouleakis, Christos Tzamos, and Manolis Zampetakis. Faster sublinear algorithms using conditional sampling. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1743–1757. SIAM, 2017.

Frank R Hampel, Peter J Rousseeuw, Elvezio M Ronchetti, and Werner A Stahel. Robust statistics: the approach based on influence functions. 1980.

Peter J Huber. Robust statistics. In *International Encyclopedia of Statistical Science*, pages 1248–1251. Springer, 2011.

M. Jha and S. Raskhodnikova. Testing and reconstruction of lipschitz functions with applications to data privacy. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 433–442, Oct 2011. doi: 10.1109/FOCS.2011.13.

Gabriella Kazai, Jaap Kamps, Marijn Koolen, and Natasa Milic-Frayling. Crowdsourcing for book search evaluation: impact of hit design on comparative system ranking. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 205–214. ACM, 2011.

Kevin A Lai, Anup B Rao, and Santosh Vempala. Agnostic estimation of mean and covariance. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 665–674. IEEE, 2016.

Roderick J. A. Little and Donald B. Rubin. *Statistical Analysis with Missing Data*. Wiley Series in Probability and Statistics. John Wiley & Sons, 2002. ISBN 9780471183860. Second edition.

Donald B. Rubin. *Multiple imputation for nonresponse in surveys*. John Wiley & Sons, 1987.

Michael Saks and C. Seshadhri. Local monotonicity reconstruction. *SIAM Journal on Computing*, 39(7):2897–2926, 2010. doi: 10.1137/080728561. URL http://dx.doi.org/10.1137/080728561.

Joseph L. Schafer. *Analysis of incomplete multivariate data*. CRC press, 1997.

Jacob Steinhardt, Gregory Valiant, and Moses Charikar. Avoiding imposters and delinquents: Adversarial crowdsourcing and peer prediction. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4439–4447, 2016. URL http://papers.nips.cc/paper/6440-avoiding-imposters-and-delinquents-adversarial-crowdsourcing-and-peer-pre

Jeroen Vuurens, Arjen P de Vries, and Carsten Eickhoff. How much spam can you take? an analysis of crowdsourcing results to increase accuracy. In *Proc. ACM SIGIR Workshop on Crowdsourcing for Information Retrieval (CIR'11)*, pages 21–26, 2011.

Paul Wais, Shivaram Lingamneni, Duncan Cook, Jason Fennell, Benjamin Goldenberg, Daniel Lubarov, David Marin, and Hari Simons. Towards building a high-quality workforce with mechanical turk. *Proceedings of computational social science and the wisdom of crowds (NIPS)*, pages 1–5, 2010.

Appendix A. Applications of Theorem 8

A.1. Optimal Travelling Salesman Tour

In this section we examine the *metric travelling salesman problem* where we are given n points (each provided by one record in $\mathcal{N}$) in a metric space $\mathcal{X}$ and we wish to find the length of the minimum cycle going through each point in the set $\mathcal{T} \subseteq \mathcal{N}$ of correct answers. As usual we let $\mathbf{x}_{\mathcal{N}}$ be the input vector with record values whose coordinates are points in the metric space $\mathcal{X}$. Our goal is to find a certification scheme for this metric travelling salesman problem. That is, the algorithm should either output a sufficiently accurate value (according to (H1)) for the minimum weight cycle going through the points in $\mathbf{x}_{\mathcal{T}}$ or find a invalid record [1]. The following lemma combined with Theorem 8 give us the desired result.

Lemma 13 *Let $f : \mathcal{D}^* \to \mathbb{R}$ be the function mapping a set of points in a metric space $\mathcal{X}$ to their minimum TSP tour and let $v_1 v_2 \dots v_n$ be the minimum TSP tour. Also, let $\mathbf{w} \in \mathbb{R}^n_+ = (w_1, \dots, w_n)$, where $w_i = d(v_{i-1}, v_i) + d(v_i, v_{i+1})$ and the second indices are mod n. Then, f is $\mathbf{w}$-continuous.*

Proof According to definition 7, we need to show that for any $S \subseteq \mathcal{N}$:

$$f(\mathbf{x}_{\mathcal{N}}) \leq f(\mathbf{x}_{\mathcal{N} \setminus S}) + \sum_{i \in S} w_i \tag{2}$$

To see why this inequality is satisfied, let T_R be the minimum TSP tour going through the points in $R = \mathcal{N} \setminus S$ and $T_{\mathcal{N}} = v_1 v_2 \dots v_n$ be the minimum TSP tour that goes through all the points in the set $\mathcal{N} \supseteq R$. Now let $j_1 < j_2 \cdots < j_r$ be the indices at which the points of the set R appear in this TSP tour. Consider two consecutive points $v_{j_k}, v_{j_{k+1}}$ in this sequence and let $P_k = \{v_{j_k+1}, v_{j_k+2}, \dots, v_{j_{k+1}-1}\}$ be the set of consecutive points in the tour $T_{\mathcal{N}}$ between v_{j_k} and $v_{j_{k+1}}$. Clearly, $\forall k : P_k \subseteq S$ and therefore the weights of those points appear in the sum that is in the rhs of equation (2). Now consider the two paths $p_{1,k} = v_{j_k}, v_{j_k+1}, \dots, v_{j_{k+1}-1}$ and $p_{2,k} = v_{j_k+1}, v_{j_k+1}, \dots, v_{j_{k+1}}$ which are both part of $T_{\mathcal{N}}$. We have that:

$$\sum_{i \in P_k} w_i = d(v_{j_k}, v_{j_k+1}) + d(v_{j_{k+1}-1}, v_{j_{k+1}}) + 2 \cdot \sum_{i=j_k+1}^{j_{k+1}-2} d(v_i, v_{i+1}) = l(p_{1,k}) + l(p_{2,k})$$

where $l(\cdot)$ denotes the length of a path. We now consider the walk that goes through all the vertices in $\mathcal{N}$ and has the following two properties:

- It respects the order in which the vertices in R are visited by T_R
- Between any two consecutive such vertices, it follows whichever path among $p_{1,k}$ and $p_{2,k}$ has smaller length in the forward and then backwards direction.

We know that $f(\mathbf{x}_{\mathcal{N}})$ smaller or equal to the walk we have just defined, since the walk goes through all the given points and even repeats the points in R [2]. Thus,

[1] Note that throughout this paper we don't consider the computational complexity of the problems, since we are more interested in the number of verifications needed. Besides that in the case of Euclidean TSP we could use the $(1+\varepsilon)$-approximation algorithm that we know in order to get similar results and avoid NP-completeness.

[2] Since we are working on a metric space, skipping points in the order that we visit them can only decrease the cost.

$$\begin{aligned} f(\mathbf{x}_{\mathcal{N}}) &\leq f(\mathbf{x}_{\mathcal{N}\setminus S}) + \sum_{k=1}^{s} 2 \cdot \min\{(d(v_{j_k}, v_{j_k+1}), d(v_{j_{k+1}-1})\} + 2 \cdot \sum_{i=j_k+1}^{j_{k+1}-2} d(v_i, v_{i+1}) \\ &\leq f(\mathbf{x}_{\mathcal{N}\setminus S}) + \sum_{k=1}^{s} \sum_{i \in P_k} w_i \\ &\leq f(\mathbf{x}_{\mathcal{N}\setminus S}) + \sum_{i \in S} w_i \end{aligned}$$

■

Using lemma 13 and theorem 8, we get the following corollary:

Corollary 14 *Let $f : \mathcal{D}^* \to \mathbb{R}$ be the function mapping a set of points in a metric space $\mathcal{X}$ to their minimum TSP tour. Then, there exists a verification scheme that uses at most $O(\frac{1}{\varepsilon}\log(\frac{1}{\delta}))$ verifications per correction.*

Proof This is a straightforward application of lemma 13 and theorem 8 since $\sum_{i \in \mathcal{N}} w_i$ contains each of the edges in the optimal TSP tour $T_{\mathcal{N}}$ exactly twice. Thus,

$$\sum_{i \in \mathcal{N}} w_i = 2f(\mathbf{x}_{\mathcal{N}})$$

■

A.2. Steiner tree

In the classic Steiner tree problem, the input is a positively weighted graph $G = (V, E, w)$ and the set of vertices V is partitioned into two disjoint sets T and U such that $V = T \cup U$. Usually T is called the set of *terminal* nodes and U the set of *Steiner* nodes. The goal is to compute a connected subgraph of G that has the smallest possible weight and has a set of vertices $T \subseteq V' \subseteq V$ that includes all *terminal* nodes and any number of steiner nodes.

Here, we are going to examine the Steiner tree problem in the following setting: We are given a fixed graph $G = (V, E)$ on $|V|$ vertices and we also have $|\mathcal{N}|$ values from the set of records $\mathcal{N}$. Each record is a node from the set V claiming that this node is in the set $T \subseteq V$ of terminal nodes that need to be connected by the tree. However, the records might be invalid and the algorithm is allowed to do verifications on those records. Let $\mathbf{x}_{\mathcal{N}}$ be the input vector whose coordinates are vertices claimed to be in the set T of terminal nodes. Similarly, let $\mathbf{x}_A$ be a vector containing only a subset $A \subseteq \mathcal{N}$ of those vertices. Our goal is again to be able to either output a sufficiently accurate answer for the cost of the optimal Steiner tree of find an invalid record.

As in the previous section we are going to use theorem 8 to achieve this. The conditions of theorem 8 are satisfied in this case due to the following lemma:

Lemma 15 *Let $G = V, E$ be a graph and $f_G : V^* \to \mathbb{R}$ be the function mapping a set of vertices $T \subseteq V$ to the minimum cost of a steiner tree connecting the vertices in T. Then, there exists a vector $\mathbf{w} \in \mathbb{R}_+^n$ such that f is $\mathbf{w}$-continuous and also $\sum_{i \in \mathcal{N}} w_i = O(f_G(\mathbf{x}_{\mathcal{N}}))$.*

Proof We need to show that there exists a vector $\mathbf{w} \in \mathbb{R}_+^n = (w_1, \dots, w_n)$, such that for any $S \subseteq \mathcal{N}$, the following inequality holds:

$$f(\mathbf{x}_{\mathcal{N}}) \leq f(\mathbf{x}_{\mathcal{N} \setminus S}) + \sum_{i \in S} w_i \tag{3}$$

We start by introducing some notation. Let t be a tree subgraph of G. We denote by H_t the Eulerian graph that results when we double each edge in t. Also, let t_A denote the optimal Steiner tree for the set $A \subseteq V$ of terminal nodes. Thus, $\forall A : f(\mathbf{x}_A) = cost(t_A)$.

Now let t_R be the optimal Steiner tree for some set $R = \mathcal{N} \setminus S \subseteq V$ of terminal nodes. In order to show equation (3), it suffices to show that there exists a tree t and a vector $\mathbf{w} \in \mathbb{R}_+^n$, such that t is a valid Steiner tree for the set $\mathcal{N}$ of terminal nodes and its cost is: $cost(t) \leq cost(t_R) + \sum_{i \in S} w_i$.

In other words, we would like to find a weight vector $\mathbf{w} \in \mathbb{R}_+^n$, such that starting from the Steiner tree t_R and using the weight assigned to the set $S = \mathcal{N} \setminus R$ as budget, we are able to construct a Steiner tree the *covers* the set $\mathcal{N}$. To keep the number of verifications low, we also require this vector to be such that $\sum_{i \in \mathcal{N}} w_i = O(f_G(\mathbf{x}_{\mathcal{N}}))$.

Now fix a specific Euler tour (i.e an ordering of the nodes) $U_{\mathcal{N}}$ for the graph $H_{t_{\mathcal{N}}}$ and also fix an Euler tour U_R for the graph H_{t_R}. Note that the cost of each Euler tour is exactly twice the cost of the corresponding Steiner tree (e.g $cost(U_R) = 2cost(t_R)$ where $cost(\cdot)$ denotes the sum of weights of all edges in the Euler tour or the tree).

We define each weight w_i to be the length of the path from the predecessor to the successor of node i in the ordering $U_{\mathcal{N}}$.

Our goal is to find a new Euler tour which directly corresponds to a valid Steiner tree [3] for the set $\mathcal{N}$ and is within our budget $\sum_{i \in S} w_i$.

Now let $U_{\mathcal{N}} = v_1 v_2 \dots v_n$ be the ordering in which the terminal nodes are visited in the Euler tour of $H_{t_{\mathcal{N}}}$ and $j_1 < j_2 \dots < j_r$ be the indices at which the points of the set $R = \mathcal{N} \setminus S$ appear in this Euler tour. Consider two consecutive points $v_{j_k}, v_{j_{k+1}}$ in this sequence and let $P_k = \{v_{j_k+1}, v_{j_k+2}, \dots, v_{j_{k+1}-1}\} \subseteq S$ be the set of consecutive points in the Euler tour $U_{\mathcal{N}}$ between v_{j_k} and $v_{j_{k+1}}$. Note that the sets P_k are mutually disjoint and therefore: $\sum_{k=1}^{r} \sum_{i \in P_k} w_i \leq \sum_{i \in S} w_i$. Also, $\sum_{i \in P_k} w_i$ is enough budget to add the set of nodes P_k in the ordering U_R between v_{j_k} and $v_{j_{k+1}}$. [4] By repeating this for all $k \in [r]$, we get the desired Steiner tree t that *covers* all nodes in $\mathcal{N}$ and is such that:

$$2 \cdot cost(t_{\mathcal{N}}) \leq 2 \cdot cost(t) \leq 2 \cdot cost(t_R) + \sum_{i \in S} w_i \Rightarrow$$

$$cost(t_{\mathcal{N}}) \leq cost(t_R) + \sum_{i \in S} \frac{w_i}{2} \Leftrightarrow$$

$$f(\mathbf{x}_{\mathcal{N}}) \leq f(\mathbf{x}_{\mathcal{N} \setminus S}) + \sum_{i \in S} w_i'$$

where $w_i' = \frac{w_i}{2}$.

Thus, f is $\frac{\mathbf{w}}{2}$-continuous and also $\sum_{i \in \mathcal{N}} w_i' = \frac{1}{2} \cdot 2 \cdot cost(U_{\mathcal{N}}) = 2 \cdot f(\mathbf{x}_{\mathcal{N}})$. ■

The following corollary is a direct application of lemma 15 and theorem 8:

[3] That is, the traversing each edge of that tree twice and in opposite directions.

[4] To be more precise here, we need an argument similar to the two paths argument in the proof of lemma 13.

Corollary 16 *Let $G = V, E$ be a graph and $f_G : V^* \to \mathbb{R}$ be the function mapping a set of vertices $T \subseteq V$ to the minimum cost of a steiner tree connecting the vertices in T. Then, there exists a verification scheme that uses at most $O(\frac{1}{\varepsilon}\log(\frac{1}{\delta}))$ verifications per correction.*

Appendix B. Proof of Theorem 9

We will first provide a simple analysis when the function f is *increasing* with record values and later extend to the general case.

Proof [Proof of Theorem 9 for increasing functions] Our weak correction scheme works by repeating the certification process enough times so that the number of times it failed is less than the number of times it succeeded. In particular, we model this procedure as a random walk on the integers starting from point C and ending once it reaches 0. We move to the right whenever the round of verifications (i.e an execution of the certification scheme) reveals some invalid record, and we move to the left otherwise.

The random walk is guaranteed to return to the origin eventually since if all invalid records are removed the certification scheme will not be able to find any additional invalid record. The only case that the weak correction scheme fails is if it returns early without removing enough invalid records having a value larger than $f(\mathbf{x}_\mathcal{T})/(1-\varepsilon)$. In such a case, at all points of the random walk the estimate was always larger than $f(\mathbf{x}_\mathcal{T})/(1-\varepsilon)$ which means that the random walk was biased with probability at least $2/3$ to the right. The probability that such a biased random walk reaches the origin is at most $\left(\frac{1/3}{2/3}\right)^C = 2^{-C}$. Setting $C = \log(1/\delta)$ times guarantees a probability of error δ. The number of verifications performed if k invalid records are found is $(C+2k)q(n,\varepsilon)$, thus the total verification complexity is $O(q(n,\varepsilon)\log(1/\delta))$. ■

We will now remove the assumption that the function f is increasing. We again use the same random walk that starts at C and ends at 0 as before. However, instead of outputting the result of the function f on the final subset of records (after all deletions), we will consider every possible intermediate subset of records during the random walk as a candidate for producing an $(1+\varepsilon)$-approximate solution. Note that, at each step i of the random walk, we run a certification scheme on some set $S_i \subseteq \mathcal{N}$. We define a subset $S \subseteq \mathcal{N}$ to be "bad" if $\frac{f(\mathbf{x}_\mathcal{T})}{f(\mathbf{x}_S)} \notin \left[1-\varepsilon, \frac{1}{1-\varepsilon}\right]$ and to be "good" otherwise.

By the definition of the certification scheme, if the set S is "bad", then an invalid record is found with probability at least $2/3$, in which case the random walk moves to the right. Otherwise, we do not have any guarantee on how the random walk will behave.

However, if at all steps the probability of finding an invalid record is more than $3/5$, then the probability that the random walk reaches 0 is less than $(\frac{2/5}{3/5})^C = (\frac{2}{3})^C < \delta/2$ for $C = O(\log(1/\delta))$. Thus given that we returned, with high probability, there must be some set S_i for which the correction scheme accepts with probability more than $2/5$. Note that, this can only be true if the set S_i is good since $2/5 > 1/3$.

At this point, given a list of these subsets, our goal is to find a "good" subset for which the certification scheme accepts with probability more than $1/3$. We know that a "good" subset exists for which the acceptance probability is more than $2/5$. We view the certification process for a subset S as sampling from a Bernoulli random variable. We say that a set S has probability p if the certification process on the set S does not find an invalid record with probability p.

Let $Test(S, \gamma)$ be a test that accepts if the probability of a set S is more than $2/5$ (call such a set "very good") and rejects if it is less than $1/3$. Such a test fails with probability γ requiring $O(\log(1/\gamma))$ samples.

The main idea behing this algorithm is to iteratively run $Test(S, \gamma)$ for all candidate subsets S with varying error probabilities γ to throw out the failing ones until a significant fraction of the subsets in our pool is "good". When this happens, we pick a subset at random and check if it is actually "good" by running $Test(S, \gamma)$ with small γ. We repeat this until we actually find a good subset and output the value on the function f on that subset. To ensure that this will eventually happen, we choose parameters appropriately, so that a constant fraction of the "bad" subsets fail while the "good" subsets pass the certifications with high enough probability.

Let K be the number of candidate subsets S_i. We have that K is equal to the number of invalid records found during the random walk process.

Our algorithm proceeds in rounds until there are at most $K/\log K$ sets remaining. In the t-th round:

- The algorithm runs $Test(S_i, 10^{-t})$ for every set S_i and discards all sets that fail.
- If the number of remaining sets is did not drop by a factor of 2 the algorithm stops and returns a set S_i uniformly at random from the remaining sets.

If the algorithm has not returned after $\log\log K$ steps, then it runs $Test(S_i, 1/K^2)$ for every remaining set S_i and returns one that passes the test.

The proposed algorithm returns a "good" set with probability more than $3/5 - o(1)$. First, notice that a "very good" set will be discarded in the first $\log\log K$ rounds with probability at most $\sum_t 10^{-t} \leq \frac{1/10}{1-1/10} = 1/9$. Hence, if the algorithm did not return after $\log\log K$ rounds, the last step returns a "good" set with high probability.

Now, suppose the algorithm returns at some round t. Let K_{t-1} be the total remaining sets before round t. The probability that the number B_t of "bad" sets remaining after round t to be more than $K_{t-1}/5$ is at most:

$$\mathbb{P}\left(B_t > K_{t-1}/5\right) \leq \exp(-B_{t-1}/10) \leq \exp(-K_{t-1}/50) \leq \exp(-K/(50\log K))$$

This is an exponentially small probability and by a union bound over all $\log\log K$ rounds it is still negligible.

Thus, assuming that $B_t < K_{t-1}/5$ and $K_t > K_{t-1}/2$, a set chosen uniformly at random is "bad" with probability $B_t/K_t \leq 2/5$.

Therefore, a good set is chosen with probability at least $3/5 - o(1)$ and thus by repeating $O(\log(1/\delta))$ times and choosing the median of the values $f(\mathbf{x}_S)$, we have that with probability $1 - \delta/2$, $\frac{f(\mathbf{x}_T)}{f(\mathbf{x}_S)} \in \left[1 - \varepsilon, \frac{1}{1-\varepsilon}\right]$.

The total number times the certification scheme is called is $O(\log(1/\delta)) \sum_t O(2^{-t} K \log 10^t) = O(K \log(1/\delta))$.

Thus, the verification complexity of the weak correction scheme is equal to $O(q(n, \varepsilon)\log(1/\delta))$ and the Theorem follows.

Appendix C. Missing Proof of Section 6

Proof [Proof of Lemma 10] We let the random variable M to be the total number of verifications until we found k valid records and let $\mathcal{M}$ be the set of samples that we observed. Also we define

$Z = |\mathcal{M} \cap \mathcal{T}| / M = k/M$. We claim that

$$\mathbb{P}\left(\frac{\sum_{i \in \mathcal{T}} x_i}{\sum_{i \in \mathcal{N}} x_i} \in \left[1 - \varepsilon, \frac{1}{1-\varepsilon}\right] \cdot Z\right) \geq 1 - \delta$$

given that $|\mathcal{M} \cap \mathcal{T}| \geq k$.

Let $q = \frac{\sum_{i \in \mathcal{T}} x_i}{\sum_{i \in \mathcal{N}} x_i}$. For tha sake of contradiction let $Z > \frac{1}{1-\varepsilon} \frac{\sum_{i \in \mathcal{T}} x_i}{\sum_{i \in \mathcal{N}} x_i}$ then $M < (1-\varepsilon)k/q$. Hence the expected number of valid records if we draw M samples according to the described distribution is at most $(1-\varepsilon)k$. But know using simple Chernoff bounds and the fact that $k \geq \frac{1}{\varepsilon^2} \log(2/\delta)$ we get that with probability at most $\delta/2$ the number of valid records found is at least k.

Similarly we can show that if $Z < (1-\varepsilon)\frac{\sum_{i \in \mathcal{T}} x_i}{\sum_{i \in \mathcal{N}} x_i}$ then with probability at most $\delta/2$ the number of valid records found is at most k. Hence we have

$$\begin{aligned}\mathbb{P}(|\mathcal{M} \cap \mathcal{T}| = k) = {} & \mathbb{P}\left(|\mathcal{M} \cap \mathcal{T}| = k \mid q \in \left[1 - \varepsilon, \frac{1}{1-\varepsilon}\right] \cdot Z\right) \mathbb{P}\left(q \in \left[1 - \varepsilon, \frac{1}{1-\varepsilon}\right] \cdot Z\right) \\ & + \mathbb{P}\left(|\mathcal{M} \cap \mathcal{T}| = k \mid q < (1-\varepsilon)Z\right) \mathbb{P}\left(q < (1-\varepsilon)Z\right) \\ & + \mathbb{P}\left(|\mathcal{M} \cap \mathcal{T}| = k \mid q > \frac{1}{1-\varepsilon}Z\right) \mathbb{P}\left(q > \frac{1}{1-\varepsilon}Z\right)\end{aligned}$$

but from the definition of the strong correction scheme $\mathbb{P}(|\mathcal{M} \cap \mathcal{T}| = k) = 1$ and as we proved

$$\mathbb{P}\left(|\mathcal{M} \cap \mathcal{T}| = k \mid q < (1-\varepsilon)Z\right) \leq \delta/2 \quad \text{and}$$

$$\mathbb{P}\left(|\mathcal{M} \cap \mathcal{T}| = k \mid q > \frac{1}{1-\varepsilon}Z\right) \leq \delta/2$$

therefore

$$\mathbb{P}(|\mathcal{M} \cap \mathcal{T}| = k) = 1 \leq \mathbb{P}\left(|\mathcal{M} \cap \mathcal{T}| = k \mid q \in \left[1 - \varepsilon, \frac{1}{1-\varepsilon}\right] \cdot Z\right) \mathbb{P}\left(q \in \left[1 - \varepsilon, \frac{1}{1-\varepsilon}\right] \cdot Z\right) + \delta$$

which implies

$$\mathbb{P}\left(q \in \left[1 - \varepsilon, \frac{1}{1-\varepsilon}\right] \cdot Z\right) \geq 1 - \delta.$$

This finally implies that our estimator is in the correct range

$$\mathbb{P}\left(Z \cdot \sum_{i \in \mathcal{N}} x_i \in \left[1 - \varepsilon, \frac{1}{1-\varepsilon}\right] \cdot \sum_{i \in \mathcal{T}} x_i\right) \geq 1 - \delta.$$

To see that $\Theta\left(\frac{1}{\varepsilon^2}\log(1/\delta)\right)$ are also necessary let $x_1 = \cdots = x_n = 1/n$ and let $|\mathcal{T}| = |\mathcal{N}| q$ where $q = 1/2$. This instance is identical with estimating the bias of a Bernoulli random variable with error at most ε and since all the x_i's are equal we can assume without loss of generality that at each step we take a uniform sample from $\mathcal{N}$. But it is well known that for estimating a Bernoulli random variable within ε with probability of failure at most δ we need at least $\Theta\left(\frac{1}{\varepsilon^2}\log(1/\delta)\right)$ total samples. Half of those samples are expected to be correct samples and hence the verification complexity for any strong correction scheme is also at least $\Theta\left(\frac{1}{\varepsilon^2}\log(1/\delta)\right)$. ■

Proof [Proof of Lemma 11] We consider a partition $\mathcal{J}$ of $\mathcal{N}$ into n/c^2 sets of size c^2 each with $x_i = 1$ for all $i \in \mathcal{N}$. Let S be a certification scheme that verifies less than $n/4c^2$ records. Then there exists a set $A \in \mathcal{J}$ such that

$$\mathbb{P}(S \text{ verifies some } j \in A) < 1/4.$$

We prove this by contradiction. Let $\mathbb{P}(C \text{ verifies some } j \in A) \geq 1/4$ for all $A \in \mathcal{J}$. Then $\mathbb{E}[\text{verification by } C] \geq \sum_{A \in \mathcal{J}} \mathbb{P}(C \text{ verifies some } j \in A) \geq n/4c^2$ and hence we have a contradiction on the assumption that S verifies less than $n/4c^2$ records. Let $\hat{s}$ be the output estimator of S then we have that

$$\begin{aligned}\mathbb{P}\left(\hat{s} \in \left[\frac{1}{c}, c\right] \cdot f(\mathbf{x}_{\mathcal{T}})\right) &= \mathbb{P}\left(\hat{s} \in \left[\frac{1}{c}, c\right] \cdot f(\mathbf{x}_{\mathcal{T}}) \mid S \text{ verifies some } j \in A\right) \mathbb{P}\left(S \text{ verifies some } j \in A\right) \\ &\quad + \mathbb{P}\left(\hat{s} \in \left[\frac{1}{c}, c\right] \cdot f(\mathbf{x}_{\mathcal{T}}) \mid S \text{ does not verify } A\right) \mathbb{P}\left(S \text{ does not verify } A\right) \\ &< 1/4 + \mathbb{P}\left(\hat{s} \in \left[\frac{1}{c}, c\right] \cdot f(\mathbf{x}_{\mathcal{T}}) \mid S \text{ does not verify } A\right)\end{aligned}$$

Now if we fix $Q \subseteq \mathbb{N}$, we observe that the quantity $\mathbb{P}\left(\hat{s} \in Q \mid S \text{ does not verify } A\right)$ does not depend on $\mathcal{T} \cap A$ since we are conditioning on the event that S does not verify any record in A. Now let j_B be an arbitrary record from the set $B \in \mathcal{J}$. We consider the following two possibilities for the set $\mathcal{T}$.

$$\mathcal{T}_0 = \bigcup_{B \in \mathcal{J}, B \neq A} \{j_B\}$$

$$\mathcal{T}_1 = \mathcal{T}_0 \cup A$$

We observe now that if $\mathcal{T} = \mathcal{T}_0$ then $f(\mathbf{x}_{\mathcal{T}}) = 1$ and if $\mathcal{T} = \mathcal{T}_1$ then $f(\mathbf{x}_{\mathcal{T}}) = c^2$. Now since $\hat{s}$ does not depend on $\mathcal{T} \cap A$ given that S does not verify A we have that we can change $\mathcal{T}$ between $\mathcal{T}_0$ and $\mathcal{T}_1$ without changing the quantity $\mathbb{P}\left(\hat{s} \in Q \mid S \text{ does not verify } A\right)$. Now

- if $\mathbb{P}\left(\hat{s} \in [1, c] \mid S \text{ does not verify } A\right) < 1/2$ then we set $\mathcal{T} = \mathcal{T}_1$ and
- if $\mathbb{P}\left(\hat{s} \in [c(c-1), c^2] \mid S \text{ does not verify } A\right) < 1/2$ then we set $\mathcal{T} = \mathcal{T}_1$.

Observe that one of the two cases has to be true. In any of these we get that

$$\mathbb{P}\left(\hat{s} \in \left[\frac{1}{c}, c\right] \cdot f(\mathbf{x}_{\mathcal{T}}) \mid S \text{ does not verify } A\right) < 1/2.$$

Hence we get that

$$\mathbb{P}\left(\hat{s} \in \left[\frac{1}{c}, c\right] \cdot f(\mathbf{x}_{\mathcal{T}})\right) < 1/4 + \mathbb{P}\left(\hat{s} \in \left[\frac{1}{c}, c\right] \cdot f(\mathbf{x}_{\mathcal{T}}) \mid S \text{ does not verify } A\right) < 3/4$$

and therefore S has to verify at least $n/4c^2$ records. ■

Appendix D. Proof of Theorem 8

Proof We set $p_i = \frac{2w_i}{3f(\mathbf{x}_{\mathcal{N}})\varepsilon}$ and we show that those values satisfy the LP (1). Thus, if we choose to verify record i with probability $\min\{2p_i, 1\}$, we get a valid (ε, δ)-certification scheme.

For any subset $S \subseteq \mathcal{N}$ by the $\mathbf{w}$-Lipschitz property we get that

$$|f(\mathbf{x}_{\mathcal{N}}) - f(\mathbf{x}_{\mathcal{N}\setminus S})| \leq \sum_{i \in S} w_i \Leftrightarrow \left| \frac{2}{3\varepsilon} - \frac{2f(\mathbf{x}_{\mathcal{N}\setminus S})}{3\varepsilon f(\mathbf{x}_{\mathcal{N}})} \right| \leq \sum_{i \in S} \frac{2w_i}{3f(\mathbf{x}_{\mathcal{N}})\varepsilon}.$$

Now if $\frac{f(\mathbf{x}_{\mathcal{N}})}{f(\mathbf{x}_{\mathcal{N}\setminus S})} > \frac{1}{1-\varepsilon}$, we have $\left| \frac{2}{3\varepsilon} - \frac{2f(\mathbf{x}_{\mathcal{N}\setminus S})}{3\varepsilon f(\mathbf{x}_{\mathcal{N}})} \right| > \frac{2}{3}$.
Also if $\frac{f(\mathbf{x}_{\mathcal{N}})}{f(\mathbf{x}_{\mathcal{N}\setminus S})} < 1 - \varepsilon$, we have $\left| \frac{2}{3\varepsilon} - \frac{2f(\mathbf{x}_{\mathcal{N}\setminus S})}{3\varepsilon f(\mathbf{x}_{\mathcal{N}})} \right| > \frac{2}{3\varepsilon(1-\varepsilon)} - \frac{2}{3\varepsilon} \geq \frac{2}{3}$.
Therefore when $\frac{f(\mathbf{x}_{\mathcal{N}})}{f(\mathbf{x}_{\mathcal{N}\setminus S})} \notin \left[1 - \varepsilon, \frac{1}{1-\varepsilon}\right]$, we have $\sum_{i \in S} p_i = \sum_{i \in S} \frac{2w_i}{3f(\mathbf{x}_{\mathcal{N}})\varepsilon} \geq \left| \frac{2}{3\varepsilon} - \frac{2f(\mathbf{x}_{\mathcal{N}\setminus S})}{3\varepsilon f(\mathbf{x}_{\mathcal{N}})} \right| \geq \frac{2}{3}$.
This means that LP (1) is satisfied. Now we can apply Theorem 6 and we conclude that the certification scheme that verifies each record independently with probability $\min\{2p_i, 1\}$, where $2p_i = \frac{4w_i}{3f(\mathbf{x}_{\mathcal{N}})\varepsilon}$, verifies at most $\frac{4\sum_{i \in \mathcal{N}} w_i}{3f(\mathbf{x}_{\mathcal{N}})\varepsilon}$ records and has probability of success at least $2/3$. In order to get probability of success δ we instead verify each record i with probability $2p_i \log(1/\delta)$ and the theorem follows. ■

Appendix E. Complete Statement and applications of Theorem 12

More precisely we are given an input $\mathbf{x}_{\mathcal{N}} = (x_1, x_2, \cdots, x_n)$ of length n, where every x_i belongs in some set $\mathcal{D}$. In this section, we will fix $\mathcal{D} = [\mathcal{D}]^d$ for some $\mathcal{D} = n^{O(1)}$ to be the discretized d-dimensional Euclidean space. Our goal is to compute the value of a symmetric function $f : \mathcal{D}^n \to \mathbb{R}_+$ with input $\mathbf{x} \in \mathcal{D}^n$. We assume that all x_i are distinct and define $\mathcal{X} \subseteq \mathcal{D}$ as the set $\mathcal{X} = \{x_i : i \in \mathcal{N}\}$. Since we consider symmetric functions f, it is convenient to extend the definition of f to sets $f(\mathcal{X}) = f(x)$.

The *conditional sampling model* allows such queries of small description complexity to be performed. In particular, the algorithm is given access to an oracle $\textsc{Cond}(C)$ that takes as input a function $C : \mathcal{D} \to \{0, 1\}$ and returns a tuple (i, x_i) with $C(x_i) = 1$ with i chosen uniformly at random from the subset $\{j \in [n] \mid C(x_j) = 1\}$. If no such tuple exists the oracle returns $\perp$.

The main result of this section is a reduction from any algorithm that uses conditional sampling to a strong correction scheme.

Theorem 17 *An algorithm that uses k conditional samples to compute a function f can produce a strong correction scheme with verification cost k.*

Proof We will show how we can implement one conditional sample using only one verification. We take all the values of the records $x_1, \ldots, x_n$ and we randomly shuffle them to get $x_{\pi_1}, \ldots, x_{\pi_n}$. Then we take one by one the records x_{π_i} with this new order and we check if $C(x_{\pi_i}) = 1$. If yes then we verify x_{π_i} and if it is valid we return it as the result of the conditional sampling oracle. If π_i is invalid then we just ignore this records without any cost and we proceed with the next record. If we finish the records and we found no valid record x_{π_j} such that $C(x_{\pi_j}) = 1$, then we return $\perp$.

It is easy to see that this procedure produces at every step a verified conditional sample. Since the conditional sampling algorithm has only this access to the data we get that any guarantees of the conditional sampling immediately transfer to this corresponding strong correction scheme. ■

The above result gives a general framework for designing strong correction schemes for several computational and learning problems. We give some of these examples below that are based on the work of Gouleakis et al. (2017). For other distributional learning tasks, one can use the conditional sampling algorithms of Canonne et al. (2014) to get efficient strong correction schemes. Some applications of Theorem 12 can be found in Appendix E.

k-means Clustering Let $\mathcal{D}$ be a metric space with distance metric $d : \mathcal{D} \times \mathcal{D} \to \mathbb{R}$, i.e. $d(x, y)$ represents the distance between x and y. Given a set of *centers* Ct we define the distance of a point x from Ct to be $d(x, Ct) = \min_{c \in Ct} d(x, c)$. Now given a set of n input points $\mathcal{X} \subseteq \mathcal{D}$ and a set of centers $Ct \subseteq \Omega$ we define the cost of Ct for $\mathcal{X}$ to be $d(\mathcal{X}, Ct) = \sum_{x \in \mathcal{X}} d(x, Ct)$. The k-means problem is the problem of minimizing the *squared cost* $d^2(\mathcal{X}, Ct) = \sum_{x \in \mathcal{X}} d^2(x, Ct)$ over the choice of centers Ct subject to the constraint $|Ct| = k$. We assume that the diameter of the metric space is $\Delta = \max_{x,y \in \mathcal{X}} d(x, y)$. In this setting we assume that the records contain the points in the d-dimensional metric space.

Corollary 18 *Let $x_1, x_2, \ldots, x_n$ be the points in the d-dimensional metric space $\mathcal{D}$ stored in the records $\mathcal{N}$ and $f(\mathbf{x}_{\mathcal{N}})$ be the optimal k-means clustering of the points $\mathbf{x}_{\mathcal{N}}$. There exists a strong correction scheme with $\tilde{O}(k^2 \log n \log(k/\delta))$ verifications that guarantees a constant approximation of the value optimal clustering, with probability of failure at most δ.*

The proof of this corollary is based on the Theorem 12 and the Theorem 2 from Gouleakis et al. (2017).

Euclidean Minimum Spanning Tree Given a set of points $\mathbf{x}_{\mathcal{N}}$ in $\mathbb{R}^d$, the minimum spanning tree problem in Euclidean space ask to compute the a spanning tree T on the points minimizing the sum of weights of the edges. The weight of an edge between two points is equal to their Euclidean distance. We will focus on a simpler variant of the problem which is to compute just the weight of the best possible spanning tree, i.e. estimate the quantity $\min_{\text{tree } T} \sum_{(x,x') \in T} \|x - x'\|_2$.

Corollary 19 *Let $x_1, x_2, \ldots, x_n$ be the points in $\mathbb{R}^d$ stored in the records $\mathcal{N}$ and $f(\mathbf{x}_{\mathcal{N}}) = \min_{\text{tree } T} \sum_{(x,x') \in T} \|x - x'\|_2$. There exists a strong correction scheme with $\tilde{O}(d^3 \log^4 n/\varepsilon^7) \cdot \log(1/\delta)$ verifications that guarantees an $(1 + \varepsilon)$-approximation of the weight of the minimum spanning tree, with probability of failure at most δ.*

The proof of this corollary is based on the Theorem 12 and the Theorem 3 from Gouleakis et al. (2017).

Remark. Observe that the value of the MST gives a 2-approximation of the metric TSP and the metric Steiner Tree problems. Hence Corollary 19 implies efficient strong correction schemes that achieve constant approximation for those problems as well.