Efficient Estimation Algorithms for Neighborhood Variance and Other Moments

Edith Cohen [*] Haim Kaplan [†]

Abstract

The *neighborhood variance* problem is as follows. Given a (directed or undirected) graph with values associated with each node, compute a data structure that for any given node v and $r \geq 0$, would quickly produce an estimate of the variance of all values of nodes that lie within distance r from v. The problem can be generalized to other moment functions and to arbitrary distance-dependent decay.

These problems are motivated by applications where the relevance of a measurement observed (or data present) at a certain location decreases with its distance, and thus the aggregate value varies by location. The centralized version of the problem is motivated by applications to query processing on graphical databases. The distributed version of the problem falls in a model we recently introduced for *spatially decaying aggregation* and is motivated by sensor or p2p networks.

We present novel algorithms for the centralized and distributed versions of the problem. Our algorithms are nearly optimal, the centralized version requires $\tilde{O}(m)$ time and the distributed version requires polylogarithmic communication per node or edge (depending on assumptions).

1 Introduction

Variance and moments are commonly used and very basic properties of data sets and distributions. We consider these problems in a *spatially-decaying* setting, where values are present at nodes of a graph (or a network). Data items present at one node are relevant to other nodes, yet, the relevance decreases with distance [5]. Thus, each location views the items through a different distribution and is interested in aggregate values accordingly.

The weight of an item as viewed from a certain location is determined by some *decay function* applied to its distance [5]. One example of a decay function is the r-threshold function, which assigns uniform weights to items within distance r and 0 weight otherwise. The respective aggregates are computed over all items present in the r-neighborhood. Generally, a decay function can be any non-increasing function.

The *spatially-decaying moments* problem is to efficiently compute a summary that would allow us to retrieve, for each node v, power ω in some fixed range, decay function, and a point a, the (approximate) weighted average of $|x-a|^\omega$ over items. For an r-threshold decay function, this aggregate value is simply the average of $|x-a|^\omega$ over all values x of items that reside at nodes in the r-neighborhood of v. When instead of an arbitrary value a we use the weighted mean (which varies from node to node), we refer to the problem as *spatially-decaying central moments*; when $\omega = 2$ this is the *spatially-decaying variance*; and for r-threshold decay this is the variance over all values in the r-neighborhood. (see Figure 1 for an example of a network, values, and respective moments.)

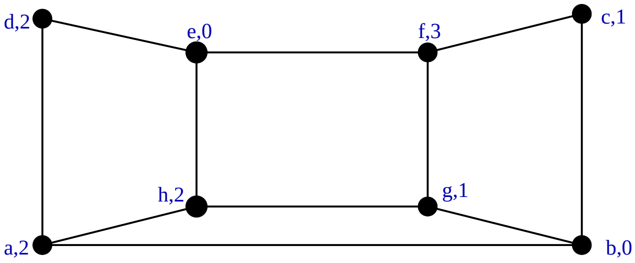

Figure 1: The 2-neighborhood of node a is $\{a, b, h, d, e, g, c\}$. The mean according to 2-threshold decay is thus 8/7, the second moment about 0 is $14/7 = 2$, and the variance is $238/343 = 34/49$. For node c the 2-neighborhood is $\{c, b, f, g, e, a\}$, the respective mean is 7/6, the second moment about 0 is $15/6 = 2.5$ and the variance is $246/216 = 41/36$.

The problem has a centralized variant, where the graph is given as input, and a distributed variant, where the nodes form a network, and the goal is to use a small amount of communication and obtain, at each node, a compact summary that would allow it to answer queries on its own neighborhoods (or arbitrary decay functions). The centralized variant of the problem is motivated

[*] AT&T Research Labs, 180 Park Ave. Florham Park, NJ, USA. Email: `edith@research.att.com`.

[†] School of Computer Science, Tel-Aviv University, Tel Aviv, Israel. Email: `haimk@cs.tau.ac.il`.

by applications in traditional graphical databases, for example, XML documents or analyzing Web structure. The distributed version of the problem is motivated by emerging applications, such as P2P networks and sensor networks, where different data items are present at nodes connected by some low-degree communication network [5].

We present algorithms that yield $(1 \pm \epsilon)$-approximate answers (for a fixed $\epsilon > 0$) for spatially-decaying moments and variance. The size of the summaries are polylogarithmic per node and the running time for the centralized version is $\tilde{O}(m)$. Our algorithms are novel for both the centralized and distributed versions of the problem.

Related work: From an algorithmic standpoint (but less so from an application standpoint), spatially-decaying aggregation generalizes time-decaying aggregation on massive data streams [6] and in particular, sliding-window aggregation for massive data streams [7, 8]: time-decaying aggregation on data streams correspond to spatially-decaying aggregation on directed path graphs, and sliding windows correspond to neighborhoods. Thus, spatially-decaying variance generalizes the sliding-windows variance problem which was studied by Babcock et al [1]. The Babcock et al techniques do not seem to carry over to the spatial setting: Exponential Histograms [7] do not seem to work in the spatial setting (see [5] for discussion). Moreover, their algorithm relies on *exact* computation of the variance and average in each bin of the histogram, an operation that seems fundamentally hard in the spatial setting. It is also not clear if the Babcock et al sliding-window variance algorithm can be extended beyond sliding windows to time-decaying variance under general decay functions [6].

The challenge in spatially-decaying aggregation is that the aggregate value (or summary) is location-dependent. Yet, we do not want to recompute it from the raw distribution for each node, as this would result in quadratic time in the centralized setting and flooding with quadratic communication in the distributed setting. The moments problem imposes additional challenges, since even beyond the issues of computational efficiency, it is not even clear how to summarize the data into a compact representation that captures sufficient information to answer the queries. Distributed computation makes the problem even more challenging, since it basically requires a very efficient way of both summarizing and communicating the essence of the data such that each node can distill the information relevant to it.

Overview and insights:

A basic ingredient in our algorithms is approximate *spatially-decaying counts*, where given binary item values the goal is to produce a *Neighborhood Summary (NH-summary)* which allows to obtain, for each node v and decay function, an approximate decaying count of values. (For the special case of threshold decay function this amounts to estimating, for any given $r \geq 0$, the count in the r-neighborhood of v). Algorithms for centralized computation of NH-summaries were introduced by the first author in [4]. Further new ideas which allow efficient distributed computation of NH-summaries and handling of general decay functions are presented by the authors in [5].

The crux of our approach is a novel technique to summarize a set of values to a poly-logarithmic size summary that allows us to retrieve an approximation of the moment about any constant. These summaries are obtained by applying a logarithmic number of global predicates to each value. Over each predicate we then compute an NH-summary for the count of values that are true for the predicate. Each NH-summary has polylogarithmic-size and can provide, for each decay function, the approximate weight of items that satisfy the predicate.

The key insights we need are in the choice of these predicates. In order to estimate neighborhood moments, we need to somehow be able to preserve and retrieve information about the distances between values and any given point. If we are only interested in moments that are about some globally-fixed point a, the problem is easy: Each value x is bucketized according to its distance from a, $|x - a|$, using buckets with exponentially growing width. We then only need to know an approximate weight of items within each bucket, something we can do using an NH-summary obtained by performing a spatially-decaying count of items in each bucket. The catch, however, is that we want the same summary to work for arbitrary choices of a. [1]

The next approach we consider is partitioning the range uniformly to a polylogarithmic number of bins and producing an NH-summary for each bin. This partially works, and only for nodes, decay functions, and points a, where "most" of the weight lies in a bin that is far from the bin that a lies in (Otherwise, too much information is lost and we can not guarantee the desired $(1 \pm \epsilon)$ approximation). Our approach is based

[1] It may seem that this approach can work for the variance, where we are interested in a moment about a specific point (the mean). Note, however, that the mean is not global and rather depends on the aggregating node and choice of the decay function. Thus, there are many (possibly linearly many) relevant "means" to consider that each value can be aggregated about. We shall see that our solution to the variance relies on the summary which can retrieve moments about arbitrary points.

on extending this attempt, by first *folding* the range of values into a smaller range (the *fold width*), and then uniformly partitioning it into a histogram with a fixed number of bins. The folding essentially amounts to discarding some values and then performing a modulus operation by the fold width. The key property is that all distances between values may decrease and become at most the fold width, but distances that are smaller than the fold width remain the same. After partitioning the fold width uniformly to some constant number of bins we obtain that all distances that are not too big and not too small (that is, are of the order of the fold width) are approximately preserved. We use a logarithmic number of different fold widths that are exponentially decreasing. When computing our estimate on the moment about some value a, we sum over different foldings. Each item is accounted for in many foldings, but there is only one folding that preserves its approximate distance from a. The larger-width foldings would bucketize it together with a, yielding a 0 contribution to the moment and the smaller-width bins will account for a contribution that is much smaller than the one corresponding to its true distance.

The computation of central moments, including the variance, uses the same summaries but requires some additional insights. The exact value of the mean is not known to the aggregating node, and simply computing the aggregate about a $(1 \pm \epsilon)$ approximation of the mean (which can be obtained using decaying sum computations) is not sufficient for obtaining $(1 \pm \epsilon)$ approximation of the variance.

We organize our presentation as follows. In section 2 we state the spatial decay model and the spatially-decaying sum problem [5]. Section 3 describes the summaries by defining the folding functions and predicates that are aggregated as spatially-decaying sums at each node. Section 4 states the algorithm that for a given decay function, point a, and power ω computes an estimate of a moment from the summary. Section 6 is concerned with the variance computation (and other central moments). The correctness proof of the algorithm in Section 4 is given in Section 5. We conclude in Section 7 with a discussion on extending our approach to higher dimensions and k-medians.

2 Preliminaries

We start by defining spatially-decaying aggregation [5], and in particular, the spatially-decaying sum problem [5]. We then proceed and define our problem of spatially-decaying moments.

We model the network as a (directed or undirected) graph $G = (V, E)$, where $V = \{v_1, \dots, v_n\}$ is the set of nodes, and there is an edge between two nodes if and only if the two nodes can communicate. We denote the number of edges by m. Edges can have nonnegative *lengths* associated with them, which correspond to distances. We denote by $\textsc{dist}(v_i, v_j)$ the distance between two nodes v_i and v_j with respect to the shortest-path metric on the edge lengths. Nodes in the network have data items associated with them. Each item $i \in I$ is specified by a pair (f_i, ℓ_i), where f_i is its value and $\ell_i \in V$ is its location.

A *decay function* is a non-increasing function $g(x) \geq 0$ defined for $x \geq 0$. The decay function determines the "weight" of a remote item as a function of its distance. The *decaying weight* of the item i as viewed by a node u is $w_{u,g}(i) = g(\textsc{dist}(u, \ell_i))$.[2] An important family of decay functions are threshold functions $\textsc{Ball}_r$ (for $r \geq 0$), defined by $\textsc{Ball}_r(x) = 1$ for $x \leq r$ and $\textsc{Ball}_r(x) = 0$ otherwise. The corresponding aggregation is over the r-neighborhood, where all data items that lie within distance r have equal weight and all further items have 0 weight. Other natural classes of decay functions are Exponential decay and Polynomial decay (see [5] for details).

An *aggregate function* is a function defined on a multiset of value-weight pairs. The goal of *spatially-decaying aggregation* is to produce summaries with respect to a particular aggregate function (or a class of functions). Each node u obtains a localized summary[3] which allows it, for any given decay function $g()$ (and any aggregate function in the set we consider), to obtain $(1 \pm \epsilon)$-approximate estimates of the value of the aggregate on the multiset $\{f_i, w_{u,g}(i)\}$.

We measure performance by the running time needed to produce these summaries and by the size of the resulting summaries. In the distributed setting we consider the amount of communication per node and storage at each node. In the sequel, $(1 \pm \epsilon)$-approximate estimates (or just estimates) means that by appropriately adjusting constants in our algorithms we can handle any fixed $\epsilon \geq 0$. To simplify the discussion, we ignore in several places scaling of ϵ by a constant factor.

A basic aggregate is the *sum* (weighted sum of values), where the value at node u for decay function $g()$ is

$$S_g(u) = \sum_i w_{u,g}(i) f_i \ .$$

(For the sum problem we assume $f_i \geq 0$ for all i.) In

[2] Our algorithms can be easily extended to a setting where each item has a "local" weight w_i^0, and its decaying weight is $w_i^0 g(\textsc{dist}(u, \ell_i))$. For simplicity of presentation we assume uniform local weights.

[3] In the centralized version of the problem one can also consider a single summary for all nodes. The algorithms we consider here produce separate summaries.

the special case where the values f_i are binary, we refer to this aggregate as the *count*. We define $W_{u,g} = \sum_i w_{u,g}(i)$ to be the decaying count of all items as viewed by u. When u or g are clear from context we will omit them from the subscript of $W_{u,g}$ and $w_{u,g}(i)$. The spatially-decaying sum problem is to obtain summaries such that each location $u \in V$, for any decay function $g()$ can retrieve an $(1 \pm \epsilon)$-approximate value of $S_g(u)$. The summaries produced by spatially-decaying sum algorithms are termed *Neighborhood-summaries* (NH-summaries) [4, 5]. NH-summaries are particularly relevant to us here since we reduce the spatially-decaying moments problem to performing logarithmically-many computations of NH-summaries. As discussed in the Introduction, [4] shows that in $\tilde{O}(m)$ time, we can obtain for each node a polylogarithmic-size NH-summary that gives $(1 \pm \epsilon)$-approximate answers with very high probability[4]. We studied distributed algorithms for NH-summaries in [5]. The communication needed per node depends on the setup. Under some assumptions, e.g., if shortest path trees are pre-computed, the summaries can be obtained using polylogarithmic communication per node.

For a set of items (with values f_i, weights $w(i)$ and $W = \sum_i w(i)$), a point ν, and a power ω, the ω-moment about ν is defined as $\sum_i w(i)(f_i - \nu)^\omega / W$. We refer to the non-normalized quantity $\sum_i w(i)(f_i - \nu)^\omega$ as the *ω-power sum about ν*. We also consider *absolute moments* defined as $\sum_i w(i)|f_i - \nu|^\omega / W$ and the respective *absolute power-sum* $\sum_i w(i)|f_i - \nu|^\omega$. Moments about the mean are termed *central moments* whereas moments about arbitrary choices of ν are termed *raw moments*.

The *spatially-decaying (absolute) power-sums* problem is to produce summaries, according to $\epsilon > 0$ and a range $[\omega_a, \omega_b]$ (where $\omega_b > \omega_a > 0$). The summaries should allow each node u to obtain, for each ν, $g()$, and $\omega \in [\omega_a, \omega_b]$, a $(1 \pm \epsilon)$-approximation of the power sum

$$\mathsf{A}^{\omega}_{\nu,g}(u) = \sum_i \left(w_{u,g}(i)|f_i - \nu|^\omega\right) \ . \tag{2.1}$$

For "pure" moments we use the notation

$$\mathsf{M}^{\omega}_{\nu,g}(u) = \sum_i \left(w_{u,g}(i)(f_i - \nu)^\omega\right) \ . \tag{2.2}$$

Note that the ω-moment is the ratio $\mathsf{M}^{\omega}_{\nu,g}(u)/W_{u,g}$ and the respective absolute moment is $\mathsf{A}^{\omega}_{\nu,g}(u)/W_{u,g}$.

Our algorithms obtain $(1+\epsilon)$-estimates for absolute power sums and thus for pure power sums with integral even values of ω (since for even powers $\mathsf{M}^{\omega}_{\nu,g}(u) \equiv \mathsf{A}^{\omega}_{\nu,g}(u)$).[5]

Central moments have particular significance – the most important such moment is the variance. The (weighted) variance of a set of values is defined as $V = \sum_i w(i)(f_i - \mu)^2/W$, where $\mu = \sum_i w(i) f_i / W$ is the (weighted) mean. The *spatially-decaying central moment* is $\mathsf{M}^{\omega}_{\mu_g(u),g}(u)/W_{u,g}$ and the *spatially-decaying variance* is thus $\mathsf{M}^{2}_{\mu_g(u),g}(u)/W_{u,g} \equiv \mathsf{A}^{2}_{\mu_g(u),g}(u)/W_{u,g}$, where $\mu_g(u) = \sum_i w_{u,g}(i) f_i / W_{u,g}$.

Moments are the ratio of the respective power sum and $W_{u,g}$. Since we can efficiently approximate $W_{u,g}$ using an NH-summary, an approximation of the numerator (the power sum) would yield approximation of the respective moment. In particular, approximate central, raw, absolute or pure moments can be obtained from the respective approximate power sums (and vice versa). In the sequel we will focus on power sums.

3 Foldings and predicates

We develop a technique to compute summaries for the spatially-decaying power sums problem. We assume (this assumption is addressed in Subsection 5.1) that items have integral values in the range $0, \ldots, R-1$.

Our algorithm defines a logarithmic number of global predicates. All nodes apply each predicate to their local items. For each predicate, the system then produces NH-summaries at all nodes. As a result, each node stores a logarithmic number of NH-summaries (one for each predicate). We now provide a high-level description of these predicates. We use mappings which we refer to as *foldings*. Each folding *excludes* part of $[0, R)$ and maps remaining (*included*) values into a range of the form $[0, R/2^{\rho j})$ for some $j \geq 0$ and $\rho \geq 2$. The range of the folding is then partitioned uniformly into B bins, where bin b ($b = 0, \ldots, B-1$) contains values that the folding maps to $[\frac{b}{B} R/2^{\rho j}, \frac{(b+1)}{B} R/2^{\rho j})$. Each bin in each folding corresponds to a predicate. This predicate is "1" for the ith item if and only if f_i is included in the folding and the image of f_i under this mapping falls in the corresponding bin. These NH-summaries allow each node u to obtain, for each folding, each bin, and each decay function g, an approximate decayed count of the items with values that are mapped by the folding to that bin. (For the special case of $\textsc{Ball}_r$ decay function, we can obtain for each r, an approximate number of items within the r-neighborhood of u that are mapped by the folding to that bin.)

The value of B is set according to the desired accuracy and communication tradeoffs. Recall that ρ is

[4] The work of [4] considers only $\textsc{Ball}_r$ decay functions, but we show in [5] that summaries that can support $\textsc{Ball}_r$ decay functions for arbitrary $r \geq 0$ can support arbitrary decay functions.

[5] $(1+\epsilon)$ approximate pure power sums with odd ω are as hard as obtaining exact neighborhood counts [5, 7].

a parameter of our construction which is at least 2. We also define $S = 2^\rho + 1$. We have a folding for each j from 0 to $\rho^{-1}\log_2(R/B)$. For convenience of presentation we assume that $B/2^{(\rho+2)}$ and $\rho^{-1}\log_2(R/B)$ are integral.

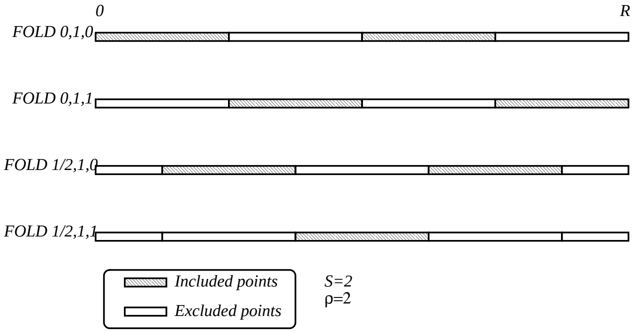

Figure 2: Included parts of the range $[0,\ldots,R)$ for the foldings $\mathrm{FOLD}_{0,1,0}$, $\mathrm{FOLD}_{0,1,1}$, $\mathrm{FOLD}_{1/2,1,0}$ and $\mathrm{FOLD}_{1/2,1,1}$ ($\rho = 2$, for simplicity shown with $S = 2$ although we assume in the analysis that $S = 2^\rho + 1$).

We now define precisely the set of foldings that we use. In addition to $j \in \{0,\ldots,\rho^{-1}\log_2(R/B)\}$, each folding $\mathrm{FOLD}_{c,j,s}$ is specified by two more parameters: $s \in \{0,\ldots,S-1\}$, and $c \in \{0,1/2\}$. We explain the role of these additional parameters next.

The folding mapping can be viewed as follows. The interval $[0,\ldots R)$ is partitioned into consecutive subintervals of size $R/2^{\rho j}$. The $c \in \{0,1/2\}$ determines at what point the partition is started: if $c = 0$ the subinterval boundaries start at 0 and if $c = 1/2$ the boundaries start at $R/2^{\rho j+1}$ ("half" subinterval shift) and end at $R - R/2^{\rho j+1}$.[6] The domain of the folding includes a subset of these subintervals that are spaced exactly S subintervals apart (s determines which of the S possible subsets of subintervals-spaced-S-apart is included.) All included subintervals are then identified (that is, a value f_i in a subinterval $[a_1,a_2]$ is mapped to $f_i - a_1$). Hence, we obtain a mapping of the range $[0,R)$ to a range $[0,R/2^{\rho j})$.

Formally, we have that the domain of the mapping is

$$\mathrm{FOLD}_{c,j,s} = \left\{ x \mid \left\lfloor \frac{(x - cR/2^{\rho j+1})}{R/2^{\rho j}} \right\rfloor \bmod S = s \right\} .$$

For $x \in \mathrm{FOLD}_{c,j,s}$ we define the image as[7]

$$\mathrm{F}_{c,j,s}(x) = (x - cR/2^{\rho j+1}) \bmod R/2^{\rho j} .$$

And the discretization to bins by

$$\mathrm{BIN}_{c,j,s}(x) = \lfloor \mathrm{F}_{c,j,s}(x) B 2^{\rho j}/R \rfloor .$$

An illustration of a range, and different foldings with the respective included items is provided in Figure 2. An illustration of the folding mapping is provided in Figure 3.

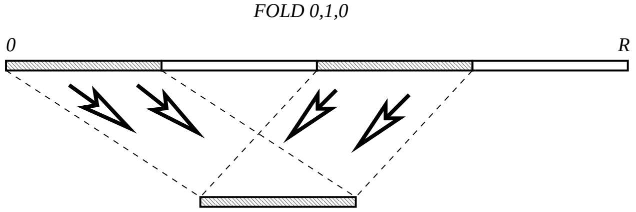

Figure 3: Included parts of the range $[0,\ldots,R)$ for the folding $\mathrm{FOLD}_{0,1,0}$ and the respective mapping of these parts to $[0,\ldots,R/4)$. ($\rho = 2$, shown with $S = 2$ although we assume in the analysis that $S = 2^\rho + 1$).

The reason for using two different partitions for each j, s, with $c = 0$ and $c = 1/2$ is to obtain the property that every subinterval of $[0,R)$ that is of length at most $R/2^{\rho j+1}$ lies within some subinterval of one of the partitions. Formally, we say that an interval $[a,b] \subset [0,R)$ is *intact* by a folding $\mathrm{FOLD}_{c,j,s}$ if all points in the interval are included in $\mathrm{FOLD}_{c,j,s}$ and the folding preserves distances within points in the interval. Equivalently, $[a,b]$ is intact if $[a,b] \subset \mathrm{FOLD}_{c,j,s}$ and $b - a = \mathrm{F}_{c,j,s}(b) - \mathrm{F}_{c,j,s}(a)$. Observe that every interval of the form $[a, a + R/2^{\rho j})$, where $a \bmod R/2^{\rho j+1} = 0$, is a *maximal intact* interval in some folding of the form $\mathrm{FOLD}_{*,j,*}$. We thus have the following property:

LEMMA 3.1. *Any interval $[a,b] \subset [0,R)$ such that $b - a \leq R/2^{\rho j+1}$ is contained in some maximal intact subinterval in a folding of the form $\mathrm{FOLD}_{*,j,*}$.*

LEMMA 3.2. *Consider a maximal intact interval $[a, a + R/2^{\rho j})$ ($a \bmod R/2^{\rho j+1} = 0$) of some $\mathrm{FOLD}_{*,j,*}$. Consider a folding $\mathrm{FOLD}_{c,j-1,s}$ such that $[a, a+R/2^{\rho j})$ is intact in that folding, and let $[d, d + R/2^{\rho(j-1)}) \supset [a, a + R/2^{\rho j})$ be a maximal intact subinterval of $\mathrm{FOLD}_{c,j-1,s}$. Then,*

$$\{\mathrm{BIN}_{c,j-1,s}(x) | x \in [a, a + R/2^{\rho j})\} \cap$$
$$\{\mathrm{BIN}_{c,j-1,s}(x) | x \in [d, d + R/2^{\rho(j-1)}) \setminus [a, a + R/2^{\rho j})\} = \emptyset$$

(the set of bins that cover $\mathrm{F}_{c,j-1,s}([a, a+R/2^{\rho j}))$, and the set of bins that cover $\mathrm{F}_{c,j-1,s}([d, d + R/2^{\rho(j-1)}) \setminus [a, a + R/2^{\rho j}))$ are disjoint.)

Proof. Since $[d, d + R/2^{\rho(j-1)})$ is a maximal intact interval of some $\mathrm{FOLD}_{*,j-1,*}$ we have that $d \bmod R/2^{\rho(j-1)+1} = 0$. It follows that $(a -$

[6] So for $c = 1/2$ we don't get exactly a partition of $[0,R)$ but of $[R/2^{\rho j+1}, R - R/2^{\rho j+1})$.

[7] We use the natural extension of the modulo operation for nonnegative reals.

d) $\textsc{mod}$ $R/2^{\rho j+1} = 0$ and that $(a + R/2^{\rho j} - d)$ $\textsc{mod}$ $R/2^{\rho j+1} = 0$. The bin partition partitions the including interval into intervals of size $(R/2^{\rho(j-1)})/B$. It thus suffices to show that $R/2^{\rho j+1}$ is divisible by $(R/2^{\rho(j-1)})/B$. This in fact holds since

$$\frac{R/2^{\rho j+1}}{(R/2^{\rho(j-1)})/B} = \frac{B2^{\rho(j-1)}}{2^{\rho j+1}} = B/2^{\rho+1} .$$

(The latter is clearly integral since we assumed that $B/2^{(\rho+2)}$ is integral.) □

The following is immediate from our definitions:

Lemma 3.3. *Consider a maximal intact subinterval I of some* $\text{FOLD}_{*,j,*}$. *Then all points $x \in [0,\ldots,R) \setminus I$ such that* $\textsc{dist}(x,I) \leq (S-1)R/2^{\rho j}$ *are not included in the folding.*

Each node stores an NH-summary for each of the B bins in each folding $\text{FOLD}_{c,j,s}$ for all c, j, s. Thus, the communication and storage amount to computing $2BS\rho^{-1}\log_2(R/B)$ NH-summaries.

Consider the viewpoint of some node u. We use the notation

$$B_{c,j,s}(b,g) = \sum_{\{i | f_i \in \text{FOLD}_{c,j,s} \wedge \text{BIN}_{c,j,s}(f_i)=b\}} w_{u,g}(i)$$

for the decaying count of items in the bth bin of $\text{FOLD}_{c,j,s}$. From the NH-summaries available at our node u we can obtain estimates $\hat{B}_{c,j,s}(b,g)$ for $B_{c,j,s}(b,g)$. (for all $g()$, foldings, and bins).

4 Computing power sums from summaries

Given ν, $g()$, and ω, we show how a node u can use its locally-available estimates on $\hat{B}_{c,j,s}(b,g)$ to estimate $\sum_i w_{u,g}(i)|f_i - \nu|^{\omega}$.

For each $j \in \{0,\ldots,\rho^{-1}\log_2(R/B)\}$ we define the intervals

$$I_j = [\max\{\nu - R/2^{\rho(j+1)+2}, 0\}, \min\{\nu + R/2^{\rho(j+1)+2}, R\}] .$$

Then for each $j \in \{0,\ldots,\rho^{-1}\log_2(R/B)\}$ the node selects one folding of the form $\text{FOLD}_{*,j,*}$ (denoted FOLD_{c_j,j,s_j}) as follows. For $j=0$ it uses the folding $\text{FOLD}_{0,0,0}$ (for all ν). For $j>0$ the node selects a folding FOLD_{c_j,j,s_j} such that I_{j-1} is intact. (Existence of such a folding is guaranteed by Lemma 3.1.) We define $\overline{I}_{j-1} = [a_j, a_j + R/2^{\rho j})$ be the maximum intact interval of FOLD_{c_j,j,s_j} which includes I_{j-1}. For convenience, we define $\overline{I}_{-1} \equiv [0,\ldots,R)$ and $\overline{I}_{\rho^{-1}\log_2(R/B)} \equiv \emptyset$. The following lemma summarizes two properties of these intervals that we need to establish correctness of the algorithm.

Lemma 4.1. 1. *$\overline{I}_j$ is contained in I_{j-1} and therefore is intact under* FOLD_{c_j,j,s_j}.

2. $\{\text{BIN}_{c_j,j,s_j}(x) | x \in \overline{I}_j\} \wedge \{\text{BIN}_{c_j,j,s_j}(x) | x \in \overline{I}_{j-1} \setminus \overline{I}_j\} = \emptyset$. *($\overline{I}_j$ is "exactly covered" by the bin partition of* FOLD_{c_j,j,s_j}*.)*

Proof. I_j is of size $R/2^{\rho(j+1)+1}$, and $\overline{I}_j$ is of size $R/2^{\rho(j+1)}$. Thus, $|\overline{I}_j| = 2|I_j|$. Since ν is the midpoint of I_j we have
$\overline{I}_j \subseteq [\max\{\nu - 3R/2^{\rho(j+1)+2}, 0\}, \min\{\nu + 3R/2^{\rho(j+1)+2}, R\}] \subseteq I_{j-1}$.
(the latter holds since $3 \leq 2^{\rho}$.) The second property follows from Lemma 3.2. □

Algorithm PowerSum(ν, w)

```
• $M \leftarrow 0$
• For $j = 0,\ldots,\rho^{-1}(\log_2(R/B))$ do as follows:
• For all $b \in \{0,\ldots,B-1\}$ such that
      $bR/(B2^{\rho j}) \in [0,\ldots,R/2^{\rho j}-1) \setminus \text{F}_{c_j,j,s_j}(\overline{I}_j)$
  (In words, for the bins of the range of $\text{F}_{c_j,j,s_j}$ that
  cover $\text{F}_{c_j,j,s_j}(\overline{I}_{j-1} \setminus \overline{I}_j)$.) do as follows:
• $M \leftarrow M + \hat{B}_{c_j,j,s_j}(b,g) \left| \frac{bR/2^{\rho j}}{B} - \text{F}_{c_j,j,s_j}(\nu) \right|^{\omega}$ .
```

5 Correctness of algorithm PowerSum

Consider an iteration of PowerSum, and the respective folding $\text{FOLD}(c_j, j, s_j)$. First note that items i are classified as either *included* or *excluded* according to whether they belong to FOLD_{c_j,j,s_j}. We further classify included items into either *internal* or *external* as follows. Items with value f_i such that $\text{F}_{c_j,j,s_j}(f_i) \in \text{F}_{c_j,j,s_j}(\overline{I}_j)$ are *internal* for FOLD_{c_j,j,s_j}. Items such that $\text{F}_{c_j,j,s_j}(f_i) \in \text{F}_{c_j,j,s_j}(\overline{I}_{j-1} \setminus \overline{I}_j)$ (i.e., all other items) are *external* for FOLD_{c_j,j,s_j}. So any item is either internal, external, or excluded. For an iteration j and an external item i, we refer to

$$w_g(i)|\text{F}_{c_j,j,s_j}(f_i) - \text{F}_{c_j,j,s_j}(\nu)|^{\omega}$$

as the *contribution* of item i. These classifications are useful as only external items "contribute" to M during the jth iteration. We will use the following property to bound the approximation error.

Lemma 5.1. *Values in $\overline{I}_{j-1}$ that are external for* FOLD_{c_j,j,s_j} *are excluded in* $\text{FOLD}_{c_{j+1},j+1,s_{j+1}}$.

Proof. Values in $\overline{I}_{j-1}$ that are external for FOLD_{c_j,j,s_j} are exactly those in $\overline{I}_{j-1} \setminus \overline{I}_j$. So we have to show that $\overline{I}_{j-1} \setminus \overline{I}_j$ is excluded by $\text{FOLD}_{c_{j+1},j+1,s_{j+1}}$. From

Lemma 3.3, all values that are not in $\overline{I}_j$ and are of distance at most $(S-1)R/2^{\rho(j+1)}$ from $\overline{I}_j$ are excluded under $\mathrm{FOLD}_{c_{j+1},j+1,s_{j+1}}$. Since I_j is of size $R/2^{\rho(j+1)+1}$ and ν is the midpoint of I_j we have that all values that are of distance at most

$$(S-1)R/2^{\rho(j+1)} + R/2^{\rho(j+1)+2} = (4S-3)R/2^{\rho(j+1)+2}$$

from ν and are not in $\overline{I}_j$ are excluded.

Since $S \geq 2^\rho + 1$ we obtain that $(4S-3) \geq 4 \cdot 2^\rho$ and thus

$$(4S-3)R/2^{\rho(j+1)+2} \geq R/2^{\rho j} \ .$$

Recall now that the interval $\overline{I}_{j-1}$ is of size $R/2^{\rho j}$ and contains ν, thus it must be the case that all points in $\overline{I}_{j-1} \setminus \overline{I}_j$ are excluded by $\mathrm{FOLD}_{c_{j+1},j+1,s_{j+1}}$. □

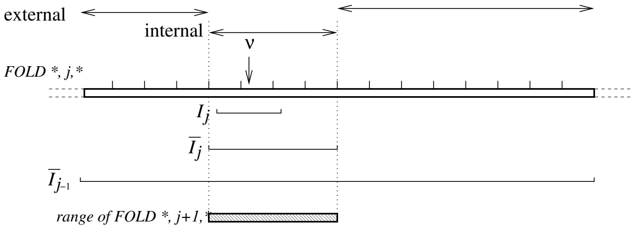

Figure 4: A folding selected according to some ν ($B = 16$, $\rho = 2$, and thus $B2^{-(\rho+2)} = 1$). The figure shows the value ν, $\overline{I}_{j-1}$ (the maximal intact interval in $\mathrm{FOLD}_{c_j,j,s_j}$) along with the histogram partition to $B = 16$ bins. The figure also shows the interval I_j (which is 1-bin wide around ν), and $\overline{I}_j$ (the maximal intact interval in $\mathrm{FOLD}_{c_{j+1},j+1,s_{j+1}}$ that contains I_j). As should be $I_j \subset \overline{I}_j \subset \overline{I}_{j-1}$ and $\overline{I}_j$ aligns with the bin partition induced on $\overline{I}_{j-1}$. The figure also shows the ranges of values that are classified as internal and external.

We conclude the correctness proof with the following two lemma. Let

$$T_j = \sum_{\substack{i \text{ external in} \\ \mathrm{FOLD}_{c_j,j,s_j}}} w(i)|\mathrm{F}_{c_j,j,s_j}(f_i) - \mathrm{F}_{c_j,j,s_j}(\nu)|^\omega$$

be the non-discretized contribution during iteration j.

Lemma 5.2. *The total contribution made to M during iteration j is a $(1+\beta)(1 \pm \frac{2^{\rho+2}}{B})^\omega$-approximation of the quantity T_j, where $(1+\beta)$ is our error in the decaying sum estimates.*

Proof. Observe that only bins of external items and all bins with external items contribute to our estimate among the bins of $\mathrm{FOLD}_{c_j,j,s_j}$. Thus, the error stems from two reasons, first we have the weight of each such bin only up to $(1+\beta)$ accuracy and second for item in a bin we sum up not the exact difference $|\mathrm{F}_{c_j,j,s_j}(f_i) - \mathrm{F}_{c_j,j,s_j}(\nu)|$ but this difference where $\mathrm{F}_{c_j,j,s_j}(f_i)$ is rounded to a bin boundary. The first component of the error clearly contribute a factor of $1+\beta$ to our overall error.

We next bound the error introduced by rounding to bins. For $j = \rho^{-1}\log_2 R/B$ the range of $\mathrm{FOLD}_{c_j,j,s_j}$ has B values and the histogram captures exact values, thus the contribution is precise. Otherwise, since item i is external, $i \notin I_j$ and therefore $\mathrm{FOLD}_{c_j,j,s_j}(f_i) - \mathrm{FOLD}_{c_j,j,s_j}(\nu) \geq R/2^{\rho(j+1)+2}$; Rounding to bin boundaries gives an additive error term of $(R/2^{\rho j})/B$ in our knowledge of $\mathrm{FOLD}_{c_j,j,s_j}(f_i)$. Thus, the relative error we get in $|\mathrm{F}_{c_j,j,s_j}(f_i) - \mathrm{F}_{c_j,j,s_j}(\nu)|^\omega$ is bounded by $(1 \pm 2^{\rho+2}/B)^\omega$. □

Lemma 5.3. *$\sum_j T_j$ is an $\left(1 + 2^{-\omega\rho+2\omega}/(1-2^{-\rho\omega})\right)$-approximation of $\sum_i w(i)|f_i - \nu|^\omega$.*

Proof. We now consider the contribution of each item i to $\sum_j T_j$ (that is, the sum, over j where i is external in $\mathrm{FOLD}_{c_j,j,s_j}$, of the contribution of i to T_j.) For each item i, we define $J(i) \in \{0, \ldots, \rho^{-1}\log_2(R/B)\}$ be such that $f_i \in \overline{I}_{J(i)-1} \setminus \overline{I}_{J(i)}$. Recall that an item i is external in the jth iteration if and only if i is included in $\mathrm{FOLD}_{c_j,j,s_j}$ and

$$\mathrm{F}_{c_j,j,s_j}(f_i) \in \mathrm{F}_{c_j,j,s_j}(\overline{I}_{j-1} \setminus \overline{I}_j) \ .$$

In particular an item i is external in the fold $\mathrm{FOLD}_{c_{J(i)},J(i),s_{J(i)}}$. Moreover, since the intervals $\overline{I}_j$ are nested, iteration $J(i)$ is the first iteration in which the item i is external. Note also that the interval between f_i and ν is intact in iteration $J(i)$ and is not intact in subsequent iterations.

Each item i contributes in several iterations. The first iteration it contributes in is iteration $J(i)$. Since the interval between f_i and ν is intact in iteration $J(i)$, the contribution of i in iteration $J(i)$ is exactly $w(i)|(f_i - \nu)|^\omega$. We next argue that its contributions in subsequent iterations are at most some constant fraction of $w(i)|(f_i - \nu)|^\omega$.

It follows from the definition of the intervals I_j that the contribution of i at iteration $J(i)$ is at least $w(i)(R/2^{\rho(J(i)+1)+2})^\omega$. Lemma 5.1 states that external items in an iteration are excluded in the next iteration. Thus, an item i does not contribute in iteration $J(i)+1$. In every iteration $j \geq J(i)+2$ the contribution of i is at most $w(i)(R/2^{\rho j})^\omega$ (since the size of the range of $\mathrm{FOLD}(*, j, *)$ is $R/2^{\rho j}$). Since the upper bound on the contribution of i in each iteration $j \geq J(i)+2$ decreases by a factor of $2^{-\rho\omega}$ we obtain that sum of the contribution of i in all these iterations together is at most

$$w(i)(R/2^{\rho(J(i)+2)})^{\omega}(1+2^{-\rho\omega}+(2^{-\rho\omega})^2+\cdots)$$
$$\leq w(i)(R/2^{\rho(J(i)+2)})^{\omega}/(1-2^{-\rho\omega}) \ .$$

Therefore the relative error contributed by the contribution of i in iterations $j \geq J(i)+2$ is at most

$$\frac{(R/2^{\rho(J(i)+2)})^{\omega}/(1-2^{-\rho\omega})}{(R/2^{\rho(J(i)+1)+2})^{\omega}} = 2^{-\omega\rho+2\omega}/(1-2^{-\rho\omega}) \ .$$

□

Combining the two lemmas, the total approximation factor is

$$(1+\beta)(1+2^{\rho+2}/B)^{\omega}(1+2^{(2-\rho)\omega}/(1-2^{-\rho\omega})) \ .$$

Assume we are interested in summaries that are good for a certain ϵ and $\omega \in [\omega_a, \omega_b]$. The second part of the approximation factor (contributed by Lemma 5.3) is decreasing with ω, thus we shall choose $\rho > 2\omega_a^{-1}$ large enough such that $2^{1+(2-\rho)\omega_a} \leq \epsilon$. The first part of the approximation factor (contributed by Lemma 5.2) is increasing with ω. By choosing a sufficeintly large B we can have $(1+2^{\rho+2}/B)^{\omega_b} \leq (1+\epsilon)$.

5.1 When values are unrestricted

Our presentation assumed that an upper bound R on the maximum value M is known to all nodes. This assumption can be dropped by using the sum S of all values in the system. Then we can use $R = S$ (The sum S is at most nM, and thus $\log S \leq \log n + \log M$.) Alternatively, we can perform $\log(Mn/S)$ count computations, where the ith computation counts the number of values that are larger than $(S/n)2^i$. As a result, we obtain an estimate on M within a factor of 2.

Another natural question is whether we can remove the dependence on R (and allow for exponentially large range of values.) A simple construction (that mimics one given for spatially-decaying sums [5]) shows that the dependence is inherent.

6 Central moments

We now show how approximate values of $\mathsf{M}^{\omega}_{\mu,g}$ for even integral $\omega \geq 2$ can be retrieved from the summaries. For brevity, since clear from context, we omit the decay function g from the subscripts.

The challenge in approximating $\mathsf{A}^{\omega}_{\mu}$ is that μ is not known to us. It can be approximated within a relative error using the sum and count aggregates, but a relative-error estimate on μ is not sufficient for obtaining a relative-error estimate on $\mathsf{A}^{\omega}_{\mu}$.

We start with some definitions and lemmas.

One-sided power sum about ν is a weighted sum of all values that are larger (or smaller) than ν. That is, $\mathsf{A}^{\omega,+}_{\nu} = \sum_{i|f_i>\nu} w(i)|f_i-\nu|^{\omega}$ or $\mathsf{A}^{\omega,-}_{\nu} = \sum_{i|f_i\leq\nu} w(i)|f_i-\nu|^{\omega}$. Absolute power sums and pure power sums with even ω can be expressed as the sum of the two one-sided sums $\mathsf{A}^{\omega}_{\nu} = \mathsf{A}^{\omega,+}_{\nu} + \mathsf{A}^{\omega,-}_{\nu}$ whereas the pure power sums with od ω are the difference $\mathsf{M}^{\omega}_{\nu} = \mathsf{A}^{\omega,+}_{\nu} - \mathsf{A}^{\omega,-}_{\nu}$.

Lemma 6.1. *A slight modification of algorithm Power-Sum allows us to obtain approximate values for each one-sided sum within an* additive *error term of* $\epsilon\mathsf{A}^{\omega}_{\nu}$.

Proof. The modification amounts to simply considering a subset of the bins that cover only the part of $\mathrm{F}_{c_j,j,s_j}(\overline{I}_{j-1}\setminus\overline{I}_j)$ that is larger (for $f_i > \nu$) or smaller (for $f_i \leq \nu$) than $\mathrm{F}_{c_j,j,s_j}(\overline{I}_j)$. Observe that this is the same additive error we obtained when approximating the sum $\mathsf{A}^{\omega}_{\nu} = \sum_i w(i)|f_i-\nu|^{\omega}$, only that it does not necessarily translate into a small relative error in the one-sided case. □

Corollary 6.1. *For each* ν *and* $\omega \in [\omega_a, \omega_b]$, $\mathsf{M}^{\omega}_{\nu}$ *can be estimated from the summaries to within an* additive term *of* $\epsilon\mathsf{A}^{\omega}_{\nu}$.

Proof. $\mathsf{M}^{\omega}_{\nu}$ is a sum or difference of two one-sided sums. Each one-sided sum can be estimated to within an additive term of $(\epsilon/2)\mathsf{A}^{\omega}_{\nu}$. Thus, their sum or difference can be estimated to within an additive term of $\epsilon\mathsf{A}^{\omega}_{\nu}$. □

An important ingredient we need is obtaining a value ν, such that the absolute ω-power sum about ν is within some constant factor of the respective absolute central power sum. That is, $\mathsf{A}^{\omega}_{\nu} = O(\mathsf{A}^{\omega}_{\mu})$. It is easy to see that an approximate (with relative error) mean does not necessarily possess this property, but fortunately, (as proved in the following lemma) an approximate median will do. Folklore knowledge is that a random value (drawn according to the weights $w(i)$) has a constant probability of being an approximate median, and this probability can be arbitrarily increased by selecting the median of a constant number of random samples. An efficient algorithm for obtaining such spatially-decaying random samples is given in [5].

Lemma 6.2. *Let* m *be such that* $\sum_{i|f_i\leq m} w(i) \geq cW$ *and* $\sum_{i|f_i\geq m} w(i) \geq cW$ *that is, the weight of items with value that is at most* m *is at least* cW*, and the weight of items with value that is at least* m *is at least* cW*.) Then*

$$\frac{\mathsf{A}^{\omega}_{m}}{\mathsf{A}^{\omega}_{\mu}} \leq 2^{\omega}(1-c)/c \ .$$

Proof. Assume wlog that $\mu > m$. Consider now items and their contributions to the power sums A_m^ω and A_μ^ω. All the values that are larger than $m + 2(\mu - m)$ have the property that their contribution to A_m^ω is at most 2^ω times their contribution to A_μ^ω. We next consider items with value at most m. Since they are closer to m than to μ, their contribution to A_m^ω is smaller than their contribution to A_μ^ω. The total contribution of these items to A_μ^ω is at least $cW|\mu - m|^\omega$. Thus $\mathsf{A}_\mu^\omega \geq cW|\mu - m|^\omega$. We next consider values in the interval $(m, m + 2(\mu - m)]$. Since the total weight of items with values at most m is at least cW, the weight of items with values in the interval $(m, m+2(\mu-m)]$ is at most $(1-c)W$. So the contribution of the items in $(m, m+2(\mu-m)]$ to A_m^ω is at most $2^\omega(1-c)W|\mu-m|^\omega$. It follows that $\mathsf{A}_m^\omega/\mathsf{A}_\mu^\omega \leq \max\{2^\omega, 2^\omega(1-c)/c\} \leq 2^\omega(1-c)/c$ (note that we always have $c \leq 1/2$). □

We will need the following inequalities:

Lemma 6.3. *For any set of integers $i_j \geq 1$ $(j = 1, \ldots n)$ we have*

$$\Pi_j \mathsf{A}_\nu^{i_j} \leq W^{n-1}\mathsf{A}_\nu^{\sum_j i_j}$$

Proof. We apply *Chebyshev's integral inequality* ([9] page 1092) which states that for any set of nonnegative, integrable, monotone (all non-decreasing or all non-increasing) functions $h_1(x), \ldots h_n(x)$ we have

$$\Pi_{j=1}^n \int_a^b h_j(x)dx \leq (b-a)^{n-1}\int_a^b (\Pi_{j=1}^n h_j(x))dx \ .$$

The claim will follow by applying this inequality with the following parameters: Let $W = \sum w(i)$ as defined earlier and assume wlog that $|f_i - \nu|$ are ordered by magnitude. Let the step function $h_j(x)$ be defined on the interval $[0, W]$ as follows, $h_j(x) = |f_i - \nu|^{i_j}$ for $x \in \left[\sum_{k<i} w(k), \sum_{k \leq i} w(k)\right)$. Note that it follows from our ordering assumption that the functions $h_j()$ are non-decreasing, as needed for the statement of the inequality. □

It is well known that (using the Binomial transform) each central moment can be expressed as a sum of powers over raw moments about any value (see [10], page 146):

$$\frac{\mathsf{M}_\mu^\omega}{W} = \sum_{k=0}^{\omega} \left(\begin{array}{c} \omega \\ k \end{array} \right) (-1)^{\omega-k}(\frac{\mathsf{M}_\nu^1}{W})^{\omega-k}\frac{\mathsf{M}_\nu^k}{W} \ . \tag{6.3}$$

In terms of power sums we obtain

$$\begin{aligned} \mathsf{M}_\mu^\omega \quad &= \quad W^{-\omega+1}(-1)^\omega(1-\omega)(\mathsf{M}_\nu^1)^\omega \\ &\quad + \sum_{k=2}^{\omega} W^{-(\omega-k)} \left(\begin{array}{c} \omega \\ k \end{array} \right) (-1)^{\omega-k}(\mathsf{M}_\nu^1)^{\omega-k}\mathsf{M}_\nu^k \ . \end{aligned} \tag{6.4}$$

In particular,

$$\begin{aligned} \mathsf{M}_\mu^2 &= -(\mathsf{M}_\nu^1)^2/W + \mathsf{M}_\nu^2 \\ \mathsf{M}_\mu^4 &= \mathsf{M}_\nu^4 - 3(\mathsf{M}_\nu^1)^4/W^3 - 4\mathsf{M}_\nu^3\mathsf{M}_\nu^1/W + 6\mathsf{M}_\nu^2(\mathsf{M}_\nu^1)^2/W^2 \ . \end{aligned}$$

We let ν be an approximate median m as in Lemma 6.2. We estimate the central moment through the polynomial sum of raw moments about $\nu = m$ (Equ. 6.4), by plugging in, for each M_m^i, our estimated quantity $\widehat{\mathsf{M}_m^i}$ and for W, the $(1 \pm \epsilon)$-approximate $\hat{W}$.

Lemma 6.4. *The additive error we obtain in our estimate is $O(\epsilon\mathsf{A}_\mu^\omega)$.*

Proof. The polynomial sum (Equ. 6.4) has a constant number of terms, where each term in the sum has the form of a constant times $W^{-(\omega-k)}\mathsf{M}_m^k(\mathsf{M}_m^1)^{\omega-k}$ (for $\omega \geq k \geq 1$), it thus suffices to bound by $O(\epsilon\mathsf{A}_\mu^\omega)$ the error introduced by the approximation of each such term. In fact, using Lemma 6.2, it suffices to bound the error of each term by $O(\epsilon\mathsf{A}_m^\omega)$.

Consider the error in a single term.

$$\left|\hat{W}^{-(\omega-k)}\widehat{\mathsf{M}_m^k}(\widehat{\mathsf{M}_m^1})^{\omega-k} - W^{-(\omega-k)}\mathsf{M}_m^k(\mathsf{M}_m^1)^{\omega-k}\right| \tag{6.5}$$

$$\leq (1+2\epsilon\omega)W^{-(\omega-k)}\left|\widehat{\mathsf{M}_m^k}(\widehat{\mathsf{M}_m^1})^{\omega-k} - \mathsf{M}_m^k(\mathsf{M}_m^1)^{\omega-k}\right| \ .$$

(using $|\hat{W} - W| \leq \epsilon W$ and assuming $\epsilon \ll \omega^{-1}$). We bound the difference $\left|\widehat{\mathsf{M}_m^k}(\widehat{\mathsf{M}_m^1})^{\omega-k} - \mathsf{M}_m^k(\mathsf{M}_m^1)^{\omega-k}\right|$. Denote $\Delta_m^k \equiv \widehat{\mathsf{M}_m^k} - \mathsf{M}_m^k$. The difference can be rewritten as $(\mathsf{M}_m^k + \Delta_m^k)(\mathsf{M}_m^1 + \Delta_m^1)^{\omega-k} - \mathsf{M}_m^k(\mathsf{M}_m^1)^{\omega-k}$. If we expand this expression the term $\mathsf{M}_m^k(\mathsf{M}_m^1)^{\omega-k}$ cancels out and we obtain a polynomial $P(\mathsf{M}_m^k, \mathsf{M}_m^1, \Delta_m^k, \Delta_m^1)$ that consists of a positive linear combination of products of M_m^k, M_m^1, Δ_m^k, and Δ_m^1. Recall now that $|\mathsf{M}_m^i| \leq \mathsf{A}_m^i$ and that $|\Delta_m^i| = |\widehat{\mathsf{M}_m^i} - \mathsf{M}_m^i| \leq \epsilon\mathsf{A}_m^i$ (for all i) (from Corollary 6.1). Therefore, if we replace each appearance of M_m^i by A_m^i and each Δ_m^i by $\epsilon\mathsf{A}_m^i$ we can only increase the absolute value of this polynomial, thus

$$P(\mathsf{A}_m^k, \mathsf{A}_m^1, \epsilon\mathsf{A}_m^k, \epsilon\mathsf{A}_m^1) \geq |P(\mathsf{M}_m^k, \mathsf{M}_m^1, \Delta_m^k, \Delta_m^1)| \ .$$

We next observe that (from the definition of the multivariate polynomial $P()$)

$$\begin{aligned} &P(\mathsf{A}_m^k, \mathsf{A}_m^1, \epsilon\mathsf{A}_m^k, \epsilon\mathsf{A}_m^1) \\ &\quad = \quad (\mathsf{A}_m^k + \epsilon\mathsf{A}_m^k)(\mathsf{A}_m^1 + \epsilon\mathsf{A}_m^1)^{\omega-k} - \mathsf{A}_m^k(\mathsf{A}_m^1)^{\omega-k} \\ &\quad = \quad ((1+\epsilon)^{\omega-k+1} - 1)\mathsf{A}_m^k(\mathsf{A}_m^1)^{\omega-k} \ . \end{aligned}$$

Using Lemma 6.3, we obtain that the difference is at most

$$\leq ((1+\epsilon)^\omega - 1)\mathsf{A}_m^\omega W^{\omega-k} \ .$$

To summarize, we have that the error (Expression 6.5) is bounded by $(1+2\omega\epsilon)((1+\epsilon)^\omega - 1)\mathsf{A}_m^\omega \leq 2\omega\epsilon\mathsf{A}_m^\omega$ (since $\epsilon \ll \omega^{-1}$) □

7 Extensions

We point out some extensions of our results.

7.1 Beyond power sums Using our $(1+\epsilon)$-approximate absolute power sums we can also approximate any fixed expression that constitutes of powers, products, ratios, and positive linear combination of $\mathsf{A}_{\nu_j}^{\omega_j}$. We next discuss approximating the sum $\sum_i w(i)h(f_i - a)$ for more general functions $h()$. By examining where properties of the $h()$ entered the analysis, we obtain that our basic technique, with polylogarithmic size summaries, can be extended to any function $h()$ that is increasing and bounded below and above by polynomials (or by slightly super and sub-polynomials functions).

7.2 Higher dimensions We now consider the case when the item values $f_i = (f_{i1}, \ldots, f_{id})$ are vectors in R^d. The query points are $\nu = (\nu_1, \ldots, \nu_d) \in R^d$. The aggregate functions are defined by constants $p_1, \ldots, p_d > 0$ and $p > 0$ (from a fixed range) and are

$$||f_i - \nu|| = \left(\sum_{j=1}^{d} |f_{ij} - \nu_j|^{p_j} \right)^{p^{-1}} .$$

(in particular, this generalizes L_p norms). We are interested in summaries that would yield approximate values of $\sum_i w(i)||f_i - \nu||$).[8] We sketch the extension of our $d = 1$ construction to $d > 1$. The size of the summaries is polylogarithmic in the number of items but the dependence on the dimension d is exponential. Each of our d-dimensional foldings maps the domain into a smaller d-dimensional *range cube*. The *widths* (edge-lengths) of these range cubes are exponentially decreasing. For each width we use $(2S)^d$ different foldings. For each possible (out of 2^d) selection of different zero or half tile-width shifts we consider a partition into sub-cubes according to the width. For each partition, we derive S^d foldings, where each folding includes a subset of the sub-cubes that are spaced S sub-cubes apart. Each folding then maps all included sub-cubes into a range cube of the respective width. The range cube of each folding is then partitioned uniformly into B^d sub-cubes (bins), and each bin corresponds to a predicate. We thus use $O((2BS)^d \log R)$ predicates.

7.3 k-medians We next consider summaries that for any set of points $\nu_1, \ldots, \nu_k$, obtain an approximate value of $F(\nu_1, \ldots, \nu_k) = \sum_i w(i) \min_{j=1}^{k} ||f_i - \nu_j||$. This is relevant for computing the k-median defined as $\arg\min_{\nu_1,\ldots,\nu_k} F(\nu_1, \ldots, \nu_k)$. We can extend our techniques to perform this task with polynomial dependence on k. Due to space limitation we defer the details to the full version. The k-median problem on data streams had been considered by Charikar, O'Callaghan, Panigraphy [3] who gave a polylogarithmic storage algorithm with linear dependence on k for any metric space. The problem on sliding windows was considered by [2] and left open the existence of polylogarithmic space algorithms. Our result is not directly comparable: On one hand, spatially-decaying aggregation generalizes sliding windows on data streams and non-decaying data streams. On the other hand, we solve a more restricted problem and address only fixed values of d.

[8] For each such "norm" it is interesting to consider the median $\arg\min_\nu \sum_i w(i)||f_i - \nu||$. Since our summaries can obtain an estimate for any ν, they can be used to estimate this median.

References

[1] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and issues in data stream systems. In *Proc. of the 2002 ACM Symp. on Principles of Database Systems (PODS 2002).* ACM, 2002.

[2] B. Babcock, M. Datar, R. Motwani, and L. O'Callaghan. Maintaining variance and k-medians in data stream windows. In *Proc. of the 2002 ACM Symp. on Principles of Database Systems (PODS 2003).* ACM, 2003.

[3] M. Charikar, L. O'Callaghan, and R. Panigraphy. Better streaming algorithms for clustering problems. In *Proc. 35th Annual ACM Symposium on Theory of Computing*, pages 30–39. ACM, 2003.

[4] E. Cohen. Size-estimation framework with applications to transitive closure and reachability. *J. Comput. System Sci.*, 55:441–453, 1997.

[5] E. Cohen and H. Kaplan. Spatially-decaying aggregation over a network: model and algorithms. In preparation, 2003.

[6] E. Cohen and M. Strauss. Maintaining time-decaying stream aggregates. In *Proc. of the 2003 ACM Symp. on Principles of Database Systems (PODS 2003).* ACM, 2003.

[7] M. Datar, A. Gionis, P. Indyk, and R. Motwani. Maintaining stream statistics over sliding windows. *SIAM J. Comput.*, 31(6):1794–1813, 2002.

[8] P. B. Gibbons and S. Tirthapura. Distributed streams algorithms for sliding windows. In *Proc. of the 14th Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 63–72. ACM, 2002.

[9] I. A. Gradshteyn and I. M. Ryzhik. *Tables of Integrals, Series, and products.* Academic Press, San Diego, CA, 6 edition, 2000.

[10] A. Papoulis. *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill Book Company, New York, second edition, 1984.