Average Distance Queries through Weighted Samples in Graphs and Metric Spaces: High Scalability with Tight Statistical Guarantees

Shiri Chechik[1], Edith Cohen[1,2], and Haim Kaplan[1]

1 **Tel Aviv University**
Israel
{schechik,haimk}@cs.tau.ac.il
2 **Google Research**
Mountain View, CA, USA
edith@cohenwang.com

Abstract

The average distance from a node to all other nodes in a graph, or from a query point in a metric space to a set of points, is a fundamental quantity in data analysis. The inverse of the average distance, known as the (classic) closeness centrality of a node, is a popular importance measure in the study of social networks. We develop novel structural insights on the sparsifiability of the distance relation via weighted sampling. Based on that, we present highly practical algorithms with strong statistical guarantees for fundamental problems. We show that the average distance (and hence the centrality) for all nodes in a graph can be estimated using $O(\epsilon^{-2})$ single-source distance computations. For a set V of n points in a metric space, we show that after preprocessing which uses $O(n)$ distance computations we can compute a weighted sample $S \subset V$ of size $O(\epsilon^{-2})$ such that the average distance from any query point v to V can be estimated from the distances from v to S. Finally, we show that for a set of points V in a metric space, we can estimate the average pairwise distance using $O(n + \epsilon^{-2})$ distance computations. The estimate is based on a weighted sample of $O(\epsilon^{-2})$ pairs of points, which is computed using $O(n)$ distance computations. Our estimates are unbiased with normalized mean square error (NRMSE) of at most ϵ. Increasing the sample size by a $O(\log n)$ factor ensures that the probability that the relative error exceeds ϵ is polynomially small.

1998 ACM Subject Classification E.1 Data Structures, F.2.1 Numerical Algorithms and Problems, G.1.3 Numerical Linear Algebra, G.2.2 Graph Theory

Keywords and phrases Closeness Centrality, Average Distance, Metric Space, Weighted Sampling

Digital Object Identifier 10.4230/LIPIcs.APPROX-RANDOM.2015.659

1 Introduction

Measures of structural centrality based on shortest-paths distances, first studied by Bavelas [3], are classic tools in the analysis of social networks and other graph datasets. One natural measure of the importance of a node in a network is its *classic closeness centrality*, defined as the inverse of its average distance to all other nodes. This centrality measure, which is also termed *Bavelas closeness centrality* or the *Sabidussi Index* [13, 14, 24], was proposed by Bavelas [4], Beauchamp [5], and Sabidussi [20]. Formally, for a graph $G = (V, E)$ with $|V| = n$ nodes, the classic closeness centrality of $v \in V$ is

$$\mathrm{CC}(v) = \frac{n-1}{\sum_{u \in V} \mathrm{dist}(u, v)} \ , \tag{1}$$

18th Int'l Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX'15) / 19th Int'l Workshop on Randomization and Computation (RANDOM'15).
Editors: Naveen Garg, Klaus Jansen, Anup Rao, and José D. P. Rolim; pp. 659–679

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

where $\mathrm{dist}(u, v)$ is the length of a shortest path between v and u in G and n is the number of nodes. Intuitively, this measure of centrality reflects the ability of a node to send goods to all other nodes.

In metric spaces, the average distance of a point z to a set V of n points, $\sum_{x \in V} \mathrm{dist}(z, x)/n$, is a fundamental component in some clustering and classification tasks. For clustering, the quality of a cluster can be measured by the sum of distances from a centroid (usually 1-median or the mean in Euclidean data). Consequently, the (potential) relevance of a query point to the cluster can be estimated by relating its average distance to the cluster points to that of the center or more generally, to the distribution of the average distance of each cluster point to all others. This classification method has the advantages of being non-parametric (making no distribution assumptions on the data), similarly to the popular k nearest neighbors [10] (kNN) classification. Average distance based classification complements kNN, in that it targets settings where the outliers in the labeled points do carry information that should be incorporated in the classifier. A recent study [16] demonstrated that this is the case for some data sets in the UCI repository, where average distance based classification is much more accurate than kNN classification.

These notions of centrality and average distance had been extensively used in the analysis of social networks and metric data sets. We aim here to provide better tools to facilitate the computation of these measures on very large data sets. In particular, we present estimators with tight statistical guarantees whose computation is highly scalable.

We consider inputs that are either in the form of an undirected graph (with nonnegative edge weights) or a set of points in a metric space. In case of graphs, distance of the underlying metric correspond to lengths of shortest paths. Our results also extend to inputs specified as directed strongly connected graphs where the distance are the round trip distances [6]. We use a unified notation where V is the set of nodes if the input is a graph, or the set of points in a metric space. We denote $|V| = n$. We use graph terminology, and mention metric spaces only when there is a difference between the two applications. We find it convenient to work with the sum of distances

$$\mathrm{W}(v) = \sum_{u \in V} \mathrm{dist}(v, u) \ .$$

Average distance is then simply $\mathrm{W}(v)/n$ and centrality is $\textsc{cc}(v) = (n-1)/\mathrm{W}(v)$. Moreover, estimates $\hat{\mathrm{W}}(v)$ that are within a small relative error, that is $(1-\epsilon)\,\mathrm{W}(u) \leq \hat{\mathrm{W}}(u) \leq (1+\epsilon)\,\mathrm{W}(u)$, imply a small relative error on the average distance, by taking $\hat{\mathrm{W}}(v)/n$, and for centrality $\textsc{cc}(v)$, by taking $\hat{\textsc{cc}}(v) = (n-1)/\hat{\mathrm{W}}(v)$.

We list the fundamental computational problems related to these measures.

- *All-nodes sums*: Compute $\mathrm{W}(v)$ of all $v \in V$.
- *Point queries (metric space):* Preprocess a set of points V in a metric space, such that given a query point v (any point in the metric space, not necessarily $v \in V$), we can quickly compute $\mathrm{W}(v)$.
- *1-median*: Compute the node u of maximum centrality or equivalently, minimum $\mathrm{W}(u)$.
- *All-pairs sum*: Compute the sum of the distances between all pairs, that is $\textsc{aps}(V) \equiv \frac{1}{2}\sum_{v \in V} \mathrm{W}(v)$.

In metric spaces, we seek algorithms that compute distances for a small number of pairs of points. In graphs, a distance computation between a specific pair of nodes u, v seems to be computationally equivalent in the worst-case to computing all distances from a single source node (one of the nodes) to all other nodes. Therefore, we seek algorithms that perform a small number of single-source shortest paths (SSSP) computations. An SSSP computation in a graph can be performed using Dijkstra's algorithm in time that is nearly linear in

the number of edges [12]. To support parallel computation, it is also desirable to reduce dependencies between the distance or single-source distance computations.

The best known exact algorithms for the problems that we listed above do not scale well. To compute $\mathrm{W}(v)$ for all v, all-pairs sum, and 1-median, we need to compute the distances between all pairs of nodes, which in graphs is equivalent to an all-pairs shortest paths (APSP) computation. To answer point queries, we need to compute the distances from the query point to all points in V. In graphs, the hardness of some of these problems was formalized by the notion of subcubic equivalence [23]. Abboud et al [1] showed that exact 1-median is subcubic equivalent to APSP and therefore is unlikely to have a near linear time solution. We apply a similar technique and show (in Section 7) that the all-pairs sum problem is also subcubic equivalent to APSP. In general metric spaces, exact all pairs sum or 1-median clearly requires $\Omega(n^2)$ distance computations.[1]

Since exact computation does not scale to very large data sets, work in the area focused on approximations with small relative errors. We measure approximation quality by the normalized root mean square error (NRMSE), which is the square root of the expected (over randomization used in the algorithm) square difference between the estimate and the actual value, divided by the mean. When the estimator is unbiased (as with sample average), this is the ratio between the standard deviation and the mean, which is called the coefficient of variation (CV). Chebyshev's inequality implies that the probability that the estimator is within a relative error of η from its mean is at least $1-(CV)^2/(\eta)^2$. Therefore a CV of ϵ implies that the estimator is within a relative error of $\eta = c\epsilon$ from its mean with probability $\geq 1-1/c^2$.

The sampling based estimates that we consider are also well concentrated, meaning roughly that the probability of a larger error decreases exponentially. With concentration, by increasing the sample size by a factor of $O(\log n)$ we get that the probability that the relative error exceeds ϵ, for any one of polynomially many queries, is polynomially small. In particular, we can estimate the sum of the distances of the 1-median from all other nodes up to a relative error of ϵ with a polynomially small error probability.

Previous work

We review previous work on scalable approximation of 1-median, all-nodes sums, and all-pairs sum. These problems were studied in metric spaces and graphs. A natural approach to approximate the centrality of nodes is to take a uniform sample S of nodes, perform $|S|$ single source distance computations to determine all distances from every $v \in S$ to every $u \in V$, and then estimate $\mathrm{W}(v)$ by $\hat{\mathrm{W}}(v) = \frac{n}{|S|}\mathrm{W}_S(v)$, where $\mathrm{W}_S(v) = \sum_{a\in S}\mathrm{dist}(v,a)$ is the sum of the distances from v to the nodes of S. This approach was used by Indyk [18] to compute a $(1+\epsilon)$-approximate 1-median in a metric space using only $O(\epsilon^{-2}n)$ distance computations (See also [17] for a similar result with a weaker bound.). We discuss this uniform sampling approach in more detail in Section 6, where for completeness, we show how it can be applied to the all-nodes sums problem.

The sample average of a uniform sample was also used to estimate all-nodes centrality [11] (albeit with weaker, additive guarantees) and to experimentally identify the (approximate)

[1] Take a symmetric distance matrix with all entries in $(1-1/n, 1]$. To determine the 1-median we need to compute the exact sum of entries in each raw, that is, to exactly evaluate all entries in the raw. This is because an unread entry of 0 in any raw would determine the 1-median. Similarly, to compute the exact sum of distances we need to evaluate all entries. Deterministically, this amounts to $\binom{n}{2}$ distance computations.

top k centralities [19]. When the distance distribution is heavy-tailed, however, the sample average as an estimate of the true average can have a large relative error. This is because the sample may miss out on the few far nodes that dominate $\mathrm{W}(v)$.

Recently, Cohen et al [6] obtained ϵ NRMSE estimates for $\mathrm{W}(v)$ for any v, using single-source distance computations from each node in a uniform sample of ϵ^{-3} nodes. Estimates that are within a relative error of ϵ for all nodes were obtained using $\epsilon^{-3} \log n$ single-source computations. This approach applies in any metric space. The estimator for a point v is obtained by using the average of the distances from v to a uniform sample for nodes which are "close" to v and estimating distances to nodes "far" from v by their distance to the sampled node closest to v. The resulting estimate is biased, but obtains small relative errors using essentially the information of single-source distances from a uniform sample.

For the all-pairs sum problem in metric spaces, Indyk [17] showed that it can be estimated by scaling up the average of $\tilde{O}(n\epsilon^{-3.5})$ distances between pairs of points selected uniformly at random. The estimate has a relative error of at most ϵ with constant probability. Barhum, Goldreich, and Shraibman [2] improved Indyk's bound and showed that a uniform sample of $O(n\epsilon^{-2})$ distances suffices and also argued that this sample size is necessary (with uniform sampling). Barhum et al. also showed that in an Euclidean space a similar approximation can be obtained by projecting the points onto $O(1/\epsilon^2)$ random directions and averaging the distances between all pairwise projections. Goldreich and Ron [15] showed that in an unweighted graph $O(\epsilon^{-2}\sqrt{n})$ distances between random pairs of points suffice to estimate the sum of all pairwise distances, within a relative error of ϵ, with constant probability. They also showed that $O(\epsilon^{-2}\sqrt{n})$ distances from a fixed node s to random nodes v suffice to estimate $\mathrm{W}(v)$, within a relative error of ϵ, with constant probability. A difficulty with using this result, however, is that in graphs it is expensive to compute distances between random pairs of points in a scalable way: typically a single distance between a particular pair of nodes s and t is not easier to obtain than a complete single source shortest path tree from s.

Contributions and overview

Our design is based on computing a single *weighted* sample that provides estimates with statistical guarantees for all nodes/points. A sample of size $O(\epsilon^{-2})$ suffices to obtain estimates $\hat{\mathrm{W}}(z)$ with a CV of ϵ for any z. A sample of size $O(\epsilon^{-2} \log n)$ suffices for ensuring a relative error of at most ϵ for all nodes in a graph or for polynomially many queries in a metric space, with probability that is at least $1 - 1/poly(n)$.

The sampling algorithm is provided in Section 2. This algorithm computes a *coefficient* γ_v for each $v \in V$ such that $\sum_v \gamma_v = O(1)$. Then for a parameter k, we obtain sampling probabilities $p_u \equiv \min\{1, k\gamma_v\}$ for $u \in V$. Using the probabilities p_v, we can obtain a Poisson sample S of expected size $\sum_u p_u = O(k)$ or a VarOpt sample [8] that has exactly that size (rounded to an integer).

We present our estimators in Section 3. For each node u, the inverse probability estimator $\widehat{\mathrm{dist}}(z,u)$ is equal to $\mathrm{dist}(z,u)/p_u$ if u is sampled and is 0 otherwise. Our estimate of the sum $\mathrm{W}(z)$ is the sum of these estimates

$$\hat{\mathrm{W}}(z) = \sum_{u \in V} \widehat{\mathrm{dist}}(z,u) = \sum_{u \in S} \widehat{\mathrm{dist}}(z,u) = \sum_{u \in S} \frac{\mathrm{dist}(z,u)}{p_u} \ . \tag{2}$$

Since $p_u > 0$ for all u, the estimates $\widehat{\mathrm{dist}}(z,u)$ and hence the estimate $\hat{\mathrm{W}}(z)$ are unbiased.

We provide a detailed analysis in Section 4. We will show that our sampling probabilities provide the following guarantees. When choosing $k = O(\epsilon^{-2})$, $\hat{\mathrm{W}}(z)$ has CV ϵ. Moreover, the

estimates have good concentration, so using a larger sample size of $O(\epsilon^{-2}\log n)$ we obtain that the relative error is at most ϵ for all nodes $v \in V$ with probability at least $1 - 1/poly(n)$.

In order to obtain a sample with such guarantees for some particular node z, the sampling probability of a node v should be (roughly) proportional to its distance $\mathrm{dist}(z,v)$ from z. Such a Probability Proportional to Size (PPS) sample of size $k = \epsilon^{-2}$ uses coefficients $\gamma_v = \mathrm{dist}(v,z)/\mathrm{W}(z)$ and has CV of ϵ. We will work with *approximate PPS* coefficients, which we define as satisfying $\gamma_v \geq c\,\mathrm{dist}(v,z)/\mathrm{W}(z)$ for some constant c. With approximate PPS we obtain a CV of ϵ with a sample of size $O(\epsilon^{-2})$. It is far from clear apriori, however, that there is a single set of *universal PPS* coefficients which are simultaneously (approximate) PPS for all nodes and are of size $\sum_v \gamma_v = O(1)$. That is, a single sample of size $O(\epsilon^{-2})$, which is independent of n and of the dimension of the space, would work for all nodes.

Beyond establishing the existence of universal PPS coefficients, we are interested in obtaining them, and the sample itself, using a near-linear computation. The dominant component of the computation of the sampling coefficients is performing $O(1)$ single-source distance computations. Therefore, it requires $O(m\log n)$ time in graphs and $O(n)$ pairwise distance queries in a metric space. A universal PPS sample of any given size k can then be computed in a single pass over the coefficients vector $\boldsymbol{\gamma}$ ($O(n)$ computation). We represent the sample S as a collection $\{(u,p_u)\}$ of nodes/points and their respective sampling probabilities. We can then use our sample for estimation using (2).

When the input is a graph, we compute single-source distances from each node in S to all other nodes in order to estimate $\mathrm{W}(v)$ of all $v \in V$. This requires $O(|S|m\log n)$ time and $O(n)$ space.

▶ **Theorem 1.** *All-nodes sums ($\mathrm{W}(v)$ for all $v \in V$) can be estimated unbiasedly as follows:*

- *With CV ϵ, using $O(\epsilon^{-2})$ single source distance computations.*
- *When using $O(\epsilon^{-2}\log n)$ single source distance computations, the probability that the maximum relative error, over the n nodes, exceeds ϵ is polynomially small.*

In a metric space, we can estimate $\mathrm{W}(x)$ for an arbitrary query point x, which is not necessarily a member of V, by computing the distances $\mathrm{dist}(x,v)$ for all $v \in S$ and applying the estimator (2). Thus, point queries in a metric space require $O(n)$ distance computations for preprocessing and $O(\epsilon^{-2})$ distance computations per query.

▶ **Theorem 2.** *We can preprocess a set of points V in a metric space using $O(n)$ time and $O(n)$ distance computations ($O(1)$ single source distance computations) to generate a weighted sample S of a desired size k. From the sample, we can unbiasedly estimate $\hat{\mathrm{W}}(z)$ using the distances between z and the points in S with the following guarantees:*

- *When $k = O(\epsilon^{-2})$, for any point query z, $\hat{\mathrm{W}}(z)$ has CV at most ϵ.*
- *When $k = O(\epsilon^{-2}\log n)$, the probability that the relative error of $\hat{\mathrm{W}}(z)$ exceeds ϵ for any of polynomially many queries z is polynomially small.*

We can also estimate all-pairs sum, using either primitive of single-source distances (for graphs) or distance computations (metric spaces).

▶ **Theorem 3.** *All-pairs sum can be estimated unbiasedly with the following statistical guarantees:*

- *CV of at most ϵ, using $O(\epsilon^{-2})$ single-source distance computations. With a relative error that exceeds ϵ with a polynomially small probability, using $O(\epsilon^{-2}\log n)$ single-source distance computations.*
- *With CV of at most ϵ, using $O(n+\epsilon^{-2})$ distance computations. With a relative error that exceeds ϵ with polynomially small probability, using $O(n+\epsilon^{-2}\log n)$ distance computations.*

The proof details are provided in Section 5. The part of the claim that uses single-source distance computations is established by using the estimate $\widehat{\text{APS}}(V) = \frac{1}{2}\sum_{z\in V}\hat{\text{W}}(z)$. When the estimates have CV of at most ϵ, even if correlated, so does the estimate $\widehat{\text{APS}}(V)$.[2] For the high probability claim, we use $O(\log n)$ single-source computations to ensure we obtain universal PPS coefficients with high probability (details are provided later), which imply that each estimate $\hat{W}(z)$, and hence the sum is concentrated.

For the second part that uses distance computations, we consider an approximate PPS distribution that is with respect to $\text{dist}(u,v)$, that is, the probability of sampling the pair (u,v) is at least $c\,\text{dist}(u,v)/\text{APS}(V)$ for some constant c. We show that we can compactly represent this distribution as the outer product of two probability vectors of size n. Using this representation we can draw $O(\epsilon^{-2})$ pairs independently in linear time, which we use for estimating the average.

Compared to the all-nodes sums algorithms of [6], our result here improves the dependency in ϵ from ϵ^{-3} to ϵ^{-2} (which is likely to be optimal for a sampling based approach), provides an unbiased estimates, and also facilitates approximate average distance oracles with very small storage in metric spaces (the approach of [6] would require the oracle to store a histogram of distances from each of ϵ^{-3} sampled nodes). For the all-pairs sum problem in graphs, we obtain an algorithm that uses $O(\epsilon^{-2})$ single source distance computations, which improves over an algorithm that does $O(\epsilon^{-3})$ single source distance computations implied by [6]. For the all pairs sum problem in a metric space, we obtain a CV of ϵ using $O(n+\epsilon^{-2})$ distance computation rather than $O(n\epsilon^{-2})$ distance computations required by the algorithms in [2, 17].

While our analysis does not optimize constants, our algorithms are very simple and we expect them to be effective in applications.

2 Constructing the sample

We present Algorithm 1 that computes a set of sampling probabilities associated with the nodes of an input graph G. We use graph terminology but the algorithm applies both in graphs and in metric spaces. The input to the algorithm is a set S_0 of base nodes and a parameter k (we discuss how to choose S_0 and k below). The algorithm consists of the following stages. We first compute a sampling coefficient γ_v for each node v such that $\sum_v \gamma_v = O(1)$. Then we use the parameter k and compute the sampling probabilities $p_v = \min\{1, k\gamma_v\}$. Finally we use the probabilities p_v to draw a sample of expected size $O(k)$, by choosing v with probability p_v. We usually apply the algorithm once with a pre-specified k to obtain a sample, but there are applications (see discussion in Section 8.4) in which we want to choose the sample size adaptively using the same coefficients.

Running time and sample size

The running time of this algorithm on a metric space is dominated by $|S_0|n$ distance computations. On a graph, the running time is $|S_0|m\log n$, and is dominated by the $|S_0|$ single-source shortest-paths computations. The expected size of the final sample S is $\sum_v p_v \le k\sum_v \gamma_v = O(k)$.

[2] In general if random variables X and Y have CV ϵ then so does their sum: $\frac{\mathsf{Var}(X+Y)}{(E(X+Y))^2} = \frac{\mathsf{Var}(X)+\mathsf{Var}(Y)+2\,\mathsf{Cov}(X,Y)}{(E(X+Y))^2} \le \frac{\mathsf{Var}(X)+\mathsf{Var}(Y)+2\sqrt{\mathsf{Var}(X)\,\mathsf{Var}(Y)}}{(E(X+Y))^2} \le \frac{\epsilon^2(E(X))^2+\epsilon^2(E(Y))^2+2\epsilon^2E(X)E(Y)}{(E(X+Y))^2} \le \epsilon^2$.

Algorithm 1 Compute universal PPS coefficients and sample

Input: Undirected graph with vertex set V or a set of points V in a metric space, base nodes S_0, parameter k

Output: A universal PPS sample S

`// Compute sampling coefficients` γ_v

foreach *node* v **do**
  $\gamma_v \leftarrow 1/n$

foreach $u \in S_0$ **do**
  Compute shortest path distances $\mathrm{dist}(u,v)$ from u to all other nodes $v \in V$
  $W \leftarrow \sum_v \mathrm{dist}(u,v)$
  foreach *node* $v \in V$ **do**
    $\gamma_v \leftarrow \max\{\gamma_v, \frac{\mathrm{dist}(u,v)}{W}\}$

foreach *node* $v \in V$ **do** `// Compute sampling probabilities` p_v
  $p_v \leftarrow \min\{1, k\gamma_v\}$

$S \leftarrow \emptyset$ `// Initialize sample`

foreach $v \in V$ **do** `// Poisson sample according to` p_v
  if $rand() < p_v$ **then**
    $S \leftarrow S \cup \{(v, p_v)\}$

return S

Choosing the base set S_0

We will show that in order to obtain the property that each estimate $\hat{\mathrm{W}}(v)$ has CV $O(\epsilon)$, it suffices that the base set S_0 includes a uniform sample of ≥ 2 nodes and we need to choose $k = \epsilon^{-2}$. Note that the CV is computed over the randomization in the choice of nodes to S_0 and of the sample we choose using the computed coeffcients. We will also introduce a notion of a *well positioned* node, which we precisely define in the sequel. We will see that when S_0 includes such a node, we also have CV of $O(\epsilon)$ with $k = \epsilon^{-2}$. This time using only the randomization in the selection of the sample. Moreover, if we choose $k = \epsilon^{-2} \log n$ and ensure that S_0 contains a well-positioned node with probability at least $1 - 1/poly(n)$ then we obtain that the probability that the relative error exceeds ϵ is polynomially small. We will see that most nodes are well positioned, and therefore, it is relatively simple to identify such a node with high probability.

3 Estimation

3.1 Centrality values for all nodes in a graph

For graphs, we compute estimates $\hat{\mathrm{W}}(v)$ for all nodes $v \in V$ as in Algorithm 2. We initialize all estimates to 0, and perform a SSSP computation from each node in $u \in S$. When scanning node v, during such SSSP computation, we add $\mathrm{dist}(u,v)/p_u$ to the estimate $\hat{\mathrm{W}}(v)$. The algorithms runs in $O(|S|m \log n)$ time, dominated by the $|S|$ SSSP computations from each node in the sample S.

3.2 Point queries (metric space)

For a query point z (which is not necessarily a member of V), we compute the distance $\mathrm{dist}(z,x)$ for all $x \in S$, and apply (2). This takes $|S|$ distance computations for each query.

Algorithm 2 Compute estimates $\hat{\mathrm{W}}(v)$ for all nodes v in the graph

Input: Weighted graph G, a sample $S = \{(u, p_u)\}$
foreach $v \in V$ **do**
  $\hat{\mathrm{W}}(v) \leftarrow 0$
foreach $u \in S$ **do**
  Perform a single-source shortest-paths computation from u.
  foreach *scanned node* $v \in V$ **do**
    $\hat{\mathrm{W}}(v) \leftarrow \hat{\mathrm{W}}(v) + \mathrm{dist}(u, v)/p_u$
return $(v, \hat{\mathrm{W}}(v))$ *for* $v \in V$

4 Correctness

We first show (Section 4.1) show that when $k = \epsilon^{-2}$, and S_0 includes either a uniform sample of size at least 2 then each estimate $\hat{\mathrm{W}}(v)$ has CV of $O(\epsilon)$. We then define well-positioned nodes in Section 4.2 and show that if S_0 contains a well positioned node we and sample size is $k = \epsilon^{-2}$ then the CV is $O(\epsilon)$ (Section 4.3) and when $k = O(\epsilon^{-2} \log n)$, the probability that the relative error exceeds ϵ is polynomially small (Section 4.5).

In Section 4.4 we establish an interesting property of our sampling coefficients: They can not grow too much even if the base set S_0 is very large. Clearly, $\sum_v \gamma_v \leq 1 + |S_0|$, but we will show that it is $O(1)$ regardless of the size of S_0.

We start with some useful lemmas.

▶ **Lemma 4.** *Suppose that S_0 contains a node u. Consider a node z such that u is the $(qn)^{th}$ closest node to z. Then for all nodes v,*

$$\gamma_v \geq \frac{1-q}{4} \cdot \frac{\mathrm{dist}(z, v)}{\mathrm{W}(z)} \ . \tag{3}$$

Proof. From the specification of Algorithm 1, the sampling coefficients γ_v satisfy

$$\gamma_v \geq \max\left\{\frac{1}{n}, \frac{\mathrm{dist}(u, v)}{\mathrm{W}(u)}\right\} \ . \tag{4}$$

Let $Q = \mathrm{dist}(z, u)$. Consider a classification of the nodes $v \in V$ to "close" nodes and "far" nodes according to their distance from z:

$$\begin{aligned} L &= \{v \in V \mid \mathrm{dist}(z, v) \leq 2Q\} \\ H &= \{v \in V \mid \mathrm{dist}(z, v) > 2Q\} \ . \end{aligned}$$

Since $\gamma_v \geq 1/n$, for $v \in L$ we have

$$\gamma_v \geq \frac{1}{n} \geq \left(\frac{1-q}{2}\right)\left(\frac{2}{1-q}\right)\frac{1}{n} = \left(\frac{1-q}{2}\right)\left(\frac{2Q}{(1-q)Q}\right)\frac{1}{n} \geq \left(\frac{1-q}{2}\right)\frac{\mathrm{dist}(z, v)}{\mathrm{W}(z)}, \tag{5}$$

where the last inequality holds since for $v \in L$ we have $\mathrm{dist}(z, v) \leq 2Q$, and since $\mathrm{W}(z) \geq (1-q)\, nQ$ if u is the (qn)th closest node to z.

For all v, we have that $\mathrm{dist}(u, v) \geq \mathrm{dist}(z, v) - Q$ by the triangle inequality. We also have $\mathrm{W}(u) \leq \mathrm{W}(z) + nQ$. Substituting into (4) we get that for every v

$$\gamma_v \geq \frac{\mathrm{dist}(u, v)}{\mathrm{W}(u)} \geq \frac{\mathrm{dist}(z, v) - Q}{\mathrm{W}(z) + nQ} \ . \tag{6}$$

In particular, for $v \in H$, we have

$$\mathrm{dist}(z,v) - Q \geq \frac{1}{2}\,\mathrm{dist}(z,v)\ . \tag{7}$$

As already mentioned, we also have $\mathrm{W}(z) \geq (1-q)\,nQ$ and thus

$$nQ \leq \frac{\mathrm{W}(z)}{1-q}\ , \tag{8}$$

and

$$\mathrm{W}(z) + nQ \leq \mathrm{W}(z)\left(1 + \frac{1}{1-q}\right) = W(z)\left(\frac{2-q}{1-q}\right)\ . \tag{9}$$

Substituting (9) and (7) in (6), we obtain that for $v \in H$,

$$\gamma_v \geq \frac{\mathrm{dist}(z,v) - Q}{\mathrm{W}(z) + nQ} \geq \frac{1}{2}\left(\frac{1-q}{2-q}\right)\frac{\mathrm{dist}(z,v)}{\mathrm{W}(z)}\ . \tag{10}$$

The lemma now follows from (5) and (10). ◀

▶ **Lemma 5.** *Consider a set of sampling coefficients γ_v such that for a node z, for all v and for some $c > 0$, $\gamma_v \geq c\frac{\mathrm{dist}(z,v)}{\mathrm{W}(z)}$. Let S be a sample obtained with probabilities $p_v = \min\{1, k\gamma_v\}$ (as in Algorithm 1), and let $\hat{\mathrm{W}}(z)$ be the inverse probability estimator as in* (2)*. Then*

$$\mathsf{Var}[\hat{\mathrm{W}}(z)] \leq \frac{\mathrm{W}(z)^2}{k \cdot c}\ . \tag{11}$$

Proof. The variance of our estimator is

$$\begin{aligned}\mathsf{Var}[\hat{\mathrm{W}}(z)] &= \sum_v \left[p_v\left(\frac{\mathrm{dist}(z,v)}{p_v} - \mathrm{dist}(z,v)\right)^2 + (1-p_v)\,\mathrm{dist}(z,v)^2\right] \\ &= \sum_v \left(\frac{1}{p_v} - 1\right)\mathrm{dist}(z,v)^2\ .\end{aligned} \tag{12}$$

Note that nodes v for which $p_v = 1$ contribute 0 to the variance. For the other nodes we use the lower bound $p_v \geq ck\frac{\mathrm{dist}(z,v)}{\mathrm{W}(z)}$.

$$\begin{aligned}\sum_{v\in V}\left(\frac{1}{p_v} - 1\right)\mathrm{dist}(z,v)^2 &= \sum_{v\in V\,|\,p_v<1}\left(\frac{1}{p_v} - 1\right)\mathrm{dist}(z,v)^2 \\ &\leq \frac{\mathrm{W}(z)}{k\cdot c}\sum_{v\in V}\mathrm{dist}(z,v) \\ &\leq \frac{\mathrm{W}(z)^2}{k\cdot c}\ .\end{aligned}$$

◀

4.1 Base set containing a uniform sample

We now consider a situation where S_0 includes a uniform sample of nodes, and consider the corresponding expected approximation quality:

▶ **Lemma 6.** *Suppose that S_0 contains a uniform random sample of b nodes. Then for any node z,*

$$\mathsf{Var}[\hat{\mathrm{W}}(z)] \leq \frac{\mathrm{W}(z)^2}{k}\frac{4b}{b-1}\ . \tag{13}$$

Proof. We apply Lemma 5 with the bound on the coefficients as in Lemma 4 with u being the closest node to z in S_0. Assume that u is the xth closest node to z. By Lemma 4 and Lemma 5 we have

$$\mathsf{Var}[\hat{\mathrm{W}}(z) \mid x] \leq \frac{\mathrm{W}(z)^2}{k} \frac{4}{1-x/n} \ . \tag{14}$$

Observe that x is a random variable which is the rank (= position in the sorted order of the nodes by distance from z) of the closest node to z in a uniform sample of size b. In particular x take values $\in \{1, 2, \ldots, n-b+1\}$ ($x = 1$ iff $u = z$). We have that the probability of rank x is

$$b\left(\frac{1}{n}\right)\left(\frac{n-x}{n-1}\right)\left(\frac{n-x-1}{n-2}\right)\cdots\left(\frac{n-x-b+2}{n-b+1}\right) \leq b\left(1-\frac{x}{n}\right)^{b-1} \ .$$

(We choose the random subset of S_0 of b nodes without replacement, we split into b events according to the step in which the node of rank x is chosen. Other items should be chosen from the $n-x$ nodes of rank larger than x.) The variance $\mathsf{Var}[\hat{\mathrm{W}}(z)]$ is the expectation, over $x \in \{1, 2, \ldots, n-b+1\}$, of $\mathsf{Var}[\hat{\mathrm{W}}(z) \mid x]$. So from (14), we get

$$\begin{aligned}
\mathsf{Var}[\hat{\mathrm{W}}(z)] &\leq \sum_{x=1}^{n-b+1} b\left(1-\frac{x}{n}\right)^{b-1}\left(\frac{\mathrm{W}(z)^2}{k}\frac{4}{(1-x/n)}\right) \\
&\leq \frac{\mathrm{W}(z)^2}{k} 4b \sum_{x=1}^{n-b+1}\left(1-\frac{x}{n}\right)^{b-2} \\
&\leq \frac{\mathrm{W}(z)^2}{k} 4b \int_0^1 (1-y)^{b-2} dy \\
&= \frac{\mathrm{W}(z)^2}{k}\frac{4b}{b-1} \ .
\end{aligned}$$

◀

It follows from Lemma 6 that if we choose $b \geq 2$ nodes uniformly into S_0 and $k = \epsilon^{-2}$, then for any node z, our estimator has $\mathsf{Var}[\hat{\mathrm{W}}(z)] = O(\epsilon^2\, \mathrm{W}(z)^2)$. This concludes the proof of the per-node (per-point) $O(\epsilon)$ bound on the CV of the estimator in the first part of Theorems 1 and 2 for a sample of size $O(\epsilon^{-2})$.

4.2 Well-positioned nodes

We provide a precise definition of a *well positioned* node. Let the *median distance* of a node u, denote by $m(u)$, be the distance between u and the $\lceil 1 + n/2 \rceil$ closest node to u in V. Let $\textsc{MinMed} = \min_{v \in V} m(v)$ be the minimum median distance of any node $v \in V$. In a metric space, we can define $m(u)$ for any point u in the space (also for $u \notin V$), and accordingly, define MinMed as the minimum $m(u)$ over all points u in the metric space.

We say that a node u is *well positioned* if $m(u) \leq 2\,\textsc{MinMed}$, that is, $m(u)$, the median distance of u is within a factor of 2 of the minimum median distance. We now show that most nodes are well positioned.

▶ **Lemma 7.** *Let v be such that is $m(v) = \textsc{MinMed}$. Then all $\lceil 1 + n/2 \rceil$ nodes in V that are closest to v are well positioned.*

Proof. Let u be one of the $\lceil 1 + n/2 \rceil$ nodes closest to v. Then $\mathrm{dist}(u, v) \leq \textsc{MinMed}$ and a ball of radius $2\,\textsc{MinMed}$ around u contains all the $\lceil 1 + n/2 \rceil$ nodes closest to v. So $m(u) \leq 2\,\textsc{MinMed}$. ◀

We are interested in well positioned nodes because of the following property:

▶ **Lemma 8.** *If u is a well positioned node, then for every node z we have that* $\mathrm{dist}(z,u) \leq 3m(z)$.

Proof. For every two nodes u and z we have that $\mathrm{dist}(u,z) \leq m(u)+m(z)$ since there must be at least one node x that is both within distance $m(u)$ from u and within distance $m(z)$ from z, and by the triangle inequality $\mathrm{dist}(u,z) \leq \mathrm{dist}(u,x)+\mathrm{dist}(x,z)$. The lemma follows since if u is well positioned then $m(u) \leq 2m(z)$. ◀

As we shall see, this means that sampling probabilities proportional to the distances from a well positioned node u approximate sampling probabilities proportional to the distances from any other node z, for nodes whose distance from z is substantially larger than $m(z)$.

4.3 Base set with a well-positioned node

We now consider the case where S_0 contains a well-positioned node. We show that in this case the coefficients γ_v satisfy what we call a *universal PPS* property:

▶ **Lemma 9.** *Suppose that S_0 contains a well-positioned node u. Then for all nodes v,*

$$\gamma_v \geq \frac{1}{18}\max_z \frac{\mathrm{dist}(z,v)}{\mathrm{W}(z)} \ . \tag{15}$$

Proof. We show that for any node z, $\gamma_v \geq \frac{1}{18}\frac{\mathrm{dist}(z,v)}{\mathrm{W}(z)}$ using a variation of the proof of Lemma 4.

We partition the nodes into two sets. A set L which contains the nodes v such that $\mathrm{dist}(z,v) \leq 6m(z)$ and a set H which contains the remaining nodes. By the definition of $m(z)$ we have that $\mathrm{W}(z) \geq m(z)(\lfloor \frac{n}{2} \rfloor - 1) \geq m(z)\frac{n}{3}$ (for $n \geq 9$). We obtain that for all $v \in L$,

$$\frac{\mathrm{dist}(v,z)}{\mathrm{W}(z)} \leq \frac{6m(z)}{m(z)\frac{n}{3}} = \frac{18}{n} \ .$$

Therefore,

$$\gamma_v \geq \frac{1}{n} \geq \frac{1}{18}\frac{\mathrm{dist}(v,z)}{\mathrm{W}(z)} \ .$$

We next consider $v \in H$. Since u is well positioned, by Lemma 8 we have that $\mathrm{dist}(z,u) \leq 3m(z)$. From the triangle inequality, $\mathrm{dist}(u,v) \geq \mathrm{dist}(z,v) - \mathrm{dist}(z,u) \geq \mathrm{dist}(z,v) - 3m(z) \geq \mathrm{dist}(z,v)/2$. We also have $\mathrm{W}(u) \leq \mathrm{W}(z) + n\,\mathrm{dist}(z,u) \leq \mathrm{W}(z) + 3nm(z) \leq 9\,\mathrm{W}(z)$. Therefore

$$\gamma_v \geq \frac{\mathrm{dist}(u,v)}{\mathrm{W}(u)} \geq \frac{(\mathrm{dist}(z,v)/2)}{9\,\mathrm{W}(z)} = \frac{1}{18}\frac{\mathrm{dist}(z,v)}{\mathrm{W}(z)} \ .$$ ◀

As a corollary, applying Lemma 5, we obtain:

▶ **Corollary 10.** *If S_0 contains a well-positioned node, then for any node z,* $\mathsf{Var}[\hat{\mathrm{W}}(z)] \leq 18\frac{\mathrm{W}(z)^2}{k}$.

4.4 Upper bound on the sum of the coefficients

One consequence of Lemma 9 is that the coefficients γ_u cannot grow too much even if the base set S_0 includes all nodes.

▶ **Corollary 11.** *Let*

$$\overline{\gamma}_v \equiv \max_z \frac{\mathrm{dist}(z,v)}{\mathrm{W}(z)} \ .$$

Then

$$\sum_v \overline{\gamma}_v = O(1) \ .$$

Proof. Consider the case where S_0 consists of a single well positioned node. By the definition of γ_v we have that $\sum_v \gamma_v \leq 2$. By Lemma 15 we have $\gamma_v \geq \frac{1}{18} \max_z \frac{\mathrm{dist}(z,v)}{\mathrm{W}(z)}$. Therefore $\sum_v \overline{\gamma}_v \leq 18 \sum_v \gamma_v \leq 36$. ◀

4.5 High probability estimates

Lastly, we establish concentration of the estimates, which will conclude the proof of the very high probability claims in Theorem 1 and 2.

We need the following lemma:

▶ **Lemma 12.** *If our sampling coefficients are* approximate PPS *for a node z, that is, there is a constant c such that for all nodes v, $\gamma_v \geq c\frac{\mathrm{dist}(z,v)}{\mathrm{W}(z)}$, and we use $k = O(\epsilon^{-2}\log n)$, then*

$$\Pr\left[\frac{|\hat{\mathrm{W}}(z) - \mathrm{W}(z)|}{\mathrm{W}(z)} \geq \epsilon\right] = O(1/poly(n)) \ .$$

Proof. We apply the Chernoff-Hoeffding bound. Let $\tau = W(z)/(ck)$. We have

$$p_v \geq \min\{1, \mathrm{dist}(z,v)/\tau\} = \min\{1, ck\,\mathrm{dist}(z,v)/\,\mathrm{W}(z)\} \ . \tag{16}$$

The contribution of a node v to the estimate $\hat{\mathrm{W}}(z)$ is as follows. If $\mathrm{dist}(z,v) \geq \tau$, then the contribution is exactly $\mathrm{dist}(z,v)$. Otherwise, the contribution X_v of node v is $\mathrm{dist}(z,v)/p_v \leq \tau$ with probability p_v and 0 otherwise.

The contributions X_v of the nodes with $\mathrm{dist}(z,v) \leq \tau$ are thus independent random variables, each in the range $[0,\tau]$ with expectation $\mathrm{dist}(z,v)$. We complete the proof by applying the Chernoff-Hoeffding bound to bound the deviation of expectation of the sum of these random variables. We defer the details to the full version of the paper. ◀

We need the condition of Lemma 12 to hold for all nodes z with probability $1 - O(1/poly(n))$. Equivalently, we would like $\boldsymbol{\gamma}$ to be universal PPS with very high probability. If so, we apply a union bound to obtain that the estimates $\hat{\mathrm{W}}(z)$ for all nodes z have a relative error of at most ϵ with probability $1 - O(1/poly(n))$. The same argument applies to polynomially many queries in metric spaces.

It follows from Lemma 9 that we obtain the universal PPS property if S_0 includes a well positioned node. We would like this to happen with very high probability. We mention several ways to achieve this effect: (i) Since most nodes are well positioned (Lemma 7), taking a uniform random sample U of $O(\log n)$ nodes, and choosing the node $u = \arg\min_{u \in U} m(u)$ with minimum distance to its $\lceil n/2 + 1 \rceil$ closest node, means that we are guaranteed with probability $1 - 1/poly(n)$ that u is well positioned. This identification step involves $O(\log n)$ single-source distance computations. (ii) Alternatively, we can ensure that S_0 contains a

well positioned node (with a polynomially small error) by simply placing $O(\log n)$ uniformly selected nodes in S_0. The computation of the coefficients will then require $O(\log n)$ single-source distance computations. (iii) Lastly, if S_0 contains $O(\log n)$ uniformly selected nodes then we can apply a direct argument that with a polynomially small error for each node z, one of the $\lceil n/2+1 \rceil$ closest nodes to z is in S_0. This means we can apply Lemma 4 with $q \leq 0.5$ to obtain that with a polynomially small error, the sampling probabilities are approximate PPS for all nodes and thus universal PPS with a polynomially small error.

To establish the second part of Theorem 2 in metric spaces, we would like to identify a well positioned node with a polynomially small ($O(1/poly(n))$) error using only $O(n)$ distance computations, which is more efficiently than by using $O(\log n)$ single-source distance computations.

To do so, we first provide a slightly relaxed definition of well positioned node and show that it retains the important properties. We will then show that a "relaxed" well positioned node can be identified with very high probability using only $O(\log^2 n)$ distance computations. When we identify such a node, we can use it in the base set S_0. This means we can use $O(n)$ distance computations in total to compute coefficients $\boldsymbol{\gamma}$ which are universal PPS with a polynomially small error. We then use $O(n)$ time to compute a sample of size $k = O(\epsilon^{-2} \log n)$, and use this sample to process point queries.

What remains is to introduce the relaxed definition of a well-positioned node and show that it has the claimed properties.

4.6 Relaxed well positioned points

For $Q \geq \lceil 1+n/2 \rceil$, we define the Q-quantile distance $m_Q(v)$ for a point v as the distance of the Qth closest point to v. We then define $\textsc{MinMed}_Q$ as the minimum Q-quantile distance over all points. Now, we define a point v to be Q well positioned if $m_{\lceil 1+n/2 \rceil}(v) \leq 2\,\textsc{MinMed}_Q$.

Now observe that at least half the points have $m_Q(v) \leq 2\,\textsc{MinMed}_Q$ and in particular are well positioned (extension of Lemma 7). Also observe that if z is Q well positioned then for any node u, $\mathrm{dist}(z,u) \leq 3m_Q(u)$ (extension of Lemma 8). We can also verify that for any $Q < 0.6n$ (any constant strictly smaller than 1 would do), a base set S_0 containing one Q well positioned point would also yield coefficients that satisfy the universal PPS property, albeit with a slightly larger constant.

We next show that we can identify a $0.6n$ well positioned point within a polynomially small error using very few distance computations:

▶ **Lemma 13.** *We can identify a* $0.6n$ *well positioned point with probability* $1 - O(1/poly(n))$ *using* $O(\log^2 n)$ *distance computations.*

Proof. We choose uniformly at random a set of points C of size $O(\log n)$. For each point in $v \in C$, we choose a uniform sample S_v of $O(\log n)$ points and compute the 0.55 quantile of $\{\mathrm{dist}(v,u) \mid u \in S_v\}$. We then return the point $v \in C$ with the minimum sample 0.55 quantile.

We refer to C as the set of candidates. Note that since at least half the points $v \in V$ are such that $m_{0.6n}(v) \leq 2\,\textsc{MinMed}_{0.6n}$, the set C contains such a point with probability $1 - O(1/poly(n))$.

The estimates are such that with probability $1 - O(1/poly(n))$, for all points in C, the sample 0.55 quantile is between the actual 0.5 and 0.6 quantiles. Therefore the point we returned (with a polynomially small error) has $m_{0.5n}$ at most the smallest $m_{0.6n}$ in C, which is at most $2\,\textsc{MinMed}_{0.6n}$. ◀

5 All-pairs sum

We now establish the claims of Theorem 3 for the all-pairs sum problem. We start with the first part of the claim, which is useful for graphs, estimates $\text{APS}(V)$ using single-source computations. To do so, we apply Algorithm 1 to compute sampling coefficients $\boldsymbol{\gamma}$ and then apply Algorithm 2 to compute estimates $\hat{\text{W}}(v)$ for all v. Finally, we return the estimate $\widehat{\text{APS}}(V) = \frac{1}{2}\sum_{z\in V}\hat{\text{W}}(z)$.

To obtain an estimate $\widehat{\text{APS}}(V)$ with CV of at most ϵ, we choose a base set S_0 that contains 2 uniformly sampled nodes when applying Algorithm 1. We then use sample size of $O(\epsilon^{-2})$ to ensure that the per-node estimates $\hat{\text{W}}(z)$ have CV of at most ϵ. Note that the estimates of different nodes are correlated, as they all use the same sample, but the CV of the sum of estimates each with CV of at most ϵ must be at most ϵ. The total time amounts to $O(\epsilon^{-2})$ single-source distance computations.

To obtain universal PPS with polynomially small error we can identify a well positioned node with a polynomially small error, which can be done using $O(\log n)$ single-source computations. We then compute the sampling coefficients $\boldsymbol{\gamma}$ for a base set that contains this well-positioned node. (Which uses a single-source distance computation). The sampling coefficients we obtain have the universal PPS property and the sample-based estimates are concentrated. A sample size of size $O(\epsilon^{-2}\log n)$ would yield a relative error of at most ϵ with probability $1 - 1/poly(n)$, for each $\hat{\text{W}}(z)$ and thus for the sum $\widehat{\text{APS}}(V)$. In total, we used $O(\epsilon^{-2}\log n)$ single-source computations.

The remaining part of this section treats the second part of the claim of Theorem 3, which applies to the all-pairs sum problem in metric spaces. We start with an overview of our approach. In order to obtain a good sample of pairs, we would like to sample pairs proportionally to $p_{ij} = \frac{\text{dist}(i,j)}{\text{APS}(V)}$. The obvious difficulty we have to overcome is that the explicit computation of the probabilities p_{ij} requires a quadratic number of distance calculations.

Our first key observation is that we can obtain a sample with (nearly) the same statistical guarantees if we relax a little the sampling probabilities and the sample size: For some constant $c \geq 1$, we work with probabilities that satisfy $p_{ij} \geq c^{-1}\frac{\text{dist}(i,j)}{\text{APS}(V)}$ and use a sample of size $k = c\epsilon^{-2}$.

We use independent sampling with replacement to compute a multiset S of pairs of points from $V \times V$. The estimator we use is the sample average inverse probability estimator:

$$\widehat{\text{APS}}(V) = \frac{1}{|S|}\sum_{(i,j)\in S}\text{dist}(i,j)/p_{ij} \ .$$

This sample average is an unbiased estimate of $\text{APS}(V)$ and has CV of at most $\sqrt{k/c}$ which is ϵ when we use sample size $k = c\epsilon^{-2}$. Moreover, each summand is by definition at most $c\,\text{APS}(V)$ and therefore we obtain concentration by a direct application of Hoeffding's inequality: The probability of a relative error that is larger than ϵ when the sample size is k is at most $2e^{-2k\epsilon^2c^{-2}}$. In particular, if we take a sample size that is $O(\epsilon^{-2}\log n)$, we obtain that the probability that the relative error exceeds ϵ is polynomially small in n.

We next discuss how we facilitate such sampling efficiently. We would like to be able to sample with respect to relaxed p_{ij} and also have the sampling probabilities available for estimation. We show that we can express a set of relaxed probabilities (for some constant c) as the outer product of two probability distributions over points, $\boldsymbol{\gamma}\boldsymbol{\rho}^T$. The distribution $\boldsymbol{\gamma}$ has the universal PPS property with respect to some constant c'. The probability distribution $\boldsymbol{\rho}$ has the property that for some constant c'', for all v, $\rho_v \geq c''\frac{\text{W}(v)}{\sum_u \text{W}(u)}$. We now observe that for some constant $c = c'c''$, for all pairs u, v, $\rho_u\gamma_v \geq c\frac{\text{dist}(u,v)}{\text{APS}(V)}$. That is, we can sample

according to $p_{uv} = \rho_u \gamma_v$ and satisfy the relaxed conditions and obtain the desired statistical guarantees.

What remains is to provide details on (i) how we use the vectors $\boldsymbol{\gamma}$ and $\boldsymbol{\rho}$ to obtain a sample of pairs and (ii) how we compute such vectors that satisfy our conditions within a polynomially small error. These are addressed in the next two subsections.

5.1 Sampling pairs using the coefficient vectors

We show how we obtain k samples (v,u) from $\gamma_v \rho_u$ efficiently, using computation that is $O(n+k)$. Many sampling schemes (with or without replacement) will have the concentration properties we seek and the implementations are fairly standard. For completeness, we describe a scheme that computes independent samples with replacement. Our scheme obtains a sample from $V \times V$ by sampling independently a point i according to the probability distributions $\boldsymbol{\gamma}$ and a point j according to distribution ρ and returning (i,j).

What remains is to describe how we can obtain k independent samples with replacement from a probability vector $\boldsymbol{\gamma}$ in time $O(n+k)$.

We arbitrarily order the points, WLOG $i \in V$ is the ith point in the order. We compute $a_i = \sum_{h<i} \gamma_h$ and associate the intervals $[a_i, a_i + \gamma_i]$ with the point i.

To randomly draw a point $i \in V$ according to $\boldsymbol{\gamma}$, we can draw a random number $x \sim U[0,1]$ and take the point $i \in V$ such that $x \in [a_i, a_i + \gamma_i)$. If we have k sorted random values, we can map all of them to points in V in $O(n)$ time using one pass on the sorted values and the sorted nodes. For completeness, we describe one way to obtain a sorted set of k independent random draws $x_1, \ldots, x_k \sim U[0,1]$ using $O(k)$ operations: (i) We draw k values $y_1, \ldots y_k$ where $y_i \sim Exp[k+1-i]$ is exponentially distributed with parameters $k+1-i$. This can be done by drawing independent uniform $u_i \sim U[0,1]$ and take $y_i = -\ln(u_i)/(k+1-i)$. (iii) Now observe that $x'_i \equiv \sum_{j \le i} y_j$ for $i \in [k]$ are k independent exponential random variables with parameter 1 which are sorted in increasing order. We can then transform x'_i to uniform random variables x_i using $x_i = 1 - \exp(-x'_i)$. Since the transformation is monotone, we obtain that x_i are sorted. Note that prefix sums of y_j and hence all x_i can be computed in $O(k)$ operations. Also note that we only need precision to the point needed to identify the point that each x_i maps into.

5.2 Computing the coefficient vectors

We recall that universal PPS coefficients can be computed using Algorithm 1 using n distance computations (and $O(n)$ additional computation), when our base set S_0 contains a well positioned point. The probability vector $\boldsymbol{\gamma}$ we work with is the universal PPS coefficients scaled to have a sum of 1.

We next discuss how we obtain the probability distribution $\boldsymbol{\rho}$. We show that given a $0.6n$ well positioned point (see Section 4.6), we can compute ρ_v that has the claimed properties with very high probability. From Lemma 13, we can identify a point that is $0.6n$ well positioned with probability at least $(1 - 1/poly(n))$, using only $O(\log^2 n)$ distance computations. We use the following lemma, which a variation of claim used for the pivoting upper bound estimate in [6]. What it roughly says is that for any node u and any node z that is within a constant times some quantile distance from u, we can get a constant factor approximation of $\mathrm{W}(u)$ from $\mathrm{W}(z)$ and $\mathrm{dist}(u,z)$.

▶ **Lemma 14.** *Consider a point u and a point z such that $\mathrm{dist}(u,z)$ is at most c times the distance of the $(qn)^{th}$ closest point to u. Then*

$$\mathrm{W}(u) \le n\,\mathrm{dist}(u,z) + \mathrm{W}(z) \le \left(1 + \frac{2c}{1-q}\right) \mathrm{W}(u) \ .$$

Proof. Left hand side is immediate from the triangle inequality. To establish the right hand side, first note that $(1-q)n$ of the points are at least as far as $\mathrm{dist}(z,u)/c$, thus $W(u) \geq \frac{(1-q)}{c} n \,\mathrm{dist}(u,z)$. From triangle inequality we have $\mathrm{W}(z) \leq \mathrm{W}(u) + n\,\mathrm{dist}(u,z)$. Combining we get:

$$\mathrm{W}(z) + n\,\mathrm{dist}(u,z) \leq \mathrm{W}(u) + 2n\,\mathrm{dist}(u,z) \leq (1 + \frac{2c}{1-q})\,\mathrm{W}(u)\ . \qquad \blacktriangleleft$$

Now consider a point z that is $0.6n$ well positioned and using the rough estimates

$$\hat{\mathrm{W}}'(u) = n\,\mathrm{dist}(u,z) + \mathrm{W}(z)$$

for all points u and accordingly the sampling probabilities

$$\rho_i = \frac{\hat{\mathrm{W}}'(i)}{\sum_j \hat{\mathrm{W}}'(j)}\ .$$

By definition, for all points u, the point z satisfies $\mathrm{dist}(u,z) \leq 3m_{0.6n}(u)$. We therefore can apply the lemma with $q = 0.6$ and $c = 3$ and obtain that for all v, $\rho_v \geq \frac{1-q}{1-q+2c} \frac{\mathrm{W}(v)}{\sum_j \mathrm{W}(j)}$. Note that given z, the vector $\boldsymbol{\rho}$ can be computed for all points using n distance computations, from z to all other points.

6 Uniform sampling based estimates

For completeness, we briefly present another solution for the all-points/nodes problem that is based on uniform sampling. The disadvantages over our weighted sampling approach is that it provides biased estimates and requires $\epsilon^{-2}\log n$ samples even when we are interested only in per-query guarantees.

To do so, we use a key lemma proved by Indyk [18, 17]. A proof of this lemma also appears in [22], and used to establish the correctness of his approximate 1-median algorithm.

▶ **Lemma 15.** *Let $Q \subset V$, $|Q| = k$ sampled uniformly at random (from all subsets of size k). Let u and v be two vertices such that $\mathrm{W}(v) \geq (1+\epsilon)\,\mathrm{W}(u)$. Then $\Pr(\mathrm{W}_Q(u) > \mathrm{W}_Q(v)) \leq e^{-\epsilon^2|Q|/64}$.*

Lemma (15) shows that if the average distance of two nodes differ by a factor larger than $1+\epsilon$, and we use a sample of size $\Omega(\epsilon^{-2})$ then the probability that the vertex of smaller average distance has larger average distance to the sample decays exponentially with the sample size. This lemma immediately implies that the 1-median with respect to a sample of size $O(\log n/\epsilon^2)$ is $(1+\epsilon)$-approximate 1-median with high probability.

To approximate all-pairs $\mathrm{W}(u)$, we use a uniform sample of size $O(\epsilon^{-2}\log n)$ and order the nodes according to the average distance to the sample. Using the lemma, and comparing to the ideal sorted order by exact $\mathrm{W}(v)$, two nodes v, u that are transposed have with high probability $\mathrm{W}(v)$ and $\mathrm{W}(u)$ within $1 \pm \epsilon$ from each other.

Recall however that the average distance to the uniform sample can be a very bad approximation of the average distance to the data set. We therefore perform adaptively another set of $O(\epsilon^{-1}\log n)$ single-source distance computations to compute exact $\mathrm{W}(v)$ of enough nodes in this nearly sorted order, so that the difference between exact $\mathrm{W}(v)$ of consecutive processed nodes is within $(1 \pm \epsilon)$.

We also mention here, for completeness, an improved approximate 1-median algorithm provided by Indyk. This algorithm only applies in metric spaces and computes a $(1+\epsilon)$-approximate 1-median with constant probability using only $O(n\epsilon^{-2})$ distance computations

(eliminating the logarithmic factor). The algorithm works in iterations, where in each iteration a fraction of the points, those with largest average distance to the current sample, are excluded from further considerations. The sample size is then increased by a constant factor, obtaining more accurate estimates for the remaining points. The final sample size used is linear, but the set of remaining nodes is very small. This algorithm only applies in metric spaces because, as we mentioned in the introduction, arbitrary distance computations are not efficient in graphs. Indyk's approach can be extended to compute any approximate quantile of the distribution with similar probabilistic guarantees.

7 Hardness of Computing Sum of All-Pairs Distances

In this section we show that if there is a truly subcubic algorithm for computing $\text{APS}(V)$, the exact sum of all pairs distances then there is a truly subcubic algorithm for computing All Pairs Shortest Paths (APSP).

Williams and Williams [23] showed that APSP is subcubic equivalent to negative triangle detection. In the *negative triangle detection problem* we are given an undirected weighted graph $G = (V, E)$ with integer weights in $\{-M, ..., M\}$ and the goal is to determine if the graph contains a negative triangle, that is, a triangle whose edge weights sum up to a negative number. Therefore to show that a subcubic algorithm for $\text{APS}(V)$ implies a subcubic algorithm to APSP it suffices to give a subcubic reduction from the negative triangle detection problem to computing $\text{APS}(V)$. We show this by the following lemma.

▶ **Lemma 16.** *Given a $O(T(n, m))$ time algorithm for computing the sum of all distances ($\text{APS}(V)$) there is $O(T(n, m) + n^2)$ time algorithm for detecting a negative triangle.*

Proof. For an input instance $G = (V, E)$ for the negative triangle detection problem we construct a graph $G' = (V', E')$ for the sum of all distances problem. The vertex set V' is the union of three copies of V, that is $V' = V_1 \cup V_2 \cup V_3$ where vertex $u_i \in V_i$, $i = 1, 2, 3$, corresponds to vertex $u \in V$. We set $E' = \{(u, v) \mid u, v \in V'\}$, that is G' is a complete graph.

Let $\omega(e)$ denote the length of an edge $e \in E$. Recall that $\omega(e) \in \{-M, ..., M\}$. Let $N = 4M$. We define the length $\omega'(e)$ of an edge $e \in E'$ as follows. For every $(u, v) \in E$ we define $\omega'(u_1, v_2) = N + \omega(u, v)$, $\omega'(u_2, v_3) = N + \omega(u, v)$, and $\omega'(u_3, v_1) = 2N - \omega(u, v)$. We set $w(e) = 3N/2$ for any other edge $e \in E'$.

We claim that $\text{APS}(V') = \sum_{(u,v)\in E'} \omega'(u, v)$ if and only if G does not contain a negative triangle. In other words, we claim that either every edge in G' is a shortest path or G contains a negative cycle.

To see the first direction, assume G contains a negative triangle $(u, v), (u, x), (x, v)$. Now consider the path $P = (u_3, x_2), (x_2, v_1)$ from u_3 to v_1. Note that the length of this path is $\omega'(u_3, x_2) + \omega'(x_2, v_1) = N + \omega(u_3, x_2) + N + \omega(x_2, v_1) < 2N - \omega(u_3, v_1) = \omega'(u_3, v_1)$, where the strict inequality follows since $(u, v), (u, x), (x, v)$ is a negative triangle. If follows that $\text{APS}(V') < \sum_{(u,v)\in E'} \omega'(u, v)$.

To see the second direction, assume that $\text{APS}(V') < \sum_{(u,v)\in E'} \omega'(u, v)$. We need to show that G has a negative triangle.

We first claim that for every edge (u, v) which does not correspond to an edge in G (and hence $w(e) = 3N/2$) we have $\omega'(u, v) = \text{dist}_{G'}(u, v)$ (regardless if G has a negative triangle or not). To see this, note that $\omega'(u, v) = 3N/2 = 6M$ and that every path from u to v that consists of more than one edge is of weights at least $2N - 2M = 6M$. The same argument also holds for every edge from V_1 to V_2 and for every edge from V_2 to V_3.

It follows that only edges $(x, y) \in E'$ such that $x \in V_3$ and $y \in V_1$ may not be shortest paths. If $\text{APS}(V') < \sum_{(u,v)\in E'} \omega'(u, v)$ then there must be an edge $(u_3, v_1) \in E'$ such that

$u_3 \in V_3$ and $v_1 \in V_1$ and the edge (u_3, v_1) is not a shortest path. It is not hard to verify that only paths of the form $(u_3, x_2), (x_2, v_1)$ such that both edges (u_3, x_2) and (x_2, v_1) correspond to edges of G, could be shorter than the path (u_3, v_1). Let $(u_3, x_2), (x_2, v_1)$ be the shortest path from u_3 to v_1. We get that $N + \omega(u_3, x_2) + N + \omega(x_2, v_1) = \omega'(u_3, x_2) + \omega'(x_2, v_1) < \omega'(u_3, v_1) = 2N - \omega(u_3, v_1)$. So $\omega(u_3, x_2) + \omega(x_2, v_1) + \omega(u_3, v_1) < 0$ and G has a negative triangle. ◀

8 Extensions and Comments

8.1 The distribution of centrality values

What can we say about the centrality distribution? First we observe that the range of average distance $\mathrm{W}(v)/n$ values is between D/n to D, where D is the diameter (maximum distance between a pair of points in V). To see the upper bound, note that the average of values that are at most D, is at most D. For the lower bound, let u and v be nodes such that $\mathrm{dist}(u, v) = D$. Then for all $h \in V$, from triangle inequality, $\mathrm{dist}(u, h) + \mathrm{dist}(h, v) \geq D$, thus, $\mathrm{W}(h) \geq D$.

▶ **Lemma 17.** *The highest average distance value must satisfy*

$$\max_{v \in V} \mathrm{W}(v)/n \geq D/2 \ .$$

Proof. Consider the two nodes u and v such that $\mathrm{dist}(u, v) = D$. From triangle inequality, any point $h \in V$ has $\mathrm{dist}(u, h) + \mathrm{dist}(h, v) \geq D$. Summing over h we obtain that $\mathrm{W}(u) + \mathrm{W}(v) \geq nD$. Therefore, either $\mathrm{W}(u)$ or $\mathrm{W}(v)$ is at least $nD/2$. ◀

▶ **Lemma 18.** *If $z = \arg\min_{v \in V} \mathrm{W}(v)$ is the 1-median, then at least half the nodes satisfy $\mathrm{W}(v) \leq 3\,\mathrm{W}(z)$.*

Proof. Take the median distance $m(z)$ from z. Then the average distance from z is at least $m(z)/2$. Thus, $n \cdot m(z) \leq 2\,\mathrm{W}(z)$. Consider now a node v that is one of the $n/2$ closest to z. For any node u, $\mathrm{dist}(v, u) \leq \mathrm{dist}(z, u) + m(z)$. Therefore,

$$\mathrm{W}(v) = \sum_u \mathrm{dist}(v, u) \leq \sum_u \mathrm{dist}(z, u) + nm(z) \leq nm(z) + \mathrm{W}(z) \leq 3\,\mathrm{W}(z) \ . \quad ◀$$

Last we observe that it is easy to realize networks where there is a large spread of centrality values. At the extreme, consider a single point (node) that has distance D to a very tight cluster of $n - 1$ points. The points in the cluster have $\mathrm{W}(v) \approx D$ whereas the isolated point has $\mathrm{W}(v) \approx nD$. More generally, networks (or data sets) containing well separated clusters with different sizes would exhibit a spread in centrality values.

A side comment is that as a corollary of the proof of Lemma 17 we obtain that the all pairs sum in metric spaces can be estimated with CV ϵ and good concentration by the scaled average of distances of $O(n\epsilon^{-2})$ pairs sampled uniformly at random – as established in [2]. This is because there are at least $n - 1$ pairwise distances that are at least $D/2$, since each point that is not an endpoint of the diameter is of distance at least $D/2$ from at least one of the endpoints. Since the maximum distance is D, this immediately implies our claim. Recall, however, that when we are restricted to using $O(\epsilon^{-2})$ single-source distance computations from a uniform sample of nodes, the estimates can have large CV, but a similar bound can still be obtained using our weighted sampling approach (see Corollary 3).

8.2 Limitation to distances

We showed that any set of points V in any metric space can be "sparsified" in the sense that a weighted sample of size $O(\epsilon^{-2})$ allows us to estimate $\mathrm{W}(v)$ for any point v in the space. We refer to such a sample as a *universal PPS* sample, since it encapsulates a PPS sample of the entries in each row of the matrix. One can ask if we can obtain similar sparsification with respect to other nonnegative symmetric matrices. We first observe that in general, the size of a universal PPS sample may be $\Omega(n)$: Consider a matrix $A_{n\times n}$ so that for $i \in [n/2]$, $A_{2i-1,2i} \gg 0$ but all other entries are 0 (or close to 0). The average of each row is dominated by the other member of the pair $(2i-1, 2i)$, and therefore, any universal PPS sample must sample most points with probability close to 1.

Such a matrix can not be realized with distances, as it violates the triangle inequality, but it can be realized when entries correspond to (absolute value) of inner products of n vectors in n-dimensional Euclidean space $\mathbb{R}^n$. In this case, the sampling question we ask is a well studied embedding problem [21], for which it is known that the size of a universal PPS sample (the terminology *leverage scores* is used) can be of size $\Theta(d\epsilon^{-2})$, where d is the dimension [9, 21]. Intuitively, the gap between the universal PPS size between distances and inner products stems from the observation that being "far" (large distance) is something that usually applies with respect to many nodes, whereas being "close" (large inner product) is a local property.

8.3 Weighted centrality

Often different points v have different importance $\beta(v)$. In this case, we would like our centrality measure to reflect that by considering a weighted average of distances

$$\frac{\sum_i \beta(i)\,\mathrm{dist}(x_i, x_j)}{\sum_i \beta(i)}\ .$$

Our results, and in particular, the sampling construction extend to the weighted setting. First, instead of uniform base probabilities $1/n$, we use PPS probabilities according to $\beta(i)/\sum_j \beta(j)$ for node i. Second, when considering distances and probabilities from a base node, we use weight equal to the product of $\beta(v)\,\mathrm{dist}(u,v)$ (product of β and distance.). Third, in the analysis, we need to take quantiles/medians with respect to β mass and not just the number of points.

8.4 Adaptive (data dependent) sampling

We showed that the number of samples needed to determine an approximate 1-median on graphs is $O(\epsilon^{-2}\log n)$, where for each sample we perform a single-source distance computation. This bound is worst case which materializes when the 1-median z is such that all other points have $\mathrm{W}(u) = (1+\epsilon)\,\mathrm{W}(z)$. In this case, only the exact 1-median qualifies as an approximate 1-median and also, since there are so many other points, some are likely to have estimated $\hat{\mathrm{W}}(u) < \hat{\mathrm{W}}(z)$ if we use a smaller sample. On realistic instances, however, we would expect a larger separation between the 1-median and most other points. This would allow us to use fewer samples if we adaptively determine the sample size. Such an approach was proposed in [7] to identify a node with approximate maximum marginal influence and similarly can be applied here for the 1-median.

9 Conclusion

Weighted samples are often used as compact summaries of weighted data. With weighted sampling, even of very skewed data, a PPS sample of size ϵ^{-2} would provide us with good estimates with CV of $O(\epsilon)$ on the total sum of the population. The surprise factor of our result, which relies only on properties of metrics, is that we can design a single set of sampling probabilities, which we termed *universal PPS*, that forms a good weighted sample from the perspectives of *any* point in the metric space. Moreover, we do so in an almost lossless way in terms of the sample size to estimation quality tradeoffs. In particular, the sample size does not depend on the number of points n or the dimension of the space. Another perhaps surprising consequence of our results is that there is a rank-1 matrix that approximates the PPS probabilities of the full pairwise distances matrix. The approximation can be expressed as the outer product of two vectors, which can be computed using a linear number of distance computations.

References

1 A. Abboud, F. Grandoni, and V. Vassilevska Williams. Subcubic equivalences between graph centrality problems, APSP and diameter. In *SODA*. ACM-SIAM, 2015.

2 K. Barhum, O. Goldreich, and A. Shraibman. On approximating the average distance between points. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, volume 4627 of *Lecture Notes in Computer Science*. Springer, 2007.

3 A. Bavelas. A mathematical model for small group structures. *Human Organization*, 7:16–30, 1948.

4 A. Bavelas. Communication patterns in task oriented groups. *Journal of the Acoustical Society of America*, 22:271–282, 1950.

5 M. A. Beauchamp. An improved index of centrality. *Behavioral Science*, 10:161–163, 1965.

6 E. Cohen, D. Delling, T. Pajor, and R. F. Werneck. Computing classic closeness centrality, at scale. In *COSN*. ACM, 2014.

7 E. Cohen, D. Delling, T. Pajor, and R. F. Werneck. Sketch-based influence maximization and computation: Scaling up with guarantees. In *CIKM*, 2014.

8 E. Cohen, N. Duffield, C. Lund, M. Thorup, and H. Kaplan. Efficient stream sampling for variance-optimal estimation of subset sums. *SIAM J. Comput.*, 40(5), 2011.

9 M. B. Cohen and R. Peng. ℓ_p row sampling by lewis weights. In *STOC*. ACM, 2015.

10 T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

11 D. Eppstein and J. Wang. Fast approximation of centrality. In *SODA*, pages 228–229, 2001.

12 M. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM*, 34(3):596–615, 1987.

13 L. C. Freeman. A set of measures of centrality based on betweeness. *Sociometry*, 40:35–41, 1977.

14 L. C. Freeman. Centrality in social networks: Conceptual clarification. *Social Networks*, 1, 1979.

15 O. Goldreich and D. Ron. Approximating average parameters of graphs. *Random Struct. Algorithms*, 32(4):473–493, 2008.

16 J. Hamidzadeh, R. Monsefi, and H. S. Yazdi. DDC: distance-based decision classifier. *Neural Computing and Applications*, 21(7), 2012.

17 P. Indyk. Sublinear time algorithms for metric space problems. In *STOC*. ACM, 1999.

18 P. Indyk. *High-dimensional Computational Geometry*. PhD thesis, Stanford University, 2000.

19 K. Okamoto, W. Chen, and X. Li. Ranking of closeness centrality for large-scale social networks. In *Proc. 2nd Annual International Workshop on Frontiers in Algorithmics*, FAW. Springer-Verlag, 2008.

20 G. Sabidussi. The centrality index of a graph. *Psychometrika*, 31(4):581–603, 1966.

21 M. Talagrand. Embedding subspaces of l_1 into l_1^n. *Proc. of the American Math. Society*, 108(2):363–369, 1990.

22 M. Thorup. Quick k-median, k-center, and facility location for sparse graphs. *SIAM J. Comput.*, 34(2):405–432, 2004.

23 V. Vassilevska Williams and R. Williams. Subcubic equivalences between path, matrix and triangle problems. In *FOCS*. IEEE, 2010.

24 S. Wasserman and K. Faust, editors. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.