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

Shiri Chechik* Edith Cohen*† Haim Kaplan*†

May 5, 2015

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 show that the average distance (and hence the centrality) for all nodes in a network can be estimated in time $O(\epsilon^{-2} mn \log n)$, where $n$ and $m$ are the number of nodes and edges. For a set of $n$ points in a general metric space, we show that using preprocessing that uses $O(n)$ distance computations we can compute a weighted sample of only $O(\epsilon^{-2})$ points such that the average distance from any query point to our points can be estimated using distance computations to the sampled points. Our estimates are unbiased with normalized mean square error (NRMSE) of at most $\epsilon$. Increasing the sample size by a $\log n$ factor ensures a very high probability of relative error at most $\epsilon$ for all nodes. Finally, we show that the average distance between all pairs of points in a metric space can be estimated using $O(n + \epsilon^{-2})$ distance computations.

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 [14, 15, 25], was proposed by Bavelas [4], Beauchamp [5], and Sabidussi [21]. Formally, for a graph $G = (V, E)$ with $|V| = n$ nodes, the classic closeness centrality of $v \in V$ is

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

where $\text{dist}(u, v)$ is the shortest-path distance 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, $\sum_{x \in V} \text{dist}(z, x)/n$, 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 \cite{11} (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 \cite{17} 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, the underlying metric are shortest paths distances. Our results also extend to inputs specified as directed strongly connected graphs where the distance are the round trip distances \cite{7}. 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

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

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

We list the fundamental computational problems related to these measures.

- **All nodes sums**: Compute $W(v)$ 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 $W(v)$.
- **1-median**: Compute the node $u$ of maximum centrality or equivalently, minimum $W(u)$.
- **All-pairs sum**: Compute the sum of the distances between all pairs, that is $\text{APS} = \frac{1}{2} \sum_{v} W(v)$.

Computationally, in metric spaces, we seek algorithms that compute a small number of distances between pairs of points. In graphs, a distance computation between a specific pair of nodes $u, v$ seems to require in the worst-case a single source computation from one of the 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 \cite{13}. 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 $W(v)$ for all $v$, all-pairs sum, and 1-median, we need to compute all pairs distances, 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 \cite{24}. Abboud et al \cite{11} 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 \textsuperscript{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\textsuperscript{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 \( \eta \) from its mean is at least 1 − \((CV)^2/(\eta)^2\). Therefore a CV of \( \epsilon \) implies that the estimator is within a relative error of \( \eta = c\epsilon \) from its mean with probability \( \geq 1 - 1/e^2 \).

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 \( \log n \) we get a relative error of \( \epsilon \) for each one of polynomially many queries with a polynomially small error probability. In particular, we can estimate the sum of the distances of the 1-median up to a relative error of \( \epsilon \) with a polynomially small error probability.

**Previous work:** We review previous work on scalable approximation of 1-median, all-nodes, and all-pairs sum. These problems were studied on both metric spaces and graphs. A natural approach to approximate the centrality of nodes is to take a uniform sample \( S \) of \( n \) single source distance computations to determine all distances from \( v \) \( \in S \) to \( u \) \( \in V \), and then estimate \( W(v) \) by \( \hat{W}(v) = \frac{n}{|S|} W_S(v) \), where \( W_S(v) = \sum_{a \in S} \text{dist}(v,a) \) is the sum of the distances from \( v \) to the nodes of \( S \). This approach was used by Indyk \textsuperscript{19} to compute a \((1 + \epsilon)\)-approximate 1-median in a metric space using only \( O(\epsilon^{-2} n) \) distance computations (See also \textsuperscript{18} for a similar result with a weaker bound.). We discuss the uniform sampling approach in more details in Section \textsuperscript{6} where for completeness, we show how it can be applied for the all-nodes problem.

The sample average of a uniform sample was also used to estimate all-nodes centrality \textsuperscript{12} (albeit with weaker, additive guarantees) and to experimentally identify the (approximate) top \( k \) centralities \textsuperscript{20}. 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 \( W(v) \).

Recently, Cohen et al \textsuperscript{7} obtained \( \epsilon \) NRMSE estimates for \( W(v) \) for any \( v \), using single-source distance computations from each node in a uniform sample of \( \epsilon^{-3} \) nodes. Estimates that are within a relative error of \( \epsilon \) for all nodes were obtained using \( \epsilon^{-3} \log n \) single-source computations. The approach applies in any metric space. The estimator for a point \( v \) is based on a hybrid of using a sample average of a uniform sample for “close” nodes to \( v \) and estimating distances to nodes “far” from \( v \) through 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 \textsuperscript{18} showed that it can be estimated by scaling up the average of \( \hat{O}(n\epsilon^{-3.5}) \) distances between pairs of points selected uniformly at random. The estimate has a relative error of at most \( \epsilon \) with constant probability. Barhum, Goldreich, and Shraibman \textsuperscript{2} improved Indyk’s bound and showed that a uniform sample of \( \hat{O}(n\epsilon^{-2}) \) distances suffices and also argued that this

---

\( ^1 \)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 incorrect 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.
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 [16] 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, with a constant probability to be within a relative error of $\epsilon$. They also showed that $O(\epsilon^{-2}\sqrt{n})$ distances from fixed node $s$ to random nodes $v$ suffice to estimate $W(v)$, with a constant probability to be within a relative error of $\epsilon$. 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: Generally, a single distance computation is almost equivalent to a single source shortest path computation.

**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 $\hat{W}(z)$ with a CV of $\epsilon$ for any $z$. A sample of size $O(\epsilon^{-2}\log n)$ suffices for ensuring a relative error of $\epsilon$ for all nodes in a graph or for polynomially many queries.

The sampling algorithm is provided in Section 2. This algorithm computes a coefficient $\gamma_v$ 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 $u \in V$, which we refer to as global PPS (Probability Proportional to Size) sampling probabilities. Using the probabilities $p_u$, we can obtain a Poisson sample $S$ of expected size $\sum_u p_u = O(k)$ or a VarOpt sample [9] that has exactly that size (to the bounding closest integers).

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

$$\hat{W}(z) = \sum_{u \in S} \frac{\text{dist}(z, u)}{p_u}. \quad (2)$$

Since $p_u > 0$ for all $u$, the estimate $\hat{W}(z)$ are unbiased.

We provide a detailed analysis in Section 4. Our global PPS probabilities have the property that when choosing $k = O(\epsilon^{-2})$, $\hat{W}(z)$ has CV $\epsilon$. Moreover, the estimates have good concentration, so using a larger sample size of $O(\epsilon^{-2}\log n)$ we obtain relative error of $\epsilon$ with high probability (at least $1 - 1/poly(n)$) for all nodes $v \in V$.

Intuitively, to obtain a sample with such guarantees for a single node $z$, the sampling probability of a node $v$ should be (roughly) proportional to its distance $\text{dist}(z, v)$ from $z$. That is, the sampling probabilities need to be PPS from the point of view of node $v$ (the “size” of a node here is its distance from $v$): A perfect PPS sample of size $\epsilon^{-2}$ has CV of $\epsilon$. It is far from clear apriori, however, that there is a single set of sampling probabilities which simultaneously provides such a guarantee for all nodes and also has total size that is $O(\epsilon^{-2})$. In particular, that it is possible to obtain such a “good for all” sample with size that does not depend on $n$ or the dimensionality of the space. We show that we can indeed obtain such a sample for any metric space.

Beyond establishing the existence of a good set of sampling probabilities, we are interested in computing these probabilities, and the sample, very efficiently, in near-linear time. The dominant component of computing the global PPS sampling probabilities is performing $O(1)$ single-source distance computations. Therefore, $O(m\log n)$ in graphs and $O(n)$ pairwise distance queries in a metric space. A global PPS sample of size $O(\epsilon^{-2})$ can then be computed in a single pass over nodes ($O(n)$ computation) from the sampling probabilities. We represent the sample $S$ as a collection $\{(u, p_u)\}$ of nodes/points and respective sampling probabilities.
We can then use our sample for estimation. When the input is a graph, we compute single-source distances from each node in $S$ to all other nodes in order to estimate $W(v)$ of all $v \in V$. This requires $O(|S|m \log n)$ computation while maintaining state $O(n)$.

**Theorem 1.1.** All-nodes $W(v)$ in a graph can be estimated unbiasedly with CV $\epsilon$ using $O(\epsilon^{-2} m \log n)$ computation. To obtain relative error of $\epsilon$ with high probability for all nodes, we need $O(\epsilon^{-2} m \log^2 n)$ computation.

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

**Theorem 1.2.** We can preprocess a set of points $V$ in a metric space using $O(n)$ distance computations ($O(1)$ single source distance computations) to generate a set of sampling probabilities and a corresponding sample $S$ of size $|S| = O(\epsilon^{-2})$. For a point query $z$, we can unbiasedly estimate $\hat{W}(z)$ with CV $\epsilon$ using the distances between $z$ and the points in $S$.

For the all-pairs sum problem we obtain the following:

**Theorem 1.3.** All-pairs sum can be estimated unbiasedly with the following statistical guarantees:

- CV of at most $\epsilon$, using $O(\epsilon^{-2})$ single-source distance computations. Within a relative error of $\epsilon$ with probability $1 - O(1/\text{poly}(n))$, using $O(\epsilon^{-2} + \log n)$ single-source distance computations.

- With CV of at most $\epsilon$ and within relative error of $\epsilon$ with probability $1 - O(1/\text{poly}(n))$, using $O(n + \epsilon^{-2})$ distance computations (metric space).

The proof is provided in Section 5. The part of the claim that uses single-source distance computations is established by using the estimate $\hat{\text{APS}}(V) = \frac{1}{2} \sum_{z \in V} \hat{W}(z)$. When the estimates have CV of at most $\epsilon$, even if correlated, so does the estimate $\hat{\text{APS}}(V)$. For the very high probability claim, we use $O(\log n)$ single-source computation to ensure we obtain global PPS sampling probabilities with very high probability (details are provided later), which imply that each estimate $\hat{W}(z)$, and hence the sum, has both CV of at most $\epsilon$ and concentration. The second part is established via a weighted sampling of $O(\epsilon^{-2})$ pairs of points roughly proportionally to distances $\text{dist}(u, v)$, which is enabled by a representation of the sampling distribution constructed from $O(n)$ distance computations.

Compared to the all-nodes sums algorithms of [7], our result here improves the dependence on $\epsilon$ from $\epsilon^{-3}$ to $\epsilon^{-2}$, which is likely to be optimal for a sampling based approach, provides an unbiased estimate, and also facilitates approximate average distance oracles with very small storage in metric spaces (the approach of [7] would require the oracle to store a histogram of distances from each of $\epsilon^{-3}$ sampled nodes). For the all-pairs sum problem on graphs, we obtain an algorithm that uses $O(\epsilon^{-2})$ single source distance computations, which improves over $O(\epsilon^{-3})$, which is implied by [7]. Compared to the all pairs sum algorithms of [18, 2] in metric spaces, we obtain a much stronger bound that is dominated by $O(n + \epsilon^{-2})$ instead of $O(ne^{-2})$ distance computations.

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 nodes. The algorithm applies in both graphs and metric spaces (replace “nodes” by “points”). The input to the algorithm is a set $S_0$ of base nodes and a parameter $k$. There are two components. We first compute a set of sampling coefficients $\gamma_v$ for each node $v$ such that $\sum_v \gamma_v = O(1)$. Then for a parameter $k$, we can compute sampling probabilities $p_v = \min\{1, k\gamma_v\}$. We can then use the probabilities $p_v$ to draw a sample of expected size $O(k)$. We would usually apply the algorithms once with a prespecified $k$ to obtain a sample, but there are applications (see discussion in Section 8.4) where we want to choose the sample size adaptively using the same coefficients.

Algorithm 1 Compute global PPS coefficients and sample

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

Output: A global PPS sample $S$

// Compute sampling coefficients

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

foreach $u \in S_0$ do
  Compute single-source distances $\text{dist}(u, v)$ from $u$ to all other nodes $v$
  $W \leftarrow \sum_v \text{dist}(u, v)$

foreach node $v \in V$ do
  $\gamma_v \leftarrow \max\{\gamma_v, \frac{\text{dist}(u, v)}{W}\}$

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 $S$

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 $\sum_v p_v \leq k \sum_v \gamma_v = O(k)$.

Choosing the base set $S_0$: Our analysis uses the following 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 $u$. Let $\text{MINMED} = \min_v m(v)$ be the minimum median distance of any node $v$. We say that a node $u$ is well positioned if $m(u) \leq 2 \text{MINMED}$, that is, $m(u)$, the median distance of $u$ is within a factor of 2 of the minimum median distance.

Let $v$ be a node of minimum median distance, that is $m(v) = \text{MINMED}$. Then all $\lceil 1 + n/2 \rceil$ nodes closest to $v$ are well positioned. Indeed, let $u$ be one of the $\lceil 1 + n/2 \rceil$ nodes closest to $v$. Then $\text{dist}(u, v) \leq \text{MINMED}$ and a ball of radius $2 \text{MINMED}$ around $u$ contains all the $\lceil 1 + n/2 \rceil$ nodes closest to $v$. So $m(u) \leq 2 \text{MINMED}$.

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

Lemma 2.1. If $u$ is a well positioned node, then for every node $z$ we have that $\text{dist}(z, u) \leq 3m(z)$.  

6
Proof. This holds since for every two nodes \( u \) and \( z \) we have that \( \text{dist}(u, z) \leq m(u) + m(z) \), this is because there must be at least one node that is both within distance \( m(u) \) from \( u \) and within distance \( m(z) \) from \( z \), and applying the triangle inequality. Now, 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 \( m(z) \). Moreover, because at least half the nodes are well positioned, it is relatively simple to ensure one is included with high probability.

We shall see that in order to obtain the property that each estimate \( \hat{W}(v) \) has CV \( O(\epsilon) \) we can choose \( k = \epsilon^{-2} \) and ensure that the base set \( S_0 \) is as follows: It suffices that \( S_0 \) includes a single well positioned node. It also suffices that the base set \( S_0 \) constitutes of a uniform sample of \( \geq 2 \) nodes.

Moreover, if we choose \( k = \epsilon^{-2} \log n \) and ensure that \( S_0 \) contains a well-positioned node (with very high probability), we obtain from concentration that all (polynomially many) queries are within relative error \( O(\epsilon) \) with very high probability.

An interesting property of our sampling coefficients is that they can not grow too much even if the 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 \( S_0 \).

3 Estimation

3.1 Centrality values for all nodes in a graph

For graphs, we compute estimates \( \hat{W}(v) \) for all nodes \( v \in V \) as in Algorithm 2. We initialize all estimates to 0, we then perform an SSSP computation from each node in \( u \in S \). When scanning node \( v \), we add \( \text{dist}(u, v)/p_u \) to the estimate \( \hat{W}(v) \). The computation of this algorithm is \( O(|S|m \log n) \), dominated by the \(|S|\) Dijkstra computations from each node in the sample \( S \).

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

Input: Weighted graph \( G \), a sample \( S = \{(u, p_u)\} \)

foreach \( v \in V \) do
  \( \hat{W}(v) \leftarrow 0 \)

foreach \( u \in S \) do
  Perform a single-source shortest-paths computation from \( u \).
  foreach scanned node \( v \in V \) do
    \( \hat{W}(v) \leftarrow \hat{W}(v) + \text{dist}(u, v)/p_u \)

return \((v, \hat{W}(v))\) for \( v \in V \)

3.2 Point queries (metric space)

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

We show that when \( k = \epsilon^{-2} \), and when \( S_0 \) includes either a uniform sample of size at least 2 or a well-positioned node, then each estimate \( \hat{W}(v) \) has CV of \( O(\epsilon) \).

If there is a well positioned node in \( S_0 \) then we also obtain concentration, which means that if we take \( k = O(\epsilon^{-2} \log n) \), we obtain with high probability relative error at most \( \epsilon \) for all nodes (or for polynomially many queries).

**Lemma 4.1.** 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 \( v \),

\[
\gamma_v \geq \frac{1 - q}{4} \frac{\text{dist}(z, v)}{W(z)}.
\]  

(3)

**Proof.** From the specification of Algorithm 1 the sampling coefficients \( \gamma_v \) satisfy

\[
\gamma_v \geq \max \left\{ \frac{1}{n}, \frac{\text{dist}(u, v)}{W(u)} \right\}.
\]  

(4)

Let \( Q = \text{dist}(u, z) \). Consider a partition of the points \( v \in V \) “close” and “far” points according to distance from \( z \):

\[
L = \{ v \in V \mid \text{dist}(z, v) \leq 2Q \}
\]

\[
H = \{ v \in V \mid \text{dist}(z, v) > 2Q \}.
\]

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

\[
\gamma_v \geq \frac{1}{n} \geq \left( \frac{1 - q}{2} \right) \left( \frac{2}{1 - q} \right) = \left( \frac{1 - q}{2} \right) \frac{2Q}{(1 - q)Q} = \left( \frac{1 - q}{2} \right) \frac{\text{dist}(z, v)}{W(z)},
\]  

(5)

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

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

\[
\gamma_v \geq \frac{\text{dist}(u, v)}{W(u)} \geq \frac{\text{dist}(z, v) - Q}{W(z) + nQ}.
\]  

(6)

In particular, for \( v \in H \), we have

\[
\text{dist}(z, v) - Q \geq \frac{1}{2} \text{dist}(z, v).
\]  

(7)

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

\[
nQ \leq \frac{W(z)}{1 - q},
\]  

(8)

and

\[
W(z) + nQ \leq W(z) \left( 1 + \frac{1}{1 - q} \right) = W(z) \left( \frac{2 - q}{1 - q} \right).
\]  

(9)
Substituting (9) and (7) in (6), we obtain that for \( v \in H \),
\[
\gamma_v \geq \frac{\text{dist}(z,v) - Q}{W(z) + nQ} \geq \frac{1}{2} \left( \frac{1 - q}{2 - q} \right) \frac{\text{dist}(z,v)}{W(z)}.
\] (10)

Combined, the lower bounds (5) for \( v \in L \) and (10) for \( v \in H \) mean that for all \( v \), \( \gamma_v \) is at least
\[
c(q) = \frac{1}{4} (1 - q) \text{ times the proportion of the contribution of dist}(u,v) \text{ to } W(u).
\]

Lemma 4.2. Consider a set of sampling coefficients \( \gamma_v \) such that for a node \( z \), for all \( v \), for some \( c > 0 \),
\[
\gamma_v \geq c \frac{\text{dist}(z,v)}{W(z)}.
\]
Consider a (Poisson or VarOpt) sample \( S \) obtained with parameter \( k \) with probabilities
\[
p_v = \min\{1, k\gamma_v\} \quad \text{(as in Algorithm 1),}
\]
and the inverse probability estimator \( \hat{W}(z) \) as in (2). Then
\[
\text{Var}[\hat{W}(z)] \leq \frac{W(z)^2}{k \cdot c}.
\] (11)

Proof. The variance of our estimator is
\[
\text{Var}[\hat{W}(z)] = \sum_v \left[ p_v \left( \frac{\text{dist}(z,v)}{p_v} - \text{dist}(z,v) \right)^2 + (1 - p_v) \text{dist}(z,v)^2 \right] = \sum_v \left( \frac{1}{p_v} - 1 \right) \text{dist}(z,v)^2.
\] (12)

Note that nodes \( v \) for which \( p_v = 1 \) contribute 0 to the variance. For the other nodes we use the lower bound \( p_v \geq c k \frac{\text{dist}(z,v)}{W(z)} \).
\[
\sum_{v \in V} \left( \frac{1}{p_v} - 1 \right) \text{dist}(z,v)^2 = \sum_{v \in V | p_v < 1} \left( \frac{1}{p_v} - 1 \right) \text{dist}(z,v)^2
\]
\[
\leq \frac{W(z)}{k \cdot c} \sum_{v \in V} \text{dist}(z,v)
\]
\[
\leq \frac{W(z)^2}{k \cdot c}.
\]

Lemma 4.3. Suppose that \( S_0 \) contains a uniform random sample of \( b \) nodes. Then for any node \( z \),
\[
\text{Var}[\hat{W}(z)] \leq \frac{W(z)^2}{k} \frac{4b}{b - 1}.
\] (13)

Proof. We apply Lemma 4.2 with the bound on the coefficients as in Lemma 4.1 with \( u \) being the closest node to \( z \) in \( S_0 \). Assume that \( u \) is the \( x \)th closest node to \( z \), we have
\[
\text{Var}[\hat{W}(z) | x] \leq \frac{W(z)^2}{k} \frac{4}{1 - x/n}.
\] (14)

Observe that \( x \sim R \) 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 \( \in [1, n - b + 1] \) \((R = 1 \text{ iff } u = z)\). We have that the probability of rank \( x \) is
\[
b \left( \frac{1}{n} \right) \left( \frac{n - x}{n - 1} \right) \left( \frac{n - x - 1}{n - 2} \right) \ldots \left( \frac{n - x - b + 2}{n - b + 1} \right) \leq b \left( \frac{1 - x}{n} \right)^{b-1}.
\]
(We choose items 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 \( \text{Var}[\hat{W}(z)] \) is the expectation, over \( x \sim R \), of \( \text{Var}[\hat{W}(z) \mid x] \). So from (14), we get

\[
\text{Var}[\hat{W}(z)] \leq n - b + 1 \sum_{x=1}^{n-b+1} b \left( 1 - \frac{x}{n} \right)^{b-1} \left( \frac{W(z)^2}{k} \frac{4}{(1 - x/n)} \right)
\]

\[
\leq \frac{W(z)^2}{k} 4b \sum_{x=1}^{n-b+1} \left( 1 - \frac{x}{n} \right)^{b-2}
\]

\[
\leq \frac{W(z)^2}{k} 4b \int_{0}^{1} (1 - y)^{b-2} dy
\]

\[
= \frac{W(z)^2}{k} \frac{4b}{b - 1}.
\]

\( \square \)

It follows from Lemma 4.3 that if we use \( b \geq 2 \) uniformly selected nodes in \( S_0 \) and \( k = \epsilon^{-2} \), then for any node \( z \), our estimator has \( \text{Var}[\hat{W}(z)] = O\left( \frac{W(z)^2}{k} \right) \). This concludes the proof of the per-node (per-point) estimation quality claims in Theorems 1.1 and 1.2.

### 4.1 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 \( \gamma_v \) satisfy what we call a global PPS property:

**Lemma 4.4.** Suppose that \( S_0 \) contains a well-positioned node \( u \). Then for all nodes \( v \),

\[
\gamma_v \geq \frac{1}{14} \max_{z} \frac{\text{dist}(z, v)}{W(z)}.
\]

**Proof.** We show that for any node \( z \), \( \gamma_v \geq \frac{1}{14} \frac{\text{dist}(z, v)}{W(z)} \) using a variation on the proof of Lemma 4.1. Since \( u \) is well positioned, \( \text{dist}(z, u) \leq 3m(z) \) (see Lemma 2.1). We partition the nodes into two sets. The set \( L \) contains the \( \geq \lfloor n/2 - 1 \rfloor \approx n/2 \) nodes \( v \) such that \( \text{dist}(v, z) \leq 6m(z) \). The set \( H \) contains the remaining nodes. Note that by definition, \( W(z) \geq m(z) \frac{n}{2} \). We obtain that for all \( v \in L \),

\[
\frac{\text{dist}(v, z)}{W(z)} \leq \frac{6m(z)}{m(z) \frac{n}{2}} = \frac{12}{n}.
\]

Therefore,

\[
\gamma_v \geq \frac{1}{n} \geq \frac{1}{12} \frac{\text{dist}(v, z)}{W(z)}.
\]

We next consider \( v \in H \). From the triangle inequality, \( \text{dist}(u, v) \geq \text{dist}(z, v) - \text{dist}(z, u) \geq \text{dist}(z, v) - 3m(z) \geq \text{dist}(z, v)/2 \). We also have \( W(u) \leq W(z) + n \text{dist}(u, z) \leq W(z) + 3nm(z) \leq 7W(z) \). Therefore

\[
\gamma_v \geq \frac{\text{dist}(u, v)}{W(u)} \geq \frac{(\text{dist}(z, v)/2)}{7W(z)} = \frac{1}{14} \frac{\text{dist}(z, v)}{W(z)}.
\]

\( \square \)

As a corollary, applying Lemma 4.2, we obtain:

**Corollary 4.5.** If \( S_0 \) contains a well-positioned node, then for any node \( z \), \( \text{Var}[\hat{W}(z)] \leq 14 \frac{W(z)^2}{k} \).
4.2 Upper bound on the coefficients sum

One consequence of Lemma 4.4 is that the coefficients $\gamma_u$ can not grow too much even if the base set $S_0$ includes all nodes.

**Corollary 4.6.**

$$\sum_u \gamma_u = O(1),$$

where

$$\gamma_u = \max_v \frac{\text{dist}(u, v)}{W(v)}.$$  

**Proof.** From obtaining the global PPS property using a single well-positioned node (Lemma 4.4) we have coefficients $\gamma_v$ that satisfy

$$\sum_v \gamma_v \leq 2$$

and at the same time, for any two nodes $v, u$,

$$\gamma_v \geq \frac{1}{14} \frac{\text{dist}(u, v)}{W(u)}.$$  

Therefore, $\sum_v \gamma_v \leq 14 \sum_v \gamma_v \leq 28$. \qed

4.3 Concentration

Lastly, we establish concentration of the estimates, which will conclude the proof of the all-nodes claim in Theorem 1.1. We need the following lemma:

**Lemma 4.7.** If $S_0$ is such that our sampling coefficients are near PPS for a node $z$, that is, there is a constant $c$ such that for all nodes $v$, $\gamma_v \geq c \frac{\text{dist}(z, v)}{W(z)}$, and we use $k = O(\epsilon^{-2} \log n)$, then

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

**Proof.** We apply the Chernoff-Hoeffding bound. We have

$$p_v \geq \min\{1, \frac{\text{dist}(z, v)}{\tau}\} = \min\{1, \frac{ck \text{dist}(z, v)}{W(z)}\}. \quad (16)$$

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

The contribution of the nodes with $\text{dist}(z, v) \leq \tau$ are thus independent random variables in the range $[0, \tau]$ with expectation $\text{dist}(z, v)$. We apply the Chernoff-Hoeffding bound to bound the error on the contribution of these nodes. The application is a fairly standard exercise, noting that (i) We do take into account the exact contributions due to nodes with $\text{dist}(z, v) > \tau$, which allows us a proportionally larger relative error for nodes with $\text{dist}(z, v) \leq \tau$. (ii) The bound statements can be rescaled to work with random variables in the range $[0, \tau]$ instead of the more common $[0, 1]$. (ii) The bounds apply also when the random variables are continuous $[0, \tau]$. \qed

In order to conclude the proof of the theorem, we would need the condition of Lemma 4.7 to hold for all nodes $z$ with very high probability $1 - O(1/poly(n))$. In this case, we can apply a union bound. Then with probability at least $1 - O(1/poly(n))$, the estimates $\hat{W}(z)$ for all nodes $z$ have a relative error of at most $\epsilon$. The same argument applies to polynomially many queries in metric spaces.

It follows from Lemma 4.4 that if we choose $S_0 = \{u\}$ such that $u$ is well positioned, then for any node $z$, our sampling coefficients satisfy the condition.
It is also simple to identify a well-positioned node with a very high probability. To do so, we recall that most nodes are well positioned. Therefore, if we take a uniform random sample $U$ of $O(\log n)$ nodes and take the node $z = \arg\min_{u \in U} m(u)$ with minimum distance to its $n/2$ closest node, then we are guaranteed with $1 - 1/poly(n)$ probability that $z$ is well positioned. This identification step amounts to $O(\log n)$ single-source distance computations and is dominated by the other component of the computation.

Alternatively, we can simply place $O(\log n)$ uniformly selected nodes in $S_0$. In this case, with very high probability, $S_0$ would contain a well positioned node. Alternatively, we also have that each node $z$ would have a node in $S_0$ within distance $m(z)$ and we can apply Lemma 4.1 with $q = 0.5$. Since (see lemma 4.6) the sum of the coefficients is $O(1)$, the sample size is still $O(k)$.

5 All-pairs sum

We now establish the claims of Theorem 1.3 for the all-pairs sum problem. The first part of the claim estimates $\text{APS}(V)$ using single-source computations. We compute estimate $\hat{W}(v)$ for all $v$, as in the previous section, and return the estimate $\hat{\text{APS}}(V) = \frac{1}{2} \sum_{z \in V} \hat{W}(z)$. An estimate with CV of at most $\epsilon$ can be obtained by using a base set $S_0$ that contains of $2$ uniformly sampled nodes. Since each individual estimates is unbiased with CV of at most $\epsilon$, so is the sum. Note that the estimates are correlated, since we use the same sample for all nodes, but the claim still holds. To obtain very high probability, we first identify a well positioned node with very high probability, which can be done using $O(\log n)$ single-source computations, and then use it in the base set. When the base set contains a well-positioned node, the sampling coefficients have the global PPS property and the estimates we obtain are concentrated.

We start with an overview of our approach. Observe that to obtain an estimate with CV of $\epsilon$ and good concentration for the sum, $\text{APS}(V)$, it suffices to sample with replacement $\epsilon^{-2}$ pairs according to probabilities $p_{ij} = \frac{\text{dist}(i,j)}{\text{APS}(V)}$ and return the average over samples of the inverse probability estimate $\text{dist}(i,j)/p_{ij}$. The obvious difficulties are that the computation of the probabilities $p_{ij}$ and even their representation alone are quadratic.

The first important observation is that we can obtain the same statistical guarantees when we relax this a little, with respect to some $c \geq 1$, and use a sample size $c\epsilon^{-2}$ and probabilities that satisfy $p_{ij} \geq c^{-1}\text{dist}(i,j)/\text{APS}(V)$. So we only seek this relaxed requirement.

An important ingredient is facilitating such sampling, while avoiding the explicit quadratic computation. For this we note that the relaxed probabilities have a simple representation as the outer product of two probability distributions over points, $\gamma \rho^T$. The distribution $\gamma$ has the global PPS property with respect to some constant. The distribution $\rho$ has the property that for some constant $c'$, for all $v$, $\rho_v \geq c' \frac{W(v)}{\sum_u W(u)}$. We now observe that for some $c''$, 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 to provide details on (i) how we use the vectors $\gamma$ and $\rho$ to obtain a sample of pairs and (ii) how we compute such vectors that satisfy our conditions with very high probability. 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. To do so, we can sample $k$ (with replacement) from each of $\gamma$ and $\rho$, and pair the samples from one set with a random permutation of the other...
To obtain a sample for one vector, say $\gamma$, we give a sample count of $\lfloor k\gamma \rfloor$ to $v$, and then varopt sample the points using probabilities $k\gamma - \lfloor k\gamma \rfloor < 1$, using a single structured pass over the vector [6]. Note that this structured sample is not independent, but satisfies the concentration bounds.

Note that the computation involved in obtaining the sample of pairs is linear $O(n)$.

### 5.2 Computing the coefficient vectors

We recall that global PPS coefficients can be computed using Algorithm 1 using $n$ distance computations (and $O(n)$ additional computation) when our sample $S_0$ contains a well positioned point.

For our purposes here, we work with a slightly relaxed definition of a well positioned point. For $Q \ge \lceil 1 + n/2 \rceil$, we define the $Q$-quantile distance $m_Q(v)$ for a point $v$ as the distance of the $Q$th closest point to $v$. We then define $\minmed_Q$ as the minimum $Q$-quantile distance over all points. Now, we define a point $v$ to be $Q$ well positioned if $m_{\lceil 1 + n/2 \rceil}(v) \le 2 \minmed_Q$.

Now observe that at least half the points have $m_Q(v) \le 2 \minmed_Q$ and in particular are well positioned. Also observe (extension of Lemma 2.1) that if $z$ is $Q$ well positioned then for any node $u$, $\text{dist}(z, u) \le 3m_Q(u)$. 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 global PPS property, albeit with a slightly larger constant.

We next show that we can identify a $0.6n$ well positioned point with very high probability using very few distance queries:

**Lemma 5.1.** 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 $\{\text{dist}(v, u) \mid u \in S_v\}$. 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 $m_{0.6n}(v) \le 2 \minmed_{0.6n}$, 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 very high probability has $m_{0.5n}$ at most the smallest $m_{0.6n}$ in $C$, which is at most $2 \minmed_{0.6n}$. $\square$

We now show that given a $0.6n$ well positioned point, we can also compute coefficients $\rho_v$ with the claimed properties with very high probability. We use the following lemma, which a variation of claim used for the pivoting upper bound estimate in [7]:

**Lemma 5.2.** Consider a point $u$ and a point $z$ such that $\text{dist}(u, z)$ is at most $c$ times the ($qn$)$^{th}$ closest point to $u$. Then

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

**Proof.** Left hand side is triangle inequality, right hand side follows from $(1 - q)n$ of the points being at least as far as $\text{dist}(z, u)/c$, thus $W(u) \ge \frac{(1 - q)}{c} n \text{dist}(u, z)$. $\square$

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

$$\tilde{W}(u) = n \text{dist}(u, z) + W(z)$$

13
for all points $u$ and accordingly the sampling coefficients

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

By definition, for all points $u$, the point $z$ satisfies $\text{dist}(u, z) \leq 3m_{0.6n}(u)$. We therefore can apply the lemma with $q = 0.6$ and $c = 3$ and obtain that for all $v, \rho_v \geq \frac{1-q}{c} \frac{\hat{W}(v)}{\sum_j \hat{W}(j)}$. Note that given $z$, the coefficients 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 [19, 18]. A proof of this lemma also appears in [23], and used to establish the correctness of his approximate 1-median algorithm.

**Lemma 6.1.** 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 $W(v) \geq (1 + \epsilon) W(u)$. Then $\Pr(W_Q(u) > W_Q(v)) \leq e^{-\epsilon^2 |Q|/64}$.

Lemma (6.1) 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 $W(u)$, 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 $W(v)$, two nodes $v, u$ that are transposed have with high probability $W(v)$ and $W(u)$ 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 $W(v)$ of enough nodes in this nearly sorted order, so that the difference between exact $W(v)$ 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 the exact sum of all pairs distances then there is a truly subcubic algorithm for computing the All Pairs Shortest Paths.

For a graph $G = (V, E, \omega)$. Let $\delta_G(u, v)$ be the shortest path distance from $u$ to $v$ in $G$ for $u, v \in V$. Let $\sigma(G) = \sum_{u,v} \delta_G(u, v)$.

We use as an intermediate step a subcubic equivalence between Negative Triangle Detection to APSP. In the negative triangle detection we are given an undirected weighted graph $G = (V, E)$ with integer weights in $\{-M, \ldots, M\}$ 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.

Vassilevska Williams and Williams showed in [24] that the Negative Triangle Detection problem has subcubic equivalence to APSP. It is therefore enough for our needs to show a subcubic reduction from the negative triangle detection problem to computing the sum of all distances. We show it in the following lemma.

Lemma 7.1. Given a $\tilde{O}(T(n, m))$ time algorithm for computing the sum of all distances, there is $\tilde{O}(T(n, m) + n^2)$ time algorithm for detecting a negative triangle.

Proof. Consider the input instance $G = (V, E, \omega)$ for the triangle detection problem. Construct the following graph $G'$ = $(V', E', \omega')$. $V'$ consists of three copies of $V$, that is $V' = V_1 \cup V_2 \cup V_3$. Consider a node $u \in V, u_i$ is the copy of $u$ in $V_i$ for $1 \leq i \leq 3$. Let $N = 4M$.

Set $E'$ to be the complete graph, that is, $E' = \{(u, v) \mid u, v \in V'\}$. Next, we set the weights of $E'$ in two steps. In the first step, for every edge $(u, v) \in E$ set the following three edges in $E'$: $(u_1, v_2), (u_2, v_3), (u_3, v_1)$. Set the weights of these edges as follows: $\omega'(u_1, v_2) = N + \omega(u, v), \omega'(u_2, v_3) = N + \omega(u, v), \omega'(u_3, v_1) = 2N - \omega(u, v)$. In the second step, set the weights of all edges that were not set in the first step to be $3N/2$. We claim that $\sigma(G') = \sum_{(u,v) \in E'} \omega'(u,v)$ if and only if $G$ does not contain a negative triangle. In other words, we are claiming 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 second last inequality follows from the fact that $(u, v)(u, x)(x, v)$ is a negative triangle. If follows that $\sigma(G') < \sum_{(u,v) \in E'} \omega'(u,v)$.

To see the second direction, assume that $\sigma(G') < \sum_{(u,v) \in E'} \omega'(u,v)$. We first claim that every edge $(u, v)$ that was added in the second step satisfies $\omega'(u, v) = \delta_G(u, v)$ (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 will also hold for every edge from $V_1$ to $V_2$ and for every edge from $V_2$ to $V_3$. Therefore, the only edges $(x, y) \in E'$ such that $(x, y)$ may not be a shortest path are edges between $V_3$ and $V_1$. If $\sigma(G') < \sum_{(u,v) \in E'} \omega'(u,v)$ 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 $(x_2, v_1)$ were added in the first step to $E'$ 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 $N + \omega(u_3, x_2) + N + \omega(x_2, v_1) = \omega'(u_3, x_2) + \omega'(x_2, v_1) < \omega'(u_3, v_1) = 2N - \omega(u_3, v_1)$. It follows that $\omega(u_3, x_2) + \omega(x_2, v_1) + \omega(u_3, v_1) < 0$. Hence, $G$ has a negative triangle. The lemma follows. \hfill \Box
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 \( W(v)/n \) 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 \( \text{dist}(u, v) = D \). Then for all \( h \in V \), from triangle inequality, \( \text{dist}(u, h) + \text{dist}(h, v) \geq D \), thus, \( W(h) \geq D \).

**Lemma 8.1.** The highest average distance value must satisfy

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

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

**Lemma 8.2.** If \( z = \arg \min_{v \in V} W(v) \) is the 1-median, then at least half the nodes satisfy \( W(v) \leq 3 W(z) \).

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

\[
W(v) = \sum_u \text{dist}(v, u) \leq \sum_u \text{dist}(z, u) + nm(z) \leq nm(z) + W(z) \leq 3 W(z) .
\]

\( \square \)

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 \( W(v) \approx D \) whereas the isolated point has \( W(v) \approx nD \). 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 8.1, we obtain that the all pairs sum in metric spaces can be estimated with \( CV \) and good concentration by the scaled average of distances of \( O(\epsilon^{-2}) \) 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 1.3).

8.2 Limitation to distances

We showed that any set of points \( V \) in any metric space can be “sparsified,” to a weighted sample of size \( O(\epsilon^{-2}) \) so that we can estimate \( W(v) \) for any point \( v \) in the space from this sample. What can we say about similarly sparsifying other nonnegative symmetric matrices? We observe that in general, the size of a global PPS sample may be \( \Omega(|V|) \), even when we are limited to use query points from \( V \): We take \( n/2 \) pairs of
points so that the members of each pair are very far from each other, but close to all other points. The row average of each row is dominated by the other member of the pair. Therefore, any global PPS sample must sample most vectors with probability close to $1$.

In particular, another natural objective is estimating the average inner product to a set $V$ of vectors in a Euclidean space $\mathbb{R}^d$, which is a well studied embedding problem [22]. In contrast to average distances, it is known that the size of a global PPS sample in this case (the terminology leverage scores is used) has worst-case dependence on the dimension of $\Omega(d\epsilon^{-2})$ [10, 22]. Intuitively, the difference in the global 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

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

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 $\beta(u) \text{dist}(u, v)$ (product of $\beta$ and distance.). Third, in the analysis, we need to take quantiles/medians with respect to $\beta$ mass and not just the number of points.

### 8.4 1-median with adaptive (data dependent) sampling

In the worst-case, when limited to using single-source distance computations, we need $O(\epsilon^{-2} \log n)$ such computations in order to identify an approximate 1-median, even if we only require a constant probability. The worst case materializes when the 1-median is some point $z$ and all other points have $W(u) = (1 + \epsilon) W(z)$. This means that 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 $\hat{W}(u) < \hat{W}(z)$ if we use a smaller sample.

Adaptive sampling is useful on realistic instances. It exploits the fact that these instances are typically “easier” in that the distribution of $W(v)$ values is skewed and hence easier to separate out. We start with a sample of size $O(\delta^{-2})$, compute estimates based on this sample, and iteratively increase the sample size only if necessary.

More precisely, when $\delta > \epsilon$, we can stop when $z = \arg \min_u \hat{W}(u)$ has $W(z) \leq (1 + \delta) \hat{W}(z)$ and also that for all $u \neq z$, $\hat{W}(u) > (1 + \delta) \hat{W}(z)$. It is possible to use $\delta > \epsilon$ only when the data has a well-separated 1-median. If there is another node $z'$ with $W(z') \leq (1 + \epsilon) W(z)$, it would be necessary to use $\delta \leq \epsilon$ to identify an approximate 1-median. When working with $\delta \leq \epsilon$, we can use a simplified stopping condition, only requiring that $W(z) \leq (1 + \delta) \hat{W}(z)$.

This can be done with starting with a small value of $k$ and increasing the sample size as necessary, taking $\delta = 1/\sqrt{k}$, using the same estimation coefficients. As we increase $k$, the sample grows in a monotone way (as the use of same coefficients makes samples for different $k$ are coordinated).

More precisely, we increase $k$ smoothly while tracking the node with minimum estimated $\hat{W}(v)$ (and when $k \leq \epsilon^{-2}$ also tracking the second smallest value). We stop when the condition is satisfied. Correctness follows by noting that using the estimated minimum can lead to large under-estimates, since expectation of
the minimum can be much smaller than minimum of expectations, but the overestimate distribution still has a controlled error.

Adaptive sampling was similarly used in [8, 7], also in this context of identifying an approximate minimum using samples. The case for adaptive sampling is to be able to use fewer samples and still obtain confidence intervals.

**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 $\epsilon^{-2}$ 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 *global PPS*, that forms a good weighted sample from the perspectives of all points 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, G. Cormode, and N. Duffield. Structure-aware sampling: Flexible and accurate summarization. *Proceedings of the VLDB Endowment*, 2011.

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

[8] E. Cohen, D. Delling, T. Pajor, and R. F. Werneck. Sketch-based influence maximization and computation: Scaling up with guarantees. In *CIKM*. ACM, 2014. full version http://research.microsoft.com/apps/pubs/?id=226623.

[9] 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.

[10] M. B. Cohen and R. Peng. $\ell_p$ row sampling by lewis weights. In *STOC*. ACM, 2015.

[11] T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.
[12] D. Eppstein and J. Wang. Fast approximation of centrality. In SODA, pages 228–229, 2001.

[13] M. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. J. Assoc. Comput. Mach., 34(3):596–615, 1987.

[14] L. C. Freeman. A set of measures of centrality based on betweenness. Sociometry, 40:35–41, 1977.

[15] L. C. Freeman. Centrality in social networks: Conceptual clarification. Social Networks, 1, 1979.

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

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

[18] P. Indyk. Sublinear time algorithms for metric space problems. In STOC. ACM, 1999.

[19] P. Indyk. High-dimensional Computational Geometry. PhD thesis, Stanford University, 2000.

[20] 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.

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

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

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

[24] V. Vassilevska Williams and R. Williams. Subcubic equivalences between path, matrix and triangle problems. In FOCS. IEEE, 2010.

[25] S. Wasserman and K. Faust, editors. Social Network Analysis: Methods and Applications. Cambridge University Press, 1994.