Pruning based Distance Sketches with Provable Guarantees on Random Graphs

Hongyang Zhang
Stanford University
hongyang@cs.stanford.edu

Huacheng Yu
Harvard University
yuhch123@gmail.com

Ashish Goel
Stanford University
ashishg@stanford.edu

ABSTRACT

Measuring the distances between vertices on graphs is one of the most fundamental components in network analysis. Since finding shortest paths requires traversing the graph, it is challenging to obtain distance information on large graphs very quickly. In this work, we present a preprocessing algorithm that is able to create landmark based distance sketches efficiently, with strong theoretical guarantees. When evaluated on a diverse set of social and information networks, our algorithm significantly improves over existing approaches by reducing the number of landmarks stored, preprocessing time, or stretch of the estimated distances.

On Erdos-Renyi graphs and random power law graphs with degree distribution exponent $2 < \beta < 3$, our algorithm outputs an exact distance data structure with space between $\Theta(n^{5/4})$ and $\Theta(n^{3/2})$ depending on the value of β, where n is the number of vertices. We complement the algorithm with tight lower bounds for Erdos-Renyi graphs and the case when β is close to two.

ACM Reference Format:
Hongyang Zhang, Huacheng Yu, and Ashish Goel. 2019. Pruning based Distance Sketches with Provable Guarantees on Random Graphs. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3308558.3313708

1 INTRODUCTION

Computing shortest path distances on large graphs is a fundamental problem in computer science and has been the subject of much study [27, 36, 37]. In many applications, it is important to compute the shortest path distance between two given nodes, i.e. to answer shortest path queries, in real time. Graph distances measure the closeness or similarity of vertices and are often used as one of the most basic metric in network analysis [29, 35, 39, 40]. In this paper, we will focus on efficient and practical implementations of shortest path queries in classes of graphs that are relevant to web search, social networks, and collaboration networks etc. For such graphs, one commonly used technique is that of *landmark-based labelings*: every node is assigned a set of landmarks, and the distance between two nodes in computed only via their common landmarks. If the set of landmarks can be easily computed, and is small, then we obtain both efficient pre-processing and small query time.

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313708

Landmark based labelings (and their more general counterpart, *Distance Labelings*), have been studied extensively [9, 36]. In particular, a sequence of results culminating in the work of Thorup and Zwick [37] showed that labeling schemes can provide a multiplicative 3-approximation to the shortest path distance between any two nodes, while having an overhead of $O(\sqrt{n})$ storage per node on average in the graph (we use the standard notation that a graph has n nodes and m edges). In the worst case, there is no distance labeling scheme that always uses sub-quadratic amount of space and provides exact shortest paths. Even for graphs with maximum degree 3, it is known that any distance labeling scheme requires $\Omega(n^{3/2})$ total space [26]. In sharp contrast to these theoretical results, there is ample empirical evidence that very efficient distance labeling schemes exist in real world graphs that can achieve much better approximations. For example, Akiba et al. [3] and Delling et al. [20] show that current algorithms can find *landmark based labelings* that use only a few hundred landmarks per vertex to obtain *exact distance*, in a wide collection of social, Web, and computer networks with millions of vertices. In this paper, we make substantial progress towards closing the gap between theoretical and observed performance. We show that natural landmark based labeling schemes can give exact shortest path distances with a small number of landmarks for a popular model of (unweighted and undirected) web and social graphs, namely the heavy-tailed random graph model. We also formally show how further reduction in the number of landmarks can be obtained if we are willing to tolerate an additive error of one or two hops, in contrast to the multiplicative 3-approximation for general graphs. Finally, we present practical versions of our algorithms that result in substantial performance improvements on many real-world graphs.

In addition to being simple to implement, landmark based shortest path algorithms also offer a qualitative benefit, in that they can directly be used as the basis of a social search algorithm. In social search [8], we assume there is a collection of keywords associated with every node, and we need to answer queries of the following form: given node v and keyword w, find the node that is closest to v among all nodes that have the keyword w associated with them. This requires an index size that is $O(L)$ times the size of the total social search corpus and a query time of $O(L)$, where L is the number of landmarks per node in the underlying landmark based algorithm; the approximation guarantee for the social search problem is the same as that of the landmark based algorithm. Thus, our results lead to both provable and practical improvements to the social search problem.

Existing models for social and information networks build on random graphs with some specified degree distribution [15, 22, 38], and there is considerable evidence that real-world graphs have power-law degree distributions [16, 23]. We will use the Chung-Lu

model [14], which assumes that the degree sequence of our graph is given, and then draws a "uniform" sample from graphs that have the same or very similar degree sequences. In particular, we will study the following question: *Given a random graph from the Chung-Lu model with a power law degree distribution of exponent β, how much storage does a landmark-based scheme require overall, in order to answer distance queries with no distortion?*

In the rest of the paper, we use the term "random power law graph" to refer to a graph that is sampled from the Chung-Lu model, where the weight (equivalently, the expected degree) of each vertex is independently drawn from a power law distribution with exponent β. We are interested in the regime when $\beta > 2$ — this covers most of the empirical power law degree distributions that people have observed on social and information networks [16]. Admittedly, real-world graphs have additional structure in addition to having power-law degree distributions [34], and hence, we have also validated the effectiveness of our algorithm on real graphs.

1.1 Our Results

Our first result corresponds to the "easy regime", where the degree distribution has finite variance ($\beta > 3$). We show that a simple procedure for generating landmarks guarantees exact shortest paths, while only requiring each node to store $\tilde{O}(\sqrt{n})$ landmarks. The same conclusion also applies to Erdős-Renyi graphs $G(n, \frac{c}{n})$ when $c > 1$, or when $c = 2 \log n$.

We then study the case where $2 < \beta < 3$. This is the most emblematic regime for power-law graphs, since the degree distribution has infinite variance but finite expectation. We present an algorithm that generates at most $\tilde{O}(n^{(\beta-2)/(\beta-1)})$ landmarks per node when $\beta \geq 2.5$; and $\tilde{O}(n^{(3-\beta)/(4-\beta)})$ landmarks per node when $2 < \beta < 2.5$. We obtain additional improvements if we allow an additive error of 1 or 2. See Figure 1 for an illustration of our results.

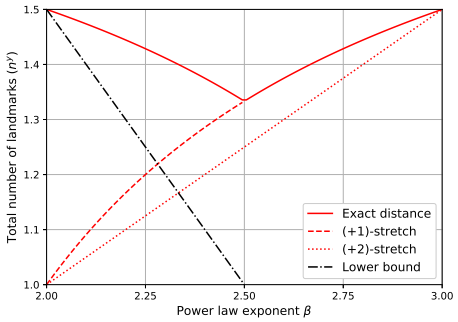

Figure 1: The x-axis is the exponent of the power law degree distribution and each value on the y-axis corresponds to a storage of $\tilde{O}(n^y)$. The lower bound is for exact distances.

While the dependence on β is complex, it is worth noting that in the entire range that we study ($\beta > 2$), the number of landmarks per node is at most $\tilde{O}(\sqrt{n})$ for *exact shortest paths*. This is in stark contrast to known impossibility results for general graphs, where no distance labeling with a multiplicative stretch less than 3 can use sub-linear space per node [26]. The query time of our algorithms is proportional to the number of landmarks per node, so we also get speed improvements.

Our algorithm is based on the pruned labeling algorithm of Akiba et al. [3], but differs in important ways. The pruned labeling algorithm initially posits that every node is a landmark for every other node, and then uses the BFS tree from each node to iteratively prune away unnecessary landmarks. In our approach, we apply a similar BFS with pruning procedure on a small subset of H (i.e. high degree vertices), but switch to lightweight local ball growing procedures up to radius l for all other vertices. As we show, the original pruned labeling algorithm requires storing $\tilde{\Omega}(n^2)$ landmarks on sparse Erdös-Rényi graphs. By growing local balls of size $\sqrt{n}$, our algorithm recovers exact distances with at most $\tilde{O}(n^{3/2})$ landmarks instead, for Erdös-Rényi graphs and random power law graphs with $\beta > 3$. Hence, our algorithm combines exploiting the existence of high-degree "central landmarks" with finding landmarks that are "locally important". Furthermore for $2 < \beta < 3$, by setting up the number of global landmarks H and the radius l suitably, we provably recover the upper bounds described in Figure 1. While the algorithmic idea is quite simple, the analysis is intricate.

We complement our algorithmic results with tight lower bounds for the regime when $\beta > 3$: the total length of any distance labeling schemes that answer distance queries exactly in this regime is almost surely $\tilde{\Omega}(n^{1.5})$. We also show that when $2 < \beta < 2.5$, any distance labeling scheme will generate labels of total size $\tilde{\Omega}(n^{3.5-\beta})$ almost surely. In particular, our algorithm achieves the optimal bound when β is close 2.

The parameter choice suggested by our theoretical analysis can be quite expensive to implement (as can earlier landmark based algorithms). We apply a simple but principled parameter tuning procedure to our algorithm that substantially improves the preprocessing time and generates a smaller set of landmarks at essentially no loss of accuracy. We conduct experiments on several real world graphs, both directed and undirected. First, compared to the pruned labeling algorithm, we find that our algorithm reduces the number of landmarks stored by 1.5-2.5x; the preprocessing time is reduced significantly as well. Next, we compare our algorithm to a variant of the distance oracle of Thorup and Zwick [37], which is believed to be theoretically optimal for worst-case graphs, as well as the distance sketch of Das Sarma et al [19] which has been found to be both efficient and useful in prior work [8]. For each graph, our algorithm substantially outperforms these two benchmarks. Details are in Section 5. It is important to note that the three algorithms we compare to also work much better on these real-world graphs than their theoretical guarantee, and we spend considerable effort tuning their parameters as well. Hence, the performance improvement given by our algorithm is particularly noteworthy.

It is worth mentioning that our technical tools only rely on bounding the growth rate of the breadth-first search. Hence we expect that our results can be extended to the related configuration model [22] as well. One limitation of our work is that the analysis does not apply directly to preferential attachment graphs, which correspond to another family of well known power law graphs. But we believe that similar results can be obtained there by adapting our analysis to that setting as well. This is left for future work.

Organizations: The rest of the paper is organized as follows. Section 2 introduces the basics of random graphs, reviews the pruned labeling algorithm and related work. Section 3 introduces our approach. We then present the analysis and experiments in Section 4 and Section 5. The lower bounds are presented in Section 6.

2 PRELIMINARIES AND RELATED WORK

2.1 Notations

Let $G = (V, E)$ be a directed graph with $n = |V|$ vertices and $m = |E|$ edges. For a vertex $x \in V$, denote by $d_{out}(x)$ the outdegree of x and $d_{in}(x)$ the indegree of x. Let $\mathcal{N}_{out}(x)$ denote the set of its out neighbors. Let $\text{dist}_G(x, y)$ denote the distance of x and y in G, or $\text{dist}(x, y)$ for simplicity. When G is undirected, then the outdegrees and indegrees are equal. Hence we simply denote by d_x the degree of every vertex $x \in V$. For an integer l and $x \in V$, denote by $\Gamma_l(x) = \{y : \text{dist}(x, y) = l\}$ the set of vertices at distance l from x. Denote by $N_l(x)$ the set of vertices at distance at most l from x.

We use notation $a \lesssim b$ to indicate that there exists an absolute constant $C > 0$ such that $a \leq Cb$. The notations $\tilde{O}(\cdot)$ and $\tilde{\Omega}(\cdot)$ hide poly-logarithmic factors.

2.2 Landmark based Labelings

In a landmark based labeling [21], each vertex x is assigned a set of forward landmarks $L_F(x)$ and backward landmarks $L_B(x)$. Each landmark set is a hash table, whose keys are vertices and values are distances. For example, if $y \in L_F(x)$, then the value associated with y would be $\text{dist}(x, y)$, which is the "forward" distance from x to y. Given the landmark sets $L_F(\cdot)$ and $L_B(\cdot)$, we estimate the distances as follows:

$$\min_{z \in L_F(x) \cap L_B(y)} \text{dist}(x, z) + \text{dist}(z, y), \forall x, y \in V. \tag{1}$$

If no common vertex is found between $L_F(x)$ and $L_B(y)$, then y is not reachable from x. In the worst case, computing set intersection takes $\Omega(\min(|L_F(x)|, |L_B(y)|))$ time.

Denote the output of equation (1) by $\hat{d}$. Clearly, we have $\hat{d} \geq \text{dist}(x, y)$ when y is reachable from x. The *additive stretch* of $\hat{d}$ is given by $\hat{d} - \text{dist}(x, y)$, and the *multiplicative stretch* is given by $\hat{d}/\text{dist}(x, y)$. When there are no errors for any pairs of vertices, such landmark sets are called *2-hop covers* [17].

There is a more general family of data structures known as labeling schemes [26], which associates a vector $\mathcal{L} : V \to \{0, 1\}^*$ for every vertex. When answering a query for a pair of vertices $x, y \in V$, only the two labels $\mathcal{L}(x)$ and $\mathcal{L}(y)$ are required without accessing the graph. The total length of $\mathcal{L}$ is given by $\sum_{x \in V} |\mathcal{L}(x)|$. It is clear from equation (1) that landmark sketches fall in the query model of labeling schemes.

2.3 The Pruned Labeling Algorithm

We review the pruned labeling algorithm [3] for readers who are not familiar. The algorithm starts with an ordering of the vertices, $\{x_1, x_2, \ldots, x_n\}$. First for x_1, a breadth first search (BFS) is performed over the entire graph. During the BFS, x_1 is added to the landmark set of every vertex.[1] Next for x_2, in addition to running BFS, a *pruning* step is performed before adding x_2 as a landmark. For example, suppose that a path of length l is found from x_2 to y. If x_1 lies on the shortest path fom x_2 to y, then by checking their landmark sets, we can find the common landmark x_1 to certify that $\text{dist}(x_2, y) = \text{dist}(x_2, x_1) + \text{dist}(x_1, y) \leq l$. In this case, x_2 is not added to y's landmark set, and the neighbors of y are pruned away. The above procedure is repeated on x_3, x_4, etc.

For completeness, we describe the pseudocode in Algorithm **??**. Note that the backward BFS procedure can be derived similarly. It has been shown that the pruned labeling algorithm is guaranteed to return exact distances [3].

Algorithm 1 PrunedLabeling (Akiba et al. [3])

Input: A directed graph $G = (V, E)$; An ordering of V, $\{x_1, x_2, \ldots, x_n\}$.

Let $O = \emptyset$, and $L_F(x) = L_B(x) = \emptyset$, for all $x \in V$
for $i = 1, \ldots, n$ **do**
    forwardBfs(x_i)
    backwardBfs(x_i)
    $O = O \cup \{x_i\}$
end for

procedure forwardBfs(x_i)
    Let Q be a priority queue and S be a hash set
    Set the priority of x_i to be zero
    while $Q \neq \emptyset$ **do**
        Let l be the minimum priority of Q
        Let u be the corresponding vertex
        $S = S \cup \{u\}$
        Let $\tilde{d} = \min_{y \in L_F(x_i) \cap L_B(u)} \text{dist}(x_i, y) + \text{dist}(y, u)$
        if $l < \tilde{d}$ **then** (otherwise u's neighbors are pruned)
            $L_B(u) = L_B(u) \cup \{x_i \to l\}$
            for $y \in \mathcal{N}_{out}(u)$ such that $y \notin O$ **do**
                Let q be the priority of y
                if $y \notin S$ and $l + 1 < q$ **then**
                    Decrease y's priority to $l + 1$
                end if
            end for
        end if
    end while
end procedure

2.4 Random Graphs

We review the basics of Erdös-Rényi random graphs. Let $G = G(n, p)$ be an undirected graph where every edge appears with probability p. It is well known that when $p \geq 2(\log n)/n$, G has only one connected component with high probability. Moreover, the neighborhood growth rate (i.e. $|\Gamma_{i+1}(x)| \,/\, |\Gamma_i(x)|$) is highly concentrated around its expectation, which is np. Formally, the following facts are well-known.

Fact 1 (Bollobás [10]). *Let $G = (V, E)$ be an undirected graph where every edge is sampled with probability $p = 2(\log n)/n$. Let $D = \lceil \frac{\log n}{\log(np)} \rceil$. Then the following are true with high probability:*

a) *The diameter of G is at most $D + 1$;*
b) *For any $x, y \in V$ and $l \leq D$, $\Pr(\text{dist}(x, y) \leq l) \leq \frac{(np)^{l+1}}{n(np-1)}$;*
c) *For any $x \in V$ and $l < D$, we have $\frac{1}{2} \leq \frac{|\Gamma_l(x)|}{(np)^l} \leq 2$.*

The Chung-Lu model: Let $p_x > 0$ denote a weight value for every vertex $x \in V$. Given the weight vector $\mathbf{p}$, the Chung-Lu model generalizes Erdös-Rényi graphs such that each edge is chosen independently with probability

$$\Pr[x \sim y] = \min\left\{\frac{p_x \cdot p_y}{\text{vol}(V)}, 1\right\}, \forall x, y \in V$$

[1] For directed graphs, there will be a forward BFS which looks at x_1's outgoing edges and its descendants, as well as a backward BFS which looks at x_1's incoming edges and its predecessors.

where $\text{vol}(V) = \sum_{x \in V} p_x$ denote the volume of V.

Random power law graphs: Let $f : [x_{\min}, \infty) \to \mathbb{R}$ denote the probability density function of a power law distribution with exponent $\beta > 1$, i.e. $f(x) = Zx^{-\beta}$, where $Z = (\beta - 1) \cdot x_{\min}^{\beta-1}$ [16]. The expectation of $f(\cdot)$ exists when $\beta > 2$. The second moment is finite when $\beta > 3$. When $\beta < 3$, the expectation is finite, but the empirical second moment grows polynomially in the number of samples with high probability. If $\beta < 2$, then even the expectation becomes unbounded as n grows.

In a random power law graph, the weight of each vertex is drawn independently from the power law distribution. Given the weight vector $\mathbf{p}$, a random graph is sampled according to the Chung-Lu model. If the average degree $\nu > 1$, then it is known that almost surely the graph has a unique giant component [15].

2.5 Related Work

Landmark based labelings: There is a rich history of study on how to preprocess a graph to answer shortest path queries [2, 4, 5, 11, 17, 37]. It is beyond our scope to give a comprehensive review of the literature and we refer the reader to survey [36] for references.

In general, it is NP-hard to compute the optimal landmark based labeling (or 2-hop cover). Based on an LP relaxation, a $\log n$ factor approximation can be obtained via a greedy algorithm [18]. See also the references [6, 7, 12, 21, 28] for a line of followup work. The current state of the art is achieved based on the pruned labeling algorithm [3, 20]. Apart from the basic version that we have already presented, bit-parallel optimizations have been used to speed up proprocessing [3]. Variants which can be executed when the graph does not fit in memory have also been studied [30]. It is conceivable that such techniques can be added on top of the algorithms that we study as well. For the purpose of this work, we will focus on the basic version of the pruned labeling algorithm. Compared to classical approaches such as distance oracles, the novelty of the pruned labeling algorithm is using the idea of *pruning* to reduce redundancy in the BFS tree.

Network models: Earlier work on random graphs focus on modeling the small world phenomenon [15], and show that the average distance grows logarithmically in the number of vertices. Recent work have enriched random graph models with more realistic features, e.g. community structures [31], shrinking diameters in temporal graphs [32].

Other existing mathematical models on special families of graphs related to distance queries include road networks [1], planar graphs and graphs with doubling dimension. However none of them can capture the expansion properties that have been observed on subnetworks of real-world social networks [34].

Previous work of Chen et al. [13] presented a 3-approximate labeling scheme requiring storage $\tilde{O}(n^{(\beta-2)/(2\beta-3)})$ per vertex, on random power law graphs with $2 < \beta < 3$. Our (+2)-stretch result improves upon this scheme in the amount of storage needed per vertex for $2 < \beta < 2.5$, with a much better stretch guarantee. Another related line of work considers compact routing schemes on random graphs. Enachescu et al. [24] presented a 2-approximate compact routing scheme using space $O(n^{1.75})$ on Erdös-Rényi random graphs, and Gavoille et al. [25] obtained a 5-approximate compact routing scheme on random power law graphs.

3 OUR APPROACH

In order to motivate the idea behind our approach, we begin with an analysis of the pruned labeling algorithm on Erdös-Rényi random graphs. While the structures of real world graphs are far from Erdös-Rényi graphs, the intuition obtained from the analysis will be useful. Below we describe a simple proposition which states that for sparse Erdös-Rényi graphs, the pruned labeling algorithm outputs $\tilde{\Omega}(n^2)$ landmarks.

Proposition 2. *Let $G = (V, E)$ be an undirected Erdös-Rényi graph where every edge appears with probability $p = 2(\log n)/n$. For any ordering of the vertices $V = \{x_1, x_2, \ldots, x_n\}$, with high probability over the randomness of G, the total number of landmarks produced by Algorithm **??** is at least $\tilde{\Omega}(n^2)$.*

Proof sketch. We first introduce a few notations. Let $r = np$ denote the growth rate of G. Let $\varepsilon = 1/\log n$. Consider a vertex x_i where $1 \le i \le \varepsilon n$. Denote by $X_{-i} = \{x_1, \ldots, x_{i-1}\}$. Consider any $u \in V$, if none of the shortest paths from x_i to u intersect with X_{-i}, then (x_i, u) is called a *bad* pair. Note that x_i must be added to u's landmark set by Algorithm **??**, because during the BFS from x_i, all estimates through X_{-i} will be strictly larger than $\text{dist}(x_i, u)$. Hence, to lower bound the total landmark sizes, it suffices to count the number of *bad* pairs. In the following, we show that in expectation for every x_i where $1 \le i \le \varepsilon n$, there are at least $n/(\log n)^3$ vertices u such that (x_i, u) are *bad*. It follows that Algorithm **??** requires at least $\varepsilon n^2/(\log n)^3 \ge \tilde{\Omega}(n^2)$ in expectation.

Let $D = \lfloor \log_r n - 2 \rfloor$. Consider $\Gamma_D(x_i)$, the set of vertices at distance equal to D from x_i. We count the number of *bad* vertices in $\Gamma_D(x_i)$ at follows. For each $1 \le k \le D$, consider the intersection $\Gamma_k(x_i) \cap X_{-i}$ and their subtree down to $\Gamma_D(x_i)$.

Starting from any $y \in \Gamma_k(x_i) \cap X_{-i}$, the subtree of y would result in *good* vertices in $\Gamma_D(x_i)$, whose distance from x_i can be correctly estimated (c.f. line 15-16 in Algorithm **??**). In expectation, the size of the intersection is $r^k\varepsilon$, because the probability that any two vertex has distance k on G is equal to r^k/n, and there are at most εn vertices in X_{-i}. Next, each y results in r^{D-k} vertices in its $(D-k)$-th level neighborhood. Combined together, the total number of *good* vertices which are covered by $\Gamma_k(x_i) \cap X_{-i}$ is at most $r^k\varepsilon \times r^{D-k} = \varepsilon r^D$. By summing over all $k \le D$, we obtain that the total number of *good* vertices in $\Gamma_D(x_i)$ is at most $D\varepsilon r^D$.

On the other hand, the size of $\Gamma_D(x_i)$ is r^D. Hence the total number of *bad* vertices is at least $(1 - D\varepsilon)r^D \ge n/\log^3 n$. To show that the the proposition holds with high probability, it suffices to apply concentration results on neighborhood growth in the arguments above. We omit the details.

□

The interesting point from the above analysis is that $\Theta(n)$ landmarks are added throughout the first εn vertices. The reason is that there are no high degree vertices in Erdös-Rényi graphs, hence the landmarks we have added in the beginning do not cover the shortest paths for many vertex pairs later.Secondly, a large number of distant vertices are added in the landmark sets, which do not lie on the shortest paths of many pairs of vertices.

Motivated by the observation, we introduce our approach as follows. We start with an ordering of the vertices. For the top H

vertices in the ordering, we perform the same BFS procedure with pruning. For the rest of the vertices, we simply grow a local ball up to a desired radius. Concretely, only the vertices from the local ball will be used as a landmark. Algorithm **??** describes our approach in full.[2] As a remark, when the input graph G is undirected, it suffices to run one of the forward or backward BFS procedures, and for each vertex, its forward and backward landmark sets can be combined to a single landmark set.

Algorithm 2 ApproximatePruning

Input: A directed graph $G = (V, E)$; An ordering of V $\{x_1, x_2, \ldots, x_n\}$; The number of global landmarks H; The set of radiuses $\{l_i\}_{i=H+1}^{n}$.

Let $O = \emptyset$, and $L_F(x) = L_B(x) = \emptyset$, for any $x \in V$.
for $i = 1, \ldots, H$ **do**
  forwardBfs(x_i)
  backwardBfs(x_i)
  $O = O \cup \{x_i\}$
end for
for $i = H + 1, \ldots, n$ **do**
  localForwardBfs(x_i, l_i)
  localBackwardBfs(x_i, l_i)
end for
procedure localForwardBfs(x_i, l_i)
  for y such that $\text{dist}(x_i, y) \le l_i - 1$ **do**
    $L_F(x_i) = L_F(x_i) \cup (y \to \text{dist}(x_i, y))$
  end for
  for y such that $\text{dist}(x_i, y) = l_i$ and $\exists z$ s.t. $\text{dist}(x, z) = l_i - 1$, $(z, y) \in E$, $d_{out}(z) \le d_{out}(y)$ **do**
    $L_F(x_i) = L_F(x_i) \cup (y \to \text{dist}(x_i, y))$
  end for
end procedure

Recall that the backward and forward BFS procedures do a breadth first search with a pruning step before enqueing a vertex (c.f. Algorithm **??**). For each x_i with $i > H$, the parameter l_i controls the depth of the local ball we grow from x_i. Furthermore, at the bottom layer, we only add vertices whose outdegree is higher than any of its predecessor to x_i's landmark set. The intuition is that vertices with higher outdegrees are more likely to be used as landmarks.

We begin by analyzing Algorithm **??** for Erdös-Rényi graphs, as a comparison to Proposition 2. The following proposition shows that without using global landmarks, local balls of suitable radius suffice to cover all the desired distances. The proof is by observing that for each vertex, if we add the closest $\sqrt{n}$ vertices to the landmark set of every vertex, then the landmark sets of every pair of vertices will intersect with high probability, i.e. we have obtained a 2-hop cover.

Proposition 3. *Let $G = (V, E)$ be an undirected random graph where each edge is sampled with probability $p = 2(\log n)/n$. By setting $H = 0$ and $l_i = l = \lceil \frac{\log n}{2 \log np} \rceil + 1$ for all $1 \le i \le n$, we have that Algorithm **??** outputs a 2-hop cover with at most $\tilde{O}(n^{3/2})$ landmarks with high probability.*

Proof. Denote by $L(x)$ the landmark set obtained by Algorithm **??**, for every $x \in V$. We will show that with high probability:

a) For all $x_i, x_j \in V$, $L(x_i) \cap L(x_j) \neq \emptyset$. This implies that $L(\cdot)$ is a 2-hop cover.
b) The size of $L(x_i)$ is less than $\tilde{O}(\sqrt{n})$, for all $x_i \in V$.

Claim a) follows because the diameter of G is at most $2l - 1$ with high probability by Fact 1. Note that $L(x_i)$ contains $N_{l-1}(x_i)$, the set of vertices with distance at most $l - 1$. If $\text{dist}(x_i, x_j) \le (l - 1) + (l - 1)$, $N_{l-1}(x_i)$ and $N_{l-1}(x_j)$ already intersect. Otherwise, since the diameter is at most $2l - 1$, these two neighborhoods must be connected by an edge e. Suppose between e's two endpoints, the one with a lower degree is on x_i's side, then the local BFS from x_i must add the other endpoint to $L(x_i)$, and vice versa. Therefore, $L(x_i)$ must intersect with $L(x_j)$.

Claim b) is because $L(x_i)$ is a subset of $N_l(x_i)$. By Fact 1, the size of $N_l(x_i)$ is at most $4(np)^l \lesssim \tilde{O}(\sqrt{n})$. Hence, the proof is complete. □

4 RANDOM POWER LAW GRAPHS

In this section we analyze our algorithm on random power law graphs. We begin with the simple case of $\beta > 3$, which generalizes the result on Erdös-Rényi graphs. Because the technical intuition is the same with Proposition 3, we describe the result below and omit the proof.

Proposition 4. *Let $G = (V, E)$ be a random power law graph with average degree $v > 1$ and power law exponent $\beta > 3$. For each $x_i \in V$, let l_i be the smallest integer such that the number of edges between $N_{l_i}(x_i)$ and $V \backslash N_{l_i}(x_i)$ is at least $\delta\sqrt{n}$, where $\delta = 5\sqrt{v \log n}$.*

*By setting $H = 0$ and $\{l_i\}_{i=1}^{n}$, Algorithm **??** outputs a 2-hop cover with high probability. Moreover, each vertex uses at most $O(\sqrt{n} \log^2 n)$ landmarks.*

Remark: The high level intuition behind our algorithmic result is that as long as the breadth-first search process of the graph grows neither too fast nor too slow, but rather at a proper rate, then an efficient distance labeling scheme can be obtained. Proposition 4 can be easily extended to configuration models with bounded degree variance. It would be interesting to see if our results extend to preferential attachment graphs and Kronecker graphs.

The case of $2 < \beta < 3$: Next we describe the more interesting case with power law exponent $2 < \beta < 3$. Here the graph contains a large number of high degree vertices. By utilizing the high degree vertices, we show how to obtain exact distance landmark schemes, (+1)-stretch schemes and (+2)-stretch schemes. The number of landmarks used varies depending on the value of β. We now state our main result as follows.

Theorem 5. *Let $G = (V, E)$ be a random power law graph with average degree $v > 1$ and exponent $2 < \beta < 3$. Let*

$$K = \begin{cases} \sqrt{n}, \text{for } 2.5 \le \beta \le 3 \\ n^{\frac{1}{(4-\beta)(\beta-1)}}, \text{for } 2 < \beta < 2.5. \end{cases}$$

Let H be the number of vertices whose degree is at least K in G. Let $\pi = \{x_i\}_{i=1}^{n}$ be any ordering of vertices V by their degrees in a non-increasing order. For each vertex $x_i \in V$, let l_i be the smallest integer such that the number of edges between $N_{l_i-1}(x_i)$ and $V \backslash N_{l_i-1}(x_i)$ is at least $\delta n^{(\beta-2)/(\beta-1)}$, where $\delta = 4v \cdot \log^2 n$.

*With ordering π, parameters H and $\{l_i\}_{i=H+1}^{n}$, Algorithm **??** outputs a 2-hop cover with high probability. Moreover, the maximum*

[2] Here we have omitted the details of the local backward BFS procedure, which can be derived similar to the local forward BFS procedure.

number of landmarks used by any vertex is at most

$$O\left(\max\left(n^{\frac{\beta-2}{\beta-1}}, n^{\frac{3-\beta}{4-\beta}}\right)\log^3 n\right).$$

The above theorem says that in Algorithm **??**, first we use vertices whose degrees are at least K as global landmarks. Then for the other vertices x_i, we grow a local ball of radius l_i, whose size is (right) above $n^{(\beta-2)/(\beta-1)}$. The two steps together lead to a 2-hop cover. We now build up the intuition for the proof.

Building up a +1-stretch scheme: First, it is not hard to show that G contains a heavy vertex whose degree is $n^{1/(\beta-1)}$, by analyzing the power law distribution. Note that $K \leq n^{1/(\beta-1)}$, hence we have added all such high degree vertices as global landmarks. This part, together with the local balls, already gives us a (+1)-stretch landmark scheme.

To see why, consider two vertices x_i, x_j. If their local balls (of size $n^{(\beta-2)/(\beta-1)}$) already intersect, then we can already compute their distances correctly from their landmark sets. Otherwise, since the bottom layers of x_i and x_j already have weight/degree $n^{(\beta-2)/(\beta-1)}$, they are at most two hops apart, by connecting to the heavy vertex with degree $n^{1/(\beta-1)}$. Recall that the heavy vertex is added to the landmark sets of every vertex. Hence, the estimated distance is at most off by one. As a remark, to get the (+1)-stretch landmark scheme, the number of landmarks needed per vertex is on the order of $n^{(\beta-2)/(\beta-1)}$. This is because we only need to use vertices whose degrees are at least $n^{1/(\beta-1)}$ as global landmarks (there are only $\log n$ of them), as opposed to H in Theorem 5.

Fixing the +1-stretch: To obtain exact distances, for each vertex on the boundary of radius $l_i - 1$, we add all of its neighbors with a higher degree to the landmark set (c.f. line 15-17 in Algorithm **??**). Whenever there is an edge connecting the two boundaries, the side with a lower degree will add the other endpoint as a landmark, which resolves the (+1)-stretch issue. For the size of landmark sets, it turns out that fixing the (+1)-stretch for the case $2 < \beta < 2.5$ significantly increases the number of landmarks needed. Specifically, the costs are $n^{(5-\beta)/(4-\beta)}$ landmarks per node.

Intuition for the +2-stretch scheme: As an additional remark, one can also obtain a (+2)-stretch landmark sketch by setting l_i in Algorithm **??** in a way such that every vertex stores the closest $\tilde{\Theta}(n^{(\beta/2)-1})$ vertices in its landmark set. This modification leads to a (+2)-stretch scheme, because for two vertices x, y, once the bottom layers of x, y have size at least $\tilde{\Theta}(n^{(\beta/2)-1})$, they are at most three hops away from each other. The reason is that with high probability, the bottom layer will connect to a vertex with weight $\Omega(\sqrt{n})$ in the next layer (which will all be connected), as it is not hard to verify that the volume of all vertices with weight $\sqrt{n}$ is $\Omega(n^{(4-\beta)/2})$. By a similar proof to Theorem 5, the maximum number of landmarks used per vertex is at most $\tilde{O}(n^{(\beta-2)/2})$.

We refer the reader to the full version for details of the full proof. The technical components involve carefully controlling the growth of the neighborhood sets by using concentration inequalities.

5 EXPERIMENTS

In this section, we substantiate our results with experiments on a diverse collection of network datasets. A summary of the findings are as follows. We first compare Algorithm **??** with the pruned labeling algorithm [3]. Recall that our approach differs from the pruned labeling algorithm by only performing a thorough BFS for a small set of vertices, while running a lightweight local ball growing procedure for most vertices. We found that this simple modification leads to 1.5-2.5x reduction in number of landmarks stored. The preprocessing time is reduced by 2-15x as well. While our algorithm does not always output the exact distance like the pruned labeling algorithm, we observe that the stretch is at most 1%, relative to the average distance.

Next we compare our approach to two approximate distance sketches with strong theoretical guarantees, Das Sarma et al. sketch [8, 19] and a variant of Thorup-Zwick's 3-approximate distance oracle [37], which uses high degree vertices as global landmarks [13]. We observe that our approach incurs lower stretch and requires less space compared to Das Sarma et al. sketch. The accuracy of Thorup-Zwick sketch is comparable to ours, but we require much fewer landmarks.

5.1 Experimental Setup

To ensure the robustness of our results, we measure performances on a diverse collection of directed and undirected graphs, with the datasets coming from different domains, as described by Table 1. Stanford, Google and BerkStan are all Web graphs in which edges are directed. DBLP (collaboration network) and Youtube (friendship network) are both undirected graphs where there is one connected component for the whole graph. Twitter is a directed social network graph with about 84% vertices inside the largest strongly connected component. All the datasets are downloaded from the Stanford Large Network Dataset Collection [33].

Table 1: Datasets used in experiments.

graph	# nodes	# edges	category	type
DBLP	317K	1.0M	Collaboration	Undirected
Twitter	81K	1.8M	Social	Directed
Stanford	282K	2.3M	Web	Directed
Youtube	1.1M	3.0M	Social	Undirected
Google	876K	5.1M	Web	Directed
BerkStan	685K	7.6M	Web	Directed

Implementation details: We implemented all four algorithms in Scala, based on a publicly available graph library.[3] The experiments are conducted on a 2.30GHz 64-core Intel(R) Xeon(R) CPU E5-2698 v3 processor, 40MB cache, 756 GB of RAM. Each experiment is run on a single core and loads the graph into memory before beginning any timings. The RAM used by the experiment is largely dominated by the storage needed for the landmark sets.

Parameters: In the comparison between the pruned labeling algorithm and our approach, we order the vertices in decreasing order by the indegree plus outdegree of each vertex.[4] Recall that there are

[3] https://github.com/teapot-co/tempest

[4] There are more sophisticated techniques such as ordering vertices using their betweenness centrality scores [20]. It is conceivable that our algorithm can be combined with such techniques.

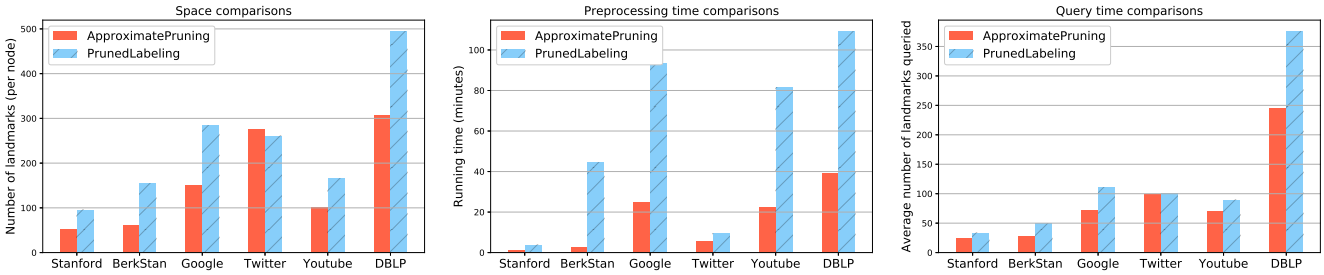

Figure 2: Comparing the efficiency of our approach to the pruned labeling algorithm.

Table 2: Measuring the accuracy of our approach.

	Stanford	BerkStan	Google	Twitter	Youtube	DBLP
Relative Average Stretch	0.37%	0.20%	0.51%	0.29%	0.33%	1.1%
Maximum Relative Stretch	21/10	10/7	8/5	4/3	4/3	7/5
Average Additive Stretch	0.046	0.030	0.060	0.014	0.018	0.075
Maximum Additive Stretch	11	3	3	1	2	2
Average Distance	12.3	14.6	11.7	4.9	5.3	6.8

two input parameters used in our approach, the number of global landmarks H and the radiuses of local balls $\{l_i\}_{i=H+1}^{n}$. To tune H, we start with 100, then keep doubling H to be 200, 400, etc. The radiuses $\{l_i\}_{i \geq H}$ are set to be 2 for all graphs.[5]

Benchmarks: For the Thorup-Zwick sketch, in the first step, $H = \sqrt{n}$ vertices are sampled uniformly at random as global landmarks. In the second step, every other vertex grows a local ball as its landmark set until it hits any of the $\sqrt{n}$ vertices. All vertices within the ball are used as landmarks. This method uses $O(n^{3/2})$ landmarks and achieves 3-stretch in worst case. In the follow up work of Chen et al. [13], the authors show a variant which uses high degree vertices as global landmarks and observe better performance. We implement Chen et al.'s variant in our experiment, and use the H vertices with the highest indegree plus outdegree as global landmarks. In the experiment, we start with H equal to $\sqrt{n}$. Then we report results for $\sqrt{n}$ multiplied by $\{2, 1/2, 1/4, 1/8\}$.

For the Das Sarma et al. sketch, first, $\log n$ sets S_i of different sizes are sampled uniformly from the set of vertices V, for $0 \leq i < \log n$, where the size of S_i is 2^i. Then a breadth first search is performed from S_i, so that every vertex $x \notin S_i$ finds its closest vertex inside S_i in graph distance. This closest vertex is then used as a landmark for x. The number of landmarks used in Dar Sarma's sketch is $n \log n$, and the worst case multiplicative stretch is $\log n$. If more accurate estimation is desired, one can repeat the same procedure multiple times and union the landmark sets together. We begin with 5 repetitions, then keep doubling it to be 10, 20 etc.

Our approach differs from the above two methods by using the idea of pruning while running BFS. This dramatically enhances performance in practice, as we shall see in our experiments.

Metrics: We measure the stretch of the estimated distances, and compute aggregated statistics over a large number of queries. For a query (x, y), if y is reachable from x, but the algorithm reports no common landmark between the landmark sets of x and y, then we count such a mistake as a "False disconnect error." On the other hand, if y is not reachable from x, then it is not hard to see that our algorithm always reports correctly that y is not reachable from x. In the experiments, we compute $\text{dist}(x, y)$ using Dijkstra's algorithm.

To measure space usage, we report the number of landmarks per node used in each algorithm as a proxy. Since the landmark sets are stored in Int to Float hash maps, the actual space usage would be eight bytes times the landmark sizes in runtime, with a constant factor overhead.

For the query time, recall that for each pair of vertices (x, y), we estimate their distance by looking at the intersection of $L_F(x)$ and $L_B(y)$ and compute the minimum interconnecting distance (c.f. equation 1). To find the minimum, we iterate through the smaller landmark set. Hence the running time is $\min(|L_F(x)|, |L_B(y)|)$ multiplied by the time for a hash map lookup, which is a small fixed value in runtime. A special case is when $y \in L_F(x)$ or $x \in L_B(y)$, where only one hash map lookup is needed. We will report the number of hash map lookups as a proxy for the query time.[6]

5.2 Comparisons to Exact Methods

We report the results comparing our approach to the pruned labeling algorithm. The pruned labeling algorithm is exact. To measure the accuracy of our approach, we randomly sample 2000 pairs of source and destination vertices. The number of global landmarks is set to be 400 for the Stanford dataset, 1600 for the DBLP dataset, and 800 for the rest of the datasets.

Figure 2 shows the preprocessing time, the number of landmarks and average query time used by both algorithms. We see that our approach reduces the number of landmarks used by 1.5-2.5x, except

[5]It follows from our theoretical analysis that the radiuses should be less than half of the average distance. As a rule of thumb, setting the radius as 2 works based on our experiments.

[6]It is conceivable that more sophisticated techniques may be devised to speedup set intersection. We leave the question for future work.

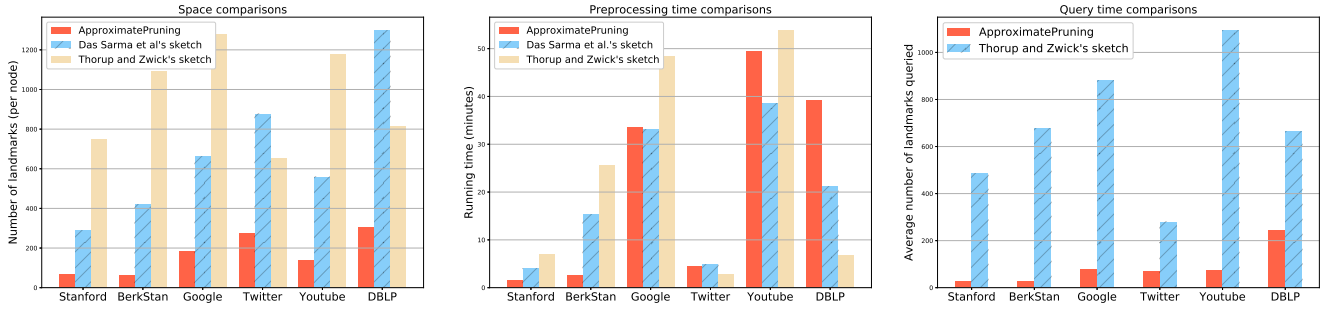

Figure 3: Comparing the efficiency of our approach to two well known distance sketches with strong theoretical guarantees.

Table 3: Measuring the stretch for all three methods.

	Stanford	BerkStan	Google	Twitter	Youtube	DBLP
DS et al. sketch	20.9%	17.2%	21.1%	11.1%	13.1%	5.4%
TZ sketch	0.30%	0.21%	0.36%	1.65%	0.03%	2.15%
Our approach	0.16%	0.20%	0.22%	0.29%	0.04%	1.1%

Table 4: Varying H in TZ sketch.

Youtube	$\sqrt{n}/2$	$\sqrt{n}/4$	$\sqrt{n}/8$	Ours
Stretch	0.04%	0.11%	0.07%	0.04%
# Landmarks	731	648	811	137
Preprocessing	37m	31m	35m	50m

on the Twitter dataset.[7] Our approach performs favorably in terms of preprocessing time and query time as well.

The accuracy of our computed estimate is shown in Table 2. We have also measured the median additive stretch, which turns out to be zero in all the experiments. To get a more concrete sense of the accuracy measures, consider the Google dataset as an example. Since the average additive stretch is 0.06 and there are 2000 pairs of vertices, the total additive stretch is at most 120 summing over all 2000 pairs! Specifically, there can be at most 120 queries with non-zero additive stretch and for all the other queries, our approach returns the exact answer. Meanwhile, among all the datasets, we observed only one "False disconnect error" in total. It appeared in the Stanford Web graph experiment, where the true distance is 80.

5.3 Comparisons to Approximate Methods

Next we compare our approach to Das Sarma et al.'s sketch (or DS et al. sketch in short) and the variant of Thorup and Zwick's sketch (or TZ sketch in short). Similar to the previous experiment, we sample 2000 source and destination vertices uniformly at random to measure the accuracy.

We start by setting the number of global landmarks to $\sqrt{n}$ in Thorup-Zwick sketch. To allow for a fair comparison, we tune our approach so that the relative average stretch is comparable or lower. Specifically, the Stanford, BerkStan and Twitter datasets use $H = 800$, the Google and DBLP datasets use $H = 1600$ and the Youtube dataset uses $H = 3200$.

Figure 3 shows the number of landmarks needed in each algorithm as well as the amount of preprocessing time consumed. Overall, our approach uses much fewer landmarks than the other two algorithms. In terms of preprocessing time, our approach is comparable or faster on all datasets, except on the DBLP network. We suspect that this may be because the degree distribution of the DBLP network is flatter than the others. Hence performing the pruning procedures on a small subset of high degree vertices are less effective in such a scenario.

We next report the relative average stretch for all three methods. As can be seen in Table 3, our approach is comparable to or slightly better than Thorup and Zwick's sketch, but much more accurate than Das Sarma et al's sketch. Note that the latter performed significantly worse than the other two approaches. We suspect that this may be because the sketch does not utilize the high degree vertices efficiently. Lastly, our approach performs favorably in the query time comparison as well. Note that the query time of Das Sarma et al.'s sketch are not reported because of the worse accuracy.

Effect of parameter choice: Note that in the above experiment, for Thorup and Zwick's sketch, we have set the number of global landmarks H to be $\sqrt{n}$. In the next experiment, we vary the value of H to $\sqrt{n}$ multiplied by $\{2, 1/2, 1/4, 1/8\}$.

First, we report a detailed comparison on the Google dataset in Figure 4. Note that when $H = 2\sqrt{n}$, the Thorup and Zwick's sketch requires over 2000 landmarks per node which is significantly larger than the other values. Hence, we dropped the data point from the plot. For our approach, we double H from 100 up to 1600. Overall, we can see that our approach requires fewer landmarks across different stretch levels.

Next, we report brief results on the Youtube dataset in Table 4 since the results are similar. The conclusions obtained from other datasets are qualitatively similar, and hence omitted.

[7] By setting the radiuses $\{l_i\}$ to be 1, we incur 0.72% relative additive stretch by using 173 landmarks per node, which improves over the pruned labeling algorithm by 1.5x.

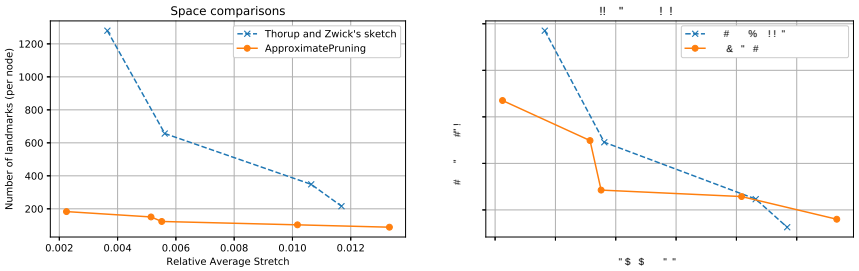

Figure 4: Varying H in TZ sketch and our approach, on the Google dataset.

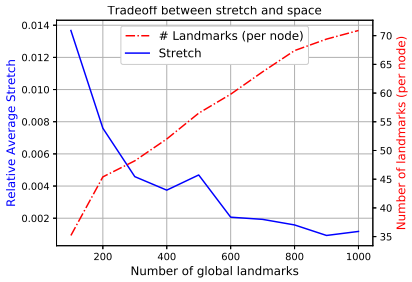

Figure 5: Tradeoff on the Stanford dataset.

5.4 More Experimental Observations

By varying the number of global landmarks used Algorithm **??**, it is possible to obtain a smooth tradeoff between stretch and number of landmarks used. As an example, we present the tradeoff curve for the Stanford Web dataset in Figure 5. Here we vary the number of global landmarks used from 100 to 1000. As one would expect, the relative average stretch decreases while the number of landmarks stored increases.

6 FUNDAMENTAL LIMITS OF LANDMARK SKETCHES

This section complements our algorithm with lower bounds. We begin by showing a matching lower bound for Erdös-Rényi graphs, saying that any 2-hop cover needs to store at least $\tilde{\Omega}(n^{3/2})$ landmarks. The results imply that the parameter dependence on n of our algorithm is tight for Erdös-Rényi graphs and random power law graphs with power law exponent $\beta > 3$. It is worth mentioning that the results not only apply to landmark sketches, but also work for the family of labeling schemes. Recall that labeling schemes associate a labeling vector for each vertex. To answer a query for a pair of vertices (x, y), only the labeling vectors of x, y are accessed. We first state the lower bound for Erdös-Rényi graphs.

Theorem 6. *Let $G = (V, E)$ be an Erdös-Rényi graph where every edge is sampled with probability $p = 2\log n/n$. With high probability over the randomness of G, any labelings which can recover all pairs distances exactly have total length at least $\Omega(n^{3/2}/\log^4 n)$.*

In particular, any 2-hop cover needs to store at least $\Omega(n^{3/2}/\log^4 n)$ many landmarks with high probability.

For a quick overview, we divide V into $\sqrt{n}$ sets of size $\sqrt{n}$ each. We will show that the total labeling length for each set of $\sqrt{n}$ vertices has to be at least $\tilde{\Omega}(n)$. By union bound over all the $\sqrt{n}$ sets, we obtain the desired conclusion. We now go into the proof details.

Proof. Denote by $r = np$. Let $d = \lfloor \frac{\log n}{2\log(np)} \rfloor - c$, where c is a fixed constant (e.g. $c = 2$ suffices). Divide V into groups of size $\sqrt{n}$. Clearly, there are $\sqrt{n}$ disjoint groups – let S be one of them. Denote by c_1 a fixed constant which will be defined later. We argue that

$$\Pr[\text{The total label length of } S \le c_1 \cdot r^{1-2c} n] \lesssim r^{1-2c}. \qquad (2)$$

Hence by Markov's inequality, with high probability except for $(\log n) r^{1-2c}\sqrt{n}$ groups, all the other groups will have label length at least $c_1 \cdot r^{1-2c} n \gtrsim \tilde{\Omega}(n)$, because $r \le 2\log n$. Hence we obtain the desired conclusion. For the rest of the proof, we focus on proving equation (2) for the group S.

Let $\{x_1, x_2, \ldots, x_{|S|}\}$ be an arbitrary ordering of S. We grow the neighborhood of each vertex in S one by one, up to level d. Denote by $G_1 = (V_1, E_1)$, where $V_1 = V$ and $E_1 = E$. For any $i \ge 1$, if $x_i \in V_i$, then we define define $T(x_i)$ to be the set of of vertices in V_i whose distance is at most d from x_i. Define $L(x_i) \subseteq T(x_i)$ to be the set of vertices in G_i whose distance is equal to d from x_i. On the other hand if $x_i \notin V_i$, then $T(x_i)$ and $L(x_i)$ are both empty. More formally,

$$T(x_i) := \begin{cases} \{y : \mathrm{dist}_{G_i}(x_i, y) \le d\}, & \text{if } x_i \in V_i \\ \varnothing, & \text{otherwise.} \end{cases}$$
$$L(x_i) := \{y \in T(x_i) : \mathrm{dist}_{G_i}(x_i, y) = d\}$$

We then define $F_i = \cup_{j=1}^{i} T(x_j)$. Denote by G_{i+1} to be the induced subgraph of G_i on the remaining vertices $V_{i+1} = V \backslash F_i$. We show that with high probability, a constant fraction of vertices $x_i \in S$ satisfy that $|L(x_i)| \ge \Omega((np)^d)$. □

Lemma 7 (Martingale inequality). *In the setting of Theorem 6, with high probability, at least $|S|/2$ vertices $x_i \in S$ satisfy that $|L(x_i)| \ge r^d/6$.*

Proof. For any $1 \le i \le |S|$, consider

$$X_i := \begin{cases} 1 & \text{if } x_i \notin V_i, \text{ or } |F_{i-1}| > |S| \cdot r^d \log n, \text{or } |L(x_i)| \ge r^d/6 \\ 0 & \text{otherwise.} \end{cases}$$

We claim that $\Pr[X_i = 1 \mid X_1, \ldots, X_{i-1}]$ with high probability. It suffices to consider the case $x_i \in V_i$ and $|F_{i-1}| \le |S| r^d \log n$. It is not hard to verify that $|F_{i-1}| \le n/\log n$ by our setting of d. Hence the size of V_i is at least $n(1 - 1/\log n)$. Note that the subgraph G_i is still an Erdös-Rényi random graph, and the number of vertices is at least $n(1 - 1/\log n)$. By Fact 1c), the size of $L(x_i)$ is at least

$$\frac{1}{2} r^d (1 - \log^{-1} n)^d \ge r^d/6,$$

since $d \le \log n$.

Thus by Azuma-Hoeffding inequality, $\sum_{i=1}^{|S|} X_i \ge 0.99\,|S|$ with high probability. We will show below that the contributions to $\sum_{i=1}^{|S|} X_i$ from $x_i \notin V_i$ and $|F_{i-1}| > |S| r^d \log n$ is less than $0.02\,|S|$. Hence by taking union bound, we obtain the desired conclusion.

First, we show that the number of x_i such that $x_i \notin V_i$ are at most $0.01|S|$ with high probability. Note that $x_i \notin V_i$ implies that

there exists some vertex x_j with $j < i$ such that $\text{dist}(x_i, x_j) \le d$. On the other hand, by Fact 1,

$$\Pr[\text{dist}(x, y) \le d] \le \frac{3r^d}{n}, \forall x, y \in S.$$

Hence, it is not hard to verify that the expected number of vertex pairs in S whose distance is at most d, is $O(|S|^2 r^{2d}/n) \lesssim |S| / \log n$, by the setting of d. By Markov's inequality, with probability $1 - 1/\log n$ only $0.01\,|S|$ vertex pairs have distance at most d in S. Hence there exists at most $0.01\,|S|$ i's such that $x_i \notin V_i$.

Secondly for all $1 \le i \le |S|$, the set of vertices T_i is a subset of $N_d(x_i)$, the set of vertices within distance d to x_i on G. By Fact 1c), the size of $N_d(x_i)$ is at most $2r^d$. Hence we have $|F_i| \le 2\,|S|\, r^d$ for all $1 \le i \le |S|$ with high probability. This proves the Lemma. □

Now we are ready to finish the proof. Given the labels of S, we can recover all pairwise distances in S. Let $\text{dist}_S : S \times S \to \mathbb{N}$ denote the distance function restricted to S. Consider the following:

a) $\exists\, |S|^2/9$ pairs (x_i, x_j) such that $\text{dist}_S(x_i, x_j) \le 2d + 1$. We know by Fact 1 that $\Pr[\text{dist}(x_i, x_j) \le 2d+1] \le 2r^{2d+1}/n$, for any $x_i, x_j \in S$. Hence the expected number of pairs with distance at most $2d+1$ in S, is at most $2\,|S|^2 \cdot r^{2d+1}/n \lesssim r^{1-2c}n$. By Markov's inequality, the probability that a random graph induces any such distance function is $r^{1-2c}n/(|S|^2/8) \lesssim r^{1-2c}$.

b) The number of pairs such that $\text{dist}_S(x_i, x_j) \le 2d + 1$ is at most $|S|^2/8$ in S. Let A denote

$$\left\{(x, y) \in S \times S \mid \text{dist}(x, y) > 2d + 1, \text{ and } |L(x)|, |L(y)| \ge r^d/6\right\}.$$

By Lemma 7 and our assumption for case b), the size of A is at least $\binom{|S|/2}{2} - |S|^2/8 \ge |S|^2/9$. For any $(x, y) \in A$, $L(x)$ and $L(y)$ are clearly disjoint. Note that the event whether there exists an edge between $L(x)$ and $L(y)$ is independent, conditional on revealing the subgraph for all $x \in S$ up to distance d. Hence

$$\begin{aligned}
&\Pr\left[\text{dist}_S(x, y) > 2d + 1,\ \forall (x, y) \in A\right] \\
&\le \prod_{(x,y)\in A} \Pr\left[\text{there is no edge between } L(x) \text{ and } L(y)\right] \\
&\le \prod_{(x,y)\in A} (1-p)^{|L(x)|\times|L(y)|} \\
&\le \exp\left(-p \times |A| \times r^{2d}/72\right) \qquad (\text{because } |L(x)|, |L(y)| \ge r^d/6) \\
&\le \exp(-c_1 r^{1-2c} n). \qquad (\text{because } |A| \ge n/9)
\end{aligned}$$

where $c_1 = 72 \times 9$ in the last line. Denote by $\kappa = c_1 \cdot r^{1-2c}n$. Note that the number of labelings of length (or number of bits) less than κ is at most 2^κ. For each labeling, the probability that it correctly gives all pairs distances is at most $\exp(-\kappa)$ by our argument above. Therefore by union bound, the probability that the total labeling length of $|S|$ is at most κ is at most $2^\kappa \cdot \exp(\kappa) \le r^{1-2c}$ for large enough n.

To recap, by combining case a) and b), we have shown that equation (2) is true. Hence the proof is complete.

Extensions to $\beta > 3$: It is worth mentioning that the lower bound on Erdös-Rényi graphs can be extended to random power law graphs with $\beta > 3$. The proof structure is similar because the degree distribution has finite variance, hence the number of high degree vertices is small. The difference corresponds to technical modifications which deal with the neighborhood growth of random graphs with constant average degree. We state the result below and leave the proof to the full version.

Theorem 8. *Let $G = (V, E)$ a random power law graph with average degree $\nu > 1$ and exponent $\beta > 3$. With high probability over the randomness of G, any labelings which can recover all pairs distances exactly have total length at least $\tilde{\Omega}(n^{3/2})$.*

In particular, any 2-hop cover needs to store at least $\tilde{\Omega}(n^{3/2})$ many landmarks with high probability.

Lower bounds for β close to 2: Next we show that the parameter dependence of our algorithm is tight when β is close to 2. Specifically, any 2-hop cover needs to store at least $\Omega(n^{3/2-\varepsilon})$ many landmarks when $\beta = 2 + \varepsilon$. Hence it is not possible to improve over our algorithm when β is close to 2. Furthermore, the lower bound holds for the general family of labeling schemes as well.

Theorem 9. *Let $G = (V, E)$ a random power law graph with average degree $\nu > 1$ and exponent $\beta = 2 + \varepsilon$ for $\varepsilon < 1/2$. With high probability over the randomness of G, any labelings which can recover all pairs distances exactly have total length at least $\Omega(n^{3/2-\varepsilon})$.*

In particular, any 2-hop cover needs to store at least $\Omega(n^{3/2-\varepsilon})$ many landmarks with high probability.

The proof is conceptually similar to Theorem 6, so we sketch the outline and leave the proof to the full version.

Let S_{high} be the set of vertices whose degrees are on the order of $\sqrt{n}$. Let S_{low} be a set of $\sqrt{n}$ vertices, where each vertex has weight between ν and 2ν. Such a set is guaranteed to exist because there are $\Theta(n)$ of them.

We first reveal all edges of G other than the ones between S_{high}. We show that at this stage, most vertices in S_{low} are more than 3 hops away from each other. If for some pair (x, y) in S_{low} whose distance is larger than three, and both x and y connect to exactly one (but different) vertex in S_{high}, then knowing whether $\text{dist}(x, y) = 3$ will reveal whether their neighbors in S_{high} are connected by an edge.

Based on the observation, we show that the total labeling length of S_{low} is at least $\tilde{\Omega}(n^{3-\beta})$. This is because the random bit between a vertex pair in S_{high} has entropy $\Omega(n^{2-\beta})$. Since there are $\Theta(n)$ pairs of vertices in S_{high}, the entropy of the labelings of S_{low} must be $\Omega(n^{3-\beta})$ (hence, its size must also be at least $\Omega(n^{3-\beta})$). Similar to Theorem 6, this argument is applied to $\sqrt{n}$ disjoint sets of "S_{low}", summing up to an overall lower bound of $\Omega(n^{7/2-\beta}) = \Omega(n^{3/2-\varepsilon})$.

7 CONCLUSIONS AND FUTURE WORK

In this work, we presented a pruning based landmark labeling algorithm. The algorithm is evaluated on a diverse collection of networks. It demonstrates improved performances compared to the baseline approaches. We also analyzed the algorithm on random power law graphs and Erdös-Rényi graphs. We showed upper and lower bounds on the number of landmarks used for Erdös-Rényi graphs and random power law graphs.

There are several possible directions for future work. One direction is to close the gap in our upper and lower bounds for random

power law graphs. We believe that any improved understanding can potentially lead to better algorithms for real world power law graphs as well. Another direction is to evaluate our approach on transportation networks, which correspond to another important domain in practice.

Acknowledgements. The authors would like to thank Fan Chung Graham, Tim Roughgarden, Amin Saberi and D. Sivakumar for useful feedback and suggestions at various stages of this work. Also, thanks to the anonymous referees for their constructive reviews. Hongyang Zhang is supported by NSF grant 1447697.

REFERENCES

[1] Ittai Abraham, Amos Fiat, Andrew V Goldberg, and Renato F Werneck. 2010. Highway dimension, shortest paths, and provably efficient algorithms. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 782–793.

[2] Ittai Abraham and Cyril Gavoille. 2011. On approximate distance labels and routing schemes with affine stretch. In *International Symposium on Distributed Computing*. Springer, 404–415.

[3] Takuya Akiba, Yoichi Iwata, and Yuichi Yoshida. 2013. Fast exact shortest-path distance queries on large networks by pruned landmark labeling. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*. ACM, 349–360.

[4] Stephen Alstrup, Søren Dahlgaard, Mathias Bæk Tejs Knudsen, and Ely Porat. 2015. Sublinear distance labeling. *arXiv preprint arXiv:1507.02618* (2015).

[5] Ingo Althöfer, Gautam Das, David Dobkin, Deborah Joseph, and José Soares. 1993. On sparse spanners of weighted graphs. *Discrete & Computational Geometry* 9, 1 (1993), 81–100.

[6] Haris Angelidakis, Yury Makarychev, and Vsevolod Oparin. 2017. Algorithmic and hardness results for the hub labeling problem. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 1442–1461.

[7] Maxim Babenko, Andrew V Goldberg, Haim Kaplan, Ruslan Savchenko, and Mathias Weller. 2015. On the complexity of hub labeling. In *International Symposium on Mathematical Foundations of Computer Science*. Springer, 62–74.

[8] Bahman Bahmani and Ashish Goel. 2012. Partitioned multi-indexing: bringing order to social search. In *Proceedings of the 21st international conference on World Wide Web*. ACM, 399–408.

[9] Hannah Bast, Daniel Delling, Andrew Goldberg, Matthias Müller-Hannemann, Thomas Pajor, Peter Sanders, Dorothea Wagner, and Renato F Werneck. 2016. Route planning in transportation networks. In *Algorithm engineering*. Springer, 19–80.

[10] Béla Bollobás. 1998. Random graphs. In *Modern Graph Theory*. Springer, 215–252.

[11] Michele Borassi, Pierluigi Crescenzi, and Luca Trevisan. 2017. An Axiomatic and an Average-Case Analysis of Algorithms and Heuristics for Metric Properties of Graphs. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 920–939.

[12] Michele Borassi, Pierluigi Crescenzi, and Luca Trevisan. 2017. An axiomatic and an average-case analysis of algorithms and heuristics for metric properties of graphs. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 920–939.

[13] Wei Chen, Christian Sommer, Shang-Hua Teng, and Yajun Wang. 2009. Compact routing in power-law graphs. In *International Symposium on Distributed Computing*. Springer, 379–391.

[14] Fan Chung and Linyuan Lu. 2002. The average distances in random graphs with given expected degrees. *Proceedings of the National Academy of Sciences* 99, 25 (2002), 15879–15882.

[15] Fan RK Chung and Linyuan Lu. 2006. *Complex graphs and networks*. Vol. 107. American mathematical society Providence.

[16] Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. 2009. Power-law distributions in empirical data. *SIAM review* 51, 4 (2009), 661–703.

[17] Edith Cohen. 1997. Size-estimation framework with applications to transitive closure and reachability. *J. Comput. System Sci.* 55, 3 (1997), 441–453.

[18] Edith Cohen, Eran Halperin, Haim Kaplan, and Uri Zwick. 2003. Reachability and distance queries via 2-hop labels. *SIAM J. Comput.* 32, 5 (2003), 1338–1355.

[19] Atish Das Sarma, Sreenivas Gollapudi, Marc Najork, and Rina Panigrahy. 2010. A sketch-based distance oracle for web-scale graphs. In *Proceedings of the third ACM international conference on Web search and data mining*. ACM, 401–410.

[20] Daniel Delling, Andrew V Goldberg, Thomas Pajor, and Renato F Werneck. 2014. Robust distance queries on massive networks. In *European Symposium on Algorithms*. Springer, 321–333.

[21] Daniel Delling, Andrew V Goldberg, Ruslan Savchenko, and Renato F Werneck. 2014. Hub labels: Theory and practice. In *International Symposium on Experimental Algorithms*. Springer, 259–270.

[22] Richard Durrett. 2007. *Random graph dynamics*. Vol. 200. Citeseer.

[23] Nicole Eikmeier and David F Gleich. 2017. Revisiting Power-law Distributions in Spectra of Real World Networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 817–826.

[24] Mihaela Enachescu, Mei Wang, and Ashish Goel. 2008. Reducing maximum stretch in compact routing. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*. IEEE.

[25] Cyril Gavoille, Christian Glacet, Nicolas Hanusse, and David Ilcinkas. 2015. Brief Announcement: Routing the Internet with very few entries. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing*. ACM, 33–35.

[26] Cyril Gavoille, David Peleg, Stéphane Pérennes, and Ran Raz. 2001. Distance labeling in graphs. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 210–219.

[27] Andrew V Goldberg and Chris Harrelson. 2005. Computing the shortest path: A search meets graph theory. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 156–165.

[28] Andrew V Goldberg, Ilya Razenshteyn, and Ruslan Savchenko. 2013. Separating hierarchical and general hub labelings. In *International Symposium on Mathematical Foundations of Computer Science*. Springer, 469–479.

[29] Hao He, Haixun Wang, Jun Yang, and Philip S Yu. 2007. BLINKS: ranked keyword searches on graphs. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*. ACM, 305–316.

[30] Minhao Jiang, Ada Wai-Chee Fu, Raymond Chi-Wing Wong, and Yanyan Xu. 2014. Hop doubling label indexing for point-to-point distance querying on scale-free networks. *Proceedings of the VLDB Endowment* 7, 12 (2014), 1203–1214.

[31] Tamara G Kolda, Ali Pinar, Todd Plantenga, and Comandur Seshadhri. 2014. A scalable generative graph model with community structure. *SIAM Journal on Scientific Computing* 36, 5 (2014), C424–C452.

[32] Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos, and Zoubin Ghahramani. 2010. Kronecker graphs: An approach to modeling networks. *Journal of Machine Learning Research* 11, Feb (2010), 985–1042.

[33] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data.

[34] Jure Leskovec, Kevin J Lang, Anirban Dasgupta, and Michael W Mahoney. 2008. Statistical properties of community structure in large social and information networks. In *Proceedings of the 17th international conference on World Wide Web*. ACM, 695–704.

[35] Michalis Potamias, Francesco Bonchi, Carlos Castillo, and Aristides Gionis. 2009. Fast shortest path distance estimation in large networks. In *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 867–876.

[36] Christian Sommer. 2014. Shortest-path queries in static networks. *ACM Computing Surveys (CSUR)* 46, 4 (2014), 45.

[37] Mikkel Thorup and Uri Zwick. 2005. Approximate distance oracles. *Journal of the ACM (JACM)* 52, 1 (2005), 1–24.

[38] Remco Van Der Hofstad. 2009. Random graphs and complex networks. *Available on http://www. win. tue. nl/rhofstad/NotesRGCN.pdf* (2009), 11.

[39] Monique V Vieira, Bruno M Fonseca, Rodrigo Damazio, Paulo B Golgher, Davi de Castro Reis, and Berthier Ribeiro-Neto. 2007. Efficient search ranking in social networks. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. ACM, 563–572.

[40] Sihem Amer Yahia, Michael Benedikt, Laks VS Lakshmanan, and Julia Stoyanovich. 2008. Efficient network aware search in collaborative tagging sites. *Proceedings of the VLDB Endowment* 1, 1 (2008), 710–721.