SAPIENZA UNIVERSITY OF ROME
DIPARTIMENTO DI INFORMATICA

# Computation over Behavioural Data

Ph.D. Dissertation

Flavio Chierichetti

Academic Year 2008-2009

# Contents

# Acknowledgements

I wish to thank my advisor Alessandro Panconesi for too many reasons to have them all listed here.

With his energy and willpower, his enthusiasm and tenacity, he has been the best possible inspiration for a generally pessimistic person like myself[1]. He has introduced me to the subjects of probability and distributed algorithms — two of the areas that I still like the best.

Then, I wish to thank Ravi Kumar for having been my mentor at Yahoo! Research, and for the (long) time he has spent working with me. His guidance has been fundamental in all of our joint work, and has deeply influenced my research outlook.

I also wish to thank Prabhakar Raghavan for having found the time to let me work with him. His deep insights on both the theoretical and the applied worlds, his ability to find interesting problems whose solutions are neither easy nor unattainable, are only two of the many reasons why I am very grateful to him.

My thanks go to Michael Mitzenmacher for having been my mentor at Harvard. I particularly thank him for having introduced me to Information Theory research, and for his refreshing think-outside-of-the-box attitude.

Then, I thank Paolo Boldi, Roberto Grossi, Romeo Rizzi, Massimo Santini and Sebastiano Vigna. Let me just mention that it is easy to obtain a Ph.D. degree in our field if you have had the fortune of being taught real computer science during high school by people like them. I wish to especially thank the first and the last persons in the list for the work we have done during my Ph.D. program — and for being great friends.

I also thank Silvio Lattanzi and Andrea Vattani for the many hours we have spent together thinking about problems.

I thank all my co-authors, Vicente Acuña, Paolo Boldi, Andrei Broder, Hilary Finucane, Vanja Josifovski, Ravi Kumar, Vincent Lacroix, Silvio Lattanzi, Zhenming Liu, Alberto Marchetti-Spaccamela, Federico Mari, Michael Mitzenmacher, Alessandro Panconesi, Sandeep Pandey, Prabhakar Raghavan, Marie-France Sagot, Mauro Sozio, Leen Stougie, Alessandro Tiberi, Eli Upfal, Sergei Vassilvitskii, Andrea Vattani and Sebastiano Vigna. Needless to say, all merit is theirs, and all mistakes are mine.

I thank the members of my thesis commission: Paolo Boldi, Alessandro Panconesi and Riccardo Silvestri.

I then want to thank Maurizio Ceccarani, my Literature and History teacher in high school, for being (and having been back then) the living proof that the italian school system can really make the difference.

Then I thank Danilo Abbasciano and Daniele Cardarelli for the many road trips we did together; I should actually thank Daniele for a number of other reasons too, but this is probably not the right context to embarass ourselves ☺. I thank Ilaria Bordino for having always been a friend, and a just critic ☺. And, finally, I would like to thank my family and all my friends.

[1] In fact, there are many concrete evidences (even a quiz on Facebook `http://apps.facebook.com/quale-profe-bahbajb/` ! ☺) that many different students feel the same way.

# Chapter 1

# Introduction

Looking at the scale of the data available nowadays, one would imagine that little computation can be carried out over it. Web-crawls of 100s of billions pages, social networks of 100s of millions people, streams of 100s of millions of queries per day, web-albums comprising billions of pictures, and so forth, pose formidable computational challenges.

Yet, web snapshots can be compressed efficiently, information spreads quickly across social networks, queries can be answered efficiently, and the best pictures of an album can be found quickly with the aid of a computer program.

In practice behavioural data, be it social networks, query logs, or preference preorders, can be dealt with with surprising efficiency.

In this thesis, we look at four different computational problems in this context:

- compressing the web graph — why can the webgraph be compressed so efficiently (both time and space-wise)?
- gossiping in social networks — why does information require so little time to be spread over social networks?
- caching user interests and the median problem in the Jaccard metric — these two problems have applications to web advertising. Under which assumptions can we efficiently approximate the median problem in the Jaccard metric?
- choosing the best pictures — is it possible to obtain the best elements of a non-totally-ordered collection in little time?

Practical instances of these computational problems can be usually dealt with efficiently. In this thesis, we try to understand why.

## 1.1 Webgraph Compressibility

The first problem we deal with is webgraph compressibility. This work, presented in Chapter 2, has been published in [26].

The webgraph — the directed graph having webpages as nodes and hyperlinks as arcs — is a mathematical object of great interest, socially and scientifically. In particular, the webgraph models social and behavioral phenomena whose graph-theoretic analysis has led to significant societal impact (witness the role of link analysis in web search).

Observe that in general, a graph can be stored using $O(\log n)$ bits per edge (for each edge, store the ids of its two endpoints). This bound is, generally, strict. The famous Erdös-Renyi $G(n, p)$ random graph model can be easily seen to *require* this many bits , for many settings of the parameter $p$.

On the other hand, an intriguing set of papers by Boldi, Santini, and Vigna [11–13] shows that the web graph is highly compressible: it can be stored in such a way that each edge requires only a small constant number — between one and three — of bits on average; other experimental studies confirm these findings [21]. These empirical results suggest the intriguing possibility that the webgraph can be

described with only $O(1)$ bits per edge on average — on the other hand, no one ever showed that any of the existing webgraph models had this property.

We prove that none of the most popular webgraph mathematical models can have this property — in particular, their entropy, averaged over the number of edges, is $\Theta(\log n)$.

This prompts the natural question: can we model the compressibility of the web graph, in particular mirroring the properties of locality and edge length distribution used by the algorithm from [12], while maintaining other well-known properties such as power law degree distribution? We answer affirmatively with a new webgraph model that, in addition to being compressible with the algorithm of [12], mimicks the other known webgraph properties such as power law in-degree distribution, high clustering coefficient, small diameter, and more.

## 1.2 Gossiping in Social Networks

Why does information (chain letters in the email network, music in file sharing networks, tweets in Twitter, etc.) spread so quickly through social networks? It is striking how this form of distributed, and decentralized, computation happens to be so fast.

To study this phenomenon analytically, some modeling has to be made:

- First of all: what is a social network? There appears to be no precise mathematical definition. According to wikipedia, "a social network is a social structure made of individuals (or organizations) called nodes, which are tied (connected) by one or more specific types of interdependency, such as friendship, kinship, financial exchange, dislike, sexual relationships, or relationships of beliefs, knowledge or prestige."

  In the literature, so-called "Preferential Attachment" graphs [5] — random graphs exhibiting the well-known power law degree distribution — are considered by [5] to be reasonable models of social networks. In [30], we take the Preferential Attachment graphs as models of social networks.

- Further, how does an individual spread information to her neighbours? We consider the simplest model that actually can capture the quick (that is, logarithmic time) information dissemination observed in practice. Initially, a single person knows some piece of information. At each time step, $(a)$ each person in the network calls a person uniformly at random from her set of neighbours in the graph, and $(b)$ when two people talk, they exchange the piece of information they might have. This corresponds to the so-called `PUSH-PULL` rumor spreading algorithm (a generalization of the randomized broadcast algorithm). If we replace $(b)$ with "$(b')$ when person $x$ calls person $y$, if $x$ has the information, then the information is sent to $y$", we obtain the well-known `PUSH` rumor spreading algorithm. On the other hand, if we replace (b) with "$(b'')$ when person $x$ calls person $y$, if $y$ has the information, then the information is sent to $x$" we obtain the `PULL` rumor spreading algorithm. These three basic rumor spreading algorithms have been introduced in 1987 by [40], and their performaces over different classes of graphs have been studied thoroughly in the last two decades [9, 17, 42, 83, 89].

  Arguably, both the PUSH and the PULL algorithm are simpler than the PUSH-PULL algorithm. In [30], we show that both PUSH and PULL require polynomially many rounds to spread the information over the Preferential Attachment graph, our model of *social networks*. In the same paper, we also show that PUSH-PULL does spread the information in polylogarithmically many rounds. That is why we claim that PUSH-PULL is the "simplest" mechanism that can spread information in social networks.

These results are not completely satisfactory, since Preferential Attachment graphs are far from being good models of social networks. In Chapter 3 (and in [31], where this work originally appeared) we look at the problem from a different angle: we show that in a graph with $n$ nodes and conductance[1] $\sigma$, PUSH-PULL spreads the information in $O(\text{poly}(\frac{\log n}{\sigma}))$ rounds. Conductance is a meaningful graph property in our case since some experimental work [76] indicates that the conductances of social networks might

[1] The conductance of a graph is the minimum, over all subsets of its nodes $S \subseteq V$, of the ratio $\frac{e(S,V-S)}{\min\{\text{vol}(S),\text{vol}(V-S)\}}$, where $\text{vol}(S)$ is the volume of the set $S \subseteq V$ — that is, the sum of the degrees of the nodes in $S$.

indeed be large (i.e., constant or inversely logarithmic in $n$). Thus our result can be seen as a theoretical justification of the empirically observed fact that information dissemination is fast over social networks.

The technical details of our proof could be of interest to the community, since they crucially leverage a seemingly unrelated result [99] from the theory of graph sparsification. The apparent relation between the two theories might possibly be exploited further.

## 1.3 User Interests and Caching

Consider an internet advertising system (be it Yahoo!, Google, Bing, Ask or Tip-Top) tasked with showing contextual advertisements (ads) on a publisher's page when a user visits the page. This so-called content-match system strives to choose and serve relevant ads using a range of cues – the profile of the user and the content on the publisher's page – to search the pool of available ads and retrieve one or more ads to display on the page. Ads and users are casted to points in some metric space, and an ad is considered to be relevant for a user if their distance is within some threshold.

The challenge here is to accomplish this task as efficiently as possible, given hundreds of millions of users and available ads. To do so, many internet advertising systems make use of a caching subsystem. The cache here does not only allow exact matches (like in traditional caching), but a cache element can be used to answer a query, if the cache element is "close" enough to the query. This relaxation is necessary, since the number of different users is too high.

In [29] and [87] we study, theoretically (in the framework of competitive analysis) and experimentally, this problem. In particular in [87] we give a heuristic that gives reasonable performance improvements over traditional LRU and LFU.

The users and the ads, in the Yahoo! system, are points in the so called Jaccard metric. In particular, the heuristic in [87] needs to solve the median problem in the Jaccard metric (given a subset of the metric, find the point in the metric that minimizes the sum of distance between itself and the points in the subset).

The Jaccard metric has been introduced in 1901 [60] and has been used in many different scientific fields exactly for clustering (botany, chemistry, cognitive sciences, ecology, geology, natural language processing, paleontology, social sciences, and web sciences, [18, 19, 23, 37, 57–59, 61, 65, 71, 86–88, 90, 93, 97, 104]). The median problem (also called Fermat's problem, or Steiner's problem) itself is of great interest in all clustering applications.

In Chapter 4 (and, originally, in [28]), we analyze the computational complexity of the jaccard median problem, showing that it does admit a PTAS (i.e., it can be approximated arbitrarily well in polynomial time), and that it does not admit a FPTAS. The same problem was analyzed for the first time in 1980 by Späth [97]. The $(1+\varepsilon)$-approximate algorithm runs in polytime for any fixed $\varepsilon$, but the exponent in the running time is polynomial in $\varepsilon^{-1}$ — that is, while being polynomial, it is very inefficient.

On the other hand, we give a linear time algorithm that gives a $1 + O(\sqrt{\varepsilon})$ approximation to the median problem if the total distance of the best median is no more than $\varepsilon$ times the number of points. Observe that the median problem (in particular in our context of users/ads clustering) has a meaningful application exactly when the total distance of the best median is small — that is, when the points *are*, indeed, close to each other. Therefore, our approximation algorithm is *fast* exactly on those input sets that are amenable to caching.

## 1.4 Preference Preorders of Pictures

Many picture albums websites assist the user in choosing the best pictures out of the collection she uploaded. These representatives are important for many reasons (they will be the most accessed ones, so they need to be cached in many servers around the world; they will help the viewers to quickly assess the quality of the album itself, etc.), and helping the user choosing them is easily seen to be a very important task.

Yet, choosing the best representatives from a set of objects is not easy at all; even the very nature of preference is questionable, and has been largely debated by psychometricians and economists (see, e.g., [36]). In particular, a much argued point is whether personal preference is a transitive relation or not; even if some experiments actually prove that preference may indeed not be transitive, it is commonly

agreed that intransitive preference lead to irrational behaviour. For this reason, it is by now accepted that preference can be modelled as a preorder (a.k.a. quasiorder), that is, a transitive, reflexive relation that is not required to be antisymmetric.

In Chapter 5 (and in [10]) we study the computational complexity of finding the best elements (where "best" is suitably defined) out of a preorder. Here, the "computational complexity" measures the most valuable resource: user time. Observe that the machine itself has not got any way of choosing the best pictures — the only thing it can do is to choose which pictures to show the user, so that she can choose which ones she likes best. The goal of the algorithm is to show the user as few pictures as possible, to avoid wasting her time.

We give almost optimal algorithms under different assumptions. In particular, we model the users in two different ways. First, we assume that the preference preorder is chosen uniformly at random — under this assumption, we show how the number of comparisons that the user has to make is essentialy *linear* in the number of representatives to be chosen.

This assumption is likely to be to strong to hold in practice, though. To work around this problem, we assume that the width of the preorder is bounded (but in this case the preorder can be adversarially chosen). The width of a preorder is the maximum number of pairwise incomparable elements in it. Assuming that this is bounded is, we believe, reasonable: how likely it is that, when shown say ten pictures, a user is incapable of choosing any two pictures out of the ten and assessing that she likes one better than the other? If this does not indeed happen, then the width of the preference preorder is upper bounded by ten.

We show that under the bounded width assumption, the user has only to answer a number of comparisons close to linear in the preorder size.

# Chapter 2

# Webgraph Compressibility

Graphs resulting from human behavior (the web graph, friendship graphs, etc.) have hitherto been viewed as a monolithic class of graphs with similar characteristics; for instance, their degree distributions are markedly heavy-tailed. Here we take our understanding of behavioral graphs a step further by showing that an intriguing empirical property of web graphs — their compressibility — cannot be exhibited by well-known graph models for the web and for social networks. We then develop a more nuanced model for web graphs and show that it does exhibit compressibility, in addition to previously modeled web graph properties.

## 2.1 Overview

There are three main reasons for modeling and analyzing graphs arising from the Web and from social networks: (1) they model social and behavioral phenomena whose graph-theoretic analysis has led to significant societal impact (witness the role of link analysis in web search); (2) from an empirical standpoint, these networks are several orders of magnitude larger than those studied hitherto (search companies are now working on crawls of 100 billion pages and beyond); (3) from a theoretical standpoint, stochastic processes built from independent random events — the classical basis of the design and analysis of computing artifacts — are no longer appropriate. The characteristics of such *behavioral* graphs (viz., graphs arising from human behavior) demand the design and analysis of new stochastic processes in which elementary events are highly dependent. This in turn demands new analysis and insights that are likely to be of utility in many other applications of probability and statistics.

In such analysis, there has been a tendency to lump together behavioral graphs arising from a variety of contexts, to be studied using a common set of models and tools. It has been observed [3, 20, 70] for instance that the directed graphs arising from such diverse phenomena as the web graph (pages are nodes and hyperlinks are edges), citation graphs, friendship graphs, and email traffic graphs all exhibit *power laws* in their degree distributions: the fraction of nodes with indegree $k > 0$ is proportional to $1/k^{\alpha}$ typically for some $\alpha > 1$; random graphs generated by classic Erdös–Rényi models cannot exhibit such power laws. To explain the power law degree distributions seen in behavioral graphs, several models have been developed for generating random graphs [2,3,14,15,22,49,69,75] in which dependent events combine to deliver the observed power laws.

While the degree distribution is a fundamental but local property of such graphs, an important global property is their compressibility — the number of bits needed to store each edge in the graph. Compressibility determines the ability to efficiently store and manipulate these massive graphs [62, 101, 107]. An intriguing set of papers by Boldi, Santini, and Vigna [11–13] shows that the web graph is highly compressible: it can be stored such that each edge requires only a small constant number — between one and three — of bits on average; a more recent experimental study confirms these findings [21]. These empirical results suggest the intriguing possibility that the Web can be described with only $O(1)$ bits per edge on average. Two properties are at the heart of the compression algorithm of Boldi and Vigna [12]. First, once web pages are sorted lexicographically by URL, the set of outlinks of a page exhibits locality; this can plausibly be attributed to the fact that nearby pages are likely to come from the same website's template. Second, the distribution of the lengths of edges follows a power law with exponent $> 1$ (the

length of an edge is the distance of its endpoints in the ordering); this turns out to be crucial for high compressibility. This prompts the natural question: can we model the compressibility of the web graph, in particular mirroring the properties of locality and edge length distribution, while maintaining other well-known properties such as power law degree distribution.

**Main results.** Our first set of results show that the best known models for the web graph cannot account for compressibility, in the sense that they require $\Omega(\log n)$ bits storage per edge on average. This holds even when these graphs are represented just in terms of their topology (i.e., with all labels stripped away). Specifically, we show that the preferential attachment model [3, 14], the ACL model [2], the copying model [69], the Kronecker product model [74], and Kleinberg's model for navigability[1] on social networks [66], all have large entropy in the above sense.

We then show our main result: a new model for the web graph that has constant entropy per edge, while preserving crucial properties of previous models such as the power law distribution of indegrees, a large number of communities (i.e., bipartite cliques), small diameter, and a high clustering coefficient. In this model, nodes lie on the line and when a new node arrives it selects an existing node uniformly at random, placing itself on the line to the immediate left of the chosen node. An edge from the new to the chosen node is added, and moreover all outgoing edges of the chosen node but one are copied (these edges are chosen at random); thus, the edges have some locality. We then show a crucial property of our model: the power law distribution of edge lengths. Intuitively, this long-get-longer effect is caused since a long edge is likely to receive the new node (which selects its position uniformly at random) under its protective wing, and the longer it gets, the more likely it is to attract new nodes. Using this, we show that the graphs generated by our model are compressible to $O(1)$ bits per edge; we also provide a linear-time algorithm to compress an unlabeled graph generated by our model.

**Technical contributions and guided tour.** In Section 2.3 and Section 2.8.2 we prove that several well-known web graph models are not compressible, i.e., they need $\Omega(\log n)$ bits per edge. In fact, we prove incompressibility even after the labels of nodes and orientations of edges are removed.

Sections 2.4 presents our new model and Sections 2.5, 2.6, and Section 2.8.8) present the basic properties of our model. Although our new model might at first sight closely resembles a prior *copying* model of [69], it differs in fundamental respects. First, our new model successfully admits the global property of compressibility which the copying model *provably* does not. Second, while the analysis of the distribution of the in-degrees is rather standard, the crucial property that edge lengths are distributed according to a power law requires an entirely novel analysis; in particular, the proof requires a very delicate understanding of the structural properties of the graphs generated by our model in order to establish the concentration of measure. Section 2.7 addresses the compressibility of our model, where we also provide an efficient algorithm to compress graphs generated by our model.

It is difficult to distinguish experimentally between graphs that require only $O(1)$ bits per edge and those requiring, say, $\varepsilon \log n$ bits. The point however is that the compressibility of our model relies upon other important structural properties of real web graphs that previous models, in view of our lower bounds, provably cannot have.

**Related prior work.** The observation of power law degree distributions in behavioral (and other) graphs has a long history [3,70]; indeed, such distributions predate the modern interest in social networks through observations in linguistics [108] and sociology [95]; see the survey by Mitzenmacher [82]. Simon [95], Mandelbrot [79], Zipf [108] and others have provided a number of explanations for these distributions, attributing them to the dependencies between the interacting humans who collectively generate these statistics. These explanations have found new expression in the form of rich-get-richer and herd-mentality theories [3, 106]. Early rigorous analyses of such models include [2, 14, 34, 69]. Whereas Kumar et al. [69] and Borgs et al. [15] focused on modeling the web graph, the models of Aiello, Chung, and Lu (ACL) [2], Kleinberg [66], Lattanzi and Sivakumar [72], and Leskovec et al. [74] addressed social graphs in which people are nodes and the edges between them denote friendship. The ACL model is in fact known not to be a good representation of the web graph [70], but is a plausible model for human social networks. Kleinberg's model of social networks focuses on their *navigability*: it is possible for a node to find a short

[1] Since navigability is a crucial property of real-life social networks (cf. [41, 77, 103]), it is tempting to conjecture that social networks are incompressible; see, for instance, [27].

route to a target using only local, myopic choices at each step of the route. The papers by Boldi, Santini and Vigna [11–13] suggests that the web graph is highly compressible (see also [1, 21, 27, 101]).

## 2.2 Preliminaries

The graph models we study will either have a fixed number of nodes or will be evolving models in which nodes arrive in a discrete-time stochastic process; for many of them, the number of edges will be linear in the number of nodes. We analyze the space needed to store a graph randomly generated by the models under study; this can be viewed in terms of the entropy of the graph generation process. Note that a naive representation of a graph would require $\Omega(\log n)$ bits per edge; entropically, one can hope for no better for an Erdös–Rényi graph. We are particularly interested in the case when the amortized storage per edge can be reduced to a constant. As in the work of Boldi and Vigna [12, 13], we view the nodes as being arranged in a linear order. To prove compressibility we then study the distribution of edge *lengths* — the distance in this linear order between the end-points of an edge.

Given a function $f : A_1 \times \cdots \times A_n \to \mathbb{R}$, we say that $f$ satisfies the *c-Lipschitz property* if, for any sequence $(a_1, \ldots, a_n) \in A_1 \times \cdots \times A_n$, and for any $i$ and $a_i' \in A_i$,

$$|f(a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_n) - f(a_1, \ldots, a_{i-1}, a_i', a_{i+1}, \ldots, a_n)| \leq c.$$

In order to establish that certain events occur w.h.p., we will make use of the following concentration result known as the *method of bounded differences* (cf. [44]).

**Theorem 1 (Method of bounded differences).** *Let $X_1, \ldots, X_n$ be independent r.v.'s. Let $f$ be a function on $X_1, \ldots, X_n$ satisfying the c-Lipschitz property. Then,*

$$\mathrm{P}\,|f(X_1, \ldots, X_n) - \mathrm{E}\,[f(X_1, \ldots, X_n)]| > t \leq 2e^{-t^2/(c^2 n)}.$$

The *Gamma function* is defined as $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$. We use these properties of the Gamma function: (i) $\Gamma(x+1) = x\Gamma(x)$, (ii) $\Gamma(x)\Gamma(x+\frac{1}{2}) = \Gamma(2x)2^{1-2x}\sqrt{\pi}$, and (iii) for constants $a, b \in \mathbb{R}$, $\lim_{n\to\infty} \frac{\Gamma(n+a)}{\Gamma(n+b)} n^{b-a} = 1$. The following lemma about the Gamma function is proved in Section 2.8.3.

**Lemma 2.** *Let $a, b \in \mathbb{R}^+$ be such that $b \neq a+1$. For each $t \in \mathbb{Z}^+$, it holds that*

$$\sum_{i=1}^{t} \frac{\Gamma(i+a)}{\Gamma(i+b)} = \frac{1}{b-a-1} \cdot \left( \frac{\Gamma(a+1)}{\Gamma(b)} - \frac{\Gamma(t+a+1)}{\Gamma(t+b)} \right).$$

## 2.3 Incompressibility of the existing models

In this section we prove the inherent incompressibility of commonly-studied random graph models for social networks and the web. We show that on average $\Omega(\log n)$ bits per edge are necessary to store graphs generated by several well-known models for web/social networks, including the preferential attachment and the copying models. In our lower bounds, we show that the random graph produced by the models we consider are incompressible, *even after removing the labels of their nodes and orientations of their edges.* Given a labeled/directed graph and its unlabeled/undirected counterpart, the latter is more compressible than the former; in fact, the gap can be arbitrarily large (see Section 2.8.1 for some examples). Thus the task of proving incompressibility of unlabeled/undirected versions of graphs generated by various models is made more challenging. Note that it is crucial to analyze the compressibility of unlabeled graphs — the experiments on web graph [12, 13] show how just the edges can be compressed using only $\approx 2$ bits per edge.

### 2.3.1 Proving incompressibility

Let $\mathcal{G}_n$ denote the set of all directed labeled graphs on $n$ nodes. Let $P_n^\theta : \mathcal{G}_n \to [0, 1]$ denote the probability distribution on $\mathcal{G}_n$ induced by the random graph model $\theta$. We consider the preferential attachment model ($\theta = \mathsf{pref}$), the ACL model ($\theta = \mathsf{acl}$), the copying model ($\theta = \mathsf{copy}$), the Kronecker multiplication model ($\theta = \mathsf{krm}$), and Kleinberg's model ($\theta = \mathsf{kl}$).

For a given $\theta$, let $\mathrm{H}(P_n^\theta)$ denote the Shannon entropy of the distribution $P_n^\theta$, that is, the average number of bits needed to represent a directed labeled random graph generated by $\theta$. Our goal is to obtain lower bounds on the representation. This is accomplished by the following min-entropy argument.

**Lemma 3 (Min-entropy argument).** *Let $\mathcal{G}_n^* \subseteq \mathcal{G}_n$, $P^+ \leq \sum_{G\in\mathcal{G}_n^*} P_n^\theta(G)$, and $P^* \geq \max_{G\in\mathcal{G}_n^*} P_n^\theta(G)$. Then, $\mathrm{H}(P_n^\theta) \geq P^+ \cdot \log(1/P^*)$.*

*Proof.*

$$\mathrm{H}(P_n^\theta) = \sum_{G\in\mathcal{G}_n} P_n^\theta(G)\log\frac{1}{P_n^\theta(G)} \geq \sum_{G\in\mathcal{G}_n^*} P_n^\theta(G)\log\frac{1}{P_n^\theta(G)} \geq \sum_{G\in\mathcal{G}_n^*} P_n^\theta(G)\log\frac{1}{P^*} \geq P^+\cdot\log\frac{1}{P^*}. \quad \square$$

Thus, to obtain lower bounds on $\mathrm{H}(P_n^\theta)$, we will upper bound $\max_{G\in\mathcal{G}_n^*} P_n^\theta(G)$ by $P^*$ and lower bound $\sum_{G\in\mathcal{G}_n^*} P_n^\theta(G)$ by $P^+$, for a suitably chosen $\mathcal{G}_n^* \subseteq \mathcal{G}_n$. For good lower bounds on $\mathrm{H}(P_n^\theta)$, $\mathcal{G}_n^*$ has to be chosen judiciously. For instance, choosing a large $\mathcal{G}_n^*$ (say, $\mathcal{G}_n$) might only yield a $P^*$ that is moderately small, while at the same time, it is important to choose a $\mathcal{G}_n^*$ such that $P^+$ is large.

Let $\mathcal{H}_n$ denote the set of all undirected unlabeled graphs on $n$ nodes. Let $\varphi : \mathcal{G}_n \to \mathcal{H}_n$ be the many-to-one map that discards node and edge labels and edge orientations. For a given model $\theta$, let $Q_n^\theta : \mathcal{H}_n \to [0,1]$ be the probability distribution such that $Q_n^\theta(H) = \sum_{\varphi(G)=H} P_n^\theta(G)$. Clearly, $\mathrm{H}(Q_n^\theta) \leq \mathrm{H}(P_n^\theta)$ and therefore, lower bounds on $\mathrm{H}(Q_n^\theta)$ are stronger and harder to obtain.

### 2.3.2 Incompressibility of the preferential attachment model

Consider the preferential attachment model ($\mathsf{pref}[\mathsf{k}]$) defined in [14]. This model is parametrized by an integer $k \geq 1$. At time 1, the (undirected) graph consists of a single node $x_1$ with 1 self-loop. At time $t > 1$,

(1) a new node $x_t$, labeled $t$, is added to the graph;

(2) a random node $y$ is chosen from the graph with probability proportional to its current degree (in this phase, the degree of $x_t$ is taken to be 1);

(3) the edge $x_t \to y$, labeled $t \mod k$, is added to the graph;[2] and

(4) if $t$ is a multiple of $k$, nodes $t-k+1,\ldots,t$ are merged together, preserving self-loops and multiedges.

For $k = 1$, note that the graphs generated by the above model are forests. Since there are $2^{O(n)}$ unlabeled forests on $n$ nodes (e.g., [84]), whose edges can be directed in at most $2^n$ ways, $\mathrm{H}(Q_n^{\mathsf{pref}[\mathsf{k}]}) = O(n)$, i.e., the graph without labels and edge orientations is compressible to $O(1)$ bits per edge. The more interesting case is when $k \geq 2$ for which we show an incompressibility bound.

We underscore the importance of a good choice of $\mathcal{G}_n^*$ in applying Lemma 3. Consider the graph $G$ having the first node of degree $k(n+1)$ and the other $n-1$ nodes of degree $k$. Clearly, $P_n^{\mathsf{pref}[\mathsf{k}]}(G) = \prod_{i=k+1}^{nk} \frac{k-1+i}{2i-1} \geq 2^{-nk}$. Thus, choosing a set $\mathcal{G}_n^*$ containing $G$, would force us to have $P^* \geq 2^{-nk}$ so that the entropy bound given by Lemma 3 would only be $H(P_n^{\mathsf{pref}[\mathsf{k}]}) \geq nk = \Theta(n)$. (A similar issue would be encountered in the unlabeled case as well.) A careful choice of $\mathcal{G}_n^*$, however, yields a better lower bound.

**Theorem 4.** $\mathrm{H}(Q_n^{\mathsf{pref}[\mathsf{k}]}) = \Omega(n\log n)$, *for $k \geq 2$.*

*Proof.* Let $G$ be a graph generated by $\mathsf{pref}[\mathsf{k}]$. Let $\deg_t(x_i)$, for $i \leq t$, be the degree of the $i$-th inserted node at time $t$ in $G$. By [35, Lemma 6], with probability $1 - O(n^{-3})$, for each $1 \leq t \leq n$, each node $x_i, 1 \leq i \leq t$, will have degree $\deg_t(x_i) < (\sqrt{t/i})\log^3 n$ in $G$.

In particular, let $t^* = \lceil \sqrt[3]{n} \rceil$. Let $\xi$ be the event: "$\exists t \geq t^*, \sum_{i=1}^{t^*} \deg_t(x_i) \geq n^{3/4}$." At time $n$, the sum of the degrees of nodes $x_1,\ldots,x_{t^*}$ can be upper bounded by

$$\sum_{i=1}^{t^*} \deg_n(x_i) \leq \sum_{i=1}^{t^*} \sqrt{\frac{n}{i}}\log^3 n = \sqrt{n}\log^3 n \sum_{i=1}^{t^*} i^{-1/2} < O(n^{3/4}),$$

w.h.p. Indeed, $\mathrm{P}\,\xi \leq O(n^{-3})$.

Now define $t^+ = \lceil \varepsilon n \rceil$, for some small enough $\varepsilon > 0$; let $n$ be large enough such that $t^* < t^+$. We call a node added after time $t^+$ *good* if it is not connected to any of the first $t^*$ nodes. To bound the number of good nodes from below, we condition on $\xi$, and we upper bound the number of bad nodes. Using a union bound, the probability that node $x_t$ for $t \geq t^*$ is bad can be upper bounded by $k \cdot n^{3/4}/(\varepsilon n) \leq O(n^{-1/4})$.

[2] In the original PA model, edges are both undirected and unlabeled: we direct and label them for simplicity of exposition. The entropy lower bound will hold for the undirected and unlabeled version of these graphs.

Let $\xi'$ be the event: "at least $(1-2\varepsilon)n$ nodes are good"; by stochastic dominance, the event $\xi'$ happens w.h.p. In our application of Lemma 3, we will choose $\mathcal{G}_n^* \subseteq \mathcal{G}_n$ to be the set of graphs satisfying $\xi \cap \xi'$. Thus, $P^+ = \mathrm{P}\,\xi \cap \xi' = 1 - o(1)$. Moreover,

$$\max_{G \in \mathcal{G}_n^*} P_n^{\mathsf{pref[k]}}(G) \leq \left( \frac{\sqrt{\frac{n}{\sqrt[3]{n}}} \log^3 n}{kn} \right)^{(1-2\varepsilon)kn} \leq \left( O(n^{-2/3+\varepsilon}) \right)^{2(1-2\varepsilon)n} \leq n^{-\frac{4}{3}n + \frac{14}{3}\varepsilon n} = \rho.$$

(Notice how, by applying Lemma 3 at this point, we already have that $\mathrm{H}(P_n^{\mathsf{pref[k]}}) \geq \Omega(n \log n)$.)

Now, we proceed to lower bound $\mathrm{H}(Q_n^{\mathsf{pref[k]}})$ through an upper bound on $|\varphi^{-1}(H)|$ for $H \in \mathcal{H}_n^{\mathsf{pref[k]}}$, by a careful counting argument. Given a $H$, it is possible to determine for each of its edges, which of the two endpoints of the edge was responsible for adding the edge to the graph. This task is trivial for edges incident to any node of degree $k$, as that node will have necessarily added all $k$ edges to the graph. So, we can remove all degree $k$ nodes from the graph and repeat this process until the graph becomes empty.

Thus, $H$ could have been produced from at most $n! \cdot (k!)^n$ labeled graphs, since there are at most $n!$ ways of labeling the nodes, and $k!$ ways of labeling each of the "outgoing" edges of each node. That is, $\left|\varphi^{-1}(H)\right| \leq n! \cdot (k!)^n \leq n^n k^{kn}$. Then, choosing $\mathcal{H}_n^* \subseteq \mathcal{H}_n$ to be the set of unlabeled graphs obtained by removing labels from $\mathcal{G}_n^*$, $\mathcal{H}_n^* = \{\varphi(G) \mid G \in \mathcal{G}_n^*\}$, we obtain $P^+ = 1 - o(1)$, and

$$\max_{H \in \mathcal{H}_n^*} Q_n^{\mathsf{pref[k]}}(H) \leq \rho \cdot n^n \cdot k^{kn} = n^{-\Omega(n)} k^{kn} = P^*.$$

Finally, an application of Lemma 3 gives $\mathrm{H}(Q_n^{\mathsf{pref[k]}}) \geq P^+ \cdot \log \frac{1}{P^*} \geq \Omega(n \log n)$, completing the proof. □

### 2.3.3 Incompressibility of other graph models

We now state the incompressibility results for other well-known graph models. Due to lack of space, the definitions of the models along with the proofs of the following results are provided in Section 2.8.2.

**Theorem 5.** $\mathrm{H}(Q_n'^{\mathsf{acl}[\alpha]}) = \Omega(n \log n)$, *for*[3] $\alpha > 1/2$.

**Theorem 6.** $\mathrm{H}(Q_n^{\mathsf{copy}[\alpha,\mathsf{k}]}) = \Omega(n \log n)$, *for* $k > 2/\alpha$.

**Theorem 7.** *Let* $\ell \geq 2$ *and* $1/\ell < \alpha < 1$. *Then, w.h.p.,* $\mathrm{H}(Q_n^{\mathsf{krm}[\mathsf{M},\mathsf{s}]}) = \Omega(m \log n)$, *where* $n = \ell^s$, $M = \alpha \cdot J_\ell$, *and* $m$ *is the number of edges.*

**Theorem 8.** $\mathrm{H}(Q_n^{\mathsf{kl}}) = \Omega(n \log n)$.

## 2.4 The new web graph model

In this section we present our new web graph model. Let $k \geq 2$ be a fixed positive integer. Our new model creates a directed simple graph (i.e., no self-loops or multiedges) by the following process.

The process starts at time $t_0$ with a simple directed *seed graph* $G_{t_0}$ whose nodes are arranged on a (discrete) line, or list. The graph $G_{t_0}$ has $t_0$ nodes, each of outdegree $k$. Here, $G_{t_0}$ could be, for instance, a complete directed graph with $t_0 = k + 1$ nodes.

At time $t > t_0$, an existing node $y$ is chosen uniformly at random (u.a.r.) as a prototype:

(1) a new node $x$ is placed to the immediate left of $y$ (so that $y$, and all the nodes on its right, are shifted one position right in the ordering),

(2) a directed edge $x \to y$ is added to the graph, and

(3) $k - 1$ edges are "copied" from $y$, i.e., $k - 1$ successors (i.e., out-neighbors) of $y$, say $z_1, \ldots, z_{k-1}$, are chosen u.a.r. without replacement and the directed edges $x \to z_1, \ldots, x \to z_{k-1}$ are added to the graph.

[3]Here we do not use the probability distribution $Q$ on the graphs of $n$ nodes — in the $\mathsf{acl}[\alpha]$ model the number of nodes is a r.v. . $Q_n'^{\mathsf{acl}[\alpha]}$ denotes the probability distribution on the graphs that can be generated by the $\mathsf{acl}[\alpha]$ model in $n$ steps.

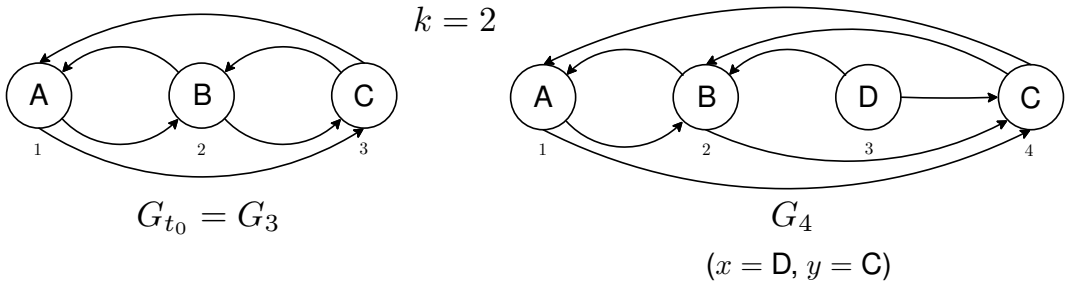


Figure 2.1: The new node $x = D$ chooses $y = C$ as its prototype. The edge $C \to B$ is copied and the new edge $D \to C$ is added for reference. Notice that all the edges incident to $C$ in $G_{t_0} = G_3$ increase their length by 1 in $G_{t_0+1} = G_4$.

See Figure 2.1 for an illustration of our model.

An intuitive explanation of this process is as follows. Consider the list of webpages ordered lexicographically by their URLs (for this ordering, a url `a.b.com/d/e` is to be interpreted as `com/b/a/d/e`.) A website owner might decide to add a new webpage to her site; to do this, she could take one of the existing webpages from her site as a prototype, modify it as needed, add an edge to the prototype for reference, and publish the new page on her site. Thus the new webpage and the prototype will be close in the URL ordering.

In our model, we can show the following:

(1) The fraction of nodes of indegree $i$ is asymptotic to $\Theta(i^{-2-\frac{1}{k-1}})$; this power law is often referred to as "the rich get richer."

(2) The fraction of edges of length[4] $\ell$ in the given embedding is asymptotic to $\Theta(\ell^{-1-\frac{1}{k}})$; analogously, we refer to this as "the long get longer."

Boldi and Vigna [12] study the distribution of *gaps* in the web graph, defined as follows. Sort the webpages lexicographically by URLs and this gives an embedding of nodes on the line. Now, if a webpage $x = z_0$ has edges to $z_1, \ldots, z_j$ in this order, the gaps are given by $|z_{i-1} - z_i|$, $1 \le i \le j$. They observe how the gap distribution in real web graph snapshots follows a power law with exponent $\approx 1.3$. Our model can capture a similar distribution for the edge *lengths*, by an appropriate choice of $k$. In fact, both the average edge length and the average gap in our model are small; intuitively, though not immediately, this leads to the compressibility result of Section 2.7. It turns out that a power law distribution of either the lengths or the gaps (with exponent $> 1$) is sufficient to show compressibility; for sake of simplicity, we focus on the former in Section 2.6.

## 2.5 The rich get richer

In this section we characterize the indegree distribution of our graph model. We show that the expected indegree distribution follows a power law. We then show the distribution is tightly concentrated.

Let

$$f(i) = \frac{k\, 2^{1+\frac{2}{k-1}}\, \Gamma\left(\frac{3}{2} + \frac{1}{k-1}\right)}{(k-1)\sqrt{\pi}} \cdot \frac{\Gamma\left(i + 1 + \frac{1}{k-1}\right)}{\Gamma\left(i + 3 + \frac{2}{k-1}\right)}.$$

It is easy to show that $\lim_{i\to\infty} f(i) \Big/ \left(\frac{k\, 2^{1+\frac{2}{k-1}}\, \Gamma\left(\frac{3}{2}+\frac{1}{k-1}\right)}{(k-1)\sqrt{\pi}} \cdot i^{-2-\frac{1}{k-1}}\right) = 1$, i.e., $f(i) = \Theta(i^{-2-\frac{1}{k-1}})$. Let $X_i^t$ denote the number of nodes of indegree $i$ at time $t$. We first show that $\mathrm{E}[X_i^t]$ can be bounded by $f(i) \cdot t \pm c$, for some constant $c$ (proof is in Section 2.8.4).

**Theorem 9.** *There is a constant $c = c(G_{t_0})$ such that*

$$f(i) \cdot t - c \le \mathrm{E}[X_i^t] \le f(i) \cdot t + c, \qquad (1)$$

*for all $t \ge t_0$ and $i \in [t]$.*

[4]The *length* of an edge $x \to y$ is the absolute difference between the positions of node $x$ and $y$ in the given embedding.

Thus, in expectation, the indegrees follow a power law with exponent $-2 - 1/(k-1)$. We now show a $O(1)$-Lipschitz property for the r.v.'s $X_i^t$ for $k = O(1)$; the proof is in Section 2.8.5. The concentration immediately follows using Theorem 1.

**Lemma 10.** *Each r.v. $X_i^t$ satisfies the $(2k)$-Lipschitz property.*

## 2.6 The long get longer

In this section we analyze the edge length distribution in our graph model. We show it follows a power law with exponent more than 1. Later, we will use this to establish the compressibility of graphs generated by our model. Let

$$g(\ell) = \frac{\Gamma\left(\ell + 1 - \frac{1}{k}\right)}{\Gamma\left(2 - \frac{1}{k}\right)\Gamma\left(\ell + 2\right)}.$$

It holds that $\lim_{\ell\to\infty} g(\ell) \Big/ \left(\ell^{-1-\frac{1}{k}} / \Gamma\left(2 - \frac{1}{k}\right)\right) = 1$, i.e., $g(\ell) = \Theta(\ell^{-1-\frac{1}{k}})$. Recall that the *length* of an edge from a node in position $i$ to a node in position $j$ is equal to $|i - j|$; we define its *circular directed length*, denoted *cd-length*, to be $j - i$ if $j > i$, and $t - (i - j)$ otherwise. Let $Y_\ell^t$ be the number of edges of length $\ell$ at time $t$. We aim to show that $Y_\ell^t \approx g(\ell) \cdot t$. It turns out to be useful to consider a related r.v. $Z_\ell^t$, which denotes the number of edges of cd-length $\ell$ at time $t$. We will first show that, w.h.p., $Z_\ell^t \approx g(\ell) \cdot t$. We will then argue that $Y_\ell^t$ is very close to $Z_\ell^t$.

The following shows that $\mathrm{E}\left[Z_\ell^t\right]$ is bounded by $g(\ell) \cdot t \pm O(1)$; the proof is in Section 2.8.6.

**Theorem 11.** *There exists some constant $c = c(G_{t_0})$ such that*

$$g(\ell) \cdot t - c \leq \mathrm{E}\left[Z_\ell^t\right] \leq g(\ell) \cdot t + c,$$

*for all $t \geq t_0$ and $\ell \in [t]$.*

Thus, the expectation of the edge lengths follows a power law with exponent $-1 - 1/k$.

To establish the concentration result, we need to analyze quite closely the combinatorial structure of the graphs generated by our model. Recall that the nodes in our graphs are placed contiguously on a discrete line (or list). At a generic time step, we use $x_i$ to refer to the $i$-th node in the ordering from left to right. Given an ordering $\pi = (x_1, x_2, \ldots, x_t)$ of the nodes, and an integer $0 \leq k < t$, a *$k$-rotation*, $\rho_k(x_i)$ maps the generic node $x_i$, $1 \leq i \leq t$, to position $1 + ((i + k) \mod t)$.

We say that two nodes $x, x'$ are *consecutive* if there exists a $k$ such that $|\rho_k(x) - \rho_k(x')| = 1$, i.e., they are consecutive if in the ordering either they are adjacent or one is the first and the other the last. Further, we say that an edge $x'' \to x'''$ *passes over* an node $x$ if there exists $k$ such that $\rho_k(x'') < \rho_k(x) < \rho_k(x''')$. Finally, two edges $x \to x'$ and $x'' \to x'''$ are said to *cross* if there exists a $k$ such that after a $k$-rotation exactly one of $x$ and $x'$ is within the positions $\rho_k(x'')$ and $\rho_k(x''')$. We prove the following characterization that will be used later in the analysis.

**Lemma 12.** *At any time, given any two consecutive nodes $x, x'$, and any positive integer $\ell$, the number of edges of cd-length $\ell$ that pass over $x$ or $x'$ (or both) is at most $C = (k+2)t_0 + 1$.*

*Proof.* Let us define $G_t^-$ as the graph $G_t$ minus the edges incident to the nodes that were originally in $G_{t_0}$. Note that, for each cd-length $\ell$, the number of the edges of cd-length $\ell$ that we remove is upper-bounded by $2t_0$ as each node can be incident to at most two edges of cd-length $\ell$, one going in, and one going out of the node. Unless otherwise noted, we will consider $G_t^-$ for the rest of the proof.

Fix the time $t$, and take any rotation $\rho$; let $x_1, \ldots, x_t$ be the nodes in the list in the left-right order given by the rotation (i.e., node $x_i$ is in position $i$ according to $\rho$). For a set of edges of the same cd-length to pass over at least one of two consecutive nodes $x, x'$ it is necessary for every pair of them to cross. We will bound, for a generic edge $e$, the number of edges that cross $e$ and have the same length as $e$. Let $t(x_a)$ be the time when $x_a$ was added to the graph. First, by definition we have that if $x_a \to x_b$, then $t(x_a) > t(x_b)$.

Second, we claim that if there exists a rotation $\rho'$ such that $x_a, x_b, x_c$ are three nodes with $\rho'(x_a) < \rho'(x_b) < \rho'(x_c)$ and $t(x_c) > t(x_b)$, then the edge $x_a \to x_c$ cannot exist. To see this, for $x_a \to x_c$ to exist it must be that $t(x_a) > t(x_c)$. We want to show inductively that all the nodes that will point to $x_c$ will be both to the left of $x_c$ and to the right of $x_b$, in the ordering implied by $\rho'$. Note that $x_c$ was not in $G_{t_0}$ since its insertion time is larger than that of $x_b$. Thus, each node placed to the immediate left of $x_c$

will point to it, and will trivially satisfy the induction hypothesis. Furthermore, each node that copies an edge to $x_c$ must be placed to the immediate left of a node pointing to $x_c$. Thus, the second claim is proved.

Third, we claim that if $x_a, x_b, x_c, x_d$ are four nodes such that the edges $x_a \to x_c$ and $x_b \to x_d$ exist, and *cross* each other, then there exists an edge $x_c \to x_d$. To see this, first note that none of these four nodes could have been part of $G_{t_0}$, for otherwise at least one of the two edges could not have been part of $G_t^-$. Fix a rotation $\rho''$ s.t. $\rho''(x_a) < \rho''(x_b) < \rho''(x_c)$; by the second claim, it must be that $t(x_b) > t(x_c)$. Thus, the edge $x_b \to x_d$ has necessarily been copied from some node, say $x_{b_1}$. Note that $\rho''(x_{b_1}) \leq \rho(x_c)$. Indeed by assumption $\rho''(x_c) > \rho''(x_b)$ and it is impossible that $\rho''(x_c) < \rho''(x_{b_1})$, for otherwise $x_b$ could not have copied from $x_{b_1}$ as $t(x_b) > t(x_c)$. Now, we know that the edge $x_{b_1} \to x_d$ exists (as before, $x_{b_1}$ is not part of $G_{t_0}$). If $x_{b_1} = x_c$, then we are done. Otherwise, there must exist an $x_{b_2}$ pointing to $x_d$ from which $x_{b_1}$ has copied the edge. Note that $\rho''(x_{b_1}) < \rho''(x_{b_2}) \leq \rho''(x_c)$. By iterating this reasoning, the claim follows.

Take any set $S$ of edges having the same length, and such that any pair of them *cross*. Given an arbitrary $\rho'''$, let $x$ be the node with the smallest $\rho'''(x)$ such that, for some $x'$, the edge $x \to x'$ is in $S$ (the nodes $x$ and $x'$ are unique). For any other edge $y \to y'$ in $S$, by the third claim, there must exist the edge $x' \to y'$. As $x'$ has outdegree $k$, it follows that $|S| \leq k + 1$.

Finally, since the seed graph $G_{t_0}$ had $k \cdot t_0$ edges and we removed at most $2t_0$ edges of cd-length $\ell$ (for an arbitrary $\ell \geq 1$) in the cut $[G_{t_0}, G_t \setminus G_{t_0}]$, we have refrained from counting at most $k \cdot t_0 + 2t_0$ edges of length $\ell$ passing over one of the nodes $x, x'$. The proof follows. □

Now we prove the $O(1)$-Lipschitz property of the r.v.'s $Z_\ell^t$, if $t_0, k = O(1)$. The concentration of the $Z_\ell^t$ will follow immediately from Theorem 1.

**Lemma 13.** *Each r.v. $Z_\ell^t$ satisfies the $((t_0+2)k+1)$-Lipschitz property.*

*Proof.* We use the stochastic interpretation as in the proof of Lemma 10. For each $\tau$, let $Z_\ell^\tau$ be the r.v. representing the number of edges of cd-length $\ell$ at time $\tau$. We consider $Y_\ell^\tau$ as a function of the trials $(Q_1, R_1)$, ..., $(Q_\tau, R_\tau)$. We show that changing the outcome of any single trial $(Q_{t'}, R_{t'})$, changes the r.v. $Z_\ell^\tau$, for fixed $\ell$, by an amount not greater than $C + k = k(t_0 + 1) + 2t_0 + 1$.

Suppose we change $(q_{t'}, r_{t'})$ to $(q'_{t'}, r'_{t'})$, going from graph $G$ to $G'$. Let $x$ be the node added at time $t'$ with the choice $(q_{t'}, r_{t'})$, and $x'$ be its equivalent with the choice $(q'_{t'}, r'_{t'})$. We show that choosing two different positions for $x$ and $x'$ can change the number of edges of cd-length $\ell$ by at most $C + k$ at any time step. Note that before time step $t'$, the cd-lengths are all equal.

By Lemma 12, at time $t > t'$, for all $\ell$, the number of edges of cd-length $\ell$ that pass over $x$ (resp., $x'$) is upper bounded by $C$. For an edge $e$, let $S_e$ be the set of edges that have been copied from $e$, directly or indirectly, including $e$ itself, i.e., $e \in S_e$ and if an edge $e'$ is copied from some edge in $S_e$, then $e' \in S_e$. It is easy to note that no two edges in $S_e$ have the same cd-length, since they all start from different nodes, but end up at the same node.

For any node $z$, if $e_1, \ldots, e_k$ are the successors of $z$, we define $S_z = S_{e_1} \cup \cdots \cup S_{e_k}$. The last observation implies that, for any fixed $\ell$, no more than $k$ edges of cd-length $\ell$ are in $S_v$ (or $S_{v'}$) at any single time step. Now, consider the following edge bijection from $G$ to $G'$: the $i$-th edge of the $j$-th inserted node in $G$ is mapped to the $i$-th edge of the $j$-th inserted node in $G'$. It is easy to see that if an edge $e$ in $G$ (resp., $G'$) does not pass over $x$ (resp., $x'$) and is not in $S_x$ (resp., $S_{x'}$), then $e$ gets mapped to an edge of the same cd-length in $G'$ (resp., $G$). Thus, the difference in the number of edges of the cd-length $\ell$ in $G$ and $G'$ is at most $C + k$. □

We now show that the number $D_t$ of edges whose length and cd-length are different (at time $t$) is very small. Since the maximum absolute difference between $Y_\ell^t$ and $Z_\ell^t$ is bounded by $D_t$, this will show that these r.v.'s are close to each other. First note that if an edge $x_i \to x_j$ has different length and cd-length, then $j < i$; call such an edge *left-directed* and let $R_t$ be the set of left-directed edges. Since $D_t \leq R_t$, it suffices to bound the latter.

**Lemma 14.** *With probability $1 - O\left(\frac{1}{t}\right)$, $R_t \leq O\left(t^{1-\frac{1}{k}+\varepsilon}\right)$, for each constant $\varepsilon > 0$.*

*Proof.* Observe that each edge $x_i \to x_j$ counted by $R_t$ is such that $j < i$. Thus, $R_{t_0}$ is equal to the number of left-directed edges in $G_{t_0}$ with its given embedding.

Further, $R_t$'s increase over $R_{t-1}$ equals the number of left-directed edges copied at step $t$ (the proximity edge is always not left-directed).

Thus, $E[R_t|R_{t-1}] = \left(1 + (k-1)\cdot\frac{1}{k(t-1)}\right)\cdot R_{t-1}$ and $E[R_t] = \left(1 + (k-1)\cdot\frac{1}{k(t-1)}\right)\cdot E[R_{t-1}]$, for each $t > t_0$. Therefore,

$$E[R_t] = R_{t_0}\cdot\prod_{i=t_0+1}^{t}\left(1+\frac{k-1}{k}\cdot\frac{1}{i}\right) = R_{t_0}\cdot\prod_{i=t_0+1}^{t}\frac{i+\frac{k-1}{k}}{i} = R_{t_0}\cdot\frac{\Gamma\left(t+\frac{k-1}{k}+1\right)\cdot\Gamma\left(t_0+1\right)}{\Gamma\left(t_0+\frac{k-1}{k}+1\right)\cdot\Gamma\left(t+1\right)}.$$

Thus, $E[R_t] = \Theta\left(t^{1-\frac{1}{k}}\right)$. We note how a $O(1)$-Lipschitz condition holds (at most $k-1$ new left-directed edges can be added at each step). Thus Theorem 1 can be applied with an error term of $O\left(\sqrt{t\log t}\right) \le O\left(t^{\frac{1}{2}+\varepsilon}\right) \le O\left(t^{1-\frac{1}{k}+\varepsilon}\right)$. The result follows. □

Applying Theorem 1, Theorem 11, Lemma 13, and Lemma 14, we obtain the following.

**Corollary 15.** *With probability* $\ge 1 - O\left(\frac{1}{t}\right)$*, it holds that*

*i.* $E[Z_\ell^t] - O\left(\sqrt{t\log t}\right) \le Z_\ell^t \le E[Z_\ell^t] + O\left(\sqrt{t\log t}\right)$*, and*

*ii.* $E[Z_\ell^t] - O\left(t^{1-1/k+\varepsilon}\right) \le Y_\ell^t \le E[Z_\ell^t] + O\left(t^{1-1/k+\varepsilon}\right)$.

Note that the concentration error term, $O(\sqrt{t\log t})$, is upper bounded by $R_t$, for each $k \ge 2$. Also, the corollary is vacuous if $\ell > t^{1/(k+2)}$.

## 2.7 Compressibility of our model

We now analyze the number of bits needed to compress the graphs generated by our model. Recall that the web graph has a natural embedding on the line via the URL ordering that experimentally gives very good compression [12,13]. Our model generates a web-like random graphs and an embedding "à-la-URL" on the line. We work with the following *BV-like compression scheme*: a node at position $p$ on the line stores its list of successors at positions $p_1, \ldots, p_k$ as a list $(p_1 - p, \ldots, p_k - p)$ of compressed integers. An integer $i \neq 0$ will be compressed using $O\left(\log\left(|i|+1\right)\right)$ bits, using Elias $\gamma$-code, for instance [107]. We show that our graphs can be compressed using $O(1)$ bits per edge using above scheme.

**Theorem 16.** *The above BV-like scheme compresses the graphs generated by our model using* $O(n)$ *bits, with probability at least* $1 - O\left(\frac{1}{n}\right)$.

*Proof.* Let $\varepsilon > 0$ be a small constant. At time $n$, consider the number of edges of length at most $L = \lceil n^\varepsilon \rceil$. Note that by Corollary 15, for each $1 \le \ell \le L$, it holds that $|Y_\ell^n - E[Z_\ell^n]| \le O\left(n^{1-1/k+\varepsilon}\right)$, with probability $1 - O\left(n^{-1}\right)$. For the rest of the proof, we implicitly condition on these events.

Lower bounding $E[Z_\ell^n]$ as in Theorem 11, we obtain the following lower bound on the number of edges of length $\le L$, using standard algebraic manipulation and Lemma[5] 2

$$\begin{aligned} S &\ge \sum_{\ell=1}^{L}\left(\frac{\Gamma\left(\ell+1-\frac{1}{k}\right)}{\Gamma\left(2-\frac{1}{k}\right)\Gamma(\ell+2)}\cdot n - c - O\left(n^{1-1/k+\varepsilon}\right)\right) \\ &\ge nk\left(1 - \frac{\Gamma\left(L+2-\frac{1}{k}\right)}{\Gamma(L+2)\Gamma\left(2-\frac{1}{k}\right)}\right) - O\left(L\cdot n^{1-1/k+\varepsilon}\right) \\ &\ge nk - O\left(n\cdot k\cdot L^{-1/k}\right) - O\left(L\cdot n^{1-1/k+\varepsilon}\right) \ge nk - O\left(n^{1-\varepsilon_1}\right), \end{aligned}$$

where $\varepsilon_1$ is a small constant.

[5] Which we use to conclude that $\frac{1}{\Gamma\left(2-\frac{1}{k}\right)}\sum_{\ell=1}^{L}\frac{\Gamma\left(\ell+1-\frac{1}{k}\right)}{\Gamma(\ell+2)} = k\cdot\left(1 - \frac{\Gamma\left(L+2-\frac{1}{k}\right)}{\Gamma(L+2)\Gamma\left(2-\frac{1}{k}\right)}\right)$.

At time $n$, the total number of edges of the graph is $nk$. Thus the number of edges of length more than $L$ is at most $O\left(n^{1-\varepsilon_1}\right)$ (notice how, for this argument to work, it is crucial to have a very strong bound on the behavior of the $Y_\ell^n$ random variables; this is why we used the Gamma function in their expressions). The maximum edge length is $O(n)$ and so each edge can be compressed in $O(\log n)$ bits. The overall contribution, in terms of bits, of the edges longer than $L$ will then be $o(n)$.

Now, we calculate the bit contribution $B$ of the edges of length at most $L$.

$$\begin{aligned} B &\leq \sum_{\ell=1}^{L} \left( O\left(\log\left(\ell+1\right)\right) \left( \frac{\Gamma\left(\ell+1-\frac{1}{k}\right)}{\Gamma\left(2+\frac{1}{k}\right)\Gamma(\ell+2)} n + c + O\left(n^{1-1/k+\varepsilon}\right)\right)\right) \\ &\leq n \cdot O\left(\sum_{\ell=1}^{L} \log\left(\ell+1\right) \cdot \ell^{-1-1/k}\right) + O\left(L \cdot n^{1-1/k+\varepsilon} \cdot \log L\right) \leq O(n), \end{aligned}$$

where the penultimate inequality follows since the $\frac{\Gamma(\cdots)}{\Gamma(\cdots)\Gamma(\cdots)}$ fraction can be upper bounded by $O(\ell^{-1-1/k})$, and the last inequality from $O\left(\ell^{-1-2\varepsilon} \cdot \log \ell\right) \leq O\left(\ell^{-1-\varepsilon}\right)$ and from the convergence of the Riemann series. The proof is complete. □

Thus, given an ordering of nodes, we can compress the graph to use $O(1)$ bits per edge using a very simple linear-time algorithm. A natural question is if it is still possible to compress this graph *without* knowing the ordering. We show that this is still possible.

**Theorem 17.** *The graphs generated by our model can be compressed using $O(n)$ bits in linear time, even if ordering of the nodes is not available.*

## 2.8 Appendix

### 2.8.1 Compressibility: Labeled vs unlabeled processes

We give some intuition on why one cannot preclude an incompressible directed/labeled graph from becoming very compressible after removing the labels and directions.

Consider the following (non-graph related) random process.

Suppose we have two bins $B_1$ and $B_2$ and suppose we toss two independent fair coins $c_1, c_2$. If $c_1$ is head (resp., tail), then we place a white (resp., black) ball in $B_1$. Analogously, if $c_2$ is head (resp., tail), then we place a white (resp., black) ball in $B_2$. Now, consider the r.v. $X$ describing the status of the two distinguishable bins. It has four possible outcomes $((W,W),(W,B),(B,W),(B,B))$ and each of them is equally likely; thus $\mathrm{H}(X) = 2$. Now, suppose we empty the bins $B_1$ and $B_2$ on a table, and let $Y$ be the event describing the status of the table after the two balls are placed on it. $Y$ has three possible outcomes $(\{W,W\},\{W,B\},\{B,B\})$ and its entropy is $\mathrm{H}(Y) = \frac{3}{2} < 2 = \mathrm{H}(X)$.

Similarly, for $n$ coins and $n$ bins, we have $\mathrm{H}(X_n) = n$ and $\mathrm{H}(Y_n) = \Theta(\log n)$. Thus, we can get an exponential gap between the entropies of the labeled (i.e., each outcome can be matched to the coin toss that determined it) and unlabeled processes.

For a graph-related example, suppose we choose a labeled transitive tournament on $n$ nodes u.a.r. There are $n!$ such graph, each equally likely, so that the entropy would be $\log(n!) = \Theta(n \log n)$. On the other hand, there exists a single unlabeled transitive tournament, i.e., the entropy of the unlabeled version is zero.

### 2.8.2 Incompressibility of other models

#### Incompressibility of the ACL model

We recall the ACL model (model A in [2]). This model ($\mathsf{acl}[\alpha]$) is parametrized by some $\alpha \in (0,1)$. At time 1, the graph consists of a single node. At time $t+1$, a coin is tossed: with probability $1-\alpha$, a new node is added to the graph and with probability $\alpha$, an edge from $x$ to $y$ is added to the graph, where node $x$ is chosen with probability proportional to the outdegree of $x$, while node $y$ is chosen randomly with probability proportional to the indegree of $y$.

We assume that $\alpha > 1/2$. This is because the edge density of the graph generated by model is $\alpha/(1-\alpha)$, w.h.p.; if $\alpha < 1/2$, then there are many more nodes than edges, an uninteresting case both in theory and in practice. Under this assumption, it is easy to show $\mathrm{H}(P_n^{\mathsf{acl}[\alpha]}) = \Omega(n \log n)$.

*Proof of Theorem 5.* Let $\alpha > 1/2$ be the parameter of the $\mathsf{acl}[\alpha]$ model. Let $\mathcal{G}'_n$ be the set[6] of all time-labeled graphs, that can be generated by $\mathsf{acl}[\alpha]$ model in $n$ time steps, where the label represents the time when a node or an edge was added to the graph. Let $\mathcal{H}'_n$ be the set of all undirected and unlabeled graphs that can be obtained by removing the orientation and (time-)labels from the graphs in $\mathcal{G}'_n$.

Let $P_n'^{\mathsf{acl}[\alpha]} : \mathcal{G}'_n \to [0,1]$ denote the probability distribution induced on $\mathcal{G}'_n$ by the model $\mathsf{acl}[\alpha]$. We define the following two events.

$\xi$: the number of edges is $\alpha n \pm o(n)$, while the number of nodes is $(1-\alpha)n \pm o(n)$, and

$\xi'$: the number of edges going from a node of $O(1)$ outdegree to a node of $O(1)$ indegree is at least $(\alpha - \varepsilon)n$, for some $\varepsilon > 0$ to be fixed later.

Our plan is first show (Lemma 18) that $\xi \wedge \xi'$ occurs with probability $1 - o(1)$. Let $\mathcal{G}'^*_n \subseteq \mathcal{G}'_n$ be the subset of $\mathcal{G}'_n$ containing the graphs satisfying $\xi \wedge \xi'$. Then, with the notation of Lemma 3, it holds that $P^+ = 1 - o(1)$. We will then show (Lemma 19) that $P^* = \max_{G' \in \mathcal{G}'^*_n} P_n'^{\mathsf{acl}[\alpha]}(G') \leq n^{-(2\alpha-\varepsilon)n}$. Given these, we can complete the proof as follows. Let $\varphi' : \mathcal{G}'_n \to \mathcal{H}'_n$, be the map that removes edge and node labels from the graphs of $\mathcal{G}'_n$. As before, $Q_n'^{\mathsf{acl}[\alpha]}(H') = \sum_{\varphi'(G')=H'} P_n'^{\mathsf{acl}[\alpha]}(G')$. Note that for each $H'$ we have that $|\varphi'(G')| \leq n!$ (as each element of the graph has one label out of the set $\{1, \ldots, n\}$). Thus,

$$\max_{G' \in \mathcal{G}'^*_n} P_n'^{\mathsf{acl}[\alpha]}(G) \leq n! \cdot n^{-(2\alpha-\varepsilon)n} \leq n^{-(2\alpha-\varepsilon)n+n} = n^{(1-2\alpha+\varepsilon)n}.$$

The proof can be concluded with an application of Lemma 3. □

**Lemma 18.** $\mathrm{P}\,\xi \wedge \xi' = 1 - o(1)$.

*Proof.* By a simple Chernoff bound, $\mathrm{P}\,\xi = 1 - o(1)$. Thus it suffices to show that $\mathrm{P}\,\xi' = 1 - o(1)$.

Let $X_i^t$ ($Y_i^t$) be the r.v. denoting the number of nodes having indegree (outdegree) $i$ at time $t$. ACL show that

$$\frac{\mathrm{E}\left[X_i^t\right]}{t} = \frac{\mathrm{E}\left[Y_i^t\right]}{t} = \frac{1-\alpha}{\alpha}\Gamma\left(1+\frac{1}{\alpha}\right)\frac{\Gamma(i)}{\Gamma\left(i+1+\frac{1}{\alpha}\right)} \pm O\left(\frac{1}{t}\right),$$

and that

$$\mathrm{P}\left|X_i^t - \mathrm{E}\left[X_i^t\right]\right| > \sqrt{2t}\log n + 2 < \exp\left(-\log^2 n\right),$$

$$\mathrm{P}\left|Y_i^t - \mathrm{E}\left[Y_i^t\right]\right| > \sqrt{2t}\log n + 2 < \exp\left(-\log^2 n\right).$$

Note that, by union bound, each of the RVs $X_i^t, Y_i^t$ can be shown to deviate from their mean by at most the stated error term w.h.p.

Let $j$ be an integer to be fixed later. An edge is *good* if it goes from a node of outdegree $\leq j$ to a node of indegree $\leq j$. Let us denote by $Z_j^t$ the number of good edges at time $t$. Note that $Z_j^{t-1} + 1 \geq Z_j^t \geq Z_j^{t-1} - 2j$. This is because at most one edge is added in a single step and adding an edge changes the degree of at most 2 nodes. Thus, the number of good edges can decrease at most $2j$ in a single step, i.e., $Z_j^t$ satisfies the $(2j)$-Lipschitz condition.

Then,

$$\mathrm{E}\left[Z_j^t\right] = \mathrm{E}\left[Z_j^{t-1}\right] + \mathrm{P}\,Z_j^t = Z_j^{t-1} + 1 - \sum_{i=1}^{2j} i\,\mathrm{P}\,Z_j^t = Z_k^{t-1} - i.$$

In order to increase the number of good edges, a node of indegree $< j$ and a node of outdegree $< j$ must be chosen as the ending and the starting point of the new edge.

$$\mathrm{P}\,Z_j^t = Z_j^{t-1} + 1 = \alpha\frac{\left(\sum_{i=1}^{j-1} iX_i^{t-1}\right)\left(\sum_{i=1}^{j-1} iY_i^{t-1}\right)}{(t-1)^2}.$$

[6]Note that here it would be unnatural to consider the previously defined class $\mathcal{G}_n$, as the number of nodes in the $\mathsf{acl}[\alpha]$ model is an r.v. The same holds for $\mathcal{H}_n$.

For the number of good edges to decrease, either the origin of the new edge has outdegree $j$, or the destination of the new edge has indegree $j$. Thus,

$$\mathrm{P}\, Z_j^t < Z_j^{t-1} \leq \frac{jX_j^{t-1}}{t-1} + \frac{jY_j^{t-1}}{t-1}.$$

By simple calculations,

$$\sum_{i=1}^{2j} i\,\mathrm{P}\, Z_j^t = Z_j^{t-1} - i \leq 2j \sum_{i=1}^{2j} \mathrm{P}\, Z_j^t = Z_j^{t-1} - i \leq 2j^2 \frac{X_j^{t-1} + Y_j^{t-1}}{t-1}.$$

Thus,

$$\mathrm{E}\left[Z_j^t\right] \geq \mathrm{E}\left[Z_j^{t-1}\right] + \alpha \frac{\mathrm{E}\left[\left(\sum_{i=1}^{j-1} iX_i^{t-1}\right)\left(\sum_{i=1}^{j-1} iY_i^{t-1}\right)\right]}{(t-1)^2} - 2j^2 \frac{\mathrm{E}\left[X_j^{t-1}\right] + \mathrm{E}\left[Y_j^{t-1}\right]}{t-1}.$$

With probability $1 - o(1)$, for all $\log^2 n \leq t \leq n$ and $1 \leq i \leq j-1$, we have

$$X_i^t = Y_i^t = (1 \pm o(1)) \frac{1-\alpha}{\alpha} \Gamma\left(1 + \frac{1}{\alpha}\right) \frac{\Gamma(i)}{\Gamma\left(i + 1 + \frac{1}{\alpha}\right)} t.$$

Thus w.h.p., for all $t \geq \log^2 n$,

$$\sum_{i=1}^{j-1} i \frac{X_i^t}{t} = \sum_{i=1}^{j-1} i \frac{Y_i^t}{t} = 1 - \frac{\Gamma(j+1)\Gamma(1+\frac{1}{\alpha})}{\Gamma(j+\frac{1}{\alpha})} \pm o(j^2).$$

As $j$ is a constant, the error term is $o(1)$. Then,

$$\mathrm{E}\left[Z_j^t\right] \geq \mathrm{E}\left[Z_j^{t-1}\right] + \alpha \left(1 - \frac{\Gamma(j+1)\Gamma(1+\frac{1}{\alpha})}{\Gamma(j+\frac{1}{\alpha})}\right)^2 - 4\frac{1-\alpha}{\alpha}\Gamma\left(1+\frac{1}{\alpha}\right) \frac{j^2\Gamma(j)}{\Gamma(j+1+\frac{1}{\alpha})} \pm o(1).$$

Note that, as $j$ grows, both $\frac{\Gamma(j+1)\Gamma(1+\frac{1}{\alpha})}{\Gamma(j+\frac{1}{\alpha})}$ and $\frac{j^2\Gamma(j)}{\Gamma(j+1+\frac{1}{\alpha})}$ tend to 0. That is, for each $\varepsilon_1$, there exists a $j = j(\varepsilon_1)$ such that

$$\mathrm{E}\left[Z_j^t\right] \geq \mathrm{E}\left[Z_j^{t-1}\right] + (1-\varepsilon_2)\alpha. \tag{2.1}$$

For each $j$, and for each $t$, we will define a $B_j^t$ in such a way that, w.h.p., $B_j^t \leq \mathrm{E}\left[Z_j^t\right]$. Let $B_j^t = 0$ for $t \leq \lceil \log^3 n \rceil$ so that the base case is trivially true. Define $B_j^t = (t - \lceil \log^3 n \rceil)(1-\varepsilon_2)\alpha$. This definition satisfies $B_j^t \leq \mathrm{E}\left[Z_j^t\right]$ for all $t$ — this can be shown by induction on (2.1). Recall that all these hold w.h.p. As we have already argued, the r.v. $Z_k^t$ satisfies the $(2j)$-Lipschitz condition, i.e., using [2, Lemma 1], $Z_j^t = \mathrm{E}\left[Z_j^t\right] \pm o(t) \geq t(1-\varepsilon_2)\alpha$, for every $t \geq \lceil \log^3 n \rceil$, w.h.p.

In particular for any $\varepsilon_2 > 0$ there exists a $j = j(\varepsilon)$ s.t. $Z_j^n \geq n(1-\varepsilon_2)\alpha$, w.h.p. □

**Lemma 19.** *Conditioned on $\xi \wedge \xi'$, $\max_{G' \in \mathcal{G}_n'^*} P_n^{\prime \mathsf{acl}[\alpha]}(G') \leq n^{-(2\alpha-\varepsilon)n}$.*

*Proof.* Since we condition on $\xi'$, there are at least $n(1-\varepsilon_2)\alpha$ good edges. These good edges are labeled with their order of arrival. For $i \geq \varepsilon_3\alpha n$, the probability that the $i$-th arrived edge is good is at most $\frac{j^2}{i^2} \leq \frac{j^2}{(\varepsilon_3\alpha n)^2}$.

The probability that all the edges with label at least $\varepsilon_3\alpha n$ are good is at most

$$\left(\frac{j}{\varepsilon_3\alpha n}\right)^{2(1-\varepsilon_3)\alpha n} \leq \left(\frac{j}{\varepsilon_3\alpha}\right)^{2(1-\varepsilon_3)\alpha n} n^{-2(1-\varepsilon_3)\alpha n} \leq n^{-(2\alpha-\varepsilon)n}.$$

Thus, the maximum probability of generating a graph in $\mathcal{G}_n'^*$, conditioned on $\xi \wedge \xi'$, is at most $n^{-(2\alpha-\varepsilon)n}$. □

**Incompressibility of the copying model**

We now turn our attention to the (linear growth) copying model ($\mathsf{copy}[\alpha, \mathsf{k}]$) of Kumar et al. [69]. This model is parametrized by an integer $k \geq 1$ and an $\alpha \in (0, 1)$. Here, $k$ represents the outdegree of nodes and $\alpha$ determines the "copying rate" of the graph. At time $t = 1$, the graph consists of a single node with $k$ self-loops. At time $t > 1$,

(1) a new node $x_t$ is added to the graph;

(2) a node $x$ is chosen uniformly at random among $x_1, \ldots, x_{t-1}$; and

(3) for each $i = 1, \ldots, k$, a $\alpha$-biased coin is flipped: with probability $\alpha$, the $i$-th outlink of $x_t$ is chosen uniformly at random from $x_1, \ldots, x_{t-1}$ and with probability $1 - \alpha$, the $i$-th outlink of $x_t$ will be equal to the $i$-th outlink of $x$, i.e., the $i$-th outlink will be "copied".

*Proof of Theorem 6.* We start by noting that the copying model with outdegree $k$ can be completely described by $k$ independent versions of the copying model with outdegree 1. We use $\mathsf{copy}[\alpha, \mathsf{k}]$ to denote the copying model with $k$ outlinks, $\mathcal{G}_{n,k}$ for the set of labeled[7] graphs on $n$ nodes that can be generated by $\mathsf{copy}[\alpha, \mathsf{k}]$, and $\mathcal{H}_{n,k}$ for the set of unlabeled graphs that can be obtained by removing labels and orientations from the graphs in $\mathcal{G}_{n,k}$.

We start with the case $k = 1$. Let $\mathrm{E}\left[X_i^t\right]$ be the expected indegree at time $t$ of the node inserted at time $i \leq t$. Then,

$$\mathrm{E}\left[X_i^t\right] = \begin{cases} 0 & t = k \\ \mathrm{E}\left[X_i^{t-1}\right]\left(1 + \frac{1-\alpha}{t-1}\right) + \frac{\alpha}{t-1} & t > i. \end{cases}$$

Note that $\mathrm{E}\left[X_i^t\right] = \frac{\alpha\Gamma(t+1-\alpha)\Gamma(i)}{(1-\alpha)\Gamma(i+1-\alpha)\Gamma(t)} - \frac{\alpha}{1-\alpha}$. We now show that $X_i^t$ satisfies a $O(1)$-Lipschitz condition, with the constant depending on $i$ and $\alpha$.

Let $Y_j^t$ denote the number of edges "copied", directly or indirectly, from the $j$-th edge until time $t \geq j$. Precisely, let us define the singleton $S_j^j = \{e_j\}$ containing the $j$-th added edge. The set $S_j^t$, $t > j$, will be defined as follows: if the $t$-th edge $e_t$ was copied from some edge in $S_j^{t-1}$, then $S_j^t = \{e_t\} \cup S_j^{t-1}$, otherwise $S_j^t = S_j^{t-1}$. With this notation, $Y_j^t = |S_j^t|$.

We now use the following concentration bound [44].

**Theorem 20 (Method of average bounded differences).** *Suppose $f$ is some function of (possibly dependent) r.v.'s $X_1, \ldots, X_n$. Suppose that, for each $i = 1, \ldots, n$, there exists a $c_i$ such that, for all pairs $x_i, x_i'$ of possible values of $X_i$, and for any assignment $X_1 = x_1, \ldots, X_{i-1} = x_{i-1}$, it holds that $|E - E'| \leq c_i$, where*

$$E = \mathrm{E}\left[f(X_1, \ldots, X_n) \mid X_i = x_i, X_{i-1} = x_{i-1}, \ldots, X_1 = x_1\right],$$

$$E' = \mathrm{E}\left[f(X_1, \ldots, X_n) \mid X_i = x_i', X_{i-1} = x_{i-1}, \ldots, X_1 = x_1\right].$$

*Let $c = \sum_{i=1}^n c_i^2$. Then,*

$$\mathrm{P}\left|f(X_1, \ldots, X_n) - \mathrm{E}\left[f(X_1, \ldots, X_n)\right]\right| > t \leq 2\exp\left(-\frac{t^2}{2c}\right).$$

Let $j$ be fixed. Our goal is to bound $c_j$ in such a way that Theorem 20 can be applied. Easily, $\mathrm{E}\left[Y_j^t\right] = \mathrm{E}\left[Y_j^{t-1}\right] \cdot \left(1 + \frac{1-\alpha}{t-1}\right)$, for $t > j$, and $Y_j^j = 1$. Then, it is easy to verify that $\mathrm{E}\left[Y_j^t\right] = \frac{\Gamma(t+1-\alpha)\Gamma(j)}{\Gamma(t)\Gamma(j+1-\alpha)}$.

Suppose we want to bound the degree of the $i$-th node $x_i$. Then, we are interested in bounding the maximum expected change $c_j$ in the degree $X_i^t$ of $x_i$ over the possible choices of the $j$-th edge, for $j = i + 1, \ldots, n$. We have $c_j \leq 2\,\mathrm{E}\left[Y_j^n\right]$.

[7] Nodes are labeled with $1, \ldots, n$ and, for each node, its outlinks are labeled with $1, \ldots k$.

Let us consider $c = \sum_{j=i+1}^{n} c_j^2$. We have

$$\begin{aligned} c &\leq \sum_{j=i+1}^{n} \left(2\,\mathrm{E}\left[Y_j^n\right]\right)^2 \\ &\leq 4\left(\frac{\Gamma(n+1-\alpha)}{\Gamma(n)}\right)^2 \cdot \sum_{j=i+1}^{n} \left(\frac{\Gamma(j)}{\Gamma(j+1-\alpha)}\right)^2 \\ &\leq a \cdot n^{2-2\alpha} \sum_{j=i+1}^{n} \frac{1}{j^{2-2\alpha}}, \end{aligned}$$

for some large enough constant $a > 0$. Thus we obtain,

$$c \leq \begin{cases} a \cdot n^{2-2\alpha} \cdot \frac{i^{2\alpha-1} - n^{2\alpha-1}}{1-2\alpha} & \alpha < \frac{1}{2} \\ a \cdot n \cdot (\log n + 1) & \alpha = \frac{1}{2} \\ a \cdot n^{2-2\alpha} \cdot \frac{n^{2\alpha-1} - i^{2\alpha-1} + 1}{2\alpha-1} & \alpha > \frac{1}{2} \end{cases}$$

Let us fix $i = \lceil \varepsilon n \rceil$. Then,

$$c \leq \begin{cases} a \cdot n^{2-2\alpha} \cdot n^{2\alpha-1} \frac{1-\varepsilon^{1-2\alpha}}{\varepsilon^{1-2\alpha}} = a \cdot n \cdot \frac{1-\varepsilon^{1-2\alpha}}{\varepsilon^{1-2\alpha}} & \alpha < \frac{1}{2} \\ a \cdot n \cdot (\log n + 1) & \alpha = \frac{1}{2} \\ a \cdot n^{2-2\alpha} \cdot n^{2\alpha-1} \frac{1}{2\alpha-1} + n^{2-2\alpha} = a \cdot n \cdot \frac{1}{2\alpha-1} + o(n) & \alpha > \frac{1}{2} \end{cases}$$

Thus, $c \leq O(n \log n)$. Applying Theorem 20, we get

$$\mathrm{P}\left|X_i^t - \mathrm{E}\left[X_i^t\right]\right| \geq 2\sqrt{c \log n} \leq 2\exp\left(\frac{4c \log n}{2c}\right) = 2\exp(2\log n) = \frac{2}{n^2}.$$

By the union bound, with probability $1 - O(1/n)$, each node $i = \lceil \varepsilon n \rceil, \lceil \varepsilon n \rceil + 1, \ldots, n$ will have degree upper bounded by $O(\sqrt{n \log n})$ (note that the expected degree of these nodes is constant). Conditioning (as in the proof of Theorem 4) on this event we obtain $\mathcal{G}^*_{n,1} \subseteq \mathcal{G}_{n,1}$, $P^+ = 1 - o(1)$, and for $k = 1$,

$$\max_{G \in \mathcal{G}^*_{n,1}} P_n^{\mathsf{copy}[\alpha,1]}(G) \leq \left(O\left(\sqrt{\frac{\log n}{n}}\right)\right)^{\alpha n}.$$

Now let us consider $\mathsf{copy}[\alpha, \mathsf{k}]$, with $k > 1$. Since outlinks are chosen independently, it holds that

$$\max_{G \in \mathcal{G}^*_{n,k}} P_n^{\mathsf{copy}[\alpha,\mathsf{k}]}(G) \leq \left(O\left(\sqrt{\frac{\log n}{n}}\right)\right)^{k\alpha n}.$$

For constant $k > 2/\alpha$, this upper bound is less than $n^{-(1+\varepsilon)n}$ for some constant $\varepsilon > 0$.

To show a lower bound on $\mathrm{H}(Q_n^{\mathsf{copy}[\alpha,\mathsf{k}]})$, we once again upper bound $\left|\varphi^{-1}(H)\right|$, for $H \in \mathcal{H}_{n,k}$. We proceed as in the proof of Theorem 4. Given $H$, for each of its nodes $v$, it is easy to determine which of the edges incident to $v$ were its outlinks in all the $G$'s such that $\varphi(G) = H$ (this can be done by induction, noting that a node of degree $k$ in $H$ in would have had in-degree 0 in $G$). As there are exactly $k$ labels for the outlinks of each node, and the number of nodes is $n$, we have that, for each $H \in \mathcal{H}_{n,k}$, $\left|\varphi^{-1}(H)\right| \leq n! \cdot (k!)^n$. The proof can be concluded as in Theorem 4. □

**Incompressibility of the Kronecker multiplication model**

We now turn our attention to the Kronecker multiplication model (krm) of Leskovec et al. [74].

Given two matrices $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{m \times m}$, their Kronecker product $A \otimes B$ is an $nm \times nm$ matrix

$$A \otimes B = \begin{pmatrix} a_{1,1}B & a_{1,2}B & \cdots & a_{1,n}B \\ a_{2,1}B & a_{2,2}B & \cdots & a_{2,n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1}B & a_{n,2}B & \cdots & a_{n,n}B \end{pmatrix},$$

where $A = \{a_{i,j}\}$ and $a_{i,j}B$ is the usual scalar product.

The Kronecker multiplication model is parametrized by a square matrix $M \in [0,1]^{\ell \times \ell}$, and a number $s$ of multiplication "steps". The graph will be composed by $\ell^s$ nodes. The edges are generated as follows. For each couple of distinct nodes $(i,j)$ in the graph an edge going from $i$ to $j$ will be added independently with probability $M^{[s]}_{i,j}$, where $M^{[s]}_{i,j} = \underbrace{M \otimes M \otimes \cdots \otimes M}_{s \text{ times}}$.

It is clear that for some choices of the matrix $M$, the graph *will* be compressible. Indeed, if $M$ has only 0/1 values then the random graph has zero entropy, as its construction is completely deterministic. On the other hand, we show here that there exists some $M$ that makes the graph incompressible. Indeed, even some $2 \times 2$ matrix $M$ would generate graphs requiring at least $\Omega(\log n)$ bits per edge. (Note that a $1 \times 1$ matrix can only produce graphs containing a single node.)

*Proof of Theorem 7.* Consider the original directed version of the graph. Note that $M^{[s]} = \alpha^s \cdot J_{\ell^s}$. Thus the events "the edge $i \to j$ is added to the graph" are i.i.d. trials, each having probability of success $\alpha^s$.

In the undirected and simple version of the graph, the events "the edge $\{i,j\}$ is added to the graph", for $i \neq j$, are again i.i.d. trials, each of probability $\beta = 1 - (1-\alpha^s)^2 = \Theta(\alpha^s)$. Thus we obtain an Erdös–Renyi $G_{n,p}$ graph with $n = \ell^s$ and $p = \Theta(\alpha^s)$. By a Chernoff bound, $m = \Theta(n^2 p)$, w.h.p. Now,

$$m = \Theta(n^2 p) = \Theta(n \cdot (\ell\alpha)^s) = \Theta\left(n \cdot \left(\ell^{1+\log_\ell \alpha}\right)^s\right) = \Theta\left(n \cdot (\ell^s)^{1-\log_\ell \frac{1}{\alpha}}\right) = \Theta\left(n^{2-\log_\ell \frac{1}{\alpha}}\right).$$

By $\alpha > \ell^{-1}$, we obtain $\log_\ell \frac{1}{\alpha} < 1$; thus $m = \Theta(n^2 p)$ is a polynomial in $n$ of degree $> 1$.

Recall that, for Lemma 3 to apply, we need to find a subset $\mathcal{H}^*_n \subseteq \mathcal{H}_n$, having large total probability $P^+$, and such that each graph in $\mathcal{H}^*_n$ has probability upper bounded by a (small) $P^*$. The condition $\{m = \Theta(n^2 p)\}$ determines our $\mathcal{H}^*_n$, giving us $P^+ = 1 - o(1)$. To upper bound $P^*$, note that each labeled version of each graph in $\mathcal{H}^*_n$ has probability $\leq p^{\Theta(n^2 p)} \leq 2^{-\Theta(s \cdot n^2 p)}$. There are at most $n! \leq 2^{O(n \log n)}$ many labeled versions of each fixed graph in $\mathcal{H}^*_n$. Thus,

$$P^* \leq 2^{O(n \log n) - \Theta(s \cdot n^2 p)} = 2^{-\Theta(s \cdot n^2 p)}.$$

By Lemma 3, we have that $\mathrm{H}(Q^{\mathsf{krm}}_n) \geq P^+ \log(1/P^*) \geq \Theta(s \cdot n^2 p)$. Noting that $s = \Theta(\log n)$ and $m = \Theta(n^2 p)$ concludes the proof. □

### Incompressibility of Kleinberg's small-world model

Recall Kleinberg's small-world model[8] (kl) [66, 67] on the line, with nodes $1, \ldots, n$. A directed labeled random graph is generated by the following stochastic process. Each node $x$ independently chooses a node $y$ with probability proportional to $1/(|x-y|)$ and adds the directed edge $x \to y$; these are the so-called *long-range* edges. In addition, the node $x$ has (fixed) directed edges to its neighbors $x-1$ and $x+1$ (the *short-range* edges).

For simplicity, we start by proving the following weaker result. After the proof, we will comment on how one can obtain the stronger claim of Theorem 8.

**Lemma 21.** $\mathrm{H}(Q^{\mathsf{kl}}_n) = \Omega(n \log\log n)$.

*Proof.* Note that in Kleinberg's one-dimensional model, the normalization factor for the probability distribution that generates long-range edges is $\Theta(\log n)$. Hence, for every node $x$, the maximum probability of choosing a particular long-range edge $x \to y$ is at most $c_1/\log n$, for some constant $c_1$. Since each node chooses edges independently, the maximum probability of generating any labeled $n$-node graph $O((c_1/\log n)^n)$, i.e., $\max_{G \in \mathcal{G}_n} P^{\mathsf{kl}}_n(G) \leq (c_1/\log n)^n$. Using Lemma 3, we conclude $\mathrm{H}(P^{\mathsf{kl}}_n) = \Omega(n \log\log n)$.

To get a lower bound on $\mathrm{H}(Q^{\mathsf{kl}}_n)$, we first obtain an upper bound on the number $\rho(H)$ of Hamiltonian paths in an undirected graph $H$ with $m$ edges (this upper bound will trivially hold for directed graphs as well). Suppose that $H$ has degree sequence $d_1 \geq \cdots \geq d_n$, with $2m = \sum_{i=1}^n d_i$. Clearly, $\rho(H) \leq n \cdot \prod_{i=1}^n d_i$,

[8] Note that an important difference between Kleinberg's small-world model and other models gere considered lies in their degree distribution. Nodes' degrees in Kleinberg's model are upper bounded by $O(\log n)$ w.h.p.; the other models we consider here have a power law degree distribution, and thus nodes of polynomial degree, w.h.p.

where the leading $n$ is for the different choices of the starting node. Applying the AM-GM inequality ($\sqrt[n]{\prod_{i=1}^{n} x_i} \leq \frac{1}{n}\sum_{i=1}^{n} x_i$, for non-negative $x_i$'s), we have that $\rho(H) \leq n \cdot \prod_{i=1}^{n} d_i \leq n \cdot (2m/n)^n$.

Let $H \in \mathcal{H}_n$. By just considering all possible permutations of the node labels, we can see that $|\varphi^{-1}(H)| \leq n!$. However, not all permutations are valid. In particular, a valid permutation preserves adjacency, hence the number of valid permutations is upper bounded by the number of Hamiltonian paths in $H$. Since $m = O(n)$ in kl, by the above argument, $\rho(H) \leq c_2^n$, for some constant $c_2$. Thus, $|\varphi^{-1}(H)| \leq c_2^n$. We have

$$Q_n^{\mathsf{kl}}(H) = \sum_{\varphi(G)=H} P_n^{\mathsf{kl}}(G) \leq |\varphi^{-1}(H)| \cdot (\max_{G \in \mathcal{G}_n} P_n^{\mathsf{kl}}(G)) \leq c_2^n \left(\frac{c_1}{\log n}\right)^n = O\left(\frac{1}{\log n}\right)^n.$$

The proof is complete by appealing to Lemma 3. □

The above lower bound can be improved as follows. First, we only consider graphs in which $\Omega(n)$ of the edges exist between nodes that are $n^{\Omega(1)}$ apart. Using a Chernoff bound it is easy to show that the graphs generated by Kleinberg's model satisfy this property w.h.p. (i.e., the $P^+$ of the Lemma 3 is $\Omega(1)$). It can then be shown that the maximum probability of generating any one of these graphs is at most $P^* = n^{-\Omega(n)}$. Once again, applying Lemma 3, we can obtain $\mathrm{H}(Q_n^{\mathsf{kl}}) = \Omega(n \log n)$; the details of this improvement are omitted in this version. Finally, we note that the similar incompressibility bounds can be obtained for the rank-based friendship model [77].

### 2.8.3 Proof of Lemma 2

*Proof.* We start by giving an expression of $\frac{\Gamma(i+a)}{\Gamma(i+b)}$, for $i \geq 1$, that we will use to telescope the sum. Consider the following chain of equations:

$$\begin{aligned}
b-a-1 &= (i+b-1)-(i+a) \\
\frac{\Gamma(i+a)}{\Gamma(i+b)} \cdot (b-a-1) &= \frac{\Gamma(i+a)}{\Gamma(i+b)} \cdot (i+b-1) - \frac{\Gamma(i+a)}{\Gamma(i+b)} \cdot (i+a) \\
\frac{\Gamma(i+a)}{\Gamma(i+b)} \cdot (b-a-1) &= \frac{\Gamma(i+a)}{\Gamma(i+b-1)} - \frac{\Gamma(i+a+1)}{\Gamma(i+b)} \\
\frac{\Gamma(i+a)}{\Gamma(i+b)} &= \frac{1}{b-a-1} \cdot \left(\frac{\Gamma(i+a)}{\Gamma(i+b-1)} - \frac{\Gamma(i+a+1)}{\Gamma(i+b)}\right)
\end{aligned}$$

Then, by telescoping on the sum terms, we get:

$$\begin{aligned}
\sum_{i=1}^{t} \frac{\Gamma(i+a)}{\Gamma(i+b)} &= \frac{\left(\frac{\Gamma(a+1)}{\Gamma(b)} - \frac{\Gamma(a+2)}{\Gamma(b+1)}\right) + \left(\frac{\Gamma(a+2)}{\Gamma(b+1)} - \frac{\Gamma(a+3)}{\Gamma(b+2)}\right) + \cdots + \left(\frac{\Gamma(a+t)}{\Gamma(b+t-1)} - \frac{\Gamma(a+t+1)}{\Gamma(b+t)}\right)}{b-a-1} \\
&= \frac{\frac{\Gamma(a+1)}{\Gamma(b)} - \frac{\Gamma(a+t+1)}{\Gamma(b+t)}}{b-a-1},
\end{aligned}$$

proving the claim. □

### 2.8.4 Proof of Theorem 9

*Proof.* For now, assume $t > t_0$. Let $x$ be the new node, and let $y$ be the node we will copy edges from; recall that $y$ is chosen u.a.r. First, we focus on the case $i = 0$. We have

$$\mathrm{E}\left[X_0^t \mid X_0^{t-1}\right] = X_0^{t-1} - \mathrm{P}\, y \text{ had indegree } 0+1,$$

as at each time step a new node (i.e., $x$) of indegree 0 is added, and the only node that could change its indegree to 1 is $y$. The probability of the latter event is exactly $X_0^{t-1}/(t-1)$. By the linearity of expectation, we get

$$\mathrm{E}\left[X_0^t\right] = \left(1 - \frac{1}{t-1}\right) \mathrm{E}\left[X_0^{t-1}\right] + 1. \tag{2.2}$$

Next, consider $i \geq 1$. According to our model, nodes $z_1, \ldots, z_{k-1}$, will be chosen without replacement from $\Gamma(y)$, the successors of $y$. The successors of the new node $x$ will then be $\Gamma(x) = \{y, z_1, \ldots, z_{k-1}\}$. Since $z_1, \ldots, z_{k-1}$ are all distinct, the graph remains simple and $|\Gamma(x)| = k$.

For each $j = 1, \ldots, k-1$, the node $z_j$ is chosen with probability proportional to its indegree; this follows since node $z_j$ was the endpoint of an edge chosen u.a.r. The probability that a particular node of indegree $i \geq 1$ gets chosen as a successor is $\frac{1}{t-1} + \frac{i(k-1)}{k(t-1)}$ (recall that all the $k$ successors of $x$ will be distinct). Thus, for $i \geq 1$,

$$\mathrm{E}\left[X_i^t\right] = \left(1 - \frac{1}{t-1} - \frac{i}{t-1}\frac{k-1}{k}\right)\mathrm{E}\left[X_i^{t-1}\right] + \left(\frac{1}{t-1} + \frac{i-1}{t-1}\frac{k-1}{k}\right)\mathrm{E}\left[X_{i-1}^{t-1}\right]. \tag{2.3}$$

For the base cases, note that $X_t^t = 0$ for each $t \geq t_0$. Also, the variables $X_i^{t_0}$ are completely determined by $G_{t_0}$. For each fixed $k$, we have $f(t) = \Theta(t^{-2-\frac{1}{k-1}})$. Thus, there is a constant $c_0$ such that for any $c \geq c_0$, and for all $t \geq t_0$, $\mathrm{E}\left[X_t^t\right]$ follows (1). The base cases $\mathrm{E}\left[X_i^{t_0}\right]$, $i = 1, 2, \ldots$, can also be covered with a sufficiently large $c$ (that has to be greater than some function of the initial graph $G_{t_0}$).

For the inductive case, we have $f(0) = \frac{1}{2}$ (by applying $\Gamma(x)\Gamma(x+\frac{1}{2}) = \Gamma(2x)\,2^{1-2x}\sqrt{\pi}$, and $\Gamma(2x+1) = 2x\,\Gamma(2x)$, with $x = 1 + \frac{1}{k-1}$). Using this, (2.2), and simple calculations, we can show that if $X_0^{t-1}$ satisfies (1), then $X_0^t$ also satisfies (1). For $i \geq 1$, we have $f(i-1) = f(i) \cdot (ik - i + 2k + 2)/(ik - i + 1)$. An easy induction on (2.3) completes the proof. □

### 2.8.5 Proof of Lemma 10

*Proof.* Our model can be interpreted as the following stochastic process: at step $t$, two independent dice, with $t-1$ and $k$ faces respectively, are thrown. Let $Q_t$ and $R_t$ be the respective outcomes of these two trials. The new node $x$ will position itself to the immediate left of the node $y$ that was added at time $Q_t$. Suppose that the (ordered) list of successors of $y$ is $(z_1, \ldots, z_k)$. The ordered list of successors of $x$ will be composed of $y$ followed by the nodes $z_1, \ldots, z_k$ with the exception of node $z_{R_t}$. Thus, the number of nodes $X_i^\tau$ of indegree $i$ at time $\tau$ can be interpreted as a function of the trials $(Q_1, R_1), \ldots, (Q_\tau, R_\tau)$.

We want to show that changing the outcome of any single trial $(Q_{t'}, R_{t'})$, changes the r.v. $X_i^\tau$ (for fixed $i$) by an amount not greater than $2k$. Suppose we change $(q_{t'}, r_{t'})$ to $(q'_{t'}, r'_{t'})$, going from graph $G$ to $G'$. Let $x$ be the node added at time $t'$ with the choice $(q_{t'}, r_{t'})$, and $x'$ be the node added with the choice $(q'_{t'}, r'_{t'})$.

Let $S, S'$ be the successors of $x$ in $G$ and $x'$ in $G'$, respectively. The proof is complete by showing inductively that at any time step $t$, and for any nodes $y, y'$ added at the same time respectively in $G, G'$, the (ordered) list of successors of $y$ and $y'$ are *close*, i.e., in each of their positions, they either have the same successor, or they have two different elements of $S \cup S'$.

If $t \leq t'$, then the proof is immediate. For $t > t'$, it is easy to see that the only edges we need to consider are the copied edges. By induction, we know that at time $t-1$, the lists of successors of the node we are copying from, in $G$ and $G'$, were close. We project those lists in such a way that the same order is induced. Thus the lists of the time $t$ node are close and the proof is complete. □

### 2.8.6 Proof of Theorem 11

*Proof.* As in the proof of Theorem 9, we start by obtaining a recurrence on the r.v.'s $Z_i^t$. Let $x$ be the node added at time $t$, and let $y, y'$ be the nodes to the immediate right and left of $x$ respectively (where $y'$ equals the last node in the ordering if $x$ is placed before the first node $y$).

Consider $Z_1^t$. For $t > t_0$, $\mathrm{E}\left[Z_1^t \mid Z_1^{t-1}\right] = Z_1^{t-1} - \mathrm{P}\,x$ enlarges an edge of cd-length $1+1$, as an edge $x \to y$ of length 1 is necessarily added to the graph, and adding $x$ can enlarge at most one edge of cd-length 1 (that is, the edge $y' \to y$ if it exists). The probability of the latter event is equal to $Z_1^{t-1}/(t-1)$. By the linearity of expectation,

$$\mathrm{E}\left[Z_1^t\right] = \left(1 - \frac{1}{t-1}\right)\mathrm{E}\left[Z_1^{t-1}\right] + 1.$$

Now consider $Z_\ell^t$, for $\ell \geq 2$ and $t > t_0$. We have,

$$\begin{aligned}\mathrm{E}\left[Z_\ell^t \mid Z_\ell^{t-1}, Z_{\ell-1}^{t-1}\right] &= Z_\ell^{t-1} - \mathrm{E}\left[\#\text{ of edges of cd-length } \ell \text{ that } x \text{ enlarged } \mid Z_\ell^{t-1}, Z_{\ell-1}^{t-1}\right] \\ &+ \mathrm{E}\left[\#\text{ of edges of cd-length } (\ell-1) \text{ that } x \text{ enlarged } \mid Z_\ell^{t-1}, Z_{\ell-1}^{t-1}\right] \\ &+ \mathrm{E}\left[\#\text{ of edges of cd-length } (\ell-1) \text{ that } x \text{ copied from } y \mid Z_\ell^{t-1}, Z_{\ell-1}^{t-1}\right].\end{aligned}$$

Recall that $x$ is placed to the left of a node $y$ chosen u.a.r. Thus, given a fixed edge of length $\ell$, the probability this edge is enlarged by $x$ is $\ell/(t-1)$. Thus,

$$\mathrm{E}\left[\#\text{ of edges of length } \ell \text{ that } x \text{ enlarged } \mid Z_\ell^{t-1}, Z_{\ell-1}^{t-1}\right] = \frac{\ell}{t-1} Z_\ell^{t-1}, \text{ and}$$

$$\mathrm{E}\left[\#\text{ of edges of length } (\ell-1) \text{ that } x \text{ enlarged } \mid Z_\ell^{t-1}, Z_{\ell-1}^{t-1}\right] = \frac{\ell-1}{t-1} Z_{\ell-1}^{t-1},$$

$$\begin{aligned}&\mathrm{E}\left[\#\text{ of edges of cd-length } (\ell-1) \text{ that } x \text{ copied from } y \mid Z_\ell^{t-1}, Z_{\ell-1}^{t-1}\right] \\ &\qquad = \sum_{j=1}^{k-1} \mathrm{P}\,\text{the } j\text{-th copied edge had cd-length } (\ell-1) \mid Z_\ell^{t-1}, Z_{\ell-1}^{t-1}.\end{aligned}$$

Note that, for each $j = 1, \ldots, k-1$, the $j$-th copied edge is chosen uniformly at random over all the edges (even if the $k-1$ copied edges are not independent). Thus,

$$\sum_{j=1}^{k-1} \mathrm{P}\,\text{the } j\text{-th copied edge had cd-length } (\ell-1) \mid Z_\ell^{t-1}, Z_{\ell-1}^{t-1} = \frac{(k-1)Z_{\ell-1}^{t-1}}{k(t-1)}.$$

By the linearity of expectation, we get for $\ell \geq 2$,

$$\mathrm{E}\left[Z_\ell^t\right] = \left(1 - \frac{\ell}{t-1}\right)\mathrm{E}\left[Z_\ell^{t-1}\right] + \left(\frac{\ell-1}{t-1} + \frac{1}{t-1}\frac{k-1}{k}\right)\mathrm{E}\left[Z_{\ell-1}^{t-1}\right].$$

The base cases can be handled as in Theorem 9. The inductive step for $\ell = 1$ can be easily shown. For $\ell \geq 2$, it suffices to note that $g(\ell-1) = k \cdot (\ell+1)/(\ell k - 1) \cdot g(\ell)$. □

### 2.8.7 Proof of Theorem 17

*Proof.* Given a node $v$ in $G$, just by looking at two-neighborhood, we can either (a) find an out-neighbor $w$ of $v$ having exactly $k-1$ out-neighbors in common with $v$, or (b) we can conclude that $v$ was part of the "seed" graph $G_{t_0}$ (having constant order). This step takes time $O(k^2) = O(1)$.

Indeed, if $v$ were not part of $G_{t_0}$, during its insertion, $v$ added a proximity edge to its "real prototype" $w$, and copied $k-1$ of $w$'s outlinks. If more than one out-neighbor of $v$ has $k-1$ out-neighbors in common with $v$, we choose one arbitrarily and we call it the "possible prototype" of $v$.

For compressing, we create an unlabeled rooted forest out of the nodes in $G_{t_0}$. A node $v$ will look for a possible prototype $w$. If such a $w$ is found, then $v$ will choose $w$ as its parent. Otherwise $v$ will be a root in the forest.

To describe $G$, it will suffice to (a) describe the unlabeled rooted forest, (b) describe the subgraph induced by the roots of the trees in the forest, and (c) for each non-root node $v$ in the forest, use $\lceil \log k \rceil$ bits to describe which of its parent's out-neighbors was not copied by $v$ in $G$. The forest can be described with $O(n)$ bits, for instance, by writing down the *down / up* steps made when visiting each tree in the forest, disregarding edge orientations (as each edge is directed from the child to the parent). This requires $O(n)$ bits. The graph induced by the roots of the trees (i.e., a subgraph of $G_{t_0}$) can be stored in a non-compressed way using $O(t_0^2) = O(1)$ bits. The third part of the encoding will require at most $O(n \log k) = O(n)$ bits. Note that it is trivial to compute each of the three encodings in linear time. □

### 2.8.8 Other properties

#### Bipartite cliques

Recall that a *bipartite clique* $K(a, b)$ is a set $A$ of $a$ nodes and a set $B$ of $b$ nodes such that each node in $A$ has an edge to every node in $B$. We can show that the graphs generated by our model contain a large number of bipartite cliques. The proof is similar to the one of [70] for the linear growth model.

**Theorem 22.** *There exists a $\beta > 0$, such that the number of bipartite cliques $K(\Omega(\log n), k)$ in our model is $\Omega(n^\beta)$, w.h.p.*

*Proof (Sketch).* Take any fixed node $x$ of the seed graph $G_{t_0}$ and a subset $S$ of $k-1$ of its successors. Divide the time steps $t - t_0$ into disjoint epochs of exponentially increasing size, i.e., of sizes $c\tau, c^2\tau, c^3\tau, \ldots$, for a large enough $\tau$. Let $j$ be the number of epochs; easily, $j = \Omega(\log n)$. Note that for $i \leq j$, the probability that at least one node added in epoch $i$ will attach itself to $x$ and copy exactly the edges in $S$ is at least a constant; also, for each $i \neq i'$, these events are independent. Thus, w.h.p., at least $\Omega(\log n)$ nodes will be *good*, i.e., will have $S \cup \{v\}$ as successors.

Now, any subset of the good nodes will form a bipartite clique with $S \cup \{v\}$. The number of subsets of size $\Omega(\log n)$ is easily shown to grow as $\Omega(n^\beta)$ for some $\beta > 0$. □

#### Clustering coefficient

Watts and Strogatz [106] introduced the concept of clustering coefficient. The *clustering coefficient* $C(x)$ of a node $x$ is the ratio of the number of edges between neighbors of $x$ and the maximum possible number[9] of such edges. The clustering coefficient $C(G)$ of a (simple) graph $G$ is the average of the clustering coefficients of its nodes.

Snapshots of the real web graph have been observed to possess a pretty high clustering coefficient. Thus, having a high clustering coefficient (that is, having a constant clustering coefficient) is a desirable property of web graphs' models.

**Theorem 23.** *Take a (directed) graph $G$ generated by our model. The clustering coefficient of $G$ is $\Theta(1)$ w.h.p.*

*Proof.* By Theorem 9, and Lemma 10, there will exists $q = \Theta(n)$ many nodes of indegree 0 w.h.p. Take any node $x$ of indegree 0, and let $y$ be the node that $x$ was copied from. Then, $x$ and $y$ shares $k - 1$ out-neighbors (the "copied" ones). The total degree of $x$ is $k$, thus the clustering coefficient of $x$ is $\geq \frac{k-1}{k(k-1)} = \frac{1}{k} \in \Omega(1)$. The clustering coefficient of the graph is the average of the clustering coefficients of its nodes; thus, in our case, it is $\geq \frac{1}{n} \cdot q \cdot \frac{1}{k} \geq \Omega(1)$.

In general, the maximum value of the clustering coefficient is 1. The claim follows. □

The previous proof also shows that, if we remove orientations from the edges of our model's graphs, the clustering coefficient of the undirected graphs we obtain is $\Theta(1)$.

#### Diameter

We now argue that, w.h.p., the undirected diameter of our random graphs is $O(\log n)$ (provided that the seed graph $G_{t_0}$ was weakly-connected). By undirected diameter, we mean the diameter of the undirected graph obtained by removing edge orientations from our graphs. Note that our graphs are almost DAGs, i.e., they are DAGs perhaps except for the nodes in the seed graph $G_{t_0}$ and therefore directed diameter is not a meaningful notion to consider.

Consider the so-called *random recursive trees*: the process starts with a single node and at each step, a node is chosen uniformly at random, and a new leaf is added as a child of that node; the process ends at the generic time $n$. A result by Szymanski [102] shows that random recursive trees on $n$ nodes have height $O(\log n)$ w.h.p.

[9] That is, $\frac{1}{2}\deg(x)(\deg(x) - 1)$ in the undirected case and $\deg(x)(\deg(x) - 1)$ in the directed case.

Consider the "proximity" edges added in step (ii) in our model, i.e., those added from the new node, to a node chosen uniformly at random. Trivially, these edges induce a random recursive forest with $t_0$ different roots corresponding to the nodes of the seed graph $G_{t_0}$. A result of [102] states that the height of a random recursive tree on $n$ nodes is $O(\log n)$ w.h.p. Thus, assuming that $G_{t_0}$ is weakly-connected implies that the (undirected) diameter of our model's graphs is $O(\log n)$ w.h.p.

# Chapter 3

# Gossiping in Social Networks

Rumour spreading, also known as randomized broadcast or randomized gossip (all terms that will be used as synonyms throughout this chapter), refers to the following distributed algorithm. Starting with one source node with a message, the protocol proceeds in a sequence of synchronous rounds with the goal of *broadcasting* the message, i.e. to deliver it to every node in the network. At round $t \geq 0$, every node that knows the message selects a uniformly random neighbour to which the message is forwarded. This is the so-called `PUSH` strategy. The `PULL` variant is specular. At round $t \geq 0$ every node that does not yet have the message selects a neighbour uniformly at random and asks for the information, which is transferred provided that the queried neighbour knows it. Finally, the `PUSH-PULL` strategy is a combination of both. In round $t \geq 0$, each node selects a random neighbour to perform a `PUSH` if it has the information or a `PULL` in the opposite case. This three strategies have been introduced by [40]. One of the most studied questions for rumour spreading concerns its completion time: how many rounds will it take for one of the above strategies to disseminate the information to all nodes in the graph, assuming a worst-case source? We will say that rumour spreading is *fast* if its completion time is poly-logarithmic in the size of the network regardless of the source, and that it is *slow* otherwise.

Randomized broadcast with its many variants is an eminently practical protocol, with implications for the study of epidemic diffusion processes. It would be very interesting to characterize a set of necessary and/or sufficient conditions for it to be fast in a given network. In this context, we do provide a very general sufficient condition– high conductance. The practical implications are that a network could be known to have, or designed to have, high expansion. Our result provides a guarantee of efficiency in such cases. Our main motivation however comes from the study of social networks. Loosely stated, we are looking after a theorem of the form *"Rumour spreading is fast in social networks"*. Our result is a good step in this direction because there are reasons to believe that social networks do have high conductance. This is certainly the case for preferential attachment models such as that of [81]. More importantly, there is some empirical evidence that this might be the case for real social networks (for which computing the real conductance is a formidable computational challenge); in particular the authors of [76] observe how in many different social networks there exist only cuts of small (logarithmic) size having small (inversely logarithmic) conductance – all other cuts appear to have larger conductance. That is, the conductance of the social networks they analyze is larger than a quantity seemingly proportional to an inverse logarithm. Knowing that rumour spreading is indeed fast for social networks would have several implications. Indeed, rumour spreading is a simplified form of viral mechanism. By understanding it in detail we might be able to say something about more complex and realistic epidemic processes, with implications that might go beyond the understanding of information dissemination in communication networks.

Perhaps surprisingly, in the case of edge expansion, see [30] for more details, there are classes of graphs for which the protocol is slow, while the problem remains open for vertex expansion. In contrast to the case of edge expansion, in this chapter we show that high conductance by itself is sufficient to ensure that rumour spreading is fast. More precisely, we show the following:

**Theorem 24.** *Given any network $G$ and any source node,* `PUSH-PULL` *broadcasts the message within $O(\log^4 n/\phi^6(G))$ many rounds, where $n$ is the number of nodes of the input graph $G$ and $\phi(G)$ is its conductance.*

Thus, if the conductance is high enough, say $\phi^{-1} = O(\log n)$ (as it has been observed to be in real social

networks [76]), then, according to our terminology, rumour spreading is fast.

We notice that the use of PUSH-PULL is necessary, as there exist high conductance graphs for which neither the PUSH, nor the PULL, strategies are *fast* on their own. Examples can be found in [30] where it is shown that in the classical preferential attachment model PUSH and PULL by themselves are slow. Although it is not known if preferential attachment graphs have high conductance, the construction of [30] also applies to the "almost" preferential attachment model of [81], which is known to have high conductance.

In terms of message complexity, we observe first that it has been determined precisely only for very special classes of graphs (cliques [63] and Erdös-Rényi random graphs [47]). Apart from this, given the generality of our class, it seems hard to improve the trivial upper bound on the number of messages– running time times number of nodes. For instance consider the "lollipop graph". Fix $\omega(n^{-1}) < \phi < o(\log^{-1} n)$, and suppose to have a path of length $\phi^{-1}$ connected to a clique of size $n - \phi^{-1} = \Theta(n)$. This graph has conductance $\approx \phi$. Let the source be any node in the clique. After $\Theta(\log n)$ rounds each node in the clique will have the information. Further it will take at least $\phi^{-1}$ steps for the information to be sent to the each node in the path. So, at least $n - \phi^{-1} = \Theta(n)$ messages are pushed (by the nodes in the clique) in each round, for at least $\phi^{-1} - \Theta(\log n) = \Theta(\phi^{-1})$ rounds. Thus, the total number of messages sent will be $\Omega(n \cdot \phi^{-1})$. Observing that the running time is $\Theta(\phi^{-1} + \log n) = \Theta(\phi^{-1})$, we have that the running time times $n$ is (asymptotically) less than or equal to the number of transmitted messages.

We also note that one cannot give fault-tolerant guarantees (that is, the ability of the protocol to resist to node and/or edge deletions) based only on conductance. A star has high conductance, but failure of the central node destroys connectivity.

As remarked, our result is based upon a connection with the spectral sparsification procedure of [99]. Roughly, the connection is as follows. The spectral sparsification procedure (henceforth ST) is a sampling procedure that, given a graph $G$, selects each edge $uv$ independently with probability

$$p_{uv} := \min\left\{1, \frac{\delta}{\min\{\deg(u), \deg(v)\}}\right\} \tag{3.1}$$

where $\deg(u)$ denotes the degree of a node $u$ and

$$\delta = \Theta\left(\frac{\log^2 n}{\phi^4}\right). \tag{3.2}$$

Spielman and Teng show that the eigenvalue spectrum of the sampled graph $ST(G)$ is, with high probability, a good approximation to that of $G$. In turn, this implies that $\phi(ST(G)) \geq \Omega(\phi^2(G))$ and that $ST(G)$ is connected (otherwise the conductance would be zero). The first thing we notice is that ST expands: after having applied ST, for each subset of vertices $S$ of at most half the total volume of $G$, the total volume of the set of vertices reachable from $S$ via edges sampled by ST is at least a constant fraction of the volume of $S$ (the volume of a set of vertices is the sum of their degrees). Intuitively, if we were to use ST to send messages across the edges it samples, we would quickly flood the entire graph. Allowing for some lack of precision for sake of clarity, the second main component of our approach is that rumour spreading stochastically dominates ST, even if we run it for poly-logarithmically many rounds. That is to say, the probability that an edge is used by rumour spreading to pass the message is greater than that of being selected by ST.

In a broad sense our work draws a connection between the theory of spectral sparsification and the speed with which diffusion processes make progress in a network. This could potentially have deeper ramifications beyond the present work and seems to be worth exploring. For instance, recently in [7, 98] introduced a more efficient sparsification technique that is able to approximate the spectrum using only $O(n \log n)$, and $O(n)$, edges, respectively. Extending our approach to the new sampler appears challenging, but not without hope. The consequence would a sharper bound on the diffusion speed. Of great interest would also be extending the approach to other diffusion processes, such as averaging. Finally, we remark that an outstanding open problem in this area is whether vertex expansion implies that rumour spreading is fast.

From the technical point of view, once the connection between spectral sparsification and rumour spreading is seen, several hurdles still remain along the way. The main problem is that ST selects edges independently, while rumour spreading exhibits complicated dependencies that make it hard to establish stochastic domination. We will see in later sections how to deal with this set of problems.

## 3.1 Related work

The literature on the gossip protocol and social networks is huge and we confine ourselves to what appears to be more relevant to the present work.

Clearly, at least diameter-many rounds are needed for the gossip protocol to reach all nodes. It has been shown that $O(n \log n)$ rounds are always sufficient for each connected graph of $n$ nodes [53]. The problem has been studied on a number of graph classes, such as hypercubes, bounded-degree graphs, cliques and Erdös-Rényi random graphs (see [53, 55, 89]). Recently, there has been a lot of work on "quasi-regular" expanders (i.e., expander graphs for which the ratio between the maximum and minimum degree is constant) — it has been shown in different settings [8, 42, 43, 54, 92] that $O(\log n)$ rounds are sufficient for the rumour to be spread throughout the graph. See also [64, 83]. Our work can be seen as an extension of these studies to graphs of arbitrary degree distribution. Observe that many real world graphs (e.g., facebook, Internet, etc.) have a very skewed degree distribution — that is, the ratio between the maximum and the minimum degree is very high. In most social networks' graph models the ratio between the maximum and the minimum degree can be shown to be polynomial in the graph order.

Mihail et al. [81] study the edge expansion and the conductance of graphs that are very similar to preferential attachment (PA) graphs. We shall refer to these as "almost" PA-graphs. They show that edge expansion and conductance are constant in these graphs.

Concerning PA graphs, the work of [30] shows that rumour spreading is fast in those networks. Although PA networks have high conductance, the present work does not supersede those results, for there it is shown a $O(\log^2 n)$ time bound.

In [17] it is shown that high conductance implies that *non-uniform* (over neighbours) rumour spreading succeeds. By non-uniform we mean that, for every ordered pair of neighbours $i$ and $j$, node $i$ will select $j$ with probability $p_{ij}$ for the rumour spreading step (in general, $p_{ij} \neq p_{ji}$). This results does not extend to the case of uniform probabilities studied in this chapter. In our setting (but not in theirs), the existence of a non uniform distribution that makes rumour spreading fast is a rather trivial matter. A graph of conductance $\phi$ has diameter bounded by $O(\phi^{-1} \log n)$. Thus, in a synchronous network, it is possible to elect a leader in $O(\phi^{-1} \log n)$ many rounds and set up a BFS tree originating from it. By assigning probability 1 to the edge between a node and its parent one has the desired non uniform probability distribution. Thus, from our point of view the existence of non uniform problem is rather uninteresting.

## 3.2 Preliminaries

We introduce notation, definitions, and recall several facts for later use.

Given a graph $G = (V, E)$, we denote by $ST(G)$ the graph on the same vertex set of $G$ whose edges have been selected by the ST-sparsification algorithm, i.e. with probability defined by Equation 3.1. We use $ST(E)$ to denote the edges of $ST(G)$.

In the spectral sparsification setting of [99] the weight of edge $uv$, surviving the sparsification procedure, is $w_{uv} := p_{uv}^{-1}$.

**Notation 25 (Weights).** *The* weight *of a set of edges $E' \subseteq E$ is defined as $w_G(E') := \sum_{e \in E'} w_e$. The* weight *of a vertex $u$ in a graph $G$ is defined as $w_G(v) := \sum_{e \ni v} w_e$. The weight of a set of vertices $S$ is defined as $w_G(S) := \sum_{u \in S} w_G(u)$.*

Given a graph $G$, the degree of a node $u$ is denoted as $\deg_G(u)$.

**Definition 26 (Volume).** *The* volume *of a set of vertices $S$ of a graph $G$ is defined to be*

$$\mathrm{Vol}_G(S) = \sum_{v \in S} \deg_G(u).$$

**Definition 27 (Volume expansion).** *Let $f$ be a randomized process selecting edges in a graph $G = (V, E)$. Given $S \subseteq V$, the set $f(S)$ is the union of $S$ and the set of all vertices $u \in V - S$ such that there exists some $v \in S$ and $uv \in E$ was selected by $f$. We say that $f$* $\alpha$-expands for *$S$ if*

$$\mathrm{Vol}_G(f(S)) \geq (1 + \alpha) \cdot \mathrm{Vol}_G(S).$$

The set of edges across the cut $(S, V - S)$ will be denoted as $\partial_G(S)$

**Definition 28 (Conductance).** *A set of vertices $S$ in a graph $G$ has* conductance $\phi$ *if*

$$w_G(\partial_G(S)) \geq \phi \cdot w_G(S).$$

*The conductance of $G$ is the minimum conductance, taken over all sets $S$ such that $w_G(S) \leq w_G(V)/2$.*

We will make use of a deep result from [99]. Specifically, it implies that the spectrum of $ST(G)$ is approximately the same as the one of $G$. It follows from [25, 33] that:

**Theorem 29 (Spectral Sparsification).** *There exists a constant $c > 0$ such that, with probability at least $1 - O(n^{-6})$, for all $S \subseteq V$ such that $w_G(S) \leq w_G(V)/2$, we have*

$$w_{ST(G)}(\partial_{ST(G)}(S)) \geq c \cdot \phi^2(G) \cdot w_{ST(G)}(S). \tag{3.3}$$

We say that an event occurs *with high probability (whp)* if it happens with probability $1 - o(1)$, where the $o(1)$ term goes to zero as $n$, the number of vertices, goes to infinity.

## 3.3 The proof

In this section we will prove Theorem 24. Before plunging into technical details, let us give an overview of the proof. The first thing we do is to show that $ST$-sparsification enjoys volume expansion. That is, there exists a constant $c > 0$ such that, for all sets $S$ of volume at most $\mathrm{Vol}_G(V)/2$,

$$\mathrm{Vol}_G(ST(S)) > (1 + c \cdot \phi^2(G)) \,\mathrm{Vol}_G(S). \tag{3.4}$$

The second, more delicate, step in the proof is to show that rumour spreading (essentially) stochastically dominates $ST$-sparsification. Assume that $S$ is the set of vertices having the message. If we run `PUSH-PULL` (henceforth $PP$, which plays the role of $f$ in Definition 27) for $T = O(\log^3 n/\phi^4)$ rounds, then $\mathrm{Vol}_G(PP(S)) \succeq \mathrm{Vol}_G(ST(S))$, where $\succeq$ denotes stochastic domination. (Strictly speaking, this is not quite true, for there are certain events that happen with probability 1 in $ST$, and only with probability $1 - o(1)$ with $PP$.)

Consider then the sequence of sets $S_{i+1} := PP(S_i)$, and $S_0 := \{u\}$ where $u$ is any vertex. These sets keep track of the diffusion via `PUSH-PULL` of the message originating from $u$ (the process could actually be faster, in the sense that $S_i$ is a subset of the informed nodes after $T \cdot i$ rounds). Then, for all $i$,

$$\mathrm{Vol}_G(S_{i+1}) = \mathrm{Vol}_G(PP(S)) \geq \mathrm{Vol}_G(ST(S)) > (1 + c \cdot \phi^2(G)) \,\mathrm{Vol}_G(S_i).$$

The first inequality follows by stochastic domination, while the second follows from Equation 3.4. Since the maximum volume is $O(n^2)$, we have that $\mathrm{Vol}(S_t) > \mathrm{Vol}(G)/2$ for $t = O(\log n/\phi^2)$. This means that within $O(T \log n/\phi^2)$ many rounds we can deliver the message to a set of nodes having more than half of the network's volume. To conclude the argument we use the fact that $PP$ is specular. If we interchange `PUSH` with `PULL` and viceversa, the same argument "backwards" shows that once we have $S_t$ we can reach any other vertex within $O(T \log n/\phi^2)$ additional many rounds. After this informal overview, we now proceed to the formal argument. In what follows there is an underlying graph $G = (V, E)$. where $n := |V(G)|$, for which we run $ST$ and $PP$.

### 3.3.1 Volume expansion of $ST$-sparsification

Our goal here is to show Equation 3.4. We begin by showing that the weight of every vertex in $ST(G)$ is concentrated around its expected value, namely its degree in $G$.

**Lemma 30.** *With probability at least $1 - n^{-\omega(1)}$ over the space induced by the random ST-sparsification algorithm, for each node $v \in V(G)$ we have that*

$$w_{ST(G)}(v) = (1 \pm o(1)) \deg_G(v).$$

*Proof.* If $\deg_G(v) \leq \delta$, then $w_{ST(G)}(v)$ is a constant random variable with value $\deg_G(v)$. If we assume the opposite we have that $E[w_{ST(G)}(v)] = \deg_G(v)$, by definition of $ST$-sparsification. Recalling the definition of $\delta$ (Equation 3.2), let $X = w_{ST(G)}(v)\delta/\deg(v)$. Then, $E[X] = \delta$. By the Chernoff bound,

$$\Pr[|X - E[X]| \geq \varepsilon E[X]] \leq 2\exp\left(-\frac{\varepsilon^2}{3}E[X]\right) = 2\exp\left(-\frac{\varepsilon^2}{3}\delta\right).$$

Since $\delta = \Theta\left(\frac{\log^2 n}{\phi^4}\right)$, if we pick $\varepsilon = \omega(\phi^2/\sqrt{\log n})$, the claim follows. □

**Corollary 31.** *Let $S \subseteq V$ be such that $\mathrm{Vol}_G(S) \leq \mathrm{Vol}_G(V)/2$. With probability at least $1 - n^{-\omega(1)}$ over the space induced by the random ST-sparsification algorithm, we have that*

$$w_{ST(G)}(S) = (1 \pm o(1))\,\mathrm{Vol}_G(S).$$

Theorem 29 states that ST-sparsification enjoys weight expansion. By means of Lemma 30 and Corollary 31 we can translate this property into volume expansion. Recall that $ST(S)$ is $S$ union the vertices reachable from $S$ via edges sampled by $ST$.

**Lemma 32 (Volume expansion of ST).** *There exists a constant $c$ such that for each fixed $S \subseteq V$ having volume at most $\mathrm{Vol}_G(V)/2$, with high probability*

$$\mathrm{Vol}_G(\mathrm{ST}(S)) > (1 + c \cdot \phi^2(G))\,\mathrm{Vol}_G(S).$$

*Proof.* By Theorem 29, $w_{ST(G)}(\partial_{ST(G)}(S)) \geq c \cdot \phi^2(G) \cdot w_{ST(G)}(S)$. Clearly,

$$w_{ST(G)}(ST(S)) \geq w_{ST(G)}(\partial_{ST(G)}(S)).$$

By Corollary 31 we have that $\mathrm{Vol}_G(ST(S)) = w_{ST(G)}(ST(S))(1 \pm o(1))$ and $\mathrm{Vol}_G(S) = w_{ST(G)}(S)(1 \pm o(1))$. The constant $c$ in Theorem 29 and the error terms in Corollary 31 can be chosen in such a way that $\mathrm{Vol}_G(ST(S)) > (1 + c' \cdot \phi^2(G))\,\mathrm{Vol}_G(S)$ for some $c' > 0$. The claim follows. □

We end this section by recording a simple monotonicity property stating that if a process enjoys volume expansion, then by adding edges expansion continues to hold.

**Lemma 33.** *Let $f$ and $g$ be a randomized processes that select each edge $e$ in $G$ independently with probability $p_e$ and $p'_e$, respectively, with $p'_e \geq p_e$. Then, for all $t > 0$ and $S$,*

$$\Pr(\mathrm{Vol}_G(g(S)) > t) \geq \Pr(\mathrm{Vol}_G(f(S)) > t).$$

*Proof.* The claim follows from a straightforward coupling, and by the fact that if $A \subseteq B$ then $\mathrm{Vol}(A) \leq \mathrm{Vol}(B)$. □

## 3.4 The road from ST-sparsification to Rumour Spreading

The goal of this section is to show that *PP*stochastically dominates *ST*. As stated the claim is not quite true and the kind of stochastic domination we will show is slightly different. Let us begin by mentioning what kind of complications can arise in proving a statement like this.

A serious issue is represented by the massive dependencies that are exhibited by *PP*. To tackle this we introduce a series of intermediate steps, by defining a series of processes that bring us from *ST* to *PP*. We will relax somewhat *PP* and *ST* by introducing two processes *PPW* and *DST* to be defined precisely later. In brief, *PPW* is the same as *PP* except that vertices select neighbours without replacement. *DST* differs from *ST* by the fact that edges are "activated" (we will come to this later) by both endpoints. Again, slightly simplifying a more complex picture for the sake of clarity, the main flow of the proof is to show that $ST \preceq DST \preceq PPW \preceq PP$, where $\preceq$ denotes stochastic domination. Let us now develop formally this line of reasoning.

The first intermediate process is called *double ST-sparsification* henceforth (DST) and it is defined as follows. *DST* is a process in which vertices select edges incident on them (similarly to what happens with *PP*). With DST each edge $e \ni u$ is *activated* independently by $u$ with probability

$$p_e := \min\left\{1, \frac{\delta}{\deg_G(u)}\right\}. \tag{3.5}$$

An edge $e = uv$ is *selected* if it is activated by at least one of its endpoints. Clearly $DST \succeq ST$ and thus it follows immediately from Lemma 33 that DST expands. We record this fact for later use.

**Lemma 34 (Volume expansion of DST).** *There exists a constant $c$ such that for each fixed $S \subseteq V$ having volume at most $\mathrm{Vol}_G(V)/2$, with high probability*

$$\mathrm{Vol}_G(\mathrm{DST}(S)) > (1 + c \cdot \phi^2(G))\,\mathrm{Vol}_G(S).$$

Therefore from now on we can forget about ST and work only with DST. The next lemma shows that with high probability after *DST*-sparsification the degree of all vertices is $O(\log^2 n/\phi^2)$.

**Lemma 35.** *Let $\xi$ be the event* "with *DST* no node will activate more than $2\delta$ edges"*. Then,*

$$\Pr(\xi) = 1 - n^{-\omega(1)}.$$

*Proof.* The only case to consider is $\deg(v) > 2\delta$. Let $X = (\#$ of edges activated by $v)$. Then $E[X] = \sum_{u \in N(v)} \frac{\delta}{\deg(v)} = \delta$. Invoking the Chernoff bound we get (see for instance [44]),

$$\Pr[X \geq 2E[X]] \leq 2\exp\left(-\Omega(\delta)\right) \leq n^{-\omega(1)}$$

for $n$ large enough. □

**Remark:** For the remainder of the section, when dealing with *DST* we will work in the subspace defined by conditioning on $\xi$. We will do so without explicitly conditioning on $\xi$, for sake of notational simplicity.

The second step to bring *PP* "closer" to *ST* is to replace *PP* with a slightly different process. This process is dubbed PP *without replacement* (henceforth PPW) and it is defined as follows. If *PPW* runs for $t$ rounds, then each vertex $u$ will select $\min\{\deg_G(u), t\}$ edges incident on itself without replacement (while *PP* does it with replacement). The reason to introduce *PPW* is that it is much easier to handle than *PP*.

**Notation 36 (Vectors of sets and time horizons).** *Given a vertex set $S \subseteq V$ we will use the notation $A := (A_u : u \in S)$ to denote a collection of vertex sets, where each $A_u$ is a subset of the neighbours of $u$. A vector of integers $T = (t_u : u \in S)$ is called a* time horizon *for $S$. Furthermore we will use the notation $\|A\| := (|A_u| : u \in S)$, to denote the time horizon for $S$ that corresponds to $A$.*

**Notation 37 (Behaviour of PPW).** *Let $S$ be a set of vertices in a graph $G$ and let $T$ be a time horizon for $S$. $\mathrm{PPW}(T, S)$ is the process where every vertex $u \in S$ activates a uniformly random subset of $\min\{\deg_G(u), t_u\}$ edges incident on itself, to perform a* `PUSH-PULL` *operation for each of them.*

Notice that *PPW* might specify different cardinalities for different vertices. This is important for the proofs to follow.

With this notation we can express the outcome of *DST* sampling. Focus on a specific set of vertices $S = (u_1, \ldots, u_k)$. We know that *DST*$(S)$ expands with respect to $S$ and we want to argue that so does *PPW*. The crux of the matter are the following two simple lemmas.

**Lemma 38.** *Let $u$ be a vertex in $G$ and $t$ a positive integer. And let $\mathrm{DST}(u)$ and $\mathrm{PPW}(t, u)$ denote, respectively, the subset of edges incident on $u$ selected by the two processes. Then,*

$$\Pr(\mathrm{DST}(u) = A_u \cup \{u\} \;\mid\; |A_u| = t) = \Pr(\mathrm{PPW}(t, u) = A_u \cup \{u\}).$$

*Proof.* With *DST* each vertex activates (and therefore selects) edges incident on itself with the same probability. If we condition on the cardinality, all subsets are equally likely. Therefore, under this conditioning, *DST*$(u)$ simply selects a subset of $t$ edges uniformly at random. But this is precisely what *PPW*$(t, u)$ does. □

**Lemma 39.** *Let $S = \{v_1, \ldots, v_{|S|}\}$ be a subset of vertices of $G$ and $T = (t_1, \ldots, t_{|S|})$ a time horizon for $S$. Then,*

$$\Pr\left(\bigwedge_{i=1}^{|S|} \mathrm{DST}(v_i) = A_{v_i} \cup \{v_i\} \;\middle|\; \|A\| = T\right) = \Pr\left(\bigwedge_{i=1}^{|S|} \mathrm{PPW}(t_i, v_i) = A_{v_i} \cup \{v_i\}\right).$$

*Proof.* This follows from Lemma 38 and the fact that under both *DST* and *PPW* vertices activate edges independently. □

In other words, for every realization $A$ of *DST* there is a time horizon $T_A$ such that the random choices of *PPW* are distributed exactly like those of *DST*. Said differently, if we condition on the cardinalities of the choices made by *DST*, then, for those same cardinalities, *PPW* is distributed exactly like *DST*. To interpret the next lemma refer to Definition 27.

**Lemma 40.** *Let $T := (2\delta, \ldots, 2\delta)$. There exists $c > 0$ such that, for all sets $S \subseteq V$,*

$$\Pr(\mathrm{PPW}(S, T) \text{ } (c\phi^2)\text{-expands for } S) \geq \Pr(\mathrm{DST}(S) \text{ } (c\phi^2)\text{-expands for } S) = 1 - o(1).$$

*Proof.* For the first inequality, recall that we are assuming that *DST* operates under conditioning on $\xi$. Thus, by Lemma 35 we have that each $u \in V$ activates at most $2\delta$ edges. Therefore $T$ majorizes every time horizon $T_S$ for which Lemma 39 holds. The last equality is derived from Lemma 35. □

We conclude the series of steps by showing that, given any set $S$, *PP* also expands with high probability.

**Lemma 41.** *Consider the* PP *process. Take any node $v$, and an arbitrary time $t_0$. Between time $t_0$ and $t_1 = t_0 + 9\delta \cdot \log n$, node $v$ activates at least* $\min(2\delta, \deg(v))$ *different edges with high probability.*

*Proof.* We split the proof into two cases, $\deg(v) \leq 3\delta$ and $\deg(v) > 3\delta$. In the former case, a straightforward coupon collector argument applies.[1].

Otherwise $\deg(v) > 3\delta$, *PP* will either choose $> 2\delta$ different edges during the $9\delta$ rounds, or it will choose at most $\leq 2\delta$ different edges. What is the probability of the latter event? In each round the probability of choosing a new edge will be $\geq \frac{\deg(v)-2\delta}{\deg(v)} \geq 1 - \frac{2\delta}{\deg(v)} \geq 1 - \frac{2}{3} = \frac{1}{3}$. But then by Chernoff bound, the probability of this event is at most $n^{-\omega(1)}$. □

To summarize, if *PP* is run $t_1 - t_0 = O(\delta \log n)$ steps, with high probability every node in *ST*$(S)$ selects at least $\min\{\deg_G(u), 2\delta\}$ many edges, and therefore dominates *PPW*.

## 3.5 The speed of *PP*

In this section, we upper bound the number of steps required by *PP* to broadcast a message in the worst case. The basic idea is that, as we have seen in the previous section, a *PP* requires $(\log^3 n/\phi^4)$ rounds to expand out of a set. Suppose the information starts at vertex $v$. Since each expansion increases the total informed volume by a factor of $(1 + \Omega(\phi^2))$ we have that after $(\log^4 n/\phi^6)$ rounds, the information will have reached a set of nodes of volume greater than half the volume of the whole graph. Consider now another node $w$. By the symmetry of the `PUSH-PULL` process, $w$ will be "told" the information by a set

[1] The probability of non-activation in $9\delta \log n$ rounds of some edge will be equal to $(1 - \frac{1}{\deg(v)})^{9\delta \log n} \leq (1 - \frac{1}{3\delta})^{9\delta \log n} \leq n^{-3}$. Thus, by union bounding over all its edges, the probability that event fails to happen is $O(n^{-2})$.

of nodes of volume bigger than half of the volume of the graph in $O(\log^4 n/\phi^6)$ many rounds. Thus the information will travel from $v$ to $w$ in $O(\log^4 n/\phi^6)$ many rounds, with high probability.

To develop the argument more formally, let us define a macro-step of PP as $2\delta \log n$ consecutive rounds. We start from a single node $v$ having the information, $S_0 = \{v\}$. As we saw, in each macro-step, with probability $\geq 1 - O(n^{-6})$ the volume of the set of nodes that happen to have the information increases by a factor $1 + \Omega(\phi^2)$, as long as the volume of $S_i$ is $\leq \frac{1}{2}\operatorname{Vol}_G(V)$.

Take any node $w \neq v$. If the information started at $w$, in $O(\log_{1+\Omega(\phi^2)} n) = O\left(\frac{1}{\phi^2}\log n\right)$ macro-steps the information will have reached a set of nodes $S = S_{O(\frac{1}{\phi^2}\log n)}$ of total degree strictly larger than $\frac{1}{2}\operatorname{Vol}_G(V)$ with probability $1 - O(n^{-6} \cdot \frac{\log n}{\phi^2}) \geq 1 - O(n^{-2}\log n)$. Note that the probability that the information, starting from some node in $S$, gets sent to $w$ in $O(\frac{1}{\phi^2}\log n)$ steps is greater than or equal to the probability that $w$ spreads the information to the whole of $S$ (we use PUSH-PULL, so each edge activation both sends and receive the information — thus by activating the edges that got the info from $w$ to $S$ in the reverse order, we could get the information from each node in $S$ to $w$. Note that the probability of the two activation sequences are exactly the same).

Now take the originator node $v$, and let it send the information for $O\left(\frac{1}{\phi^2}\log n\right)$ macro-rounds (for a total of $O\left(\frac{\log^4 n}{\phi^6}\right)$ many rounds). With high probability, the information will reach a set of nodes $S_v$ of volume strictly larger than $\frac{1}{2}\operatorname{Vol}_G(V)$. Take any other node $w$, and grow its $S_w$ for $O(\frac{1}{\phi^2}\log n)$ rounds with the aim of letting it grab the information. Again, after those many rounds, $w$ will have grabbed the information from a set of volume at least $\frac{1}{2}\operatorname{Vol}_G(V)$ with probability $1 - O(n^{-2}\log n)$. As the total volume is $\operatorname{Vol}_G(V)$ the two sets will intersect — so that $w$ will obtain the information with probability $1 - O(n^{-2}\log n)$. Union bounding over the $n$ nodes, gives us the main result: with probability $\geq 1 - O(n^{-1}\log n) = 1 - o(1)$, the information gets spread to the whole graph in $O\left(\frac{\log^4 n}{\phi^6}\right)$ rounds.

# Chapter 4

# User Interests and Caching

A caching subsystem is an integral component of any large-scale system, especially one where computing the answer anew — whether it involves reading a page off disk, as in the case of operating systems [94], or concerns buffer management issues in database systems [32, 46, 91, 100], or means recomputing the top results, as in the case of search engines [73] — is expensive. The basic paging problem is to devise a strategy to maintain a set of $k$ pages in fast memory. At each point in time, a request arrives for a page $p$. If $p$ is in the cache (*cache hit*), then it is served immediately. Otherwise (*cache miss*), the system fetches $p$ from slower storage and serves it (and typically stores it in the cache.) The seminal paper of Sleator and Tarjan [96] showed that no deterministic paging algorithm is $o(k)$-competitive and the simple LRU (Least Recently Used) strategy achieves that bound. Although a classical topic, the caching problem continues to receive attention both from practitioners [73], as well as theoreticians. On the theoretical side, there are many variations on this simple framework (cf. [16]), with several long-standing conjectures.

In [29] we consider the case when the pages lie in a metric space, and a cache hit occurs if there is a "similar" page in the cache. Our primary motivation comes from caching/buffer management questions in approximate nearest-neighbor applications such as multimedia systems [52] and contextual advertising on the web [87]; we describe the latter in more detail. Today, many web pages, in particular blogs, are supported by advertisement (ad) revenue. When a user navigates to a blog, the on-line service provider (OSP), e.g., Yahoo! or Google, selects the most relevant ad to display to the user based on the user characteristics and the page content. For example, a user from a New York IP address navigating to a page about Bangkok may be shown ads by travel agents advertising cheap flights. Since the OSPs are typically paid only in the event of a click, it is in their best interest to select and show the most relevant ad.

Consider then the problem faced by an OSP — upon arrival of user $u$ on page $p$, it must select the most relevant ad to show to the user.[1] The total number of potential candidate ads is in the hundreds of millions, and the computation to select the most relevant ad is typically expensive. Thus, it is prudent to cache the results from previous visits by $u$ to $p$. In the classical formulation, a cached result would only be valuable if a user with identical characteristics to $u$ landed on a page identical to $p$. However, if $u$ and $v$ are similar (say, $v$ is from New Jersey whereas $u$ was from New York), then when $v$ navigates to the same page $p$ about Bangkok, the same travel agency ads are likely to be among the most relevant, and the OSP should use the cached results to its advantage. Obviously, as the similarity between $u$ and $v$ decreases, the results becomes less relevant, and the OSP needs to be cognizant of the threshold at which the cached result would no longer be useful.

We call the problem of caching in this framework *similarity caching*. There can be two variants of similarity caching: a hard constraints variant, and a soft constraints one.

Formally there exists a metric space $(X, d)$ over the queries. In the latter case a threshold radius $r$ is given . When a query $p$ arrives, if there is a stored query $q$ in the cache with $d(p, q) \leq r$, then the cost to the algorithm is 0, as the request can be satisfied using the cache. Otherwise, we have to *fault* — that is, compute a solution for $p$, at a cost of 1 (and possibly add it to the cache, evicting some other solution if necessary). In this variant of the problem one aims to minimize the number of faults.

In the soft constraints problem, one aims to minimize the number of faults plus the sum of the

[1] In practice, there are many other constraints to be be considered, e.g., advertiser budgets. We ignore these here.

distances $d(p, q)$, over all queries $q$ in the stream with their answer $p$. In this variant, one can answer a query $q$ with an arbitrarily bad point $p$ — but the penalty incurred by the algorithm increases with $d(p, q)$.

Our findings in [29] indicate that the hard constraint version, while being easier to state, is computationally very hard (under the so called "competitive analysis" measure). On the other hand, we give a very good algorithms exist for the soft version of the problem.

The interested reader is referred to [29] for the actual results and their proofs.

In [87], we conduct an experimental analysis of a heuristic based on the algorithm given in [29] for the soft constraints version of the problem. The heuristic, and the algorithm, crucially leverage on a "median" sub-routing — that is, given a (multi-)set $S$ of points of the metric what is their median point (the point of the metric that minimizes the total distance to the set)?

In general (that is, for general metrics) finding the median point is a hard problem, but a trivial 2-approximation exists (one of the points of $S$ happens to be a 2-approximate median). From an inapproximability point of view, there exist metrics where the median problem cannot be approximated to better than $2 - \varepsilon$ in polynomial time if P $\neq$ NP.

In practice, and in particular in the system used at Yahoo!, the queries — and the ads — are part of a very specific metric space, the weighted, or generalized, Jaccard space.

In a forthcoming paper [28], we show that the median problem in the Jaccard space admits a PTAS, and does not admit a FPTAS if P $\neq$ NP. While our PTAS requires $(nm)^{\varepsilon^{-\Theta(1)}}$ time, it happens to be very fast (linear in the input) when the median point has small total distance to the input points. In fact, for the intended application, either the median point (which is, the answer to a query) *has* this property (the answer is close to the user queries) or it is unclear if actually finding it has any usefulness. Our experimental work in [87] indicates that it is often the case that this property holds, thanks to the properties of the users (in a sense, they are well-clustered), and of the queries they generate.

The PTAS, therefore, is not only of theoretical interest but happens to be a "practical" algorithm when the input points are well-clustered — something that happens in our user-generated data.

In the rest of this chapter, we will present the PTAS, and the hardness results, of [28].

## 4.1 Jaccard Coefficient

The Jaccard coefficient is a widely used set similarity measure, introduced more than a century ago [60]. For two sets $X, Y$, it is defined to be $J(X, Y) = |X \cap Y|/|X \cup Y|$. The Jaccard distance between the sets, defined as $D(X, Y) = 1 - J(X, Y)$, is known to be a metric. A natural generalization of Jaccard similarity, independently proposed several times over many years [37,57,58,61,65,80,93,97], is to consider $n$-dimensional non-negative vectors $X, Y$ and define $J(X, Y) = \frac{\sum_{i=1}^{n} \min(X_i, Y_i)}{\sum_{i=1}^{n} \max(X_i, Y_i)}$; the generalized Jaccard distance, $D(X, Y) = 1 - J(X, Y)$, still remains a metric. Here we study the computational complexity of the median problem in the Jaccard distance metric, namely, given a family $\mathcal{S}$ of input sets (or vectors), find a set (vector) $M^*$ that minimizes $\sum_{X \in \mathcal{S}} D(M^*, X)$.

The use of the Jaccard metric and Jaccard median is common in many scientific fields: biology [71], botany [59], cognitive sciences [86], ecology [90], geology [93], natural language processing [37, 57, 61, 65], paleontology [88,93], psychology [58,104], web sciences [18,87], and so on. In the field of computer science, Broder et al [18, 19] introduced "shingles" and min-wise independent permutations for sketching the Jaccard distance; the sets in their case were the web documents, viewed as a bag of words. Charikar [23] gave a way of sketching arbitrary non-negative vectors in a way that preserves their generalized Jaccard distance.

The Jaccard median problem itself was studied more than two decades ago. Späth [97] showed a "canonical" structural property of the optimal Jaccard median: for each coordinate, its value has to agree with that of some input. This makes the search space finite, albeit exponential ($|\mathcal{S}|^n$). Watson [105] gave a vertex-descent algorithm for Jaccard median and showed that his algorithm terminates and always returns an optimal median. Unfortunately, he did not show any bounds on its running time. In fact, other than these two pieces of work, nothing substantial is known about the complexity of finding or approximating the Jaccard median.

**Our results.** Here, following [28], we fully study the computational complexity of the general Jaccard median problem. We begin by showing that the problem is NP-hard. Interestingly, our proof shows that the Jaccard median problem remains NP-hard even in the following two special cases: (a) when the input sets are not allowed to be repeated (i.e., $\mathcal{S}$ cannot be a multi-set) and (b) when all the input sets consists of exactly two elements (i.e., $|X| = 2, \forall X \in \mathcal{S}$) but the sets themselves are allowed to be repeated (i.e., $\mathcal{S}$ can be a multi-set). Our proofs in fact show that unless P = NP, there can be no FPTAS for finding the Jaccard median.

We then consider the problem of approximating the Jaccard median. Our main result is a polynomial-time approximation scheme (PTAS) for the general Jaccard median problem. While it is trivial to obtain a 2-approximation for the problem (the best of the input vectors achieves this and this bound is tight, see section 4.7.1), obtaining a $(1 + \varepsilon)$-approximation turns out to require new ideas, in particular, understanding the structure of the optimal solution.

We first show how to find a $(1 + \varepsilon)$-approximate median for the binary (i.e., set) version of the Jaccard metric. This is done by combining two algorithms. The first algorithm uses random projections on a carefully selected subspace and outputs an additive approximation; the quality translates to a multiplicative approximation provided the optimum is large. The second algorithm focuses on the case when the optimum is small and obtains a multiplicative approximation — this algorithm leverages on certain structural properties of an optimal solution.

To obtain a PTAS for the general Jaccard median problem, we consider three different cases. If the value of the optimum is very small ($O(\varepsilon)$), then we show how the Jaccard median problem can be "linearized" and give a PTAS based on linear programming. If the value of the optimum is $\Omega(\varepsilon)$, then there are two sub-cases. If the ratio between the maximum and the minimum coordinate values is polynomial, then we map the input instance to a polynomially-sized binary instance, solve using the PTAS for the binary case, and show how this approximate solution can be mapped back to an approximate solution in the original space. If the ratio between the maximum and the minimum coordinate values is super-polynomial, then we show how one can modify the instance so as to guarantee that the ratio becomes polynomial and that each approximate solution for the modified instance is also an approximate solution for the original instance.

**Related work.** The median problem has been actively studied for many different metric spaces. The hardness of finding the best median for a set of points out of a (typically exponentially) large set of candidates strictly depends on the metric considered. For instance, the median problem has been shown to be hard for edit distance on strings [39,85], for the Kendall $\tau$ metric on permutations [6,45], but can be solved in polynomial time for the Hamming distance on sets (and more generally, for the $\ell_1$ distance on real vectors), and for the Spearman footrule metric on permutations [45]. The general metric $k$-median problem has also been studied in the literature; see, for example, [4, 24].

## 4.2 Preliminaries

Let $U = \{x_1, \ldots, x_n\}$ be the ground set.

**Definition 42 (Binary Jaccard measures).** *Given $X, Y \subseteq U$, their* Jaccard similarity *is defined as*

$$J(X,Y) = \begin{cases} \frac{|X \cap Y|}{|X \cup Y|} & \text{if } X \cup Y \neq \varnothing, \\ 1 & \text{if } X \cup Y = \varnothing, \end{cases}$$

*and the* Jaccard distance *is defined as $D(X,Y) = 1 - J(X,Y)$.*

It is known that $D(X,Y)$ is a metric; see, for instance, [23]. Let $S_1, \ldots, S_m$ be (not necessarily distinct) subsets of $U$, and let $\mathcal{S} = \{S_1, \ldots, S_m\}$; let $X \subseteq U$. We define the Jaccard similarity between $X$ and $\mathcal{S}$ to be $J(X,\mathcal{S}) = \sum_{Y \in \mathcal{S}} J(X,Y)$. If $X \neq \varnothing$, we have

$$J(X,\mathcal{S}) = \sum_{Y \in \mathcal{S}} \frac{|X \cap Y|}{|X \cup Y|} = \sum_{x \in X} \sum_{Y \ni x} \frac{1}{|Y \cup X|}.$$

The Jaccard distance of $X$ to $\mathcal{S}$ is defined to be $D(X,\mathcal{S}) = \sum_{Y \in \mathcal{S}} D(X,Y) = |\mathcal{S}| - J(X,\mathcal{S})$.

**Definition 43 (Jaccard distance median).** *For a given $\mathcal{S}$, $M^* \subseteq U$ is said to be an* optimal Jaccard distance (1-)median *if $D(M^*, \mathcal{S}) = \min_{X \subseteq U} D(X, \mathcal{S})$. For $\alpha \geq 1$, $M \subseteq U$ is said to be an* $\alpha$-approximate Jaccard distance (1-)median*, if $D(M^*, \mathcal{S}) \leq D(M, \mathcal{S}) \leq \alpha D(M^*, \mathcal{S})$.*

Analogously, a median problem with respect to Jaccard similarity can be defined (observe, though, that an approximation algorithm for Jaccard distance median might not approximate the Jaccard similarity median, and viceversa). Unless otherwise specified, we use Jaccard median to denote the Jaccard distance median problem. We assume throughout the paper that $\varnothing \notin \mathcal{S}$. The case $\varnothing \in \mathcal{S}$ is easy — just check the value of $\varnothing$ as a candidate Jaccard median, remove $\varnothing$ from $\mathcal{S}$, solve for the remaining sets, and then return the best solution.

For an element $x \in U$, we will refer to the number of sets it is present as its degree. Thus, $\deg_{\mathcal{S}}(x) = |\{S \in \mathcal{S} : x \in S\}|$. When it is clear from the context, we will simply write $\deg(x)$.

The Jaccard measures can be generalized to non-negative real vectors (sets being binary vectors); the corresponding Jaccard distance is also known to be a metric.

**Definition 44 (Generalized Jaccard measures).** *Given two non-negative $n$-dimensional real vectors $X, Y$, their* Jaccard similarity *is defined as*

$$J(X,Y) = \begin{cases} \frac{\sum_{i=1}^{n} \min(X_i, Y_i)}{\sum_{i=1}^{n} \max(X_i, Y_i)} & \text{if } \sum_{i=1}^{n} \max(X_i, Y_i) > 0, \\ 1 & \text{if } \sum_{i=1}^{n} \max(X_i, Y_i) = 0, \end{cases}$$

*and the* Jaccard distance *is defined as $D(X, Y) = 1 - J(X, Y)$.*

### 4.2.1 Tanimoto Similarity

Let us now comment on how our results relate to the so-called Tanimoto Similarity between non-negative real vectors, defined to be

$$T(V,W) = \begin{cases} \frac{V \cdot W}{V \cdot V + W \cdot W - V \cdot W} & \text{if } V \cdot V + W \cdot W - V \cdot W > 0 \\ 1 & \text{otherwise} \end{cases}$$

where $A \cdot B$ is the dot product between vectors $A$ and $B$, $A \cdot B = \sum_i A(i) \cdot B(i)$.

We point out that the Tanimoto Similarity has often been confused with the Jaccard one. While they do coincide on binary vectors, they are generally different; for instance, take the 1-dimensional vectors (1) and (2) — their Jaccard similarity is $\frac{1}{2}$, while their Tanimoto similarity is $\frac{2}{3}$.

Further, the so-called Tanimoto Distance $D_T(V, W) = 1 - T(V, W)$ is in general *not* a metric. Indeed we have $D_T((1),(2)) + D_T((2),(3)) = \frac{1}{3} + \frac{1}{7} = \frac{10}{21} < \frac{12}{21} = \frac{4}{7} = D_T((1),(3))$; that is, the triangle inequality is violated by the triple of points $(1), (2), (3)$.

On the other hand, there exist sets of real vectors that where $d_T$ is actually a metric. Lipkus [78] showed that, for any $n \geq 1$, and for any sequence of non-negative real numbers $w_1, w_2, \ldots, w_n$, the set of vectors $\mathcal{W} = \{0, w_1\} \times \{0, w_2\} \times \cdots \times \{0, w_n\}$ (that is, the set of all vectors $V$ such that, for each coordinate $1 \leq i \leq n$, either $V(i) = w_i$ or $V_i = 0$) forms a metric under the Tanimoto distance $D_T$.

We observe that $\mathcal{W}$ under the Tanimoto distance isometrically embeds into the Jaccard space. Indeed, if we map each vector $V \in \mathcal{W}$, to the vector $\phi(V) = (V(1)^2, V(2)^2, \ldots, V(n)^2)$, then for each pair of vector $V, W \in \mathcal{W}$ it holds that $D_T(V, W) = D(\phi(V), \phi(W))$ — that is, the embedding is isometric. It is then easy to check that if $M$ is an $\alpha$-approximate median for some subset of $\phi(\mathcal{W})$ with the Jaccard distance, then $(\sqrt{M(1)}, \sqrt{M(2)}, \ldots, \sqrt{M(n)})$ is an $\alpha$-approximate median for the corresponding subset of $\mathcal{W}$ with the Tanimoto distance. It follows that our algorithm is a PTAS for the subset of real vectors that Lipkus pointed out to form a metric under the Tanimoto distance.

## 4.3 Hardness of the Jaccard median

In this section we study the hardness of the Jaccard median problems. Since our focus will be on finding the optimum, both Jaccard distance median and Jaccard similarity median can be treated interchangeably, i.e., the optimal Jaccard distance median is the same as the optimal Jaccard similarity median.

First, we describe a gadget that will be central in our reductions; this gadget appears to be "unique" in many aspects. For $t \in \mathbb{Z}^+$, let $B_t = K_{3t,3t-2}$ be the complete bipartite graph; let $L$ denote the set of nodes on the left side, $R$ denote the set of nodes on the right side, and $C$ denote the set of edges in $B_t$. Let $U = L \cup R$ and each edge $e = (u, v)$ in $C$ represents the set $\mathcal{S}_e = \{u, v\}$ and let $\mathcal{S}_B = \cup_{e \in C}\{\mathcal{S}_e\}$ be an instance of the Jaccard median problem.

Let $\mathcal{M}_B^*$ denote the set of all subsets of $U$ such that for each $M^* \in \mathcal{M}_B^*$, we have $|L \cap M^*| = t$ and $R \subseteq M^*$, i.e., each $M^* \in \mathcal{M}_B^*$ consists of exactly $t$ nodes from $L$ and all nodes from $R$. We show that the optimal Jaccard median[2] must come from the set $\mathcal{M}_B^*$ and quantify the gap between any near-optimal solution (proved in Appendix 4.7.2).

**Lemma 45.** *For the instance $\mathcal{S}_B$, every $M^* \in \mathcal{M}_B^*$ is an optimal median and $J(M^*, \mathcal{S}_B) > 3t - 2$. Furthermore, $J(M^*, \mathcal{S}_B) - J(M, \mathcal{S}_B) \geq t^{-2}/32$ for $M^* \in M_B^*$ and $M \notin M_B^*$.*

In our reductions we will overlay a graph on $L$, bijectively mapping nodes to $G$ to nodes in $L$. There are two competing forces in play for selecting the best Jaccard median. On the one hand, the gadget ensures that we want to select exactly $t$ nodes from $L$; on the other we would like to pick the densest subset in $G$. We make sure the gain from selecting exactly $t$ nodes from $L$ is a stronger force, either by duplicating every edge in $\mathcal{S}_B$ as in section 4.3.1, or diluting the contribution of edges in $G$, as in section 4.3.2. Given that the optimum median selects exactly $t$ nodes from $G$, we show that it must select those forming the $t$-densest subgraph.

### 4.3.1 The multi-set, edge case

In this section we show that the Jaccard median problem restricted to the case when each set $S$ in the instance $\mathcal{S}$ has exactly two elements from the universe (i.e., each set can be thought of as an "edge" in a graph whose nodes are the elements of the universe) is NP-hard. However, we need to allow $\mathcal{S}$ to be a multi-set.

Our reduction will use a custom-defined problem, whose NP-hardness we will prove in Section 4.7.2. Given a graph $G(V, E)$, having maximum degree $\Delta \geq \frac{1}{3} \cdot |V|$, and having no node $v \in V$ such that $\frac{5}{18} \cdot |V| < \deg(v) < \Delta$, the $\frac{1}{3}$-Quasi-Regular-Clique problem is defined to be: does $G$ contain a clique of size at least $\frac{1}{3} \cdot |V|$?

**Theorem 46.** *The Jaccard median problem, where each set in the instance has two elements, is NP-hard.*

We prove the NP-hardness by reducing from $\frac{1}{3}$-Quasi-Regular-Clique. Without loss of generality, assume $|V| = 3t$, where $t \in \mathbb{Z}^+$. We consider the bipartite gadget $B_t = (L, R, C)$ described before and for each edge in $C$, replicate it $320t^5$ times in order to obtain the bipartite multi-graph $B = (L, R, C')$. Next we overlay the graph $G(V, E)$ onto $L$, bijectively mapping nodes in $V$ to nodes in $L$ and adding appropriate edges among the nodes in $L$ according to $E$; let $B' = (L, R, C' \cup E)$ be the resulting multi-graph.

Each edge $e = (u, v)$ in $B'$ is interpreted as the set $S_e = \{u, v\}$. Let $\mathcal{S}_B = \cup_{e \in C'} S_e$ be the family corresponding to the edges in $B$ and let $\mathcal{S}_G = \cup_{e \in E} S_e$ be the family corresponding to the edges in $G$. Observe that each set $M \in \mathcal{M}_B^*$ (that is, each set $M = R \cup L'$, with $L' \subseteq L$ and $|L'| = t$), has the same Jaccard similarity $c_1 = J(M, \mathcal{S}_B)$ to $\mathcal{S}_B$. Define $c_2 = \binom{t}{2}\frac{2}{4t-2} + t\,(\Delta - (t-1))\,\frac{1}{4t-1}$, where $\Delta$ is the maximum degree in the $\frac{1}{3}$-Quasi-Regular-Clique instance. We ask: does there exist a Jaccard median $M^*$ of total Jaccard similarity $J(M^*, \mathcal{S}) \geq c_1 + c_2$?

We continue the proof in Appendix 4.7.2.

**Corollary 47.** *The Jaccard median problem, where each set in the instance has two elements, does not admit an FPTAS if $P \neq NP$.*

*Proof.* In the proof of Theorem 46, we have shown it is NP-hard to approximate the Jaccard median problem to within an additive factor $\frac{2}{(4t-2)(4t-1)}$. In our instances, $m = \Theta(t^7)$ and $n = \Theta(t)$. Note that the number of sets $m$ is an upper bound on the total Jaccard distance of any median. It follows that it is NP-hard to approximate the Jaccard median problem to within a multiplicative factor of $1 + o\left(m^{-9/7}\right)$ or $1 + o\left(n^{-9}\right)$. It follows that no FPTAS exists for the problem if P $\neq$ NP.

[2]We remark that, in this regard, $K_{3t,3t-2}$ seems to be very special — choosing $K_{at,at-b}$, for $a$ not a multiple of 3 or $b \neq 2$, does not seem to give equal or similar guarantees.

### 4.3.2 The set, hyperedge case

In this section we show that the Jaccard median problem restricted to the case when $\mathcal{S}$ is not a multi-set, is NP-hard. However, we need that the sets in the instances have cardinalities more than two, i.e., they are like "hyperedges".

**Theorem 48.** *The Jaccard median problem, where the instance does not contain duplicate sets, is NP-hard.*

*Proof.* As before, we prove the NP-hardness by reducing from $\frac{1}{3}$-Quasi-Regular-Clique. The steps of the reduction are similar to the previous case. Let $|V| = 3t$ and we consider $B = B_t = (L, R, C)$. Next we overlay the graph $G$ onto $L$, bijectively mapping nodes in $V$ to nodes in $L$ and adding appropriate edges among the nodes in $L$ according to $E$; let $B' = (L \cup R, C \cup E)$ be the resulting graph.

From $B'$, we construct an instance of the Jaccard median problem, whereby for each edge $e = (u, v)$ in $B'$ that came from $B$, we create the set $S_e = \{u, v\}$ and for each edge $e = (u, v)$ in $B'$ that came from $G$, we create the set $S_e = \{u, v, \alpha_e^1, \ldots, \alpha_e^k\}$ where $k = t^7$. Since each edge has unique $\alpha_e^i$'s, these $\alpha$ nodes have degree one and we refer to them as *fake nodes* as they belong neither to $G$ nor to $B$. Let $\mathcal{S}_B = \cup_{e \in C} S_e$ be the family corresponding to the edges in $B$ and let $\mathcal{S}_G = \cup_{e \in E} S_e$ be the family corresponding to the edges in $G$. Let $\mathcal{S} = \mathcal{S}_G \cup \mathcal{S}_B$ be the instance of the Jaccard median problem and let $M^*$ be its optimal Jaccard median.

**Lemma 49.** *Given $M^*$ as above, $M^* \in \mathcal{M}_B^*$ and that the subgraph in $G$ induced by the nodes in $L^* = M^* \cap L$ is a clique.*

The proof is in Appendix 4.7.2. It is easy to see that the reduction from $\frac{1}{3}$-Quasi-Regular-Clique is complete.

We note that the same no-FPTAS claim that we made for Jaccard median with 2-elements set, can be made for Jaccard median with no duplicate sets.

## 4.4 A PTAS for binary Jaccard median

First, we consider the binary Jaccard median problem. Here, we split the analysis based on the quality of the (yet) unknown optimal median. First, suppose the optimal median is large, say, $\Omega(\varepsilon m)$. In this case we obtain an algorithm (Section 4.4.1) that returns an additive $O(\varepsilon^2 m)$-approximation to the optimal median; clearly, this additive approximation translates to a $(1 + O(\varepsilon))$-multiplicative approximation. Next, we obtain an algorithm (Section 4.4.2) that returns a $(1 + O(\varepsilon))$-multiplicative approximation, assuming the optimal median has value $O(\varepsilon^2 m)$. Thus, by running the two algorithms in tandem, and returning the better solution, we are guaranteed to have a PTAS.

### 4.4.1 A PTAS when the optimal median is large

In this section we show how to obtain an additive $O(\varepsilon m)$-approximation in time $(nm)^{\frac{1}{\varepsilon^{O(1)}}}$. As stated before, when the optimal median is $\Omega(\varepsilon m)$, this immediately gives a PTAS. This algorithm first guesses the number of elements in the optimal median, and then proceeds to "densify" the instance, removing the sets whose sizes are too far away from the size of the optimal median, and removing those elements that are not present in too many sets. Intuitively, these steps will be justified since the sets whose sizes are too far away from optimal will always be far, regardless of the actual choice of median and removing elements that appear in a small number of sets will not affect the solution too much.

If the dense instance has too many elements, we subsample further in order to reduce the total number of elements to at most $O(\log(nm)/\varepsilon^6)$. At this point we can afford to try all of the possible subsets, to find a solution $M_c$, which we call the *seed median*, which will be optimal on this restricted space. Finally, we show how to generalize the seed median to the full space of dense elements by solving a linear program and then rounding it randomly.

The flow of the algorithm is presented below:
(1) Guess $t$, the size of the optimal median $M^*$.
(2) Densify the instance by considering only the set family: $\mathcal{S}_t = \{S_j \in \mathcal{S} \mid \varepsilon t \leq |S_j| \leq \frac{t}{\varepsilon}\}$. Keep only

the elements $U_t$ present in at least $\varepsilon m$ sets in $\mathcal{S}_t$.
(3) If $|U_t| \leq 9\varepsilon^{-6}\ln(nm)$, then try all subsets of $U_t$, and return its subset $M$ minimizing $D(M, \mathcal{S})$.
(4) Otherwise (a) subsample elements $\mathcal{P}_t \subseteq U_t$ by selecting each element with probability $\frac{9\ln(nm)}{\varepsilon^6|U_t|}$ and (b) for every subset $M_c$ of $\mathcal{P}_t$, generalize this seed median $M_c$ from a solution on $\mathcal{P}_t$ to a solution $M$ on $U_t$. Finally return $M$ that minimizes $D(M, \mathcal{S})$.

We give the complete description of the algorithm in Appendix 4.7.9.

Note that the median returned by the algorithm consists of only the elements in $U_t$. We first show that restricting only to sets in $\mathcal{S}_t$ adds at most an $\varepsilon m$ to the cost of the solution (Lemma 50, proved in Appendix 4.7.3); then we show that by restricting only to the elements in $U_t$ increases the cost by at most an additional $\varepsilon m$ (Lemma 51, proved in Appendix 4.7.3).

**Lemma 50.** *Fix $t$ and $\mathcal{S}_t$ as above. Let $M^*$ be the optimal median for $\mathcal{S}$, $M_t^*$ be the optimal median for $\mathcal{S}_t$, and $M$ be such that $D(M, \mathcal{S}_t) \leq D(M_t^*, \mathcal{S}_t) + \alpha$. Then $D(M, \mathcal{S}) \leq D(M^*, \mathcal{S}) + \alpha + \varepsilon m$.*

**Lemma 51.** *Fix an arbitrary integer $k$, and let $M$ be any subset of $U$. If $T \subseteq U$ is the set of elements of degree $\leq k$ then $D(M \setminus T, \mathcal{S}) \leq D(M, \mathcal{S}) + k$.*

So far we have shown that the optimal median on the instance consisting of $\mathcal{S}_t$ with the elements in $U_t$ is an $(\varepsilon m)$-approximate median to the original instance. Now, if $|U_t|$ is sufficiently small, i.e., $|U_t| = O(\frac{\ln nm}{\varepsilon^6})$, then we can just enumerate all of the subsets of $U_t$ to find the optimal median.

Otherwise (i.e., $U_t$ is relatively large), we proceed to subsample elements from $U_t$ with probability $p = \frac{9\ln(nm)}{\varepsilon^6|U_t|}$. Let $P \subseteq U_t$ be the set of sampled elements. An easy application of Chernoff bound shows that $|P| = O(\ln(nm)/\varepsilon^6)$ with high probability. Furthermore, as the following shows, the size of the intersection between any two sets $A, B \subseteq U_t$ is either small, or is well-preserved; proof is in Appendix 4.7.3

**Lemma 52.** *For any $A, B \subseteq U_t$, and let $C = A \cap B$. Then, with probability $\geq 1 - O(nm)^{-3}$, if $|C| \geq \varepsilon^4|U_t|$, then $(1-\varepsilon)p|C| \leq |C \cap P| \leq (1+\varepsilon)p|C|$ and if $|C| < \varepsilon^4|U_t|$, then $|C \cap P| \leq 6\varepsilon^4 p|U_t|$.*

At this point the algorithm proceeds to look at all possible subsets of $P$ as the seed medians, $M_c$. We now show how to generalize the seed to a median on the full set $U_t$. Let $M_t^*$ be the optimal median on $U_t$ and let $M_c = M_P^* = M_t^* \cap P$. The condition we require is that the generalization of $M_P^*$ to the ground set $U_t$ happens to be an $\varepsilon m$ (additive) approximate median on $\mathcal{S}_t$.

For a candidate median $M_c$, let $\mathcal{S}_t(M_c) \subseteq \mathcal{S}_t$ be the sets that have a "large-enough" intersection with $M_c$. Formally, let $\mathcal{S}_t(M_c) = \{S \in \mathcal{S}_t \mid |M_c \cap S| > \frac{54\ln(nm)}{\varepsilon^2}\}$. To generalize $M_c$, we solve the following system $\mathcal{L}$ of linear inequalities: $(\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x}_{|\mathbf{U_t}|})$

$$\mathcal{L} = \begin{cases} 0 \leq \mathbf{x_i} \leq 1 & \forall i, 1 \leq i \leq |U_t| \\ \sum_{x_i \in S \cap U_t} \mathbf{x_i} \leq (1-\varepsilon)^{-1} \cdot |S \cap M_c| \cdot p^{-1} & \forall S \in \mathcal{S}_t(M_c) \\ \sum_{x_i \in S \cap U_t} \mathbf{x_i} \geq (1+\varepsilon)^{-1} \cdot |S \cap M_c| \cdot p^{-1} & \forall S \in \mathcal{S}_t(M_c) \\ \sum_{x_i \in U_t} \mathbf{x_i} \leq (1-\varepsilon)^{-1} \cdot |M_c| \cdot p^{-1} & \\ \sum_{x_i \in U_t} \mathbf{x_i} \geq (1+\varepsilon)^{-1} \cdot |M_c| \cdot p^{-1} & \end{cases}$$

If there exists a solution $(\hat{\mathbf{x}}_\mathbf{1}, \ldots, \hat{\mathbf{x}}_{|\mathbf{U_t}|})$, compute $M$ by select each element $x_i \in U_t$ with probability $\hat{\mathbf{x}}_i$ independently.

We begin by showing that unless the optimum solution $M^*$ has a very small intersection with $U_t$, there will be some solution to the set $\mathcal{L}$ of linear inequalities. We say that some subset $Y \subseteq U_t$, we defined its $\mathcal{L}$-assignment as $\{\mathbf{y}_i\}_{i=1}^{|U_t|}$, where $\mathbf{y}_i = 1$ if $x_i \in Y$ and $\mathbf{y}_i = 0$ otherwise.

**Lemma 53.** *Let $M^*$ be the optimal median with $|M^*| = t$. Fix $U_t$, and let $M_t^* = M \cap U_t$. Select $P \subseteq U_t$ as above, and let $M_P^* = M^* \cap P$. Then, either $|M_t^*| < \varepsilon^2 t$ or with high probability, the $\mathcal{L}$-assignment of $M_t^*$ satisfies $\mathcal{L}$.*

**Theorem 54.** *Let $M^*$ be the optimal median, and $M$ be the best median produced by the algorithm above. Then, with high probability $D(M^*, \mathcal{S}) \leq D(M, \mathcal{S}) + O(\varepsilon m)$.*

*Proof.* As before, let $t = |M^*|$, and use $U_t$ and $P$ as above. For ease of notation, denote by $M_t^* = M^* \cap U_t$ and $M_P^* = M^* \cap P$. And suppose the conditions of Lemma 53 hold. Let $M$ be the solution reconstructed by the algorithm when $M_c = M_P^*$, or $M = \varnothing$ if $|M_t^*| < \varepsilon^2 t$.

Let $\mathcal{S}_N = \{S \in \mathcal{S}_t \mid |S \cap M_t^*| \geq \frac{6\varepsilon^2 t}{(1-\varepsilon)}\}$. Observe that for every set $S \in \mathcal{S}_t \setminus \mathcal{S}_N$,

$$D(M_t^*, S) \geq 1 - \frac{\frac{6\varepsilon^2 t}{1-\varepsilon}}{\varepsilon t} = 1 - \frac{6\varepsilon}{(1-\varepsilon)} = 1 - O(\varepsilon).$$

Therefore for such sets $S$ any median $M$, $D(M, S) \leq D(M^*, S) + O(\varepsilon)$.

To bound $D(M, \mathcal{S})$, observe that:

$$D(M, \mathcal{S}) = D(M, \mathcal{S}_N) + D(M, \mathcal{S}_t - \mathcal{S}_N) + D(M, \mathcal{S} \setminus \mathcal{S}_t).$$

Lemmas 50 and 51 imply that $D(M, \mathcal{S} \setminus \mathcal{S}_t) \leq D(M^*, \mathcal{S} \setminus \mathcal{S}_t) + O(\varepsilon m)$. Therefore what remains to show is that the median $M$ is such that $D(M, S_N) \leq D(M_t^*, S_N) + O(\varepsilon m)$.

Suppose that $|M_t^*| < \varepsilon^2 t$, then $\mathcal{S}_N = \varnothing$ and the proof is complete. Otherwise, for each set $S \in S_N$, notice that $|S \cap M_t^*| \geq 6\varepsilon^2 t(1-\varepsilon)^{-1} \geq 6\varepsilon^4 |U_t|(1-\varepsilon)^{-1}$, and therefore $|S \cap M_t^* \cap P| \geq 6\varepsilon^4 |U_t| p = \frac{54 \ln(nm)}{\varepsilon^2}$. Therefore $\mathcal{S}_N \subset \mathcal{S}_t(M_c)$.

Let $\mathbf{y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_{|U_t|}\}$ be any solution to the system $\mathcal{L}$ when $M_c = M_P^*$. Then for every $S \in \mathcal{S}_N$ we have that

$$\sum_{x_i \in S \cap U_t} \mathbf{y}_i \geq \frac{|S \cap M_P^*|}{(1+\varepsilon) \cdot p}.$$

Since $|S \cap (M_t^* \cap P)| = |S \cap M_P^*| \geq \varepsilon^{-2} 54 \ln nm$, an easy application of Chernoff bounds shows that with high probability the randomized rounding of $\mathbf{y}$ will approximate $|S \cap M_t^*|$ to within a $(1 \pm \varepsilon)$ factor. That combined with the fact that $\sum_i \mathbf{y}_i$ is also concentrated with high probability, implies that for any $S \in \mathcal{S}_N$, $J(M, S) \geq J(M_t^*, S) - O(\varepsilon)$; thus $D(M, S_N) \leq D(M_t^*, S_N) + O(\varepsilon m)$. The proof is complete.

In the next sections we show a polytime algorithm that produces a $(1 + O(\sqrt{\varepsilon}))$-approximate median if the optimal median has value $\leq \varepsilon m$. The two algorithms together form a PTAS.

### 4.4.2 A PTAS when the optimal median is small

In this section we provide an algorithm that works when the optimal median is very good, and the average distance from a set to the median is $\varepsilon$.

**Definition 55 ($\varepsilon$-good instance).** *An instance $\mathcal{S}$ on $m$ sets is* $\varepsilon$-good *if the cost of the optimal median is less than $\varepsilon m$.*

In this section we show an algorithm that achieves a $(1 + O(\sqrt{\varepsilon}))$ approximate median to $\varepsilon$-good instances in time $O(nm)$.

We begin by proving several structural properties of any $\varepsilon$-good instance. First, for the instance $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$, denote by $\mu$ the median of the input sizes, $\{|S_1|, |S_2|, \ldots, |S_m|\}$.

Any $\varepsilon$-good instance has the following properties:

- The size of the best median set, $M^*$, is $(1 \pm O(\varepsilon))\mu$. (Lemma 56).
- There are many – $(1 - O(\sqrt{\varepsilon}))\mu$ high degree elements (elements present in at least $(1 - O(\sqrt{\varepsilon}))m$ sets), and all of them are part of each near optimal median. (Lemma 57 and Lemma 58).

This set of properties suggests the following natural algorithm:

1. Find the set of all high degree elements, and add them to the optimal median (This adds at least $(1 - O(\sqrt{\varepsilon}))\mu$ elements).
2. Greedily select another $O(\sqrt{\varepsilon} + \varepsilon)\mu$ elements to add to the median. Since we are only adding a small number of extra elements to the set, denominator does not change by much, but the size of respective intersections is maximized.

We now proceed to formalize the statements of the lemmas and analyze the final algorithm.

**Lemma 56.** *Fix $0 < \varepsilon_1 \leq \frac{1}{6}$. If a set $M \subseteq X$ is such that $d(M, \mathcal{S}) \leq \varepsilon_1 \cdot m$ then*

$$(1 - 3\varepsilon_1) \cdot \mu \leq |M| \leq (1 + 3\varepsilon_1) \cdot \mu.$$

Intuitively, consider a median $M$ with $|M| > (1 + \varepsilon)\mu$. Then on at least half of the sets (those whose sizes are less than $\mu$), the distance between $M$ and $S_i$ will be at least $\frac{1}{1+\varepsilon}$, leading to a contradiction of goodness of $M$. We present the formal proof in Appendix 4.7.4.

We next lower bound the number of high degree elements (proof in Appendix 4.7.4). Let $M^*$ be the optimal median, and let $D = d(M^*, \mathcal{S})$.

**Lemma 57.** *Fix some $0 < \varepsilon_2 \leq \frac{2-\sqrt{3}}{3}$. We say that an element $j \in X$ has high degree if $\deg(j) = |\{S_i \mid j \in S_i \in \mathcal{S}\}| \geq (1 - \sqrt{2\varepsilon_2}) \cdot m$. If $D \leq \varepsilon_2 \cdot m$ then there exist at least $(1 - \sqrt{2\varepsilon_2}) \cdot \mu$ high degree elements.*

Finally, the next lemma states that each high degree element is part of any near optimal median (proof in Appendix 4.7.4).

**Lemma 58.** *Fix $0 < \varepsilon_4 < \frac{3}{100}$. Let $X^* \subseteq X$ be the set of the elements having degree $\geq \left(1 - \sqrt{\varepsilon_4}\right) m$. Take any $M \subseteq X$ such that $d(M, \mathcal{S}) \leq \varepsilon_4 \cdot m$. If $X^* - M \neq \varnothing$, it holds that $d(M \cup X^*, \mathcal{S}) < d(M, \mathcal{S})$.*

At this point we know that every near optimal median contains no more than $(1+O(\varepsilon))\mu$ elements, out of which at least $(1-O(\sqrt{\varepsilon}))$ are the easily found dense elements. Thus, we need to chose at most $O(\sqrt{\varepsilon}\mu)$ extra elements to include in the solution. The difficulty of finding the optimal median stems from the fact that as we add extra elements to a candidate median, the total contribution of each set to the overall distance changes due to *both* the change in the numerator and the change in the denominator. However, since we have an approximation to the bound on the size of the optimal median, we can effectively freeze the denominators, knowing that we are making at most an $1 + \sqrt{\varepsilon}$ approximation to the final solution. Once the denominators are frozen, the problem is simpler and the greedy algorithm is optimal.

Formally, let $M$ be the set of at least $(1 - O(\sqrt{\varepsilon}))\mu$ dense elements guaranteed by Lemma 58. For an element $x_i \notin M$ let the weight of $x$ be $\sum_{S_j \ni x_i} \frac{1}{|S_j \cup M|} - \sum_{S_j \not\ni x_i} \frac{1}{|S_j \cup M|}$. Set $N^*$ to be the set found by by greedily selecting elements in order of decreasing weight, stopping when either (a) the size of $N^*$ is $O(\sqrt{\varepsilon})$ or (b) the weight of the element in consideration is non-positive.

**Theorem 59.** *Let $M$ and $N^*$ as above. Then $D(M^*, \mathcal{S}) \geq \frac{1}{1+O(\sqrt{\varepsilon})} \cdot D(M \cup N^*, \mathcal{S})$.*

We give the proof in Appendix 4.7.4. Therefore, the solution $M \cup N^*$ found by the algorithm is an $(1 + O(\sqrt{\varepsilon}))$ approximation to the optimal median.

## 4.5 A PTAS for generalized Jaccard median

In the generalized Jaccard median problem, we are given a (multi-)set of vectors $\mathcal{V} = \{V_1, \ldots, V_m\}$, where the generic $V_i$ is a real vector on $n$ non-negative coordinates, $V_i \in \mathbf{R}^n_{\geq 0}$.

Then, the Jaccard distance between two such vectors is

$$D(V, W) = 1 - \frac{\sum_{i=1}^n \min(V(i), W(i))}{\sum_{i=1}^n \max(V(i), W(i))} = \frac{\sum_{i=1}^n |V(i) - W(i)|}{\sum_{i=1}^n \max(V(i), W(i))},$$

if $V$ and $W$ aren't two zero-vectors, and 0 otherwise. If $V, W$ are binary vectors, this simplifies to the previously defined Jaccard set distance.

We give a solution to the median problem on non-binary instances. We defer the technical details to Appendix 4.7.5 and give a high-level view here.

The algorithm of Section 4.7.6 returns a $(1 + O(\varepsilon))$-multiplicative approximate median $M$ if the value of the optimal median $M^*$ is $O(\varepsilon)$ — so, if the total distance between the median and the input vectors is much less than 1. This algorithm is needed as the other two algorithms for general Jaccard (Sections 4.7.7 and 4.7.7) only guarantee to return a median $M$ of total distance $D(M, \mathcal{V}) \leq (1+O(\varepsilon^2)) \cdot D(M^*, \mathcal{V}) + O(\varepsilon^2)$ — that is, they both make a multiplicative error of $(1 + O(\varepsilon^2))$ and an additive error of $O(\varepsilon^2)$. Then, if we run the algorithm of Section 4.7.6, and the algorithms of Section 4.7.7-4.7.7, and return the best

solution they produce we are guaranteed to return a $(1 + O(\varepsilon))$-approximate median, as either the total distance of the optimal median is $O(\varepsilon)$ — so that the former algorithm will succeed — or it will be $\Omega(\varepsilon)$, in which case the latter algorithms will return a multiplicative $(1 + O(\varepsilon))$ approximate solution. We now comment on the latter two algorithms.

The algorithm of Section 4.7.7 transforms a weighted input instance having "polynomial spread" (that is, such that the ratios between the maximum and the minimum non-zero value of each coordinate are at most polynomial) into a set instance in such a way that an approximate solution for the set instance can then be mapped into the original instance while preserving the stated approximation guarantee. On the other hand, the algorithm of Section 4.7.7 transforms an arbitrary weighted instance into one having polynomial spread in such a way that the solution to the new instance can be mapped back in the original space while preserving the approximation guarantee.

Observe how the generalized Jaccard algorithms might return medians that are not "canonical" — that is, that might contain coordinate values that are not part of any of the input vectors. As shown by [97], each optimal median is canonical — so that limiting the search space to contain only canonical vectors does not reduce the quality of the optimal solution. Therefore one might want to define the Jaccard median problem as one having a finite search space (the one spawned by the coordinate values of its input vectors — having $\leq m^n$ possible solutions). In Section 4.7.8 we show how the "canonical" and the "not-necessarily canonical" problems are essentially the same. We give a polynomial algorithm that transforms a non-canonical median into a canonical one of smaller total distance. This let us give a PTAS for the canonical version of the problem, too. Further, the Lemma of Section 4.7.8 is used to get a non-FPTAS result for "not-necessarily canonical" Jaccard, starting from the non-FPTAS proof for "canonical" Jaccard of Section 4.3.

## 4.6 Conclusions

We have studied the median problem in the Jaccard metric. We gave a PTAS that returns a $(1 + \varepsilon)$-approximate median in time $(nm)^{\frac{1}{\varepsilon^{O(1)}}}$, and showed how the problem does not admit a FPTAS, if P $\neq$ NP. Our PTAS makes use of a number of different algorithmic ideas, while our hardness result leverages on a gadget that appears to be unique in many ways.

It would be interesting to study the $k$-median problem on the Jaccard metric. We know the $k$-median problem on the Jaccard metric to be APX-complete if $k$ is unbounded. It remains an open question whether it admits a PTAS for constant $k > 1$. Also, the similarity-median problem (that is, finding the set, or vector, that maximizes the similarity to the input points) seems to merit consideration. We can show that the easy 2-approximation for median (i.e., return the best of the input points) happens to be a $\Omega(\sqrt{m})$ approximation for the similarity-median problem — on the other hand, we can also show a 2-approximation algorithm for this problem. Does the similarity-median problem admit a PTAS?

## 4.7 Appendix

### 4.7.1 Tightness of 2-approximation

Given a set of points of an arbitrary metric, for which one wants to compute the median, one of the input points in the set is a $\leq \left(2 - \frac{2}{n}\right)$-approximation to the optimal median. Here, we show that this bound is tight for the Jaccard metric. Taker an instance of $n$ sets, $\mathcal{S} = \{S_1, \ldots, S_n\}$, such that $S_i = \{x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n\}$ for $i = 1, \ldots n$. Then, the distance between any two sets in the collection will be $1 - \frac{n-2}{n} = \frac{2}{n}$. Therefore, the optimal point (and in fact, any point) in the collection will act as a median of total distance $(n-1) \cdot \frac{2}{n} = 2 - \frac{2}{n}$. Now consider the set $M = \{x_1, x_2, \ldots, x_n\}$. The distance of $M$ to an arbitrary $S_i \in \mathcal{S}$ will be $1 - \frac{n-1}{n} = \frac{1}{n}$. Its total distance will thus be $n \cdot \frac{1}{n} = 1$. The "optimal-input-point" algorithm is then exactly a $2 - \frac{2}{n}$ approximation to the median problem, even for the Jaccard metric.

### 4.7.2 Proofs from Section 4.3

**Proof of Lemma 45**

*Proof.* Consider any $M \subseteq U$ with $|M \cap L| = a$ and $|M \cap R| = b$. We derive the conditions under which $M$ is an optimal Jaccard median. Specifically, we show that for $M$ to be an optimal Jaccard median, $a = t$ and $b = 3t - 2$. First note that we can explicitly write

$$J(M, \mathcal{S}_B) = ab\frac{2}{(a+b)} + a(3t-2-b)\frac{1}{(a+b+1)} + b(3t-a)\frac{1}{(a+b+1)} = \frac{3t(a+b)^2 - 2a^2}{(a+b)(a+b+1)}.$$

From this,

$$\frac{\partial J(M, \mathcal{S}_B)}{\partial b} = \frac{4a^3 + (4b+3t+2)a^2 + 6abt + 3b^2t}{((a+b)(a+b+1))^2}.$$

Since $\frac{\partial J(M,\mathcal{S}_B)}{\partial b} > 0$ for all $a, b$, we have that $J(M, \mathcal{S}_B)$ is monotonically increasing in $b$ and is hence maximized at $b = 3t - 2$, i.e., if $M$ is an optimal Jaccard median, then $R \subseteq M$.

Likewise, we obtain

$$\frac{\partial J(M, \mathcal{S}_B)}{\partial a} = \frac{a^2(3t-2-4b) - 2ab(2b-3t+2) + 3b^2t}{((a+b)(a+b+1))^2},$$

and using the optimality condition $b = 3t - 2$,

$$\frac{\partial J(M, \mathcal{S}_B)}{\partial a} \mid_{b=3t-2} = (3t-2) \cdot \frac{3t(3t-2) - 2a(3t-2) - 3a^2}{((a+3t-2)(a+3t-1))^2}. \tag{4.1}$$

Since $t \geq 1$, setting (4.1) to zero gives us a quadratic equation in $a$. It is easy to see that the quadratic equation has a positive root at

$$a_r = \left(t - \frac{2}{3}\right) \cdot \left(2\sqrt{1 + \frac{3}{6t-4}} - 1\right).$$

We now show that $a_r \in (t-1, t)$. Since $6t - 4 \geq 0$, we have $a_r > t - \frac{2}{3} > t - 1$. Moreover,

$$\begin{aligned}
a_r &= \left(t - \frac{2}{3}\right) \cdot \left(2\sqrt{1 + \frac{3}{6t-4}} - 1\right) \\
&\leq \left(t - \frac{2}{3}\right) \cdot \left(2 + \frac{3}{6t-4} - 1\right) \text{ by Taylor expansion and since } t \geq 1 \\
&= \left(t - \frac{2}{3}\right) + \frac{1}{2} \\
&< t.
\end{aligned}$$

We then note that $\frac{\partial J(M,\mathcal{S}_B)}{\partial a} \mid_{a=t-1,b=3t-2} > 0$ since (4.1) evaluates to $\frac{(3t-2)(10t-7)}{((a+3t-2)(a+3t-1))^2}$ at $a = t-1$, and $\frac{\partial J(M,\mathcal{S}_B)}{\partial a} \mid_{a=t,b=3t-2} < 0$ since (4.1) evaluates to $\frac{(-2t)(3t-2)}{((a+3t-2)(a+3t-1))^2}$ at $a = t$. Moreover, since $a \in \mathbb{Z}$ in our case, this implies that (4.1) attains its maximum value at either $a = t - 1$ or $a = t$. It is easy to see that the maximum indeed occurs at $a = t$:

$$\begin{aligned}
&J(M, \mathcal{S}_B) \mid_{a=t,b=3t-2} - J(M, \mathcal{S}_B) \mid_{a=t-1,b=3t-2} \\
&= \frac{3t(4t-3) + 4t^2 - 2(2t-1)(4t-1)}{(4t-1)(4t-2)(4t-3)} \\
&\geq \frac{3t-2}{(4t-1)(4t-2)(4t-3)} \geq \frac{t^{-2}}{32}.
\end{aligned}$$

Hence, $M$ is optimal if and only if $M \in \mathcal{M}_B^*$, and for each $M \in \mathcal{M}_B^*$, $J(M, \mathcal{S}_B) = \frac{2t(3t-2)}{4t-2} + \frac{2t(3t-2)}{4t-1} > (3t-2)$. And, the second best solution occurs at $a = t-1$ and $b = 3t-2$ and is lower than the optimum value by $t^{-2}/32$.

**Corollary 60.** *For an instance $\mathcal{S}_B$ where each edge has multiplicity $\ell$, every $M^* \in \mathcal{M}_B^*$ is an optimal median. Furthermore, $J(M^*, \mathcal{S}_B) - J(M, \mathcal{S}_B) \geq \ell \cdot t^{-2}/32$ for $M^* \in \mathcal{M}_B^*$ and $M \notin \mathcal{M}_B^*$.*

**Proof of Theorem 46**

*Proof.* First of all, observe that each clique of size $t$ in the original graph contains only nodes of degree $\Delta$. Further, if such a clique $H$ exists then the median $M^* = H \cup R$ is such that $J(M^*, \mathcal{S}) = c_1 + c_2$. Indeed,

$$\begin{aligned} J(M^*, \mathcal{S}) &= J(M^*, \mathcal{S}_B) + J(M^*, \mathcal{S}_G) \\ &= c_1 + \sum_{\substack{S_e \in \mathcal{S}_G \\ |S_e \cap H| = 2}} \frac{2}{t + |R|} + \sum_{\substack{S_e \in \mathcal{S}_G \\ |S_e \cap H| = 1}} \frac{1}{t + |R| + 1} \\ &= c_1 + \binom{t}{2} \frac{2}{4t - 2} + t \cdot (\Delta - (t-1)) \cdot \frac{1}{4t-1} \\ &= c_1 + c_2 \end{aligned}$$

Conversely, let $\mathcal{S} = \mathcal{S}_G \cup \mathcal{S}_B$ be the instance of the Jaccard median problem and let $M^*$ be one of its Jaccard medians of value $\geq c_1 + c_2$.

Let $L^* = M^* \cap L$. We claim that $M^* \in \mathcal{M}_B^*$ and that the subgraph in $G$ induced by the nodes in $L^*$ is a clique. Supposing the claim is true, it is easy to see that the reduction from $\frac{1}{3}$-Quasi-Regular-Clique is complete.

We now prove the claim. In particular, first we show that $M^* \in \mathcal{M}_B^*$. We have $J(M^*, \mathcal{S}) = J(M^*, \mathcal{S}_G) + J(M^*, \mathcal{S}_B)$. From Corollary 60 we know that $J(M^*, \mathcal{S}_B)$ is maximized when $M^* \in \mathcal{M}_B^*$ (with $J(M^*, \mathcal{S}_B) = c_1$ in that case), and for any $M^* \in \mathcal{M}_B^*$ and $M \notin \mathcal{M}_B^*$, $J(M^*, \mathcal{S}_B) - J(M, \mathcal{S}_B) \geq 320 \cdot t^5 \cdot t^{-2}/32 = 10t^3$. Further, we know that $J(M, \mathcal{S}_G) = \sum_{S_e \in \mathcal{S}_G} J(M, S_e) \leq |\mathcal{S}_G| \leq 9t^2$ for any $M$. Thus, for any $M \notin \mathcal{M}_B^*$ we have that

$$J(M, \mathcal{S}) = J(M, \mathcal{S}_G) + J(M, \mathcal{S}_B) \leq 9t^2 + J(M^*, \mathcal{S}_B) - 10t^3 \leq c_1 - t^3,$$

a contradiction. Hence, $M^* \in \mathcal{M}_B^*$.

Given this, we next claim $J(M^*, \mathcal{S}_G)$ has value $c_2$ if $L^*$ is a clique, and value $\leq c_2 - \frac{2}{(4t-2)(4t-1)}$ otherwise. Suppose $k \leq \binom{t}{2}$ edges of $G$ are completely inside $L^*$. Then at most $\Delta t - 2k$ edges will have a single endpoint in $L^*$ (since the maximum degree is $\Delta$), thus:

$$\begin{aligned} J(M^*, \mathcal{S}_G) &= \sum_{\substack{S_e \in \mathcal{S}_G \\ |S_e \cap L^*| = 2}} \frac{2}{t + |R|} + \sum_{\substack{S_e \in \mathcal{S}_G \\ |S_e \cap L^*| = 1}} \frac{1}{t + |R| + 1} \\ &\leq k \cdot \frac{2}{4t-2} + (\Delta t - 2k) \cdot \frac{1}{4t-1} = k \cdot \frac{2}{(4t-2)(4t-1)} + \frac{\Delta t}{4t-1}. \end{aligned}$$

The latter equals $c_2$ if $k = \binom{t}{2}$. Also, if $L^*$ is not a clique, then $k < \binom{t}{2}$ and $J(M^*, \mathcal{S}_G) \leq c_2 - \frac{2}{(4t-2)(4t-1)}$. Thus $J(M^*, \mathcal{S}) \leq c_1 + c_2 - \frac{2}{(4t-2)(4t-1)}$, a contradiction.

**Proof of Lemma 49**

Let fake$(M)$ denote the set of fake nodes in a subset $M$. First, we show that $M^*$ does not contain any fake nodes, i.e., fake$(M^*) = \varnothing$.

We begin by showing that any $M^*$ must have a high Jaccard similarity score on $\mathcal{S}_B$. Let $M_B^* = M^* \cap (L \cup R)$ denote the non-fake nodes in $M^*$. Suppose $J(M_B^*, \mathcal{S}_B) < 2t$. Then using Lemma 61 we can conclude that:

$$J(M^*, \mathcal{S}) = J(M^*, \mathcal{S}_B) + J(M^*, \mathcal{S}_G) < J(M_B^*, \mathcal{S}_B) + 1.5 \leq 2t + 1.5$$

where $J(M^*, \mathcal{S}_B) < J(M_B^*, \mathcal{S}_B)$ because $\mathcal{S}_B$ does not contain any fake nodes. However, any solution $M' \in \mathcal{M}_B^*$ has $J(M', \mathcal{S}_B) > 2.1t$ for $t > 5$ (from Lemma 45), therefore $M^*$ cannot be the optimal solution.

On the other hand, suppose $M^*$ is such that $J(M_B^*, \mathcal{S}_B) \geq 2t$. Then:

$$J(M_B^*, \mathcal{S}) - J(M^*, \mathcal{S}) = (J(M_B^*, \mathcal{S}_B) - J(M^*, \mathcal{S}_B)) + (J(M_B^*, \mathcal{S}_G) - J(M^*, \mathcal{S}_G)).$$

Lemmas 61 and 62 together show that $(J(M_B^*, \mathcal{S}) - J(M^*, \mathcal{S})) > 0$, which is a contradiction. Hence, $M^*$ does not contain any fake nodes.

Next we show that $M^* \in \mathcal{M}_B^*$. From Lemma 45 we know that $J(M^*, \mathcal{S}_B)$ is maximized when $M^* \in \mathcal{M}_B^*$ and for any $M^* \in \mathcal{M}_B^*$ and $M \notin \mathcal{M}_B^*$, $J(M^*, \mathcal{S}_B) - J(M, \mathcal{S}_B) \geq t^{-2}/32$. Also, from Lemma 61, $J(M^*, \mathcal{S}_G) \leq 2|E|/k$ for any set $M^*$ with $\text{fake}(M^*) = \varnothing$. Hence, for any $M^* \in \mathcal{M}_B^*$ and $M \notin \mathcal{M}_B^*$,

$$J(M^*, \mathcal{S}) - J(M, \mathcal{S}) = (J(M^*, \mathcal{S}_B) - J(M, \mathcal{S}_B)) + (J(M^*, \mathcal{S}_G) - J(M, \mathcal{S}_G)) \geq \frac{t^{-2}}{32} - \frac{2|E|}{k} > 0,$$

since $t \geq 3$ and $k = t^7$. In other words, our choice of parameters guarantees that $M^* \in \mathcal{M}_B^*$; thus, $|L^*| = t$.

Given this, we next claim $J(M^*, \mathcal{S}_G)$ (and therefore $J(M^*, \mathcal{S})$) is maximized when $L^*$ induces a clique in $G$. In particular, let the induced graph contain $f$ full-edges (i.e., edges with both end points in $L^*$) and $h$ half-edges (i.e., edges with exactly one end point in $L^*$ and the other end point in $L \setminus L^*$.) Since the degree of each node in $G$ is bounded by $\Delta$, it is easy to see that $h \leq (|L^*| \cdot \Delta) - 2f = t\Delta - 2f$. By definition,

$$\begin{aligned} J(M^*, \mathcal{S}_G) &= \frac{2f}{4t-2+k} + \frac{h}{4t-1+k} \\ &\leq \frac{2f}{4t-2+k} + \frac{t\Delta - 2f}{4t-1+k} = \frac{2f}{(4t-1+k)(4t-2+k)} + \frac{t\Delta}{4t-1+k} = c \end{aligned}$$

Since $c$ is increasing in $f$, it is maximized when $f = \binom{t}{2}$. Observe that $J(M^*, \mathcal{S}_G)$ actually equals this maximum value if $L^*$ induces a clique since in that case $f = \binom{t}{2}$ and each of the nodes of $L^*$ will have degree $\Delta$ and $h = t\Delta - 2f$. Hence, $L^*$ is a clique iff $J(M^*, \mathcal{S}_G)$ is maximized.

**Lemma 61.** *Fix $t$ large enough, if $|\text{fake}(M)| = O(t^2)$, then $J(M, \mathcal{S}_G) < 0.03$. Otherwise, $J(M, \mathcal{S}_G) < 3/2$.*

*Proof.* For each $e = (u, v) \in E$, let $T_e = M \cap \{u, v\}$ and let $F_e = (M \cap S_e) \setminus \{u, v\}$, i.e., $T_e$ corresponds to the true nodes and $F_e$ corresponds to the fake nodes from set $S_e$ that are present in $M$. Let $T = \cup_{e \in E} T_e$ and $F = \text{fake}(M) = \cup_{e \in G} F_e$. Then,

$$\begin{aligned} J(M, \mathcal{S}_G) &= \sum_{e \in E} J(M, S_e) \\ &= \sum_{e \in E} \left( \frac{|T_e \cap S_e|}{|T \cup F \cup S_e|} + \frac{|F_e \cap S_e|}{|T \cup F \cup S_e|} \right) \\ &\leq \sum_{e \in E} \left( \frac{2}{|T \cup F \cup S_e|} + \frac{|F_e|}{|T \cup F \cup S_e|} \right) \\ &\leq \frac{2|E|}{k} + \sum_{e \in E} \frac{|F_e|}{|F \cup S_e|} \\ &\leq \frac{18t^2}{k} + \frac{|F|}{\max\{|F|, k\}} \end{aligned}$$

If $|F| = O(t^2)$, then since $k = t^7$ and $t \geq 3$, we have $J(M, \mathcal{S}_G) = O(t^{-5}) < 0.03$. Otherwise, $J(M, \mathcal{S}_G) < 18t^{-5} + 1 < 3/2$ for $t \geq 3$.

**Lemma 62.** *Let $M \subset L \cup R$, such that $J(M, \mathcal{S}_G) \geq 2t$. Let $F$ be any set of fake nodes ($|F| > 0$). If $|F| \leq 40t$, then $J(M, \mathcal{S}_B) - J(M \cup F, \mathcal{S}_B) \geq 0.035$ and if $|F| > 40t$, then $J(M^*, \mathcal{S}_B) - J(M^* \cup F, \mathcal{S}_B) \geq 1.55$.*

*Proof.* For any $M$ as specified above, let $f$ be the number of edges in $B$ with both endpoints in $M$ and $h$ be the number of edges in $B$ with one endpoint in $M$. Then,

$$J(M^*, \mathcal{S}_B) = \frac{f}{4t-2} + \frac{h}{4t-1},$$

and the condition on $M$ implies that $f + h \geq 7t^2$.

Since the nodes in $F$ do not have any edges in $B$, we know that

$$J(M \cup F, \mathcal{S}_B) = \frac{f}{4t - 2 + |F|} + \frac{h}{4t - 1 + |F|}.$$

Hence,

$$\begin{aligned} J(M, \mathcal{S}_B) - J(M \cup F, \mathcal{S}_B) &= \frac{f|F|}{(4t-2)(4t-2+|F|)} + \frac{h|F|}{(4t-1)(4t-1+|F|)} \\ &\geq \frac{f|F|}{(4t)(4t+|F|)} + \frac{h|F|}{(4t)(4t+|F|)} \\ &= \frac{|F|\frac{(f+h)}{t}}{4(4t+|F|)} \geq \frac{7t|F|}{4(4t+|F|)}. \end{aligned}$$

If $|F| \leq 40t$, we can bound the fraction with $\frac{7t}{4(44t)} \geq 0.035$. If $|F| \geq 40t$, then can bound the fraction with $\frac{7|F|}{4(1.1|F|)} \geq 1.55$.

$\frac{1}{3}$-Quasi-Regular-Clique

We prove the NP-hardness of $\frac{1}{3}$-Quasi-Regular-Clique.

**Lemma 63.** $\frac{1}{3}$-Quasi-Regular-Clique *is NP-hard.*

*Proof.* We prove the NP-hardness by reducing from $\frac{2}{3}$-Clique (see [56]): given a graph $G = (V, E)$, does $G$ contain a clique of size at least $\frac{2}{3} \cdot |V|$?

We let $n = |V|$. Observe that if $G$ has fewer than $\frac{2}{3} \cdot n$ nodes of degree $\geq \frac{2}{3} \cdot n$, then we can conclude (in polytime) that the answer to the problem is no. We assume the contrary: there exist at least $\frac{2}{3} \cdot n$ nodes of degree $\geq \frac{2}{3} \cdot n$. Then, the volume of $V$ will be $\geq \frac{4}{9} \cdot n^2$. If we let $\Delta$ denote the maximum degree of $G$, we have that $\frac{2}{3} \cdot n \leq \Delta < n$. Also, an upperbound for the volume of $V$ is $\Delta \cdot n < n^2$.

We create a new graph $G' = (V', E')$ that contains $G$ as a node subgraph (that is, $V \subseteq V'$ and $E \subseteq E'$). Its node set $V'$ will also contain $n$ new nodes, for a total of $|V'| = 2n$ many nodes. Its edge set will contain all the edges in $E$, and possibly some new edges going from the nodes in $V$ to the ones in $V' - V$; these new edges will be added as follows. As long as there exists some node $v \in V$, such that $\deg_{G'}(v) < \Delta$, we choose arbitrarily one node $v'$ in $V' - V$, having degree $\deg_{G'}(v') \leq \frac{5}{9} \cdot n$, and we add the edge $\{v, v'\}$ to $E'$. Observe that such a node $v'$ will always exist: each time we add an edge we increase the total degree of $V$; then, since the volume of $V$ had value $\geq \frac{4}{9} \cdot n^2$ at the beginning and cannot get past $n^2$, we have that no more than $\frac{5}{9} \cdot n^2$ edges will need to be added. Further, since all $n$ nodes in $V' - V$ had degree 0 in the beginning, it is possible to add $\leq \frac{5}{9} \cdot n^2$ edges, with each edge having a single endpoint in $V' - V$, in such a way that the maximum degree in $V' - V$ remains bounded by $\leq \frac{5}{9} \cdot n$.

In the end, for each $v \in V$, we will have $\deg_{G'}(v) = \Delta \geq \frac{2}{3} \cdot n = \frac{1}{3} \cdot |V'|$ and, for each $v' \in V' - V$, $\deg_{G'}(v') \leq \frac{5}{9} \cdot n = \frac{5}{18} \cdot |V'|$.

We claim that $G$ had a clique of size $\geq \frac{2}{3} \cdot n$ iff $G'$ has a clique of size $\geq \frac{2}{3} \cdot n = \frac{1}{3} \cdot |V'|$.

Indeed, if $G$ had such a clique $C$, then $C \subseteq V$ will also be a clique in $G'$. On the other hand, suppose there exists some clique $C' \subseteq V'$ in $G'$ of size $|C| \geq \frac{2}{3} \cdot n$. Then, by the upperbound on the degree of the nodes in $V' - V$, $C'$ must be composed only of nodes in $V$. But then, $C'$ will also be a clique in $G$ by construction.

### 4.7.3 Proofs from Section 4.4.1

**Proof of Lemma 50**

*Proof.* We can write $D(M, \mathcal{S}) = D(M, \mathcal{S}_t) + D(M, \mathcal{S} \setminus \mathcal{S}_t)$. Consider any set $S \in \mathcal{S} \setminus \mathcal{S}_t$. Suppose $|S| \leq \varepsilon t$ (the other case is similar). We have

$$D(M^*, S) = 1 - \frac{|S \cap M^*|}{|S \cup M^*|} \geq 1 - \frac{\varepsilon t}{t} = 1 - \varepsilon \geq D(M, S) - \varepsilon.$$

Therefore,

$$D(M,\mathcal{S}) = D(M,\mathcal{S}_t) + D(M,\mathcal{S}\setminus\mathcal{S}_t) \le D(M^*,\mathcal{S}_t) + \alpha + D(M^*,\mathcal{S}\setminus\mathcal{S}_t) + \varepsilon|\mathcal{S}\setminus\mathcal{S}_t| \le D(M^*,\mathcal{S}) + \alpha + \varepsilon m.$$

**Proof of Lemma 51**

*Proof.* Consider the total similarity of $M$,

$$J(M,\mathcal{S}) = \sum_{x\in M}\sum_{\substack{S_j\ni x\\ S_j\in\mathcal{S}}}\frac{1}{|S_j\cup M|} = \sum_{x\in M\cap T}\sum_{\substack{S_j\ni x\\ S_j\in\mathcal{S}}}\frac{1}{|S_j\cup M|} + \sum_{x\in M\setminus T}\sum_{\substack{S_j\ni x\\ S_j\in\mathcal{S}}}\frac{1}{|S_j\cup M|}.$$

The first sum can be bounded as

$$\sum_{x\in M\cap T}\sum_{\substack{S_j\ni x\\ S_j\in\mathcal{S}}}\frac{1}{|S_j\cup M|} \le \sum_{x\in M\cap T}\sum_{\substack{S_j\ni x\\ S_j\in\mathcal{S}}}\frac{1}{|M|} \le \sum_{x\in M\cap T}\frac{k}{|M|} \le k.$$

Let us now turn our attention to the total similarity of $M\setminus T$,

$$J(M\setminus T,\mathcal{S}) = \sum_{x\in M\setminus T}\sum_{\substack{S_j\ni x\\ S_j\in\mathcal{S}}}\frac{1}{|S_j\cup(M\setminus T)|} \ge \sum_{x\in M\setminus T}\sum_{\substack{S_j\ni x\\ S_j\in\mathcal{S}}}\frac{1}{|S_j\cup M|}.$$

Then,

$$J(M,\mathcal{S}) \le k + J(M\setminus T,\mathcal{S}) \quad\Longrightarrow\quad D(M\setminus T,\mathcal{S}) \le D(M,\mathcal{S}) + k.$$

**Proof of Lemma 52**

*Proof.* By the Chernoff bound, if $X$ is the sum of $k$ independent 0/1-random variables, each with expectation $q$, it holds that

$$\Pr\left[|X-kq| > \varepsilon kq\right] \le 2\exp\left(-\frac{\varepsilon^2}{3}kq\right),$$

and if $u > 2ekq$, then

$$\Pr\left[X > u\right] \le 2^{-u}.$$

In our case $|C\cap P|$ is the sum of $|C|$ independent 0/1-random variables each with expectation $p$. When $|C| \ge \varepsilon^4|U_t|$, we have

$$\begin{aligned}\Pr[||C\cap P| - p|C|| > \varepsilon p|C|] &\le 2\exp\left(-\frac{\varepsilon^2}{3}p|C|\right)\\ &\le 2\exp\left(-\frac{\varepsilon^2}{3}\cdot\left(\varepsilon^4|U_t|\right)\cdot\left(9\varepsilon^{-6}|U_t|^{-1}\ln(nm)\right)\right)\\ &= 2\exp\left(-3\ln(nm)\right) \le O\left(\frac{1}{nm}\right)^3.\end{aligned}$$

If $|C| < \varepsilon^4|U_t|$, we have $2ep|C| < 6\varepsilon^4 p|U_t| = u$, so the second bound from above can be applied. Observe that $u = 54\varepsilon^{-2}\ln(nm) \ge 3\lg(nm)$ and thus

$$\Pr[|C\cap P| > 6\varepsilon^4 p|U_t|] \le \left(\frac{1}{nm}\right)^3.$$

**Proof of Lemma 53**

*Proof.* With high probability, the conditions in Lemma 52, hold for every intersection $C = M_t^* \cap S$, with $S \in \mathcal{S}$. Let $\{\mathbf{y}_i\}_{i=1}^{|U_t|}$ be the $\mathcal{L}$-assignment of $M_t^*$. Fix a set $S \in \mathcal{S}_t(M_P^*)$. The first constraint of $\mathcal{L}$,

$$\sum_{x_i \in S \cap U_t} \mathbf{y}_i \leq \frac{|S \cap M_P^*|}{(1-\varepsilon)p}$$

is equivalent to

$$|M_t^* \cap S| \leq \frac{|M_P^* \cap S|}{(1-\varepsilon)p} = \frac{|(M_t^* \cap S) \cap P|}{(1-\varepsilon)p}.$$

In other words, it states that the intersection $M_t^* \cap S$ is well preserved under the sample $P$. This is exactly the condition guaranteed by Lemma 52, provided that $|M_t^* \cap S| \geq \varepsilon^4 |U_t|$. Assume to the contrary that $|M_t^* \cap S| < \varepsilon^4 |U_t|$. Then, the size of $|M_t^* \cap S \cap P| \leq 6\varepsilon^4 |U_T| p = \frac{54 \ln(nm)}{\varepsilon^2}$; therefore $S \notin \mathcal{S}_t(M_P)$.

The second constraint is similar. Finally, the second to last constraints says that $|M_t^*| \leq \frac{|M_t^* \cap P|}{(1-\varepsilon)p}$. We first derive a bound on $|U_t|$. Since each set in $\mathcal{S}_t$ has at most $t/\varepsilon$ elements, the multiset of elements present in some set $S \in \mathcal{S}_t$ is at most $|\mathcal{S}_t| t/\varepsilon$. Furthermore, since the elements in $U_t$ have degree at least $\varepsilon |\mathcal{S}_t|$, the total number of such elements can be at most $\frac{|\mathcal{S}_t| t/\varepsilon}{\varepsilon |\mathcal{S}_t|}$. Therefore $|U_t| \leq t\varepsilon^{-2}$.

We know by assumption that $|M_t^*| \geq \varepsilon^2 t \geq \varepsilon^4 |U_t|$. Therefore $|M_t^*|$ satisfies the conditions of Lemma 52, and $|M_t^*| \leq \frac{|M_t^* \cap P|}{(1-\varepsilon)p}$, as we needed to show.

### 4.7.4 Proofs from Section 4.4.2

**Proof of Lemma 56**

*Proof.* Let us define $\tilde{\varepsilon}_1 = 3 \cdot \varepsilon_1$, and take an arbitrary set $M \subseteq X$ having size $> (1 + 3\varepsilon_1) = (1 + \tilde{\varepsilon}_1)\mu$ (resp., $< (1 - 3\varepsilon_1) = (1 - \tilde{\varepsilon}_1)\mu \leq (1 + \tilde{\varepsilon}_1)^{-1}\mu$).

Let $\mathcal{S}' \subseteq \mathcal{S}$ be such that $S_i \in \mathcal{S}'$ iff $|S_i| \leq \mu$ (resp., $|S_i| \geq \mu$). Note that $|\mathcal{S}'| \geq \frac{m}{2}$.

Note that, for each $S_i \in \mathcal{S}'$, it holds that $\frac{|S_i \cap M|}{|S_i \cup M|} < \frac{\mu}{(1+\tilde{\varepsilon}_1)\mu} = \frac{1}{1+\tilde{\varepsilon}_1}$ because $|S_i \cap M| \leq |S_i| \leq \mu$ and $|S_i \cup M| \geq |M| > (1 + \tilde{\varepsilon}_1)\mu$ (resp., $|S_i \cap M| \leq |M| < (1 + \tilde{\varepsilon}_1)^{-1}\mu$ and $|S_i \cup M| \geq |S_i| \geq \mu$).

The total Jaccard Similarity of $M$ is then given by:

$$\begin{aligned}
J(M, \mathcal{S}) &= \sum_{S_i \in \mathcal{S}} \frac{|S_i \cap M|}{|S_i \cup M|} \\
&= \sum_{S_i \in \mathcal{S}'} \frac{|S_i \cap M|}{|S_i \cup M|} + \sum_{S_i \in \mathcal{S} - \mathcal{S}'} \frac{|S_i \cap M|}{|S_i \cup M|} \\
&< \sum_{S_i \in \mathcal{S}'} \frac{1}{1+\tilde{\varepsilon}_1} + \sum_{S_i \in \mathcal{S} - \mathcal{S}'} 1 \\
&= |\mathcal{S}'| \cdot \frac{1}{1+\tilde{\varepsilon}_1} + (m - |\mathcal{S}'|) \\
&\leq \frac{m}{2} \cdot \frac{1}{1+\tilde{\varepsilon}_1} + \frac{m}{2} \\
&= m \cdot \left(1 - \frac{\tilde{\varepsilon}_1}{2 + 2\tilde{\varepsilon}_1}\right).
\end{aligned}$$

Thus, the total distance is $> \frac{\tilde{\varepsilon}_1}{2+2\tilde{\varepsilon}_1} \cdot m \geq \frac{1}{3}\tilde{\varepsilon}_1$, for $\tilde{\varepsilon}_1 \leq \frac{1}{2}$ — that is, $\varepsilon_1 \leq \frac{1}{6}$.

**Proof of Lemma 57**

We need one more technical lemma before proving Lemma 57. We begin by showing that almost all of the sets have their size in $(1 \pm O(\sqrt{\varepsilon}))\mu$. Intuitively, if there are many sets whose size is far from the size of the near optimal median (as bounded in Lemma 56), then each of those sets contributes at least an $O(\sqrt{\varepsilon})$ to the overall distance, leading to a contradiction.

**Lemma 64.** *Fix $0 < \varepsilon_3 < \frac{1}{6}$. Let $\mathcal{S}' \subseteq \mathcal{S}$ be the class of sets $S_i$ of sizes $(1-4\sqrt{\varepsilon_3})\mu \leq |S_i| \leq (1+4\sqrt{\varepsilon_3})\mu$. If $\mathcal{S}$ is an $\varepsilon_3$-good instance, then $|\mathcal{S}'| \geq \left(1-\sqrt{\varepsilon_3}\right) m$.*

*Proof.* Suppose $|\mathcal{S} - \mathcal{S}'| > \sqrt{\varepsilon_3} \cdot m$; that is, suppose that more than $\sqrt{\varepsilon_3} \cdot m$ sets have size $\leq \left(1-4\sqrt{\varepsilon_3}\right) \cdot \mu$ or $\geq \left(1+4\sqrt{\varepsilon_3}\right) \cdot \mu$. By $\varepsilon_3 \leq \frac{1}{6}$ and lemma 56, the best median $M^*$ will have size $(1-3\varepsilon_3) \cdot \mu \leq |M^*| \leq (1+3\varepsilon_3) \cdot \mu$.

If $|S_i| \leq \left(1-\sqrt{\varepsilon_3}\right) \cdot \mu$, then

$$J(M^*, S_i) \leq \frac{|M^* \cap S_i|}{|M^* \cup S_i|} \leq \frac{|S_i|}{|M^*|} \leq \frac{1-4\sqrt{\varepsilon_3}}{1-3\varepsilon_3} = 1 - \frac{4\sqrt{\varepsilon_3} - 3\varepsilon_3}{1-3\varepsilon_3} \leq 1 - 4\sqrt{\varepsilon_3} + 3\varepsilon_3.$$

On the other hand, if $|S_i| \geq \left(1+4\sqrt{\varepsilon_3}\right) \mu$, we have

$$J(M^*, S_i) \leq \frac{|M^* \cap S_i|}{|M^* \cup S_i|} \leq \frac{|M^*|}{|S_i|} \leq \frac{1+3\varepsilon_3}{1+4\sqrt{\varepsilon_3}} = 1 - \frac{4\sqrt{\varepsilon_3} - 3\varepsilon_3}{1+4\sqrt{\varepsilon_3}},$$

In both cases, $J(M^*, S_i) \leq 1 - \sqrt{\varepsilon_3}$, by $\varepsilon_3 \leq \frac{1}{6}$.

The total Jaccard similarity will be,

$$\begin{aligned}
J(M^*, \mathcal{S}) &= \sum_{S_i \in \mathcal{S}} J(M^*, S_i) \\
&= \sum_{S_i \in \mathcal{S}'} J(M^*, S_i) + \sum_{S_i \in \mathcal{S} - \mathcal{S}'} J(M^*, S_i) \\
&\leq \sum_{S_i \in \mathcal{S}'} 1 + \sum_{S_i \in \mathcal{S} - \mathcal{S}'} (1 - \sqrt{\varepsilon_3}) \\
&= |\mathcal{S}'| + |\mathcal{S} - \mathcal{S}'| \, (1 - \sqrt{\varepsilon_3}) \\
&< (1 - \sqrt{\varepsilon_3}) \, m + \sqrt{\varepsilon_3} m \, (1 - \sqrt{\varepsilon_3}) \\
&= m - \varepsilon_3 m.
\end{aligned}$$

Thus, the total distance will be $> \varepsilon_3 \cdot m$, a contradiction.

We are now ready to prove Lemma 57.

*Proof.* Let $X' \subseteq X$ be the set of high degree elements. Let $M^*$ be the optimal median. By lemma 56, $(1-3\varepsilon_2)\mu \leq |M^*| \leq (1+3\varepsilon_2)\mu$. Note that the total Jaccard similarity $J(M^*, \mathcal{S})$ can be written as

$$\begin{aligned}
J(M^*, \mathcal{S}) &= \sum_{S_i \in \mathcal{S}} \frac{|S_i \cap M^*|}{|S_i \cup M^*|} \\
&= \sum_{S_i \in \mathcal{S}} \sum_{x \in S_i \cap M^*} \frac{1}{|S_i \cup M^*|} \\
&= \sum_{x \in M^*} \sum_{S_i \ni x} \frac{1}{|S_i \cup M^*|} \\
&\leq \sum_{x \in M^*} \sum_{S_i \ni x} \frac{1}{|M^*|} \\
&\leq \frac{1}{|M^*|} \sum_{x \in X' \cap M^*} \left( \sum_{S_i \ni x} 1 \right) + \frac{1}{|M^*|} \sum_{x \in (X - X') \cap M^*} \left( \sum_{S_i \ni x} 1 \right) \\
&\leq \frac{1}{|M^*|} \sum_{x \in X' \cap M^*} m + \frac{1}{|M^*|} \sum_{x \in (X - X') \cap M^*} \left( \left(1 - \sqrt{2\varepsilon_2}\right) \cdot m \right)
\end{aligned}$$

Now, suppose by contradiction that $|X'| < \left(1-\sqrt{2\varepsilon_2}\right) \cdot \mu = T$. The overall number of terms in the two sums of the previous expression is $\leq |M^*|$; also the higher the number of terms of the first sum, the

higher the expression's value. Thus,

$$\begin{aligned}
J(M^*,\mathcal{S}) &< \frac{1}{|M^*|}\cdot T\cdot m + \frac{1}{|M^*|}\cdot(|M^*|-T)\cdot\left(\left(1-\sqrt{2\varepsilon_2}\right)\cdot m\right)\\
&= \left(1-\sqrt{2\varepsilon_2}\right)\cdot m + \frac{T}{|M^*|}\cdot\sqrt{2\varepsilon_2}\cdot m\\
&= \left(1-\sqrt{2\varepsilon_2}\right)\cdot m + \frac{T}{(1-3\varepsilon_2)\cdot\mu}\cdot\sqrt{2\varepsilon_2}\cdot m\\
&= \left(1-\sqrt{2\varepsilon_2}\right)\cdot m + \frac{\left(1-\sqrt{2\varepsilon_2}\right)\cdot\mu}{(1-3\varepsilon_2)\cdot\mu}\cdot\sqrt{2\varepsilon_2}\cdot m\\
&= \left(1-\sqrt{2\varepsilon_2}\right)\cdot m + \left(1-\frac{\sqrt{2\varepsilon_2}-3\varepsilon_2}{1-3\varepsilon_2}\right)\cdot\sqrt{2\varepsilon_2}\cdot m\\
&= m - \frac{\sqrt{2\varepsilon_2}-3\varepsilon_2}{1-3\varepsilon_2}\cdot\sqrt{2\varepsilon_2}\cdot m\\
&= \left(1-\frac{2\varepsilon_2-3\sqrt{2\varepsilon_2^3}}{1-3\varepsilon_2}\right)\cdot m
\end{aligned}$$

This implies $D(M^*,\mathcal{S}) > \frac{2-3\sqrt{2\varepsilon_2}}{1-3\varepsilon_2}\cdot\varepsilon_2\cdot m \geq \varepsilon_2\cdot m$ (where the last inequality is implied by $\varepsilon_2 \leq \frac{2-\sqrt{3}}{3}$). This is a contradiction; thus $|X'| \geq \left(1-\sqrt{2\varepsilon_2}\right)\cdot\mu$.

**Proof of Lemma 58**

*Proof.* Fix an arbitrary $x^* \in X^* - M$. We will show that $D(M\cup\{x^*\},\mathcal{S}) < D(M,\mathcal{S})$, so that the main statement will be proved.

Note that, for any $Y \subseteq X$, it holds that $J(Y,\mathcal{S}) = \sum_{y\in Y}\sum_{S_i\ni y}\frac{1}{|S_i\cup Y|}$.

By lemma 64, there exist at least $(1-\sqrt{\varepsilon_4})m$ sets of size $\leq \left(1+4\sqrt{\varepsilon_4}\right)\mu$. The element $x^*$ has degree $\geq \left(1-\sqrt{\varepsilon_4}\right)m$ so it will be part of at least $(1-2\sqrt{\varepsilon_4})m$ sets of size $\leq \left(1+4\sqrt{\varepsilon_4}\right)\mu$. Let $\mathcal{S}'_{x^*}$ be the class of these sets.

By lemma 56, the set $M$ will have size $(1-3\varepsilon_4)\cdot\mu \leq |M| \leq (1+3\varepsilon_4)\cdot\mu$. So, for $S_i \in \mathcal{S}'_{x^*}$ we can lower bound the term $\frac{1}{|S_i\cup M|}$ (which will be in the following chain of inequalities) with

$$\frac{1}{|S_i\cup M|} \geq \frac{1}{|S_i|+|M|} \geq \frac{1}{\left(1+4\sqrt{\varepsilon_4}\right)\mu + (1+3\varepsilon_4)\cdot\mu} \geq \frac{1}{\left(2+7\sqrt{\varepsilon_4}\right)\mu}.$$

Also, we will use the inequality $|M| \geq (1-3\varepsilon_4)\mu$ in the following chain.

So, $J(M \cup \{x^*\}, \mathcal{S}) - J(M, \mathcal{S})$ equals

$$\begin{aligned}
& \sum_{x \in M \cup \{x^*\}} \sum_{\substack{S_i \\ x \in S_i \in \mathcal{S}}} \frac{1}{|S_i \cup M \cup \{x^*\}|} - \sum_{x \in M} \sum_{\substack{S_i \\ x \in S_i \in \mathcal{S}}} \frac{1}{|S_i \cup M|} \\
= & \sum_{\substack{S_i \\ x^* \in S_i \in \mathcal{S}}} \frac{1}{|S_i \cup M \cup \{x^*\}|} + \sum_{x \in M} \sum_{\substack{S_i \\ x \in S_i \in \mathcal{S}}} \left( \frac{1}{|S_i \cup M \cup \{x^*\}|} - \frac{1}{|S_i \cup M|} \right) \\
= & \sum_{\substack{S_i \\ x^* \in S_i \in \mathcal{S}}} \frac{1}{|S_i \cup M|} - \sum_{x \in M} \sum_{\substack{S_i \\ x \in S_i \in \mathcal{S} \\ x^* \notin S_i}} \frac{1}{|S_i \cup M| \left(|S_i \cup M| + 1\right)} \\
> & \sum_{S_i \in \mathcal{S}'_{x^*}} \frac{1}{|S_i \cup M|} - \sum_{x \in M} \sum_{\substack{S_i \\ x \in S_i \in \mathcal{S} \\ x^* \notin S_i}} \frac{1}{|M|^2} \\
\geq & \sum_{S_i \in \mathcal{S}'_{x^*}} \frac{1}{\left(2 + 7\sqrt{\varepsilon_4}\right) \cdot \mu} - \sum_{x \in M} \frac{\sqrt{\varepsilon_4} \cdot m}{|M|^2} \\
\geq & \frac{\left(1 - 2\sqrt{\varepsilon_4}\right) \cdot m}{\left(2 + 7\sqrt{\varepsilon_4}\right) \cdot \mu} - \frac{\sqrt{\varepsilon_4} \cdot m}{|M|} \\
\geq & \frac{\left(1 - 2\sqrt{\varepsilon_4}\right) \cdot m}{\left(2 + 7\sqrt{\varepsilon_4}\right) \cdot \mu} - \frac{\sqrt{\varepsilon_4} \cdot m}{(1 - 3\varepsilon_4) \cdot \mu} \\
= & \frac{m}{\mu} \left( \frac{1 - 2\sqrt{\varepsilon_4}}{2 + 7\sqrt{\varepsilon_4}} - \frac{\sqrt{\varepsilon_4}}{1 - 3\varepsilon_4} \right) = \frac{m}{\mu} \cdot \frac{1 - 4\varepsilon_4^{1/2} - 10\varepsilon_4 + 6\varepsilon_4^{3/2}}{2 + 7\varepsilon_4^{1/2} - 6\varepsilon_4 - 21\varepsilon_4^{3/2}}.
\end{aligned}$$

Note that the latter is positive for $\varepsilon_4 \leq c$, for some positive constant $c$ (in particular for some $c \geq 0.0319\ldots$). Thus if $\varepsilon_4 \leq c$ then $J(M \cup \{x^*\}, \mathcal{S}) - J(M, \mathcal{S}) > 0$ — equivalently, $D(M \cup \{x^*\}, \mathcal{S}) < D(M, \mathcal{S})$.

**Proof of Theorem 59**

*Proof.* For any solution $M \cup N$, we have:

$$D(\mathcal{S}, M \cup N) = \sum_{S_j \in \mathcal{S}} \left( 1 - \frac{|S_j \cap (M \cup N)|}{|S_j \cup (M \cup N)|} \right) = \sum_{S_j \in \mathcal{S}} \frac{|S_j \cup (M \cup N)| - |S_j \cap (M \cup N)|}{|S_j \cup (M \cup N)|}$$

If we restrict the size of $N$, with $|N| < O(\sqrt{\varepsilon}\mu)$ elements, then for each set $S_j$,

$$|S_j \cup M| \leq |S_j \cup (M \cup N)| \leq |S_j \cup M| + O((\sqrt{\varepsilon})\mu) \leq |S_j \cup M|(1 + O(\sqrt{\varepsilon})),$$

where the last inequality follows from the lower bound on the size of $M$. For any set T, let $D_T$ as the distance with each denominator fixed to be $|S_j \cup T|$.

$$D_T(\mathcal{S}, A) = \sum_{S_j \in \mathcal{S}} \frac{|S_j \cup A| - |S_j \cap A|}{|S_j \cup T|} = \sum_{S_j \in \mathcal{S}} \sum_{x_i \in S_j \triangle A} \frac{1}{|S_j \cup T|},$$

where for two sets $U$ and $V$, $U \triangle V$ denotes their symmetric difference. Then we have

$$D_M(\mathcal{S}, M \cup N) \geq D(\mathcal{S}, M \cup N) \geq \frac{1}{1 + O(\sqrt{\varepsilon})} \cdot D_M(\mathcal{S}, M \cup N). \tag{4.2}$$

Let $N$ be such that $N \cap M = \varnothing$. It's easy check that $D_M$ can be rewritten as:

$$\begin{aligned}
& D_M(\mathcal{S}, M \cup N) = \sum_{S_j \in \mathcal{S}} \sum_{x_i \in S_j \triangle (M \cup N)} \frac{1}{|S_j \cup M|} = \\
& = \left( \sum_{x_i \in M} \sum_{S_j \not\ni x_i} \frac{1}{|S_j \cup M|} + \sum_{x_i \notin M} \sum_{S_j \ni x_i} \frac{1}{|S_j \cup M|} \right) - \sum_{x_i \in N} \left( \sum_{S_j \ni x_i} \frac{1}{|S_j \cup M|} - \sum_{S_j \not\ni x_i} \frac{1}{|S_j \cup M|} \right)
\end{aligned}$$

Let $N^*$ be the set that minimizes $D_M(\mathcal{S}, M \cup N^*)$, under the constraints $M \cap N^* = \varnothing$ and $|N^*| < O(\sqrt{\varepsilon} \cdot \mu)$. If we define the weight of an element $x_i \notin M$ to be $\sum_{S_j \ni x_i} \frac{1}{|S_j \cup M|} - \sum_{S_j \not\ni x_i} \frac{1}{|S_j \cup M|}$, then $N^*$ can be found by greedily selecting elements in order of decreasing weight, stopping when either (a) the size of $N^*$ has reached its limit, or (b) the weight of the element in consideration is non-positive.

Let $M^* = M \cup M'$, $M' \cap M = \varnothing$, be the optimal solution. Recall that $|M'| \leq O(\sqrt{\varepsilon}) \cdot |M|$. Then:

$$\begin{aligned} D(\mathcal{S}, M^*) &\geq \frac{1}{1 + O(\sqrt{\varepsilon})} \cdot D_M(\mathcal{S}, M^*) \\ &\geq \frac{1}{1 + O(\sqrt{\varepsilon})} \cdot D_M(\mathcal{S}, M \cup N^*) \\ &\geq \frac{1}{1 + O(\sqrt{\varepsilon})} \cdot D(\mathcal{S}, M \cup N^*) \end{aligned}$$

Where the first and last inequality follow from Equation 4.2 and the second from the optimality of $N^*$.

### 4.7.5 Algorithms for Generalized Jaccard metric

### 4.7.6 Very Good Medians

The algorithm in this section returns a $1 + O(\varepsilon)$ approximate median if the optimal median has total distance $\leq \varepsilon$.

**Lemma 65.** *There exists a polynomial time algorithm producing a median $M$ such that $D(M, \mathcal{V}) \leq (1 + \frac{\varepsilon}{1-\varepsilon}) \cdot D(M^*, \mathcal{V})$, if the optimal median $M^*$ is such that $D(M^*, \mathcal{V}) \leq \varepsilon$.*

*Proof.* If two generic $A, B$ vectors have Jaccard distance $\leq \delta$, it must be that

$$\sum_i \min\left(A(i), B(i)\right) \geq (1 - \delta) \cdot \sum_i \max\left(A(i), B(i)\right).$$

so that

$$\sum_i A(i) \geq (1 - \delta) \cdot \sum_i \max(A(i), B(i))$$

Now, take two vectors $A', B'$ and suppose their distance $D(A', B') \leq \varepsilon$,

$$D(A', B') = \frac{\sum_i |A'(i) - B'(i)|}{\sum_i \max(A'(i), B'(i))} \leq \varepsilon,$$

using $\delta := \varepsilon$, $A := A'$ and $B := B'$ in the previous equation we obtain $\frac{1}{1-\varepsilon} \cdot \sum_i A'(i) \geq \sum_i \max(A'(i), B'(i))$, and

$$\frac{\sum_i |A'(i) - B'(i)|}{\frac{1}{1-\varepsilon} \cdot \sum_i A'(i)} \leq \frac{\sum_i |A'(i) - B'(i)|}{\sum_i \max(A'(i), B'(i))} \leq \varepsilon,$$

and

$$\sum_i |A'(i) - B'(i)| \leq \frac{\varepsilon}{1 - \varepsilon} \cdot \sum_i A'(i).$$

Further, observe that if we have two vectors $A'', B''$ such that $\sum_i |A''(i) - B''(i)| \leq \frac{\varepsilon}{1-\varepsilon} \cdot \sum_i A''(i)$ then,

$$D(A'', B'') = \frac{\sum_i |A''(i) - B''(i)|}{\sum_i \max(A''(i), B''(i))} \leq \frac{\sum_i |A''(i) - B''(i)|}{\sum_i A''(i)} \leq \frac{\frac{\varepsilon}{1-\varepsilon} \cdot \sum_i A''(i)}{\sum_i A''(i)} = \frac{\varepsilon}{1 - \varepsilon}.$$

Now consider the following linear program:

$$\begin{cases} \mathbf{m_i} \geq 0 & \forall \text{ coordinate } i \\ \mathbf{t_i^j} \geq |\mathbf{m_i} - V_j(i)| & \forall V_j \in \mathcal{V}, \forall \text{ coordinate } i \\ \sum_i \mathbf{t_i^j} \leq \frac{\varepsilon}{1-\varepsilon} \cdot \sum_i V_j(i) & \forall V_j \in \mathcal{V} \\ \\ \mathbf{min} \sum_j \frac{1}{\sum_i V_j(i)} \cdot \sum_i \mathbf{t_i^j}. \end{cases}$$

(Observe how the inequality $\mathbf{t_i^j} \geq |\mathbf{m_i} - V_j(i)|$ can be replaced by the two inequalities $\mathbf{t_i^j} \geq \mathbf{m_i} - V_j(i)$ and $\mathbf{t_i^j} \geq V_j(i) - \mathbf{m_i}$.)

We claim that if an optimal median $M^*$ for $\mathcal{V}$ has total distance $D(M^*, \mathcal{V}) \leq \varepsilon$ then the solution linear program is feasible, and each of its optimal solutions are $(1 + \frac{\varepsilon}{1-\varepsilon})$ approximations to $D(M^*, \mathcal{V})$. That is, if $\mathbf{M}^* = (\mathbf{m_1^*}, \mathbf{m_2^*}, \ldots, \mathbf{m_n^*})$ is an optimal solution to the linear program then $D(\mathbf{M}^*, \mathcal{V}) \leq (1 + \frac{\varepsilon}{1-\varepsilon}) \cdot D(M^*, \mathcal{V})$.

Indeed, to observe that the linear program is feasible, take $\mathbf{m_i} = M^*(i)$ for each $i$, and $\mathbf{t_i^j} = |M^*(i) - V_j(i)|$ for each $i, j$. Since $D(M^*, \mathcal{V}) \leq \varepsilon$ it must be that, for each $V_j \in \mathcal{V}$, also $D(M^*, V_j) \leq \varepsilon$. Then, setting $A' := V_j(i)$ and $B' := M(i)$ in the previous equation, we obtain

$$\sum_i \mathbf{t_i^j} = \sum_i |M^*(i) - V_j(i)| \leq \frac{\varepsilon}{1-\varepsilon} \cdot \sum_i V_j(i),$$

so all the constraints are satisfied. The value of the objective function will then be

$$f^* = \sum_j \frac{\sum_i |\mathbf{m_i} - V_j(i)|}{\sum_i V_j(i)} = \sum_j \frac{\sum_i |M^*(i) - V_j(i)|}{\sum_i V_j(i)}.$$

For each $j$ we apply the previous inequaility with $A := V_j$, $B := \mathbf{M}$ and $\delta = \varepsilon$, obtaining $\sum_i V_j(i) \geq (1-\varepsilon) \cdot \sum_i \max(V_j(i), \mathbf{m_i})$. Then,

$$f^* \leq \frac{1}{1-\varepsilon} \cdot \sum_j \frac{\sum_i |M^*(i) - V_j(i)|}{\sum_i \max(V_j(i), \mathbf{m_i})} = \frac{1}{1-\varepsilon} \cdot D(M^*, \mathcal{V}).$$

Now take any optimal solution $\mathbf{M}^* = (\mathbf{m_1^*}, \mathbf{m_2^*}, \ldots, \mathbf{m_n^*})$ to the linear program. Consider the function that the linear program is minimizing,

$$f = \sum_j \frac{\sum_i \mathbf{t_i^j}}{\sum_i V_j(i)},$$

since $\mathbf{M}^*$ is optimal we will have $\mathbf{t_i^j} = |\mathbf{m_i^*} - V_j(i)|$, for each $i, j$, and

$$f = \sum_j \frac{\sum_i |\mathbf{m_i^*} - V_j(i)|}{\sum_i V_j(i)}.$$

Observe that if we were to use the vector $\mathbf{M}^*$ as a median, we would have total distance

$$D(\mathbf{M}^*, \mathcal{V}) = \sum_j \frac{\sum_i |\mathbf{m_i^*} - V_j(i)|}{\sum_i \max(\mathbf{m_i^*}, V_j(i))} \leq \sum_j \frac{\sum_i |\mathbf{m_i^*} - V_j(i)|}{\sum_i V_j(i)} = f.$$

Further, since $f$ is optimal, and $f^*$ is feasible, we will have $f \leq f^*$, and

$$D(\mathbf{M}^*, \mathcal{V}) \leq f \leq f^* \leq \frac{1}{1-\varepsilon} \cdot D(M^*, \mathcal{V}),$$

so $\mathbf{M}^*$ is an $\frac{1}{1-\varepsilon}$-approximate median.

### 4.7.7 Medians that are not Very Good

#### Polynomial Spread Instances

Given an input set $\mathcal{V}$, not all null, let $\alpha$ be their minimum non-zero coordinate value,

$$\alpha = \alpha_{\mathcal{V}} = \min_{\substack{V \in \mathcal{V} \\ 1 \leq i \leq n \\ V(i) > 0}} V(i),$$

and let $\beta$ be their maximum coordinate value,

$$\beta = \beta_{\mathcal{V}} = \max_{\substack{V \in \mathcal{V} \\ 1 \leq i \leq n}} V(i).$$

Observe that if all the input vectors are all-zero vectors, then the input is a set instance, and then the optimal median is trivally the all-zero vector. Otherwise both $\alpha$ and $\beta$ are well-defined, and we define the "spread" of $\mathcal{V}$ as $\sigma = \frac{\beta}{\alpha}$.

Suppose that the spread is polynomial, $\sigma \leq (n \cdot m)^{O(1)}$.

Let us scale the vectors by $\alpha^{-1}$ — obtaining the multi-set of vectors $\mathcal{V}_\alpha = \left\{\alpha^{-1} \cdot V \mid V \in \mathcal{V}\right\}$. Then, the minimum non-zero coordinate in $\mathcal{V}_\alpha$ will be 1, while the maximum will be $\sigma$. Let $k = \left\lceil \xi^{-1} \right\rceil$. Observe that $\xi \leq k^{-1}$. Given a vector $V$ on $n$ coordinates, having each coordinate value $\leq \sigma$, we define its *expansion* $e_{k,\sigma}(V) = e(V)$ as a 0/1 vector on $n \cdot k \cdot \lceil \sigma \rceil$ coordinates, as follows:

$$e(V) = (\underbrace{1,1,\ldots,1}_{\substack{t_1 = \lceil k \cdot V(1) \rceil \\ \text{times}}}, \underbrace{0,0,\ldots,0}_{\substack{\lceil k \cdot \sigma \rceil - t_1 \\ \text{times}}}, \underbrace{1,1,\ldots,1}_{\substack{t_2 = \lceil k \cdot V(2) \rceil \\ \text{times}}}, \underbrace{0,0,\ldots,0}_{\substack{\lceil k \cdot \sigma \rceil - t_2 \\ \text{times}}}, \ldots, \underbrace{1,1,\ldots,1}_{\substack{t_n = \lceil k \cdot V(n) \rceil \\ \text{times}}}, \underbrace{0,0,\ldots,0}_{\substack{\lceil k \cdot \sigma \rceil - t_n \\ \text{times}}}).$$

We then use the PTAS for binary instances to obtain a $(1+\varepsilon)$ approximation of the following binary instance:

$$\mathcal{V}_S = \{e_{k,\sigma}(V) \mid V \in \mathcal{V}_\alpha\} = \{e_{k,\sigma}(\alpha^{-1} \cdot V) \mid V \in \mathcal{V}\}.$$

We show that distances are well-preserved by this mapping.

**Lemma 66.** *Let $V, W$ be any two non-negative real vectors, having minimum coordinate value $\geq \alpha$ and maximum coordinate value $\leq \beta$. Let $\xi > 0$ be sufficiently small. Let $\sigma = \frac{\beta}{\alpha}$, and $k = \left\lceil \xi^{-1} \right\rceil$. Then,*

$$D(V,W) - \xi \leq D(e_{k,\sigma}(\alpha^{-1} \cdot V), e_{k,\sigma}(\alpha^{-1} \cdot W)) \leq D(V,W) + \xi.$$

*Proof.* If $V = W$, the claim is trivial as they will both be mapped to the same vector. Otherwise, $D(V,W) > 0$, and

$$\begin{aligned} D(V,W) &= 1 - \frac{\sum_{i=1}^{n} \min\{V(i), W(i)\}}{\sum_{i=1}^{n} \max\{V(i), W(i)\}} \\ &= 1 - \frac{\sum_{i=1}^{n} \min\left\{\alpha^{-1} \cdot V(i), \alpha^{-1} \cdot W(i)\right\}}{\sum_{i=1}^{n} \max\{\alpha^{-1} \cdot V(i), \alpha^{-1} \cdot W(i)\}} \\ &= D(\alpha^{-1} \cdot V, \alpha^{-1} \cdot W). \end{aligned}$$

Now, let $V' = e_{k,\sigma}(\alpha^{-1} \cdot V)$. For any $i = 1, \ldots, n$, consider

$$V'_i = \sum_{j=(i-1)\cdot\lceil k\cdot\sigma\rceil+1}^{i\cdot\lceil k\cdot\sigma\rceil} V'(j).$$

Then $\frac{1}{k} \cdot V'_i = \frac{1}{k} \cdot \left\lceil k \cdot \alpha^{-1} \cdot V(i) \right\rceil \leq \alpha^{-1} \cdot V(i) + k^{-1} \leq \alpha^{-1} \cdot V(i) + \xi$, and $\frac{1}{k} \cdot V'_i \geq \alpha^{-1} \cdot V(i)$. As $\alpha \leq V(i)$ by definition, we have that $\frac{\xi}{\alpha^{-1} \cdot V(i)} \leq \xi$. Thus,

$$\alpha^{-1} \cdot V(i) \leq V'_i \leq (1+\xi) \cdot \alpha^{-1} \cdot V(i).$$

Analogously, if $W' = e_{k,\sigma}(\alpha^{-1} \cdot W)$, we have

$$\alpha^{-1} \cdot W(i) \leq W'_i \leq (1+\xi) \cdot \alpha^{-1} \cdot W(i).$$

Then,

$$\begin{aligned} D(V',W') &= 1-\frac{\sum_{i=1}^{n}\min\{V'(i),W'(i)\}}{\sum_{i=1}^{n}\max\{V'(i),W'(i)\}} \\ &\leq 1-\frac{\sum_{i=1}^{n}\min\left\{\alpha^{-1}\cdot V(i),\alpha^{-1}\cdot W(i)\right\}}{\sum_{i=1}^{n}\max\{(1+\xi)\cdot\alpha^{-1}\cdot V(i),(1+\xi)\cdot\alpha^{-1}\cdot W(i)\}} \\ &= 1-\frac{1}{1+\xi}\cdot\frac{\sum_{i=1}^{n}\min\{V(i),W(i)\}}{\sum_{i=1}^{n}\max\{V(i),W(i)\}} \\ &= 1-\frac{1}{1+\xi}\cdot(1-D(V,W)) \\ &= \frac{1}{1+\xi}\cdot D(V,W)+\frac{\xi}{1+\xi} \\ &\leq D(V,W)+\xi \end{aligned}$$

And,

$$\begin{aligned} D(V',W') &= 1-\frac{\sum_{i=1}^{n}\min\{V'(i),W'(i)\}}{\sum_{i=1}^{n}\max\{V'(i),W'(i)\}} \\ &\geq 1-(1+\xi)\cdot\frac{\sum_{i=1}^{n}\min\{V(i),W(i)\}}{\sum_{i=1}^{n}\max\{V(i),W(i)\}} \\ &= -\xi+(1+\xi)\cdot D(V,W)\geq D(V,W)-\xi \end{aligned}$$

Our algorithm will fix $\xi=\frac{\varepsilon^2}{m}$ and map the weighted Jaccard instance into a binary instance. Then, it will compute an approximate median $M$ of the binary instance. We see $M$ as a binary vector on $n\cdot\lceil\xi^{-1}\rceil\cdot\lceil\sigma\rceil$ coordinates. The next lemma shows how one can compute in polytime a $(1+\varepsilon)$ multiplicative approximation if the optimal median has total distance $\geq\varepsilon$, and the spread is polynomial.

**Lemma 67.** *Fix $\mathcal{V}$, and take any candidate median $M$ for $\mathcal{V}_S$. A real vector $M'$ such that $D(M',\mathcal{V})\leq D(M,\mathcal{V}_S)+\xi\cdot m$ can be found in time $(n\cdot m\cdot\xi^{-1}\cdot\sigma)^{O(1)}$.*

*Proof.* Let $w_i=\sum_{j=(i-1)\cdot\lceil k\cdot\sigma\rceil+1}^{i\cdot\lceil k\cdot\sigma\rceil}M(j)$, be the number of 1's in the block of coordinates of $M$ corresponding to the $i$-th coordinate of the original real vector space. Set $w'_i=\max(w_i,k)$.

We create a 0/1 vector $A$ from $M$, by pushing all the 1's of $M$ on the left side of their respective blocks:

$$A=(\underbrace{1,1,\ldots,1}_{\substack{w'_1\\ \text{times}}},\underbrace{0,0,\ldots,0}_{\substack{\lceil k\cdot\sigma\rceil-w'_1\\ \text{times}}},\underbrace{1,1,\ldots,1}_{\substack{w'_2\\ \text{times}}},\underbrace{0,0,\ldots,0}_{\substack{\lceil k\cdot\sigma\rceil-w'_2\\ \text{times}}},\ldots,\underbrace{1,1,\ldots,1}_{\substack{w'_n\\ \text{times}}},\underbrace{0,0,\ldots,0}_{\substack{\lceil k\cdot\sigma\rceil-w'_n\\ \text{times}}}).$$

Observe that for each $V_S\in\mathcal{V}_S$, we will have $D(A,V_S)\leq D(M,V_S)$ — this is directly implied by the fact that each such $V_S$ has all its 1 coordinates on the left sides of its blocks, and that each $V_S$ has at least $k$ many 1's in each block.

Further, observe that $A$ is the $e_{k,\sigma}$ expansion of the vector $\frac{1}{k}\cdot(w'_1,w'_2,\ldots,w'_n)$. Let $M'=\alpha\cdot\frac{1}{k}\cdot(w'_1,w'_2,\ldots,w'_n)$. By lemma 66, we have that, for each real vector $V\in\mathcal{V}$,

$$D(M',V)-\xi\leq D(A,e_{k,\sigma}(\alpha^{-1}\cdot V)),$$

or, equivalently,

$$D(M',V)\leq D(A,e_{k,\sigma}(\alpha^{-1}\cdot V))+\xi.$$

Thus,

$$\begin{aligned} D(M', \mathcal{V}) &= \sum_{V \in \mathcal{V}} D(M', V) \\ &\leq \sum_{V \in \mathcal{V}} \left( D(A, e_{k,\sigma}(\alpha^{-1} \cdot V)) + \xi \right) \\ &= \sum_{V_S \in \mathcal{V}_S} \left( D(A, V_S) + \xi \right) \\ &\leq \sum_{V_S \in \mathcal{V}_S} \left( D(M, V_S) + \xi \right) \\ &= D(M, \mathcal{V}_S) + \xi \cdot m. \end{aligned}$$

**Arbitrary Spread Instances**

Let $\mathcal{V}$ be an arbitrary Jaccard median instance. To compute the median of $\mathcal{V}$, we start by guessing the largest coordinate value of one of its optimal medians (observe that by [97], and lemma 69, this coordinate value will be shared with the median by at least one input vector), and (a) we remove all the sets that would be too far to a median having such a (large) coordinate value (these would be the sets having too small coordinate values), and (b) we set to zero those coordinate values that were much smaller than our guess (by doing this, we do not distort distances by much). This way, we obtain an instance having polynomial spread - so that the polynomial spread algorithm can be applied.

More precisely, for each input coordinate value $\alpha$ (there are $\leq n \cdot m$ such values), we

- remove all sets having a coordinate value larger than $\alpha \cdot \frac{n}{\varepsilon}$, or having total weight less than $\varepsilon \cdot \alpha$ obtaining the class $\mathcal{V}_\alpha$,

$$\mathcal{V}_\alpha = \left\{ V_j \mid V_j \in \mathcal{V} \wedge \max_i V_j(i) \leq \alpha \cdot \frac{n}{\varepsilon} \wedge \sum_i V_j(i) \geq \varepsilon \cdot \alpha \right\}.$$

- For each vector $V_j \in \mathcal{V}_\alpha$, we set to zero all its coordinates having value $\leq \alpha \cdot \frac{\varepsilon^2}{n \cdot m}$, obtaining a vector $V'_j$,

$$V'_j(i) = \begin{cases} 0 & \text{if } V_j(i) \leq \frac{\alpha \cdot \varepsilon^2}{n \cdot m} \\ V_j(i) & \text{otherwise} \end{cases}$$

- Finally, we let $\mathcal{V}'_\alpha$ be the resulting instance,

$$\mathcal{V}'_\alpha = \left\{ V'_j \mid V_j \in \mathcal{V}_\alpha \right\}.$$

The spread of $\mathcal{V}'_\alpha$ will be $\leq \frac{n^2 \cdot m^2}{\varepsilon^3}$. We then apply the polynomial spread algorithm to obtain a $(1+O(\varepsilon))$-approximate median $M$ for $\mathcal{V}_\alpha$. We now show that, given the appropriate choice of $\alpha$, $M$ will be an approximately optimal median for $\mathcal{V}$.

**Lemma 68.** *Let $M^*$ be an optimal median for $\mathcal{V}$, and let $\alpha = \max_i M^*(i)$. If $M$ is a $(1 + O(\varepsilon))$-approximate median for $\mathcal{V}'_\alpha$, then*

$$D(M, \mathcal{V}) \leq (1 + O(\varepsilon)) \cdot D(M^*, \mathcal{V}) + O(\varepsilon).$$

*Proof.* We start by showing that $M$ is an approximate median for $\mathcal{V}_\alpha$. The observation that $M^*$ is at distance $\geq 1 - \varepsilon$ to each vector in $\mathcal{V} - \mathcal{V}_\alpha$ will complete the proof — as any median is at distance $\leq 1$ from each vector in $\mathcal{V} - \mathcal{V}_\alpha$.

Let $W$ be any non-negative vector on $n$ coordinates. First of all, observe that, for each $V'_j \in \mathcal{V}'_\alpha$, we have

$$\sum_i \max(V_j(i), W(i)) \geq \sum_i \max(V'_j(i), W(i)) \geq \sum_i \max(V_j(i), W(i)) - \frac{\alpha \cdot \varepsilon^2}{m},$$

and $\sum_i \max(V_j(i), W(i)) \geq \sum_i V_j(i) \geq \varepsilon \cdot \alpha$. Then,

$$\sum_i \max(V_j(i), W(i)) \geq \sum_i \max(V'_j(i), W(i)) \geq \left(1 - \frac{\varepsilon}{m}\right) \sum_i \max(V_j(i), W(i)).$$

Further,

$$\sum_i |V_j(i) - W(i)| + \frac{\alpha \cdot \varepsilon^2}{m} \geq \sum_i \left|V'_j(i) - W(i)\right| \geq \sum_i |V_j(i) - W(i)| - \frac{\alpha \cdot \varepsilon^2}{m}.$$

Let us now show that the values of a median $M'$ for $\mathcal{V}_\alpha$ and $\mathcal{V}'_\alpha$ are very close to each other; we start with an upper bound on $D(M', \mathcal{V}'_\alpha)$,

$$\begin{aligned}
D(M', \mathcal{V}'_\alpha) &= \sum_{V'_j \in \mathcal{V}'_\alpha} D(M', V'_j) \\
&= \sum_{V'_j \in \mathcal{V}'_\alpha} \frac{\sum_i \left|V'_j(i) - M'(i)\right|}{\sum_i \max(V'_j(i), M'(i))} \\
&\leq \frac{1}{1 - \frac{\varepsilon}{m}} \cdot \sum_{V'_j \in \mathcal{V}'_\alpha} \frac{\sum_i |V_j(i) - M'(i)| + \frac{\alpha \cdot \varepsilon^2}{m}}{\sum_i \max(V_j(i), M'(i))} \\
&= \frac{1}{1 - \frac{\varepsilon}{m}} \cdot \sum_{V'_j \in \mathcal{V}'_\alpha} \left( D(M', V_j) + \frac{\frac{\alpha \cdot \varepsilon^2}{m}}{\sum_i \max(V_j(i), M'(i))} \right) \\
&\leq \frac{1}{1 - \frac{\varepsilon}{m}} \cdot \sum_{V'_j \in \mathcal{V}'_\alpha} \left( D(M', V_j) + \frac{\frac{\alpha \cdot \varepsilon^2}{m}}{\varepsilon \cdot \alpha} \right) \\
&\leq \frac{1}{1 - \frac{\varepsilon}{m}} \cdot \sum_{V'_j \in \mathcal{V}'_\alpha} D(M', V_j) + \frac{\varepsilon}{1 - \frac{\varepsilon}{m}} \\
&\leq \frac{1}{1 - \varepsilon} \cdot D(M', \mathcal{V}_\alpha) + \frac{\varepsilon}{1 - \varepsilon}
\end{aligned}$$

We proceed to the lower bound,

$$\begin{aligned}
D(M', \mathcal{V}'_\alpha) &= \sum_{V'_j \in \mathcal{V}'_\alpha} \frac{\sum_i \left|V'_j(i) - M'(i)\right|}{\sum_i \max(V'_j(i), M'(i))} \\
&\geq \sum_{V'_j \in \mathcal{V}'_\alpha} \frac{\sum_i |V_j(i) - M'(i)| - \frac{\alpha \cdot \varepsilon^2}{m}}{\sum_i \max(V_j(i), M'(i))} \\
&\geq D(M', \mathcal{V}_\alpha) - \sum_{V'_j \in \mathcal{V}'_\alpha} \frac{\frac{\alpha \cdot \varepsilon^2}{m}}{\sum_i \max(V_j(i), M'(i))} \\
&\geq D(M', \mathcal{V}_\alpha) - \varepsilon
\end{aligned}$$

Now, take an optimal median $M^*$ for $\mathcal{V}$; using the previous upper bound, with $M' := M^*$, we obtain

$$D(M^*, \mathcal{V}'_\alpha) \leq (1 + O(\varepsilon)) \cdot D(M^*, \mathcal{V}_\alpha) + O(\varepsilon).$$

Since $M$ is an $(1+O(\varepsilon))$-approximate median for $\mathcal{V}'_\alpha$, we will also have $D(M, \mathcal{V}'_\alpha) \leq (1+O(\varepsilon)) \cdot D(M^*, \mathcal{V}'_\alpha)$; thus,

$$D(M, \mathcal{V}'_\alpha) \leq (1 + O(\varepsilon)) \cdot D(M^*, \mathcal{V}_\alpha) + O(\varepsilon).$$

Finally, by $D(M, \mathcal{V}'_\alpha) \geq D(M, \mathcal{V}_\alpha) - \varepsilon$, we obtain that

$$D(M, \mathcal{V}_\alpha) \leq (1 + O(\varepsilon)) \cdot D(M^*, \mathcal{V}_\alpha) + O(\varepsilon).$$

Now we consider the vectors in $\mathcal{V} - \mathcal{V}_\alpha$. Take some $V_j \in \mathcal{V} - \mathcal{V}_\alpha$; we will either have,

- $\max_i V_j(i) \geq \alpha \cdot \frac{n}{\varepsilon}$, in which case

$$\sum_i \max(M^*(i), V_j(i)) \geq \sum_i V_j(i) \geq \max_i V_j(i) \geq \alpha \cdot n \cdot \varepsilon^{-1}.$$

Further,

$$\sum_i \min(M^*(i), V_j(i)) \leq \sum_i M^*(i) \leq \sum_i \alpha = n \cdot \alpha.$$

Then,

$$D(M^*, V_j) = 1 - \frac{\sum_i \min(M^*(i), V_j(i))}{\sum_i \max(M^*(i), V_j(i))} \geq 1 - \frac{n \cdot \alpha}{\alpha \cdot n \cdot \varepsilon^{-1}} = 1 - \varepsilon.$$

- or we will have $\sum_i V_j(i) < \varepsilon \cdot \alpha$. Then,

$$\sum_i \max(M^*(i), V_j(i)) \geq \sum_i M^*(i) \geq \max_i M^*(i) = \alpha.$$

On the other hand,

$$\sum_i \min(M^*(i), V_j(i)) \leq \sum_i V_j(i) \leq \varepsilon \cdot \alpha.$$

Again, these entail $D(M^*, \mathcal{V} - \mathcal{V}_\alpha) \geq 1 - \varepsilon$.

The maximum Jaccard distance is 1. Therefore, $D(X, \mathcal{V} - \mathcal{V}_\alpha) \leq (1 + O(\varepsilon)) \cdot D(M^*, \mathcal{V} - \mathcal{V}_\alpha)$ for each vector $X$ — and in particular for $X = M$.

Putting everything together, we get

$$\begin{aligned} D(M, \mathcal{V}) &= D(M, \mathcal{V}_\alpha) + D(M, \mathcal{V} - \mathcal{V}_\alpha) \\ &\leq ((1 + O(\varepsilon)) \cdot D(M^*, \mathcal{V}_\alpha) + O(\varepsilon)) + ((1 + O(\varepsilon)) \cdot D(M^*, \mathcal{V} - \mathcal{V}_\alpha)) \\ &= (1 + O(\varepsilon)) \cdot D(M^*, \mathcal{V}) + O(\varepsilon) \end{aligned}$$

### 4.7.8 "Canonical" medians

In this section, we prove how a simple polynomial rounding technique can transform non-canonical medians to canonical ones without decreasing their value. We say that a median $M$ for $\mathcal{V}$ is canonical iff, for each $i$, $M(i)$ is equal to $V(i)$ for some $V \in \mathcal{V}$.

In previous sections, (a) we showed how no FPTAS' exist for the problem of finding canonical medians for Jaccard (assuming P $\neq$ NP), and (b) we gave a PTAS for finding not-necessarily-canonical medians.

As each non-canonical median can be transformed into a canonical one of smaller or equal total distance, we have that the value of the optimal medians is the same in the canonical and not-necessarily canonical problems. The algorithm of this section, if given a non-canonical median, returns a canonical median of smaller value in polynomial time. Thus, if P $\neq$ NP, no FPTAS' exist for the not-necessarily canonical median problem, either.

Further, with this section, we show the existence of a PTAS for the canonical medians problem.

This section's lemma is quite similar to one of Späth [97], that shows how each optimal Jaccard median is canonical. We have it here in this form for completeness.

**Lemma 69.** *Let $M$ be a median for $\mathcal{V}$. Suppose there exists a coordinate $j$, such that $M(j) \notin \{V(j) \mid V \in \mathcal{V}\}$. Then,*

*i. if $M(j) > \max_{V \in \mathcal{V}} V(j)$, then*

$$M_j^- = \left( M(1), M(2), \ldots, M(j-1), \max_{\substack{V \in \mathcal{V} \\ V(j) < M(j)}} V(j), M(j+1), \ldots, M(n) \right)$$

*is a better median than $M$;*

*ii. if* $M(j) < \min_{V\in\mathcal{V}} V(j)$*, then*

$$M_j^+ = \left( M(1), M(2), \ldots, M(j-1), \min_{\substack{V\in\mathcal{V} \\ V(j)>M(j)}} V(j), M(j+1), \ldots, M(n) \right)$$

*is a better median than* $M$*;*

*iii. otherwise, either* $M_j^-$ *or* $M_j^+$ *is a better median than* $M$*.*

*Proof.* The first two cases are easy. If $M(j) > \max_{V\in\mathcal{V}} V(j)$, then, for each $V \in \mathcal{V}$, it holds that $\max(V(j), M(j)) = M(j) > M_j^-(j) = \max(V(j), M_j^-(j))$ and $\min(V(j), M(j)) = V(j) = \min(V(j), M_j^-(j))$. That is,

$$D(M,V) = 1 - \frac{\sum_j \min(V(j), M(j))}{\sum_j \max(V(j), M(j))} \geq 1 - \frac{\sum_j \min(V(j), M_j^-(j))}{\sum_j \max(V(j), M_j^-(j))} = D(M_j^-, V).$$

For the second case observe that if $M(j) < \min_{V\in\mathcal{V}} V(j)$, then, for each $V \in \mathcal{V}$, it holds that $\max(V(j), M(j)) = V(j) = \max(V(j), M_j^+(j))$ and $\min(V(j), M(j)) \leq M(j) \leq M_j^+(j) = \min(V(j), M_j^+(j))$. So that, again, $D(M,V) \geq D(M_i^+, V)$.

Consider the third case. Let $M_i'$ be such that $M_i'(j) = M(j)$, for each $j \neq i$. We define $s_{V,i} = \sum_{j\neq i} \min(V(j), M(j))$ and $S_{V,i} = \sum_{j\neq i} \max(V(j), M(j))$; then

$$D(V,M) = 1 - \frac{\min(V(i), M(i)) + s_{V,i}}{\max(V(i), M(i)) + S_{V,i}},$$

and

$$D(V,M_i') = 1 - \frac{\min(V(i), M_i'(i)) + s_{V,i}}{\max(V(i), M_i'(i)) + S_{V,i}}.$$

Further let $\mathcal{V}_< = \{V \mid V \in \mathcal{V} \wedge V(i) < M(i)\}$ and $\mathcal{V}_> = \{V \mid V \in \mathcal{V} \wedge V(i) > M(i)\}$. Observe that $\mathcal{V}_< \cup \mathcal{V}_> = \mathcal{V}$. Let us define $\delta = D(M,\mathcal{V}) - D(M_i',\mathcal{V})$; then,

$$\begin{aligned}
\delta &= \sum_{V\in\mathcal{V}} \left( \frac{\min(V(i), M_i'(i)) + s_{V,i}}{\max(V(i), M_i'(i)) + S_{V,i}} - \frac{\min(V(i), M(i)) + s_{V,i}}{\max(V(i), M(i)) + S_{V,i}} \right) \\
&= \sum_{V\in\mathcal{V}_<} \left( \frac{\min(V(i), M_i'(i)) + s_{V,i}}{\max(V(i), M_i'(i)) + S_{V,i}} - \frac{\min(V(i), M(i)) + s_{V,i}}{\max(V(i), M(i)) + S_{V,i}} \right) + \\
&\quad \sum_{V\in\mathcal{V}_>} \left( \frac{\min(V(i), M_i'(i)) + s_{V,i}}{\max(V(i), M_i'(i)) + S_{V,i}} - \frac{\min(V(i), M(i)) + s_{V,i}}{\max(V(i), M(i)) + S_{V,i}} \right) \\
&= \sum_{V\in\mathcal{V}_<} \left( \frac{V(i) + s_{V,i}}{M_i'(i) + S_{V,i}} - \frac{V(i) + s_{V,i}}{M(i) + S_{V,i}} \right) + \sum_{V\in\mathcal{V}_>} \left( \frac{M_i'(i) + s_{V,i}}{V(i) + S_{V,i}} - \frac{M(i) + s_{V,i}}{V(i) + S_{V,i}} \right) \\
&= \sum_{V\in\mathcal{V}_<} \left( (V(i) + s_{V,i}) \left( \frac{1}{M_i'(i) + S_{V,i}} - \frac{1}{M(i) + S_{V,i}} \right) \right) + \sum_{V\in\mathcal{V}_>} \frac{M_i'(i) - M(i)}{V(i) + S_{V,i}} \\
&= \sum_{V\in\mathcal{V}_<} \left( (V(i) + s_{V,i}) \cdot \frac{M(i) - M_i'(i)}{(M_i'(i) + S_{V,i})\,(M(i) + S_{V,i})} \right) + \sum_{V\in\mathcal{V}_>} \frac{M_i'(i) - M(i)}{V(i) + S_{V,i}} \\
&= (M_i'(i) - M(i)) \left( \sum_{V\in\mathcal{V}_>} \frac{1}{V(i) + S_{V,i}} - \sum_{V\in\mathcal{V}_<} \frac{V(i) + s_{V,i}}{(M_i'(i) + S_{V,i})\,(M(i) + S_{V,i})} \right)
\end{aligned}$$

Let $A = \sum_{V\in\mathcal{V}_>} \frac{1}{V(i)+S_{V,i}}$ and $B(x) = \sum_{V\in\mathcal{V}_<} \frac{V(i)+s_{V,i}}{(x+S_{V,i})(M(i)+S_{V,i})}$. Observe that $0 < x_1 < x_2$ implies $B(x_1) > B(x_2)$. Then,

$$\delta = (M_i'(i) - M(i)) \cdot \left(A - B_{M_i'(i)}\right)$$

We will either choose $M'_i = M_i^+$ or $M'_i = M_i^-$. Suppose that $A - B_{M_i^+} > 0$. Then, choosing $M'_i = M_i^+$ will guarantee that $\delta > 0$ (as $M_i^+(i) - M(i) > 0$) and therefore $D(M, \mathcal{V}) > D(M_i^+, \mathcal{V})$.

On the other hand, if $A - B_{M_i^+} < 0$, then we will also have $A - B_{M_i^-} < 0$, by $B_{M^-(i)} \geq B_{M^+(i)}$. Thus, choosing $M'_i = M_i^-$ will give $D(M, \mathcal{V}) > D(M_i^-, \mathcal{V})$ (as $\delta > 0$, by $M_i^-(i) - M(i) < 0$).

So, to obtain a canonical median $M^*$ out of a non-canonical median $M$, one can do the following. Suppose $M$ is non-canonical on some coordinate $j$. Then either $M_j^+$ or $M_j^-$ are better medians than $M$ — update $M$ to be the optimal between $M_j^+$ and $M_j^-$. Now, either $M$ is canonical, or it will have one fewer non-canonical coordinate — in the latter case, repeat the process. After $n$ iterations, $M$ will necessarily be canonical.

### 4.7.9 $O(\varepsilon m)$ additive approximation algorithm

We present the full pseudocode for the algorithm in Algorithm 4.1.

**Algorithm 4.1** An $O(\varepsilon m)$ additive approximation polynomial algorithm, assuming $\varnothing \notin \mathcal{S}$.

$\mathcal{C} \leftarrow \{\varnothing\}$
**for all** $t \in [1, \ldots, n]$ **do**
  Let $\mathcal{S}_t = \{S \in \mathcal{S} \mid \varepsilon \cdot t \leq |S| \leq \frac{t}{\varepsilon}\}$ be the class of sets having size in $\left[\varepsilon \cdot t, \frac{t}{\varepsilon}\right]$.
  For each $x \in U$, let $\deg_t(x) = |\{S \in \mathcal{S}_t \mid x \in S\}|$ be the degree of $x$ in $\mathcal{S}_t$.
  Let $U_t = \{x \in U \mid \deg_t(x) \geq \varepsilon \cdot m\}$ be the set of "high degree" elements in $\mathcal{S}_t$.
  **if** $|U_t| \leq 9 \cdot \varepsilon^{-6} \cdot \ln(nm)$ **then**
    **for all** $Y_t \subseteq U_t$ **do**
      $\mathcal{C} \leftarrow \mathcal{C} \cup \{Y_t\}$
    **end for**
  **else**
    $P \leftarrow \varnothing$
    **for all** $x \in U_t$ **do**
      Flip a coin with head probability $p = 9 \cdot \varepsilon^{-6} \cdot |U_t|^{-1} \cdot \ln(nm)$.
      **if** it comes up head **then**
        $P \leftarrow P \cup \{x\}$.
      **end if**
    **end for**
    **for all** $M_c \subseteq P$ **do**
      Let $\mathcal{S}_t(M_c) = \left\{S \in \mathcal{S}_t \mid |S \cap M_c| > 6 \cdot \varepsilon^4 \cdot |U_t| \cdot p\right\}$ be the subset of $\mathcal{S}_t$ containing sets with "non-vanishing" intersection with $M_c$.
      Solve the following system of linear inequalities in $(\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x}_\tau)$,

$$\begin{cases} 0 \leq \mathbf{x_i} \leq 1 & \forall i, 1 \leq i \leq \tau \\ \sum_{x_i \in S \cap U_t} \mathbf{x_i} \leq (1-\varepsilon)^{-1} \cdot |S \cap M_c| \cdot p^{-1} & \forall S \in \mathcal{S}_t(M_c) \\ \sum_{x_i \in S \cap U_t} \mathbf{x_i} \geq (1+\varepsilon)^{-1} \cdot |S \cap M_c| \cdot p^{-1} & \forall S \in \mathcal{S}_t(M_c) \\ \sum_{x_i \in U_t} \mathbf{x_i} \leq (1-\varepsilon)^{-1} \cdot |M_c| \cdot p^{-1} \\ \sum_{x_i \in U_t} \mathbf{x_i} \geq (1+\varepsilon)^{-1} \cdot |M_c| \cdot p^{-1} \end{cases}$$

      **if** there exists a solution $(\hat{\mathbf{x}}_\mathbf{1}, \ldots, \hat{\mathbf{x}}_\tau)$ **then**
        $Y_t \leftarrow \varnothing$
        **for all** $x_i \in U_t$ **do**
          Flip a coin with head probability $\hat{\mathbf{x}}_\mathbf{i}$.
          **if** it comes up head **then**
            $Y_t \leftarrow Y_t \cup \{x_i\}$.
          **end if**
        **end for**
        $\mathcal{C} \leftarrow \mathcal{C} \cup \{Y_t\}$
      **end if**
    **end for**
  **end if**
**end for**
Output the optimal candidate median in $\mathcal{C}$.

# Chapter 5

# Preference Preorder

Choosing the best representatives from a set of objects is not an easy task; even the very nature of preference is questionable, and has been largely debated by psychometricians and economists (see, e.g., [36]). In particular, a much argued point is whether personal preference is a transitive relation or not; even if some experiments actually prove that preference may indeed not be transitive, it is commonly agreed that intransitive preference lead to irrational behaviour. For this reason, it is by now accepted that preference can be modelled as a *preorder* (a.k.a. *quasiorder*), that is, a transitive, reflexive relation that is not required to be antisymmetric; for sake of simplicity, in this chapter we shall mainly focus our attention on *partial orders*, although all our algorithms can be easily extended to preorders (we discuss this issue in Section 5.5).

So, in an abstract sense, we shall consider the problem of choosing the "best" $t$ elements from a set $X$ of $n$ elements subject to a given partial order; the exact notion of "best elements" will be discussed shortly[1]. In our intended application, $X$ is the set of all pictures, whereas $t$ is the number of pictures we are going to put in our album, and the order reflects the preference of some given, rational subject.

An algorithm for this problem will present a sequence of pairs of pictures to the subject and ask her to choose which of the two pictures she likes best, or if she is completely indifferent about the two pictures. The computational cost we are going to take into consideration is the number of comparisons the subject is requested to perform; notice, in particular, that we shall not be considering the number of instructions executed or the space occupied by the data (even though all our algorithms have a polynomial running time).

Apart for pictures, the problem may find more interesting applications to other areas; for example, our algorithms suggest a new way to perform rank aggregation in information-retrieval tasks. Rank aggregation is the problem of combining ranking results from various sources; in the context of the Web, the main applications include building meta-search engines, combining ranking functions, selecting documents based on multiple criteria, and improving search precision through word associations. Here, by rank we mean an assignment of a relevance score to each candidate. Traditional approaches to rank aggregation [45, 50] assume that ranks be aggregated in a single rank value that satisfies suitable properties. Instead, we propose to produce a partial order by intersection, and then identify suitable "best elements" of the partial order obtained in this way.

For more information on the possible applications of the problem we discuss, we direct the reader to an independently developed paper by Daskalakis, Karp, Mossel, Riesenfeld and Verbin [38] that contains some results similar in nature to the ones discussed here.

## 5.1 Basic setup

A *partial order* on a set $X$ is a binary relation $\preceq$ on $X$ that is reflexive ($x \preceq x$ for all $x \in X$), antisymmetric ($x \preceq y$ and $y \preceq x$ imply $x = y$) and transitive (if $x \preceq y$ and $y \preceq z$ then $x \preceq z$); a *poset* is a set endowed with a partial order; here, all posets will be finite. We let $x \prec y$ mean $x \preceq y$ and $x \neq y$.

[1] Note that this problem is significantly different from the complete reconstruction of the poset, a task whose complexity has been largely studied in the literature, starting from the seminal paper [51].

Two elements $x, y \in X$ such that either $x \preceq y$ or $y \preceq x$ are called *comparable*, otherwise they are called *incomparable*. A set of pairwise comparable (incomparable, resp.) elements is called a *chain* (*antichain*, resp.). The *width* of a poset is the maximum cardinality of an antichain.

We say that $x$ *covers* $y$ iff $y \prec x$ and, for every $z$, $y \preceq z \preceq x$ implies either $y = z$ or $z = x$. Intuitively, $x$ is "just better" than $y$.

A *maximal (minimal, resp.) element* in a subset $Y$ of a poset $(X, \preceq)$ is any element $y \in Y$ such that $y \preceq y' \in Y$ ($Y \ni y' \preceq y$, resp.) implies $y = y'$. Maximal elements are pictures for which we cannot find a better one (but there might be many incomparable maximal elements).

The notation $x{\downarrow}$ denotes the set of all elements smaller than or equal to $x$.

## 5.2 Setting our goal

We wish to give a sound and meaningful definition of what are the "best" elements of a given poset; this notion is quite obvious in the case of a total order, but turns out to be much subtler when partial orders are involved.

As a first attempt, an *upper set* seems a good idea. Given a poset $(X, \preceq)$, a set $Y \subseteq X$ is an *upper set* iff $y \preceq z$ and $y \in Y$ imply $z \in Y$. Indeed, if we choose an upper set we are sure that no picture more valuable than the ones chosen has been missed.

There is however a significant problem with this definition when we try to give an intuitive, user-related meaning to incomparability. Our friends coming from Mongolia have pictures from Lake Hovsgol and also from the Gobi Desert[2]. It is reasonable to assume that it is very difficult to define preferences between pictures containing just sand and pictures containing just water, so every picture from Gobi could be incomparable with every picture from Lake Hovsgol. But in this case the pictures from Gobi would be an upper set, and a subset of Gobi pictures that is also an upper set could provide a valid answer.

In other words, if we interpret incomparability as "different setting" or "different topic" or "different subject", we would like, say, to have at least the best picture for every setting, or topic, or subject. Once we have the best pictures, if we still have room we would like to get the *second* best pictures, and so on: intuitively, we would like to *peel* the top of the poset by iteratively choosing the best pictures available. This idea suggests a more stringent definition:

**Definition 70.** *Let $(X, \preceq)$ be a poset, and define a sequence $X_0$, $X_1$, $X_2$, ... of disjoint subsets of $X$, called* levels*, where $X_t$ is the set of maximal elements in $X \setminus (X_0 \cup X_1 \cup \cdots \cup X_{t-1})$ (the sequence is ultimately $\varnothing$). Let us define a new order $\sqsubseteq$ on $X$ by setting $X_0 \sqsupseteq X_1 \sqsupseteq X_2 \sqsupseteq \cdots$ (elements within each $X_i$ are incomparable). A* top set *is an upper set w.r.t. $\sqsupseteq$; a top set of size $t$ is also called a $t$-top set.*

More explicitly, top sets are exactly sets of the form $X_0 \cup X_1 \cup \cdots \cup X_{i-1} \cup X'_i$, where $X'_i \subseteq X_i$.

Note that $\sqsubseteq$ is an extension of $\preceq$ (if $x \preceq y$ and $y \in X_i$ necessarily $x \in X_j$ for some $j > i$): as a consequence, all top sets are upper sets (for $\preceq$). The extension is in general proper: if we consider the set $X = \{ x, y, x', y' \}$ ordered by $x \preceq x'$, $y \preceq y'$, we have that additionally $x \sqsubseteq y'$.

At this point, we are in a position to give a more precise definition of our problem: we are given a poset $(X, \preceq)$ of $n$ elements, and an integer $t \leq n$, and we must output a $t$-top set[3]; the poset is available only implicitly, in the sense that we do not know $\preceq$ in advance, but our algorithm can use an oracle that, given two distinct elements $x, y \in X$, answers either "$x \prec y$", or "$x \succ y$" or "$x$ and $y$ are incomparable": such a call to the oracle is named *comparison* (or query), and we want our algorithm to perform as few comparisons as possible.

To measure the loss w.r.t. a hypothetical optimal algorithm, we will say that an algorithm is $\theta$*-slow* if it never makes more than

$$\theta \cdot \min_{\mathscr{A}} \max_{|P|=n} q(\mathscr{A}, P, t)$$

queries to return the $t$-top set of a poset of size $n$, where $q(\mathscr{A}, P, t)$ is the number of queries performed by algorithm $\mathscr{A}$ on the poset $P$ of size $n$ to return a $t$-top set.

[2] Incidentally, "Gobi" in Mongolian means "desert", which makes the expression "Gobi Desert" bizarre at best.

[3] It may be worth noting that almost all the algorithms we are going to present can be modified so to output not only a $t$-top set $T$, but also, for each $x \in T$, the level $i$ to which $x$ belongs (i.e., such that $x \in X_i$).

Of course, there is a trivial algorithm that performs $\binom{n}{2}$ comparisons and rebuilds the whole poset, and in the worst case we cannot hope to perform a smaller number of comparisons: every algorithm must perform at least $\binom{n}{2}$ comparison on a poset made of $n$ incomparable elements to output a top set of $n-1$ elements. Indeed, if $x$ and $y$ are never compared and, say, $y$ is not in the output, the algorithm would fail for a poset in which $x \prec y$ (note that for this proof to work the choice to use top sets vs. upper sets is essential). This consideration can be generalised to any $t < n$:

**Proposition 71.** *Every correct algorithm must, in the worst case, perform at least*

$$\frac{1}{2}t(2n-t-1) \geq \frac{1}{2}tn$$

*comparisons ($0 < t < n$) to determine a $t$-top set of a $n$-elements poset.*

*Proof.* Assume that the algorithm outputs a $t$-top set $T$ of a poset of $n$ elements without comparing every element of $T$ with every element of $P$; then, there is some $x \in T$ and some $y \in P$ that have not been compared, and the algorithm would fail on a poset in which the only comparable pair is $x \prec y$. Indeed, in that case $x$ should not be in the output, as $t < n$ and there are $n-1$ maximal elements. Hence, the algorithm performed at least $\binom{t}{2} + t(n-t)$ comparisons, which is the left-hand side of the inequality in the statement (the right-hand side follows from $t < n$).

We note in passing that a similar lower bound holds for upper sets. The previous proof can be easily amended to provide a $t(n-t)$ lower bound on the number of comparisons (if you output $t$ elements and you did not check against some of the remaining $n-t$, your output might not be an upper set). On the other hand, finding an upper set in $O(t(n-t))$ comparisons is trivial: if $t \leq n/2$ just look in a brute-force manner for a maximal element ($n-1$ comparisons), then find another maximal element in the remaining $n-1$ elements ($n-2$ comparisons) and so on. This requires $(n-1)+(n-2)+\cdots+(n-t) = tn - t(t+1)/2 \leq \frac{3}{2}t(n-t)$ comparisons. If $t > n/2$ we can work backwards, eliminating iteratively minimal elements, and returning the remaining ones.

Maybe surprisingly, an absolutely analogous approach yields a 2-slow algorithm for top sets. The only difference is that once we find a maximal element $m$, we start searching for elements incomparable with $m$ in the poset, and we output them. Some care, however, must be exercised, as once an incomparable element has been output, all smaller elements are not good candidates for the output, even if they are incomparable with $m$. So we must keep track of which elements of the poset should be checked for incomparability with $m$, and update them each time we output a new incomparable element. The details are given in algorithm 5.1.

**Algorithm 5.1** A 2-slow algorithm.

**top**($P$: a poset, $t$: an integer)

```
1:  C ← ∅ {Candidate set: elements of the level to be peeled are to be found in here.}
2:  L ← P {Elements left for the next round.}
3:  for t times do
4:    if C = ∅ then
5:      {We prepare to peel a new level}
6:      C ← L
7:      L ← ∅
8:    end if
9:    set m to a maximal element of C
10:   output m
11:   add m↓ minus m to L {Keep all elements smaller than m for the next round...}
12:   remove m↓ from C {... but remove m and all smaller elements from the candidates.}
13: end for
```

**Theorem 72.** *Algorithm 5.1 outputs a $t$-top set.*

*Proof.* Let $O$ be the set of already output elements. We show that at the start of the loop, $C \cup L = P \setminus O$, and the maximal elements of $C$ are the maximal elements of $P \setminus (O \cup L)$. Indeed, the execution of the if

statement (i.e., when $C$ is empty, hence $L = P \setminus O$) does not change these facts. When we output $m$ we make it disappear from $C$ and appear in $O$, preserving the equation above (the elements strictly smaller than $m$ are transferred from $C$ to $L$). Finally, since we remove $m{\downarrow}$ from $C$, $m$ also disappears from the set of maximal elements of $C$, and no new maximal element appears. This implies that the second property is preserved, too.

To conclude the proof, we note that just after the $i$-th execution of the if statement, $C$ will contain the entire poset $P$ minus the first $i$ levels. This is certainly true at the first reassignment, and thus the maximal elements of $C$ are exactly the first level of $P$. Since we remove them one by one, by the time we execute the second reassignment we will have output exactly the first level, so we will now assign to $C$ the entire poset minus the first level, and so on.

**Theorem 73.** *Algorithm 5.1 requires no more than $t(2n-t-1)$ comparisons to output a $t$-top set of a poset with $n$ elements. Thus, it is 2-slow.*

*Proof.* The algorithm performs some comparisons only when finding maximal elements, and when computing $x{\downarrow}$. In both cases at most $|C|-1$ comparisons are needed, using a brute-force approach. Note that at the start of the $i$-th iteration of the loop $|C|+|L| = n-i$, as several elements are moved between $C$ and $L$, but exactly one element is output (and removed from $C$) at each iteration, so in particular $|C| \leq n-i$. The total number of comparison is then bounded by $2\left((n-1)+\cdots+(n-t)\right) = 2tn-t(t+1) = t(2n-t-1)$. The last claim then follows immediately from Proposition 71.

## 5.3 Small-width posets

The reader might have noticed that our lower bound uses very wide posets (i.e., with width $\Theta(n)$). It is thus possible that by limiting the width we can work around the lower bound, getting a better algorithm for small-width posets. Indeed, we expect that settings, topics or subjects should be significantly fewer than the number of pictures, or it would be very difficult to choose a small best subset.

Looking into Algorithm 5.1, it is clear that we are somehow wasting our time by scanning for one maximal element at a time. A more efficient strategy could be building the set of maximal elements in one shot, peeling them, building the new set of maximal elements, and so on.

There are two difficulties with this approach: first of all, if we need a very small top set we could make much more queries than necessary, as the set of maximal elements could be significantly larger than the required top set. Second, rebuilding the set of maximal elements after each peeling could be expensive.

There is not much we can do about the first problem, but our next algorithm (Algorithm 5.2) tries to address the second one. If we divide $P$ into two subsets and compute the respective sets of maximal elements, the union of these two sets will certainly contain all maximal elements of $P$ (plus some spurious elements). We can apply this reasoning recursively, building the set of maximal elements incrementally. To avoid an expensive recomputation after each peeling, we will arrange the elements of the poset $P$ arbitrarily on the leaves of a complete binary tree $T$. The binary tree then induces naturally a hierarchical partition of the elements of $P$—just look at the leaves of each node of given depth. We will keep track of the maximal elements of each subset of the partition, by suitably labelling each node of the tree. Initially, we will find at the root the set $M$ of maximal elements of $P$, which we can output. Then, we will remove the elements of $M$ from all labels in the tree, and recompute the new labels. The process will continue until we have output enough elements.

Note that the width of the poset does not appear explicitly in the description of the algorithm. Indeed, it will surface only during the analysis, when we shall put to good use the fact that the labels on each node cannot be of cardinality larger than the poset width.

In what follows we shall tacitly assume that the algorithm uses some data structure to keep track of the comparisons that have already been performed; in other words, for every pair $x, y \in P$, we are able to tell whether $x, y$ have ever been compared and, if so, the result of the comparison. This assumption makes it possible to bound the number of comparisons using just the *number of pairs ever compared*.

**Theorem 74.** *Algorithm 5.2 is correct.*

*Proof.* To prove the statement, it is sufficient to show that after the $i$-th call to **completeLabelling** the label of the root is the $i$-th level of the input poset $P$. Given a node $v$ of $T$, we define the $v$-dominated

**Algorithm 5.2** An algorithm for small-width posets

**top**(t: an integer)
1: let $T$ be a complete binary tree labelled on subsets of $P$ and with $n$ leaves
2: label each leave of $T$ with a singleton containing a distinct element of $P$
3: label the remaining nodes with $\varnothing$
4: **while** $t > 0$ **do**
5: **completeLabelling**($T$)
6: let $A$ be the label of the root of $T$
7: output $u \leftarrow \min\{\, t, a \,\}$ elements from $A$
8: $t \leftarrow t - u$
9: remove the elements of $A$ from all the labels
10: **end while**

**completeLabelling**($T$: a binary tree)
1: recursively consider all non-leaf nodes $v$ of $T$, bottom-up
2: let $z_0$, $z_1$ be the two children of $v$
3: **for** $k \leftarrow 0, 1$ **do**
4: **for** $x$ a label of $z_k$ **do**
5: if no label of $z_{1-k}$ is greater than $x$ add $x$ to the label of $v$
6: **end for**
7: **end for**

poset $v{\Downarrow}$ as the subset of elements of $P$ contained in leaves dominated by $v$ with the order inherited from $P$. It is immediate to show by induction that if each node $v$ of $T$ is labelled by a (possibly empty) set of maximal elements of $v{\Downarrow}$, then after a call to **completeLabelling** each node will be labelled by the set of *all* maximal elements of $v{\Downarrow}$. Indeed, during the recomputation of the labels we compute the maximal elements of the union of the labels of the children. But the labels of the children contain (by induction) all maximal elements of the respective dominated posets, and all maximal elements of $v{\Downarrow}$ must appear in the labels of the children of $v$ (if we partition a partial order in two disjoint subsets, all maximal elements of the partial order are still maximal elements of the subset they belong).

Thus, the label of the root after the first iteration is the first level of $P$. Then, either the algorithm stops, or we remove the first level and call again **completeLabelling**. Since now the tree contains just $P$ minus its first level, and all labels are still maximal, the label of the root will be the second level, and so on.

**Theorem 75.** *Algorithm 5.2 finds a $t$-top set of a poset with $n$ elements and width at most $w$ using*

$$\frac{3}{2}wn + wt(\lfloor \log n \rfloor - \lceil \log w \rceil)$$

*comparisons.*

*Proof.* We split the estimate of the number of comparisons in two parts. Define the *depth* of a node in the standard way (the root has depth 0, and the children of nodes at depth $d$ have depth $d+1$). We shall call the nodes of depth smaller than $\lfloor \log n \rfloor - \lceil \log w \rceil$ *interesting*, and the remaining nodes *uninteresting*. Note that the depth of the deepest node performing comparisons is $\lceil \log(n+1) \rceil - 2$, and that

$$\lceil \log(n+1) \rceil - \lfloor \log n \rfloor = 1 \text{ for all integers } n.$$

Since we never repeat a comparison, all comparisons that could be ever performed on uninteresting nodes are very easily bounded:

$$\sum_{i=\lfloor \log n \rfloor - \lceil \log w \rceil}^{\lceil \log(n+1) \rceil - 2} 2^i \left(2^{\lceil \log(n+1) \rceil - 2 - i}\right)^2 \leq n 2^{\lceil \log w \rceil - 1}.$$

We now focus the rest of the proof on the interesting nodes. The number of elements contained in each child of an interesting node can be just bounded by $w$, so in principle each time we need to update the

labels of an interesting node we might need $w^2$ comparisons. This very rough estimate can be significantly improved noting that we cannot increase indefinitely the number of elements in a child (as it is bounded by $w$). Indeed, each time we add new elements in a node, this generates some new comparisons in its parent. Nonetheless, as long as we *add* elements, the number of overall comparisons performed by the parent approaches $w^2$, but can never get past it. So if we estimate an initial cost

$$\sum_{i=0}^{\lfloor \log n \rfloor - \lceil \log w \rceil - 1} 2^i w^2 \leq 2^{\lfloor \log n \rfloor - \lceil \log w \rceil} w^2 \leq n 2^{-\lceil \log w \rceil} w^2,$$

this estimate will cover all *additions*, as long as elements are never deleted.

Before considering the deletion of elements, let us bound the sum $n2^{\lceil \log w \rceil - 1} + n2^{-\lceil \log w \rceil} w^2$:

$$\begin{aligned} n2^{\lceil \log w \rceil - 1} + n2^{-\lceil \log w \rceil} w^2 &= n\left(2^{\lceil \log w \rceil - 1} + w^2 2^{-\lceil \log w \rceil}\right) \\ &= nw\left(\frac{1}{w}2^{\lceil \log w \rceil - 1} + w2^{-\lceil \log w \rceil}\right) = nw\left(2^{\lceil \log w \rceil - \log w - 1} + 2^{\log w - \lceil \log w \rceil}\right). \end{aligned}$$

We let $x = \lceil \log w \rceil - \log w$ (note that $x \in [0\,..\,1)$) so the previous expression becomes $nw\left(2^{x-1} + 2^{-x}\right)$. The function $g(x) = 2^{x-1} + 2^{-x}$ decreases for $x < \frac{1}{2}$ and increases for $x > \frac{1}{2}$. Thus, in the interval $[0\,..\,1)$, $g(x) \leq \max\{\, g(0), g(1) \,\} = \frac{3}{2}$. We conclude that

$$n2^{\lceil \log w \rceil - 1} + n2^{-\lceil \log w \rceil} w^2 \leq \frac{3}{2} nw.$$

Now, what happens when elements are deleted? In this case, the deleted element can be replaced by a new one, and its cost is not included in our previous estimate. So we must fix our bound by adding $w$ comparisons for each node in which a deletion happens.

Let us make the above considerations formal. Each interesting node $v$ has a *bonus* of $w^2$ comparisons included in the estimate above. Let $v$ ambiguously denote the number of labels of a node $v$ *before* a call to **completeLabelling** (so in particular $v = 0$ for all interesting $v$ at the first iteration), and $v^*$ the number of labels of $v$ after the call. If $\ell$ and $r$ are the left and right child of $v$ we show that, provided $w^2 - \ell r$ bonus comparisons are available before the call, $w^2 - \ell^* r^*$ are still available afterwards.

When we call **completeLabelling**, the labels of each node $v$ must be updated. The update involves comparing new elements that appeared in the children of $v$; at most $(\ell^* - \ell)r + (r^* - r)\ell + (\ell^* - \ell)(r^* - r)$ comparison are needed to update $v$. But since $w^2 - \ell r - (\ell^* - \ell)r - (r^* - r)\ell - (\ell^* - \ell)(r^* - r) = w^2 - \ell^* r^*$ the invariant is preserved. The invariant is trivially true before the first call, so we conclude that the costs of **completeLabelling** are entirely covered by the bonus.

Finally, when we remove elements from the tree, each removed element potentially alters $\lfloor \log n \rfloor - \lceil \log w \rceil$ interesting nodes, reducing by 1 the number of its labels. Thus, to keep the invariant true we might need to cover some costs: with the same notation as above, if, for instance, the set of labels of $\ell$ becomes smaller we need to compensate $r+1 \leq w$ comparisons to keep the invariant true for node $v$ (symmetrically for $r$). Thus, removing an element requires patching the bonus by at most $w(\lfloor \log n \rfloor - \lceil \log w \rceil)$ additional comparisons.

All in all, emitting $t$ elements will require a fixed cost of $\frac{3}{2}nw$ comparisons, plus at most $w(\lfloor \log n \rfloor - \lceil \log w \rceil)$ comparisons for each deleted element; since the deleted elements are bounded by $t$ the result follows easily.

Note that if $w = o(t)$ Algorithm 5.2 is advantageous over Algorithm 5.1, as

$$wt \log \frac{n}{w} = nt\left(\log \frac{n}{w}\right) / \left(\frac{n}{w}\right) = o(nt).$$

The opposite happens if $t = o(w)$. To see that this is not an artifact of the analysis of the algorithm, just note that on a poset formed by $w$ chains, each of height $\lceil n/w \rceil$, if we distribute each level (which has width $w$) on a set of adjacent leaves *all* possible comparisons on uninteresting nodes will be performed during the first call to **completeLabelling**, so the algorithm actually requires $\Omega(wn) = \omega(tn)$ comparisons if $t > 0$.

As we will now show, the difference between the case $t = o(w)$ and the case $w = o(t)$ is not an artifact of our analysis or of our algorithms, but rather an intrinsic feature of the problem.

**Theorem 76.** *Every correct algorithm must, in the worst case, perform at least*

$$\frac{w}{2}(n-t)$$

*queries to get a $t$-top set of a $n$-elements poset of width $w \leq t$.*

*Proof.* Suppose that a certain algorithm is trying to determine the $t$-top set of a $n$-elements poset of width $w$; we build the poset adversarially using a $w$-colourable graph $G$ with $n$ vertices $\{0, 1, \ldots, n-1\}$ (each vertex represents an element of the poset). At the beginning, the graph $G$ contains $n-t$ isolated vertices $\{0, 1, \ldots, n-t-1\}$, $\lfloor t/w \rfloor$ cliques containing $w$ vertices each and possibly (if $w$ does not divide $t$) a clique containing the last $t \bmod w$ elements; let us denote by $U$ the elements of this last clique.

Every time the algorithm queries $x :: y$, we decide the answer as follows:

- if one element belongs to $U$ and the other does not, we answer that the element in $U$ is larger;
- otherwise, if $x$ and $y$ are adjacent or can be made adjacent while still keeping the graph $w$-colourable, we answer "incomparable" and add the edge $\{x, y\}$, if it is not already in the graph;
- otherwise, we answer $\prec$ or $\succ$ depending on whether $x < y$ or $x > y$; note that in such a case every $w$-colouring assigns the same colour to $x$ and $y$: we say that $x$ and $y$ are *tainted*; when a vertex becomes tainted, it has degree at least $w-1$, and it obviously remains tainted thereafter.

At any time, you can build a poset of width $w$ compatible with all the answers given so far as follows: take any $w$-colouring of $G$, say $c : \{0, 1, \ldots, n-1\} \to \{0, 1, \ldots, w-1\}$ and consider the partial order $\preceq_c$ defined by $x \preceq_c y$ iff $c(x) = c(y)$ and $x \leq y$, or $x < n - t \bmod w \leq y$. In such a poset, the only $t$-top set is $\{n-t, n-t+1, \ldots, n-1\}$, so this must be the correct output: indeed, the last $n-t \bmod w$ elements are larger than all the other ones; once these are removed, the elements $\{n-t, \ldots, n-t \bmod w - 1\}$ form $\lfloor t/w \rfloor$ cliques, each corresponding to a layer in the poset.

Note that if the algorithm gives an output when a vertex $z$ out of the first $n-t$ ones is not tainted, the output would be wrong if we made $z$ the largest among the elements not in $U$ with the same colour as $z$. But a vertex $z$ out of the first $n-t$ ones must have been subject of at least $w$ queries to become tainted. We conclude that $w(n-t)/2$ queries are necessary to guarantee that all of the first $n-t$ vertices are tainted.

With the same construction as in the previous proof, but using a $t$-colourable graph and adding a $w$-sized clique of *smallest* elements, we can obtain the following result for the case $t < w$:

**Corollary 77.** *Every correct algorithm must, in the worst case, perform at least*

$$\frac{t}{2}(n-t-w)$$

*queries to get a $t$-top set of a $n$-elements poset of width $w > t$.*

The previous theorems imply that, when $t \geq w$ and $t = o\left(n/\log \frac{n}{w}\right) \supseteq o(n/\log n)$, Algorithm 5.2 is $(3+\varepsilon)$-slow; when $t < w$ and $w = o(n)$ Algorithm 5.1 is $(4+\varepsilon)$-slow. Here $\varepsilon = \varepsilon(n)$ and $\lim_{n\to\infty} \varepsilon(n) = 0$.

## 5.4 A probabilistic algorithm

We now attack the problem from a completely different viewpoint. Since our lower bounds are based on very peculiar posets, we resort to a wonderful asymptotic structure theorem proved by Kleitman and Rothschild [68]: almost all posets of $n$ elements are *good*, that is, they are made of three levels of approximate size $n/4$, $n/2$, and $n/4$, respectively[4]. The idea is that of writing the algorithm as if *all* posets were good. In this way we will get an asymptotically very fast (albeit "probably" useless) algorithm that returns top sets on almost all posets.

Before embarking on this task, we should point out that we are not claiming that the structure of good posets reflects a real-world situation, especially not in the case of pictures; rather, it should be considered

[4]The original paper creates levels by stripping *minimal* elements, but by duality all results are valid also in our case.

a promising starting point for further investigation, and a base case that is useful when nothing is known about the actual distribution of inputs.

Stated in a slightly more precise form, Kleitman and Rothschild prove the following property. If you draw at random a poset $(X, \preceq)$ uniformly among all posets of $n$ elements, then with probability $1 - O(1/n)$ the poset is *good*, that is,[5]

- $X$ can be partitioned into three antichains $L_1$, $L_2$ and $L_3$;
- $\left||L_i| - \frac{n}{4}\right| \leq \sqrt{n} \log n$ for $i = 1, 3$;
- every element in $L_i$ only covers elements in $L_{i+1}$ (for $i = 1, 2$); hence, in particular, every element in $L_3$ is minimal and every element in $L_1$ is maximal;
- every non-maximal (non-minimal, resp.) element is covered by (covers, resp.) at least $n/8 - n^{7/8}$ elements.
- every element in $L_1$ ($L_3$) covers (is covered by, resp.) at least $n/4 - n^{7/8}$ elements of $L_2$.

In other words, almost every poset is made by just three antichains, and contains about $n/4$ minimal and maximal elements, whereas the remaining elements are just "sandwiched" between a maximal and a minimal element. The main idea of our first algorithm is that if we need no more than $\big(1/4 - \varepsilon\big)n$ top elements, they are easy to find if the poset is good—and almost all posets *are* good. Indeed, top elements are so easy to find that we can just extract three elements at random, hoping that they form a chain, and in that case take the largest element of the chain, as described in Algorithm 5.3 (the bound on the number of comparisons can be immediately derived from the condition of the while loop).

**Algorithm 5.3** An $O(t + \log n)$ algorithm for good posets and $t \leq \big(1/4 - \varepsilon\big)n$.

**top**($P$: a poset of order $n$, $t$: a positive integer, with $t \leq \big(1/4 - \varepsilon\big)n$)
1: $t' \leftarrow \max\left(64t/(5\varepsilon), 256 \ln n/(5\varepsilon)\right)$
2: $T \leftarrow \varnothing$
3: $i \leftarrow 0$
4: **while** $|T| < t$ **and** $i < t'$ **do**
5: choose u.a.r. without replacement 3 elements $x_1, x_2, x_3$ from $P$
6: perform all comparisons between $x_1, x_2, x_3$
7: **if** the $x_i$'s form a chain and $x_{\bar{\imath}}$ is the maximum **then**
8: $T \leftarrow T \cup \{\, x_{\bar{\imath}} \,\}$
9: $P \leftarrow P \setminus \{\, x_{\bar{\imath}} \,\}$
10: **end if**
11: $i \leftarrow i + 1$
12: **end while**
13: **if** $|T| = t$ **then**
14: **return** $T$
15: **else**
16: **fail**
17: **end if**

Let $\xi_{\text{fail}}$ be the event that the algorithm fails, and let $\xi_{\text{good}}$ be the event that the chosen poset is good. Then

$$P[\xi_{\text{fail}}] = P[\xi_{\text{fail}} \mid \xi_{\text{good}}]P[\xi_{\text{good}}] + P[\xi_{\text{fail}} \mid \overline{\xi_{\text{good}}}]P[\overline{\xi_{\text{good}}}] \leq$$
$$\leq P[\xi_{\text{fail}} \mid \xi_{\text{good}}] + P[\overline{\xi_{\text{good}}}] \leq P[\xi_{\text{fail}} \mid \xi_{\text{good}}] + O\left(\frac{1}{n}\right).$$

So, $P[\xi_{\text{fail}} \mid \xi_{\text{good}}] = O(1/n)$ entails $P[\xi_{\text{fail}}] = O(1/n)$.

[5]The result proved in [68] is actually stronger, but we are only quoting the properties of good posets that we will be needing in our proof.

**Algorithm 5.4** An $O(n)$ algorithm for good posets and every $t$.

**top**($P$: a poset of order $n$, $t$: a positive integer)

1: $P_1 \leftarrow \varnothing, P_2 \leftarrow \varnothing, P_3 \leftarrow \varnothing$
2: $P' \leftarrow P$
3: **while** $|P'| \geq 524 \ln n$ **do**
4: **for all** $x_1 \in P'$ **do**
5: choose u.a.r. without replacement two elements $x_2, x_3$ from $P$
6: perform all comparisons between $x_1, x_2, x_3$
7: **if** there is a permutation $\pi \in S_3$ such that $x_{\pi(1)} \succ x_{\pi(2)} \succ x_{\pi(3)}$ **then**
8: **for** $i = 1, 2, 3$ **do**
9: $P_i \leftarrow P_i \cup \{ x_{\pi(i)} \}$
10: **end for**
11: **end if**
12: $P' \leftarrow P - (P_1 \cup P_2 \cup P_3)$
13: **end for**
14: **end while**
15: **for all** $x \in P'$ **do**
16: $\ell \leftarrow$**getLevel**$(x)$
17: $P_\ell \leftarrow P_\ell \cup \{ x \}$
18: **end for**
19: given the subdivision of $P$ into its three layers $P_1, P_2, P_3$, determine a $t$-top set $T$ by first drawing elements $P_1$, then possibly from $P_2$, and finally, if necessary, from $P_3$
20: **return** $T$

**getLevel**$(x)$

1: choose u.a.r. with replacement $q = \lceil 7.5 \ln n \rceil$ elements $y_1, y_2, \ldots, y_q$ from the poset
2: **if** $x$ is comparable with at least one $y_i$ **then**
3: **return** 1, 3 or 2 depending on whether $x$ is maximal, minimal, or neither in $\{x, y_1, \ldots, y_q\}$
4: **else**
5: **fail**
6: **end if**

The following lemma can be shown using Chernoff's bounds, with the observation that during the execution of the algorithm, every time we choose $x_1, x_2, x_3$, such triple of elements with constant probability will be a chain.

**Lemma 78.** *The probability that the Algorithm 5.3 fails is at most $O(1/n)$.*

It is an easy observation that we can modify Algorithm 5.3 to select *minimal* elements, hence we can use it when $t \geq \left(\frac{3}{4} + \varepsilon\right) n$ by selecting $n - t$ minimal elements and giving in output the rest of the poset.

We point out that the algorithm can fail in two different ways: unconsciously, if the given poset was not good, or consciously, if the poset was good but not enough maximal elements were found.

### 5.4.1 An $O(n)$ algorithm for every $t$

We just presented fast algorithms for small $t$ (less than $\left(\frac{1}{4} - \varepsilon\right) n$) and large $t$ (more than $\left(\frac{3}{4} + \varepsilon\right) n$). For the remaining cases, we provide an alternative linear algorithm.

Consider Algorithm 5.4: it tries to classify the elements of an (assumed good) poset into its three levels ($P_1$ should contain maximal elements, $P_3$ minimal elements, and $P_2$ the elements in the intermediate layer).

Let us first observe that if we run Algorithm 5.4 on a good poset $P$ the elements put in the sets $P_1$, $P_2$ and $P_3$ are classified correctly after the loop at line 3. Moreover

**Lemma 79.** *In a good poset, the loop at line 3 performs $O(n)$ comparisons with probability $1 - O(1/n)$.*

*Proof.* Consider a single execution of the loop at line 4, and let $X$ be the random variable denoting the number of times the condition at line 7 was true. For any given $x_1$, the probability to extract a chain

involving $x_1$ is at least $(1/8 - o(1))^2 \geq 1/64 - o(1)$; hence $\mu = E[X] \geq |P'|/65$. Chernoff's bound, if $|P'| \geq 524 \ln n$, implies

$$P\left[X < \frac{|P'|}{130}\right] \leq P[X < \mu/2] \leq e^{-\mu/8} \leq e^{-|P'|/520} = O(n^{-\rho})$$

for some $\rho > 1$. Letting $P'_0, P'_1, \ldots$ be the cardinality of $P'$ after 0, 1, ... executions of the loop at line 3, we have $|P'_0| = n$ and for each $i > 0$ the size reduction $|P'_i| \leq 129|P'_{i-1}|/130$ happens with probability $1 - O(n^{-\rho})$.

Notice that as long as the loop is repeated for a number of times that is $o(n^{\rho-1})$, the probability that the size reduction does not happen at some step is bounded by $O(1/n)$. Since after $k = O(\log n)$ steps $|P'_k| < 524 \ln n$, we conclude that with probability $1 - O(1/n)$ the loop is exhausted having performed a size reduction at each step, which also imply that the overall number of comparisons performed will be at most

$$\sum_{i=0}^{k-1} 3|P'_i| \leq \sum_{i=0}^{k-1} 3 \left(\frac{129}{130}\right)^i n = O(n).$$

We are now ready to prove correctness of Algorithm 5.4:

**Theorem 80.** *With probability $1-O(1/n)$, Algorithm 5.4 finds a $t$-top set performing $O(n)$ comparisons.*

*Proof.* As before, we will just show that $P[\xi_{\text{fail}} \mid \xi_{\text{good}}] = O(1/n)$, which is enough to obtain the result. In the case of good posets, by Lemma 79 during the loop at line 3 $O(n)$ comparisons are performed with probability $1 - O(1/n)$. In the second loop (at line 15) no more than $O(\log^2 n)$ comparisons are performed. In this part of the algorithm, though, it may happen that some element is misclassified: the probability that $x$ is non-maximal (non-minimal) and still no element $y$ that $x$ covers (covered by $x$) is found is at most

$$\left(1 - \frac{1}{8} + O(n^{-1/8})\right)^{\lceil 7.5 \ln n \rceil} \leq n^{7.5 \ln\left(\frac{7}{8} + O(n^{-1/8})\right)} \leq O\left(n^{-1.001}\right).$$

So, once more, with probability $1 - O(1/n)$, all elements of $P'$ are correctly classified at the end of the second loop.

### 5.4.2 A probabilistic lower bound and a proof of optimality

In this section, we will prove that no algorithm that fails with small probability can perform less than $\Omega(\min(t, n-t) + \log n)$ queries, for $0 < t < n$ (this assumption is necessary, because for $t = 0$ and $t = n$ zero queries are sufficient). As a consequence, we shall obtain that our probabilistic algorithms can be merged to obtain an asymptotically optimal one.

We say that a probabilistic algorithm is *successful* if, for some $\eta > 0$, its failure probability is at most $O(n^{-\eta})$. We will show that no successful algorithm can perform less than $\Omega(\min(t, n-t) + \log n)$ queries.

Our first lemma states that, for a probabilistic algorithm to be successful, it must perform $\Omega(\min(t, n-t))$ queries; otherwise, indeed, the failure probability does not even go to zero:

**Lemma 81.** *Suppose that a probabilistic algorithm is given that never performs more than $q(n,t)$ queries. If $q(n,t) = o(\min(t, n-t))$ and $0 < t < n$ then with nonvanishing probability the algorithm fails to return a $t$-top set when the input is chosen u.a.r. among the posets of cardinality $n$.*

*Proof.* We can restrict without loss of generality to good posets, since they are a nonvanishing fraction of all posets.

Let us call "compared" an element that has been used in a comparison at least once during the execution of the algorithm. As $q(n,t)$ is $o(n)$, $o(t)$ and $o(n-t)$, we have that the compared elements are at most

- a $o(1)$ fraction of the elements in the output,
- a $o(1)$ fraction of the elements not in the output,

- a $o(1)$ fraction of the elements in the poset level $L_i$, for each $i = 1, 2, 3$.

Now, for a given execution of the algorithm on an input poset of $n$ elements, let $\mathscr{P}$ be the class of good posets of $n$ elements that are compatible with the answers provided to the algorithm. The third property above says that, for every uncompared element $u$, the number of posets in $\mathscr{P}$ where $u \in L_1$ ($u \in L_2$, $u \in L_3$, respectively) is $1/4 + o(1)$ ($1/2 + o(1)$, $1/4 + o(1)$, respectively) of the total.

The assumption $0 < t < n$ implies that there is some uncompared element $x$ in the output, and some uncompared element $y$ not in the output. But now

- if $t < (3/4 - \varepsilon)n$, the output is wrong for all posets in $\mathscr{P}$ where $x$ is in $L_3$;
- if $t > (1/4 + \varepsilon)n$, the output is wrong for all posets in $\mathscr{P}$ where $y$ is in $L_1$.

Each of the two latter events ($x \in L_3$ and $y \in L_1$) has probability at least $1/4 + o(1)$, so in both cases the algorithm would fail with probability at least $1/4 + o(1)$.

We need something more than this, though: we need to prove that, if just $o(\log n)$ queries are performed, then the probability that the algorithm fails is still too high. To show this result, we first state a combinatorial lemma concerning the number of good posets with prescribed sets of elements in their middle layer: for a given set $X$, let $\mathscr{M}_X$ be the set of good posets of $n$ elements in which $X$ is contained in the middle layer.

**Lemma 82.** *Let $A \subseteq \{0, 1, \ldots, n-1\}$ and $x \in \{0, 1, \ldots, n-1\} \setminus A$; if $a = |A| = o(n)$ then*

$$\frac{|\mathscr{M}_{A \cup \{x\}}|}{|\mathscr{M}_A|} = \frac{1}{2} + o(1).$$

*Proof.* Let us first restrict our attention to the part of $\mathscr{M}_A$ with prescribed cardinalities $\ell_1, \ell_2, \ell_3$ of the three layers (of course, $\ell_2 \geq a$). We can divide this set into classes, according to the three sets of elements $L_1, L_2, L_3$ that fall in each layer; these classes will all have the same cardinality, by a simple relabelling argument, so we can limit ourselves to studying how many of these classes will be such that $x \in L_2$. This is equivalent to counting the number of ways of dividing an $n - a - 1$ elements set into three parts of cardinalities $\ell_1, \ell_2 - a - 1, \ell_3$ over the number of ways of dividing an $n - a$ elements set into three parts of cardinality $\ell_1, \ell_2 - a, \ell_3$, so:

$$\frac{|\mathscr{M}_{A \cup \{x\}}|}{|\mathscr{M}_A|} = \binom{n-a-1}{\ell_1, \ell_2 - a - 1, \ell_3} \Big/ \binom{n-a}{\ell_1, \ell_2 - a, \ell_3} = \frac{\ell_2 - a}{n - a} = \frac{1}{2} + o(1),$$

recalling that in a good poset $\ell_2 = n/2 + o(n)$, and $a = o(n)$ by hypothesis.

We can now prove our second probabilistic lower bound:

**Lemma 83.** *Suppose that a probabilistic algorithm is given that never performs more than $q(n, t)$ queries. If $q(n, t) = o(\log n)$ and $0 < t < n$ then, for all $\eta > 0$, with probability $\omega(n^{-\eta})$ the algorithm fails to return a $t$-top set when the input is chosen u.a.r. among the posets of cardinality $n$.*

*Proof.* Let $x_1, x_2, \ldots, x_k$, with $k = o(\log n)$, be the elements queried by the algorithm, in the order in which they are first queried; let $\xi_{\text{good}}$ be the event that the poset is good, and $\xi_i$ be the event that the poset is good and $x_i \in L_2$.

We start by showing that all the different elements queried by the algorithm will belong to the middle layer $L_2$ with nonnegligible probability, that is,

$$P[\xi_1 \cap \cdots \cap \xi_k] = \omega(n^{-\eta}), \quad \forall \eta > 0. \tag{5.1}$$

By the chain rule we get,

$$\begin{aligned} P[\xi_1 \cap \cdots \cap \xi_k] &= P[\xi_{\text{good}}] P[\xi_k \cap \xi_{k-1} \cap \ldots \cap \xi_1 \mid \xi_{\text{good}}] = \\ &= P[\xi_{\text{good}}] \prod_{i=1}^{k} P[\xi_i \mid \xi_{i-1} \cap \ldots \cap \xi_1 \cap \xi_{\text{good}}]. \end{aligned}$$

By Lemma 82, $P[\xi_i \mid \xi_{i-1} \cap \ldots \cap \xi_1 \cap \xi_{\text{good}}] = 1/2 + o(1)$, as long as $i = o(n)$. Since $P[\xi_{\text{good}}] = 1 - O(1/n)$, we have, for all $\eta > 0$,

$$P[\xi_1 \cap \cdots \cap \xi_k] = \left(1 - O\left(\frac{1}{n}\right)\right)\left(\frac{1}{2} + o(1)\right)^{o(\log n)} = \omega(n^{-\eta}).$$

Now, to obtain the contradiction, consider two cases:

- if $t < (3/4 - \varepsilon)n$, we can assume that at least one output element $x$ has not been compared (because otherwise with the above probability the output contains only elements from the middle layer, missing the elements from the top layer);
- if $t > (1/4 + \varepsilon)n$, we can assume that at least one element $y$ not in the output has not been compared (because otherwise the output contains all elements from the top and bottom layers, but misses some elements from the middle layer).

In both cases, we can proceed as in the last part of the proof of Lemma 81.

It is now straightforward to obtain the following theorem:

**Theorem 84.** *Every successful algorithm performs at least $\Omega(\min(t, n-t) + \log n)$ queries to return a $t$-top set ($0 < t < n$) of a poset of order $n$ chosen uniformly at random.*

Now, we can merge Algorithm 5.3 and Algorithm 5.4 as follows:

- if $0 < t < \big(1/4 - \varepsilon\big)n$ we apply Algorithm 5.3, performing $O(t + \log n)$ queries;
- if $\big(1/4 - \varepsilon\big)n \le t \le \big(3/4 + \varepsilon\big)n$ we apply Algorithm 5.4, performing $O(n)$ queries;
- if $\big(3/4 + \varepsilon\big)n < t < n$ we apply a modified version of Algorithm 5.3 that searches for $n - t$ minimal elements, and returns the top set containing all the elements but those found; in this case, we perform $O((n-t) + \log n)$ queries.

The algorithm obtained in this way performs $O(\min(t, n-t) + \log n)$ queries, so it is optimal in the class of successful algorithms (i.e., it is $\theta$-slow with respect to the optimal algorithm in the class, for some suitable constant $\theta$).

## 5.5 Preorders

All our algorithms can be extended to work on preorders, as we informally show in this section. A *preorder* is a set endowed with a reflexive, transitive relation $\preceq$ (which however might fail to be antisymmetric). In a preorder, if $x \preceq y$ and $y \preceq x$ we say that $x$ and $y$ are *equivalent* and write $x \sim y$.

Suppose we take a picture of the same scene twice, to increase the probability that at least one is not blurred, and that both copies turn out to be perfect shots. In this case, $x \sim y$, as we can clearly compare the two pictures, and we have no preference about one or the other. This situation is very different from the sea/desert dilemma, in which we are not able to express an opinion about which picture is better: in the case of the two shots, we want just one of the two pictures, whereas in the incomparability case we would like to keep both.

There are at least two possible ways to extend the notion of $t$-top sets to preorders. One could just search for a $t$-top set of the poset $P/\sim$ obtained by quotienting the original preorder $P$ with respect to the equivalence relation $\sim$. In this case, the output of our algorithms should be a set of equivalence classes. Alternatively, as a perhaps more sensible solution, we might return an arbitrarily chosen representative for each equivalence class.

To treat preorders, we add a new possible answer to queries (denoted, of course, by $x \sim y$). We change the first two algorithms in the following way: instead of handling antichains, we manipulate quasi-antichains, that is, sets of $\sim$-equivalent classes that are pairwise *incomparable* (i.e., any two elements from distinct classes are incomparable, and any two elements within the same class are $\sim$-equivalent). When the algorithm wants the query $X :: Y$ to be answered (where $X$ and $Y$ are $\sim$-equivalence classes), we choose arbitrarily an element in $x \in X$ and an element $y \in Y$ and we pass the query $x :: y$ to the oracle.

If the result of the query is $x \sim y$, we just merge the two classes $X, Y$, adding a class $X \cup Y$ and removing both $X$ and $Y$. Otherwise, we let the algorithm proceed as before.

It is easy to check that this strategy works and that it does not increase the running time of the algorithms. The deterministic lower bounds we gave trivially hold for preorders as well, since each partial order is also a preorder.

### 5.5.1 Preorders in the probabilistic setting

To handle preorders in the probabilistic setting, we use Marcel Erné's results [48, page 253, lemma 4.6] stating that the ratio between the number of preorders of order $n$ (therein labelled $A(n)$) and the number of posets of order $n$ ($A_0(n)$) is

$$\frac{A(n)}{A_0(n)} = 1 + 2^{-n/2+O(\sqrt{n}\log n)} \leq \frac{1}{1-O(1/n)}.$$

As a consequence, almost all preorders are good posets; more precisely, since the ratio between the numbers of posets and preorders is at least $1 - O(1/n)$, and at least $1 - O(1/n)$ of these posets are good, the fraction of preorders that turn out to be good posets is at least $(1 - O(1/n))^2 = 1 - O(1/n)$.

This observation is sufficient to show that both the probabilistic algorithms and the corresponding lower bounds hold for preorders as well.

## 5.6 Conclusions

We have introduced the problem of finding a set of $t$ best elements (that we called top set) from a poset of $n$ elements. The first two solutions we have provided are deterministic: one requires $O(tn)$ comparisons, while the other needs $O(wn + wt\log(n/w))$ (where $w$ is the width of the poset).

In general, the former algorithm is 2-slow (meaning that, for all posets, it does not pose more than twice the number of queries that the best algorithm would pose in the worst case).

For posets of limited width $w$, under the assumption that $t, w$ are not too large, we showed an inherent difference between the cases $t < w$ and $t \geq w$: if $t < w = o(n)$, the first algorithm is $(4+\varepsilon)$-slow, whereas if $o(n/\log n) = t \geq w$ the second algorithm is $(3+\varepsilon)$-slow (here, $\varepsilon = \varepsilon(n)$ and $\lim_{n\to\infty} \varepsilon(n) = 0$).

Then, to beat the lower bounds used to determine the slowness of the two deterministic algorithms, we assumed that the input poset was chosen uniformly at random between all posets of size $n$. Under this assumption, we were able to obtain a probabilistic algorithm capable of determining $t$-top sets in time $O(\min(t, n-t) + \log n)$ with probability $1 - O(1/n)$. This probabilistic algorithm happens to be $O(1)$-slow in the class of all algorithms that choose their input u.a.r. and fail with at most an inverse polynomial probability (that is, $O(n^{-\eta})$ for some $\eta > 0$). In other words, our algorithm is asymptotically optimal under the stated success probability.

Finally, we considered preorders instead of just posets. A top set of a preorder is a top set of the poset obtained by quotienting the preorder with respect to its equivalence relation. We showed that all our upper and lower bounds hold also in the case of preorders.

# Bibliography


[1] M. Adler and M. Mitzenmacher. Towards compressing web graphs. In *Data Compression Conference*, pages 203–212, 2001.

[2] W. Aiello, F. Chung, and L. Lu. Random evolution in massive graphs. In *Proc. 42nd IEEE Symposium on Foundations of Computer Science*, pages 510–519, 2001.

[3] R. Albert and A.-L. Barabasi. Emergence of scaling in random networks. *Science*, pages 509–512, 1999.

[4] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit. Local search heuristics for k-median and facility location problems. *SIAM J. Comput.*, 33(3):544–562, 2004.

[5] A. L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 1999.

[6] J. Bartholdi, C. A. Tovey, , and M. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice Welfare*, 6, 2, 1989.

[7] J. D. Batson, D. A. Spielman, and N. Srivastava. Twice-ramanujan sparsifiers. In *Proceedings of STOC*, 2009.

[8] P. Berenbrink, R. Elsässer, and T. Friedetzky. Efficient randomized broadcasting in random regular networks with appliations in peer-to-peer systems. In *Proceedings of PODC*, 2008.

[9] N. Berger, C. Borgs, J. T. Chayes, and A. Saberi. On the spread of viruses on the internet. In *SODA*, 2005.

[10] P. Boldi, F. Chierichetti, and S. Vigna. Pictures from mongolia - extracting the top elements from a partially ordered set. *Theory of Computing Systems*, 44(2), 2009.

[11] P. Boldi, M. Santini, and S. Vigna. Permuting web graphs. In *Proc. of WAW 2009*, 2009.

[12] P. Boldi and S. Vigna. The webgraph framework I: Compression techniques. In *Proc. 13th International World Wide Web Conference*, pages 595–601, 2004.

[13] P. Boldi and S. Vigna. Codes for the world-wide web. *Internet Mathematics*, 2(4):405–427, 2005.

[14] B. Bollobás, O. Riordan, J. Spencer, and G. E. Tusnády. The degree sequence of a scale-free random graph process. *Random Structures and Algorithms*, 18(3):279–290, 2001.

[15] C. Borgs, J. T. Chayes, C. Daskalakis, and S. Roch. First to market is not everything: An analysis of preferential attachment with fitness. In *Proc. 39th Annual ACM Symposium on Theory of Computing*, pages 135–144, 2007.

[16] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

[17] S. P. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Gossip algorithms: design, analysis and applications. *IEEE Transactions on Information Theory*, 52, 2006.

[18] A. Broder. On the resemblance and containment of documents. In *SEQUENCES '97: Proceedings of the Compression and Complexity of Sequences 1997*. IEEE Computer Society, 1997.

[19] A. Z. Broder, M. Charikar, A. M. Frieze, and M. Mitzenmacher. Min-wise independent permutations. *JCSS*, 60:630–659, 2000.

[20] A. Z. Broder, S. R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. In *Proc. 9th International World Wide Web Conference*, pages 309–320, 2000.

[21] G. Buehrer and K. Chellapilla. A scalable pattern mining approach to web graph compression with communities. In *Proc. 1st International Conference on Web Search and Data Mining*, pages 95–106, 2008.

[22] J. Carlson and J. Doyle. Highly optimized tolerance: A mechanism for power laws in designed systems. *Phys. Rev. E*, 60:1412, 1999.

[23] M. Charikar. Similarity estimation techniques from rounding algorithms. In *Proc. of ACM STOC 2002*, 2002.

[24] M. Charikar, S. Guha, E. Tardos, and D. B. Shmoys. A constant-factor approximation algorithm for the $k$-median problem. *JCSS*, 65(1):129–149, 2002.

[25] J. Cheeger. A lower bound for smallest eigenvalue of laplacian. *Problems in Analysis*, 1970.

[26] F. Chierichetti, R. Kumar, S. Lattanzi, A. Panconesi, and P. Raghavan. Models for the compressible web. In *FOCS*, 2009.

[27] F. Chierichetti, R. Kumar, M. Mitzenmacher, A. Panconesi, P. Raghavan, and S. Lattanzi. On compressing social networks. In *KDD*, 2009.

[28] F. Chierichetti, R. Kumar, S. Pandey, and S. Vassilvitskii. Finding the jaccard median. In *SODA*, 2010.

[29] F. Chierichetti, R. Kumar, and S. Vassilvitskii. Similarity caching. In *PODS*, 2009.

[30] F. Chierichetti, S. Lattanzi, and A. Panconesi. Rumor spreading in social networks. In *ICALP*, 2009.

[31] F. Chierichetti, S. Lattanzi, and A. Panconesi. Rumor spreading and graph conductance. In *SODA*, 2010.

[32] H. T. Chou and D. J. DeWitt. An evaluation of buffer management strategies for relational database systems. *Algorithmica*, 1(3):311–336, 1986.

[33] F. R. K. Chung. Spectral graph theory. In *CBMS Regional Conference Series in Mathematics*. American Mathematical Society, 1997.

[34] C. Cooper and A. M. Frieze. A general model of web graphs. *Random Structures and Algorithms*, 22(3):311–335, 2003.

[35] C. Cooper and A. M. Frieze. The cover time of the preferential attachment graph. *Journal of Combinatorial Theory, Ser. B*, 97(2):269–290, 2007.

[36] T. Cowan and P. Fishburn. Foundations of preference. In *Essays in Honor of Werner Leinfellner*, pages 261–271. D. Reidel, Dordrecht, 1988.

[37] I. Dagan. Contextual word similarity. In R. Dale, H. Moisl, and H. Somers, editors, *A Handbook of Natural Language Processing: Techniques and Applications for the Processing of Language as Text*. Marcel Dekker Inc., 2000.

[38] C. Daskalakis, R. M. Karp, E. Mossel, S. Riesenfeld, and E. Verbin. Sorting and selection in posets. In *SODA*, 2009.

[39] C. de la Higuera and F. Casacuberta. Topology of strings: Median string is np-complete. *Theoretical Computer Science*, 230:39 – 48, 2000.

[40] A. J. Demers, D. H. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. E. Sturgis, D. C. Swinehart, and D. B. Terry. Epidemic algorithms for replicated database maintenance. In *Proceedings of PODC*, 1987.

[41] M. Dodds and Watts. An experimental study of search in global social networks. *Science*, 301(5634):827–829, 2003.

[42] B. Doerr, T. Friedrich, and T. Sauerwald. Quasirandom broadcasting. In *Proceedings of SODA*, 2008.

[43] B. Doerr, T. Friedrich, and T. Sauerwald. Quasirandom rumor spreading: Expanders, push vs. pull, and robustness. In *Proceedings of ICALP*, 2009.

[44] D. Dubhashi and A. Panconesi. *Concentration of Measure for the Analysis of Randomised Algorithms*. Cambridge University Press, 2009.

[45] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 613–622, New York, NY, USA, 2001. ACM Press.

[46] W. Effelsberg and T. Härder. Principles of database buffer management. *ACM Transactions on Database Systems*, 9(4):560–595, 1984.

[47] R. Elsässer. On the communication complexity of randomized broadcasting in random-like graphs. In *Proceedings of SPAA*, 2006.

[48] M. Erné. Struktur- und Anzahlformeln für Topologien auf endlichen Mengen. *Manuscripta Mathematica*, 11:221–259, 1974.

[49] A. Fabrikant, E. Koutsoupias, and C. H. Papadimitriou. Heuristically optimized trade-offs: A new paradigm for power laws in the internet. In *Proc. 29th International Colloquium on Automata, Languages and Programming*, pages 110–122, 2002.

[50] R. Fagin, R. Kumar, and D. Sivakumar. Efficient similarity search and classification via rank aggregation. In *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 301–312, New York, NY, USA, 2003. ACM Press.

[51] U. Faigle and G. Turán. Sorting and recognition problems for ordered sets. *SIAM J. Comput.*, 17(1):100–113, 1988.

[52] F. Falchi, C. Lucchese, S. Orlando, R. Perego, and F. Rabitti. A Metric Cache for Similarity Search. In *Proc. 6th Workshop on Large-Scale Distributed Systems for Information Retrieval*, 2008.

[53] U. Feige, D. Peleg, P. Raghavan, and E. Upfal. Randomized broadcast in networks. *Algorithms*, 1, 1990.

[54] T. Friedrich and T. Sauerwald. Near-perfect load balancing by randomized rounding. In *Proceedings of STOC*, 2009.

[55] A. Frieze and G. Grimmett. The shortest-path problem for graphs with random arc-lengths. *Algorithms*, 10, 1985.

[56] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[57] G. Grefenstette. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers, Norwell, MA, USA, 1994.

[58] R. A. M. Gregson. *Psychometrics of Similarity*. Academic Press, 1975.

[59] B. Huntley and H. J. B. Birks. The past and present vegetation of the morrone birkwoods national nature reserve. *Journal of Ecology*, 67, 1979.

[60] P. Jaccard. Distribution de la flore alpine dans le bassin des dranses et dans quelques régions voisines. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:241–272, 1901.

[61] D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall, second edition, February 2008.

[62] C. Karande, K. Chellapilla, and R. Andersen. Speeding up algorithms on compressed web graphs. In *Proc. 2nd International Conference on Web Search and Data Mining*, pages 272–281, 2009.

[63] R. Karp, C. Schindelhauer, S. Shenker, and B. Vöcking. Randomized rumor spreading. In *Proceedings of FOCS*, 2000.

[64] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *Proceedings of FOCS*, 2003.

[65] B. Kendrick. Quantitative characters in computer taxonomy. *Phenetic and Phylogenetic Classification*, 1964.

[66] J. Kleinberg. Navigation in a small world. *Nature*, 406:845, 2000.

[67] J. Kleinberg. The small-world phenomenon: An algorithmic perspective. In *Proc. 37th Annual ACM Symposium on Theory of Computing*, pages 163–170, 2000.

[68] D. Kleitman and B. L. Rothschild. Asymptotic enumeration of partial orders on a finite set. *Trans. of the American Mathematical Society*, 205:205–220, 1975.

[69] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal. Stochastic models for the web graph. In *Proc. 41st IEEE Symposium on Foundations of Computer Science*, pages 57–65, 2000.

[70] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the web for emerging cybercommunities. In *Proc. 8th International World Wide Web Conference*, pages 403–416, 1999.

[71] H. R. Lasked. Light dependent activity patterns among reef corals: Montastrea cavernosa. *Biological Bullettin*, 156, 1979.

[72] S. Lattanzi and D. Sivakumar. Affiliation networks. In *Proc. 41st ACM Symposium on Theory of Computing*, 2009.

[73] R. Lempel and S. Moran. Predictive caching and prefetching of query results in search engines. In *Proc. 12th International Conference on World Wide Web*, pages 19–28, 2003.

[74] J. Leskovec, D. Chakrabarti, J. Kleinberg, and C. Faloutsos. Realistic, mathematically tractable graph generation and evolution, using kronecker multiplication. In *European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 133–145, 2005.

[75] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: Densification laws, shrinking diameters, and possible explanations. In *Proc. 11th Conference on Knowledge Discovery and Data Mining*, pages 177–187, 2005.

[76] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Statistical properties of community structure in large social and information networks. In *WWW*, 2008.

[77] D. Liben-Nowell, J. Novak, R. Kumar, P. Raghavan, and A. Tomkins. Geographic routing in social networks. *Proc. National Academy of Sciences*, 102(33):11623–11628, 2005.

[78] A. Lipkus. A proof of the triangle inequality for the tanimoto distance. *Journal of Mathematical Chemistry*, 26, 1999.

[79] B. Mandelbrot. An informational theory of the statistical structure of languages. In W. Jackson, editor, *Communication Theory*, pages 486–502. Butterworth, 1953.

[80] E. Marczewski and H. Steinhaus. On a certain distance of sets and the corresponding distance of functions. *Colloquium Mathematicum*, 6, 1958.

[81] M. Mihail, C. H. Papadimitriou, and A. Saberi. On certain connectivity properties of the internet topology. *J. Comput. Syst. Sci.*, 72(2):239–251, 2006.

[82] M. Mitzenmacher. A brief history of generative models for power law and lognormal distributions. *Internet Mathematics*, 1(2), 2003.

[83] D. Mosk-Aoyama and D. Shah. Fast distributed algorithms for computing separable functions. *IEEE Transactions on Information Theory*, 54, 2008.

[84] L. Mutafchiev. The largest tree in certain models of random forests. *Random Structures and Algorithms*, 13(3-4):211–228, 1998.

[85] F. Nicolas and E. Rivals. Complexities of the centre and median string problems. In *Proc. of CPM 2003*, 2003.

[86] D. N. Osherson, J. Stern, O. Wilkie, M. Stob, and E. E. Smith. Default probability. *Cognitive Science*, 15, 1991.

[87] S. Pandey, A. Z. Broder, F. Chierichetti, V. Josifovski, R. Kumar, and S. Vassilvitskii. Nearest-neighbor caching for content-match applications. In *Proc. of WWW 2009*, 2009.

[88] M. E. Patzkowsky and S. M. Holland. Biofacies replacement in a sequence stratigraphic framework; middle and upper ordovician of the nashville dome, tennessee, usa. *Palaios*, 14, 1999.

[89] B. Pittel. On spreading a rumor. *SIAM Journal on Applied Mathematics*, 47, 1987.

[90] I. C. Prentice. Non-metric ordination methods in ecology. *Journal of Ecology*, 65, 1977.

[91] G. M. Sacco and M. Schkolnick. Buffer management in relational database systems. *ACM Transactions on Database Systems*, 11(4):473–498, 1986.

[92] T. Sauerwald. On mixing and edge expansion properties in randomized broadcasting. In *Proceedings of ISAAC*, 2007.

[93] J. J. Sepkoski. Quantified coefficients of association and measurement of similarity. *Mathematical Geology*, 6, 1974.

[94] A. Silberschatz and P. B. Galvin. *Operating System Concepts*. John Wiley & Sons, Inc., New York, NY, USA, 2000.

[95] H. Simon. On a class of skew distribution functions. *Biometrika*, 42:425–440, 1955.

[96] D. D. Sleator and R. E. Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28:202–208, 1985.

[97] H. Späth. The minisum location problem for the jaccard metric. *OR Spektrum*, 3:91–94, 1981.

[98] D. A. Spielman and N. Srivastava. Graph sparsification by effective resistances. In *Proceedings of STOC*, 2008.

[99] D. A. Spielman and S.-H. Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of STOC*, 2004.

[100] M. Stonebraker. Operating system support for database management. *Communications of the ACM*, 24(7):412–418, 1981.

[101] T. Suel and J. Yuan. Compressing the graph structure of the web. In *Data Compression Conference*, pages 213–222, 2001.

[102] J. Szymanski. On the complexity of algorithms on recursive trees. *Theoretical Computer Science*, 74(3):355–361, 1990.

[103] J. Travers and S. Milgram. An experimental study of the small world problem. *Sociometry*, 32(4):425–443, 1969.

[104] A. Tversky. Features of similarity. *Psychological Review*, 84, 1977.

[105] G. A. Watson. An algorithm for the single facility location problem using the jaccard metric. *SIAM Journal on Science and Statistical Computing*, 4:748–756, 1983.

[106] D. Watts and S. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):409–410, 1998.

[107] I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kauffman Publishers, 2 edition, 1999.

[108] G. K. Zipf. *Human Behavior and the Principle of Least Effort*. Addison-Wesley, 1949.