The Complexity of LSH Feasibility

Flavio Chierichetti[a,1], Ravi Kumar[b,2], Mohammad Mahdian[b,2]

[a]*Sapienza University, Dipartimento di Informatica, Rome, Italy.*
[b]*Google, Mountain View, CA, USA.*

Abstract

In this paper we study the complexity of the following feasibility problem: given an $n \times n$ similarity matrix S as input, is there a locality sensitive hash (LSH) for S? We show that the LSH feasibility problem is NP-hard even in the following strong promise version: either S admits an LSH or S is at ℓ_1-distance at least $n^{2-\epsilon}$ from every similarity that admits an LSH. We complement this hardness result by providing an $\tilde{O}(3^n)$ algorithm for the LSH feasibility problem, which improves upon the naïve $n^{\Theta(n)}$ time algorithm; we prove that this running time is tight, modulo constants, under the Exponential Time Hypothesis.

1. Introduction

Locality Sensitive Hashing (LSH) has become as an indispensable tool in the algorithmicist's toolkit. Given a universe of objects and a similarity function between two objects, the goal is to succinctly represent each object in a way that the similarity between two objects can be efficiently approximated by merely using their respective representations [1, 2]; we assume that the similarity values lie in $[0, 1]$. A popular way to construct an LSH for a similarity function is to carefully pick a hash function family so that each object is represented by a random hash value and the probability that the hashes for two objects collide is precisely their similarity. LSH has a variety of applications in massive data processing, including in algorithms for the data stream and sketching models.

Email addresses: flavio@di.uniroma1.it (Flavio Chierichetti), ravi.k53@gmail.com (Ravi Kumar), mahdian@gmail.com (Mohammad Mahdian)

[1]Corresponding author. This work was done while the author was at Cornell University, supported by the NSF grant CCF-0910940.

[2]Part of the work was done while the author was at Yahoo! Research.

Much of the past work on LSH has been on expanding and understanding the family of similarities for which an LSH can be constructed. For example, Broder et al. [3–5] obtained an LSH for Jaccard similarity between two sets, using min-wise independent permutations; this was recently generalized to the entire family of rational set similarities [6]. Charikar [7] obtained an LSH for the cosine similarity for vectors in the Euclidean space, using random projections, and an LSH for the earth-mover's distance between strings, using LP rounding. Datar et al. [8] obtained an LSH for p-norms in Euclidean spaces, using p-stable distributions. See the survey by Andoni and Indyk [9] and Andoni's thesis [10].

LSH feasibility problem. A symmetric similarity function over a universe U *admits a LSH* (or is *LSHable*) iff there exists a probability distribution on the hash functions operating on U that guarantees that, for every two elements of U, the probability that they get hashed to the same value is equal to their similarity (formal definition in Section 2). The general problem of finding necessary conditions for a similarity to be LSHable has been addressed in [6, 7].

In particular, the family of LSHable similarities was shown to be closed under transformations by probability generating functions [6]. Those works were motivated by the following basic question: is it possible to find a property that characterizes LSHable similarities and that can be computed efficiently? In other words, given a similarity, can we determine if it is LSHable? We call this the *LSH feasibility* problem and study here its computational complexity.

A practical motivation for our work stems from a large body of entity-matching applications in large-scale information retrieval. In this setting, we have a collection of entities (e.g., restaurants) obtained from a variety of web sources (e.g., `yelp.com`, `zagat.com`, `opentable.com`); note that this collection could inadvertently contain duplicate entries. We also have a function that gives a similarity score between a pair of entities (e.g., a hand-built function that takes into account the edit distance between the restaurant names, the geographic distance between their locations, the Hamming distance between their phone numbers, etc. [11]). This similarity function is designed to reflect the human interpretation of when two entities are possibly the same; often, it is machine-learned through labeled examples and feature extraction. Since the number of entities is typically huge, naïve methods are infeasible for the duplicate-detection problem and LSH is an obvious candidate of choice. Such an approach is fundamentally based on the ability to obtain an LSH for the machine-learned similarity or determin-

ing that it is not possible to do so. See [12] for a similar scenario that arises in Computational Biology.

Such feasibility problems have been studied in related contexts. Avis and Deza [13] considered the ℓ_1 embeddability feasibility problem and showed that it is NP-hard under Turing reductions. Edmonds [14] showed that embedding into ℓ_∞^2 is easy whereas into ℓ_∞^3 is NP-complete. Onak [15] considered the problem of testing properties of set of points in metric spaces and focused on testing dimensionality reduction in the ℓ_1 and ℓ_∞ metrics; see also the earlier work of Krauthgamer and Sasson [16]. The problem of computing a low-distortion embedding between two metric spaces has been addressed as well [17–20]; see also the thesis of Sidiropoulos [21].

JPM feasibility problem. Another instance of a feasibility problem related to LSH is the following: given a non-negative matrix of joint probabilities, is the matrix realizable by events such that the probability of two events occurring together is given by the corresponding matrix entry? We call this the *Joint Probability Matrices (JPM) feasibility* problem.[3] The JPM problem (or the equivalent Bernoulli covariance matrix realizability) has been studied under various disguises by the statistics community. A large number of heuristics have been proposed for this problem [22–24] but none of these methods, however, is guaranteed to run in polynomial time. Note that a necessary condition for feasibility is that the covariance matrix has to be symmetric positive semi-definite. Since the collection of correlation matrices forms a convex set, there has been a lot of work on understanding its extreme points; see [25] and the references therein. In [26], the authors give a necessary and sufficient condition for a matrix to be a correlation matrix of a binary random vector. The condition, however, is the feasibility of a set of inequalities of exponential size. If the joint probability matrix is allowed to contain "don't cares," then the feasibility problem was proved to be NP-hard by Koller and Megiddo [27].

Main results. Let S be an $n \times n$ similarity matrix. We show that distinguishing between the case when S is LSHable and when S is at least $n^{2-\epsilon}$ in ℓ_1 distance from any LSHable similarity, is NP-complete. The hardness proof is built in two steps. First, we consider an inverse polynomial gap

[3] Reid Barton asked the following question on March 26, 2010 in the MathOverflow forum (`http://mathoverflow.net/questions/19406/correlation-of-bernoulli-random-variables`): for which correlation matrix can one construct Bernoulli random variables with this correlation? Clearly, this question is related to the JPM problem.

version of the problem and show its hardness by using the hardness of edge-coloring of cubic graphs. In order to prove the correctness of the reduction, we have to show non-LSHability of similarities. For this, we analyze the dual of a natural LP formulation of the LSH feasibility problem. Second, by using a tensor construction, we amplify the hardness gap from inverse polynomial to $n^{2-\epsilon}$.

We complement this hardness result by providing an $\tilde{O}(3^n)$ algorithm for this problem; note that the naïve algorithm runs in time $n^{\Theta(n)}$. Our algorithm is obtained by constructing an efficient separation oracle for the dual of the LP formulation of the problem; the oracle is implemented by a dynamic program. The running time of our algorithm is optimal, modulo constants, assuming the Exponential Time Hypothesis.

Finally, we consider the JPM problem and prove it is NP-hard under Turing reductions, once again by leveraging on the dual of an LP formulation of the problem.

2. Preliminaries

Let U be a universe of n elements. A *similarity function* $S : U^2 \to [0, 1]$ is a symmetric function such that $S(u, v) = 1$ if and only if $u = v$; we will use the terms similarity function and similarity matrix interchangeably. Let $\mathcal{H} = \{h : U \to R\}$ be a maximal family of non-isomorphic (hash) functions over U. In order to include all the non-isomorphic hash functions over U, we impose $|R| \geq |U|$. For simplicity, we write $\mathcal{H} = \{h_1, h_2, \ldots\}$. Given a similarity function S, the *LSH feasibility* problem asks if there is a probability distribution over the hash functions in $\mathcal{H}$ such that for all $u, v \in U$, $\Pr_{h\in\mathcal{H}}[h(u) = h(v)] = S(u, v)$; if this holds, we say that S is *LSHable*. It was shown in [6, 7] that for a similarity matrix S to be LSHable, S has to be positive semi-definite and the distance $\overline{S} = 1 - S$ must admit an isometric embedding into ℓ_1 .

It is easy to see that $|\mathcal{H}|$ is the number of equivalence relations on a set of n distinguishable elements, which is the nth Bell number B_n [28]. It is known that $n \ln n > \ln B_n > n \ln n - O(n \ln \ln n)$ (see [29]).

We observe that $S : U^2 \to [0, 1]$ is LSHable if and only if a specific linear system is feasible. Indeed, consider the matrix M with rows indexed by elements in $\binom{U}{2} \cup \{\star\}$ and columns indexed by hash functions in $\mathcal{H}$. The $\star$-row of M will be all-ones, and $M(\{u, v\}, h_i)$, for $u, v \in U$ and $h_i \in \mathcal{H}$, will equal 1 if $h_i(u) = h_i(v)$, and 0 otherwise. Furthermore, let $S^\star$ be the vector with coordinates indexed by elements in $\binom{U}{2} \cup \{\star\}$ such that $S^\star(\{u, v\}) = S(u, v)$, for each $\{u, v\} \in \binom{U}{2}$, and $S^\star(\star) = 1$. Then, S is

LSHable if and only if there exists a vector p, indexed by the elements in $\mathcal{H}$ that satisfies the *primal*:

$$Mp \quad = \quad S^\star; \qquad\qquad p \quad \geq \quad \mathbf{0}. \tag{1}$$

Note that the solution vector p is a probability distribution over $\mathcal{H}$; thus, this forms an LSH for S.

We observe that the primal system has $\binom{n}{2}+1$ equality constraints, plus the non-negativity constraints on the variables. The number of variables equals $|\mathcal{H}| = B_n$. Since $B_n = n^{\Theta(n)}$, if we try to solve the primal system (1) directly to check the LSH feasibility, we will obtain an algorithm with deterministic running time $n^{\Theta(n)}$. By Farkas's lemma, we also have that S is *not* LSHable if and only if there exists a vector π (with coordinates indexed by the elements in $\binom{U}{2} \cup \{\star\}$) that satisfies the *dual*:

$$\pi M \quad \geq \quad \mathbf{0}; \qquad\qquad \pi S^\star \quad < \quad 0. \tag{2}$$

Given a hash function h, we say that a set $W \subseteq U$ is *h-uniform* if $h(a) = h(b)$ for each $a, b \in W$ and is *h-shattered* if $h(a) \neq h(b)$ for each $a, b \in W$. Furthermore, if W is h-uniform but $W \cup \{a\}$ is not h-uniform $\forall a \in U \setminus W$, we say that W is a *maximal* h-uniform set (or an equivalence class under h).

The *ℓ_1-distance* between two similarities S, S' over the same universe set U is defined as $\ell_1(S, S') = \sum_{\{u,v\} \in \binom{U}{2}} |S(u,v) - S'(u,v)|$.

3. An $\tilde{O}(3^n)$ algorithm for LSH feasibility

In this section we give an $\tilde{O}(3^n)$ algorithm that can determine if an $n \times n$ similarity matrix S is LSHable (for simplicity, we will assume that the values of S can be expressed as a ratio of poly(n)-bit integers.) This algorithm will complement the lower bound in Corollary 9. To obtain the algorithm, we will give a separation oracle that runs in time $\tilde{O}(3^n)$ for the dual problem given by (2). The ellipsoid algorithm [30] can then be used to check LSH feasibility in time $\tilde{O}(3^n)$. The algorithm employs a dynamic program to "construct" the hash function whose equivalence classes have minimum total π-weight (defined in line 7 of Algorithm 1). If this hash function has a total π-weight smaller than $-\pi(\star)$, then its inequality will be violated; otherwise, no hash functions' inequalities will be violated. This observation immediately implies the following.

Proposition 1. *Algorithm 1 returns a violated inequality of the dual system (2), if one exists.*

Algorithm 1 A separation oracle for the dual system (2).

Input: A similarity $S : U^2 \to [0, 1]$ and a vector π indexed by elements in $\binom{U}{2} \cup \{\star\}$.

1: **if** $\sum_{\{u,u'\}\in\binom{U}{2}} (\pi(\{u, u'\}) \cdot S(u, u')) + \pi(\star) \geq 0$ **then**
2: **return** "the $\pi S^* < 0$ inequality is violated."
3: Let V and P be two vectors indexed by all the $2^{|U|}$ subsets of U. (V and P will contain, respectively, the values and the partitions induced by the partial solutions.)
4: Let $V[\varnothing] \leftarrow 0$, and $V[Y] \leftarrow \perp$ for each $\varnothing \neq Y \subseteq U$.
5: Let $P[\varnothing] \leftarrow \varnothing$, and $P[Y] \leftarrow \perp$ for each $\varnothing \neq Y \subseteq U$.
6: **for all** $X \subseteq U$ **do**
7: Let x be the total π-weight of the pairs in X, i.e., $x \leftarrow \sum_{\{u,u'\}\in\binom{X}{2}} \pi(\{u, u'\})$.
8: **for all** $Y \subseteq U \setminus X$ **such that** $V[Y] \neq \perp$ **do**
9: **if** $V[X \cup Y] = \perp$ **or** $x + V[Y] < V[X \cup Y]$ **then**
10: $V[X \cup Y] \leftarrow x + V[Y]$
11: $P[X \cup Y] \leftarrow \{X\} \cup P[Y]$
12: **if** $V[U] + \pi(\star) < 0$ **then**
13: **return** "the inequality of the hash function corresponding to the partition $P[U]$ is violated."
14: **return** "all the inequalities are satisfied."

We now observe that the separation oracle runs in time $\tilde{O}(3^n)$.

Proposition 2. *Algorithm 1 runs in time $\tilde{O}(3^n)$, with $n = |U|$.*

Proof. If $f(n)$ is the total number of executions of the lines inside the loop at line 8, we have that the running time of Algorithm 1 is $O(f(n)\,\mathrm{poly}(n))$. Moreover,

$$f(n) \leq \sum_{X\subseteq U} \sum_{Y\subseteq U\setminus X} 1 = \sum_{X\subseteq U} 2^{|U\setminus X|} = \sum_{X\subseteq U} 2^{n-|X|} = \sum_{i=0}^{n} \binom{n}{i} 2^{n-i} = 3^n. \quad \square$$

In Appendix A, we observe that a separation oracle analogous to Algorithm 1 works for the more general problem of finding the closest LSHable similarity to S.

4. Hardness of LSH feasibility

In this section we show that the LSH feasibility problem is NP-hard. First we show that it is NP-hard to distinguish between the case when a

similarity is LSHable and the case when the closest LSHable similarity is at ℓ_1-distance at least $\Omega(n^{-2})$ from the given similarity. In Section 4.2, we amplify this gap to $n^{2-\epsilon}$.

4.1. Inverse polynomial gap hardness

First note that the problem is in NP. Indeed, system (1) has $O(n^2)$ constraints. Then, by Carathéodory's theorem [31], if the system is feasible, it admits a solution with $O(n^2)$ non-zero variables. A non-deterministic algorithm can first guess the non-zero variables and then solve the polynomial-sized linear system they induce.

We show the hardness by reducing from the problem of coloring regular graphs. Holyer [32] showed that determining if a 3-regular graph admits an edge 3-coloring is NP-hard. Since the line graph of a 3-regular graph is a 4-regular graph, it follows that determining if a 4-regular graph admits a node 3-coloring is NP-hard. We reduce from the latter problem.

Theorem 3. *The LSH feasibility problem is NP-hard even if the similarity S, operating on the universe U, assumes only values in $\{0, 1/6, 1/3, 2/3, 1\}$. More specifically, it is NP-hard to distinguish between the following two alternatives:*

- *S is LSHable, and*
- *every LSHable similarity S' operating on U satisfies $\ell_1(S, S') > \frac{2}{3|U|^2}$.*

Proof. Given a 4-regular graph G, we create a similarity $S = S(G)$ as described in Algorithm 2. We give a pictorial representation of S on the set $U_{v,w} \cup U_{w,v}$ (Figure 1) and pictorial representations of S on the sets $\Delta_{v,1}$, $\Delta_{v,2}$, and $\Delta_{v,3}$ (Figure 2).

Observe that the construction runs in polynomial time, and that $|U| = O(|V(G)|)$. We will show below that $S(G)$ is LSHable if (Lemma 4) and only if (Lemma 5) G is 3-colorable. Moreover, in Lemma 5 we will prove that if G is not 3-colorable, then $S(G)$ is at ℓ_1-distance more than $\frac{2}{3|U|^2}$ from any LSHable similarity. □

Lemma 4. *$\chi(G) \leq 3 \implies S(G)$ is LSHable.*

Proof. Consider any valid 3-coloring $\phi : V \to [3]$ of the nodes of G. Let $\pi_1, \pi_2, \pi_3, \pi_4, \pi_5, \pi_6$ be the six distinct permutations on the set $[3]$. For $i \in [6]$, we define $\phi_i : V \to [3]$ as $\phi_i(v) = \pi_i(\phi(v))$. Then, for each $i \in [6]$, $\phi_i(v) \neq \phi_i(w)$ for each $\{v, w\} \in E$, i.e., ϕ_i is a valid coloring of G.

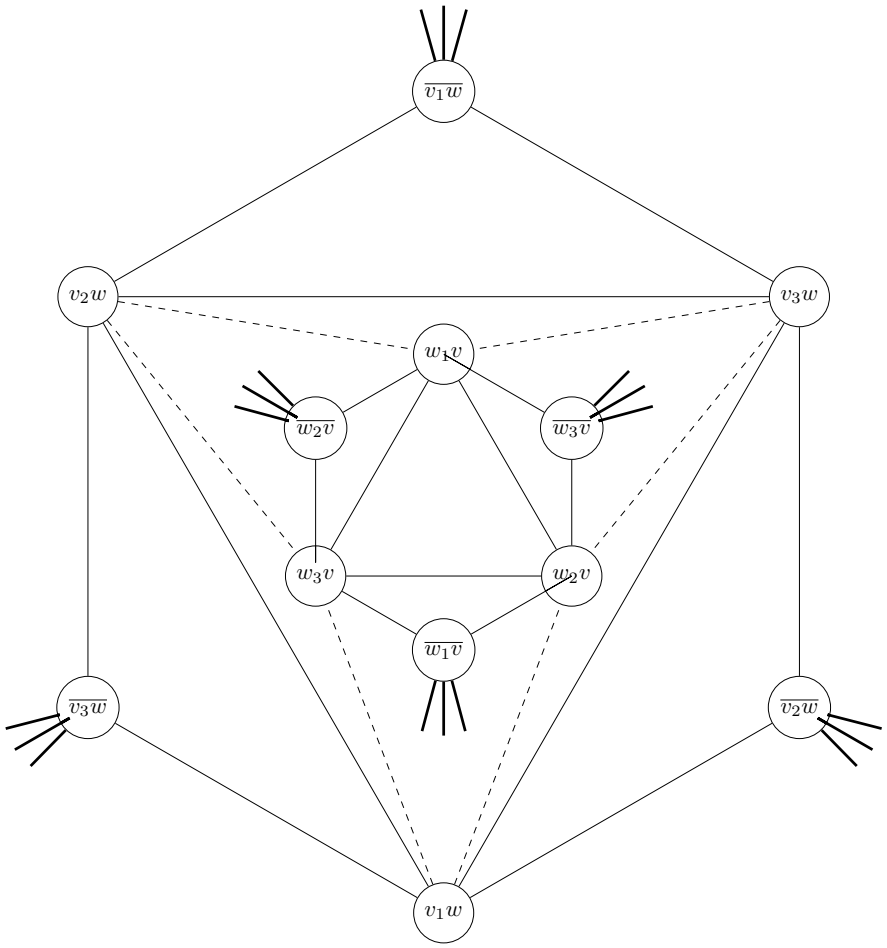

Figure 1: The weighted graph induced by the similarity S projected on the set $X_{\{v,w\}} = U_{v,w} \cup U_{w,v}$ of the elements of an edge $\{v, w\} \in E$. The six outer elements form the set $U_{v,w}$, and the six inner ones form the set $U_{w,v}$. A bold edge corresponds to similarity $2/3$, a light edge to similarity $1/3$, and a dashed edge to similarity $1/6$. The absence of an edge corresponds to similarity 0.

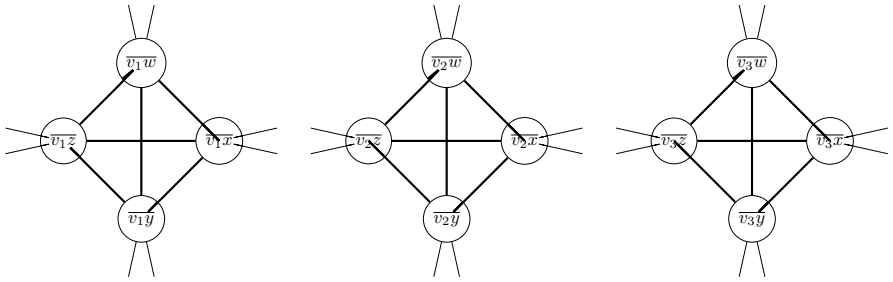

Figure 2: The weighted graph induced by the similarity S projected on the tetrahedrons (left to right) $\Delta_{v,1}$, $\Delta_{v,2}$, and $\Delta_{v,3}$ for a node $v \in V$ with neighbor set $N(v) = \{w, x, y, z\}$. A bold edge corresponds to similarity $2/3$, a light edge to similarity $1/3$, and no edge to similarity 0.

Algorithm 2 The NP-hardness construction.

Input: A 4-regular graph $G = (V, E)$.

Output: A similarity matrix S and a universe U.

1: Let $n = |V|$ and $m = |E| = 2n$. For each edge $\{v, w\} \in E$, we create two sets $U_{v,w}$ and $U_{w,v}$ of six unique elements each. The universe $U = \bigcup_{\{v,w\}\in E} (U_{v,w} \cup U_{w,v})$; thus $|U| = 12m = 24n$.

2: For an (ordered) pair (v, w) of nodes such that $\{v, w\} \in E$, let $U_{v,w} = \bigcup_{i=1}^{3} \{v_i w, \overline{v_i w}\}$. The elements $\overline{v_i w}$ are called *negated*.

3: We describe the similarity S between the elements in an arbitrary set $U_{v,w}$. First of all, for each $i \in [3]$, let $\{j, k\} = [3] \setminus \{i\}$ and let $\Gamma_{v,w,i} = \{\overline{v_i w}, v_j w, v_k w\}$. We will refer to the set $\Gamma_{v,w,i}$ as the $v_i w$-*triangle*. Observe that $U_{v,w} = \cup_{i=1}^{3} \Gamma_{v,w,i}$. For each $i \in [3]$ and for each pair of distinct elements a, b in the triangle $\Gamma_{v,w,i}$, let $S(a, b) = 1/3$. Furthermore, for each pair of distinct $i, j \in [3]$, let $S(\overline{v_i w}, \overline{v_j w}) = 0$. (See Figure 1 for an illustration of $U_{v,w}$ and the similarities of its elements.)

4: We now describe the similarity S between an element of $U_{v,w}$ and an element of $U_{w,v}$. For each $i \in [3]$, $S(v_i w, w_i v) = 0$, and for each of the six ordered pairs of distinct $i, j \in [3]$, we set $S(v_i w, w_j v) = 1/6$. For each $i \in [3]$, and for each $a \in U_{w,v}$ (resp., $a \in U_{v,w}$), we set $S(\overline{v_i w}, a) = 0$ (resp., $S(\overline{w_i v}, a) = 0$).

5: Take any node $v \in V$, and let $N(v) = \{w, x, y, z\}$ be the set of its neighbors. For each $i \in [3]$, let $\Delta_{v,i} = \{\overline{v_i w}, \overline{v_i x}, \overline{v_i y}, \overline{v_i z}\}$. We refer to the set $\Delta_{v,i}$ as the v_i-*tetrahedron*. For every $v \in V$, $i \in [3]$, and every two distinct $a, b \in \Delta_{v,i}$, let $S(a, b) = 2/3$.

6: Finally, two distinct elements $a, b \in U$ have $S(a, b) = 0$ if there exists no edge $\{v, w\} \in E$ such that $a, b \in U_{v,w} \cup U_{w,v}$, and if there exists no $v \in V, i \in [3]$ such that $a, b \in \Delta_{v,i}$.

For each $i \in [6]$, we define a hash function h_i, operating on the universe set U, that will be chosen with probability $1/6$ in the LSH. We will describe h_i by listing its maximal h_i-uniform sets:

(i) for each $v \in V$ and for each neighbor w of v, the $v_{\phi_i(v)}w$-triangle $\Gamma_{v,w,\phi_i(v)}$ will be a maximal h_i-uniform set;

(ii) for each $v \in V$, $j \in [3] \setminus \{\phi_i(v)\}$, the v_j-tetrahedron $\Delta_{v,j}$ will be a maximal h_i-uniform set;

(iii) for each $\{v, w\} \in E$, $\{v_{\phi_i(v)}w, w_{\phi_i(w)}v\}$ will be a maximal h_i-uniform set.

We observe that if the coloring ϕ assigned at most two distinct colors

to the nodes of V, i.e., $|\phi(V)| \leq 2$, then the six hash functions $h_1, \ldots, h_6$ would not be distinct.

Observe that each $\Gamma_{v,w,i}$-triangle will be a maximal uniform set in exactly two of the six chosen hash functions: the h_k's for which $\pi_k(\phi(v)) = i$. Furthermore, no other chosen hash function will map two elements of $\Gamma_{v,w,i}$ to the same class. Therefore, given any two elements of any triangle $\Gamma_{v,w,i}$, the probability that they will be hashed to the same value is exactly $1/3$. Moreover, no chosen hash function will map an element in $\Gamma_{v,w,i}$ and another element in $\Gamma_{v,w,j}$, $j \neq i$, to the same value. Therefore, the value-constraints induced by the similarity S are satisfied in every set $U_{v,w}$.

Each $\Delta_{v,i}$ tetrahedron will be a maximal uniform set in exactly four of the six chosen hash functions: the h_k's for which $\pi_k(\phi(v)) \neq i$. As before, no other chosen hash function will map two elements of $\Delta_{v,i}$ to the same class. It follows that, given any two elements of any $\Delta_{v,i}$ tetrahedron, the probability that they will be hashed to the same value is exactly $2/3$.

Furthermore, for each edge $\{v, w\} \in E$, and for each (ordered) pair of distinct colors $i, j \in [3]$, the two elements $\{v_i w, w_j v\}$ will be hashed to the same value in exactly one of the chosen hash functions: the h_k for which $\pi_k(\phi(v)) = i$ and $\pi_k(\phi(w)) = j$ (observe that h_k exists because ϕ is a valid coloring of G). Therefore their similarity will be exactly $1/6$.

The proof is complete by observing that no other pair of elements is ever hashed to the same value. □

Lemma 5. *$\chi(G) > 3 \Longrightarrow S(G)$ is at ℓ_1-distance more than $\frac{2}{3|U|^2}$ from any similarity that is LSHable.*

Proof. The idea is to give a solution to the dual (2) of the LSH feasibility system of each similarity T which is close enough to S. We aim to find a vector π (with coordinates indexed by elements in $\binom{U}{2} \cup \{\star\}$) such that $\pi M \geq \mathbf{0}$ and $\pi T^\star < 0$; this shows no such p exists and hence T is not LSHable.

We will now define a vector π; we will later show that it satisfies the inequalities above, for each similarity T such that $\ell_1(T, S(G)) \leq \frac{2}{3|U|^2}$. Let $S = S(G)$.

Define $\pi(\star) = 96m - 4$. Let $A = -6, B = -12, C = 3, D = -8$. For each

$\{u, u'\} \in \binom{U}{2}$, define

$$\pi(\{u,v\}) = \begin{cases} 12\binom{|U|}{2} & \text{if } S(u,v) = 0, \\ A & \text{if } S(u,v) = 1/6, \\ B & \text{if } S(u,v) = 1/3 \text{ and at least one of } u \text{ or } v \text{ is negated}, \\ C & \text{if } S(u,v) = 1/3 \text{ and neither } u \text{ nor } v \text{ is negated}, \\ D & \text{if } S(u,v) = 2/3. \end{cases}$$

Let T be any similarity such that $\ell_1(T,S) = \delta \leq \frac{2}{3|U|^2}$. Let T^* be the right-hand side vector in the LSH feasibility system of T. We will prove that $\pi T^\star < 0$ and that $\pi M \geq \mathbf{0}$.

($\pi T^\star < 0$). We start by computing the value of $\pi S^\star$:

$$\begin{aligned} \pi S^\star &= 2m \cdot \left(\frac{6}{2} \cdot \frac{1}{6} \cdot A + 6 \cdot \frac{1}{3} \cdot B + 3 \cdot \frac{1}{3} \cdot C + \frac{9}{2} \cdot \frac{2}{3} \cdot D \right) + \pi(\star) \\ &= -96m + \pi(\star) = -4. \end{aligned}$$

Since $S^\star(\star) = T^\star(\star)$ it follows that $\ell_1(S^\star, T^\star) = \delta$. Furthermore, on every coordinate $\{u, u'\}$, we have $|\pi(\{u,u'\})| \leq 12\binom{|U|}{2} < 6|U|^2$. Therefore, $|\pi S^\star - \pi T^\star| < 6|U|^2 \ell_1(S^\star, T^\star) = 6|U|^2\delta \leq 4$. Since $\pi S^\star = -4$, it follows that $\pi T^\star < 0$.

($\pi M \geq \mathbf{0}$). Let M_h be the h-column of the matrix M. We will show $\pi M_h \geq 0$, for each $h \in \mathcal{H}$, thus proving our claim. Let us start by considering the set $X_{\{v,w\}} = U_{v,w} \cup U_{w,v}$ for an arbitrary edge $\{v,w\} \in E$. The total contribution $Q_{\{v,w\}}$ of the elements in $X_{\{v,w\}}$ to πM_h equals

$$Q_{\{v,w\}} = \sum_{u \in X_{\{v,w\}}} \sum_{u' \in U \setminus \{u\}} \left(\frac{1}{2} \cdot \pi(\{u,u'\}) \cdot M_h(\{u,u'\}) \right).$$

Since the sets $X_{\{v,w\}}$ form a partition U, we have

$$\pi M_h = \sum_{\{v,w\} \in E} Q_{\{v,w\}} + \pi(\star) = \sum_{\{v,w\} \in E} Q_{\{v,w\}} + 96m - 4.$$

Now, note that if there are two elements u, v with $S(u,v) = 0$ and $h(u) = h(v)$, then $\pi M_h \geq 0$; indeed, since $\pi(\{u',v'\}) \geq -12$ for each $u', v' \in U$, we have $\pi M_h \geq \pi(\{u,v\}) - 12 \cdot \binom{|U|}{2} \geq 0$. Therefore, for the rest of the proof, we can assume that

$$h(u) = h(v) \implies S(u,v) > 0. \tag{3}$$

Assuming (3), we will prove

$$\forall \{v,w\} \in E, \quad Q_{\{v,w\}} \geq -96, \tag{4}$$

and

$$\chi(G) > 3 \implies \exists \{v,w\} \in E \text{ such that } Q_{\{v,w\}} \geq -92. \tag{5}$$

Proving (4) and (5) is sufficient since $\pi M_h \geq -((m-1)\cdot 96 + 92) + \pi(\star) = 0$.

We now proceed to showing (4) and (5). For each $i \in [3]$, and $\{j,k\} = [3] \setminus \{i\}$, and $\{v,w\} \in E$, let $X_{v,w,i} = \{\overline{v_i w}, v_j w, v_k w, w_i v\}$. Given the hash function h, and $\{a,b\} \in \binom{U}{2}$, define $Z_{\{a,b\}} = \pi(\{a,b\}) \cdot M_h(\{a,b\})$. Let

$$Q_{v,w,i} = \sum_{\{a,b\} \in \binom{\left\{\overline{v_i w}, w_j v, w_k v\right\}}{2}} Z_{\{a,b\}} + \frac{1}{2} \cdot \left(Z_{w_j v, v_i w} + Z_{w_k v, v_i w} + \sum_{x \in N(w) \setminus \{v\}} Z_{\overline{w_i v}, \overline{w_i x}} \right),$$

where $\{j,k\} = [3] \setminus \{i\}$, and $N(w)$ is the set of neighbors of w in G. Under (3), $Q_{v,w,i}$ is the total contribution to the sum $Q_{\{v,w\}}$ of the elements in $X_{v,w,i}$ (where the contribution of an element that belongs to two distinct $X_{v,w,i}$ and $X_{x,y,j}$ sets is split equally between the two; observe that no element is part of three distinct such sets.) Thus, $Q_{\{v,w\}} = \sum_{i=1}^{3} (Q_{v,w,i} + Q_{w,v,i})$.

If $Z_{v_j w, w_i v} \neq 0$, then $i \neq j$ and $h(v_j w) = h(w_i v)$; in this case, we will say that $v_j w$ and $w_i v$ form a *bridge* under h. Given h, the set $X_{v,w,i}$ will contain $0, 1$, or 2 bridges: the number of bridges will be equal to 0 if none of the pairs $\{v_j w, w_i v\}, \{v_k w, w_i v\}$ (with $\{j,k\} \in [3] \setminus \{i\}$) forms a bridge under h, to 1 if only one of them forms a bridge, and to 2 if both form bridges. We will say that $X_{v,w,i}$ contains a *crest* of width ℓ iff the element $\overline{v_i w}$ has the same hash of exactly ℓ elements in $\{\overline{v_i x}, \overline{v_i y}, \overline{v_i z}\}$. We say that $X_{v,w,i}$ contains a crest if it contains a crest of positive width, and a *full crest* if the width is maximum, i.e., 3. Furthermore, we will say that the set $X_{v,w,i}$ contains a *heavy triangle* iff the elements $\overline{v_i w}, v_j w, v_k w$ (i.e., the elements of $\Gamma_{v,w,i}$) are all hashed to the same value. Under (3), if $X_{v,w,i}$ contains a heavy triangle, then $w_i v$ is hashed to a value distinct to the common hash value of $\overline{v_i w}, v_j w, v_k w$. Furthermore, no element in $\{\overline{v_i x}, \overline{v_i y}, \overline{v_i z}\}$ can be hashed to the same value of $\overline{v_i w}$. Therefore $\Gamma_{v,w,i}$ will form a maximal h-uniform class (i.e., an equivalence class). In other words, under (3), if $X_{v,w,i}$ contains a heavy triangle, then it does not contain a crest, and it does not contain bridges.

We now state a key structural property of our construction.

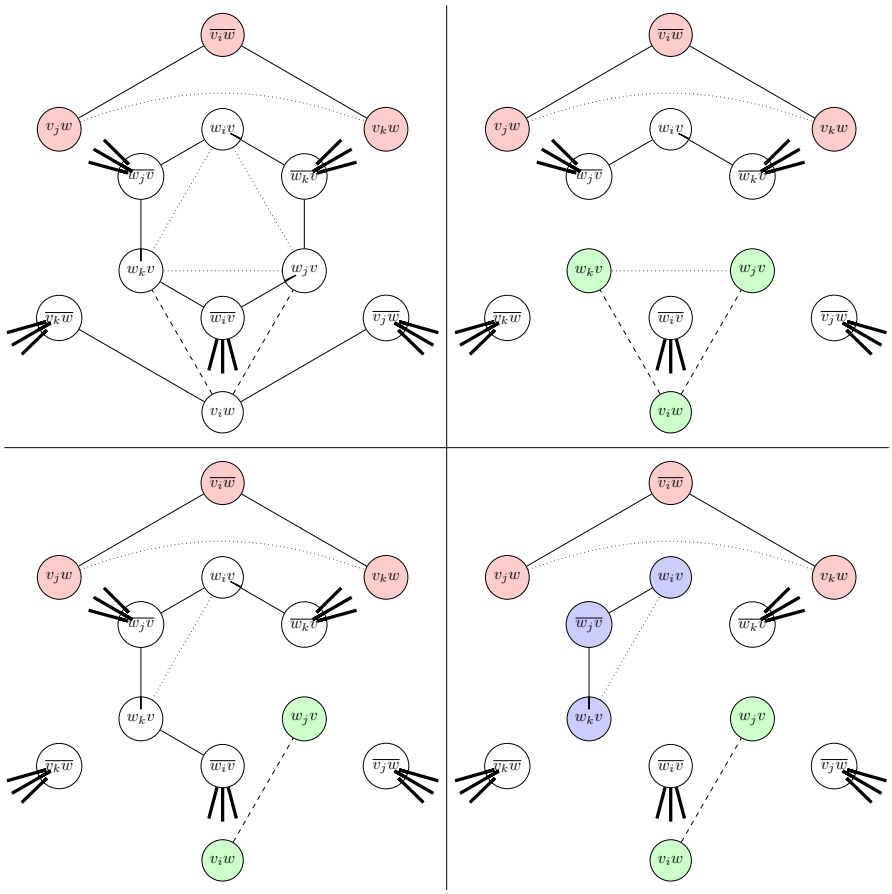

Figure 3: The weighted graph induced by the proposed solution π to the dual system, projected on the set $U_{v,w} \cup U_{w,v}$ of the elements of an edge $\{v, w\} \in E$. The six outer elements form the set $U_{v,w}$, and the six inner ones form the set $U_{w,v}$. In the solution π, a dashed edge corresponds to $A = -6$, a light edge to $B = -12$, a dotted edge to $C = 3$, and a bold edge to $D = -8$. The colors red, green and blue correspond to three distinct hash values (nodes in white have unspecified, and possibly non-uniform, hashes; the white nodes' hashes are different from the colored ones).

Lemma 6 (Structural property)**.** *Given a hash function h, under (3), and for every $\{v, w\} \in E$, one of the following two alternatives holds: (i) there exists i, j, k, with $\{i, j, k\} = [3]$, such that $X_{v,w,i}$ and $X_{w,v,j}$ contain heavy triangles, $X_{v,w,j}, X_{v,w,k}, X_{w,v,i}, X_{w,v,k}$ contain full crests, and the nodes $v_i w, w_j v$ form a bridge, in which case $Q_{\{v,w\}} = -96$;*[4] *or (ii) $Q_{\{v,w\}} \geq -92$.*

Proof. Fix a $X_{v,w,i}$. Given a hash function h, and under (3), one of the following disjoint possibilities must hold:

• $X_{v,w,i}$ has a heavy triangle. Then, it contains no bridges and no crest and hence $Q_{v,w,i} = 2B + C = -21$.

• $X_{v,w,i}$ has two bridges (top-right quadrant of Figure 3.) Then, h maps each of $w_i v, v_j w, v_k w$ to the same value, and $X_{v,w,i}$ $\overline{v_i w}$ can only have the same hash of elements in $\{\overline{v_i x} \mid x \in N(v)\}$. Therefore, $Q_{v,w,i} \geq \frac{2}{2}A + C + \frac{3}{2}D = -15$.

• $X_{v,w,i}$ has one bridge, wlog, $v_j w, w_i v$. Then, $h(v_j w) \neq h(v_k w)$, $h(v_j w) \neq h(\overline{v_i w})$, and either $X_{v,w,i}$ contains a crest or $h(\overline{v_i w}) = h(v_k w)$. Thus, either $Q_{v,w,i} \geq \frac{1}{2}A + \frac{3}{2}D$ or $Q_{v,w,i} \geq \frac{1}{2}A + B$. Since $\frac{1}{2}A + \frac{3}{2}D = \frac{1}{2}A + B = -15$, in either case we have $Q_{v,w,i} \geq -15$.

• $X_{v,w,i}$ has zero bridges and no heavy triangles. Then, either $X_{v,w,i}$ contains a crest, or it does not and at most one of $h(\overline{v_i w}) = h(v_j w)$, $h(\overline{v_i w}) = h(v_k w)$ holds. Therefore we have either $Q_{v,w,i} \geq \frac{3}{2}D$ or $Q_{v,w,i} \geq B$. Since $\frac{3}{2}D = B = -12$, in either case we obtain $Q_{v,w,i} \geq -12$.

Furthermore, observe that for any (v, w) such that $\{v, w\} \in E$, at most one of the sets $X_{v,w,1}, X_{v,w,2}, X_{v,w,3}$ can contain a heavy triangle. Indeed, if we let $\{i, j, k\} = [3]$, and if $X_{v,w,i}$ and $X_{v,w,j}$ both contained heavy triangles, then we would have both $h(\overline{v_i w}) = h(v_k w)$ and $h(\overline{v_j w}) = h(v_k, w)$. Therefore, we would also get $h(\overline{v_i w}) = h(\overline{v_j w})$, which contradicts (3).

Observe that if, for some $\{v, w\} \in E$, none of $X_{v,w,1}$, $X_{v,w,2}$, $X_{v,w,3}$, $X_{w,v,1}$, $X_{w,v,2}$, $X_{w,v,3}$ contains a heavy triangle, then we would have $Q_{\{v,w\}} \geq 6 \cdot (-15) = -90$. So (ii) would be satisfied. We then assume wlog that $X_{v,w,i}$ contains a heavy triangle (top-left quadrant of Figure 3) and hence $Q_{v,w,i} = -21$. We let $\{j, k\} = [3] \setminus i$. Observe that $X_{v,w,j}$ and $X_{v,w,k}$ cannot contain heavy triangles. Therefore the only possible bridges contributing to the sum $Q_{\{v,w\}}$ are $v_i w, w_j v$ and $v_i w, w_k v$, and they are both part of $X_{w,v,i}$.

Suppose that $X_{w,v,i}$ contains both bridges (top-right quadrant of Figure 3). Then, $Q_{w,v,i} \geq -15$. On the other hand, for $\{j, k\} = [3] \setminus \{i\}$,

[4] See a representation of this partition in the lower-right quadrant of Figure 3.

we would have that $X_{v,w,j}$ and $X_{v,w,k}$ would both contain a single bridge, and therefore $Q_{v,w,j}, Q_{v,w,k} \geq -15$. On the other hand, $X_{w,v,j}$ and $X_{w,v,k}$ would contain zero bridges, and therefore $Q_{w,v,j}, Q_{w,v,k} \geq -12$. Therefore, we would have $Q_{\{v,w\}} \geq -21 + 3 \cdot (-15) + 2 \cdot (-12) = -90$. Thus (ii) would be satisfied.

Suppose $X_{w,v,i}$ contains zero bridges, then every set $X_{v,w,1}$, $X_{v,w,2}$, $X_{v,w,3}$, $X_{w,v,1}$, $X_{w,v,2}$, $X_{w,v,3}$ contains zero bridges. Since there can be at most one heavy triangle in $X_{w,v,1}, X_{w,v,2}, X_{w,v,3}$, and since there exists only one heavy triangle in $X_{v,w,1}, X_{v,w,2}, X_{v,w,3}$, we would have that $Q_{\{v,w\}} \geq 2 \cdot (-21) + 4 \cdot (-12) = -90$. Thus (ii) would be satisfied.

We then assume that $X_{w,v,i}$ contains exactly one bridge, wlog, $v_i w, w_j v$, and that bridge is the only bridge contributing to the sum $X_{\{v,w\}}$ (lower-left quadrant of Figure 3).

Recall that $X_{v,w,i}$ contains a heavy triangle. It follows that $Q_{w,v,i} \geq -15, Q_{v,w,j} \geq -15$ and $Q_{v,w,k} \geq -12, Q_{w,v,k} \geq -12$. Now, suppose that $T_{w,v,j}$ does not contain a heavy triangle; then $Q_{w,v,j} \geq -15$. Therefore we would have $Q_{\{w,v\}} = -21 + 3 \cdot (-15) + 2 \cdot (-12) = -90$. Thus (ii) would be satisfied.

Finally, suppose $X_{w,v,j}$ contains a heavy triangle (lower-right quadrant of Figure 3). Let c be the total of the widths of the crests of $X_{v,w,j}, X_{v,w,k}, X_{w,v,i}$, and $X_{w,v,k}$. Observe that $c \leq 12$, and $c = 12$ iff each of $X_{v,w,j}, X_{v,w,k}, X_{w,v,i}, X_{w,v,k}$ contains a full crest. Then, $Q_{\{v,w\}} = 2 \cdot (-21) + (-6) + \frac{1}{2} \cdot (-8) \cdot c = -48 - 4c$. Therefore, either all $X_{v,w,j}, X_{v,w,k}, X_{w,v,i}, X_{w,v,k}$ contain a full crest (and, since $c = 12$, (i) holds), or (ii) holds (because $Q_{\{v,w\}} \geq -48 - 4 \cdot 11 = -92$.)

This completes the proof of the structural property. □

We claim that Lemma 6 is enough to establish (4) and (5). Easily, Lemma 6 $\implies$ (4). Note that (5) is satisfied if there exists $\{v, w\} \in E$ for which Lemma 6(ii) holds. Thus we have to prove that if $\chi(G) > 3$, then it is impossible that Lemma 6(i) holds for all edges $\{v, w\} \in E$. Suppose the contrary. Then, under h, the $4m$ full crests form m maximal equivalence classes of size 4 each. In particular, for each v, if $N(v) = \{w, x, y, z\}$, then there must exist $j', k' \in [3]$, $j' \neq k'$, such that both $\Delta_{v,j'} = \{\overline{v_{j'}w}, \overline{v_{j'}x}, \overline{v_{j'}y}, \overline{v_{j'}z}\}$ and $\Delta_{v,k'} = \{\overline{v_{k'}w}, \overline{v_{k'}x}, \overline{v_{k'}y}, \overline{v_{k'}z}\}$ form a maximal equivalence class. Observe that each of the classes hits all the four edge-gadget of which v is part. We will then say that v is assigned color $i' \in [3] \setminus \{j', k'\}$. Given the two maximal equivalence classes, between the three sets $X_{v,w,i'}, X_{v,w,j'}, X_{v,w,k'}$, the set $X_{v,w,i'}$ is the only set that could contain a heavy triangle and by Lemma 6(i), it must contain it. Therefore, the bridge of the edge $\{v, w\}$ must hit the element $v_{i'}w$ and in fact, the bridge of each

edge $\{v, v'\}$, $v' \in N(v)$ must hit the element $v_{i'}v'$.

Analogously, for a neighbor w of v, with $N(w) = \{v, x', y', z'\}$, there must exists $j'', k'' \in [3]$, $j'' \neq k''$, such that $\Delta_{w,j''} = \{\overline{w_{j''}v}, \overline{w_{j''}x'}, \overline{w_{j''}y'}, \overline{w_{j''}z'}\}$ and $\Delta_{w,k''} = \{\overline{w_{k''}v}, \overline{v_{k''}x'}, \overline{v_{k''}y'}, \overline{v_{k''}z'}\}$ form maximal equivalence classes; we will then say that w would be assigned color $i'' \in [3] \setminus \{j'', k''\}$. Again, the bridge of each edge $\{w, w'\}$, $w' \in N(w)$, must hit the element $w_{i''}w'$.

Now observe that $i' \neq i''$, for otherwise there could be no bridge for the edge $\{v, w\}$: bridges can only connect two elements having two different indices in $\{1, 2, 3\}$. Since we have assumed that Lemma 6(i) holds for every edge in E, the 3-coloring that we have built will be valid. Therefore, if Lemma 6(i) holds for every edge in E, then $\chi(G) \leq 3$. This contradicts the assumption and therefore there must exist an edge in E for which Lemma 6(ii) holds. □

Finally, to stress the difference between the LSH feasibility problem and the ℓ_1-embeddability problem (which was shown to be NP-hard under Turing reductions in [13]), we observe that the similarity S produced by Algorithm 2 is such that $1 - S$ is a metric that always embeds isometrically into ℓ_1, regardless of whether S is LSHable or not. The embedding can be constructed using the following cut metrics. Assign a weight of:
(i) 1/12 to the cut induced by $\{v_i w, w_j v\}$, for each $\{v, w\} \in E$ and each $\{i, j\} \in \binom{[3]}{2}$,
(ii) 1/6 to each of the two cuts induced by the triangles $\Gamma_{v,w,i}$ and $\Gamma_{w,v,i}$, for each $i \in [3], \{v, w\} \in E$, and
(iii) 1/3 to the cut induced by $\Delta_{v,i}$ for each $v \in V$ and each $i \in [3]$.

In other words, if we let $C_1, C_2, \ldots$ be the sets that we used to define the above cuts, we can create a coordinate for each cut C_k. The value of element a's vector in its kth coordinate will be 0 if $a \notin C_k$, and the weight that we gave to C_k otherwise. An easy calculation shows that the ℓ_1-distance between the vectors corresponding to every two elements is exactly 1 minus their similarity.

4.2. Gap amplification

In this section we amplify the LSH feasibility hardness gap in Theorem 3 up to something close to the maximum possible. We start with a fact that we will use in the amplification.

Lemma 7. *Let k and t be integers such that $2 \leq k \leq t/2$. If $U_1, \ldots, U_k$ are pairwise disjoint sets of size t each, then there exists a set $\mathcal{C} \subseteq U_1 \times \cdots \times U_k$ of size $|\mathcal{C}| \geq (t/k)^2$ such that for each $\{C, C'\} \in \binom{\mathcal{C}}{2}$, it holds that C and C' agree in at most one coordinate.*

Proof. Let $\mathcal{D} = U_1 \times \cdots \times U_k$; we have $|\mathcal{D}| = t^k$.

Given an arbitrary set $D \in \mathcal{D}$, the number of sets $D' \in \mathcal{D} \setminus \{D\}$ that agree in at least two coordinates with D is

$$\begin{aligned}
\Delta &= \sum_{i=2}^{k-1} \left(\binom{k}{i} (t-1)^{k-i} \right) = \sum_{i=2}^{k} \left(\binom{k}{i} (t-1)^{k-i} \right) - 1 \\
&\leq \sum_{i=2}^{k} \left(\frac{k^i}{i!} (t-1)^{k-i} \right) - 1 \leq \frac{1}{2} \sum_{i=2}^{k} \left(k^i t^{k-i} \right) - 1 \\
&= \frac{t^{k-2}k^2}{2} \sum_{i=0}^{\infty} \left(\frac{k}{t} \right)^i - 1 = \frac{t^{k-2}k^2}{2} \frac{1}{1-\frac{k}{t}} - 1 \\
&\leq t^{k-2}k^2 - 1.
\end{aligned}$$

If we consider the graph G with vertex set $\mathcal{D}$ and an edge $\{D, D'\} \in \binom{\mathcal{D}}{2}$ iff D and D' agree in at least two coordinates, we have that the maximum degree in G is Δ. Now, let $\mathcal{C}$ be a maximum independent set of G. Then, for any two $\{C, C'\} \in \binom{\mathcal{C}}{2}$, we will have that C and C' agree in at most one coordinate. Furthermore, we will have

$$|\mathcal{C}| \geq \frac{|\mathcal{D}|}{\Delta + 1} \geq \frac{t^k}{t^{k-2}k^2} = \frac{t^2}{k^2}.$$

□

Theorem 8. *Fix any $\epsilon > 0$, and let T be a similarity operating on the universe V. Then it is NP-hard to distinguish between the following two alternatives:*

- *T is LSHable, and*
- *every LSHable similarity T' operating on V satisfies $\ell_1(T, T') > |V|^{2-\epsilon}$.*

Proof. Consider the matrix S produced by Algorithm 2 and used in the reduction of Theorem 3. Let $k \times k$ be its size, i.e., let k be the size of the universe set U of S. The reduction is such that k is polynomial (and, in fact, linear) in the size of the original graph G.

Pick an arbitrarily small constant $\epsilon > 0$, and make $t = \lceil k^{1/\epsilon} \rceil$ copies of each element $u_i \in U$. The universe of the new similarity T is given by $V = \{u_{i,p} \mid u_i \in U \wedge p \in [t]\}$; let $n = |V| = kt$. Observe that $t^2/k^{O(1)} = n^{2-O(\epsilon)}$.

Let $\alpha = 1 - k^{-4}$. Given $u_{i,p}, u_{j,q} \in V$, let $T(u_{i,p}, u_{j,q}) = \alpha \cdot S(u_i, u_j)$, if $i \neq j \vee p \neq q$, and $T(u_{i,p}, u_{j,q}) = 1$ otherwise.[5]

Recall that Theorem 3 states that it is NP-hard to distinguish if S is LSHable or if it is at distance more than $\frac{2}{3k^2}$ from an LSHable similarity. We show the following.

(i) S is LSHable $\implies$ T is LSHable. Let p_h be the probability that was given by the LSH for S to the hash function h on the domain U. For each such hash function h, we create a new hash function h' on domain U_t such that, for each $u_{i,p} \in U$, $h'(u_{i,p}) = h(u_i)$. We will be choosing this h' with probability αp_h. Finally, we add $1 - \alpha$ to the probability of choosing the identity hash function h' over V (i.e., $|h'(V)| = |V|$). This hashing scheme is easily seen to be an LSH for T.

(ii) S is at ℓ_1-distance more than $\frac{2}{3k^2}$ from any LSHable similarity $\implies$ T is at ℓ_1-distance at least $n^{2-O(\epsilon)} = |V|^{2-O(\epsilon)}$ from any LSHable similarity. For each $i = 1, \ldots, k$, let $V_i = \{u_{i,p} \mid p \in [t]\}$. Observe that $V_1, \ldots, V_k$ form an equipartition of V. Lemma 7 guarantees that there exists a class $\mathcal{C} \subseteq V_1 \times \cdots \times V_k$ of size $|\mathcal{C}| \geq (t/k)^2$ such that any two distinct $C, C' \in \mathcal{C}$ share at most one element. Pick any $C \in \mathcal{C}$, and take any two distinct elements $u_{i,p}, u_{j,q} \in C$. Observe that it must hold $i \neq j$. By construction, we have that $T(u_{i,p}, u_{j,q}) = \alpha S(u_i, u_j)$. Since S has range $[0, 1]$, we have

$$\begin{aligned} |T(u_{i,p}, u_{j,q}) - S(u_i, u_j)| &\leq |\alpha S(u_i, u_j) - S(u_i, u_j)| \\ &\leq (1 - \alpha)\, S(u_i, u_j) \\ &\leq 1 - \alpha = k^{-4}. \end{aligned}$$

If we let T_C be the projection of T on the elements in C, we then have that $\ell_1(T_C, S) \leq \binom{k}{2} k^{-4} \leq \frac{1}{2k^2}$. Since S is at distance at least $\frac{2}{3k^2}$ from any LSHable similarity, the triangle inequality implies that T_C is at distance at least $\frac{2}{3k^2} - \frac{1}{2k^2} = \frac{1}{6k^2}$ from any LSHable similarity. Observe that the projections T_C, for each $C \in \mathcal{C}$, involve pairwise disjoint entries of the original similarity T (indeed, if $\{C, C'\} \in \binom{\mathcal{C}}{2}$ then C and C' share at most one element and, therefore, no entries of the similarity matrix). Therefore, the ℓ_1-distance between T and a LSHable similarity over the same universe set is at least $|\mathcal{C}| \frac{1}{6k^2} \geq \frac{1}{6} t^2 k^{-4} = n^{2-O(\epsilon)}$. □

[5] We observe that we could have set the similarity between two objects $u_{i,p}, u_{j,q} \in V$, $u_{i,p} \neq u_{j,q}$ to be equal to $S(u_i, u_j)$, instead of $\alpha \cdot S(u_i, u_j)$. This would have made the proof slightly simpler, but it would have also created pairs of distinct elements having similarity 1 with each other.

The proof can be easily modified to prove a hardness-gap of $n^{\frac{2}{p}-\epsilon}$ under the ℓ_p-distance between similarities, for each fixed $1 \leq p < \infty$, and for each fixed $\epsilon > 0$. Since the maximum ℓ_p-distance between two similarities on a universe of n elements is $\binom{n}{2}^{\frac{1}{p}} = \Theta\left(n^{\frac{2}{p}}\right)$, the ℓ_p-hardness result is also near-tight.

4.3. A time lower bound under the ETH

We now observe that under the Exponential Time Hypothesis (ETH) [33], our proof of Theorem 3 directly entails a lower bound on the time complexity of algorithms for the LSH feasibility problem.

Corollary 9. *Assuming the ETH, determining whether a similarity operating on a universe of size n is LSHable requires time at least $2^{\Omega(n)}$.*

Proof. Recall that the proof of Theorem 3 is a reduction from the 3-edge-colorability problem on cubic graphs [32]. Starting from an n-node graph, we build a similarity operating on a universe of size $\Theta(n)$. In [32], Holyer showed how to reduce 3-SAT to the 3-edge-colorability problem on cubic graphs. His proof reduces a 3-SAT formula on n variables and m clauses to a cubic graph with $\Theta(n+m)$ nodes.

The ETH [33] claims that there exists a constant $\delta > 0$ such that 3-SAT cannot be solved in time $o\left(2^{\delta n}\right)$. Moreover, Impagliazzo and Paturi [33] show that for each $\epsilon > 0$, a 3-SAT formula on n variables and m clauses can be reduced in time $O(2^{\epsilon n} \cdot \text{poly}(m))$ to the disjunction of $2^{\epsilon n}$ 3-SAT formulas on the same n variables, with each formula containing at most $m' = \Theta(n)$ clauses.

Therefore, if for each constant $\epsilon > 0$, one could solve the LSH feasibility problem in time at most $O\left(2^{\epsilon n}\right)$, where n is the size of the universe of the input similarity, one could also solve 3-SAT in time $O\left(c^{\,\epsilon n} \cdot \text{poly}(m)\right)$, for some constant $c > 1$ and for all $\epsilon > 0$, contradicting the ETH. □

The algorithm in Section 3 matches the lower bound of Corollary 9. We remark that the n in the exponent of the upper and lower bounds is *not* the size of the input of the LSH feasibility problem (which is $\Omega(n^2)$).

5. Hardness of JPM feasibility

In this section we consider the *Joint Probability Matrices (JPM)* feasibility problem: given a symmetric matrix $\{p_{i,j}\}_{1 \leq i,j \leq n}$ of non-negative numbers in $[0,1]$, are there random events $E_1, \ldots, E_n$ and a probability space such

that for every $i, j \in [n]$, $\Pr[E_i \wedge E_j] = p_{i,j}$? There is a significant connection between these two problems, at least at an intuitive level. In the LSH feasibility problem, we ask whether there exists a weighted collection of partitions into cliques that satisfies some probabilistic constraints. In the JPM feasibility problem, we ask whether there exists a weighted collection of cliques that satisfies some probabilistic constraints. As we mentioned earlier, in the case where some of the $p_{i,j}$'s can be unspecified, then the problem is already known to be NP-hard [27].

Note that it is not a priori obvious that for positive instances of the JPM problem, one can find a concise representation of the joint probability distribution of the events $E_1, \ldots, E_n$, since in general, representing such a distribution might require $2^n - 1$ numbers (the probabilities of each of the 2^n subsets of events). However, as we will see, the JPM problem is a linear system with $O(n^2)$ constraints; therefore, if an instance is positive, then it can be supported by a probability space with $O(n^2)$ atoms.

Let x be a vector indexed by all the subsets of $[n]$. Using $x(S)$, with $S \subseteq [n]$, we aim to denote the probability of the event $E_S = \bigwedge_{i \in S} E_i \wedge \bigwedge_{i \notin S} \overline{E}_i$. Let A be a matrix whose columns are indexed by the subsets of $[n]$, and with rows indexed by $[n]^2 \cup \{\star\}$. The entry $A((i,j), S)$ will be equal to 1 if $i, j \in S$, and 0 otherwise. The $\star$-row of A will be all-ones. Furthermore, let the vector $P^\star$ be indexed by $[n]^2 \cup \{\star\}$, with $P^\star((i,j)) = P(i,j)$, for $1 \leq i, j \leq n$, and $P^\star(\star) = 1$.

Then, the JPM problem is equivalent to the following linear system:

$$Ax \quad = \quad P^\star; \qquad\qquad x \quad \geq \quad 0. \tag{6}$$

Proposition 10. *For any feasible instance of* JPM*, there is a solution where the support of the joint probability distribution of the events has size at most* $n^2 + 1$*.*

Proof. If the polytope defined by the above system of linear constraints is non-empty, then it must have a basic feasible solution. The number of non-zero coordinates in this basic feasible solution cannot exceed the number of constraints, which is $n^2 + 1$. □

By the Farkas's Lemma, (6) is *infeasible* iff there is a vector y, with entries in $[n]^2 \cup \{\star\}$, such that

$$yA \quad \geq \quad \mathbf{0}; \qquad\qquad yP^\star \quad < \quad 0. \tag{7}$$

Observe that (7) has a solution iff, for each $S \subseteq [n]$, we have $\sum_{(i,j) \in S^2} y(i,j) > \sum_{(i,j) \in [n]^2} P(i,j) \cdot Y(i,j)$. We now present a Turing reduction from Clique

to JPM, i.e., we assume that we have access to an oracle that given an instance of JPM, finds a solution, and using this oracle, give a polynomial-time algorithm that solves CLIQUE. The proof proceeds as follows: We define a particular polytope and use the JPM oracle to construct a separation oracle for the polar of this polytope. We will then use the Ellipsoid algorithm with this separation oracle to optimize over this polytope, and will prove that optimizing over this polytope in a certain direction will give us a solution to CLIQUE.

Theorem 11. *The JPM feasibility problem is NP-hard under Turing reductions.*

Proof. We define a polytope R as follows: R is in the n^2-dimensional Euclidean space $\mathbb{R}^{n\times n}$, with each coordinate corresponding to a pair $(i,j) \in [n]\times[n]$. For each subset $S \subseteq [n]$, we define a point $a_S \in \mathbb{R}^{n\times n}$ that has a 1 in every coordinate (i,j) with $i,j \in S$ and a 0 in the other coordinates. The polytope R is defined as the convex hull of the 2^n points a_S. This polytope is essentially the polar of the dual polytope defined above.

Next, we show that given an oracle that finds a solution for JPM, we can give a separation oracle for R. The separation oracle for R needs to solve the following problem: given a point $b \in \mathbb{R}^{n\times n}$, decide if this point is in the convex hull of the points a_S, and if it is not, find a separating hyperplane. Note that b is in the convex hull of a_S's if there is a convex combination of these points that equals b, i.e., if there are non-negative values x_S such that $b = \sum_S x_S a_S$, and $\sum_S x_S = 1$. This is precisely equivalent to the matrix $[b_{ij}]$ being a feasible instance of JPM: if $[b_{ij}]$ is a feasible instance of JPM, there are events $E_1,\ldots,E_n$ such that $\Pr[E_i \wedge E_j] = b_{ij}$, and it is easy to see that for $x_S = \Pr[\bigwedge_{i\in S} E_i \wedge \bigwedge_{i\notin S} \overline{E}_i]$, we have $b = \sum_S x_S a_S$. Conversely, if there are x_S's such that $b = \sum_S x_S a_S$, we can define a distribution by picking a set S with probability x_S, and then letting all events E_i for $i \in S$ be true and the E_i's for $i \notin S$ not be true. Clearly, for this probability distribution, $b_{ij} = \Pr[E_i \wedge E_j]$ for every i and j. Therefore, using the JPM oracle, we can decide if the given point b is inside the polytope R.

If the point b is not in the polytope R, we need to find a separating hyperplane. We do this as follows: first, using binary search, we find the largest value $\alpha \geq 0$ such that $\alpha.b$ is in R; such a value should exist since $\bar{0}$ is in R. By the definition of α, αb must be on a facet of R. The solution of JPM gives a way of writing αb as a convex combination of a_S's. The points a_S that appear in this convex combination must be on the same facet as αb, and therefore their linear combination defines the facet that αb is on. It is

easy to see that this hyperplane separates b from R.[6]

Given the separation oracle and the Ellipsoid algorithm, we can optimize over the polytope R in any given direction. Now, consider the following direction: for a given graph $G = (V, E)$, we define the vector g as follows: $g_{ij} = 1$ if $(i, j) \in E$ and is $-n^2$ otherwise. Consider the problem of optimizing over R in the direction of g. We can assume, without loss of generality, that the optimal point is one of the nodes of this polytope, since otherwise we can use the JPM oracle to write this point as a convex combination of nodes, and then output any of the nodes in the convex combination as an optimal solution. Therefore, this optimization is equivalent to finding a point a_S that maximizes ga_S. The value of ga_S is negative if there are two nodes in S that are not connected by an edge. Therefore, for ga_S to be positive, S needs to be a clique in G, and the value of this objective function is precisely $\binom{|S|}{2}$. Thus, by optimizing over R in the direction given by g, we can solve the maximum clique problem in G. □

References

[1] P. Indyk, R. Motwani, P. Raghavan, S. Vempala, Locality-preserving hashing in multidimensional spaces, in: Proc. STOC, 1997, pp. 618–625.

[2] P. Indyk, R. Motwani, Approximate nearest neighbors: Towards removing the curse of dimensionality, in: Proc. STOC, 1998, pp. 604–613.

[3] A. Broder, S. Glassman, M. Manasse, G. Zweig, Syntactic clustering of the web, in: Proc. WWW, 1997, pp. 391–404.

[4] A. Broder, On the resemblance and containment of documents, in: Proc. SEQUENCES, 1997, pp. 21–29.

[5] A. Broder, M. Charikar, A. Frieze, M. Mitzenmacher, Min-wise independent permutations, J. Computing and System Sciences 60 (3) (2000) 630–659.

[6] F. Chierichetti, R. Kumar, LSH-preserving functions and their applications, in: Proc. SODA, 2012, pp. 1078–1094.

[6] We observe that it is possible that αb falls on a smaller dimensional facet of R. However, in this case, we can simply perturb b by a very small amount and redo the calculations.

[7] M. S. Charikar, Similarity estimation techniques from rounding algorithms, in: Proc. STOC, 2002, pp. 380–388.

[8] M. Datar, N. Immorlica, P. Indyk, V. Mirrokni, Locality-sensitive hashing scheme based on p-stable distributions, in: Proc. SoCG, 2004, pp. 253–262.

[9] A. Andoni, P. Indyk, Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions, Communications of the ACM 51 (1) (2008) 117–122.

[10] A. Andoni, Nearest neighbor search: the old, the new, and the impossible, Ph.D. thesis, MIT (2009).

[11] N. Dalvi, A. Dasgupta, V. Rastogi, T. Sarlos, Optimal hashing schemes for entity matching, in: Proc. WWW, 2013, pp. 295–306.

[12] J. Buhler, Provably sensitive indexing strategies for biosequence similarity search, in: Proc. RECOMB, 2002, pp. 90–99.

[13] D. Avis, M. Deza, The cut cone, L1 embeddability, complexity, and multicommodity flows, Networks 21 (6) (1991) 595–617.

[14] J. Edmonds, Embedding into ℓ_∞^2 is easy, embedding into ℓ_∞^3 is NP-complete, Discrete and Computational Geometry 39(4) (2008) 747–765.

[15] K. Onak, Testing properties of set of points in metric spaces, in: Proc. ICALP, 2008, pp. 515–526.

[16] R. Krauthgamer, O. Sasson, Property testing of data dimensionality, in: Proc. SODA, 2003, pp. 18–27.

[17] M. Badoiu, P. Indyk, A. Sidiropoulos, Approximation algorithms for embedding general metrics into trees, in: Proc. SODA, 2007, pp. 512–521.

[18] M. Badoiu, J. Chuzhoy, P. Indyk, A. Sidiropoulos, Low-distortion embeddings of general metrics into the line, in: Proc. STOC, 2005, pp. 225–233.

[19] M. Badoiu, J. Chuzhoy, P. Indyk, A. Sidiropoulos, Embedding ultrametrics into low-dimensional spaces, in: Proc. SOCG, 2006, pp. 187–196.

[20] M. Badoiu, K. Dhamdhere, A. Gupta, Y. Rabinovich, H. Raecke, R. Ravi, A. Sidiropoulos, Approximation algorithms for low-distortion embeddings into low-dimensional spaces, in: Proc. SODA, 2005, pp. 119–128.

[21] A. Sidiropoulos, Computational metric embeddings, Ph.D. thesis, MIT (2008).

[22] L. J. Emrich, M. R. Piedmonte, A method for generating high-dimensional multivariate binary variates, The American Statistician 45 (4) (1991) 302–304.

[23] A. J. Lee, Generating random binary deviates having fixed marginal distributions and specified degrees of association, The American Statistician 47 (3) (1993) 209–215.

[24] C. G. Park, T. Park, D. W. Shin, A simple method for generating correlated binary variates, The American Statistician 50 (4) (1996) 306–310.

[25] C.-K. Li, B.-S. Tam, A note on extreme correlation matrices, SIAM J. Matrix. Anal. Appl. 15 (3) (1994) 903–908.

[26] N. R. Chaganty, H. Joe, Range of correlation matrices for dependent Bernoulli random variables, Biometrika 93 (1) (2006) 197–206.

[27] D. Koller, N. Megiddo, Constructing small sample spaces satisfying given constraints, SIAM J. Discrete Math. 7 (2) (1994) 260–274.

[28] E. T. Bell, Exponential numbers, The American Mathematical Monthly 41 (7) (1934) pp. 411–419.

[29] N. De Bruijn, Asymptotic Methods in Analysis, Dover Books on Mathematics Series, Dover Publications, 1970.

[30] M. Grötschel, L. Lovász, A. Schrijver, The ellipsoid method and its consequences in combinatorial optimization, Combinatorica 1 (1981) 169–197.

[31] C. Carathéodory, über den variabilitätsbereich der fourierschen konstanten von positiven harmonischen funktionen, Rendiconti del Circolo Matematico di Palermo 32 (1) (1911) 193–217. doi:10.1007/BF03014795.

[32] I. Holyer, The NP-completeness of edge-coloring, SIAM J. Computing 10 (4) (1981) 718–720.

[33] R. Impagliazzo, R. Paturi, On the complexity of k-SAT, J. Computing and System Sciences 62 (2) (2001) 367 – 375.

[34] A. K. Lenstra, H. W. Lenstra, L. Lovász, Factoring polynomials with rational coefficients, Mathematische Annalen 261 (1982) 515–534.

Appendix A. The closest LSH-feasible similarity problem

First of all, let us define the closest LSH feasible similarity problem through a LP. The vector p will be defined as in the system (1). The matrix M', will be the matrix M of system (1) without its $\star$-row. Let δ be a vector indexed by elements in $\binom{U}{2}$. Then, the *primal LP* will have the p and δ as variables, and will be equal to:

$$\left\{\begin{array}{rcl} \min \ \mathbf{1}\cdot\delta & & \\ M\cdot p & \geq & S-\delta \\ M\cdot p & \leq & S+\delta \\ \mathbf{1}\cdot p & = & 1 \\ p & \geq & \mathbf{0} \\ \delta & \geq & \mathbf{0} \end{array}\right. \tag{A.1}$$

Obviously a solution of (A.1) will form a LSH (through p) for a similarity S' whose ℓ_1-distance to S will be equal to the value of the objective function of (A.1).

Let π be a vector indexed by the elements in $\binom{U}{2}$. If we take the dual of (A.1) we get:

$$\left\{\begin{array}{rcl} \max \ S\cdot\pi + x & & \\ \pi\cdot M + \mathbf{1}\cdot x & \leq & \mathbf{0} \\ \pi & \leq & \mathbf{1} \\ \pi & \geq & -\mathbf{1} \end{array}\right. \tag{A.2}$$

We rewrite (A.2) in an equivalent way:

$$\left\{\begin{array}{rcl} \min \ S\cdot\pi + x & & \\ \pi\cdot M + \mathbf{1}\cdot x & \geq & \mathbf{0} \\ \pi & \leq & \mathbf{1} \\ \pi & \geq & -\mathbf{1} \end{array}\right. \tag{A.3}$$

The dual (A.3) is very similar to the dual (2). The only differences are that (i) in (A.3), the variables $\pi(\{u,u'\})$ are bounded in $[-1,1]$, while

those same variables were unbounded in (2), (ii) the variable $\pi(\star)$ in (2) is replaced by x in (A.3), and (iii) the constraint $\pi S^\star < 0$ in (2) is replaced by the objective function in (A.3). Therefore, we can easily modify the separation oracle in Algorithm 1 to get a separation oracle for (A.3). The only additional checks that have to be made are the $\binom{|U|}{2}$ double inequalities $-1 \leq \pi(\{u, u'\}) \leq 1$ for each $\{u, u'\} \in \binom{U}{2}$.

Since we have a separation oracle for the dual, we can also optimize the primal [30, 34] and therefore we can find the closest LSHable similarity in time $\tilde{O}(3^n)$.