Algorithms for Diversity and Clustering in Social Networks through Dot Product Graphs *

Matthew Johnson[1], Daniël Paulusma[1], and Erik Jan van Leeuwen[2]

[1] School of Engineering and Computer Science, Durham University, England
{matthew.johnson2,daniel.paulusma}@durham.ac.uk
[2] Max-Planck Institut für Informatik, Saarbrücken, Germany
erikjan@mpi-inf.mpg.de

Abstract. In this paper, we investigate a graph-theoretical model of social networks. The *dot product model* assumes that two individuals are connected in the social network if their attributes or opinions are similar. In the model, a d-dimensional vector $\mathbf{a}^v$ represents the extent to which individual v has each of a set of d attributes or opinions. Then two individuals u and v are assumed to be friends, that is, they are connected in the graph model, if and only if $\mathbf{a}^u \cdot \mathbf{a}^v \geq t$, for some fixed, positive threshold t. The resulting graph is called a *d-dot product graph*.
We consider diversity and clustering in social networks by using a d-dot product graph model for the network. Diversity is considered through the size of the largest independent set of the graph, and clustering through the size of the largest clique. We present both positive and negative results on the potential of this model. We obtain a tight result for the diversity problem, namely that it is polynomial-time solvable for $d = 2$, but NP-hard for $d \geq 3$. We show that the clustering problem is polynomial-time solvable for $d = 2$. To our knowledge, these results are also the first on the computational complexity of combinatorial optimization problems on dot product graphs. We also give new insights into the structure of dot product graphs.
We also consider the situation when two individuals u and v are connected if and only if their preferences are not antithetical, that is, if and only if $\mathbf{a}^u \cdot \mathbf{a}^v \geq 0$, and the situation when two individuals u and v are connected if and only if their preferences are neither antithetical nor "orthogonal", that is, if and only if $\mathbf{a}^u \cdot \mathbf{a}^v > 0$. For these two cases we prove that the diversity problem is polynomial-time solvable for any fixed d and that the clustering problem is polynomial-time solvable for $d \leq 3$.

Keywords. social network; d-dot product graph; independent set; clique.

1 Introduction

Social networks are often modeled by a graph in order to use advanced algorithmic (or statistical) tools. Indeed, there is a large body of literature on (random)

* An extended abstract of this paper has appeared in the Proceedings of ISAAC 2013 [19].

graph models for social networks (see, for example, the surveys by Newman [31] and Snijders [40]). These studies have proposed many models for social networks, offering different explanations of why connections are made in the network (see the partial overview in Liben-Nowell and Kleinberg [27]). For example, the models of Simon [39], Price [34], and Barabási and Albert [3] famously propose that if you have many friends, you are more likely to make further new friends. A similar idea was recently considered from an algorithmic perspective by Bhawalkar et al. [6].

We consider a different predictor for connections in a social network, namely the degree of similarity of attributes and opinions of different individuals. Generally, individuals with similar attributes or opinions are more likely to be connected. This is known as the *homophily principle* and is well-studied in sociological research (see, for example, the survey by McPherson et al. [29]). To model the attributes of an individual u, we can associate them with a vector $\mathbf{a}^u$, where an entry a_i^u expresses the extent to which u has an attribute or opinion i [42]. For example, a positive value of a_i^u could indicate that u likes item i, whereas a negative value suggests that u dislikes item i. We call this a *vector model*.

There are many ways to measure similarity using a vector model (see, for example, [1, 17, 23, 25, 42]). We will use the dot product as a similarity measure. This measure is closely related to the cosine measure, which was studied before by researchers in information retrieval and social networks (see e.g. [8, 9]). The dot product measure leads to the *dot product model* for social networks, which is defined as follows. Consider a social network that consists of a set V of individuals, together with a vector model $\{\mathbf{a}^u \mid u \in V\}$. Let

$$\mathrm{sim}(u, v) = \mathbf{a}^u \cdot \mathbf{a}^v = \sum_{i=1}^{d} a_i^u a_i^v.$$

If the similarity $\mathrm{sim}(u, v)$ is at least some specified *threshold* $t > 0$, then we view the preferences of u and v to be sufficiently close together for u and v to be connected, that is, to be friends within the network. This immediately implies a graph $G = (V, E)$, where $(u, v) \in E$ if and only if $\mathrm{sim}(u, v) \geq t$. Such a graph is called a *dot product graph* of *dimension* d, or a *d-dot product graph*. The vector model $\{\mathbf{a}^u \mid u \in V\}$ together with the threshold t is called a *d-dot product representation* of G.

The dot product model has a long tradition, both in the study of social networks (see, for example, Breiger [5]) and in (algorithmic) graph theory (see, for example, Reiterman et al. [35–37] and particularly Fiduccia et al. [12]). Below, we survey some of the recent work and how it relates to social networks.

The dot product graph as a model for social networks was formalized by Nickel [32], Young and Scheinerman [43, 44], Minton [30], and Scheinerman and Tucker [38]. In particular, these works consider a randomized version of the dot product model, where the dot product of two vectors constitutes the probability that an edge occurs between the corresponding vertices. This randomized version of the model fits in a long line of research on random graph models for social networks, such as the classic Erdös-Rényi graph model [11], the Kronecker graph model [25] and the multiplicative attribute graphs model [23] (which generalizes the Kronecker graph model). The random dot product graph model exhibits the main characteristics that one would expect from a model for social networks, such as the property that two vertices are more likely to be adjacent if they have a common neighbour, the small-world principle, and a power-law degree distribution [32]. Studies into the dot product model were also motivated by the work of Papadimitriou et al. [33] and Caldarelli et al. [7]. Moreover, dot product graphs share some ideas with low-complexity graphs [2].

Dot product graphs have been studied from the perspective of (algorithmic) graph theory mostly with respect to the question of determining the *dot product dimension* of a graph: the minimum dimension d for which a graph has a d-dot product representation. This notion is well defined, as every graph on m edges has a dot product representation of dimension m [12]. Observe that, in the context of social networks, the dot product dimension can be seen as the smallest number of preferences needed to determine all friendship relations and non-relations between any two individuals in the network. Hence, the dot product dimension is a measure of the social complexity of a network [30].

The work of Fiduccia et al. [12] implies that deciding whether a graph has dot product dimension 1 takes polynomial time. However, Kang and Müller [21] showed the problem of deciding whether a graph has dot product dimension d is NP-hard for all fixed $d \geq 2$ (membership in NP is still open). They also proved that an exponential number of bits is sufficient and can be necessary to store a d-dot product representation of a dot product graph. Kang et al. [20] gave a tight bound of 4 on the dot product dimension of a planar graph. Fiduccia et al. [12] conjectured that any graph on n vertices has dot product dimension at most $\frac{n}{2}$; Li and Chang [26] recently confirmed this conjecture for a number of graph classes.

In this paper, we study how the complexity of computing structural properties of a social network is influenced by the complexity of the network's dot product model. Note that many standard structural properties, such as the graph diameter and the clustering coefficient, are easy to compute even on general graphs. Therefore, we consider two more advanced structural properties that give information on diversity and clustering in the network. These properties relate to classic graph optimization problems that are NP-hard to compute on general graphs, but whose computational complexity on dot product graphs was unknown. In fact, to the best of our knowledge (see also Spinrad [41, p. 309]), no algorithmic work on graph optimization problems on d-dot product graphs for $d \geq 2$ has been done prior to this work.

The main observation from our study is that when computing information on diversity and clustering properties of a social network, it is helpful if the network has small dot product dimension. When the network has small dot product dimension, we give positive results, in the sense of polynomial-time algorithms, for the studied problems. When the network does not have small dot product dimension, we observe clear barriers that prevent us from generalizing our algorithms. Additionally, we give a hardness result for one of the problems. This furthers our understanding of the scope of this particular model for social networks, but more importantly suggests that future studies on dot product models should focus on investigating approximation or fixed-parameter algorithms for the studied problems.

This main observation is supported by the following results. First, we consider diversity, by finding (the size of) a largest group of individuals in the network that are different-minded, and thus pairwise disconnected. This corresponds to the well-known Independent Set problem, which is NP-hard on general graphs [22]. On 1-dot product graphs the problem is known to be solvable in polynomial time, since such graphs consist of at most two connected components, each of which is a threshold graph [12], and Independent Set has a trivial polynomial-time algorithm for threshold graphs[3]. However, its complexity on d-dot product graphs for $d \geq 2$ is open. We settle this by proving that Independent Set is polynomial-time solvable on 2-dot product graphs, but becomes NP-hard on 3-dot product graphs.

Second, we consider clustering, by finding (the size of) a largest group of individuals in the network that are like-minded, and thus pairwise connected. This corresponds to the well-known Clique problem, which is NP-hard on general graphs [22]. Again, on 1-dot product graphs a trivial polynomial-time algorithm is known using the relation to threshold graphs [12], but its complexity has not been analyzed on d-dot product graphs for $d \geq 2$. We give initial insights into the complexity of this problem and show that it is polynomial-time solvable on 2-dot product graphs.

We remark that our complexity results depend on a number of lemmas on the structure of dot product graphs which are of independent interest.

To complement these results, we consider two variants of the dot product model. For the first variant, we model the scenario in which two individuals are connected if their preferences are not antithetical. That is, consider the graph where two individuals u, v are connected if and only if $\mathbf{a}^u \cdot \mathbf{a}^v \geq 0$. We call such a graph a *d^0-dot product graph*. Recall that in d-dot product graphs, the threshold t for connectivity must be greater than zero, and hence the definition of d^0-dot product graphs is different. Moreover, the structure of d^0-dot product graphs is substantially different from that of d-dot product graphs. To illustrate this, we prove that Independent Set is polynomial-time solvable on d^0-dot product graphs for any fixed d and that Clique is polynomial-time solvable if $d \leq 3$.

For the second variant, we model the situation in which two individuals are connected in the model if their preferences are neither antithetical nor orthogonal. Consider the graph that is obtained when two vertices u, v are adjacent if and only if $\mathbf{a}^u \cdot \mathbf{a}^v > 0$. We call this a *$d^+$-dot product graph*. It follows from

[3] A possible definition of a threshold graph states that G is a threshold graph if it can be constructed from a single vertex by repeatedly adding an isolated vertex or a dominating vertex (that is, a vertex adjacent to all other vertices) [15, 28]. Using this definition, a polynomial-time algorithm for Independent Set (and for Clique) can be easily derived.

Setting	INDEPENDENT SET	CLIQUE
d-DPG (≥ 1)	in P for $d \leq 2$ NP-hard for $d \geq 3$	in P for $d \leq 2$? for $d \geq 3$
d^0-DPG (≥ 0)	in P for $d \geq 0$	in P for $d \leq 3$? for $d \geq 4$
d^+-DPG (> 0)	in P for $d \geq 0$	in P for $d \leq 3$? for $d \geq 4$

Table 1. An overview of our results for the problems INDEPENDENT SET and CLIQUE on d-dot product graphs (the first row), d^0-dot product graphs (the second row), and d^+-dot product graphs (the third row), respectively, for fixed dimension d.

Fiduccia et al. [12] that the graph class where two vertices are adjacent if and only if $\mathbf{a}^u \cdot \mathbf{a}^v > t$ for some $t > 0$ is equivalent to the class of d-dot product graphs. However, we prove that the structure of d^+-dot product graphs is different from that of d-dot product graphs and that of d^0-dot product graphs. Still, we can show that INDEPENDENT SET is polynomial-time solvable on d^0-dot product graphs for any fixed d, as is CLIQUE when $d \leq 3$.

We provide an overview of our results in Table 1.

Organization. In Section 3, we prove several structural results about d-dot product graphs. In Section 4, we consider the complexity of INDEPENDENT SET and CLIQUE on dot product graphs. In Section 5, we study the computational complexity of these problems on d^0-dot product graphs and d^+-dot product graphs.

2 Preliminaries

All graphs that we consider are finite, undirected, and have neither loops nor multiple edges. For undefined graph terminology we refer to Diestel [10].

Let $G = (V, E)$ be a graph. We denote the neighbourhood of a vertex $u \in V$ by $N(u) = \{v \mid (u, v) \in E\}$. A subset $U \subseteq V$ is *independent* if no two vertices in U are joined by an edge, and U is a *clique* if every two vertices of U are adjacent. Given $U \subseteq V$, $G[U]$ denotes the subgraph of G induced by U, that is, it has vertex set U and an edge between two vertices of U if and only if G has an edge between them. The *complement* of G has vertex set V and an edge between two distinct vertices if and only if these vertices are not adjacent in G.

A graph is a *comparability graph* if there exists an assignment of exactly one direction to each of its edges such that (a, c) is a directed edge whenever (a, b) and (b, c) are directed edges. The complement of a comparability graph is called a *co-comparability graph.*

A graph is *p-partite* if its vertex set can be partitioned into p independent sets (some of which may be empty). If $p = 2$, then the graph is called *bipartite.* The complement of a p-partite graph is called a co-p-partite graph. Observe that the vertex set of a co-p-partite graph can be partitioned into at most p cliques. The complement of a bipartite graph is called *co-bipartite.*

The *length* of a cycle is its number of edges. The *girth* of a graph is the length of its shortest induced cycle.

Throughout the paper, we assume a computational model where operations on vectors and numbers take constant time. In particular, we assume that simple operations such as addition, multiplication, and division take constant time. Since we never use complex operations, such as cosines or square roots, we believe that this is a reasonable assumption. Moreover, it enables a cleaner presentation of the results.

3 Structure of d-Dot Product Graphs

In this section, we describe some of the structure that can be found in d-dot product graphs and which we need in our algorithms later on. Fiduccia et al. [12, Theorem 20] proved that 1-dot product graphs have at most two nontrivial components, each of which are threshold graphs. We show that d-dot product graphs, and in particular 2-dot product graphs, exhibit similar interesting structural properties.

From now we assume that $d \geq 2$. The reason for doing this is that our polynomial-time results on Independent Set and Clique in Section 4 for the case $d = 2$ readily carry over to the case $d = 1$: we can represent a $(d-1)$-dot product graph as a d-dot product graph for all $d \geq 2$ by adding a zero entry to all vectors of any of its $(d-1)$-dot product representations.

We call a d-dot product representation of a graph *clean* if it contains no two vectors $\mathbf{a}^u$ and $\mathbf{a}^v$ with $\mathbf{a}^u = \gamma \mathbf{a}^v$ for some $\gamma \geq 0$.

Lemma 1. *Given a d-dot product graph G without isolated vertices and a d-dot product representation of G, we can compute a clean d-dot product representation of G in polynomial time.*

Proof. Let $G = (V, E)$ be a d-dot product graph, and let $\{\mathbf{a}^u \mid u \in V\}$ be a d-dot product representation of G. Let t be the threshold. We consider the vertices of G consecutively and do as follows. Let $u \in V$. If there is no vertex $v \neq u$ such that $\mathbf{a}^u = \gamma \mathbf{a}^v$ for some $\gamma \geq 0$, then we do not modify $\mathbf{a}^u$. Otherwise, let

$$\delta = \frac{\min_w |t - \mathbf{a}^u \cdot \mathbf{a}^w|}{t},$$

where w ranges over all vertices (except u) that do not neighbor u. Should u be adjacent to all other vertices, we define $\delta = 1$. Note that, in both cases, $\delta > 0$. Let $\mathbf{b}^u = (1+\delta)\mathbf{a}^u$. Then for any vertex $w \neq u$:

$$\mathbf{b}^u \cdot \mathbf{a}^w = (1+\delta)(\mathbf{a}^u \cdot \mathbf{a}^w)$$

By the choice of δ, observe that $\mathbf{b}^u \cdot \mathbf{a}^w > t$ if $\mathbf{a}^u \cdot \mathbf{a}^w \geq t$, and $\mathbf{b}^u \cdot \mathbf{a}^w < t$ if $\mathbf{a}^u \cdot \mathbf{a}^w < t$ for any vertex $w \neq u$; in particular $\mathbf{b}^u \cdot \mathbf{a}^w \neq t$ for any $w \neq u$.

Now let $\epsilon = (\min_w |t - \mathbf{b}^u \cdot \mathbf{a}^w|)/\beta$, where w ranges over all vertices except u and where $\beta > 1$ is a sufficiently large constant (we will explain how to choose

β later). Note that $\epsilon > 0$, because $\mathbf{b}^u \cdot \mathbf{a}^w \neq t$ for all $w \neq u$. Let $\mathbf{e}^u$ be such that $e_i^u = \epsilon^u/(dm)$, where $m = \max_w \max_j |a_j^w|$. Note that, for all $w \neq u$,

$$(\mathbf{b}^u + \mathbf{e}^u) \cdot \mathbf{a}^w = \mathbf{b}^u \cdot \mathbf{a}^w + \mathbf{e}^u \cdot \mathbf{a}^w,$$

and therefore, $(\mathbf{b}^u+\mathbf{e}^u)\cdot\mathbf{a}^w \geq t$ if and only if $\mathbf{a}^u\cdot\mathbf{a}^w \geq t$ for any $w \neq u$. Hence, by setting $\mathbf{a}^u$ to $\mathbf{b}^u + \mathbf{e}^u$, we retain a d-dot product representation of G. Moreover, by choosing β sufficiently large we find that there is no $v \neq u$ with $\mathbf{a}^u = \gamma\mathbf{a}^v$ for some $\gamma \geq 0$. Clearly, this representation can be obtained in polynomial time. As mentioned, we iteratively apply this procedure to all u to obtain a clean representation in polynomial time (note that we only appropriately adjust the vector corresponding to the vertex u that is under consideration and hence the occurrence of a new pair of vectors in which one vector is a scalar multiple of the other is not possible). □

Throughout the remainder of this section, we assume that we are given a d-dot product graph $G = (V, E)$ for some $d \geq 2$ together with a d-dot product representation with vectors $\{\mathbf{a}^u \mid u \in V\}$ and threshold t. For solving Independent Set and Clique, we can preprocess G by removing any isolated vertices. Hence, by Lemma 1, we may assume without loss of generality that the given representation is clean.

In the remainder of the paper, whenever we are given a representation of a dot product graph, we do not always distinguish between a vertex and the corresponding vector and so may, for example, speak about the length of a vertex without confusion.

We introduce several notations for vertices u and v to measure the angle between $\mathbf{a}^u$ and $\mathbf{a}^v$ in the plane defined by $\mathbf{a}^u$ and $\mathbf{a}^v$. We assume a fixed direction of rotation, so that we can speak of the angle *from* $\mathbf{a}^u$ *to* $\mathbf{a}^v$. This angle, denoted by ϕ_{uv}, thus is always positive. Then we define θ_{uv} so that $\theta_{uv} = \phi_{uv}$ if $\phi_{uv} \leq \pi$ and $\theta_{uv} = \phi_{uv} - 2\pi$ otherwise. Note that this implies that $-\pi < \theta_{uv} \leq \pi$ and $\theta_{uv} = -\theta_{vu}$ (unless $\theta_{uv} = \theta_{vu} = \pi$).

We say that a vertex u is *short* if $||\mathbf{a}^u|| \leq \sqrt{t}$; otherwise, it is *long*. Note that we can decide whether u is short in polynomial time by checking whether $||\mathbf{a}^u||^2 \leq t$. We first provide two lemmas about short vertices.

Lemma 2. *Let v be a short vertex. Then $G[N(v)]$ is co-2^{d-1}-partite.*

Proof. We may assume that the representation is rotated such that $\mathbf{a}_1^v = z$ for some $0 < z \leq \sqrt{t}$ and that $\mathbf{a}_i^v = 0$ for all $i = 2, \ldots, d$, i.e., that $\mathbf{a}^v$ is the d-dimensional unit vector scaled by some $z > 0$. Observe that $u \in N(v)$ if and only if $\mathbf{a}_1^u \geq t/z \geq \sqrt{t}$. Associate with each vertex $u \in N(v)$ a $(d-1)$-dimensional sign vector $\boldsymbol{s}^u$, where $\boldsymbol{s}_i^u = 1$ if $\mathbf{a}_{i+1}^u \geq 0$ and $\boldsymbol{s}_i^u = -1$ otherwise for $i = 1, \ldots, d-1$. Observe that the sign-vectors naturally partition the vertices of $N(v)$ into 2^{d-1} equivalence classes. Moreover, any two vertices u, w in an equivalence class are adjacent, because $\mathbf{a}^u \cdot \mathbf{a}^w = \sum_{i=1}^d \mathbf{a}_i^u\mathbf{a}_i^w \geq t + \sum_{i=2}^d \mathbf{a}_i^u\mathbf{a}_i^w \geq t$, as $\mathbf{a}_i^u \geq 0$ if and only if $\mathbf{a}_i^w \geq 0$ for any $i = 2, \ldots, d$. Therefore, each equivalence class induces a clique, and thus $G[N(v)]$ is co-2^{d-1}-partite. □

Lemma 2 shows in particular that $G[N(v)]$ is co-bipartite if $d = 2$.

Lemma 3. *The set of short vertices is an independent set.*

Proof. For any two vertices v and w, we have $\boldsymbol{a}^v \cdot \boldsymbol{a}^w = \|\boldsymbol{a}^v\| \, \|\boldsymbol{a}^w\| \cos\theta_{vw}$. Assume that v and w are short. As we assume that our d-dot product representation of G is clean, $\cos\theta_{vw} < 1$ which, combined with the definition of short, implies that $\boldsymbol{a}^v \cdot \boldsymbol{a}^w < t$, and hence the vertices are not adjacent. □

We say that a vertex v is *between* vertices u and w if $\boldsymbol{a}^v$ can be written as a nonnegative linear combination of $\boldsymbol{a}^u$ and $\boldsymbol{a}^w$. In other words, v is between u and w if $\mathbf{a}^v$ lies in the plane defined by $\mathbf{a}^u$ and $\mathbf{a}^w$ and $\boldsymbol{a}^v$ lies within the smaller of the two angles defined by $\boldsymbol{a}^u$ and $\boldsymbol{a}^v$ in this plane.

We require a result that in the case that $t = 1$ is implied by Lemma 28 of Fiduccia et al. [12]. The generalization to all t can easily be obtained by copying their proof, so here we will state it without proof.

Lemma 4. *Suppose $d = 2$. Let u, v, and w be vertices such that v is between u and w. If u is adjacent to w, and v is adjacent to neither u nor w, then v is short.*

We now present two lemmas about the neighbourhoods of vertices.

Lemma 5. *Let $L = \{u \in V \mid \|\boldsymbol{a}^u\| > \sqrt{t}\}$. If $d = 2$, then $G[N(v) \cap L]$ is a co-comparability graph for all $v \in V$.*

Proof. Number the vertices $u \in N(v) \cap L$ by increasing value of θ_{vu}. Let u_i denote the ith vertex in the order. Consider some $i < j < k$ such that $(u_i, u_k) \in E$. Note that $\boldsymbol{a}^{u_j}$ is between $\boldsymbol{a}^{u_i}$ and $\boldsymbol{a}^{u_k}$. Since $u_j \in L$, it follows from Lemma 4 that one of $(u_i, u_j), (u_k, u_j) \in E$. The existence of such an ordering implies that $G[N(v) \cap L]$ is a co-comparability graph, due to Kratsch and Stewart [24]. □

Lemma 6. *Let $u, v, w \in V$ be such that v is between u and w. If u is adjacent to w and $\|\boldsymbol{a}^v\| \geq \|\boldsymbol{a}^w\|$, then u is adjacent to v.*

Proof. Without loss of generality θ_{uw} is positive. Notice that $\theta_{uv} < \theta_{uw}$, because v is between u and w, and thus $\cos\theta_{uv} > \cos\theta_{uw}$. Hence,

$$\begin{aligned}
\boldsymbol{a}^u \cdot \boldsymbol{a}^v &= \|\boldsymbol{a}^u\| \, \|\boldsymbol{a}^v\| \cos\theta_{uv} \\
&> \|\boldsymbol{a}^u\| \, \|\boldsymbol{a}^w\| \cos\theta_{uw} \\
&= \boldsymbol{a}^u \cdot \boldsymbol{a}^w \\
&\geq t,
\end{aligned}$$

and so u and v are adjacent. □

4 Diversity and Clustering in Social Networks

In this section, we consider the complexity of computing diversity and clustering in social networks through Independent Set and Clique, respectively, on a dot product graph model of the network. We first prove that Independent Set is polynomial-time solvable if $d \leq 2$ and NP-hard if $d \geq 3$. We then prove that Clique is polynomial-time solvable if $d \leq 2$.

As before, throughout we have a d-dot product graph $G = (V, E)$ and a clean d-dot product representation with vectors $\{\mathbf{a}^u \mid u \in V\}$ and threshold t.

We first consider Independent Set in the case $d \leq 2$. Recall that we may assume without loss of generality that $d = 2$. Armed with the structural results of the previous section, we can prove the following theorem.

Theorem 1. Independent Set *is solvable in $O(n^3)$ time on 2-dot product graphs on n vertices, when a representation of the graph is known.*

Proof. Let G be a 2-dot product graph. We describe how to find a maximum-size independent set of G. In fact, we will describe how to find, for each long vertex u of G, an independent set of G that has largest size over all independent sets of G that contain u. This is sufficient as the maximum-size independent set of G is either the largest of these $O(n)$ sets or the set of all short vertices which is also independent by Lemma 3; we use this latter fact repeatedly in the proof.

So let u be a fixed long vertex of G. Let G_u be the graph obtained by removing all vertices that neighbour u and their incident edges. Note that u is in every maximum-size independent set of G_u. Hence, if we can find a maximum-size independent set of G_u, then we will have found an independent set of G that has largest size over all independent sets of G that contain u.

We define a total (or linear) ordering $\prec$ of the vertices of G_u by ordering the vertices by increasing angle ϕ_{uv} of their vector representation from $\boldsymbol{a}^u$. Using the square of the cosine formula, $\prec$ can be computed in quadratic time using just dot-products.

Claim 1: Let v and w be adjacent vertices in G_u such that θ_{vw} is positive. Then $v \prec w$.

Proof. Any vertex between v and w is, by Lemma 4, either short or adjacent to one of them. Since we have removed all neighbours of u from G to obtain G_u, we find that v and w are not neighbours of u in G. Moreover, u is long. Hence, u cannot be between v and w, and thus $\phi_{uv} < \phi_{uw}$. Therefore, $v \prec w$. This proves Claim 1.

An intuitive understanding of the statement of Claim 1 would be that no edge can 'jump over' $\mathbf{a}^u$. This gives G_u a linear structure, instead of the circular structure of G. Our algorithm exploits this linear structure to find a maximum-size independent set.

We now relate the ordering $\prec$ to betweenness.

Claim 2: Let v, w, x be vertices in G_u where v and w are adjacent. Then x is between v and w and θ_{vw} is positive if and only if $v \prec x \prec w$.

Proof. One direction is an extension of Claim 1 and the other is trivially true.

For a long vertex v in G_u, let $J(v)$ be a largest independent set in the subgraph of G_u that contains all vertices up to and including v in the ordering $\prec$ such that $v \in J(v)$. Let $j(v) = |J(v)|$. For a pair of long vertices v and w in G_u with $w \prec v$, let $S(w, v)$ be the set of vertices x such that x is short, $w \prec x \prec v$ and x is not adjacent to either v or w. Let $s(w, v) = |S(w, v)|$.

Claim 3: For each pair of non-adjacent long vertices v and w with $w \prec v$ in G_u, $j(v) \geq j(w) + s(w, v) + 1$.

Proof. Note that the claim will follow if we can show that $J(w) \cup S(w, v) \cup \{v\}$ is an independent set. All we need to show is that no vertex in $S(w, v) \cup \{v\}$ is adjacent to a vertex in $J(w)$.

Suppose that v is adjacent to a vertex x in $J(w)$. We know v and w are not adjacent so $x \neq w$ and $x \prec w \prec v$. Hence, w is between x and v (by Claim 2), and the adjacency of x and v implies, by Lemma 4, that w is short; a contradiction.

If a vertex $y \in S(w, v)$ is adjacent to any vertex x in $J(w)$, then $x \neq w$ by the definition of $S(w, v)$. But x is adjacent to w using Lemma 6 and noting that w is long, y is short and w is between x and y. This contradiction proves Claim 3.

Claim 4: For each long vertex $v \neq u$ in G_u, $j(v)$ is the maximum, over all long vertices w with $w \prec v$ and v and w non-adjacent, of $j(w) + s(w, v) + 1$.

Proof. Note that the set of long vertices that precede v includes the isolated vertex u so the maximum is well-defined, and the previous claim tells us that $j(v)$ is no less than this maximum. We must show that it is no larger. Let w be the long vertex that is last in the ordering amongst all long vertices in $J(v) \setminus \{v\}$ (as $J(v)$ contains u we can always find such a vertex). The subset of $J(v)$ containing only w and preceding vertices is independent and contains at most $j(w)$ vertices. The only other vertices in $J(v)$ are short vertices between w and v and v itself. Thus $j(v) \leq j(w) + s(w, v) + 1$, and Claim 4 is proved.

Note that j can easily be computed since $j(u) = 1$, and Claim 4 tells us that if we consider the vertices in order we can find the remaining values.

For each long vertex v in G_u, let $S^+(v)$ contain each vertex w such that w is short, $v \prec w$ and v is not adjacent to w. Let $s^+(v) = |S^+(v)|$. Let m be the maximum, over all long vertices v in G_u, of $j(v) + s^+(v)$.

Claim 5: Let J be a maximum-size independent set in G_u. Then $|J| = m$.

Proof. Let v be a long vertex in G_u. We shall show that $J(v) \cup S^+(v)$ is an independent set. Let w be a vertex in $S^+(v)$ and suppose that x is a vertex in $J(v)$ adjacent to w. By the definition of $S^+(v)$, we have $x \neq v$, so $x \prec v \prec w$. By Claim 2, v is between x and w and, by Lemma 4, v is either short or adjacent to x or w. This contradiction shows that $J(v) \cup S^+(v)$ is an independent set. So $|J| \geq j(v) + s^+(v)$ for all long vertices v and hence $|J|$ is at least m.

Now let z be the long vertex in J that is latest in the ordering. Let J_1 be the subset of J containing z and preceding vertices. Hence, $|J_1| \leq j(z)$. The vertices of $J \setminus J_1$ are short vertices later than z in the ordering, so there are at most $s^+(z)$ of them. Thus $|J| \leq j(z) + s^+(z) \leq m$, and Claim 5 is proved.

We omit the details but it is straightforward to show that j and s^+, and so also m, can be computed in $O(n^2)$ time. The corresponding sets of vertices, and thus a maximum-size independent set of G_u, can also be found. By repeating for each u, a maximum-size independent set of G is found in time $O(n^3)$. □

We contrast this positive result with the following result.

Theorem 2. *For any $d \geq 3$,* Independent Set *is NP-hard on d-dot product graphs, when a representation of the graph is not known.*

Proof. Recall that an s-subdivision of an edge is the operation in which the edge is replaced by an $(s+1)$-edge path (implying that s new vertices are created).

Claim: If Independent Set is NP-hard on some graph class $\mathcal{G}$, it is NP-hard on the graph class obtained by 2-subdividing each edge of each graph of $\mathcal{G}$.

Proof. Let G be a graph and let G' be the graph obtained by 2-subdividing some edge (u, v) of G. Let x, y be the new vertices, where y is adjacent to v. Then any independent set I of G can be turned into an independent set of G' of size $|I| + 1$ by adding either x or y (depending on whether $v \in I$ or not, respectively). Conversely, any (non-empty) independent set I' of G' corresponds to an independent set of G of size $|I'| - 1$ by restriction; observe that if both u and v are in I' then neither x nor y belongs to I', so we can remove one of u, v to obtain an independent set of size $|I'| - 1$.

Applying the above idea iteratively shows that a graph G has an independent set of size k if and only if the graph obtained by 2-subdividing each edge of G has an independent set of size $k + |E(G)|$. This yields the required NP-hardness reduction, and proves the claim.

Since Independent Set is NP-hard on planar graphs [13], the claim implies that it remains NP-hard on 2-subdivisions of planar graphs. Note that such graphs are planar and have girth at least 9. Kang et al. [20] observed that planar graphs of girth at least 5 are 3-dot product graphs. So Independent Set is NP-hard on the class of 3-dot product graphs, which is a subclass of d-dot product graphs for all $d > 3$. □

The structural results of the previous section provide enough structure to solve Clique in polynomial time on 2-dot product graphs.

Theorem 3. Clique *is solvable in $O(n^4)$ time on 2-dot product graphs on n vertices, even when no representation of the graph is known.*

Proof. Call a vertex v *weak* if $N(v)$ is co-bipartite. Note that we can determine whether a vertex v is weak in quadratic time by attempting to 2-color the complement. We first find a largest clique that contains a weak vertex.

Claim: Clique can be solved in $O(n^{2.5})$ time on co-bipartite graphs.

Proof. To solve CLIQUE on co-bipartite graphs, it suffices to solve INDEPENDENT SET on bipartite graphs, because any clique in a graph corresponds to an independent set in the complement of the graph. To solve INDEPENDENT SET on bipartite graphs, it suffices to solve VERTEX COVER[4] on bipartite graphs, because the complement of an independent set of a graph is a vertex cover and vice versa. To solve VERTEX COVER on bipartite graphs, it suffices to solve MATCHING[5] on bipartite graphs, because of König's Theorem (see, for example, [10]). Solving MATCHING on bipartite graphs takes $O(n^{2.5})$ time [18]. The claim follows.

Following the claim, we can find a largest clique containing a weak vertex in $O(n^{3.5})$ time.

We now find a largest clique that only contains vertices that are not weak. Let L be the set of vertices that are not weak. By Lemma 2, each vertex of L must be long in each 2-dot product representation of the graph. Hence, for any $v \in L$, $G[N(v) \cap L]$ is a co-comparability graph by Lemma 5. We now observe that CLIQUE on co-comparability graphs is INDEPENDENT SET on comparability graphs. The latter problem can be reduced to a maximum-flow computation [14], which takes $O(n^3)$ time. Hence, we can find this largest clique in $O(n^4)$ time.

Finally, we return the largest of the two cliques that we found. Observe that the algorithm only uses the graph and does not require a representation to be given. □

5 Structure and Complexity for Variants of the Model

In this section, we consider two variants of the dot product graph model, which model that two individuals are connected if and only if their preferences are not antithetical, or are neither antithetical nor orthogonal. In the introduction, we defined the d^0-dot product graph and the d^+-dot product graph model for these cases. Recall that if $\{\mathbf{a}^u \mid u \in V\}$ is a representation of $G = (V, E)$, then

- $(u, v) \in E$ if and only if $\mathbf{a}^u \cdot \mathbf{a}^v \geq 0$ when G is a d^0-dot product graph, and
- $(u, v) \in E$ if and only if $\mathbf{a}^u \cdot \mathbf{a}^v > 0$ when G is a d^+-dot product graph.

We study the complexity of computing the diversity and clustering on these models, that is, of INDEPENDENT SET and CLIQUE, on d^0-dot product graphs and d^+-dot product graphs.

Throughout this section, we assume without loss of generality that all vectors in a representation have *non-zero* length. Observe that vertices whose corresponding vectors have length 0 are adjacent to all other vertices in a d^0-dot product graph and are isolated in a d^+-dot product graph. Hence, for solving

[4] A *vertex cover* of a graph G is a subset of its vertices such that at least one endpoint of each edge is in the subset. Then VERTEX COVER asks to find a vertex cover of smallest size.

[5] A *matching* of a graph is a subset of its edges such that no two edges in the subset share an endpoint. Then MATCHING asks to find a matching of largest size.

Independent Set and Clique, we can preprocess the instance by removing any such vertices.

First, we describe the structure of independent sets in d^0-dot product graph. The following lemma is equivalent to Lemma 18 of Fiduccia et al. [12].

Lemma 7. *For all $d \geq 1$, every independent set in a d^0-dot product graph has size at most $d+1$.*

Independent sets in d^+-dot product graphs have a different structure.

Lemma 8. *For all $d \geq 1$, every independent set in a d^+-dot product graph has size at most $2d$.*

Proof. Recall that we assume that at all vectors in a representation have non-zero length. Observe that the statement of the lemma is equivalent to stating that the maximum number of (non-zero) vectors in $\mathbb{R}^d$ with pairwise non-positive dot products is at most $2d$.

We apply induction on d. Let $G = (V, E)$ be a d^+-dot product graph with representation $\{\mathbf{a}^u \mid u \in V\}$. The lemma is readily seen to hold for $d = 1$.

Let $d \geq 2$ and suppose that the claim holds for dimension $d-1$. Let $I = \{u_1, \ldots, u_p\}$ for some $p \geq 1$ be an independent set of G. Without loss of generality, $\mathbf{a}^{u_1} = (1, 0, \ldots, 0)$. Consider the $(d-1)^+$-dot product graph $G' = (V', E')$ obtained from G by removing all first coordinates from the vectors $\mathbf{a}$ and then removing vectors of zero length. We claim that $I \cap V'$ is an independent set in G'. To see this, let $v, w \in I \cap V'$. Since v, w are independent of u_1, $\mathbf{a}_1^v, \mathbf{a}_1^w \leq 0$, and thus $\mathbf{a}_1^v \mathbf{a}_1^w \geq 0$. As $v, w \in I$, $\mathbf{a}^v \cdot \mathbf{a}^w \leq 0$ and thus $\sum_{i=2}^d \mathbf{a}_i^v \mathbf{a}_i^w \leq -(\mathbf{a}_1^v \mathbf{a}_1^w) \leq 0$. Hence, $\sum_{i=2}^d \mathbf{a}_i^v \mathbf{a}_i^w \leq 0$. Therefore, $I \cap V'$ is indeed an independent set.

By induction, we find that $|I \cap V'| \leq 2d - 2$. Notice that the only vertices w that are in G but not in G' are those for which $\mathbf{a}_i^w = 0$ for $i = 2, \ldots, d$. Suppose that $I \setminus \{u_1\}$ contains two such vertices, say v, w. They must satisfy $\mathbf{a}_1^v, \mathbf{a}_1^w < 0$ in order to be independent from u_1. It follows that $\mathbf{a}^v \cdot \mathbf{a}^w > 0$ and thus that v and w are adjacent, a contradiction. Therefore $I \setminus \{u_1\}$ can contain at most one vertex that is in G but not in G'. Hence $|I| \leq |I \cap V'| + 2 \leq 2d$. □

The proofs of Lemmas 7 and 8 can be turned into constructions to show that the given bounds are tight. The lemmas show that d^0-dot product graphs and d^+-dot product graphs have different structure, which is also different from the structure of d-dot product graphs. Moreover, using exhaustive enumeration, the two lemmas immediately imply the following.

Theorem 4. *For all $d \geq 1$,* Independent Set *is solvable in $O(n^{d+1})$ time on d^0-dot product graphs and in $O(n^{2d})$ time on d^+-dot product graphs on n vertices, even when no representation of the graph is known.*

We now consider Clique on d^0-dot product graphs and d^+-dot product graphs. For $d = 2$, it suffices to observe that a set of vertices forms a clique if and only if their corresponding vectors lie in the nonnegative quadrant (after an appropriate rotation). However, this structural observation does not generalize to higher dimensions, as evident from the counterexamples by Gray and Wilson [16] for $d = 3$ and $d \geq 5$; see Appendix A for a counterexample for the case $d = 4$. Instead, we follow a different approach, which leads to a polynomial-time algorithm for all $d \leq 3$.

For any hyperplane h with normal $\mathbf{n}$, let h^+ be the half-space $\{p \mid p \cdot \mathbf{n} \geq 0\}$ and let h^- be the half-space $\{p \mid p \cdot \mathbf{n} \leq 0\}$. Note that any two vectors $\mathbf{a}, \mathbf{b}$ induce a hyperplane with normal $\mathbf{a} \times \mathbf{b}$, where $\times$ is the cross product operation. We refer to the monograph by Barvinok [4] for any undefined terminology on cones.

Theorem 5. *For all $d \leq 3$,* Clique *can be solved in $O(n^{4.5})$ time on d^0-dot product graphs and d^+-dot product graphs on n vertices, when a representation of the graph is known.*

Proof. We assume that $d = 3$ (fewer dimensions are a special case). Let $G = (V, E)$ be a 3^0-dot product graph or a 3^+-dot product graph with representation $\{\mathbf{a}^v \mid v \in V\}$. We note that if a basis change is applied, then the resulting vectors are still a representation of the same kind (3^0-dot product or 3^+-dot product) for G. We first give a structural result, where we essentially show that any clique C of G induces a basis such that the vectors of C lie in two octants with respect to this basis. Then, we give an algorithm that finds this basis for a maximum clique by guessing limited information about the clique, and use the basis to obtain a maximum clique of G.

We start with the structural result. Let C be any clique of G. Let $\mathcal{K}$ denote the conic hull of $\mathbf{a}^v$ for all vertices $v \in C$, that is, $\mathcal{K} = \{\sum_{v \in C} \lambda_v \mathbf{a}^v \mid \lambda_v \geq 0\}$. We call $\mathcal{K}$ the *cone corresponding to C*. The structural result considers the case that $\mathcal{K}$ is not a ray. Since $\mathcal{K}$ is generated by a finite set, its extreme rays are vectors that correspond to vertices of C. Let u be any vertex such that $\mathbf{a}^u$ spans an extreme ray of $\mathcal{K}$, and let h_u denote the hyperplane with normal $\mathbf{a}^u$. Because $\mathcal{K}$ is the conic hull of vectors corresponding to a clique, $\mathbf{p} \cdot \mathbf{a}^u \geq 0$ for any $\mathbf{p} \in \mathcal{K}$ (this is true both when G is a 3^0-dot product graph or a 3^+-dot product graph). Hence, $\mathcal{K} \subseteq h_u^+$.

Let w be any vertex such that $\mathbf{a}^w$ spans an extreme ray of $\mathcal{K}$ that is not spanned by u and such that the hyperplane h_{uw} induced by $\mathbf{a}^u$ and $\mathbf{a}^w$ contains a facet of $\mathcal{K}$. Since h_{uw} contains a facet of $\mathcal{K}$, either $\mathcal{K} \subseteq h_{uw}^+$ or $\mathcal{K} \subseteq h_{uw}^-$. Assume without loss of generality that $\mathcal{K} \subseteq h_{uw}^+$, and let $\mathbf{t}$ denote the normal of h_{uw} that lies in h_{uw}^+. Finally, let $\mathbf{w}'$ denote the projection of $\mathbf{a}^w$ onto h_u. By definition, $\mathbf{t}$, $\mathbf{a}^u$, $\mathbf{w}'$ are pairwise orthogonal. Moreover, as $\mathcal{K} \subseteq h_u^+ \cap h_{uw}^+$ and $h_u^+ \cap h_{uw}^+$ is the union of two octants in the basis induced by $\mathbf{t}$, $\mathbf{a}^u$, $\mathbf{w}'$, we find that $\mathcal{K}$ is a subset of two octants in the basis induced by $\mathbf{t}$, $\mathbf{a}^u$, $\mathbf{w}'$.

We now turn the insight of the structural result into an algorithm. The algorithm consists of two phases.

In the first phase of the algorithm, we ensure that we find a maximum clique if the cone corresponding to some maximum clique is a ray. Therefore, we iterate over all $v \in V(G)$ and find the set X of vertices u for which $\mathbf{a}^u$ spans the same ray as $\mathbf{a}^v$. The set X is a clique irrespective of whether G is a 3^0-dot product graph or a 3^+-dot product graph. We keep a maximum clique found over all choices of v.

In the second phase of the algorithm, we ensure that we find a maximum clique if the cone corresponding to some maximum clique is not a ray. Iterate over all n^2 ordered pairs (u, w) of the vertices of G such that $\mathbf{a}^u$ and $\mathbf{a}^w$ do not span the same ray. Define h_u as the plane with normal $\mathbf{a}^u$, and define h_{uw} as the plane induced by $\mathbf{a}^u$ and $\mathbf{a}^w$. Consider $h_u^+ \cap h_{uw}^+$ (we also consider $h_u^+ \cap h_{uw}^-$ in a similar way). Let $\mathbf{t}$ denote the normal of h_{uw} that lies in h_{uw}^+ and let $\mathbf{w}'$ denote the projection of $\mathbf{a}^w$ onto h_w. Note that $h_u^+ \cap h_{uw}^+$ is the union of two octants in the basis induced by $\mathbf{t}$, $\mathbf{a}^u$, $\mathbf{w}'$. As any octant induces a clique, $h_u^+ \cap h_{uw}^+$ induces a co-bipartite graph H. We can find H in linear time as the graph induced by the vertices whose corresponding vectors have positive or strictly positive dot product with both $\mathbf{a}^v$ and $\mathbf{t}$. Since H is co-bipartite, we can find a maximum clique of H in $O(n^{2.5})$ time, as it reduces to finding a maximum matching in a bipartite graph, which takes $O(n^{2.5})$ time [18]. We then keep a maximum clique over all choices of u, w. The output of the algorithm is a largest of the two cliques kept in the first and second phase.

Note that the algorithm runs in $O(n^{4.5})$ time, as claimed. To see correctness, let C be a maximum clique. If the cone corresponding to C is a ray, then the algorithm considers C in the first phase, and thus outputs a clique of size $|C|$. If the cone corresponding to C is not a ray, then by our structural result there will be a choice of u, w for which $u, w \in C$ and h_{uw} contains a facet of $\mathcal{K}$, where $\mathcal{K}$ is the cone corresponding to C. For this choice of u, w, the algorithm finds a clique of size $|C|$. Hence, our algorithm outputs a maximum clique of G. $\square$

6 Conclusions

This paper provided the first study of algorithms that compute diversity and clustering in social networks that are modeled as dot product graphs by solving Independent Set and Clique on these graphs. We focussed on classical complexity only, and both approximability and parameterized complexity remain largely unexplored.

Our exploration of the complexity of Clique on d-dot product graphs leaves further open problems. The current approach for $d = 2$ does not seem to extend to d-dot product graphs for $d \geq 3$, as our structural results (e.g., Lemma 2) seem to indicate that we need to solve Clique on co-p-partite graphs for $p \geq 3$. However, this problem is NP-hard, as Independent Set is NP-hard on 2-subdivisions of planar graphs (as shown in the proof of Theorem 2). Hence, further structural insight into d-dot product graphs is needed to resolve the complexity of Clique on these graphs.

We observe that our polynomial-time algorithms for Independent Set and Clique on 2-dot product graphs generalize well-known polynomial-time algorithms for these problems on interval graphs, because interval graphs have a 2-dot product representation [12, Theorem 21]. At the same time, we are unaware of any nontrivial superclasses of 2-dot product graphs, in particular for which Independent Set and Clique are polynomial-time solvable.

Another line of research that should be pursued is to find algorithms for Independent Set and Clique on d-dot product graph that do not require a representation to be given. Although for social networks the necessary data to obtain a representation should be readily available, this is not true in general. Given the difficulty in building a representation from the graph (see [21]), such representation-less algorithms could prove useful, and would be interesting from a theoretical point of view. We note that our algorithms for Clique on 2-dot product graphs and for Independent Set on d^0-dot product graphs and on d^+-dot product graphs already do not require a representation to be given.

Finally, we note that the dot product graph model of social networks might be able to capture more problems for social networks as graph optimization problems.

References

1. L.A. Adamic, E. Adar, Friends and neighbors on the Web, Social Networks 25 (2003) 211–230.
2. S. Arora, D. Steurer, A. Wigderson, Towards a Study of Low-Complexity Graphs, Proc. ICALP 2009, LNCS 5555 (2009) 119–131.
3. A.-L. Barabási and R. Albert, Emergence of Scaling in Random Networks, Science 286 (1999) 509–512.
4. A. Barvinok, A Course in Convexity, American Mathematical Society, 2003.
5. R.L. Breiger, The Duality of Persons and Groups, Social Forces 53 (1974) 181–190.
6. K. Bhawalkar, J.M. Kleinberg, K. Lewi, T. Roughgarden and A. Sharma, Preventing Unraveling in Social Networks: The Anchored k-Core Problem, Proc. ICALP 2012, LNCS 7392 (2012) 440–451.
7. G. Caldarelli, A. Capocci, P. de Los Rios, M.A. Muñoz, Scale-Free Networks from Varying Vertex Intrinsic Fitness, Phys. Rev. Lett. 89 (2002), 258702.
8. C. Cattuto, D. Benz, A. Hotho, G. Stumme, Semantic Grounding of Tag Relatedness in Social Bookmarking Systems, Proc. ISWC 2008, LNCS 5318 (2008) 615–631.
9. D.J. Crandall, D. Cosley, D.P. Huttenlocher, J.M. Kleinberg, S. Suri, Feedback effects between similarity and social influence in online communities, Proc. KDD 2008, ACM (2008) 160–168.
10. R. Diestel, Graph Theory, Springer-Verlag, 2005.
11. P. Erdös and A. Rényi, On random graphs. Publicationes Mathematicae Debrecen 6 (1959) 290–297.
12. C.M. Fiduccia, E.R. Scheinerman, A. Trenk and J.S. Zito, Dot product representations of graphs, Discrete Mathematics 181 (1998) 113–138.
13. M.R. Garey, D.S. Johnson and L. Stockmeyer, Some simplified NP-complete graph problems, Theor. Comp. Sci. 1 (1976) 237–267.
14. M.C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, Acad. Press, 1988.
15. M.C. Golumbic, A. Trenk, Tolerance Graphs, Cambridge University Press, 2004.
16. L.J. Gray and D.G. Wilson, Nonnegative factorization of positive semidefinite nonnegative matrices, Linear Algebra and its Applications 31 (1980) 119–127.
17. P.D. Hoff, A.E. Raftery, M.S. Handcock, Latent Space Approaches to Social Network Analysis, J. Am. Stat. Assoc. 97 (2002) 1090–1098.
18. J.E. Hopcroft, R.M. Karp, An $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs, SIAM J. Comput. 2 (1973) 225–231.

19. M. Johnson, D.Paulusma, E.J. van Leeuwen, Algorithms to measure diversity and clustering in social networks through dot product graphs, Proc. ISAAC 2013, LNCS 8283 (20-13) 130–140
20. R.J. Kang, L. Lovász, T. Müller and E.R. Scheinerman, Dot product representations of planar graphs. Electr. J. Comb. 18 (2011).
21. R.J. Kang and T. Müller, Sphere and dot product representations of graphs, Discrete and Computational Geometry 47 (2012) 548–568.
22. R.M. Karp, Reducibility among Combinatorial Problems", Complexity of Computer Computations, Plenum Press (1972) 85–103.
23. M. Kim, J. Leskovec, Multiplicative Attribute Graph Model of Real-World Networks, Proc. WAW 2010, LNCS 6516 (2010) 62–73.
24. D. Kratsch and L. Stewart, Domination on cocomparability graphs, SIAM J. Disc. Math. 6 (1993) 400–417.
25. J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos and Z. Ghahramani, Kronecker graphs: an approach to modeling networks, Journal of Machine Learning Research 11 (2010) 985–1042.
26. B. Li and G.J. Chang, Dot product dimensions of graphs, Discrete Applied Mathematics 166 (2014) 159–163.
27. D. Liben-Nowell, J. Kleinberg, The Link Prediction Problem for Social Networks, Proc. CIKM 2003 (2003) 556–559.
28. N.V.R. Mahadev, U.N. Peled, Threshold Graphs and Related Topics, Annals of Discrete Mathematics 56, North Holland, 1995.
29. M. McPherson, L. Smith-Lovin, J.M. Cook, Birds of a Feather: Homophily in Social Networks, Annual Rev. Sociology 27 (2001) 415–444.
30. G. Minton, Dot Product Representations of Graphs, PhD dissertation, Harvey Mudd College, 2008.
31. M.E.J. Newman, The Structure and Function of Complex Networks, SIAM Review 45 (2003) 167–256.
32. C.M.L. Nickel, Random Dot Product Graphs: A Model for Social Networks, PhD dissertation, Johns Hopkins University, 2007.
33. C.H. Papadimitriou, P. Raghavan, H. Tamaki, S. Vempala, Latent Semantic Indexing: A Probabilistic Analysis, J. Comput. Syst. Sci. 61 (2000) 217–235.
34. D.J. de S. Price, A general theory of bibliometric and other cumulative advantage processes, J. American Society for Information Science 27 (1976) 292–306.
35. J. Reiterman, V. Rödl and E. Šiňajová, Embeddings of graphs in Euclidean spaces, Discrete and Computational Geometery 4 (1989) 349–364.
36. J. Reiterman, V. Rödl and E. Šiňajová, Geometrical embeddings of graphs. Discrete Mathematics 74 (1989) 291–319.
37. J. Reiterman, V. Rödl and E. Šiňajová, On embedding of graphs into Euclidean spaces of small dimension, J. Combin. Theory B 56 (1992) 1–8.
38. E.R. Scheinerman, K. Tucker, Modeling graphs using dot product representations, Computational Statistics 25 (2010) 1–16.
39. H.A. Simon, On a class of skew distribution functions, Biom. 42 (1955) 425–440.
40. T.A.B. Snijders, Statistical Models for Social Networks, Annual Rev. Sociology 37 (2011) 131–153.
41. J.P. Spinrad, Efficient Graph Representations, Fields Institute Monographs, American Mathematical Society, 2003.
42. D.J. Watts, P.S. Dodds, M.E.J. Newman, Identity and Search in Social Networks, Science 296 (2002) 1302–1305.
43. S.J. Young and E.R. Scheinerman, Random dot product graph models for social networks, Proc. WAW 2007, LNCS 4863 (2007) 138–149.

44. S.J. Young and E.R. Scheinerman, Directed random dot product graphs, Internet Mathematics 5(2008) 91–111.

A Counterexample for the Case $d = 4$

Gray and Wilson [16] showed there exist sets of four vectors in $\mathbb{R}^3$ and sets of $\lfloor d/2 \rfloor + 3$ vectors in $\mathbb{R}^d$ for any $d \geq 5$ such that all vectors in the set have pairwise nonnegative dot product and that no orthogonal transformation can map all vectors of the set to the nonnegative orthant. Moreover, they showed that for any three vectors in $\mathbb{R}^3$ and any four vectors in $\mathbb{R}^4$ that have pairwise nonnegative dot product, there exists an orthogonal transformation that maps all vectors to the nonnegative orthant. Note that this gives a tight upper and lower bound for $\mathbb{R}^3$. We now give a tight upper bound for $\mathbb{R}^4$.

Proposition 1. *There is a set of five vectors in $\mathbb{R}^4$ with pairwise nonnegative dot product such that no orthogonal transformation can map all vectors in the nonnegative orthant.*

Proof. The idea of the proof is similar to the construction of Gray and Wilson [16] for $d \geq 5$. Consider the following five vectors:

$$\begin{aligned} v_1 &= (\ \ 1, \ \ 1, 0, 0\ ) \\ v_2 &= (\ \ 0, \ \ 0, 1, 0\ ) \\ v_3 &= (\ \ 0, \ \ 0, 0, 1\ ) \\ w_1 &= (-1, \ \ 2, 1, 3\ ) \\ w_2 &= (\ \ 2, -1, 1, 1\ ) \end{aligned}$$

Note that the five vectors indeed have pairwise nonnegative dot product. Moreover, v_1, v_2, v_3 are pairwise orthogonal and w_1 and w_2 are orthogonal. It is crucial to observe, however, that both w_1 and w_2 have positive dot product with each of v_1, v_2, v_3.

Suppose there is an orthogonal transformation T that maps v_1, v_2, v_3, w_1, w_2 to the nonnegative orthant. Recall that orthogonal transformations preserve the dot product between any two vectors. Hence, in particular, $T(v_1), T(v_2), T(v_3)$ are pairwise orthogonal, and $T(w_1)$ and $T(w_2)$ are orthogonal. Also note that all coordinates of $T(v_1), T(v_2), T(v_3), T(w_1), T(w_2)$ are nonnegative. If $T(v_i)$ and $T(v_j)$ are strictly positive in the same coordinate for $i, j \in \{1, 2, 3\}, i \neq j$, then their dot product is strictly positive, contradicting their orthogonality. Therefore, at least two of $T(v_1), T(v_2), T(v_3)$ span a coordinate axis. Without loss of generality, these are the third and fourth coordinate axes. Since T preserves the value of the dot product between any two vectors, it follows from the crucial observation above that $T(w_1)$ and $T(w_2)$ are both strictly positive in the third and fourth coordinate. As all coordinates of both $T(w_1)$ and $T(w_2)$ are nonnegative, the dot product of $T(w_1)$ and $T(w_2)$ is strictly positive, contradicting their orthogonality. Hence, T cannot exist. $\square$