Spectral Relaxations and Fair Densest Subgraphs*

Aris Anagnostopoulos
Sapienza, University of Rome

Luca Becchetti
Sapienza, University of Rome

Adriano Fazzone
Sapienza, University of Rome

Cristina Menghini
Sapienza, University of Rome

Chris Schwiegelshohn
Sapienza, University of Rome

ABSTRACT

Reducing hidden bias in the data and ensuring fairness in algorithmic data analysis has recently received significant attention. In this paper, we address the problem of identifying a densest subgraph, while ensuring that none of one binary protected attribute is disparately impacted.

Unfortunately, the underlying algorithmic problem is NP-hard, even in its approximation version: approximating the densest fair subgraph with a polynomial time algorithm is at least as hard as the densest subgraph problem of at most k vertices, for which no constant approximation algorithms are known.

Despite such negative premises, we are able to provide approximation results in two important cases. In particular, we are able to prove that a suitable spectral embedding allows recovery of an almost optimal, fair, dense subgraph hidden in the input data, whenever one is present, a result that is further supported by experimental evidence. We also show a polynomial time, 2-approximation algorithm, whenever the underlying graph is itself fair. We finally prove that, under the small set expansion hypothesis, this result is tight for fair graphs.

The above theoretical findings drive the design of heuristics, which we experimentally evaluate on a scenario based on real data, in which our aim is to strike a good balance between diversity and highly correlated items from Amazon co-purchasing graphs.

CCS CONCEPTS

• Theory of computation → Graph algorithms analysis; **• Information systems →** *Web searching and information discovery.*

KEYWORDS

densest subgraph; fairness; spectral graph analysis

ACM Reference Format:
Aris Anagnostopoulos, Luca Becchetti, Adriano Fazzone, Cristina Menghini, and Chris Schwiegelshohn. 2020. Spectral Relaxations and Fair Densest Subgraphs. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3340531.3412036

*Supported by the ERC Advanced Grant 788893 AMDROMA "Algorithmic and Mechanism Design Research in Online Markets", the EC H2020RIA project "SoBigData++" (871042), and the MIUR PRIN project ALGADIMAR "Algorithms, Games, and Digital Markets".

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3412036

1 INTRODUCTION

The identification of dense subgraphs is a fundamental primitive in community detection and graph mining [19, 25, 35, 40, 46]. Given an underlying graph $G = (V, E)$, the density of a node set $S \subseteq V$ is defined as $\frac{2\cdot|E\cap S\times S|}{|S|}$. In most mining scenarios, communities are assumed to have a high intra-community density versus a lower inter-community density. In this sense, density is arguably the most natural measure of quality for evaluating and comparing communities in graphs (see [12] for an extensive survey.)

In this paper, we consider the densest subgraph problem with fairness constraints. Specifically, we are given a binary labeling of the nodes of the graph $\ell : V \to \{-1, 1\}$. The labeling corresponds to an attribute that ideally should be uncorrelated with community membership. Our goal is to compute a set of nodes $S \subseteq V$ of maximum density while ensuring that S contains an equal number of representatives of either label. The problem has a number of motivating applications, some of which are discussed below.

Mitigation of Polarization. Social networks are very prone to polarization among users [9]: reinforcement of user preferences can lead to feedback loops. For example, recommender systems incentivize disagreement minimization, leading to echo chambers among users with similar preferences. This problem has been considered for example by Musco et al. [34], who studied the problem of identifying a graph of connections between users (of two different opinions), such that polarization and disagreement are simultaneously minimized. The notions behind the fair densest subgraph problem are closely related: Its goal is to maximize agreement while avoiding polarization.

Team Formation. In crowdsourcing, team formation consists in identifying a set of workers, whose collective skill set includes all skills that are required for processing some given jobs. Lappas et al. [30] proposed subgraph density as a way of modeling the effectiveness of multiple individuals when working together. The potential benefits of team diversity are well documented in organizational psychology [24] studies and also highlighted by recent work (e.g., see [32] and follow-up work). Diversity in turn can be naturally modeled via fairness constraints.

Diversity in Association Rule Mining. Sozio and Gionis [43] study dense subgraphs for association rule mining: Given a set of tags used to label objects, the densest subgraph problem allows to determine additional related tags that can be used for a better description of the objects. It is common that the tags that are added are semantically identical to those already used. We argue that an appropriate labeling of the tags followed by solving the fair densest subgraph problem allows recovery of a set of tags that are not only closely related, but also unique.

Algorithmic Fairness. As pioneered by Chierichetti et al. [14], there has recently been considerable work on clustering data sets using the disparity-of-impact measure. Conceptually, the aim is to perform data analysis such that the resulting clustering or classifier does not discriminate based on some protected attribute. In our case, finding a densest subgraph such that a protected attribute is not disparately impacted is equivalent to the definition of the fair densest subgraph problem.

1.1 Contributions

As it turns out (see Section 3), the fair densest subgraph problem is intractable in general, whereas its unconstrained counterpart can be solved optimally through network flow [22]. Nevertheless, we have some quantifiable results regarding approximation algorithms in special cases. We can show that, if the underlying graph itself is fair, there exists a 2-approximation algorithm. We further show that, assuming the widely used small set expansion hypothesis [38], this is the best possible. We also consider the case where the graph itself is not fair and we instead aim for a proportional representation. For this, in our opinion more flexible variant of the problem, we show that the results for fair graphs can be extended.

Although this worst-case behavior is discouraging, the possibility of effective algorithms on practical instances is not ruled out. To this end, we identify properties that, if satisfied by some subgraph of the network under consideration, will afford recovery of an approximately fair, dense subgraph. More precisely, our goal in this respect is to design a heuristic that

(a) has a quantifiable guarantee if the underlying graph is well-behaved and
(b) is practically viable.

Our main result is a spectral algorithm that satisfies both of these requirements. In particular, the practical viability of our algorithm underscores that our notion of a well-behaved graph is a realistic one. As a candidate application, we consider the scenario of providing diverse recommendations of high quality, using data from the Amazon product co-purchasing graph. Our experiments not only confirm the quality of the output solutions, but also the scalability of our approach, which may not be the case for a conventional combinatorial approximation algorithm.

Overview of our approach. Our approach builds on the finding [27, 33] that the densest subgraph problem admits a spectral formulation. Specifically, an approximate densest subgraph can be computed by selecting nodes for inclusion according to the magnitudes of the corresponding entries in the main eigenvector of G's adjacency matrix. Unfortunately, this approach does not afford balanced solutions in general. In a nutshell, we sidestep this issue by first projecting the adjacency matrix onto a suitable "fair" subspace, an operation that corresponds to the enforcement of "soft" fairness constraints. To see why the conventional spectral approach of [27] may not work[1] and why our approach mitigates the issue, Figure 1 presents plots obtained from Amazon books on US politics [29]. The books are labeled as either conservative or liberal, which corresponds to the labels -1 or 1. As described above, a candidate application may be to find a selection of books that are of interest to multiple readers, while mitigating potential polarization along political lines.

On the left, we observe the books ordered according to their corresponding entries in the main eigenvector of the adjacency matrix of the co-purchase graph. Books are also colored according to political orientation. We can observe that, whereas liberal books are well distributed, conservative ones are clustered. On the right we observe the results after application of our spectral embedding, which affords recovery of a subgraph of the co-purchase graph that is both dense and approximately balanced. Note that now conservative books are also well distributed along the principal component.

1.2 Related Work

Densest Subgraph. Identifying dense subgraphs is a key primitive in a number of applications; see [18, 20, 21, 46]. The problem can be solved optimally in polynomial time [22]. On the contrary, the fair densest subgraph problem is highly related to the densest subgraph problem limited to at most k nodes, which cannot be approximated up to a factor of $n^{1/(\log\log n)^c}$ for some $c > 0$ assuming the exponential time hypothesis [31] and for which state-of-the-art methods yield an $O(n^{1/4+\varepsilon})$ approximation [6].

Algorithmic Fairness. Fairness in algorithms received considerable attention in the recent past, see [23, 45, 47, 49] and references therein. The closely related notion of disparate impact was first proposed by Feldman et al. [17]. It has since been used by Zafar et al. [48] and Noriega-Campero et al. [37] for classification and Celis et al. [10, 11] for voting and ranking problems. Another problem that received considerable attention is fair clustering. This was first proposed as a problem by Chierichetti et al. [14] in the case of a binary protected attribute. It was then investigated for various objectives and more color classes in theirs and subsequent work [1, 4, 5, 26, 39, 42].

Most closely related to our work are some recent works [28, 41, 44]. From those, the works of Samadi et al.[41] and Tantipongpipat et al. [44]. consider the problem of executing a principal component analysis in a fair manner. Specifically, given a matrix A where the rows are colored (e.g., every row corresponds to a man or a woman), they ask for an algorithm that finds a rank-k matrix A' whose residual error $\|A - A'\|$ is small for both types of rows simultaneously. Whereas our method is similarly based on using the principal component in a fair manner, the difference is that we may be forced to treat the classes differently, if we aim to uncover a dense subgraph as illustrated in the example mentioned previously and illustrated in Figure 1.

The paper by Kleindessner et al. [28] considers spectral clustering problems such as normalized cut. Like our work, they project the Laplacian matrix of a graph G onto a suitable "fair" subspace, and then run k-means on the subspace spanned by the smallest resulting

[1] In fact, this applies to any approach based on unconstrained maximization of the induced subgraph's density.

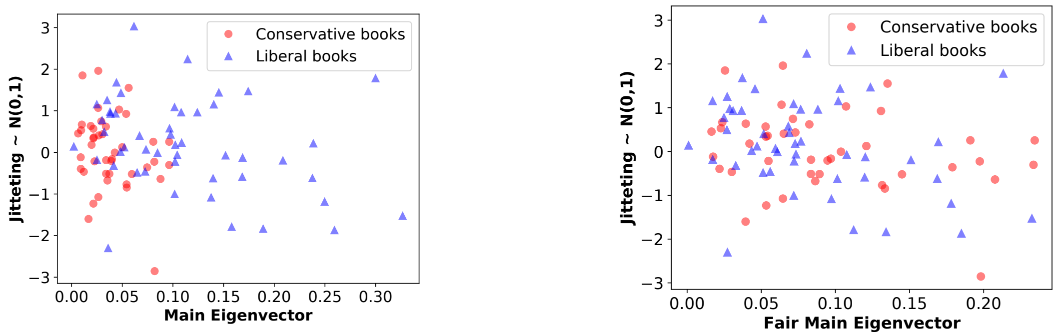


Figure 1: Projection of books (see Section 4) onto the first principal component. (Left) Original data. (Right) Data after spectral embedding. Books are ordered on the x axis according to their corresponding entries in the main eigenvector, whereas on the y axis we have random noise for visualization.

eigenvectors. Under a fair version of the stochastic block model, they show that this algorithm recovers planted fair partitions. Our work continues this idea by applying the technique to the densest subgraph problem.

1.3 Preliminaries and Notation

We consider an undirected graph $G(V, E, w)$, where V is the set of n nodes, $E \subset V \times V$ is the set of edges, and $w : E \longrightarrow \mathbb{R}_{\geq 0}$ is a weight function. We denote the (weighted) adjacency matrix of G by A. For a subset $E' \subset E$ of the edges, we let $w(E') = \sum_{e \in E'} w(e)$. For unweighted graphs we have $w(e) = 1$ for each $e \in E$. For a node $u \in V$, its (weighted) degree (often called volume) is $d_u = \sum_{e \cap \{v\} \neq \emptyset} w(e)$. We also let $d_{\max} = \max_u d_u$. For a $S \subseteq V$, we denote by G_S the induced subgraph. The *density* $D_S(G)$ of $S \subseteq V$ is simply the average degree of G_S, namely,

$$D_S(G) = \frac{2 \cdot w(E \cap S \times S)}{|S|}.$$

We omit G from $D_S(G)$, whenever it is clear from context.

A *coloring* of the vertices is simply a map $c : V \longrightarrow [\ell]$ of V, where $[\ell] := \{1, 2, \ldots, \ell\}$. A set $S \subset V$ is called *fair* if $|S \cap \{v \in V \mid c(v) = 1\}| = |S \cap \{v \in V \mid c(v) = 2\}| = \cdots = |S \cap \{v \in V \mid c(v) = \ell\}|$. A graph is called fair if V is fair. In the remainder, we provide positive results for the important case $\ell = 2$. In this case, for simplicity of exposition, we denote the colors *r*ed and *b*lue and we use $Red := \{v \in V \mid c(v) = red\}$ and $Blue := \{v \in V \mid c(v) = blue\}$ to refer to nodes of the respective color.

Definition 1.1 (Fair Densest Subgraph Problem). Given a (weighted) graph $G(V, E, w)$ and a coloring c of its vertices, identify a fair subset $S \subseteq V$ that maximizes D_S.

The fair densest subgraph problem is obviously a constrained version of the densest subgraph problem. It turns out to be considerably harder than its (polynomially solvable) unconstrained counterpart, as we show in Section 3.

Linear algebra notation. We denote by $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ the eigenvalues of A and by v_i its ith eigenvector. We also set $\lambda = \max\{\lambda_2, |\lambda_n|\}$. Note that we always have $\lambda_1 \leq d_{\max}$. For a subset $S \subset V$, we denote by χ its normalized indicator vector, where S is understood from context. Namely, $\chi_i = 1/\sqrt{|S|}$ if $i \in S$, $\chi_i = 0$ otherwise. Finally, for a vector $x \in \mathbb{R}^n$, we let $\|x\| = \sqrt{\sum_{i=1}^n x_i^2}$, the 2-norm of x.

2 SPECTRAL RELAXATIONS FOR THE FAIR DENSEST SUBGRAPH

As observed in Kannan and Vinay [27], the densest subgraph problem admits a spectral formulation. In particular, if we let x be an indicator vector over the vertex set, then the indicator vector of the vertex subset maximizing density is the maximizer of the following expression:

$$\max_{x \in \{0,1\}^n} \frac{x^T A x}{x^T x}.$$

Now, assume that each node is colored with one of two colors, red or blue. In the optimal solution x^* one of the colors might be overrepresented. To formulate the problem of computing a fair solution, we can add the constraint

$$\sum_{\text{node } i \text{ is red}} x_i = \sum_{\text{node } i \text{ is blue}} x_i$$
$$\Leftrightarrow \sum_{\text{node } i \text{ is red}} x_i - \sum_{\text{node } i \text{ is blue}} x_i = 0.$$

If we define the (unit 2-norm) vector

$$f_i = \begin{cases} \frac{1}{\sqrt{n}} & \text{if node } i \text{ is red} \\ -\frac{1}{\sqrt{n}} & \text{if node } i \text{ is blue,} \end{cases}$$

the above constraint can be described as $f^T x = 0$. We call such an x *fair.* Conversely, very unbiased solutions will have high, in absolute value, inner products with f.

Fair Densest Subgraph: Spectral Relaxation. Based on the considerations above, our approach transforms the input data (in this case the adjacency matrix A) by first projecting them onto the kernel of f. Namely, we first consider the following formulation of the fair densest subgraph problem:

$$\max_{x \in \{0,1\}^n} \frac{2x^T (I - ff^T) A (I - ff^T) x}{x^T x}.$$

It should be noted that, for any fair subset S with indicator x, we have $\frac{2x^T Ax}{x^T x} = \frac{2x^T(I-ff^T)A(I-ff^T)x}{x^T x}$. Conversely, for any indicator vector $x \notin span(I - ff^T)$, the objective value can only decrease after we project to the kernel of f.

We next note that, by the discussion in the beginning of the section, by relaxing x to be an arbitrary vector, the above expression is maximized by the main eigenvector of $(I - ff^T)A(I - ff^T)$. The above relaxation corresponds to replacing hard fairness constraints with soft ones.

It is straightforward to encode more complicated fairness constraints using this technique. Suppose, for example, that we are given ℓ colors, and wish to output a subgraph such that every color is featured equally often. This induces a set of constraints

$$\begin{aligned} \sum_{\text{node } i \text{ is red}} x_i &= \sum_{\text{node } i \text{ is blue}} x_i \\ \sum_{\text{node } i \text{ is red}} x_i &= \sum_{\text{node } i \text{ is green}} x_i \\ &\vdots \\ \sum_{\text{node } i \text{ is red}} x_i &= \sum_{\text{node } i \text{ is yellow}} x_i \end{aligned}$$

The vectors satisfying all of these constraints lie in the nullspace of some $\ell - 1$ dimensional subspace S. Assume that F is a matrix such that the columns of F form an orthogonal basis of S. Then the above technique leads to the problem

$$\max_{x \in \{0,1\}^n} \frac{2x^T(I - FF^T)A(I - FF^T)x}{x^T x}.$$

More generally, this technique can be extended to any system of linear constraints. One only has to merely find a suitable basis and project A onto said basis.

We note that even though the technique can handle these more complicated constraints, leveraging this in an algorithm with provable guarantees seems very difficult. Nevertheless, our experiments dealing with multiple colors showcase that we can still tackle more complicated fairness constraints with success in practice, see Section 4.1.

2.1 Recovery of Dense Fair Subgraphs in Almost Regular Graphs

To prove our main result we need the following definition:

Definition 2.1. A graph $H = (V_H, E_H)$ is (d, ϵ)-regular if a d exists, such that $(1 - \epsilon)d \le d_i \le (1 + \epsilon)d$, for every $i \in V_H$.

THEOREM 2.2. *Assume we have a graph $G = (V, E, w)$ with a 2-coloring of the nodes. Assume the spectrum of A satisfies $\lambda_1 \ge 4\lambda$. Assume further that G contains a fair subset S such that: (1) G_S is (d, ϵ)-regular and (2) $d \ge (1 - \theta)d_{\max}$. In this case, it is possible to recover all but $16(\epsilon + \theta)|S|$ of the vertices in S in polynomial time.*

Intuitively, the result above states that, if the underlying network G is an expander containing an almost-regular, dense and fair subgraph, we can approximately retrieve it in polynomial time. Succinctly, this follows because, under these assumptions, the indicator vector of S forms a small angle with the main eigenvector of $(I - ff^T)A(I - ff^T)$.

PROOF OF THEOREM 2.2. For this proof, we denote by $\hat{\lambda}_1 \ge, \dots, \ge \hat{\lambda}_n$ the eigenvalues of $(I - ff^T)A(I - ff^T)$ and by $\hat{v}_i$ the ith associated eigenvector. For a vertex i of G_S we denote by $\hat{d}_i$ its degree in G_S. We denote by χ the indicator vector of S and we let $m = |S|$.

As a first step, we summarize straightforward, yet useful properties of the spectrum of $(I - ff^T)A(I - ff^T)$.

CLAIM 1. *Whenever $\hat{\lambda}_i \neq 0$ we have:*

$$(I - ff^T)\hat{v}_i = \hat{v}_i \text{ and } \hat{\lambda}_i = \hat{v}_i^T A\hat{v}_i \tag{1}$$

PROOF. If $\hat{\lambda}_i \neq 0$, we have:

$$(I - ff^T)A(I - ff^T)\hat{v}_i = \hat{\lambda}_i\hat{v}_i.$$

Because $(I - ff^T)$ is a projection matrix, if we pre-multiply both members of the above equation by $(I - ff^T)$ we have:

$$(I - ff^T)A(I - ff^T)\hat{v}_i = \hat{\lambda}_i(I - ff^T)\hat{v}_i.$$

Subtracting the first equation from the second and recalling that $\hat{\lambda}_i \neq 0$, we immediately obtain the first part of the claim. The second part follows immediately from the first:

$$\hat{\lambda}_i = \hat{v}_i^T(I - ff^T)A(I - ff^T)\hat{v}_i = \hat{v}_i^T A\hat{v}_i.$$

□

It should be noted that, as a consequence of Claim 1, we always have:

$$\hat{\lambda}_1 = \hat{v}_1^T(I - ff^T)A(I - ff^T)\hat{v}_1 = \hat{v}_1^T A\hat{v}_1 \le v_1^T Av_1 = \lambda_1.$$

Note that this last property does not apply to the other eigenvalues in general. The first important, technical step to prove Theorem 2.2 is showing that the hypothesis $\lambda_1 \ge 4\lambda$ implies that $\hat{\lambda}_2$ cannot be too large.

LEMMA 2.3. *Assume the spectrum of A satisfies the condition $\lambda_1 \ge 4\lambda_2$. Then $\hat{\lambda}_2 \le \frac{3}{4}\lambda_1$.*

PROOF. We express $\hat{v}_1$ and $\hat{v}_2$ as $\hat{v}_1 = \beta v_1 + q$ and $\hat{v}_2 = \gamma v_1 + z$, where q and z respectively denote $\hat{v}_1$'s and $\hat{v}_2$'s components orthogonal to v_1, the main eigenvector of A. Note that, because v_1, $\hat{v}_1$ and $\hat{v}_2$ have unit norms, we have $\beta^2 + \|q\|^2 = 1$ and $\gamma^2 + \|z\|^2 = 1$. Next:

$$\hat{\lambda}_2 = (\gamma v_1 + z)^T A(\gamma v_1 + z)^T = \gamma^2\lambda_1 + z^T Az, \tag{2}$$

where the first equality follows from Claim 1, and the second follows because $z \in \mathbf{span}(v_2, \dots, v_n)$ by definition and the v_i's form an orthonormal basis.

We next show that $\gamma^2 \le 1/2$. Assume on the contrary that $\gamma^2 > 1/2$. We show that this implies $\beta^2 \le 1/2$ and that the latter in turns brings to a contradiction. Because the $\hat{v}_i$'s are orthonormal, we have $\hat{v}_1^T\hat{v}_2 = 0$, which implies $|\beta\gamma| = |q^T z|$. As a consequence, if $\beta^2 > 1/2$, we would have $|\beta\gamma| \ge 1/2$, whereas $\beta^2 + \|q\|^2 = 1$ and $\gamma^2 + \|z\|^2 = 1$ would imply $\|q\|^2 < 1/2$ and $\|z\|^2 < 1/2$, whence $|q^T z| \le \|q\|\|z\| < 1/2$ from the Cauchy–Schwarz inequality, a contradiction. On the other hand, if $\beta^2 \le 1/2$:

$$\begin{aligned} \lambda_1 = \hat{\lambda}_1 &= (\beta v_1 + q)^T A(\beta v_1 + q)^T = \beta^2\lambda_1 + q^T Aq \\ &= \beta^2\lambda_1 + \|q\|^2\frac{q^T Aq}{\|q\|^2} \le \frac{1}{2}\lambda_1 + (1 - \beta^2)\lambda_2 \le \frac{1}{2}\lambda_1 + \lambda_2. \end{aligned}$$

The last expression is strictly less than λ_1, whenever $\lambda_2 < \lambda_1/2$. Since we are assuming $\lambda_1 \ge 4\lambda_2$, this is clearly a contradiction.

Assume therefore that $\gamma^2 \leq 1/2$. In this case, Equation (2) implies:

$$\hat{\lambda}_2 = \gamma^2\lambda_1 + z^T Az \leq \gamma^2\lambda_1 + \|z\|^2 \max_{w\perp v_1} \frac{w^T Aw}{\|w\|^2}$$
$$= \gamma^2\lambda_1 + (1-\gamma^2)\lambda_2 \leq \frac{\lambda_1}{2} + \lambda_2 \leq \frac{3}{4}\lambda_1.$$

□

The second step is showing that Lemma 2.3 implies that the indicator vector of the fair densest subgraph is close to $\hat{v}_1$:

Lemma 2.4. *Assume the hypotheses of Theorem 2.2 hold. Then:*

$$\|\chi - \hat{v}_1\|^2 \leq 4(\epsilon + \theta).$$

Proof. We begin by noting that $\chi^T f = 0$ by definition, which implies $(I - ff^T)\chi = \chi$. We therefore have:

$$\chi^T(I - ff^T)A(I - ff^T)\chi = \frac{\sum_{i\in S}\hat{d}_i}{m} \geq (1-\epsilon)d, \tag{3}$$

Next, we decompose χ along its components respectively parallel and orthogonal to $\hat{v}_1$, namely, $\chi = \alpha\hat{v}_1 + z$, and we note that $\|z\|^2 = 1 - \alpha^2$, as both $\hat{v}_1$ and χ are unit norm vectors. Set $B = (I - ff^T)A(I - ff^T)$ for the sake of space. We have:

$$\chi^T B\chi = (\alpha\hat{v}_1 + z)^T B(\alpha\hat{v}_1 + z) = \alpha^2\hat{\lambda}_1 + z^T Bz$$
$$\leq \alpha^2\hat{\lambda}_1 + \hat{\lambda}_2\|z\|^2 \leq \alpha^2\hat{\lambda}_1 + (1-\alpha^2)\hat{\lambda}_2. \tag{4}$$

Putting together (3) and (4) yields $\alpha^2 \geq \frac{(1-\epsilon)d - \hat{\lambda}_2}{\hat{\lambda}_1 - \hat{\lambda}_2}$. Now:

$$\|\chi - v\|^2 \leq 1 - \frac{(1-\epsilon)d - \hat{\lambda}_2}{\hat{\lambda}_1 - \hat{\lambda}_2}$$
$$\leq 1 - \frac{(1-\epsilon)(1-\theta)d_{\max} - \hat{\lambda}_2}{\lambda_1 - \hat{\lambda}_2}$$
$$\leq 1 - \frac{(1-\epsilon)(1-\theta)\lambda_1 - \hat{\lambda}_2}{\lambda_1 - \hat{\lambda}_2}$$
$$< 1 - \frac{\lambda_1 - \hat{\lambda}_2 - (\epsilon+\theta)\lambda_1}{\lambda_1 - \hat{\lambda}_2}$$
$$= \frac{(\epsilon+\theta)\lambda_1}{\lambda_1 - \hat{\lambda}_2} \leq 4(\epsilon+\theta).$$

Here, the second inequality follows from our hypotheses on d and because $\hat{\lambda}_1 \leq \hat{\lambda}$, the third inequality follows because the main eigenvalue of an adjacency matrix is upper-bounded by the maximum degree of the underlying graph, and the last inequality follows from Lemma 2.3. □

Corollary 2.5. *Under the hypotheses of Lemma 2.4, for all but at most $16m(\epsilon + \theta)$ vertices in V we have: (1) $\hat{v}_1(i) \geq \frac{1}{2\sqrt{m}}$ if $i \in S$, (2) $\hat{v}_1(i) < \frac{1}{2\sqrt{m}}$ otherwise.*

The algorithm. Our algorithm is based on a sweep of $\hat{v}_1$ [27, 33]. In particular, we run Algorithm GSA (see Algorithm 1) with $M = (I - ff^T)A(I - ff^T)$ and $\Delta = 16(\epsilon + \theta)$.

Corollary 2.5 ensures that i) the above algorithm always returns a solution, ii) the solution returned by the algorithm will not be worse than the one obtained by picking i if $\hat{v}_1(i) \geq \frac{1}{2\sqrt{m}}$ and rejecting it otherwise. This concludes the proof of Theorem 2.2. □

Algorithm: General Sweep Algorithm (GSA)
Data: Non-negative $n \times n$ matrix M, parameter Δ
Result: Subset $S \subseteq V$
$\hat{S} = \emptyset$; $\hat{D} = 0$;
Compute v_1 = main eigenvector of M;
Sort nodes $i \in V$ in nonincreasing order of $v_1(i)$;
// Assume w.l.o.g. that $\{1, \ldots, n\}$ is the resulting ordering of the nodes in V;
for $s = 1$ *to* n **do**
    $S = \{1, \ldots, s\}$
    Compute D_S = density of the subgraph induced by S
    if $D_S > \hat{D}$ *AND* $||S \cap Red| - |S \cap Blue|| \leq \Delta|S|$ **then**
        $\hat{S} = S$; $\hat{D} = D_S$
    end
end
return $\hat{S}$

Algorithm 1: General Sweep Algorithm (nonincreasing).

The running time of the algorithm is dominated by computing the first eigenvector and the projecting of the rows of the Laplacian onto said eigenvector. This can be done, up to $(1 + \varepsilon)$ precision, in linear time.

3 HARD CONSTRAINTS AND HARDNESS OF APPROXIMATION

In general, enforcing fairness can make an easy problem intractable and this is the case for the densest subgraph problem. In this context, spectral relaxations can be regarded as a way to mitigate this issue, by enforcing soft fairness constraints to virtually any problem that is amenable to an algebraic formulation.

Nevertheless, in some cases it might be important to assess the *price of fairness*, by comparing the achievable quality of fair solutions to that of solutions for the original, unconstrained problem. In this section, we complement our algorithmic treatment of fairness with hardness results and approximation algorithms for specific cases. Some proofs are omitted for the sake of space, but they are available in the full version of the paper.[2] Some of our hardness results are based on the *small set expansion hypothesis*, which we now describe.

Consider a d-regular weighted graph G and, for every $S \subset V$, denote by $\Phi(S)$ the *expansion* (or conductance) of S [38]. Given two constants $\delta, \eta \in (0, 1)$, the small set expansion problem [38] $SSE(\delta, \eta)$ asks to distinguish between the following two cases:

- **Completeness** There exists a set of nodes $S \subset V$ of size $\delta \cdot |V|$ such that $\Phi(S) \leq \eta$.
- **Soundness** For every set of nodes $S \subset V$ of size $\delta \cdot |V|$, $\Phi(S) \geq 1 - \eta$.

Our hardness proofs are based on the small set expansion hypothesis defined as follows.

Conjecture 3.1 (SSEH). *For every $\eta > 0$ there exists a $\delta := \delta(\eta) > 0$ such that $SSE(\eta, \delta)$ is NP-hard.*

[2] https://arxiv.org/abs/1905.13651

Recall from Section 1.2 that, whereas the densest subgraph problem is polynomially solvable, the best approximation for the densest at-most-k subgraph problem is in $O(n^{1/4})$ [7] and cannot be approximated up to a factor of $n^{1/(\log \log n)^c}$ for some $c > 0$ assuming the exponential time hypothesis [31] . The next theorem implies that these inapproximability results for the densest at-most-k subgraph problem hold also for the fair densest subgraph problem, showing that fairness constraints can drastically affect hardness of this problem.

Theorem 3.2. *The densest fair subgraph problem is at least as hard as the densest at most k subgraph problem. Moreover, any α-approximation to the densest at-most-k subgraph is a 2α approximation to densest fair subgraph.*

Proof. Consider an arbitrary graph $G(V, E)$. We consider V to be colored red. Add k blue nodes with no edges. Then the density of the fair densest subgraph is, up to a multiplicative factor of exactly $\frac{1}{2}$, equal to the density of the densest-at-most-$2k$ subgraph. Conversely, running an algorithm for densest k-subgraph with $k = \min(|Blue|, |Red|)$, and balancing out the resulting subgraph in post processing decreases the density by at most a factor 2. (This latter part is explained in more detail in the following theorem). □

When the input graph G is itself fair, we can provide stronger bounds.

1 **Input:** Graph $G(V, E, w)$
2 Compute the densest subgraph S
3 W.l.o.g $|S \cap Blue| \geq |S \cap Red|$
4 While $|S \cap Blue| > |S \cap Red|$, add an arbitrary node $v \in Red \setminus S$ to S
5 Return S

Algorithm 2: Approximate Fair Densest Subgraph

Theorem 3.3. *Given a fair graph $G(V, E, w)$, Algorithm 2 computes a fair set $S \subset V$, such that $2D_S \geq OPT$, where OPT is the density of the fair densest subgraph.*

Proof. We refer to the set S computed after line 2, and 4 as S_1 an S_2, respectively. Because S_1 is the unconstrained densest subgraph, $D_{S_1} > OPT$. For S_2, we observe that $|S_2| = S_1 + |S_1 \cap \text{Blue}| - |S_1 \cap \text{Red}| \leq 2 \cdot |S_1|$, hence $D_{S_2} = \frac{w(E_{S_2})}{|S_2|} \geq \frac{w(E_{S_1})}{2|S_1|} \geq \frac{OPT}{2}$. □

The running times of both algorithms depend on the running time of the subroutines used to compute dense subgraphs. Unconstrained dense subgraphs can be found by solving a linear program or by computing a max flow [13, 22]. A faster $(1 + \varepsilon)$ approximation that runs in time $O(n\text{polylog}(n))$ also exits [2, 15].

For the densest k-subgraph problem, the currently best algorithm that computes an $O(n^{1/4+\varepsilon})$ approximation runs in time $n^{O(1/\varepsilon)}$ [6].

We conclude this section by showing that approximating the fair densest subgraph problem beyond a factor of 2 is at least as hard as solving $SSE(\eta, \delta)$. Therefore, barring a major algorithmic breakthrough, Algorithm 2 is optimal. The proof is provided in the full version of the paper and it is based on the following idea: In regular graphs, for a given set of nodes S, the expansion $\Phi(S)$ is related to the density of S. We can use this, so that, given a graph G, we can carefully construct a colored graph G' such that finding the optimal fair densest subgraph in G' gives an estimate of the largest-expansion node set in G.

Theorem 3.4. *If SSEH holds, computing a $(2 - \varepsilon)$ approximation of the fair densest subgraph problem in fair graphs is NP-hard for any $\varepsilon > 0$.*

4 EXPERIMENTAL ANALYSIS

Worst case bounds are often uninformative when compared with empirical behaviour. Algorithm 2 is (assuming that the underlying graph is fair) theoretically optimal and therefore theoretically superior to the spectral recovery schemes. As we now describe, the empirical performance between these approaches paints the opposite picture.

Overview. To test the performances of our algorithms on real data we used two publicly available datasets: PolBooks [29] and Amazon products metadata [36]. Both (explicitly or implicitly) contain undirected unweighted graphs, whose nodes are products from the Amazon catalog, and an edge between two nodes exists if the corresponding products are frequently co-purchased by the same buyers. Moreover, for both datasets, each product belongs to exactly one category.

We tested our methods in a scenario in which, given a (not necessarily fair) labeled graph, our only interest lies in finding fair subgraphs with high density. In this context, we are considering the density of the provided solution as a quality indicator: the higher the density, the better the quality of a solution.

For our experiments we used an Intel Xeon 2.4GHz with 24GB of RAM running Linux Ubuntu 18.04 LTS. All methods have been implemented in Python3 using the functionalities provided by NetworkX [3] and SciPy [4] libraries.

Datasets. The PolBooks dataset [29] is an undirected unweighted graph[5], whose nodes represent books on US politics included in the Amazon catalog, and an edge between two books exists if both books are frequently co-purchased by the same buyers. Each book is further labeled depending on its political stance, possible labels being *liberal*, *neutral*, and *conservative*. For our experiments, we considered only the subgraph induced by *liberal* and *conservative* books, obtaining 92 nodes (43 of which were associated with a *conservative* worldview, 49 with a *liberal* worldview) for 374 edges in total.

The Amazon products metadata dataset [36] contains descriptions for 15.5 million Amazon products [6]. For a single product, we only considered the product id (*asin* field), the category the product belongs to (*main_cat* field), and the set of frequently co-purchased products (*also_buy* field). It should be noted that in this dataset, each node belongs to exactly one (main) Amazon category so that, together, these three fields allow recovery of a large, undirected, labeled graph, with products as nodes, categories as labels, and

[3] https://networkx.github.io/documentation/stable
[4] https://www.scipy.org
[5] http://www.casos.cs.cmu.edu/computational_tools/datasets/external/polbooks/polbooks.gml.
[6] https://nijianmo.github.io/amazon/index.html

edges representing frequent co-purchasing product pairs. For this dataset, we leveraged the co-purchasing relation among products to naturally extract undirected and unweighted labeled graphs. In more detail, for each pair (ℓ_1, ℓ_2) of Amazon main categories, we extracted the undirected subgraph induced by the subset of nodes of category ℓ_1 (ℓ_2) that have at least one neighbor from category ℓ_2 (ℓ_1). We did not consider graphs with fewer than 100 nodes. In this way, we retrieved 299 subgraphs of two categories (colors), with sizes ranging between 103 and 33,922 nodes. We extended and applied this procedure to triples (ℓ_1, ℓ_2, ℓ_3) and quadruples $(\ell_1, \ell_2, \ell_3, \ell_4)$ of labels, obtaining 1,147 subgraphs of three categories (colors), with sizes ranging between 352 and 30,135 nodes, and 1,408 subgraphs of four categories (colors), with sizes ranging between 1,521 and 30,086 nodes.

Algorithms. We compare the performance of the following algorithms, which for simplicity we describe in the two-colors scenario:

2-DFSG. The optimal 2-approximation algorithm (Algorithm 2) based on Goldberg's optimal algorithm for the densest subgraph problem [22], described in Section 3.

Spectral Algorithms. Following [27, 33] and Theorem 2.2, we ran a variety of eigenvector rounding algorithms. These are all variants of a modified version of the General Sweep Algorithm (Algorithm 1) used in the proof of Theorem 2.2 that sorts the entries of the main eigenvector of M four times (instead of a single one) according to the following criteria: (1) nonincreasing; (2) nondecreasing; (3) nonincreasing absolute values; (4) nondecreasing absolute values. With these premises, we consider the following spectral algorithms. The first two are just the modified version of Algorithm 1 with different choices for M, whereas **PS** and **FPS** perform a slightly modified sweep that always affords a fair solution.

Single Sweep (SS). This algorithm is simply (Algorithm 1), when all previously mentioned sorting criteria are used, with $M = A$ and $\Delta = 0$.

Fair Single Sweep (FSS). It is the execution of **SS**, this time on matrix $(I - ff^T)A(I - ff^T)$ instead of A.

Paired Sweep (PS). Paired Sweep is a modification of **SS** in which the fairness constraint is satisfied by construction in each subgraph produced by the rounding algorithm. This is done by considering the subsets *Red* and *Blue* of the nodes, sorting each of them separately according to the values of the corresponding entries in the main eigenvector of A and then, for each $s = 1, \dots, \min\{|Red|, |Blue|\}$ considering the candidate set of nodes of cardinality $2s$ obtained by taking the first s nodes from each ordered subset. For a pseudocode, we refer to Algorithm 3.

Fair Paired Sweep (FPS). It is the execution of **PS**, this time on matrix $(I - ff^T)A(I - ff^T)$ instead of A.

4.1 Results

Figure 2 shows the performance of our algorithms on PolBooks dataset through the Pareto front of the subgraphs generated by each algorithm during its execution w.r.t. density and balance [7]. **PS** and **FPS** by construction only return fair solutions whereas the other algorithms potentially have trade-offs. In particular, the **2-DFSG** (Algorithm 2) starts at the unconstrained optimum and proceeds to add nodes that increase balance while potentially decreasing density.

Figure 3 shows the distributions of the normalized density, over the entire set of Amazon instances (for two, three, and four colors), of the fair subgraphs retrieved by different algorithms. Normalization, performed to make solutions for different instances comparable, is done by scaling to the optimal density of the unconstrained problem, making the maximum possible value on the y-axis equal to 1. Experimental results represented in Figure 3 (a, b, and c) show that spectral heuristics based on the paired-sweep technique (**PS**

[7] Given two color classes *Red* and *Blue*, we define the *balance* of a subgraph containing x *Red* and y *Blue* nodes as $\min\left(\frac{x}{y}, \frac{y}{x}\right)$.

```
Data: Graph G(V, E), with V = Red ∪ Blue, n × n adjacency
      matrix M, parameter Δ
Result: Subset S ⊆ V
Ŝ = ∅; D̂ = 0;
Compute v_1 = main eigenvector of M;
Sort nodes i ∈ Red and nodes j ∈ Blue in non increasing
  order wrt v_1
// Assume w.l.o.g. that Π_red = {1, ..., |Red|} and
Π_blue = {1, ..., |Blue|} is resulting ordering of
nodes in V;
Fuse node i from Π_red with node i from Π_blue
for s = 1 to min(|Red|, |Blue|) do
    S = {1, ..., s}
    Compute D_S = density of the subgraph induced by S
    if D_S > D̂ AND ||S ∩ Red| − |S ∩ Blue|| ≤ Δ|S| then
        Ŝ = S; D̂ = D_S
    end
end
return Ŝ
```

Algorithm 3: Paired Sweep Algorithm.

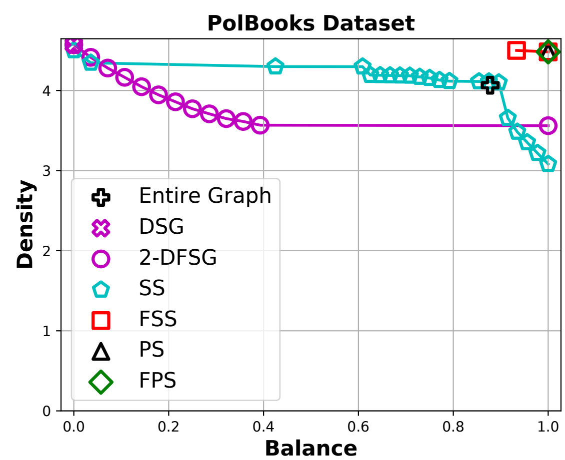
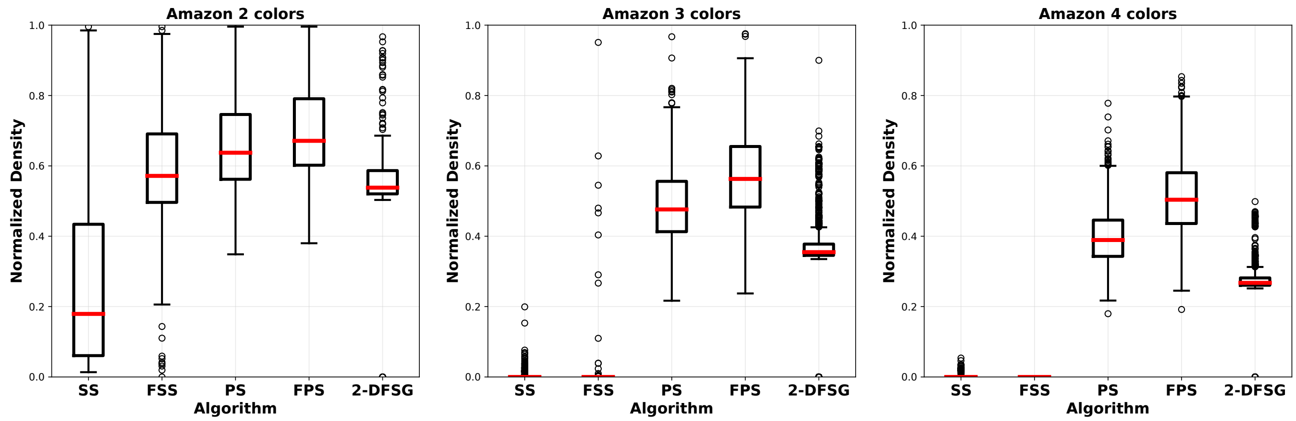


Figure 2: Pareto front of the subgraphs generated by each algorithm, w.r.t. density and balance, on PolBooks dataset.



(a) AMAZON 2 colors: 299 subgraphs. (b) AMAZON 3 colors: 1147 subgraphs. (c) AMAZON 4 colors: 1408 subgraphs.

Figure 3: Performance of our algorithms on AMAZON dataset for 2,3 and 4 colors on 299, 1147 and 1408 samples (subgraphs) respectively. Reported are aggregates over all generated subgraphs, with unfair solutions receiving a density of 0, see Table 1.

and **FPS**) consistently outperform **2-DFSG** algorithm, despite its theoretical optimality (proved in a two-color scenario and in presence of a fair input graph), regardless of the number of considered colors. In more detail, the **FPS** heuristic is the method that achieves the maximum median density. According to Figure 3 (b and c), it is evident that for a number of colors greater than two, the spectral methods that do not rely on the paired-sweep technique (**SS** and **FS**) are not the appropriate approaches for tackling the problem. Focusing on the two-color scenario, depicted in Figure 3 (a), we have that, with the exception of **SS**, which uses the original adjacency matrix and whose distribution is skewed towards lower density values, performances of spectral heuristics are comparable, with **FPS** achieving the highest median density. Always in the two-colors scenario, we can observe that algorithms run on $(I - ff^T)A(I - ff^T)$ (**FSS** and **FPS**) respectively outperform their counterparts (**SS** and **PS**) run on A.

We report in Table 1 the percentage of instances each algorithm is not able to solve, that is, for which it does not return a fair solution and, consequently, we assign a density equal to 0.

#Colors	#Samples	SS	FSS	PS	FPS	2-DFSG
2	299	0	0.33	0	0	3.01
3	1147	73.93	95.55	0	0	5.31
4	1408	92.54	99.64	0	0	1.91

Table 1: Percentages of unfair solutions for AMAZON dataset.

The data reported in Table 1 confirm the observation that spectral methods that do not rely on the paired-sweep technique essentially fail to recover a dense fair subgraph in a context that involves more than two colors: the **SS** and **FSS** methods provided unfair solutions for almost all samples when the number of considered colors is greater than 2. As noted previously, **PS** and **FPS** cannot return unfair solutions: this is the reason behind the presence of zeros in their columns. It is worth to say that **2-DFSG** (Algorithm 2) results in an unfair solution if the original graph is unbalanced and the unconstrained densest subgraph cannot be made fair via line 4. This justifies the presence of quantities greater than zero in the last column.

AMAZON dataset	2 Colors	3 Colors	4 Colors
#Samples	299	1147	1408
2-DFSG	46388 (*101391*)	151049 (*152898*)	127834 (*75276*)
FPS	360 (*659*)	1083 (*2073*)	745 (*524*)
PS	424 (*842*)	1130 (*2106*)	775 (*572*)
FSS	465 (*861*)	1652 (*2185*)	1369 (*984*)
SS	463 (*859*)	1665 (*2216*)	1368 (*986*)

Table 2: Average and standard deviation of the running times (in msec) of all proposed methods on AMAZON dataset: 2, 3, and 4 colors.

Table 2 reports that spectral methods are faster than **2-DFSG**. Indeed, the average running time of the **2-DFSG** method is of two orders of magnitude greater than the one required by the spectral methods. This is coherent with the fact that the **2-DFSG** method requires solving the Max-Flow problem, which is computationally expensive.

Table 3 reports execution time and solution quality of all proposed methods on three not small-sized AMAZON subgraphs with 2, 3 and 4 colors each. In particular, for what concerns the quality of the provided solutions, the results provided in Table 3 are completely in line with with the information extracted from Figure 3 and Table 1. The relation among execution times are also in line with what provided in Table 2, moreover, we can see that on the considered instances (2, 3, and 4 colors, 100K nodes and 1.1M edges) the **2-DFSG** method requires slightly more than one hour of computation, against 91sec required by the paired spectral heuristics (**PS** and **FPS**). These results suggest that the spectral approaches are suitable for dealing with not small-sized graphs.

AMAZON dataset	2 Colors	3 Colors	4 Colors
#Nodes, #Edges	108230/1851733	108185/1132578	108220/1360241
2-DFSG	4126002/*0.50*	3618960/*0.34*	3991358/*0.27*
FPS	36199/*0.65*	11467/*0.45*	31988/*0.61*
PS	91582/*0.56*	39327/*0.45*	32643/*0.50*
FSS	33074/*0.51*	17358/*NoFairSol*	45465/*NoFairSol*
SS	26429/*0.21*	24161/*NoFairSol*	32324/*NoFairSol*

Table 3: Running time (in msec) and solution quality (expressed as normalized density of the retrieved fair subgraph to the optimal density of the unconstrained problem) of all proposed methods on three AMAZON subgraphs with 2, 3, and 4 colors each. Each subgraph has roughly 100K nodes and 1.1M edges.

5 CONCLUSION AND FUTURE WORK

In this work, we studied graphs with an arbitrary 2-coloring. For these graphs, the densest-fair-subgraph problem consists in finding a subgraph with maximal induced degree under the condition that both colors occur equally often. We observed that the problem is closely related to the densest-at-most-k subgraph problem and thus has similar strong inapproximability results. On the positive side, we presented an approximation algorithm under the assumption that the graph itself is fair, and a more involved spectral recovery algorithm inspired by the work of Kleindessner et al. [28] on stochastic block models.

In practice, the spectral recovery algorithm tends to dominate the approximation algorithm. We interpret these results as showing that (1) an approximation algorithm may not be the correct way to attack this problem, and (2) as previous work also suggests [28, 41], spectral relaxations seem to be an inexpensive tool to improve the fairness of algorithms geared towards recovery and learning.

Future work might consider extending this approach to more involved fairness constraints with provable guarantees. Empirically, we already observed that, although the spectral algorithms retain a good behavior both theoretically and empirically, the performance of the approximation algorithm deteriorates. We identify two key problems that may be more manageable. First, one might consider the case where the graph only has two colors, but the colors may overlap, that is, a node can be both red and blue. Clearly, the approximation results still hold in this case. Can one improve the analysis of the spectral recovery scheme, depending on the degree of overlap? Second, one might consider the case of multiple disjoint colors, each of equal size. Such considerations have been studied in clustering literature [3, 4, 8, 16]. Is it possible to derive similar results for densest subgraph?

REFERENCES

[1] Backurs, A., Indyk, P., Onak, K., Schieber, B., Vakilian, A., and Wagner, T. Scalable fair clustering. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA* (2019), pp. 405–413.

[2] Bahmani, B., Goel, A., and Munagala, K. Efficient primal-dual graph algorithms for mapreduce. In *Algorithms and Models for the Web Graph - 11th International Workshop, WAW 2014, Beijing, China, December 17-18, 2014, Proceedings* (2014), pp. 59–78.

[3] Becchetti, L., Bury, M., Cohen-Addad, V., Grandoni, F., and Schwiegelshohn, C. Oblivious dimension reduction for k-means: beyond subspaces and the johnson-lindenstrauss lemma. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019.* (2019), pp. 1039–1050.

[4] Bera, S. K., Chakrabarty, D., Flores, N., and Negahbani, M. Fair algorithms for clustering. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada* (2019), pp. 4955–4966.

[5] Bercea, I. O., Gross, M., Khuller, S., Kumar, A., Rösner, C., Schmidt, D. R., and Schmidt, M. On the cost of essentially fair clusterings. *CoRR abs/1811.10319* (2018).

[6] Bhaskara, A., Charikar, M., Chlamtac, E., Feige, U., and Vijayaraghavan, A. Detecting high log-densities: an $O(n^{1/4})$ approximation for densest k-subgraph. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010* (2010), pp. 201–210.

[7] Bhaskara, A., Charikar, M., Vijayaraghavan, A., Guruswami, V., and Zhou, Y. Polynomial integrality gaps for strong SDP relaxations of densest k-subgraph. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012* (2012), pp. 388–405.

[8] Böhm, M., Fazzone, A., Leonardi, S., and Schwiegelshohn, C. Fair clustering with multiple colors. *CoRR abs/2002.07892* (2020).

[9] Boxell, L., Gentzkow, M., and Shapiro, J. Is the internet causing political polarization? evidence from demographics. Tech. rep., National Bureau of Economic Research, mar 2017.

[10] Celis, L. E., Huang, L., and Vishnoi, N. K. Multiwinner voting with fairness constraints. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.* (2018), pp. 144–151.

[11] Celis, L. E., Straszak, D., and Vishnoi, N. K. Ranking with fairness constraints. In *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic* (2018), pp. 28:1–28:15.

[12] Chakraborty, T., Dalmia, A., Mukherjee, A., and Ganguly, N. Metrics for community analysis: A survey. *ACM Comput. Surv. 50*, 4 (2017), 54:1–54:37.

[13] Charikar, M. Greedy approximation algorithms for finding dense components in a graph. In *Approximation Algorithms for Combinatorial Optimization, Third International Workshop, APPROX 2000, Saarbrücken, Germany, September 5-8, 2000, Proceedings* (2000), pp. 84–95.

[14] Chierichetti, F., Kumar, R., Lattanzi, S., and Vassilvitskii, S. Fair clustering through fairlets. In *Proceedings of the 30th Annual Conference on Neural Information Processing Systems (NIPS)* (2017), pp. 5036–5044.

[15] Esfandiari, H., Hajiaghayi, M., and Woodruff, D. P. Brief announcement: Applications of uniform sampling: Densest subgraph and beyond. In *Proceedings of the 28th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA 2016, Asilomar State Beach/Pacific Grove, CA, USA, July 11-13, 2016* (2016), pp. 397–399.

[16] Feldman, D., Schmidt, M., and Sohler, C. Turning big data into tiny data: Constant-size coresets for k-means, PCA and projective clustering. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013* (2013), pp. 1434–1453.

[17] Feldman, M., Friedler, S. A., Moeller, J., Scheidegger, C., and Venkatasubramanian, S. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2015), pp. 259–268.

[18] Fratkin, E., Naughton, B. T., Brutlag, D. L., and Batzoglou, S. Motifcut: regulatory motifs finding with maximum density subgraphs. *Bioinformatics 22*, 14 (2006), e150–e157.

[19] Galimberti, E., Bonchi, F., and Gullo, F. Core decomposition and densest subgraph in multilayer networks. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017* (2017), pp. 1807–1816.

[20] Gibson, D., Kumar, R., and Tomkins, A. Discovering large dense subgraphs in massive graphs. In *Proceedings of the 31st International Conference on Very Large Data Bases, Trondheim, Norway, August 30 - September 2, 2005* (2005), pp. 721–732.

[21] Gionis, A., Junqueira, F., Leroy, V., Serafini, M., and Weber, I. Piggybacking on social networks. *Proceedings of the VLDB Endowment 6*, 6 (2013), 409–420.

[22] Goldberg, A. V. Finding a maximum density subgraph. Tech. Rep. UCB/CSD-84-171, EECS Department, University of California, Berkeley, 1984.

[23] Hardt, M., Price, E., and Srebro, N. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain* (2016), pp. 3315–3323.

[24] Horwitz, S. The compositional impact of team diversity on performance: Theoretical considerations. *Human Resource Development Review 4* (06 2005), 219–245.

[25] Hu, S., Wu, X., and Chan, T. H. Maintaining densest subsets efficiently in evolving hypergraphs. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017* (2017), pp. 929–938.

[26] Huang, L., Jiang, S. H., and Vishnoi, N. K. Coresets for clustering with fairness constraints. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14*

December 2019, Vancouver, BC, Canada (2019), H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds., pp. 7587–7598.

[27] Kannan, R., and Vinay, V. *Analyzing the structure of large graphs.* Rheinische Friedrich-Wilhelms-Universität Bonn Bonn, 1999.

[28] Kleindessner, M., Samadi, S., Awasthi, P., and Morgenstern, J. Guarantees for spectral clustering with fairness constraints. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA* (2019), pp. 3458–3467.

[29] Krebs, V. Polbook-network-dataset, v. krebs, unpublished. http://www.orgnet.com/.

[30] Lappas, T., Liu, K., and Terzi, E. Finding a team of experts in social networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2009), KDD '09, pp. 467–476.

[31] Manurangsi, P. Almost-polynomial ratio eth-hardness of approximating densest k-subgraph. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017* (2017), pp. 954–961.

[32] Marcolino, L. S., Jiang, A. X., and Tambe, M. Multi-agent team formation: diversity beats strength? In *Twenty-Third International Joint Conference on Artificial Intelligence* (2013).

[33] McSherry, F. Spectral partitioning of random graphs. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA* (2001), pp. 529–537.

[34] Musco, C., Musco, C., and Tsourakakis, C. E. Minimizing polarization and disagreement in social networks. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018* (2018), pp. 369–378.

[35] Nasir, M. A. U., Gionis, A., Morales, G. D. F., and Girdzijauskas, S. Fully dynamic algorithm for top-k densest subgraphs. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017* (2017), pp. 1817–1826.

[36] Ni, J., Li, J., and McAuley, J. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (Hong Kong, China, Nov. 2019), Association for Computational Linguistics, pp. 188–197.

[37] Noriega-Campero, A., Bakker, M., Garcia-Bulle, B., and Pentland, A. Active fairness in algorithmic decision making. *CoRR abs/1810.00031* (2018).

[38] Raghavendra, P., and Steurer, D. Graph expansion and the unique games conjecture. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010* (2010), pp. 755–764.

[39] Rösner, C., and Schmidt, M. Privacy preserving clustering with constraints. In *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic* (2018), pp. 96:1–96:14.

[40] Rozenshtein, P., Bonchi, F., Gionis, A., Sozio, M., and Tatti, N. Finding events in temporal networks: Segmentation meets densest-subgraph discovery. In *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018* (2018), pp. 397–406.

[41] Samadi, S., Tantipongpipat, U. T., Morgenstern, J. H., Singh, M., and Vempala, S. The price of fair PCA: one extra dimension. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.* (2018), pp. 10999–11010.

[42] Schmidt, M., Schwiegelshohn, C., and Sohler, C. Fair coresets and streaming algorithms for fair k-means. In *Approximation and Online Algorithms - 17th International Workshop, WAOA 2019, Munich, Germany, September 12-13, 2019, Revised Selected Papers* (2019), pp. 232–251.

[43] Sozio, M., and Gionis, A. The community-search problem and how to plan a successful cocktail party. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, July 25-28, 2010* (2010), pp. 939–948.

[44] Tantipongpipat, U., Samadi, S., Singh, M., Morgenstern, J. H., and Vempala, S. S. Multi-criteria dimensionality reduction with applications to fairness. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada* (2019), H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds., pp. 15135–15145.

[45] Thanh, B. L., Ruggieri, S., and Turini, F. k-nn as an implementation of situation testing for discrimination discovery and prevention. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2011), pp. 502–510.

[46] Tsourakakis, C. E., Bonchi, F., Gionis, A., Gullo, F., and Tsiarli, M. A. Denser than the densest subgraph: extracting optimal quasi-cliques with quality guarantees. In *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013* (2013), pp. 104–112.

[47] Zafar, M. B., Valera, I., Gomez Rodriguez, M., and Gummadi, K. P. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th International Conference on World Wide Web (WWW)* (2017), pp. 1171–1180.

[48] Zafar, M. B., Valera, I., Gomez-Rodriguez, M., and Gummadi, K. P. Fairness constraints: Mechanisms for fair classification. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA* (2017), pp. 962–970.

[49] Zafar, M. B., Valera, I., Gomez-Rodriguez, M., Gummadi, K. P., and Weller, A. From parity to preference-based notions of fairness in classification. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA* (2017), pp. 228–238.