Preprint of:
P. Papadakos and G. Konstantakis, bias goggles: Graph-based Computation of the Bias of Web Domains through the Eyes of Users, 42nd European Conference On Information Retrieval (ECIR 2020), Lisbon, Portugal, Apr. 2020

bias goggles: Graph-based Computation of the Bias of Web Domains through the Eyes of Users

Panagiotis Papadakos[1,2][0000−0001−8926−4229] and Giannis Konstantakis[2]

[1] Institute of Computer Science, FORTH-ICS, Greece
[2] Computer Science Department, University of Crete, Greece
papadako@ics.forth.gr, jkonstan@csd.uoc.gr

Abstract. Ethical issues, along with transparency, disinformation, and bias, are in the focus of our information society. In this work, we propose the *bias goggles* model, for computing the bias characteristics of web domains to user-defined concepts based on the structure of the web graph. For supporting the model, we exploit well-known propagation models and the newly introduced `Biased-PR` PageRank algorithm, that models various behaviours of biased surfers. An implementation discussion, along with a preliminary evaluation over a subset of the greek web graph, shows the applicability of the model even in real-time for small graphs, and showcases rather promising and interesting results. Finally, we pinpoint important directions for future work. A constantly evolving prototype of the *bias goggles* system is readily available.

Keywords: Bias · Web Graph · Propagation Models · Biased PageRank

1 Introduction

There is an increasing concern about the potential risks in the consumption of abundant biased information in online platforms like Web Search Engines (WSEs) and social networks. Terms like echo chambers and filter-bubbles [26] depict the isolation of groups of people and its aftereffects, that result from the selective and restrictive exposure to information. This restriction can be the result of helpful personalized algorithms, that suggest user connections or rank highly information relevant to the users' profile. Yet, this isolation might inhibit the growth of informed and responsible humans/citizens/consumers, and can also be the result of malicious algorithms that promote and resurrect social, religious, ethnic, and other kinds of discriminations and stereotypes.

Currently, the community focus is towards the transparency, fairness, and accountability of mostly machine learning algorithms for decision-making, classification, and recommendation in social platforms like twitter. However, social platforms and WSEs mainly act as gateways to information published on the web as common web pages (e.g., blogs and news). Unfortunately, users are unaware of the bias characteristics of these pages, except for obvious facts (e.g., a page in a political party's web site will be biased towards this party).

In this work, we propose the *bias goggles* model, where users are able to explore the biased characteristics of web domains for a specific biased concept

(i.e., a bias goggle). Since there is no objective definition of what bias and biased concepts are [27], we let users define them. For these concepts, the model computes the *support* and the *bias score* of a web domain, by considering the *support* of this domain for each aspect (i.e., dimension) of the biased concept. These *support* scores are calculated by graph-based algorithms that exploit the structure of the web graph and a set of user-defined seeds representing each aspect of bias. As a running example we will use the biased concept of greek politics, that consists of nine aspects of bias, each one representing a popular greek party, and identified by a single seed; the domain of its homepage.

In a nutshell, the main contributions of this work are:

- the *bias goggles* model for computing the bias characteristics of web domains for a user-defined concept, based on the notions of `Biased Concepts` (`BC`s), `Aspects of Bias` (`AB`s), and the metrics of the *support* of the domain for a specific `AB` and `BC`, and its *bias score* for this `BC`,
- the introduction of the `Support Flow Graph (SFG)`, along with graph-based algorithms for computing the `AB` *support* score of domains, that include adaptations of the `Independence Cascade` (`IC`) and `Linear Threshold` (`LT`) propagation models, and the new `Biased-PageRank` (`Biased-PR`) variation that models different behaviours of a biased surfer,
- an initial discussion about performance and implementation issues,
- some promising evaluation results that showcase the effectiveness and efficiency of the approach on a relatively small dataset of crawled pages, using the new `AGBR` and `AGS` metrics,
- a publicly accessible prototype of *bias goggles*.

The rest of the paper is organized as follows: the background and the related work is discussed in § 2, while the proposed model, and its notions and metrics are described in § 3. The graph-based algorithms for computing the *support* score of a domain for a specific `AB` are introduced in § 4. The developed prototype and related performance issues are discussed in § 5, while some preliminary evaluation results over a relatively small dataset of web pages are reported in § 6. Finally, § 7 concludes the paper and outlines future work.

2 Background & Related Work

Social platforms have been found to strengthen users' existing biases [21] since most users try to access information that they agree with [18]. This behaviour leads to rating bubbles when positive social influence accumulates [24] and minimizes the exposure to different opinions [31]. This is also evident in WSEs, where the personalization and filtering algorithms lead to echo chambers and filter bubbles that reinforce bias [4,12]. Remarkably, users of search engines trust more the top-ranked search results [25] and biased search algorithms can shift the voting preferences of undecided voters by as much as 20% [8].

There is an increasingly growing number of discrimination reports regarding various protected attributes (e.g., race, gender, etc.) in various domains, like in ads [29,7] and recommendation systems [13], leading to efforts for defining principles

of accountable[3], auditing [28] and de-bias algorithms [1], along with fair classifiers [14,34,6]. Tools that remove discriminating information[4], flag fake news[5], make personalization algorithms more transparent[6], or show political biases in social networks[7] also exist. Finally, a call for equal opportunities by design [16] has been raised regarding the risks of bias in the stages of the design, implementation, training and deployment of data-driven decision-making algorithms [3,11,20].

There are various efforts for measuring bias in online platforms[27]. Bias in WSEs has been measured as the deviation from the distribution of the results of a pool of search engines [23] and the coverage of SRPs towards US sites [30]. Furthermore, the presence of bias in media sources has been explored through human annotations [5], by exploiting affiliations [32], the impartiality of messages [33], the content and linked-based tracking of topic bias [22], and the quantification of data and algorithmic bias [19]. However, this is the first work that provides a model that allows users to explore the available web sources based on their own definitions of biased concepts. The approach exploits the web graph structure and can annotate web sources with bias metrics on any online platform.

3 The *bias goggles* Model

Below we describe the notions of `Biased Concepts` (`BC`s) and `Aspects of Bias` (`AB`s), along with the *support* of a domain for an `AB` and `BC`, and its *bias score* for a `BC`. Table 1 describes the used notation.

3.1 Biased Concepts (BCs) and Aspects of Bias (ABs)

The interaction with a user begins with the definition of a `Biased Concept` (`BC`), which is considered the goggles through which the user wants to explore the web domains. `BC`s are given by users and correspond to a concept that can range from a very abstract one (e.g., god) to a very specific one (e.g., political parties). For each `BC`, it is required that the users can identify at least two `Aspects of Bias` (`AB`s), representing its bias dimensions. `AB`s are given by the users and correspond to a non-empty set of seeds (i.e., domains) $\mathcal{S}$, that the user considers to fully support this bias aspect. For example, consider the homepage of a greek political party as an aspect of bias in the biased concept of the politics in Greece. Notice, that an `AB` can be part of more than one `BC`s. Typically, an `AB` is denoted by $\texttt{AB}_{\texttt{sign}(\mathcal{S})}$, where $\texttt{sign}(\mathcal{S})$ is the signature of the non-empty set of seeds $\mathcal{S}$. The $\texttt{sign}(\mathcal{S})$ is the SHA1 hash of the lexicographic concatenation of the normalized Second Level Domains (SLDs)[8] of the urls in $\mathcal{S}$. We assume that all seeds in $\mathcal{S}$ are incomparable and support with the same strength this `AB`.

[3] http://www.fatml.org/resources/principles-for-accountable-algorithms
[4] http://www.debiasyourself.org/
[5] https://www.facebook.com/help/572838089565953
[6] https://facebook.tracking.exposed/
[7] http://politecho.org/
[8] We follow the standard URL normalization method (see `https://en.wikipedia.org/wiki/URI_normalization`) and get the SLD of an url.

Symbol	Description
$\mathcal{W}$	the set of crawled Web pages
$\mathtt{p}$	a page in $\mathcal{W}$
$\mathtt{dom(p)}$	the normalized SLD of page $\mathtt{p}$
$\mathtt{doms}(\mathcal{W})$	the set of normalized SLDs in $\mathcal{W}$
$\mathtt{dom}$	an SLD in $\mathtt{doms}(\mathcal{W})$
$\mathtt{link}_{\mathtt{p,p'}}$	a link from page $\mathtt{p}$ to $\mathtt{p'} \mid \mathtt{p},\ \mathtt{p'} \in \mathcal{W}$
$\mathtt{link}_{\mathtt{dom,dom'}}$	a link from domain $\mathtt{dom}$ to $\mathtt{dom'} \mid \mathtt{dom},\ \mathtt{dom'} \in \mathtt{doms}(\mathcal{W})$
$\mathtt{links}(\mathcal{W})$	the set of crawled links between pages in $\mathcal{W}$
$\mathtt{dom(links}(\mathcal{W}))$	the set of crawled links between the domains in $\mathtt{doms}(\mathcal{W})$
$\mathtt{inv(link}_{\mathtt{p,p'}})$	the inverse link of $\mathtt{link}_{\mathtt{p,p'}}$, i.e., $\mathtt{link}_{\mathtt{p',p}}$
$\mathtt{inv(link}_{\mathtt{dom,dom'}})$	the inverse link of $\mathtt{link}_{\mathtt{dom,dom'}}$, i.e., $\mathtt{link}_{\mathtt{dom',dom}}$
$\mathtt{inv(links}(\mathcal{W}))$	the set of inverse links between the pages in $\mathcal{W}$
$\mathtt{inv(dom(links}(\mathcal{W})))$	the set of inverse links between the domains in $\mathtt{doms}(\mathcal{W})$
$\mathtt{G}(\mathcal{W})$	the graph with $\mathtt{doms}(\mathcal{W})$ as nodes and $\mathtt{dom(links}(\mathcal{W}))$ as edges
$\mathtt{outInvLinks(dom)}$	the set of $\mathtt{link}_{\mathtt{p,*}} \in \mathtt{inv(links}(\mathcal{W})) \mid \mathtt{p} \in \mathcal{W},\ \mathtt{dom(p)} = \mathtt{dom}$
$\mathtt{outInvLinks(dom,dom')}$	the set of $\mathtt{link}_{\mathtt{p,p'}} \in \mathtt{inv(links}(\mathcal{W})) \mid \mathtt{p,p'} \in \mathcal{W},\ \mathtt{dom(p)} = \mathtt{dom}, \mathtt{dom(p')} = \mathtt{dom'}$
$\mathtt{neigh(dom)}$	the set of all $\mathtt{dom'} \in \mathtt{doms}(\mathcal{W}) \mid \mathtt{link}_{\mathtt{dom,dom'}} \in \mathtt{dom(links}(\mathcal{W}))$
$\mathtt{invNeigh(dom)}$	the set of all $\mathtt{dom'} \in \mathtt{doms}(\mathcal{W}) \mid \mathtt{link}_{\mathtt{dom,dom'}} \in \mathtt{inv(dom(links}(\mathcal{W})))$
$\mathtt{w}_{\mathtt{dom,dom'}}$	the weight of the $\mathtt{link}_{\mathtt{dom,dom'}}$
$\mathtt{SFG}(\mathcal{W})$	the weighted graph with $\mathtt{doms}(\mathcal{W})$ as nodes and $\mathtt{inv(dom(links}(\mathcal{W})))$ as edges where $\mathtt{w}_{\mathtt{dom,dom'}} = \frac{\mathtt{outInvLinks(dom,dom')}}{\mathtt{outInvLinks(dom)}}$
$\mathcal{S}$	a non-empty set of normalized domain urls (i.e., seeds)
$\mathtt{sign}(\mathcal{S})$	the signature of a set of seeds
$\mathtt{AB}_{\mathtt{sign}(\mathcal{S})}$	an Aspect of Bias (AB) as identified by $\mathtt{sign}(\mathcal{S})$
$\mathtt{seeds}(\mathtt{AB}_{\mathtt{sign}(\mathcal{S})})$	the set of seeds that define $\mathtt{AB}_{\mathtt{sign}(\mathcal{S})}$
$\mathcal{A}^{\mathcal{U}}$	the universe of all available ABs
$\mathcal{A}$	a non-empty set of ABs $\mid \mathcal{A} \subseteq \mathcal{A}^{\mathcal{U}}, \lvert\mathcal{A}\rvert \geq 2$
$\mathtt{BC}_{\mathcal{A}}$	a Biased Concept (BC) as defined by $\mathcal{A}$
$\boldsymbol{d}_{\mathcal{A}}$	an $\lvert\mathcal{A}\rvert$-dimensional vector holding the ABs of $\mathtt{BC}_{\mathcal{A}}$
$\boldsymbol{d}_{\mathcal{A}}[i]$	the AB stored in dimension i of $\boldsymbol{d}_{\mathcal{A}}$
$\mathtt{sup}(\boldsymbol{d}_{\mathcal{A}}[i], \mathtt{dom})$ $\mathtt{sup}(\mathtt{AB}_{\mathtt{sign}(\mathcal{S})}, \mathtt{dom})$	*support* score of domain $\mathtt{dom}$ regarding AB $\boldsymbol{d}_{\mathcal{A}}[i]$
$\boldsymbol{s}^{\mathtt{dom}}_{\boldsymbol{d}_{\mathcal{A}}}$	vector holding *support* scores $\forall \boldsymbol{d}_{\mathcal{A}}[i] \in \boldsymbol{d}_{\mathcal{A}}$ for domain $\mathtt{dom}$
$\boldsymbol{s}^{\mathtt{dom}}_{\boldsymbol{d}_{\mathcal{A}}}[i]$	*support* score of dimension i of $\boldsymbol{s}^{\mathtt{dom}}_{\boldsymbol{d}_{\mathcal{A}}}$
$\mathtt{sup}(\boldsymbol{s}^{\mathtt{dom}}_{\boldsymbol{d}_{\mathcal{A}}})$ $\mathtt{sup}(\mathtt{BC}_{\mathcal{A}}, \mathtt{dom})$	*support* score of domain $\mathtt{dom}$ regarding BC $\mathtt{BC}_{\mathcal{A}}$
$\mathtt{bias}(\mathtt{BC}_{\mathcal{A}}, \mathtt{dom})$ $\mathtt{bias}(\boldsymbol{s}^{\mathtt{dom}}_{\boldsymbol{d}_{\mathcal{A}}})$	*bias score* of $\mathtt{dom}$ for BC $\mathtt{BC}_{\mathcal{A}}$
$\mathbf{1}_{\lvert\mathcal{A}\rvert}$	An $\lvert\mathcal{A}\rvert$-dimensional vector with *support* 1 in all dimensions

Table 1. Description of the used notation. The first part describes the notation used for the Web Graph, while the second the notation for the proposed model.

Assumption 1. ***Incomparable Seeds Support.*** *The domains in the set of seeds $\mathcal{S}$ are incomparable and equally supportive of the* $\mathtt{AB}_{\mathtt{sign}(\mathcal{S})}$.

The user-defined BC of the set of ABs $\mathcal{A} \subseteq \mathcal{A}^{\mathcal{U}}$, where $|\mathcal{A}| \geq 2$ and $\mathcal{A}^{\mathcal{U}}$ the universe of all possible ABs in the set of domains $\mathtt{doms}(\mathcal{W})$ of the crawled pages $\mathcal{W}$, is denoted by $\mathtt{BC}_{\mathcal{A}}$ and is represented by the pair $< \boldsymbol{d}_{\mathcal{A}}, \mathtt{desc}_{\mathcal{A}} >$. The $\boldsymbol{d}_{\mathcal{A}}$ is an $|\mathcal{A}|$-dimensional vector with $|\mathcal{A}| \geq 2$, holding all $\mathtt{AB}_{\mathtt{sign}(\mathcal{S})} \in \mathcal{A}$ of this BC in lexicographic order. $\mathtt{desc}_{\mathcal{A}}$ is a user-defined textual description of this BC. In this work, we assume that all ABs of any BC are orthogonal and unrelated.

Assumption 2. ***Orthogonality of Aspects of Bias.*** *All* ABs *in a user-defined* BC *are considered orthogonal.*

Using the notation, our running example is denoted as $\texttt{BC}_{\mathcal{R}} =< \texttt{d}_{\mathcal{R}}, \texttt{desc}_{\mathcal{R}} >$, where $\texttt{d}_{\mathcal{R}}$ is a vector that holds lexicographically the SHA1 signatures of the nine ABs singleton seeds of greek political parties $\mathcal{R} = \{$ {"anexartitoiellines.gr"}, {"antidiaploki.gr"}, {"elliniki − lisi.gr"}, {"kke.gr"}, {"mera25.gr"}, {"nd.gr"}, {"syriza.gr"}, {"topotami.gr"}, {"xryshaygh.com"}$\}$, and $\texttt{desc}_{\mathcal{R}}$ = "politics in Greece" is its description.

3.2 Aspects of Bias Support & Biased Concepts Support

A core metric in the proposed model is the *support* score of a domain dom to an aspect of bias $\texttt{AB}_{\texttt{sign}(\mathcal{S})}$, denoted as $\texttt{sup}(\texttt{AB}_{\texttt{sign}(\mathcal{S})}, \texttt{dom})$. The *support* score ranges in $[0, 1]$, where 0 denotes an unsupportive domain for the corresponding AB, and 1 a fully supportive one. We can identify three approaches for computing this *support* for a dataset of web pages: a) the graph-based ones that exploit the web graph structure and the relationship of a domain with the domains in $\texttt{seeds}(\texttt{AB}_{\texttt{sign}(\mathcal{S})})$, b) the content-based ones that consider the textual information of the respective web pages, and c) the hybrid ones that take advantage of both the graph and the content information. In this work, we focus only on graph-based approaches and study two frequently used propagation models, the Independence Cascade (IC) and Linear Threshold (LT) models, along with the newly introduced Biased-PageRank (Biased-PR), that models various behaviours of biased surfers. The details about these algorithms are given in § 4.

In the same spirit, we are interested about the *support* of a specific domain dom to a biased concept $\texttt{BC}_{\mathcal{A}}$, denoted by $\texttt{sup}(\texttt{BC}_{\mathcal{A}}, \texttt{dom})$. The basic intuition is that we need a metric that shows the relatedness and *support* to all or any of the aspects in $\mathcal{A}$, which can be interpreted as the relevance of this domain with any of the aspects of the biased concept $\texttt{BC}_{\mathcal{A}}$. A straightforward way to measure it, is the norm of the $\boldsymbol{s}^{\texttt{dom}}_{\boldsymbol{d}_{\mathcal{A}}}$ vector that holds the *support* scores of dom for each AB in $\mathcal{A}$, normalized by the norm of the $\mathbf{1}_{|\mathcal{A}|}$ vector. This vector holds the *support* scores of a 'virtual' domain that fully supports all bias aspects in $\texttt{BC}_{\mathcal{A}}$. Specifically,

$$\texttt{sup}(\texttt{BC}_{\mathcal{A}}, \texttt{dom}) = \frac{\left\|\boldsymbol{s}^{\texttt{dom}}_{\boldsymbol{d}_{\mathcal{A}}}\right\|}{\left\|\mathbf{1}_{|\mathcal{A}|}\right\|} = \frac{\sqrt{\sum_{i=1}^{|\mathcal{A}|} \boldsymbol{s}^{\texttt{dom}}_{\boldsymbol{d}_{\mathcal{A}}}[i]^2}}{\sqrt{|\mathcal{A}|}} = \frac{\sqrt{\sum_{\texttt{AB}_{\texttt{sign}(\mathcal{S})} \in \mathcal{A}} \texttt{sup}(\texttt{AB}_{\texttt{sign}(\mathcal{S})}, \texttt{dom})^2}}{\sqrt{|\mathcal{A}|}} \tag{1}$$

The $\texttt{sup}(\texttt{BC}_{\mathcal{A}}, \texttt{dom})$ value ranges in $[0, 1]$. By using the above formula two domains might have similar *support* scores for a specific BC, while the *support* scores for the respective aspects might differ greatly. For example, consider two domains dom and dom′, with dom fully supporting only one aspect in $\mathcal{A}$ and dom′ fully supporting another aspect in $\mathcal{A}$. Then $\texttt{sup}(\texttt{BC}_{\mathcal{A}}, \texttt{dom}) \sim \texttt{sup}(\texttt{BC}_{\mathcal{A}}, \texttt{dom}')$. Below we introduce the *bias score* of a domain regarding a specific BC, as a way to capture the leaning of a domain to specific ABs of a BC.

3.3 Bias Score of Domain Regarding a Biased Concept

The *bias score* of a domain regarding a BC tries to capture how biased the domain is over any of its ABs, and results from the *support* scores that the domain has

for each aspect of the BC. For example, consider a domain dom that has a rather high *support* for a specific AB, but rather weak ones for the rest ABs of a specific BC. This domain is expected to have a high *bias score*. On the other hand, the domain dom′ that has similar *support* for all the available ABs of a BC can be considered to be unbiased regarding this specific BC.

We define the *bias score* of a domain dom for $\texttt{BC}_{\mathcal{A}}$ as the distance of the $\boldsymbol{s}^{\texttt{dom}}_{\boldsymbol{d}_{\mathcal{A}}}$ vector from the $\mathbf{1}_{|\mathcal{A}|}$ vector, multiplied by its support $\texttt{sup}(\texttt{BC}_{\mathcal{A}}, \texttt{dom})$. The *bias score* takes values in $[0, 1]$. Specifically,

$$\texttt{bias}(\boldsymbol{s}^{\texttt{dom}}_{\boldsymbol{d}_{\mathcal{A}}}) = \texttt{dist}(\boldsymbol{s}^{\texttt{dom}}_{\boldsymbol{d}_{\mathcal{A}}}, \mathbf{1}_{|\mathcal{A}|}) * \texttt{sup}(\texttt{BC}_{\mathcal{A}}, \texttt{dom}) \tag{2}$$

We use the cosine similarity to define the distance metric, as shown below:

$$\texttt{dist}(\boldsymbol{s}^{\texttt{dom}}_{\boldsymbol{d}_{\mathcal{A}}}, \mathbf{1}_{|\mathcal{A}|}) = 1 - \texttt{cosSim}(\boldsymbol{s}^{\texttt{dom}}_{\boldsymbol{d}_{\mathcal{A}}}, \mathbf{1}_{|\mathcal{A}|}) = 1 - \frac{\boldsymbol{s}^{\texttt{dom}}_{\boldsymbol{d}_{\mathcal{A}}} \cdot \mathbf{1}_{|\mathcal{A}|}}{\left\|\boldsymbol{s}^{\texttt{dom}}_{\boldsymbol{d}_{\mathcal{A}}}\right\| \left\|\mathbf{1}_{|\mathcal{A}|}\right\|} \tag{3}$$

4 Graph-based Computation of Aspects of Bias Support

In this section, we discuss the graph-based algorithms that we use for computing the *support* score of a domain regarding a specific AB. We focus on the popular Independence Cascade (IC) and Linear Threshold (LT) propagation models, along with the newly introduced Biased-PageRank (Biased-PR) algorithm.

Let $\mathcal{W}$ be the set of crawled web pages, $\texttt{doms}(\mathcal{W})$ the set of normalized SLDs in $\mathcal{W}$, $\texttt{links}(\mathcal{W})$ the set of crawled links between the domains in $\texttt{doms}(\mathcal{W})$, and $\texttt{G}(\mathcal{W})$ the corresponding graph with $\texttt{doms}(\mathcal{W})$ as nodes and $\texttt{links}(\mathcal{W})$ as edges. With $\texttt{link}_{\texttt{dom},\texttt{dom}'}$ we denote a link from domain dom to $\texttt{dom}' \mid \texttt{dom}, \texttt{dom}' \in \texttt{doms}(\mathcal{W})$, while $\texttt{inv}(\texttt{link}_{\texttt{dom},\texttt{dom}'})$ inverses the direction of a link and $\texttt{inv}(\texttt{links}(\mathcal{W}))$ is the set of inverse links in $\mathcal{W}$. Furthermore, for the links we assume that:

Assumption 3. ***Equally Supportive Links.*** *Any link* $\texttt{link}_{\texttt{dom},\texttt{dom}'}$ *from the domain* dom *to the domain* dom′ *in the set of crawled domains* $\mathcal{W}$*, is considered to be of supportive nature (i.e.,* dom *has the same support stance as* dom′ *for any* AB*). All links in a domain are equally supportive and independent of the importance of the page they appear in.*

Although the above assumption might not be precise, since links from a web page to another are not always of supportive nature (e.g., a web page critizing another linked one), or of the same importance (e.g., links in the homepage versus links deeply nested in a site), it suffices for the purposes of this first study of the model. Identification of the nature of links and the importance of the pages they appear is left as future work. Given that the assumption holds, part or whole of the *support* of dom′ regarding any AB can flow to dom through $\texttt{inv}(\texttt{link}_{\texttt{dom},\texttt{dom}'})$. Specifically, we define the **Support Flow Graph** as:

Support Flow Graph (SFG) Definition. *The* SFG *of a set of web pages* $\mathcal{W}$ *is the weighted graph that is created by inversing the links in* $\texttt{G}(\mathcal{W})$ *(i.e., the graph with* $\texttt{doms}(\mathcal{W})$ *as nodes and* $\texttt{inv}(\texttt{links}(\mathcal{W}))$ *as edges). The weight of each edge is*

$\mathtt{w}_{\mathtt{dom},\mathtt{dom}'} = \frac{\mathtt{outInvLinks(dom,dom')}}{\mathtt{outInvLinks(dom)}}$ *(i.e., the number of outgoing inverse links of pages in the domain* dom *that link to pages in the domain* dom′*, divided by the total outgoing inverse links of pages in the domain* dom*), and takes a value in* $[0, 1]$.

So, given an $\mathtt{SFG}(\mathcal{W})$ and the $\mathtt{seeds}(\mathtt{AB}_{\mathtt{sign}(\mathcal{S})})$ of an AB we can now describe how the *support* flows in the nodes of the $\mathtt{SFG}(\mathcal{W})$ graph. All algorithms described below return a map M holding $\mathtt{sup}(\mathtt{AB}_{\mathtt{sign}(\mathcal{S})}, \mathtt{dom})\ \forall\ \mathtt{dom} \in \mathtt{doms}(\mathcal{W})$.

Algorithm 1: IC *Support* Computation

input : SFG($\mathcal{W}$) : the Support Flow Graph of $\mathcal{W}$
seeds($\mathtt{AB}_{\mathtt{sign}(\mathcal{S})}$) : the set of seeds of $\mathtt{AB}_{\mathtt{sign}(\mathcal{S})}$
n : the number of experiments
output : a map M holding $\mathtt{sup}(\mathtt{AB}_{\mathtt{sign}(\mathcal{S})}, \mathtt{dom})\ \forall\ \mathtt{dom} \in \mathtt{doms}(\mathcal{W})$

```
1  L ← ∅                                          // list holding the support maps of each experiment
2  for i ← 1 to n do                                                          // for each experiment
3  |  A ← seeds(AB_sign(S))                                  // set of active nodes for next iteration
4  |  I ← doms(W) \ A                                                  // set of inactive nodes
5  |  M ← mapWithZeros(doms(W))                                // map with 0 support for domains
6  |  while A ≠ ∅ do
7  |  |  C ← A                                                  // active nodes in this iteration
8  |  |  A ← ∅                                                  // active nodes in next iteration
9  |  |  foreach dom ∈ C do                                     // for each current active domain
10 |  |  |  N ← invNeigh(dom) ∩ I                    // get all inactive inverse neighbors
11 |  |  |  foreach dom' ∈ N do                                                 // for each neighbor
12 |  |  |  |  r ← random(0, 1))                          // get a random value in [0,1]
13 |  |  |  |  if r ≤ w_dom,dom' then                                 // successful experiment
14 |  |  |  |  |  A ← A ∪ {dom'}                    // activate node for next iteration
15 |  |  |  |  |  M ← setOne(M, dom')                          // set dom' support to 1
16 |  |  |  |  |  I ← I \ {dom'}                          // remove dom' from inactive
17 |  |  L ← L ∪ {M}                                  // hold support values map to list
18 return average(L)                                  // map with average support values
```

4.1 Independence Cascade (IC) Model

The IC propagation model was introduced by Kempe et al. [17], and a number of variations have been proposed in the bibliography. Below, we describe the basic form of the model as adapted to our needs. In the IC propagation model, we run n experiments. Each run starts with a set of activated nodes, in our case the $\mathtt{seeds}(\mathtt{AB}_{\mathtt{sign}(\mathcal{S})})$, that fully *support* the $\mathtt{AB}_{\mathtt{sign}(\mathcal{S})}$. In each iteration there is a history independent and non-symmetric probability of activating the neighbors of the activated nodes associated with each edge, flowing the *support* to the neighbors of the activated nodes in the $\mathtt{SFG}(\mathcal{W})$. This probability is represented by the weights of the links of an activated node to its neighbors, and each node, once activated, can then activate its neighbors. The nodes and their neighbors are selected in arbitrary order. Each experiment stops when there are no new activated nodes. After n runs we compute the average *support* score of nodes, i.e., $\mathtt{sup}(\mathtt{AB}_{\mathtt{sign}(\mathcal{S})}, \mathtt{dom})\ \forall\ \mathtt{dom} \in \mathtt{doms}(\mathcal{W})$. The algorithm is given in Alg. 1.

4.2 Linear Threshold (LT) Model

The LT model is another widely used propagation model. The basic difference from the IC model is that for a node to become active we have to consider

the *support* of all neighbors, which must be greater than a threshold $\theta \in [0, 1]$, serving as the resistance of a node to its neighbors joint *support*. Again, we use the *support* probabilities represented by the weights of the `SFG` links. The full algorithm, which is based on the static model introduced by Goyal et al. [10], is given in Alg. 2. In each experiment the thresholds θ get a random value.

Algorithm 2: `LT` *Support* Computation

input : $\texttt{SFG}(\mathcal{W})$: the Support Flow Graph of $\mathcal{W}$
$\texttt{seeds}(\texttt{AB}_{\texttt{sign}(\mathcal{S})})$: the set of seeds of $\texttt{AB}_{\texttt{sign}(\mathcal{S})}$
n : the number of experiments
output : a map M holding $\texttt{sup}(\texttt{AB}_{\texttt{sign}(\mathcal{S})}, \texttt{dom})\ \forall\ \texttt{dom} \in \texttt{doms}(\mathcal{W})$

```
L ← ∅                                          // list holding the support maps of each experiment
for i ← 1 to n do                                                          // for each experiment
    N ← seeds(AB_sign(S))                          // set of active nodes for next iteration
    A ← ∅                                                    // set of all active nodes
    I ← doms(W) \ N                                          // set of inactive nodes
    M ← mapWithZeros(doms(W))                        // map with 0 support for domains
    T ← mapWithRandom(doms(W)                 // random value θ in [0,1] for each node
    while N ≠ ∅ do
        C ← N                                          // active nodes in this iteration
        N ← ∅                                          // active nodes in next iteration
        A ← A ∪ C                                                  // add to all nodes
        foreach dom ∈ (⋃_{c∈C} invNeigh(c) ∩ I) do     // for inactive invNeigh of active
            N ← neigh(dom) ∩ N                                 // get all active neighbors
            jointSup ← 1 − ∏_{dom'∈N} (1 − w_{dom,dom'}) // compute joint support value in [0, 1]
            if jointSup ≥ getValue(T, dom) then // joint support bigger than threshold
                N ← N ∪ {dom}                          // activate node for next iteration
                M ← setOne(M, dom)                                 // set dom support to 1
                I ← I \ {dom}                                // remove node from inactive
    L ← L ∪ {M}                                      // hold support values map to list
return average(L)                                    // map with average support values
```

4.3 `Biased-PageRank` (`Biased-PR`) Model

We introduce the `Biased-PR` variation of PageRank [9] that models a biased surfer. The biased surfer always starts from the biased domains (i.e., the seeds of an `AB`), and either visits a domain linked by the selected seeds or one of the biased domains again, with some probability that depends on the modeled behaviour. The same process is followed in the next iterations. The `Biased-PR` differs to the original PageRank in two ways. The first one is how the score (*support* in our case) of the seeds is computed at any step. The *support* of all domains is initially 0, except from the *support* of the seeds that have the value $\texttt{init}_{\texttt{seeds}} = 1$. At any step, the *support* of each seed is the original PageRank value, increased by a number that depends on the behaviour of the biased surfer. We have considered three behaviours: a) the `Strongly Supportive (SS)` one, where the support is increased by $\texttt{init}_{\texttt{seeds}}$ and models a constantly strongly biased surfer, b) the `Decreasingly Supportive (DS)` one, where the support is increased by $\texttt{init}_{\texttt{seeds}}/\texttt{iter}$, modeling a surfer that becomes less biased the more pages he/she visits, and c) the `Non-Supportive (NS)` one, with no increment, modeling a surfer that is biased only on the initial visiting pages, and afterwards the *support* score is computed as in the original PageRank. `Biased-PR` differs also on how the biased surfer is teleported to another domain when he/she reaches a sink (i.e., a domain that has no outgoing links). The surfer randomly teleports

with the same probability to a domain in any distance from the seeds. If a path from a node to any of the seeds does not exist, the distance of the node is the maximum distance of a connected node increased by one. Since the number of nodes at a certain distance from the seeds increase as we move away from the seeds, the teleporting probability for a node is greater the closer the node is to the seeds. We expect slower convergence for `Biased-PR` than the original PageRank, due to the initial zero scores of non-seed nodes. The algorithm is given in Alg.3.

Algorithm 3: `Biased-PR` *Support* Computation

input : $\texttt{SFG}(\mathcal{W})$: the Support Flow Graph of $\mathcal{W}$
$\texttt{seeds}(\texttt{AB}_{\texttt{sign}(\mathcal{S})})$: the set of seeds of $\texttt{AB}_{\texttt{sign}(\mathcal{S})}$
`behaviour` : bias user behaviour. One of `SS`, `DS`, `NS`
$\theta_{\texttt{conv}}$: converge threshold
`d` : damping factor
output : a map M holding $\texttt{sup}(\texttt{AB}_{\texttt{sign}(\mathcal{S})}, \texttt{dom})\ \forall\ \texttt{dom} \in \texttt{doms}(\mathcal{W})$

```
   // ----- INIT PART -----
 1 init_s = 1                                                  // initial support of seeds
 2 iter ← 0                                                    // counts iterations
 3 conv ← false                                  // holds if the algorithm has converged
 4 M ← mapWithZeros(doms(W))                          // map with 0 support for domains
 5 foreach dom ∈ seeds(AB_sign(S)) do                 // initialize support for each seed
 6 |  M ← addSupport(M, dom, init_s)     // add init_s support value for domain dom in map M
   // D is a map with keys the distinct minimum distances of nodes from seeds in SFG,
   // and values the number of nodes with this minimum distance
 7 D ← distinctMinDistancesAndCounts(doms(W), seeds(AB_sign(S)))
 8 E ← mapWithZeros(doms(W))            // map that holds the teleportation probabilities
 9 foreach dom ∈ doms(W) do              // find teleportation probability for each node
10 |  minDist = findMinDistanceFromSeeds(dom, seeds(AB_sign(S)))
11 |  E ← addProbability(E, dom, 1/(minDist * getValue(D, minDist)))   // compute probability
   // ----- MAIN PART -----
12 while !conv do                                             // alg has not finished
13 |  M' ← mapWithZeros(doms(W))          // new map with ranks (0 support for domains)
14 |  foreach dom ∈ doms(W) do                                          // for each node
15 |  |  tele ← getValue(D, dom)                    // find teleport probability of node
16 |  |  sup ← Σ_{dom'∈neigh(dom)} (getValue(M, dom')/w_{dom,dom'})     // compute joint support
17 |  |  if dom ∈ seeds(AB_sign(S)) && behaviour == SS then     // support to seeds - SS
18 |  |  |  sup ← sup + init_s
19 |  |  if dom ∈ seeds(AB_sign(S)) && behaviour == DS then     // support to seeds - DS
20 |  |  |  sup ← sup + init_s/(iter + 1)
21 |  |  final = (1 − d) * tele + d * sup                       // final support score
22 |  |  M' ← addSupport(M, dom, final)                          // add support to map
23 |  M' ← normalize(M')                                          // normalize values
24 |  conv ← checkConvergence(M, M', θ_conv)        // all supports changed less than θ_conv?
25 |  M ← M'                                          // prepare map for next iteration
26 |  iter ← iter + 1                                             // increase counter
27 return M                                                // map with support values
```

5 Perfomance & Implementation Discussion

Due to size restrictions we provide a rather limited discussion about the complexities and the cost of tuning the parameters of each algorithm. The huge scale of the web graph has the biggest performance implication to the the graph-based computation of the `ABs` *support*. What is encouraging though, is that the algorithms are applied over the compact `SFG` graph, that contains the SLDs of the pages and their corresponding links. The complexity of `IC` is in $\mathcal{O}(n * |\texttt{doms}\mathcal{W}| * |\texttt{dom}(\texttt{links}(\mathcal{W})|)$, where n is the number of experiments. `LT` is much slower though since we have

to additionally consider the joint *support* of the neighbors of a node. Finally, the `Biased-PR` converges slower than the original PageRank, since the algorithm begins only with the seeds, spreading the support to the rest nodes. Also, we must consider the added cost of computing the shortest paths of the nodes from the seeds. For the relatively small `SFG` used in our study (see § 6), the `SS` converges much faster than the `DS` and `NS`, which need ten times more iterations.

For newly introduced `AB`s though, the computation of the *support* scores of the domains can be considered an offline process. Users can submit `AB`s and `BC`s into the *bias goggles* system and get notified when they are ready for use. However, what is important is to let users explore in real-time the domains space for any precomputed and commonly used `BC`s. This can be easily supported by providing efficient ways to store and retrieve the signatures of already known `BC`s, along with the computed *support* scores of domains of available `AB`s. Inverted files and trie-based data structures (e.g., the space efficient burst-tries [15] and the cache-consious hybrid or pure HAT-tries[2]) over the SLDs and the signatures of the `AB`s and `BC`s, can allow the fast retrieval of offsets in files where the *support* scores and the related metadata are stored. Given the above, the computation of the *bias score* and the *support* of a `BC` for a domain is lightning fast. We have implemented a prototype[9] that allows the exploration of predefined `BC`s over a set of mainly greek domains. The prototype offers a REST API for retrieving the bias scores of the domains, and exploits the open-source project crawler4j[10]. We plan to improve the prototype, by allowing users to search and ingest `BC`s, `AB`s and domains of interest, and develop a user-friendly browser plugin on top of it.

6 Experimental Evaluation Discussion

Evaluating such a system is a rather difficult task, since there are no formal definitions of what bias in the web is, and there are no available datasets for evaluation. As a result, we based our evaluation over `BC`s for which it is easy to find biased sites. We used two `BC`s for our experiments, the *greek politics* (`BC1`) with 9 `AB`s, and the *greek football* (`BC2`) with 6 `AB`s. For these `BC`s, we gathered well known domains, generally considered as fully supportive of only one of the `AB`s, without inspecting though their link coverage to the respective seeds, to avoid any bias towards our graph based approach. Furthermore, we did not include the original seeds to this collection. In total, we collected 50 domains for `BC1` and 65 domains for `BC2`, including newspapers, radio and television channels, blogs, pages of politicians, etc. This collection of domains is our gold standard.

We crawled a subset of the greek web by running four instances of the crawler: one with 383 sites related to the greek political life, one with 89 sport related greek sites, one with the top-300 popular greek sites according to Alexa, and a final one containing 127 seeds related to big greek industries. We black-listed popular sites like facebook and twitter to control the size of our data and avoid crawling non-greek domains. The crawlers were restricted to depth seven for each domain, and free to follow any link to external domains. In total we downloaded

[9] http://pangaia.ics.forth.gr/bias-goggles
[10] https://github.com/yasserg/crawler4j

Alg.	n	BC1 - Political Parties (9 ABs) t (s)	bias AGBR	bias AGS	BC2 - Sports Teams (6 ABs) t (s)	bias AGBR	bias AGS
IC	1000	**0.399**	217.571	0.2619	**0.287**	320.408	0.1362
	$k/2$	0.622	230.798	0.3024	0.497	326.005	0.1365
	k	0.945	**234.535**	0.361	0.681	328.667	0.1675
LT	1000	129.0	230.611	0.2354	87.5	322.585	0.1528
	$k/2$	512.9	230.064	0.3621	322.5	**328.698**	0.1364
	k	966.1	231.087	0.3626	663.1	327.786	0.1659
Biased-PR	SS $(40, 31)$	34.8	227.999	**0.5569**	17.9	261.788	**0.4745**
	DSS $(319, 391)$	260.5	129.829	0.3730	219.4	163.165	0.4344
	NSS $(306, 458)$	231.4	32.602	0.3041	207.6	34.905	0.4052

Table 2. Experimental results over two BCs.

893,095 pages including 531,296,739 links, which lead to the non-connected SFG graph with 90,419 domains, 288,740 links (on average 3.1 links per domain) and a diameter $k = 7,944$. More data about the crawled pages, the gold standard, and the SFG graph itself are available in the prototype's site.

Below we report the results of our experiments over an i7-5820K 3.3GHz system, with 6 cores, 15MB cache and 16GB RAM memory, and a 6TB disk. For each of the two BCs and for each algorithm, we run experiments for various iterations n and Biased-PR variations, for the singleton ABs of the 9 political parties and 6 sports teams. For Biased-PR we evaluate all possible behaviours of the surfer using the parameters $\theta_{conv} = 0.001$ and $d = 0.85$. We also provide the average number of iterations for convergence over all ABs for Biased-PR. We report the run times in seconds, along with the metrics Average Golden Bias Ratio (AGBR) and Average Golden Similarity (AGS), that we introduce in this work. The AGBR is the ratio of the average bias score of the golden domains, as computed by the algorithms for a specific BC, divided by the average bias score of all domains for this BC. The higher the value, the more easily we can discriminate the golden domains from the rest. On the other hand, the AGS is the average similarity of the golden domains to their corresponding ABs. The higher the similarity value, the more biased the golden domains are found to be by our algorithms towards their aspects. A high similarity score though, does not imply high support for the golden domains or high disimilarity for the rest. The perfect algorithm will have high values for all metrics. The results are shown in Table 2.

The difference in BC1 and BC2 results implies a less connected graph for BC2 (higher AGBR values for BC2), where the support flows to less domains, but with a greater interaction between domains supporting different aspects (smaller AGS values). What is remarkable is the striking time performance of IC, suggesting that it can be used in real-time and with excellent results (at least for AGBR). On the other hand, the LT is a poor choice, being the slowest of all and dominated in any aspect by IC. Regarding the Biased-PR only the SS variation offers exceptional performance, especially for AGS. The DS and NS variations are more expensive and have the worst results regarding AGBR, especially the NSS that avoids bias. In most cases, algorithms benefit from more iterations. The SS variation of Biased-PR

needs only 40 iterations for BC1 and 31 for BC2 to converge, proving that less nodes are affected by the seeds in BC2. Generally, the IC and the SS variation of Biased-PR are the best options, with the IC allowing the real-time ingestion of ABs. But, we need to evaluate the algorithms in larger graphs and for more BCs.

We also manually inspected the top domains according to the *bias* and *support* scores for each algorithm and each BC. Generally the support scores of the domains were rather low, showcasing the value of other support cues, like the content and the importance of pages that links appear in. In the case of BC1, except from the political parties, we found various blogs, politicians homepages, news sites, and also the national greek tv channel, being biased to a specific political party. In the case of BC2 we found the sport teams, sport related blogs, news sites, and a political party being highly biased towards a specific team, which is an interesting observation. In both cases we also found various domains with high support to all ABs, suggesting that these domains are good unbiased candidates. Currently, the *bias goggles* system is not able to pinpoint false positives (i.e pages with non supportive links) and false negatives (i.e., pages with content that supports a seed without linking to it), since there is no content analysis. We are certain that such results can exist, although we were not able to find such an example in the top results of our study. Furthermore, we are not able to distinguish links that can frequently appear in users' content, like in the signatures of forum members.

7 Conclusion & Future Work

In this work, we introduce the *bias goggles* model that facilitates the important task of exploring the bias characteristics of web domains to user-defined biased concepts. We focus only on graph-based approaches, using popular propagation models and the new Biased-PR PageRank variation that models biased surfers behaviours. We propose ways for the fast retrieval and ingestion of aspects of bias, and offer access to a developed prototype. The results show the efficiency of the approach, even in real-time. A preliminary evaluation over a subset of the greek web and a manually constructed gold standard of biased concepts and domains, shows promising results and interesting insights that need futher research.

In the future, we plan to explore variations of the proposed approach where our assumptions do not hold. For example, we plan to exploit the supportive, neutral or oppositive nature of the available links, as identified by sentiment analysis methods, along with the importance of the web pages they appear in. Content-based and hybrid approaches for computing the *support* scores of domains are also in our focus, as well as the exploitation of other available graphs, like the graph of friends, retweets, etc. In addition interesting aspects include how the *support* and *bias scores* of multiple BCs can be composed, providing interesting insights about possible correlations of different BCs, as well as how the *bias* scores of domains change over time. Finally, our vision is to integrate the approach in a large scale WSE/social platform/browser, in order to study how users define bias, create a globally accepted gold standard of BCs, and explore how such tools can affect the consumption of biased information. In this way, we will be able to evaluate and tune our approach in real-life scenarios, and mitigate any performance issues.

References

1. G. Adomavicius, J. Bockstedt, C. Shawn, and J. Zhang. *De-biasing user preference ratings in recommender systems*, volume 1253, pages 2–9. CEUR-WS, 2014.
2. N. Askitis and R. Sinha. Hat-trie: a cache-conscious trie-based data structure for strings. In *Proceedings of the thirtieth Australasian conference on Computer science-Volume 62*, pages 97–105. Australian Computer Society, Inc., 2007.
3. T. Bolukbasi, K. Chang, J. Y. Zou, V. Saligrama, and A. Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *CoRR*, abs/1607.06520, 2016.
4. E. Bozdag. Bias in algorithmic filtering and personalization. *Ethics and Inf. Technol.*, 15(3):209–227, Sept. 2013.
5. C. Budak, S. Goel, and J. M. Rao. Fair and balanced? quantifying media bias through crowdsourced content analysis. *Public Opinion Quarterly*, 80(S1):250–271, 2016.
6. S. Corbett-Davies, E. Pierson, A. Feller, S. Goel, and A. Huq. Algorithmic decision making and the cost of fairness. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 797–806. ACM, 2017.
7. C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. S. Zemel. Fairness through awareness. In *ITCS*, pages 214–226, 2012.
8. R. Epstein and R. E. Robertson. The search engine manipulation effect (seme) and its possible impact on the outcomes of elections. *PNAS*, 112(20), 2015.
9. D. F. Gleich. Pagerank beyond the web. *SIAM Review*, 57(3):321–363, 2015.
10. A. Goyal, F. Bonchi, and L. V. Lakshmanan. Learning influence probabilities in social networks. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 241–250. ACM, 2010.
11. S. Hajian, F. Bonchi, and C. Castillo. Algorithmic bias: From discrimination discovery to fairness-aware data mining. In *KDD*, pages 2125–2126. ACM, 2016.
12. A. Hannak, P. Sapiezynski, A. Molavi Kakhki, B. Krishnamurthy, D. Lazer, A. Mislove, and C. Wilson. Measuring personalization of web search. In *WWW*, pages 527–538. ACM, 2013.
13. A. Hannak, G. Soeller, D. Lazer, A. Mislove, and C. Wilson. Measuring price discrimination and steering on e-commerce web sites. In *Internet Measurement Conference*, pages 305–318, 2014.
14. M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. In *NIPS*, pages 3315–3323, 2016.
15. S. Heinz, J. Zobel, and H. E. Williams. Burst tries: a fast, efficient data structure for string keys. *ACM Transactions on Information Systems (TOIS)*, 20(2):192–223, 2002.
16. W. House. Big data: A report on algorithmic systems, opportunity, and civil rights. *Washington, DC: Executive Office of the President, White House*, 2016.
17. D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.
18. D. Koutra, P. N. Bennett, and E. Horvitz. Events and controversies: Influences of a shocking news event on information seeking. In *WWW*, pages 614–624, 2015.
19. J. Kulshrestha, M. Eslami, J. Messias, M. B. Zafar, S. Ghosh, I. Shibpur, I. K. P. Gummadi, and K. Karahalios. Quantifying search bias: Investigating sources of bias for political searches in social media. In *CSCW*, 2017.

20. B. Lepri, J. Staiano, D. Sangokoya, E. Letouzé, and N. Oliver. The tyranny of data? the bright and dark sides of data-driven decision-making for social good. In *Transparent data mining for big and small data*, pages 3–24. Springer, 2017.
21. Z. Liu and I. Weber. Is twitter a public sphere for online conflicts? a cross-ideological and cross-hierarchical look. In *SocInfo*, pages 336–347, 2014.
22. H. Lu, J. Caverlee, and W. Niu. Biaswatch: A lightweight system for discovering and tracking topic-sensitive opinion bias in social media. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 213–222. ACM, 2015.
23. A. Mowshowitz and A. Kawaguchi. Measuring search engine bias. *Information Processing & Management*, 41(5):1193–1205, 2005.
24. L. Muchnik, S. Aral, and S. J. Taylor. Social influence bias: A randomized experiment. *Science*, 341(6146):647–651, 2013.
25. B. Pan, H. Hembrooke, T. Joachims, L. Lorigo, G. Gay, and L. Granka. In google we trust: Users' decisions on rank, position, and relevance. *Journal of Computer-Mediated Communication*, 12(3):801–823, 2007.
26. E. Pariser. *The filter bubble: What the Internet is hiding from you*. Penguin UK, 2011.
27. E. Pitoura, P. Tsaparas, G. Flouris, I. Fundulaki, P. Papadakos, S. Abiteboul, and G. Weikum. On measuring bias in online information. *ACM SIGMOD Record*, 46(4):16–21, 2018.
28. C. Sandvig, K. Hamilton, K. Karahalios, and C. Langbort. Auditing algorithms: Research methods for detecting discrimination on internet platforms. *Data and discrimination: converting critical concerns into productive inquiry*, 2014.
29. J. L. Skeem and C. T. Lowenkamp. Risk, race, and recidivism: predictive bias and disparate impact. *Criminology*, 54(4):680–712, 2016.
30. L. Vaughan and M. Thelwall. Search engine coverage bias: evidence and possible causes. *Information processing & management*, 40(4):693–707, 2004.
31. I. Weber, V. R. K. Garimella, and A. Batayneh. Secular vs. islamist polarization in egypt on twitter. In *ASONAM*, pages 290–297, 2013.
32. F. M. F. Wong, C. W. Tan, S. Sen, and M. Chiang. Quantifying political leaning from tweets and retweets. *ICWSM*, 13:640–649, 2013.
33. M. B. Zafar, K. P. Gummadi, and C. Danescu-Niculescu-Mizil. Message impartiality in social media discussions. In *ICWSM*, pages 466–475, 2016.
34. M. B. Zafar, I. Valera, M. G. Rodriguez, and K. P. Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *WWW*, 2017.