CHAPTER 4

Analysis of the Impact of Algorithms on Siloing Users: Special Focus on YouTube

Joseph Vybihal and Mika Desblancs

School of Computer Science, McGill University, Montreal, Quebec, Canada

CONTENTS

DOI: 10.1201/9781003261247-6

INTRODUCTION

The impact of artificial intelligence (AI) and technology on society is undeniable. In *Weapons of Math Destruction,* Cathy O'Neil (2017) states that "If we had been clear-headed, we all would have taken a step back at this point to figure out how math had been misused … But instead … new mathematical techniques were hotter than ever … A computer program could speed through thousands of resumes or loan applications in a second or two and sort them into neat lists, with the most promising candidates on top" (Crawford, 2021). Kate Crawford in her recent book *Atlas of AI* mentions that "I've argued that there is much at stake in how we define AI, what its boundaries are, and who determines them: it shapes what can be seen and contested" (Crawford, 2021). Recently, the Senate Commerce subcommittee (2021) hearings with whistleblower, Frances Haugen, calling for transparency claiming that Facebook entices users to keep scrolling to increase the opportunity for advertisers to reach these users resulting in side effects harmful to children and women (Senate Commerce Subcommittee, 2021). The misuse of algorithms, unintentional or intentional, is a great concern given that society is very trusting in computers.

The Internet has changed the speed at which information can be shared as anyone with an Internet connection is able to disseminate their ideas to a wide audience. While this phenomenon is not new, YouTube has recently received a lot of attention. The low barrier of entry to publishing videos on the platform has caused a spike in the amount of independent political pundits; a little less than half of the most popular news channels are independent of any news organization and over a third center around a single personality. Furthermore, given that over 73% of adults and 94% of 18–24-year-olds turn to YouTube for news (Stocking et al., 2020), understanding what kind of content is being watched and information disseminated is crucial to measuring the platform's impact on today's democracies. In recent years, YouTube has been accused of being a vector for online radicalization and in 2020 had to ban open white supremacists such as Stefan Molyneux and Richard Spencer (Stocking et al., 2020). Along with its hosting of incendiary content, the YouTube recommendation system has also been accused of gradually pulling users towards more extreme content by recommending and promoting videos that become more "hard-core". A *New York Times* article claimed in 2018 that the recommendation system had a bias toward inflammatory content and users who watched mainstream news sources were quickly presented with far-right videos (Tufekci, 2018). The article argued that this tendency was not a bug but a feature; the algorithm was simply exploiting our natural attraction towards watching more inflammatory content. In June 2020, Google announced that it changed its guidelines to better manage racist content, terminated over 25,000 channels, and took steps to limit recommending inflammatory content (BBC News, 2020). YouTube's impact on political news dissemination remains important yet Google's changes have received little scrutiny.

The impact that recommendation systems have on individuals is a function of the form of interaction the individual engages with the recommendation system. This form of interaction is higher order, in that it is not controlled by the recommendation system but is centered on the user's behavior. For example, the recommendation system may show the user a set of recommendations some of which the user has already viewed. The user might not view the repeated recommendations again. It is this higher order search space the user explores: the things they want to see. That search space is constrained by what the recommendation system offers to the user, but the user further constrains the space. To study user siloes, we need to traverse this higher order search space and see how it impacts the recommendation system. In this chapter, we suggest ways to access this higher order search space (or Meta Space).

Our study focuses on the impact of YouTube's recommendation system in placing individuals into silos, unintentionally encouraging people to focus on a point-of-view while ignoring other views that may balance or critique their bias. The chapter will focus on experiments our group conducted with the YouTube platform. Our experiments explore different perusing techniques. We discuss these results in relation to the YouTube algorithm's current ability to silo users.

RECOMMENDATION SYSTEM MECHANICS

In contrast to a recommendation system, a traditional database stores information in records that are relatively independent from one another. For example, when depositing $5.00 at an ATM machine, the bank's database will record this transaction as a database record. The record will contain, as a minimum, the date, user ID, account number, and amount. If we wanted to retrieve information from the database, we would need key information to search on, like the date of the transaction. We could ask, "Display all the transaction at a particular date" or "list all the dates Alice deposited $5.00". To perform a more advanced search the record would need to have *metadata*, information (a field in the record) not directly related to the transaction that created the record. If metadata is present, then we could also perform more in-depth queries like *cross-linking* (e.g., all account holders that are teenagers) and the ability to *group information* (e.g., list all of Alice's bank accounts). These deeper searches are defined by an organization's purpose for doing such reporting. Google, for example, claims they record only statistical metadata and do not record specific statistics about individuals. They then profit from that information through targeted advertisement or the sharing of those statistics with interested third parties. Population statistics instead of tracking an individual's statistics (Jannach et al., 2021).

A recommendation system's database is more like a graph. Each database record is a node of the graph, and every arc is metadata. At least two types of metadata are stored: categorization information, which is supplied by content creators and may also be auto-generated by the host system (e.g., Genre, publisher, tags), and population statistics ("type" of people who frequently visit a node, "click" interaction tracking, group "join" tracking, search "query" writing). In addition to the content database (the YouTube videos), the system also keeps a traditional database of statistics about users (individual's personal interactions with the system) that may or may not be private information (e.g., Google claims they share population statistics but not personal statistics). The rationale is that the way an individual interacts with the system statistically is proportional to what the individual wants.

The goal of the recommendation system is to *document* and *exploit* interaction statistics for an individual. Business goals are related to *profit* and product *loyalty*, which may conflict with social concerns.

The process of documenting interaction statistics requires methods to associate a user with their interaction statistics, the common way is a password-protected user account. This restricts the number of individuals who can access the account, resulting in higher confidence that the statistics belong to the individual. An indirect way of creating an account is through the IP address of the device used to access the system. The likelihood that the IP-based account refers to a single individual is less likely, but not zero given the use of cell phones and personal laptops. At worst, the IP-based account provides statistics about a closed community (like a family or friend group). Recommendation systems can use the IP metadata to realize that two accounts, one an IP-based account and the other a username/password account, when on the same IP address are related in some way. Recommendation systems can also use the IP metadata to associate accounts resulting in cross-recommendations between presumed "family" and "friend group" members. For example, your spouse searches for diapers and it also shows up in your YouTube feed.

A user of a recommendation system interacts with the system in two modes: passive click (click a video to play) and active interaction (input search key or join a group). Each mode results in the gathering of statistics. Passive click is the default mode. In default mode the system watches the user recording specific statistics: video selected, length of video watched, pause length in front of a video, time of day user interacts with the system, view habits by time of day, associations by IP address with other users, genres frequency, etc. In active mode, the user typically enters a query into the search box for specific information. The intentional search is a strong indicator of interest in the topic. The user can also "join" a "channel" or "group" to indicate a strong preference for the material published by the provider of that channel. Commonly in a modern recommendation system, the user begins in default mode after logging in. The user is presented with a list of recommendations even before interacting with the system based on their past behavior. The way the individual interacts with this initial offering reinforces (or modifies) the statistical decisions made by the system.

A user session is measured as the period between when a user logs into the system (this could be automated by simply starting the app) and when the user logs out of the system (or automated by closing the app). During that session, the user performs a series of actions over time. These actions are related to a traversal of a graph. The root node of the graph contains the initial recommendations presented to the user when they first login. How they respond to the initial recommendations determines the path they follow through the graph. Each node of the graph is a set of recommendations presented to the user. After selecting a recommendation, the system presents additional recommendations (the next node in the graph). We define depth as being the number of nodes traversed from the root node to some stopping point. For example, the user can choose to "stop" by returning to the root node at any time by selecting a "home" button. The root node may or may not display the same information. If it displays the same information, then this is a cyclic graph. If the algorithm displays different nodes, then this is a tree (a graph without cycles). Note, it is not required that the algorithm that displays the root node recommendation is the same algorithm that presents the follow-up node recommendations.

Recommendations are based on a *policy*. A policy is a set of rules or a strategy for selecting videos from the system's catalog. These policies are *black boxes* to an outside investigator. These policies are proprietary. However, the actual policy can be teased out based on how it responds to specific inputs. These policies will manifest themselves based on observations of the form: videos grouped together, videos shown (hid) to (from) the user, node population, and the evolution of the root node over subsequent sessions.

RECOMMENDATION SYSTEM WEAKNESSES

The weaknesses in a recommendation system are directly related to the policies or technologies in question. The algorithm may not be intentionally constructed as biased. The recommendation is a function of the limits in the policy or technology. These limits can be expressed in many ways: algorithmic artifacts, fairness, business interests, etc. Let us look at two examples that highlight algorithmic artifacts.

Example 1: Artificial neural network job placement and implicit bias.

An artificial neural network (ANN) is a statistical pattern matcher. Given a set of example data, it can on its own determine the most common features belonging to the example set of data. The larger the number of examples, the greater its ability to extract even finer features that are in common to all the parameters in the example set. Assume we use an ANN to help an organization select its next employee. Assume we have decided to create an example set of all the greatest employees of this organization. Without prejudice, we give the ANN the curriculum vitae (CV) of all the greatest employees. This would contain information like their name, place of birth, school, degree, where they lived, age, gender, etc. It is hard to know what features an ANN identifies as significant since that knowledge is kept as fractional numbers (scores and weights) stored within the cells of a matrix the ANN uses to identify significant features. These fractional number patterns are determined via the interaction of the algorithm with the examples presented to the ANN. What if it so happens to be that the greatest employees for this organization happen to come from affluent neighborhoods and that past managers favored hiring men? The ANN would select strongly for men from affluent neighborhoods. This is called *implicit bias*. The organization did not intentionally want the algorithm to select affluent men. Notice that the bias arises from both the dataset used for learning (the selected CVs – we can call this the *policy*) and the way ANNs detect important features (the algorithm – we can call this the *technology*).

Example 2: Sorting genre preferences and silos.

Simple tracking algorithms use counting and thresholding when making recommendations. Let us look at this simple algorithm: count the number of times a user visits a genre and sort that tracking list from highest to lowest. Display recommendations in sorted order for all values greater than a threshold value. Values below the threshold would not be displayed. The threshold is used to reduce the volume of recommended objects and to remove uninteresting objects. The threshold could be an absolute number, or it could be a percentage. An absolute number would have the effect that recommendations would not be made for a given metadata until the user visits that metadata n times, where n is the absolute threshold value. A percentage value would display recommended objects regardless of the number of times the user visited a genre. For example, if the user is new to the system and only visited a single provider, then that visit constitutes 100% of the statistics, and assuming the threshold is 50%, it would display the object. A side-effect of this type of algorithm is observed when a user, for some reason on a particular day, becomes interested in a topic and visits that topic frequently resulting in the visit count (for that metadata) to increase causing it to be sorted to the top of the list and recommended more frequently to the user. This is useful to the user for a time, but when the user is no longer interested in this topic the system will continue to recommend the topic (since it remains sorted above the threshold). The user must actively interact with objects tagged with different metadata to force these new meta statistics to sort above their previous interest. This is an example of a silo effect. When the algorithm begins to favor recommending one topic above others based on the sorted metadata count. The unintended effect is to show only one kind of information to the user over a long period of time. This is manageable for users with diverse interaction habits since they purposefully influence the statistics, but for passive users who rely on the default behavior of the system, the objects presented to them could hover around specific content for longer periods of time, while at the same time hiding from sight other content. In the above example, the technology of counting, sorting, and thresholding has a siloing effect.

In analysis, it is important to identify not only the policy but the technology. In some cases, the policy and technology are highly integrated and can be viewed as the same thing. In the literature, policy is often viewed as integrated (Steck et al., 2021). But we argue that policy and technology should be kept separate when possible. We define *policy* as human choices in terms of input data used for learning. *Technology* is defined as the side effects of algorithms.

HOW TO ANALYZE A RECOMMENDATION SYSTEM?

This section is an introduction to software probing, where probing will be discussed in general terms. The next section, Analysis of YouTube, will describe the way we implemented these general principles.

Recommendation systems are proprietary software and are therefore black boxes. The only way to determine the nature of a black box (to determine the policy it uses) is to probe and see how it behaves with input stimuli (datasets). The output of the black box given the input stimuli form statistics that is proportional to the underlying policy and technology.

To probe a black box, we need to be aware of three important algorithmic qualities: (a) upper and lower bounds, (c) high-probability cycles, and (c) edge cases. Upper and lower bound analysis determines the behavior of the algorithm during extreme probing. This helps determine the algorithm's boundaries. In other words, beyond this point, the algorithm cannot go. High-probability cycles identify the most common way users interact with the system and how the system behaves in those conditions. Edge cases are low-probability cycles, conditions in which the algorithm is rarely in, but must still be considered for a small population of use cases.

These three important algorithmic qualities express themselves differently based on the underlying policy and technology that make up the recommendation system. These policies and technologies produce a recommendation search space the user interacts with, but there is a meta-search space that is an expression of the combined interaction of the policies, technologies, and the user. For example, the recommendation system's recommendations will express differently when using a list, a tree, or a graph (as seen in the examples discussed previously). Assuming a graph, the number of connections and the quality of the metadata on the connections determine the type of queries that can be invoked on the graph. This impacts the quality of the recommendations displayed to the user. If the recommendation has an object the user has previously seen, the user may remove that object from consideration. We referred to this previously as the user's Meta-Space, it is the search space in the user's mind. That is the real search space.

What boundaries, common cycles, and edge cases does the user's meta-search space have?

A search space is often viewed as a graph since cycles are observed in recommendation systems. In other words, the same recommendation is encountered later. However, users interact with a recommendation system on a session basis. This means the user interacts with the recommendation system over a period, and then they exit the recommendation system and visit it again at another time. This period represents an entry point into the graph (called that root) and the interactions with the system during the session (click, then new recommended objects) is the *search path*. It is acceptable to think that during a session a user will rarely interact with a recommended object more than once (meaning, if they watch a movie, they probably will not watch the movie again during the same session). If this is the case, then during the session the graph reduces to a tree. In other words, the user will ignore objects they have seen before and only search through pathways they have not previously followed in that session. It is possible that the policy would take note of cyclic interactions by the user (user watches a movie twice in the same session) to strengthen certain types of recommended objects, however, since this can be assumed to rarely happen, this behavior could be analyzed during an edge case probe, not for boundary or common-cycle analysis.

A recommendation system logs user interactions during each session. These interactions form statistics that modify the set of objects recommended during a subsequent session (or during the same session as they travel down a search path). These recommended objects can be classified and reflect the underlying policy. Studying how the root or search path changes during a session and between sessions reveal the underlying policies employed by the recommendation system. With enough data points, a distribution can be graphed giving confidence that a particular policy exists within the recommendation system.

Assuming the meta-search-space reduces to a tree, then (1) the upper and lower bound analyses of a tree are depth-first and breadth-first search, (2) the common cycles can be estimated from surveys of large populations of users, and (3) the edge cases must be handled case-by-case (one example was given previously).

Depth-first and breadth-first searches are extreme behavior interactions with a tree. In both cases, the tree is searched exhaustively. Depth-first search interacts with the tree by selecting the first recommendation presented, regardless of what it is, until it exhausts that path. Then, it goes up the branch and selects the second recommendation, and so on. In breadth-first, every recommendation is watched in the ply before going down to the next ply. Breadth-first search focuses on the set of recommended objects presented to the user, and how watching all the objects impacts subsequent recommendations. In some recommendation systems, depth-first and breadth-first interactions are unrealistic. However, this type of probing reveals features of the underlying policy. In other words, general conclusions about features of the policy of the form "at most it can do this" or "in the worst case, it can do that" are revealed.

Surveying a large population of users will tease out common practices. A common practice is defined by a set of interactions users commonly perform during a session. For example, login, see initial offering, pick the best object, watch it, pick from the next set of recommendations, watch that, go back to root, see the offerings, watch another one, then logout. Writing probing software that follows these common practices, together with the boundary cases, helps to reduce the range of divergent paths. In other words, the change in recommendations observed from the common practices probing is compared to the boundary probing. If a common practice sequence of interactions has a depth or breadth element (truncated to a finite length), then extreme behavior can be extrapolated using the boundary analysis. It can be said "most users experience this x but some users will experience that y", where x is the common cycle and y is the boundary cycle. In other words, if a user does x for a while and then happens to do y, a statistical prediction can be made.

ANALYSIS OF YOUTUBE

Previous Work

A *New York Times* article cited by Tufekci (2018) claimed that the YouTube recommendation system had a bias toward inflammatory content and users who watched mainstream news sources were quickly presented with far-right videos. In 2019, Ledwich and Zaitsev (2020) published a paper that contradicted these claims. They found that

YouTube's recommendation algorithm actively discourages viewers from visiting radicalizing or extremist content, instead favoring mainstream media and cable news. The authors built a dataset of over 816 political and influential YouTube channels. Each of these channels was manually annotated using a list of 18 tags. Using individual videos from tagged content creators, the authors collected information about the recommended videos with a scraping script. The authors' algorithm viewed and collected information about these videos using an anonymous account that had not "watched" any previous videos. The YouTube recommendation system had no "user history" on which to base its recommendations. The authors investigated the recommendations of an individual video without traveling down the recommendation space. Thus, their study focuses on the recommendation system's political behavior for isolated videos without history. Given that recommendation systems base their recommendations on historical user interaction statistics, we do not believe their study accurately portrays the YouTube recommendation system.

Ledwich and Zaitsev (2020) list of 816 political channels was later extended to include annotations for over 6,500 content creators (Clark & Zaitsev, 2020). The authors used a channel discovery and classification method which generates political affiliation tags using user channel subscriptions. The model can predict political affiliation tags with higher degree of agreement with human annotators, agreeing between 84% and 97% of the time. The model achieved precision and recall values of 89.1% and 77.9% for left-leaning channels, and 86.3% and 92.3% for right-wing channels. We have chosen to re-use an updated list of annotated political channels from January 1, 2022, generated by Clark and Zaitsev's method with 11,645 content creators (Clark & Zaitsev, 2020).

Our Work

We classify videos using tags associated with YouTube channels from the 11,645 content-creators annotated dataset described previously. For our scraping, we created different authorized YouTube accounts to take into account user history. We have chosen to assume that the user will not view a video multiple times during a YouTube session. However, they may repeat view a video at another session. By session, we mean logging back into a YouTube account after having deleted its entire account-linked user history and data. We delete an account's entire account-linked user history and data before starting another session so that a user's history built over the course of an expansion does not influence the recommended videos of another session. To properly mimic user engagement in a video, the algorithm "watches" videos for three minutes or until the maximum length of the video is reached, whichever is shorter. By "watching" the video, we hope to better mimic the human user interaction on YouTube.

Given the above, the upper and lower bound analysis is equivalent to a depth-first and breadth-first search of the recommendation tree. The high-probability cycle for the most common user behavior was determined using a survey of 187 YouTube users. We leave edge cases for future work.

We present only the upper bound, lower bound, and common-cycle experiments. Figure 4.1 is the expansion algorithm.

```
Begin:

   if depth or breadth:
        root_rec = YouTube_Search(random(Curated_list));
   else:
        root_rec = YouTube_Search(common_use_behavior());

   recs = Top(root, n); // top n recommended videos

   count = 0;
   Expansion(recs, count);

End.

Expansion(recs, count):

   if count == m: stop // m is based on probing depth

   if depth:
        Expansion(Top(YouTube_Search(depth_select_first(recs))), n), count+1);
   if breadth:
        Expansion(Top(YouTube_Search(breath_select_next(recs))), n), count+1);
   if common-cycle:
        Expansion(Top(YouTube_Search(random_or_politicalBias_select_one(recs))), n), count+1);

End.
```

FIGURE 4.1 Expansion algorithm.

Next we detail the different expansion types. Each expansion description follows algorithm 1. Figure 4.2 depicts breadth-first boundary experiment. Figure 4.3 depicts depth-first boundary experiment. Figure 4.4 depicts important results from our survey. Table 4.1 depicts the curated left-/right-leaning seed videos.

BREADTH-FIRST EXPANSION

In breadth-first expansion (see Figure 4.2), the algorithm begins with the user selecting a YouTube video of interest using active search. This search places the user at what we call the *root recommendations* of the tree. It is the only part of the algorithm that uses a directed search. It is an automated search using a predetermined (curated) set of providers of interest (see Table 4.1) divided into left- or right-wing content-creator sets. Depending on the search bias, a video is randomly selected from that list of videos. The resultant root node is a set of video recommendations dependent on this search. In breadth-first search, we will visit every child video in that resultant list of recommended videos (the ply) before continuing to the next ply. To reduce complexity of our analysis, the algorithm visits only the first three recommended videos irrespective of what the videos contain. Each of those three videos becomes the root of another recursive expansion. In this way, the tree expands ply by ply. The algorithm continues until a ply depth of 4. For each ply, we compute the number of videos whose metadata corresponds to the initial search metadata, in this case, its left- or right-wing rating over the total number of recommendations in that ply. We call this the *silo percentage*, and how it changes over time, we call this the *silo percentage evolution*.

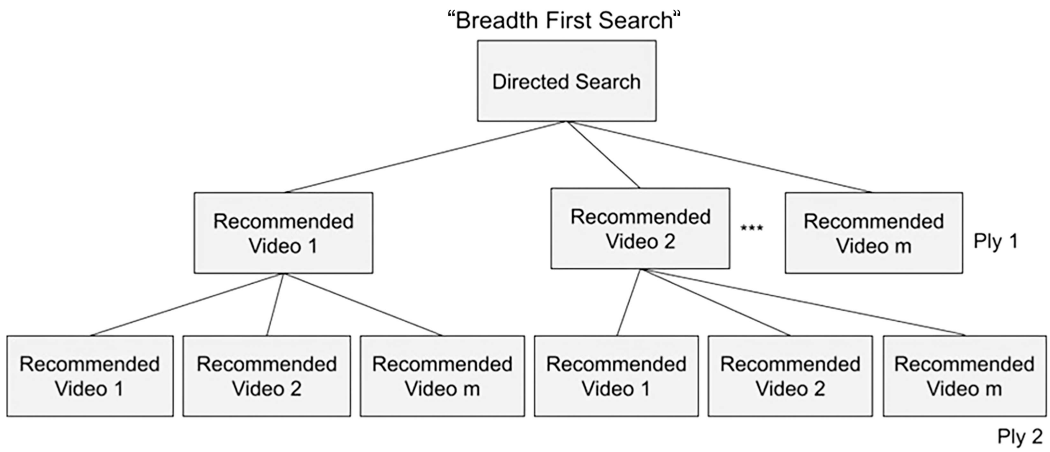

FIGURE 4.2 Breadth-First expansion of YouTube recommendations.

DEPTH-FIRST EXPANSION

In depth-first expansion (see Figure 4.3), the algorithm begins by performing the same initial active mode search as in breadth-first search from the curated list. Then, the algorithm takes the first video recommended, regardless of its content, and watches it. The algorithm is then presented with a new list of recommended videos and watches the very first one, regardless of its content. It repeats this watching of the first recommended video repeatedly up to a ply depth of 6. Then, it returns to the root and watches the next

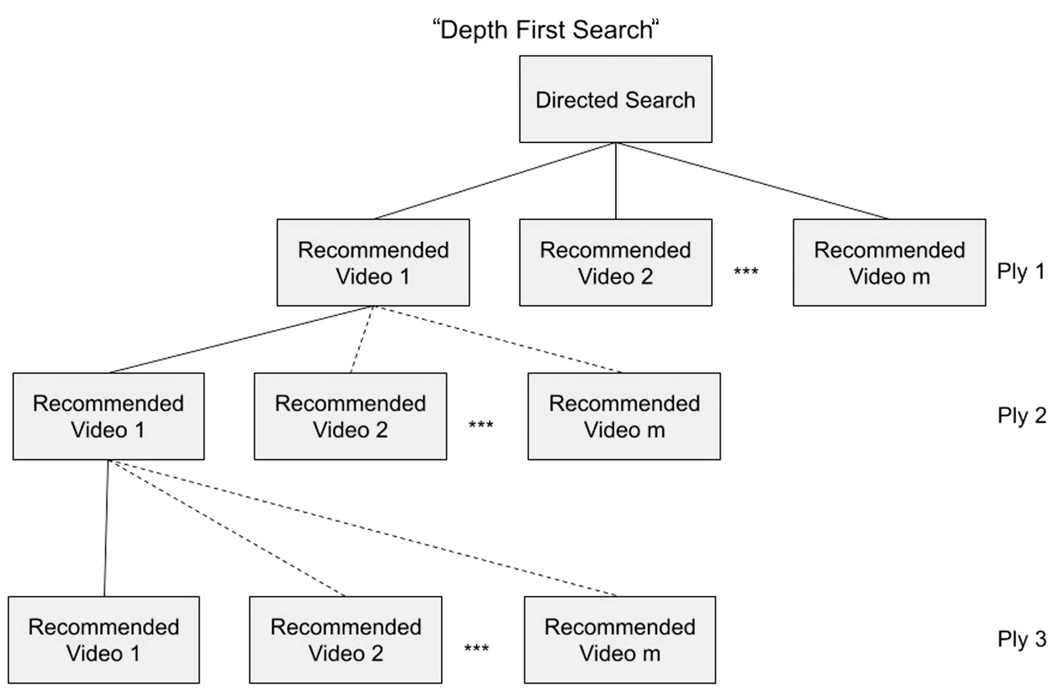

FIGURE 4.3 Depth-First expansion of YouTube recommendations.

recommendation presented, starting a new depth-first traversal of the recommendation tree. Importantly, the original root video recommendations are saved and remain unchanged throughout the entire session. We do this five times. We calculate the silo percentage by following each depth-first search path. We call this a *dive*. We compare the number of silo videos by the number of videos watched from root to leaf for each “dive”. We then compute the growth/decrease of the silo bias between each “dive”, comparing the first dive with the second, the second with the third, and so on. We call this the *silo evolution value*.

COMMON-CYCLE EXPANSION

In common-cycle expansion, the algorithm uses the technique most reported in our survey. We surveyed 187 YouTube users. The summary of the survey results and some important conclusions can be found in Figure 4.4.

Given the results from the survey, the most common procedure YouTube users follow is directed search by interest (A and D with E) with randomness (F) that sometimes challenges their beliefs (G). They do a depth-directed search up to a ply of 2 or 3 before returning to the homepage (H). They repeat this process a few times. We will call this *the common-cycle procedure*.

The common-cycle experiment will use the following variation of the common-cycle procedure. The algorithm will start by determining the user’s intent by using the following rule:

- 80% of the time it will choose to start with an initial left-/right-wing seed from the curated list of videos (Table 4.1) to generate the root recommendations of interest, based on its initial bias.

```
(A) 82% Looks for things they like. (W 77%)
(B) 50% Go back to home page. (W 38%)
(C) 92% Logged into YouTube. (W 88%)
(D) 50% Search for a video. (W 62%)
(E) 56% Same topic scan. (W 54%)
(F) 65% Random - catch attention. (W 63%)
(G) 85% Watch videos challenging beliefs (W 22%)
(H) 81% Depth of 2 or 3 (W 80%)
(I) 95%: in their 20s, student, 90% North American.

Conclusions:
- Three data points greatest difference between men/women (B), (D), (G)
- Most are logged into YouTube (C)
- Look for things they like (A) (some randomly (F))
- search depth 2 or 3.
- W = women
```

FIGURE 4.4 Survey Results for Common Cycle.

TABLE 4.1 Example of Curated Left- and Right-Wing Videos

Title	Content creator	Political classification
A Short History of Slavery	PragerU	Right
Yes, Censorship is Bad	Sargon of Akkad	Right
Leftists MELT DOWN After Elon Musk Condemns Vaccine Mandates	Ben Shapiro	Right
Ben Shapiro Gets SCHOOLED by Neil Degrasse Tyson on Trans Issues	Vaush	Left
The Easy Answer of YouTube Conservatism	Three Arrows	Left

- 10% of the time it will choose a video from the first 25 videos automatically presented to it when it first logged in (called the *homepage videos*).
- 10% of the time it will scan the homepage videos for an opposing point of view, choosing a random homepage video if no partisan video is found.

The algorithm then performs a depth-first search of the recommendation space guided by the user's intent for a ply depth of 2. This will be repeated six times.

For each expansion type, we also compute a *silo evolution value of the user's homepage*. The homepage is the initial set of recommended videos presented to the user when they log into YouTube (at the start of a session). To track the silo evolution of the homepage, the algorithm scrapes a session's YouTube homepage's top 10 videos. Scrapes will occur before the start of each new *ply* in the breadth-first expansion and before each *dive* in the depth and common-cycle expansion. As more probes are performed, the homepage will change over time. We are interested in whether siloing occurs right from the start of their YouTube session.

The analysis of YouTube will be based on three experiments using the three previous expansion types. Experiment 1 uses bread-first search, experiment 2 uses depth-first search, and experiment 3 uses the implementation of the common-cycle procedure described above. For experiments 1 and 2, our analysis will compare *dive statistics* with *dive evolution statistics* between each dive. We will also compute *homepage state statistics* and *homepage state evolution statistics*. We will then generate a table and make observations. For more detailed explanations of the algorithms please refer to the GitHub page (Desblancs, 2022).

RESULTS AND ANALYSIS

Breadth-First Expansion

The results from the depth-first search experiment are summarized in Table 4.2, Figures 4.5 and 4.6. The experiment visited 7,426 videos, which represent trees of width 3 and depth of 4, where each "dive" started at the root, using the root's 5 videos as the beginning.

TABLE 4.2 Tentative Breadth-First Search Experiment Results

Ply	TTL L	TTL R	TTL O
0	1	0	2
1	2	0	7
2	4	0	20
3	5	0	64
Ply	**TTL L**	**TTL R**	**TTL O**
0	2	0	1
1	3	0	6
2	4	1	22
3	8	3	64

Table 4.2 shows the number of videos seen per ply, categorized by left-leaning (TTL L), right-leaning (TTL R), and other videos (TTL O). These initial results show that videos of type Other are favored. Two breath dives are shown for a left-leaning initial directed search. As the dive progresses, the TTL O columns increase faster than the TTL R and TTL L columns.

Silo Percentage and Evolution

In Figure 4.5, we see how breadth-first evolves over depth. In the picture on the left, we have a left-leaning initial directed search. We see that the recommended videos at ply zero show more left-leaning videos, as expected. But as the algorithm progresses breadthwise down the tree, the "Other" videos dominate. A similar story exists for the right-leaning videos. An interesting artifact of our data is that the right-leaning videos are always not the most prevalent, even at ply zero. Also notice that at ply 4 the curve changes direction. We believe this is an artifact of our algorithm. In future experiments, we will dive deeper to see if this pattern persists.

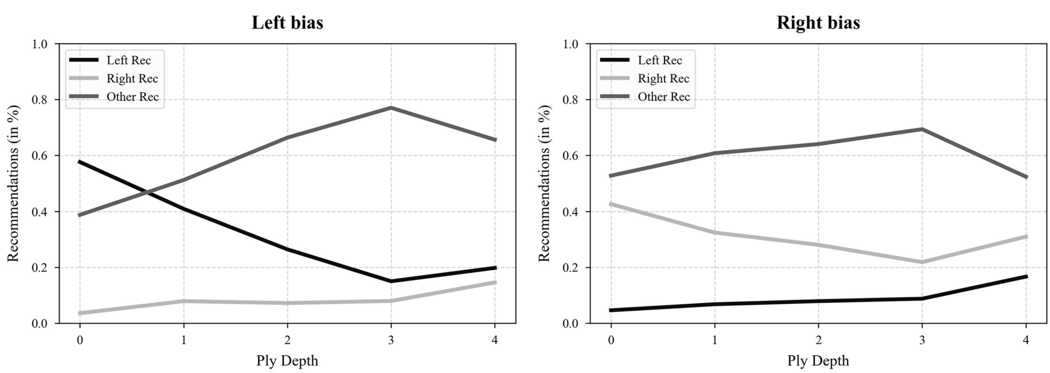

FIGURE 4.5 Average video recommendation types per ply in a Breadth-First Left- and Right-wing tree (in %).

Homepage Ply Percentages

The evolution of the homepage is an important property. It is the initial offering to the user when they login and can influence their outlook. Figure 4.6 shows our experimental results. You can see in breadth-first the homepage is always dominated by "Other" videos. If the user is left or right leaning, the remaining videos reflect their leaning, and the opponent videos (to a much lesser degree) still appear in the feed.

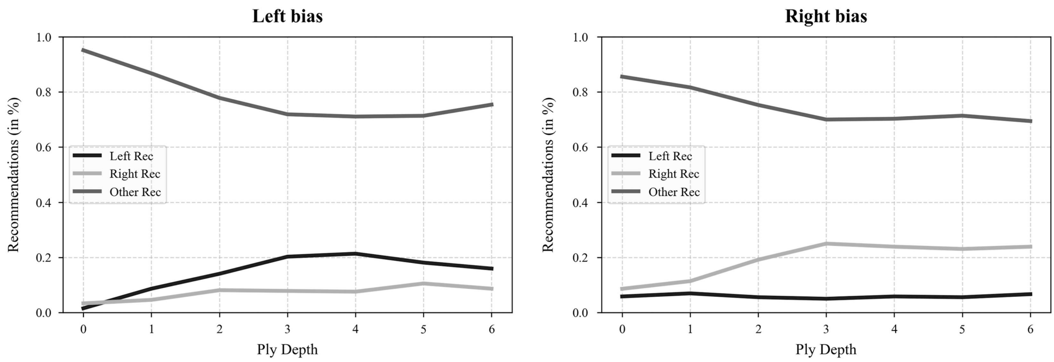

FIGURE 4.6 Average homepage recommendation types per ply after each dive in a Breadth-First Left- and Right-wing tree (in %).

Depth-First Expansion

The results from the Depth-First Search Experiment are summarized in Figures 4.7 and 4.8. The experiment visited 2,448 videos, which represents trees of width 5 and depth of 6, where each "dive" started at the root, using the root's 5 videos as the beginning. This was not an exhaustive depth-first search of every ply, but only the root.

Dive Statistics and Evolution

Figure 4.7 shows the dive statistics and evolution with the percentage of right-, left-wing, and neutral (Other) videos in the top 5 suggested videos at the different video positions, where the position is the order in which videos were visited. The algorithm goes to a ply depth of 6, which is indicated by the vertical dashed lines. After each dashed line the algorithm returns to the root and starts its next dive. Interesting artifacts of the data to notice: (a) at the end of the first dive the bias is nullified by YouTube resulting in a steep rise in neutral videos and a sharp decline in the left/right videos. (b) We see again that right-leaning videos perform more poorly than left-leaning videos. Further study is needed; for example, are left-leaning videos more entertainment based, while right-leaning are more news-based?

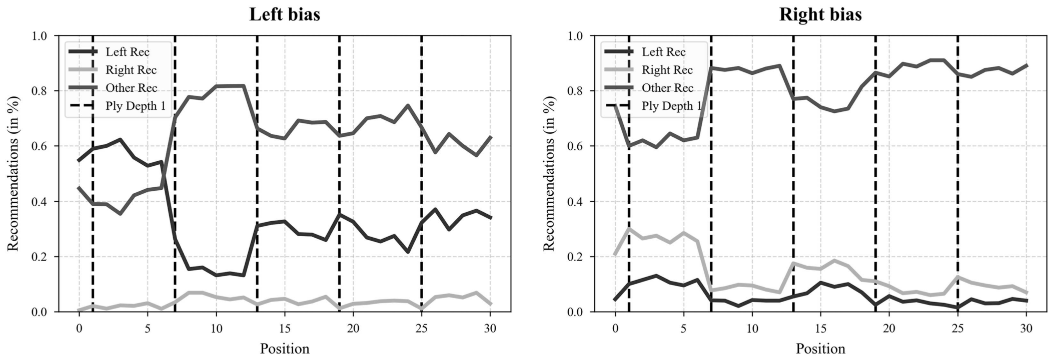

FIGURE 4.7 Average video recommendation types per position in a Depth-First Right- and Left-Wing Tree (in %).

Homepage State and Evolution

Figure 4.8 graphs out the homepage right-, left-wing, and other recommendation percentages on the homepage after dive *n* along with its evolution after the different dives, where 0 is a snapshot when we first login and 6 is a snapshot after the sixth (last) dive. The homepage evolution compared to the breadth-first graphs.

In summary for depth and breadth search, we see that the bias affected greatly the outcome. If the user selected "other" they were presented with left and other videos, but mostly other. If they selected "left" they were presented with left and other videos, but mostly left.

The dive recommendation bias frequency evolution graphs also show this siloing effect is especially strong for trees with left-wing biases. The frequency with which left-wing videos are recommended is greater than the reverse situation. Furthermore, regardless of the tree bias, videos presenting opposite political points of view are rarely recommended.

The homepage evolution shows slight levels of siloing for both biases. However, it is not as strong as the siloing which occurs in the recommendation space. Interestingly, for both political biases, political bias video frequency stabilizes to its highest values directly after the first dive, indicating that while siloing is slight, it happens during the first dive.

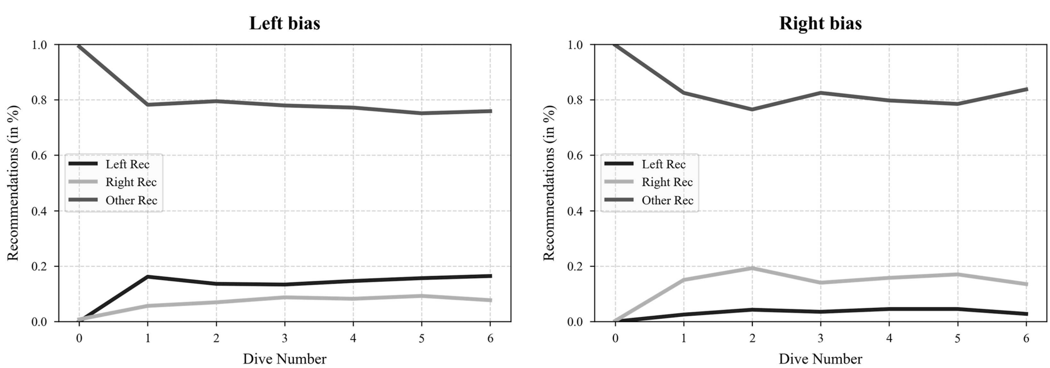

FIGURE 4.8 Average homepage recommendation types after each dive in a Depth-First Left- and Right-wing tree (in %).

Common-Cycle Expansion

The results from the Directed Search Experiment are summarized in Figures 4.9 and 4.10. The experiment visited 1,895 videos. Which represented directed dives of depth 3, returning to the root, and then performing another directed dive, following the *variation of the common-cycle procedure*.

Dive Statistics and Evolution

Read Figure 4.9 in the same way as Figure 4.8 with the notable exception that black dotted vertical lines now indicate points in the algorithm where it has gone back to the homepage instead of to the directed search root recommendations. Notice the artifacts of this figure: (a) siloing occurs to a greater extent for both biases in the common expansion. (b) The neutral (Other) videos compete for dominance. (c) In the right bias, the neutral (Other) videos dominate. An interesting question to explore is whether the competing neutral (Other) video is an attempt to counter siloing through distraction by YouTube.

Homepage State and Evolution

Figure 4.10 is read like Figure 4.8. The artifacts to note in this figure are (a) the neutral (Other) videos dominate the graph but lose ground as bias persistence continues. (b) The homepage reflects the user's bias to a strong degree.

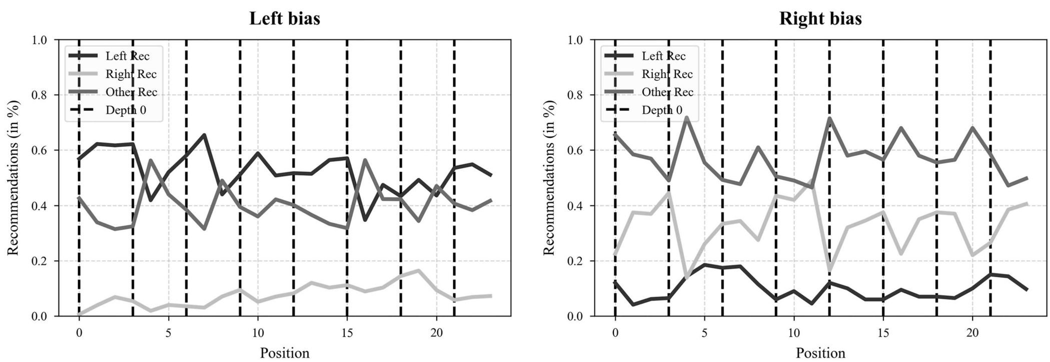

FIGURE 4.9 Average video recommendation types per position in a common-cycle expansion Left- and Right-wing tree (in %).

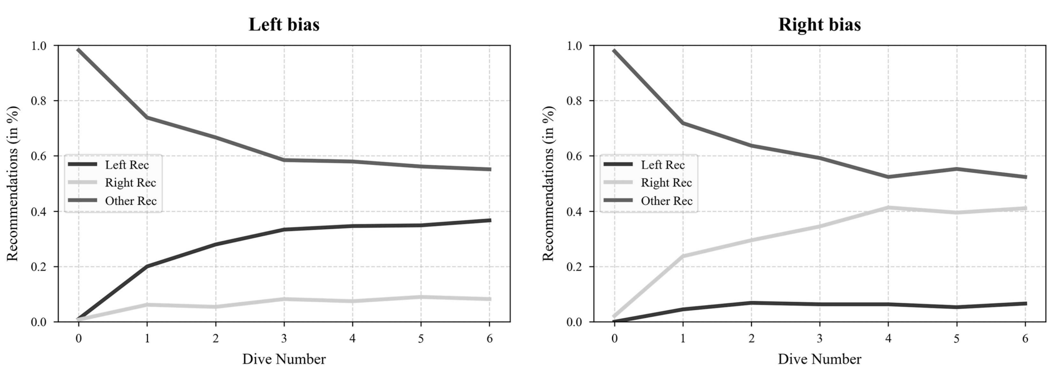

FIGURE 4.10 Average homepage recommendation types per dive in a survey-inspired Right-wing tree (in %).

In summary, following the common-cycle procedure, siloing is present for both political biases with the opposite bias rarely being recommended. However, the recommendation system seems more sensitive to recommending left-wing content than right-wing content. On scrapes with a left bias, the recommendation system suggested an average of around 50% of left-wing recommendations while only suggesting around 30% of right-wing content. Unlike the depth- and breadth-first expansions, however, the frequency of biased recommended videos does not follow periods of constant growth, decreases nor stabilization. The YouTube algorithm also seemed to be sensitive to neutral (Other) choices resulting in a rapid rise in neutral recommendations. However, this rise would decrease when not exploited.

CONCLUSION

The purpose of a good recommendation system is to give the user what they want to see. Based on our experiments, YouTube does this. If the user is left or right leaning and selects 80% of the time their bias of choice, the YouTube platform will present the user with the left- or right-leaning videos, rarely presenting opposing videos. The silo effect is present in our experiments. The user must intentionally select opposing videos to see those videos in their feed. However, the recommender is sensitive to selection changes and displays opposing videos quickly.

The evolution of the user's homepage over time will reflect their bias to a greater degree with little to oppose. Even though the homepage is dominated by neutral videos, these videos do not help expose the user to opposing views. In many cases, this effect is not important, as when purchasing diapers. But it is important when viewing videos on the "benefits" of not eating, or in politics, or conspiracy videos. This homepage bias dominance can lead to confirmation bias and siloed thought.

Interesting questions to study further: What is the reason for the dominance of neutral (Other) videos, is this an artifact of the data, or is this an intentional distraction strategy by YouTube? Is the effect that left-leaning videos are presented more often than right-leaning videos real? To what extent does homepage bias dominance lead to confirmation bias and siloed thought?

REFERENCES

BBC News (2020). YouTube bans prominent white supremacist channels. BBC Business. *BBC News*, https://www.bbc.com/news/business-53230986

Clark, S., & Zaitsev, A. (2020). Understanding YouTube communities via subscription-based channel embeddings. *arXiv* [Preprint] (2020). arXiv:2010.09892.

Crawford, K. (2021). *The atlas of AI*. Yale University Press, p. 217.

Desblancs, M. (2022). Analysis of the impact of algorithms in siloing users: Special focus on YouTube (Version 1.0.0) https://github.com/mika-jpd/YouTube_Radicalization_Recommendations [Computer software]

Jannach, D., Pu, P., Ricci, F., & Zanker, M. (2021). Recommender systems: Past, present, future. *AI Magazine*, *42*, 3–6. 10.1609/aaai.12012

Ledwich, M., & Zaitsev, A. (2020). Algorithmic extremism: Examining YouTube's rabbit hole of radicalization. *First Monday*, *25*(3), 10.5210/fm.v25i3.10419

O'Neil, C. (2017). *Weapons of math destruction*. New York: Broadway Book, p. 13.

Senate Commerce Subcommittee (2021). Facebook whistleblower Frances Haugen testifies before senate commerce committee. C-SPAN. https://www.youtube.com/watch?v=GOnpVQnv5Cw

Steck, H., Baltrunas, L., Elahi, E., Liang, D., Raimond, Y., & Basilico, J. (2021). Deep learning for recommender systems: A netfil case study. *AI Magazine*, *42*, 7–18. 10.1609/aaai.12013

Stocking, G., Van Kessel, P., Barthel, M., Eva Matsa, K., & Khuzam, M. (2020). *A closer look at the channels producing news on YouTube – And the videos themselves*. Pew Research Center. https://www.pewresearch.org/journalism/2020/09/28/a-closer-look-at-the-channels-producing-news-on-youtube-and-the-videos-themselves/

Tufekci, Z. (2018). Youtube, the great radicalizer. *The New York Times*, www.nytimes.com/2018/03/10/opinion/sunday/youtube-politics-radical.html