Community-based Recommendations on Twitter: Avoiding The Filter Bubble

Quentin Grossetti, Cédric Du Mouza, Nicolas Travers

Community-based Recommendations on Twitter: Avoiding The Filter Bubble

Quentin Grossetti[1], Cédric du Mouza[1], and Nicolas Travers[1,2]

[1] CEDRIC Lab, CNAM Paris, France
firstname.lastname@cnam.fr
[2] Léonard de Vinci Pôle Universitaire, Research Center, Paris La Défense, France
nicolas.travers@devinci.fr

Abstract. Due to their success, social network platforms are considered today as a major communication mean. In order to increase user engagement, they rely on recommender systems to personalize individual experience by filtering messages according to user interest and/or neighborhood. However some recent results exhibit that this personalization of content might increase the *echo chamber* effect and create *filter bubbles*. These filter bubbles restrain the diversity of opinions regarding the recommended content. In this paper, we first realize a thorough study of communities on a large Twitter dataset to quantify how recommender systems affect users' behavior and create filter bubbles. Then we propose the *Community Aware Model* (CAM) to counter the impact of different recommender systems on information consumption. Our results show that *filter bubbles* concern up to 10% of users and our model based on similarities between communities enhance recommender systems.

Keywords: Twitter · Communities · Filter Bubble · Recommender System

1 Introduction

Social networking has become a major way to share and discover information on the Internet. Users generally connect since they know each other in real life or share a common interest. Since received content from the flow is related to people with whom they are connected to, users may consequently find their opinions constantly echoed back which creates an *echo chamber* [8], that may skew their point of view. Moreover, it has been theorized that this phenomenon is reinforced by recommender systems [18] massively used to enhance users' engagement by personalizing individual experience. Consequently, they tend to focus on highly relevant messages mainly based on users' neighborhood and/or interests. Recently critics argued that such systems are impoverishing user opportunities to be displayed to diversified information, so called the "*filter bubble*".

The link between Recommender Systems (RS) and filter bubbles is not clearly characterized in literature and we particularly target this issue in this paper. So we first extract communities with a traditional community detection algorithm

in a real `Twitter` dataset. This algorithm which relies mainly on topological properties (so not topic-centric) will group people who are close and strongly connected in the network, because they know each other, are geographically close and/or share common interests. Then we perform analysis to detect *filter bubbles* by measuring how often messages leave their community of origin and we try to understand how RS focus on content originated from a reduced number of communities. To achieve this we propose to characterize users by a community profile based on their interactions with communities through messages provenance. Then we show that recommendations provided by RS may differ from users' community profile and generate a *filter bubble* for some users. Therefore, we advocate the fact that *filter bubbles* can be characterized by topology-based communities, further works on opinion mining are out of the scope of this article.

Our second objective is to tackle this *filter bubble* effect for these users through a re-ranking of their recommendations to be more respectful of their community profile. Our proposal can be deployed on top of any RS without modifying its implementation. We show that our solution significantly improve the quality of recommendations by matching more closely users' community profile and by reducing the *filter bubble* effect at a limited computation cost.

In a nutshell, the main contributions of the paper are:

1. A community analysis to study how information is propagated through communities to characterize *echo chambers*,
2. A measure and an analysis of the *filter bubble* effect from respectively a community and a user's point of view,
3. A novel re-ranking strategy that relies on users' community profile and the community network to reduce the *filter bubble* effect.

2 Related Work

Most popular social network platforms such as *Facebook*, *Baidu*, *Twitter* or *Instagram* gather millions of users. To help them find relevant content, these platforms largely rely on RS. Recently, some works have shown that these platforms have to face two simultaneous effects that affect user points of view. First, the "*echo chamber*" phenomenon means that some users tend to consume only information from the same ideological alignment. This leads to biased opinions. The second effect, due to the personalization of content from recommender systems, traps users in a "*filter bubble*" as described by Eli Pariser [18].

Studies on "*echo chambers*" were initially conducted in social sciences to investigate how people tend to bind with similar people, creating communities and having difficulties to access opposite view points. It has been partially described and analyzed in [7, 16]. They conclude that people tend to choose news articles from sources aligned with their political opinions. [3] also shows that people tend to connect to each other on social platforms following an homophily behavior, so to bind with similar people. A large study [5] focusing on filter bubbles and echo chambers states that this phenomenon is not limited to the digital era since social media users only mimic traditional offline reading habits. In short, *echo*

# nodes	2,2M	# edges	325.5M	# tweets	3,002M
avg. path length	3.7	avg. out-deg.	57.8	max out-deg.	349K
diameter	15	avg. in-deg.	69.4	max in-deg.	185K

Table 1: Main features of the `Twitter` dataset

chambers is a natural phenomenon which has existed for a long time before Facebook and the echo chamber on social networks is due to this real-life behavior, homophily, which is only replicated on social platforms.

While it is commonly admitted that *echo chambers* exist, there is no indisputable evidence of the existence of "*Filter Bubbles*". Indeed, it is unclear whether recommender system algorithms amplify the echo chamber phenomenon or not. Some studies have tried to quantify this phenomenon. [5] studied web-browsing habits of 50,000 US-located people. To our knowledge, this is the largest study on *filter bubbles* and *echo chambers* phenomena. They observed a counterintuitive behavior: users with the highest "ideological segregation" rely more on recommender systems to find new information but also are more exposed to opposite perspectives. Thus, people using recommendation systems (RS) are the ones seeing more different points of view. Another study [17] related to movie recommendations made by the *GroupLens* team has a similar conclusion. This work on *filter bubbles* asserts that RS actually lower the chances of being trapped into a *filter bubble*. *Facebook* also conducted a similar study [1] on their algorithm which is used to filter the feed of users. They conclude that it only decreases by 1% the chances of seeing posts corresponding to opposing views.

Models aiming at bursting an *echo chamber* to create more "peaceful" debates on a specific topic, such as gun control or Obamacare, have been presented in [6]. In this work the authors propose to add edges between people having opposite views in order to reduce controversy in the network. [12] proposes a model where the user gives a specific point of view in order to see how recommendation change based on this new perspective. A similar idea is developed in [8]. However, these solutions are difficult to deploy in practice because they rely on the will of the users to change their viewpoints. Our approach largely differs from existing work since it is, to the best of our knowledge, the first approach to use communities as a tool to observe *echo chambers* and *filter bubbles* effects, and to propose a re-ranking strategy of the recommendation to reduce the filter bubble phenomenon.

3 Community Analysis

In order to estimate the importance of the *filter bubbles* and *echo chambers*' phenomena induced by recommender systems' usage, we first extract communities from the social graph with the traditional `Louvain` method. Then we try to have a better understanding of the communities these algorithms produce and we study the behavior of users regarding the community they belong to.

3.1 Twitter dataset

We present here the main characteristics of our `Twitter` dataset introduced in *[anonymous]*. It is based on a connected component extracted from the graph made provided by Kwak et al. [14] which has been updated since 2017 thanks to the Twitter API [3]. We collected the incoming edges (followers), out-coming edges (followees) and all the tweets published by the associated accounts. Observe that due to the API limit we only retrieved the last 3,200 messages for each node. Table 1 summarizes the main features of the dataset.

We can notice that, with more than 2 million users and 3 billion messages, we have a mean number of 1,375 published tweets per user. We detect that around 12% of these tweets, so on average 150 tweets per user, correspond to a retweet action. Our analysis also exhibits that 92% of the tweets are never retweeted. It means that recommendations mainly focus on a small part of the messages. As shown in [11] users tend to have more similar profiles with users within a 2-hop distance in the graph (called homophily [13]). This homophily has an impact on information propagation: people close to each other in the network tend to have a higher number of retweets in common.

3.2 Communities' detection

To characterize the *echo chamber* phenomenon and the information propagation between users, we identify and study communities in our dataset. Scalable community detection algorithms are proposed in literature, like `Infomap`, `Louvain` and `Label Propagation`. Note that these methods only use the network topology and not topics, user profiles or exchanged content to extract communities. Moreover they associate users to a single community. The `Louvain` algorithm we have adopted is tailored for directed graphs [15, 4]. It consequently suits to the `Twitter` network. It maximizes the modularity of clusters inside the graph that will produce denser components (*i.e.,* maximizing the number of connection triplets). However note that we also performed similar work for `Infomap` and got very similar results. To explain the filter bubble effect, we try to understand the rationale for the formation of a community. We first label the communities according to their main feature(s). Remember that a user belongs to a single "community" according to the considered community-detection algorithms, and that these communities are built by considering only the topology of the underlying social graph. We focus on the 105 more representative communities, *i.e.*, those with more than 100 users identified by the `Louvain` method. To determine the labels, we adopt the following three-step process:

1) Most followed users inside each community are selected (most central users),
2) We find most frequent terms occurring in the tweets of these users and we check important features from their profiles like *age*, *location*, *language*, *etc.*,
3) Based on these two kinds of information we provide the most representative tag for each entity.

[3] https://dev.twitter.com/rest/public

Some improvements may be considered like performing named-entity extraction rather than only relying on term frequencies, for instance. However it turned out that our basic strategy provides good labeling since users who have strong common interests, such as "Sports" for instance, are highly connected and form a community we effectively tagged as "Sports".

3.3 The community network

We exploit here the detected communities to enlighten the echo chamber effect. The objective is to quantify how information spreads outside the community to which it has been attached to. We first link a tweet to a community, then we find out how many communities it reaches. This quantification could be seen as a *propagation measure* inside the social network. This allows us to study the presence of echo chambers at both users and communities level.

Community Membership of a Message. To track messages "activity" we need to identify the way to attach messages to a community. Two options can mainly achieve this: a message belongs to the community from which it occurs first or to the community in which it obtained most likes/retweet.

It appears that 90% of retweeted messages obtain a high popularity in the community from which it comes from. The remaining 10% belong to small communities and naturally become famous when they reach larger communities. In the following we decide to identify the message community membership based on the community where it was written initially in order to emphasize the influence of small communities on bigger ones.

Correlation Between Popularity and Spread. Now we have communities and messages, we can measure the popularity of messages and how they propagate throughout the *community network*.

Figure 1 shows the distribution of retweeted messages with respect to the number of reached communities. We can see that 80% of retweeted messages reach at most 2 communities, and among them, half remains internal to the community they belong to. This distribution is characterized by a *Power Law*: $Cx^{-\alpha}$ (with $C = 200$, $\alpha = 2.2$ and $x_{min} > 1$ for probabilities). As expected C is really high stating that the probability that a tweet remains in a community is high. According to α, this classical value (typically between 2 and 3) indicates that communities have far connections between each other. This experiment confirms the fact that most of the messages are rarely retweeted while few very popular messages reach high numbers of communities. It underlines the existence of an echo-chamber effect inside communities.

According to this analysis, we conclude that most of the tweets hardly ever leave their community, especially if they are not popular.

4 Filtering Bubble

The objective is to analyze how recommender systems create or reinforce the *echo chamber* phenomenon at community and user levels. We study the filter bubble

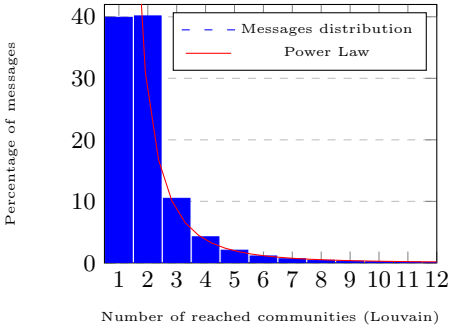

Fig. 1: Msg. wrt. # reached communities

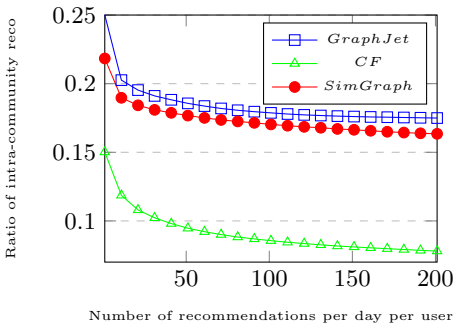

Fig. 2: Ratio of intra-community reco

effect with three different recommendation systems: *GraphJet* [19] proposed by *Twitter*, Collaborative Filtering [2] (called *CF*) and *SimGraph* [11]. To achieve this, we consider recommendations produced for samples of 25 users randomly extracted from each community obtained by `Louvain`.

4.1 Community-level approach

A global approach to quantify the *filter bubble* effect is to compute the proportion of intra-community recommendations. When the proportion of users' recommendations belongs to its own community is too high (*intra-community recommendations*, opposite of the diversity), it implies that a *filter bubble* effect could lead to the reinforcement (or apparition) of an *echo chamber* effect.

In Figure 2 we plot the ratio of intra-community recommendations regarding the number of recommendations proposed per day (for each user). We find out that *GraphJet* tends to propose less "diverse" recommendations than *CF* with on average 23% of intra-community recommendations. This could be explained by the random walk-based algorithm behind *GraphJet* that would give more opportunities to recommend messages in the neighborhood, which corroborates conclusions of Figure 1. At the opposite *CF* computes similarities between users from the whole graph independently from the topology and tends to provide more diversified recommendations than other solutions, in terms of community provenance. *SimGraph* results are between *CF* and *GraphJet* since it mixes both topology and similarity (*i.e.,* homophily).

We also notice that independently of the number of recommendations proposed, the diversity is constant after 20 recommendations. Consequently, in the following we fix the recommendation number to 20 per day. As expected, in Figure 2 *filter bubbles* aren't visible due to average values over every user.

To study the filter bubble at community scale, we display in Figure 3 the ratio of intra-community recommendations per community along with their size for the CF recommendation algorithm. Community labels come from Section 3.2. Due to space limitations, we do not display Figures for the other algorithms but they behave similarly. We observe that for all recommendation algorithms, there is a logarithmic correlation between community size and intra-cluster recommen-

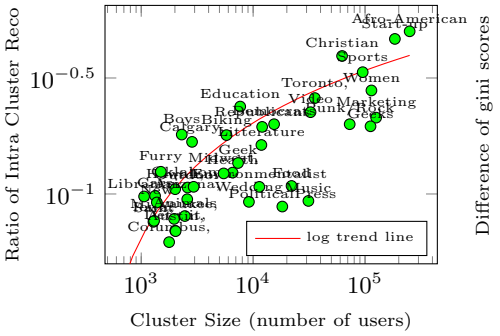

Fig. 3: CF intra-Recommendations

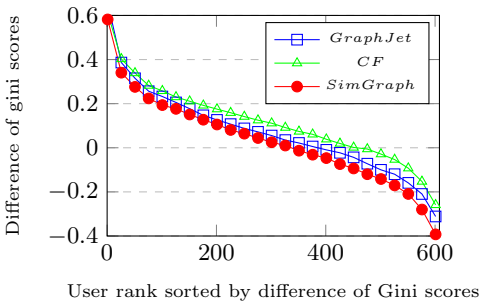

Fig. 4: Diff. of Gini coefficients between likes and recos

dations. The rationale is that the bigger a community is, the higher the chances are for its users to receive a recommendation from this community. However this experiment reveals that a global approach isn't sufficient to exhibit a particular community being concerned by a filter bubble.

4.2 Local approach

Since we cannot detect filter bubbles with a global approach at community-level, we attempt to see whether this phenomenon can be observed at user-level. Therefore, we analyze communities' diversity for which recommended tweets are issued from. For this, we apply for each user the *Gini* coefficient [9] on the aggregate number of received recommendations per community. The *Gini* coefficient measures the ratio of inequalities within a set of values, *i.e.,* its diversity.

Users with high Gini scores seem to be trapped into a filter bubble. It is due to the RS which provides recommendations issued from few different communities. However after analyzing their profiles, we observe that these users have in fact a very specific usage of the platform (*e.g.,* football player's account only interact with sports messages). Therefore the RS by recommending only sports messages just follow the usage of the user maintaining the *echo-chamber* effect.

Consequently we believe that we must consider users' profile in the platform to determine if they are in a bubble or not. We thus consider the difference between user's interactions and RS recommendations. We propose to show this effect by computing the difference between the Gini coefficient of users' profile (list of effectively "*liked*" communities) and the one from the recommender system (list of "*recommended*" communities). Results are plotted in Figure 4. High values mean that the recommendations are too diversified compared to the real user behavior while low values lead to a *bubble effect* with fewer communities concerned by recommendations compared to the real user behavior.

We see that 30% of the users are faced with less diversified recommendations than their own profile. This effect is mainly due to a frequent behavior of the user who "likes" many messages from a particular community and less frequently from "random" ones. However, recommender systems focus mostly on this main community and provide recommendations mainly issued from this community.

5 CAM - a Community-Aware Model

Thanks to this preliminary but essential study, we are now able to detect a filter-bubble effect on users' community profile with topology-based communities. We propose in the following our Community-Aware Model whose objective is to reduce the filter-bubble impact. It can be deployed on top of a RS and it enhances it with a new scoring function which permits re-ranking the recommendations. Observe that our approach is consequently independent of the choice of the RS and may be consequently deployed in any existing social network platform.

5.1 Community profiles

So consider a user u and a social network where n communities were detected by a community detection algorithm. Let $\overrightarrow{Pu}$ be the user's u community profile represented as a normalized vector: $\overrightarrow{Pu} = (pc_1, pc_2, \ldots, pc_n)$ where pc_i denotes the rate of messages from the community c_i among all the messages he liked.

Suppose that a recommender system RS produces a list of recommendations $LReco_u$ for the user u from which only the top-k items are extracted and presented to u. The main idea is to re-rank $LReco_u$ by considering, for each message, its community of origin. The end goal consists in finding a top-k which corresponds more precisely to the user community profile $\overrightarrow{Pu}$.

Note that naive models which attempt to pick up the required number of messages from $LReco_u$ in each community of $\overrightarrow{Pu}$ wouldn't be successful. Indeed, due to too low recommendation scores or to a period where the corresponding community is less active, some communities from a profile $\overrightarrow{Pu}$ are not present (or insufficiently present) in $LReco_u$. Besides, with such naive approaches, a message with a high recommendation score which is not issued from a community appearing in $\overrightarrow{Pu}$ will also be discarded, even if the community is topologically and/or thematically closed to, which contributes to the filter bubble effect.

Since our community analysis reveals that some communities are thematically very close to, we propose that our re-ranking model takes into account this similarity and consequently modifies the scores produced by RS even for messages from communities which are not in $\overrightarrow{Pu}$.

Our model relies on the impact of items on communities called $\overrightarrow{V_U}$ and the user's profile $\overrightarrow{Pu}$. It tries to minimize the distance between $\overrightarrow{V_U}$ and $\overrightarrow{Pu}$.

5.2 Community similarity score

We first need to determine a measure of similarity between communities which takes into account 1) topology, 2) semantic information and 3) flows of information between these communities. We propose the following similarity measure to estimate how similar two communities can be.

Definition 1. *(Community Similarity Score) The asymmetric similarity measure between a community c_i and c_j is estimated as follows:*

$$sim(c_i, c_j) = \alpha\ Links(c_i \to c_j) + \beta\ Sem(c_i, c_j) + \gamma\ Flow(c_i \to c_j) \quad (1)$$

where Links is the ratio of the number of links from c_i which are directed to c_j among its outgoing links, Sem represents the similarity (see Section 6.1) between the main topics of c_i and c_j, and Flow corresponds to the link importance which relies on the proportion of circulating tweets (retweets) from c_i to c_j. α, β and γ are constants which can be tuned according to the behavior of the underlying RS (see Section 6.2) in order to target relevance and/or filter bubbles.

Based on this similarity measure we can build the Community Similarities Matrix ($CSM = (sim_{ij})_{1 \leq i,j \leq n}$). Observe that this matrix is not symmetric since we consider links' direction and information propagation (flow).

5.3 Community-aware recommendations

We consider that each item I is associated to a community score vector $\overrightarrow{I}$ which captures how this item is thematically and topologically close to each community. To compute the vector $\overrightarrow{I}$ of an item I we rely on the community-similarity matrix CSM. So $\overrightarrow{I}$ corresponds to the community similarities from column c_i of the CSM matrix to which community I is associated to.

Our model intends to propose a set of recommendations U, selected from the recommendation list $LReco_u$ produced by RS, with a community score vector $\overrightarrow{V_U}$ which matches as much as possible the user profile $\overrightarrow{Pu}$. The community score vector $\overrightarrow{V_U}$ of a set of recommendations U is the aggregation of different normalized community score vectors of each item in U: $\overrightarrow{V_U} = \sum_{I \in U} \overrightarrow{I} / || \sum_{I \in U} \overrightarrow{I} ||$.

Finding the set of recommended items U whose community profile $\overrightarrow{V_U}$ matches as much as possible the profile $\overrightarrow{Pu}$ can be modeled as a distance minimization problem between $\overrightarrow{V_U}$ and $\overrightarrow{Pu}$:

$$\begin{cases} U = argmax_{Lreco_u} |\overrightarrow{Pu} - \overrightarrow{V_U}| \\ |U| = k \end{cases} \tag{2}$$

However, determining the new recommendations based only on the distance with the user profile, regardless of the importance of the recommended content, may lead to recommend content of lower interest for the user. So another objective for our approach consists in the following maximization problem:

$$\begin{cases} U = argmax_{Lreco_u} \sum_{I \in U} recom(u, I) \\ |U| = k \end{cases} \tag{3}$$

where $recom(u, I)$ denotes the score of item I for user u provided by the RS.

Consequently the objective of our re-ranking algorithm is expressed as a multi-objective optimization problem determined by both Equations 2 and 3.

5.4 Avoiding the filter bubble

A traditional strategy to determine a solution to a multi-objective optimization problem is scalarization where no solution satisfies both objectives. Scalarizing is

Ranking	Origin community	Score
1	A	0.8
2	A	0.7
3	A	0.5
4	B	0.4
5	C	0.1

Table 2: Recommendation scores for Joe

3-item set and their score	
{1,2,3} 0.81	{1,2,4} 0.40
{1,2,5} 0.57	{1,3,4} 0.41
{1,3,5} 0.54	{1,4,5} 0.48
{2,3,4} 0.42	{2,3,5} 0.47
{2,4,5} 0.47	{3,4,5} 0.43

Table 3: Scores for all 3-item combinations

an a priori method, which transforms the multi-objective optimization problem into a single-objective optimization problem.

To achieve this transformation, we propose to integrate the recommendation score when estimating the community score vector $\overrightarrow{I}$. Since our objective is to get a high global recommendation score, we attempt to discard first from our recommendation set, items with a low recommendation score.

Thus, we adopt for our community score vector $\overrightarrow{I}$ this new definition:

$$\overrightarrow{I} = \frac{1}{recom(u, I)} \times \overrightarrow{CSM(c_i)} \tag{4}$$

With this new definition, an item with a low recommendation score will significantly increase the different components of its community score vector. This item will have a high impact on $\overrightarrow{V_U}$ and increase the profile distance. Thus, this item is more likely to be replaced by another one in the final item set.

Example 1. Consider a user Joe to whom a recommender system proposes a list of recommendations $Reco_{Joe}$. Assume for this example that we limit the recommendations to the top-3 scores, so Joe receives the three recommendations originated from the community A. We suppose that there are only 3 communities and that there exists no similarity between them. Therefore CSM is the identity matrix. We assume that Joe interacts equally with these three communities; therefore his profile is: $\overrightarrow{P_{Joe}} = (0.33, 0.33, 0.33)$

To re-rank the items by considering the user profile and the relevance of the messages, we compute the distance from Equation 2 with the community score vector computed with Equation 4.

We display in Table 3 the $distance = |\overrightarrow{P_{Joe}} - \overrightarrow{V_{Joe}}|$ for the different 3-item combinations. For our example, we see that the score for the best combination is 0.40 and corresponds to {1, 2, 4}. We see in Table 2 that these items have a high recommendation score, and this set better matches the user profile.

To determine the top-k recommendation set, we theoretically need to compute all the combinations of k items from U extracted from $LReco_u$ which consists of N items has a complexity of $\binom{N}{k}$. We escape the exponential complexity by adopting an interchange algorithm. So we initialize the recommendation set with the k top-rated items. Then we check for each of the $N-k$ remaining items if we can reduce the distance with the user profile by replacing one recommendation by this item. This algorithm has a N^2 complexity.

6 Experiments

We first detail the experimental protocol we adopted to measure the benefits of our re-ranking model *CAM* for both the relevance of recommendations and the number of users suffering from the filter bubble effect. We also study the impact of the different parameters in our model, *i.e.,* semantic similarities, the flow and the topological similarity, on the overall results. Our experiments reveal that our model can be tailored for different RS to provide significant gains. For all our experiments, we use the `Twitter` dataset described in Section 3.

6.1 Settings

To measure the filter bubble effect, we use the *Gini* coefficient (see Section 4.2) for a sample of users from our dataset. More precisely we measure the difference between the user's Gini coefficient computed for his community profile and the one computed for the community distribution of the recommendations. We consider that a user is affected by a filter bubble if this difference is lower than a given threshold of -0.2 (bottom right of Figure 4). This -0.2 corresponds to the inflection point observed in Figure 4 which characterizes 10% of the users.

We select the largest communities, with at least 1,000 users, from the USA found by the Louvain clustering method. They represent 38 communities. For each of these communities, we randomly extract 16 users, leading to 608 selected users. This choice of 16 users corresponds to the maximum number of users for the smallest community that retweeted at least twice, therefore users that can be targeted by a RS to give sufficient messages to re-rank. We chose to balance all the communities by an equal number of users.

Then we select messages' retweets which were retweeted at least twice. This constitutes a set of $132,389,409$ sharing actions, timely ordered. We split the set in two: the first 90% of actions (the oldest retweets) compose the training set and the last 10% the test set. While the former set is used to train the three methods, the latter one allows checking the recommendations with real retweets. Note that the test set captures 66 days of retweets from users in our dataset.

Then for each recommender system we compare *CF*, *GraphJet* and *SimGraph*, we observe the recommendations computed during the test set with and without applying our CAM algorithm. To estimate the CAM re-ranking score we determine its three components *Links*, *Semantic* and *Flow* as follows:

- The number of directed edges between communities is used to compute the *Links* weight, capturing the topological proximity between communities.
- In order to compute the *Semantic* similarity we rely on `Word2Vec` trained on Google News data [10]. We extract most frequent words from communities and combined them to create a vector thanks to `Word2Vec`. This method allows us to compute semantic distance between communities.
- We measure the *Flow* weight from the network of communities based on the proportion of tweets that circulates between corresponding communities through retweets (flow proximity).

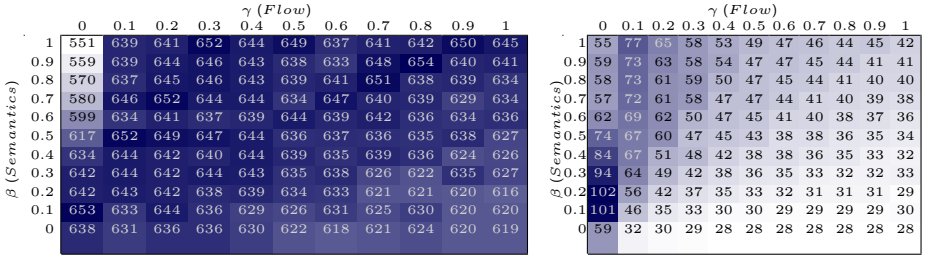

Fig. 5: *hits* and users suffering from filter bubble for *GraphJet* w.r.t. γ and β

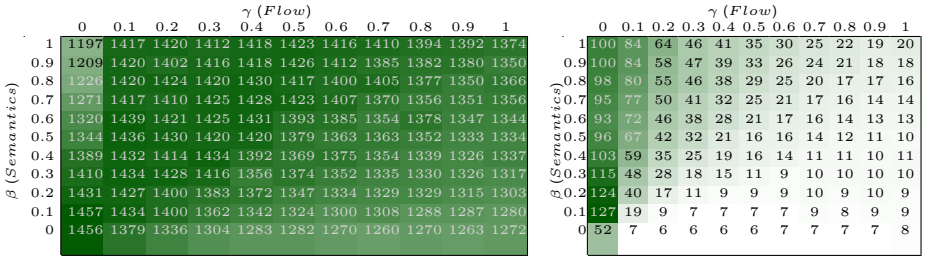

Fig. 6: *hits* and users suffering from filter bubble for CF model w.r.t. γ and β

We consider that a message is a *hit* if it is recommended to a user based on the training set, and we detect that it leads indeed to an interaction (retweet/like) in the test set. This prediction task can be seen as a relevance measure.

6.2 Studying weights' impact

The re-ranking algorithm relies on the similarities between communities (Equation 1). Since similarity scores depend on α, β and γ, we perform experiments to study the impact of each weight on the re-ranking quality. Each weight is bounded between 0 and 1 and we adopt a 0.1 padding for our experiments providing 11 different values for every weight which leads to $11^3 = 1,331$ different weights configurations. For space reason, we displayed only results with α set to 0.5 which showed a lower impact than β and γ that are considered here.

In Figure 5 we plot the results for *GraphJet* from `Twitter`. The left table shows the number of accurate predictions (*hits*) made by the system w.r.t. β and γ weights. The right table represents the number of users among our 608 selected users who suffer from a bubble effect (those with a Gini difference lower than the -0.2 threshold). Results for respectively the collaborative system (CF) and for *SimGraph* are presented respectively in Figure 6 and 7. We first observe a high variability of results depending on our parameters. The number of accurate recommendations - *hits* - ranges from 492 to 653 for `GraphJet` for instance, so a 32% difference. We notice the same order of variability for both CF and *SimGraph*. The variability is even more important for the number of users facing a filter bubble. For instance, we see that 6 users at a minimum are concerned by a filter bubble for the CF model while in the worst configuration there are 128 users concerned. So we see that the given weights to the different dimensions (semantics and messages flow) have an important impact on the quality of

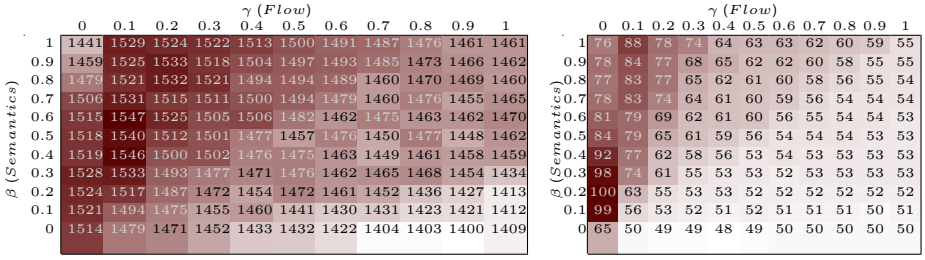

Fig. 7: *hits* and users suffering from filter bubble for *SimGraph* w.r.t. γ and β

	Initial		**Best configuration**	
	Hits	Filter Bubble	Hits	Filter Bubble
GraphJet	552	5.4%	630 (+14%)	4.6% (−15%)
CF	1,400	2.7%	1,348 (−3%)	0.9% (−64%)
SimGraph	1,468	10.0%	1,491 (+2%)	7.0% (−23%)

Table 4: CAM approach benefits

the recommendations and they allow us to efficiently boost the relevance of recommendations or to decrease the number of users suffering from filter bubbles. Obviously, the two scores are linked: the more we narrow users' interests the more chances we have to make accurate recommendations but the more users are proposed the same kinds of recommendations.

Overall we observe that the three RS tested show similar key trends when changing α, β and γ. Reducing weight β (the semantics similarity between communities) allows tweets whose topic is different from the users niche interest to be more likely recommended and therefore lowers the number of users suffering from the bubble effect. On the other hand, lowering this weight also induces that some relevant items will not be recommended to the user. The γ weight (the flow between communities) has an opposite effect. Higher γ values tend to recommend items from different communities with which the user is used to interacting but also to decrease at the same time the number of hits. Finally, we observe that α has a similar impact β on the results but with a lower amplitude.

Our experiments show that there does not exist a configuration where both the relevance of recommendations and the number of users in a filter bubble are optimized. So the different weights in our CAM model may be tuned according to the objectives of the recommender system. Thanks to our experiments, we can also determine for each recommender system the configurations which minimize the number of users suffering from the filter bubble effect (see below).

6.3 Gains Achieved with the CAM Approach

Our next experiment aims at illustrating the gain that we achieve by deploying the *CAM* model on top of existing recommender systems. So for each recommender system, we select the best weight setting to minimize the number of users affected by filter bubbles according to our observations in Figure 5, 6 and 7.

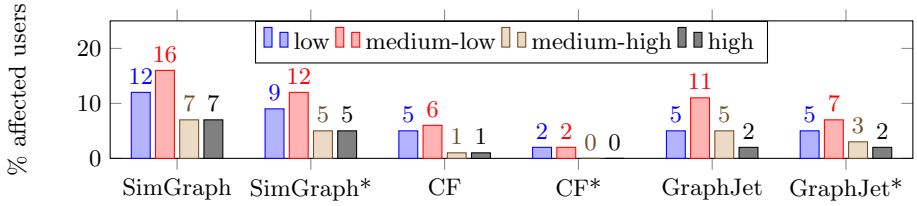

Fig. 8: Filter bubble users w.r.t. their activity without or with* CAM

We present in Table 4 the percentage of users facing a filter bubble with and without re-ranking the recommendations for the different recommender systems, along with the total number of *hits* we get. We observe that our re-ranking model successfully decreases the number of users affected by the filter bubble by 15% for $GraphJet$, 64% for CF and 23% for $SimGraph$. Additionally, by matching more to the user's profile we also improve the relevance of recommendations and boost the number of accurate predictions especially for $Graphjet$. Only for CF, removing 64% of filter bubble effects on affected users slightly decreases their relevance: -3% of hits.

Our model seems to remove users more successfully from a filter bubble for the CF model. This could be explained by choice possibilities of the re-ranking step. Sometimes, $GraphJet$ and $SimGraph$ hardly produce recommendations far in the social network, narrowing the possibilities of re-ranking while CF could compute a large list of recommendations for all users [11].

6.4 Users Activity and Filter Bubbles

In Figure 8, we investigate the link between users' activity, *i.e.*, number of messages they interacted with, and the filter bubble. Users are assigned to a category (*i.e.,* low, medium-low, medium-high, high) according to the number of interactions they made on the platform. For each of these categories, we plot the percentage of users affected by a filter bubble. We observe that most users concerned by this phenomenon have a low activity. Users with low or medium-low activities correspond to more than 70% of affected users.

Due to fewer interactions, recommender systems focus on the known interest of these users. Therefore, this limits the scope of possible recommendations. Consequently, using CAM allows highlighting items that were poorly considered by the underlying recommender system, and impact those users much more.

7 Conclusion

In this paper we have presented a thorough study on information flow on *Twitter* and we showed that the filter bubble phenomenon only concerns a minority of users. We proposed the CAM approach which relies on similarities between communities to re-rank lists of recommendations in order to weaken the filter

bubble effect for these users. Moreover our approach is able to boost the accuracy of *GraphJet* recommendations by increasing the prediction by 14%.

For future works, we want to investigate better partitioning strategies for *Twitter*. Even if we showcased filter bubbles with topology-based communities, our approach can reasonably be enhanced by finding location and/or opinion-based community detection algorithms to better detect filter bubble effects.

We also wish to study the evolution of the links between communities, since retweets evolve over time, it will have an impact on the similarity measure.

References

1. Bakshy, E., Messing, S., Adamic, L.A.: Exposure to ideologically diverse news and opinion on facebook. Science **348**(6239), 1130–1132 (2015)
2. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: UAI'18. pp. 43–52 (1998)
3. Colleoni, E., Rozza, A., Arvidsson, A.: Echo chamber or public sphere? predicting political orientation and measuring political homophily in twitter using big data. Journal of Communication **64**(2), 317 – 332 (2014)
4. Dugué, N., Labatut, V., Perez, A.: A community role approach to assess social capitalists visibility in the Twitter network. SNAM **5**(1), 26 (2015)
5. Flaxman, S., Goel, S., Rao, J.M.: Filter bubbles, echo chambers, and online news consumption. Public Opinion Quarterly **80**(S1), 298–320 (2016)
6. Garimella, K., De Francisci Morales, G., Gionis, A., Mathioudakis, M.: Reducing controversy by connecting opposing views. In: WSDM. pp. 81–90 (2017)
7. Garrett, R.K.: Echo chambers online?: Politically motivated selective exposure among internet news users. JCC **14**(2), 265–285 (2009)
8. Gillani, N., Yuan, A., Saveski, M., Vosoughi, S., Roy, D.: Me, my echo chamber, and I: introspection on social media polarization. CoRR **abs/1803.01731** (2018)
9. Gini, C.: Variabilità e mutabilità. Libreria Eredi Virgilio Veschi (1912)
10. Google: Word2vec. `https://code.google.com/archive/p/word2vec/` (2013)
11. Grossetti, Q., Constantin, C., du Mouza, C., Travers, N.: An homophily-based approach for fast post recommendation in microblogging systems. In: Proc. Intl. Conf. on Extending Database Technology (EDBT). pp. 1–12. Austria (2018)
12. Kamishima, T., Akaho, S., Asoh, H., Sakuma, J.: Enhancement of the neutrality in recommendation. In: Decisions@ RecSys. pp. 8–14 (2012)
13. hyung Kang, J., Lerman, K.: Using Lists to Measure Homophily on Twitter. In: AAAI. pp. 26–32 (2012)
14. Kwak, H., Lee, C., Park, H., Moon, S.B.: What is twitter, a social network or a news media? In: WWW. pp. 591–600 (2010)
15. Leicht, E.A., Newman, M.E.J.: Community structure in directed networks. Phys. Rev. Lett. **100**, 118–122 (Mar 2008)
16. Munson, S.A., Resnick, P.: Presenting diverse political opinions: how and how much. In: Human Factors in Computing Systems. pp. 1457–1466. ACM (2010)
17. Nguyen, T.T., Hui, P., Harper, F.M., Terveen, L.G., Konstan, J.A.: Exploring the filter bubble: the effect of using recommender systems on content diversity. In: WWW. pp. 677–686 (2014)
18. Pariser, E.: Beware online "filter bubbles" (2011), `https://www.ted.com/talks/eli_pariser_beware_online_filter_bubbles`,
19. Sharma, A., Jiang, J., Bommannavar, P., Larson, B., Lin, J.: GraphJet: Real-time Content Recommendations at Twitter. PVLDB **9**(13), 1281–1292 (2016)