



L'aspect topologique des recommandations

Erwan Le Merrer, Gilles Trédan

► To cite this version:

Erwan Le Merrer, Gilles Trédan. L'aspect topologique des recommandations. ALGOTEL 2017 - 19èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications, May 2017, Quiberon, France. 4p. hal-01517738

HAL Id: hal-01517738

<https://hal.science/hal-01517738>

Submitted on 3 May 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

L'aspect topologique des recommandations

Erwan Le Merrer¹, Gilles Trédan²

¹Technicolor, France ²LAAS/CNRS, France

La recommandation joue un rôle central dans le e-commerce et dans l'industrie du divertissement. L'intérêt croissant pour la transparence algorithmique nous motive dans cet article à observer les résultats de recommandations sous la forme d'un graphe capturant les navigations proposées dans l'espace des items. Nous argumentons qu'une telle approche en « boîte noire » est utile dans le cas d'une exploration limitée à un utilisateur : nous illustrons une topologie tirée de recommandations à un utilisateur de Youtube, fournissons ses caractéristiques clés, et montrons qu'elle renseigne sur la connaissance de cet utilisateur par le système.

Nous montrons ensuite que l'analyse de cette topologie d'aborder la question du biais potentiel dans ces recommandations. Nous postulons que les systèmes de recommandation produisent naturellement des topologies cohérentes, et qu'une manipulation de ces résultats par l'ajout de liens biaisés a toutes les chances de violer cette cohérence (à la manière des liens longs d'un modèle « petit monde »). Ce postulat est supporté par l'analyse d'un modèle génératif basé sur les k NN et par l'exploitation du crawl Youtube, en ciblant la prédiction de liens « Recommandé pour vous » (i.e., biaisés ou non par Youtube).[†]

Mots-clés : Graphes de terrain, recommandation, modèle Watts-Strogatz, biais, transparence algorithmique.

The output of recommender systems are benchmarked by researchers and practitioners based on their precision and recall performances on test datasets [4]. Yet, while those metrics have proven useful for assessing the performances of recommenders, we find that the *graph* data-structure has not been applied for studying and learning about the recommendations made to users (i.e., the recommenders' outputs). We argue that graph theory and the wide spectrum of graph algorithms available for data mining complex networks can be as well leveraged for complementing studies about recommender results.

Some major service providers, such as Youtube, comment on the high level implementation of their recommender, without specifying details that would allow a transparent use by the public [1]. Recently, there is an increased demand for accountability of the services provided by those platforms, that can be viewed as black-boxes operating remotely, and that a user interacts with by providing her profile or by calling API operations [3]. We observe in next section the results of recommendations to a user as a graph.

1 User recommendations as a topology

On the web-page of an enjoyed item, recommendations take the simple form of a set of displayed items, for the user to interact with. We propose to go beyond the collection of this flat item-set, by diving into each proposed item, recursively, up to a limited depth h (for obvious practical reasons). On the canonical example of crawling from an item web-page (e.g., video), where k other related items are recommended, and where each item is only recommended once in total, we would obtain a balanced tree of $n_{tree}^h(k) = \frac{k^{h+1}-1}{k-1}$ nodes.

We use the example of video recommendation in this paper. In the mainstream Youtube platform, we start crawling recommended video web-pages from a given video (a popular one from Lady Gaga), with $h = 4$ and noticing that 19 videos are recommended at each page ($k = 19$). There are two modes for viewing videos in such a system ; case \mathbb{N} : a user is new to the platform, and case \mathbb{C} : she is a returning user (thus has a history, identified by Youtube with a cookie passed by our web-crawler). Drastic differences occur when building the graphs for both scenario (the two crawls are separated by only few seconds) ; \mathbb{N} : $n_{\mathbb{N}}^4(19) = 14,121$ nodes and has 36,435 edges. In case \mathbb{C} , $n_{\mathbb{C}}^4(19) = 8,786$ nodes and has 24,731 edges. Both graphs are displayed on Figure 1. First observation is that as $n_{tree}^4(19) = 137,561$, there is a high

[†]. This paper is a part of the technical report available under reference [2].

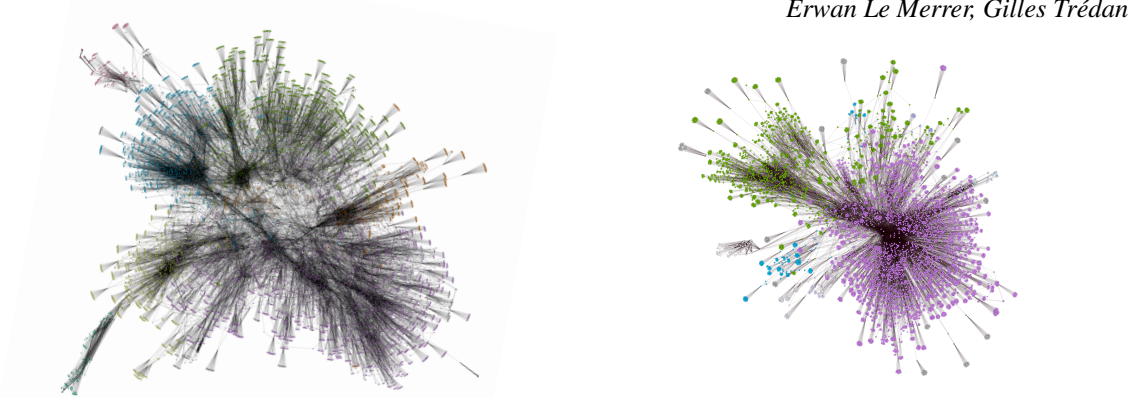


FIGURE 1: 4-hops graphs of recommendations from a Youtube video, new (left) vs returning user (right).

redundancy of recommendations, for both \mathcal{X} and \mathcal{C} . Very interestingly, crawl \mathcal{C} contains around 38% less nodes than \mathcal{X} . This has to be interpreted by the recommender system “knowing” the user, and then trying at each video access step to insist in the recommendation of what it thinks is best suited for that user to enjoy. In case \mathcal{X} of a new user, the recommender presents videos related to the start one as well, but also includes in its recommendations videos from different categories (sport, news, ...), probably in the hope to gain knowledge faster about the user by varying its propositions. Those phenomena are confirmed by analyzing graph structures : a search for main clustered components (through the modularity algorithm with $p = 5.0$) indicates 7 components of size at least 1% of the graph nodes for the graph in experiment \mathcal{X} , versus only 3 for case \mathcal{C} (i.e., more precise recommendations lead to fewer clusters of interest for the returning user). Node colors corresponds to components they belong to, on Figure 1. Another key difference is the degree distribution, with 3 nodes having more than 100 in-neighbors for \mathcal{X} , versus 10 in \mathcal{C} : well ranked videos are consistently recommended to the returning user.

An immediate conclusion drawn from the structural analysis of both \mathcal{X} and \mathcal{C} graphs, is that their topological comparison informs about the degree of knowledge of the system about the observing user, even under limited exploration scope. Having discussed the possibility for a user to crawl a recommender system and analyze its outputs under the form of a graph, we now propose an application using this representation. [‡]

2 Exploitation of graphs of recommendations for bias detection

A recently discussed topic is the influence of online medias, and their capacity to shape user opinion based on item recommendations. We now define the setup considered in our work.

Recommender Model We consider that recommenders, given an item $i \in I$ (along with some other type of information like the user profile) return a score $s_R^i : I \mapsto [0, 1]$ typically capturing items similar to i [4]. The output of such a recommender system R is then exploited by a service that selects the subset of best matching items that will get recommended when a user u consults i (typically a simple sort operation). Let $R_i(u) \subset I$ be this set of recommended items to u at a web-page : $\forall j \in R_i(u), j' \in I \setminus R_i(u), s_R^i(u, j) \geq s_R^i(u, j')$. By selecting a recommended item $j \in R_i(u)$, u enjoys j , but as well gets recommended items similar to j , namely $R_j(u)$. In this context, an instance of a graph discussed in Section 1 is a directed graph $G_R^h(u) = (I, E_R)$ in which $(i, j) \in E_R \Leftrightarrow j \in R_i(u)$. In a system that biases recommendations, the user is proposed certain items (for economical, or legal reasons for instance) : this translates in *biased edges* toward items from set I .

Bias in a graph of recommendations The service officially exploits the recommender R (such as the one advertised in [1]), that can be accessed as a graph of recommendations $G_R^h(u)$. A service may aim at adding recommendation bias to G_R^h for orienting u 's navigation by including some biased edges. Let E_B this set of biased edges. Resulting observable graph is $G^h(u) = (I, E_R \cup E_B)$. The central use case we are interested in relies on the vision of recommendations under the form of a topology.

We now ask the following question : *having access to $G^h(u)$, can user u decide whether a given edge of that graph is biased or not ?* Hereafter, we propose a model for addressing this question, that is based on the small-world parallel with recommendation graphs.

[‡]. We propose models for graphs of recommendations in [2], notably one explaining observation bias that a user might encounter.

L'aspect topologique des recommandations

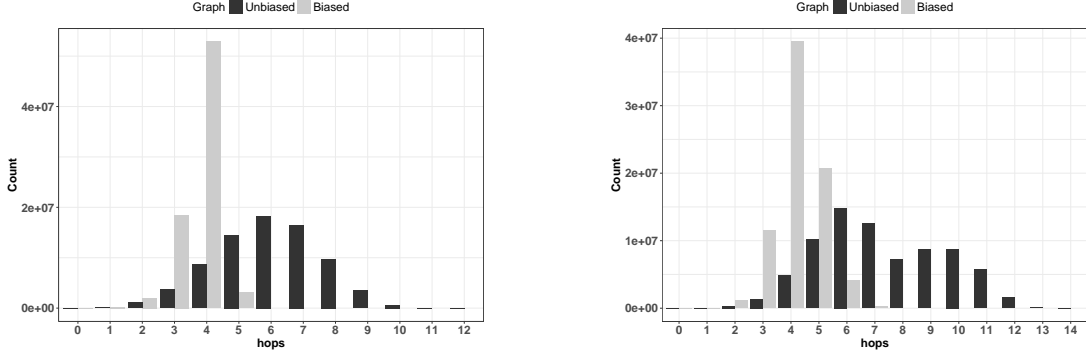


FIGURE 2: Distribution of graph path lengths for the experiments : model (left) and Youtube crawl (right).

2.1 Towards algorithms for tagging long biased edges

We propose to study potential bias in recommendation graphs, in relation to the *locality* or not of recommended items. For avoiding a formal - and possibly debatable - definition of bias, we instead state two of its most probable topological consequences on G^h :

Proposition 1 *Biased edges impact graph structure* : if the service bothers to bias recommendations to a user, it is because it effectively impacts that user navigation among items. That is, users do not navigate the same way in an unbiased graph G_R^h than in a graph G^h containing biased edges. Therefore, the existence of biased edges must significantly change the properties of the observed graph of recommendations.

Proposition 2 *Recommenders leverage item proximity, and this appears in a graph observation* : most of recommenders exploit some underlying coherence among the items. Collaborative filtering exploits correlations in users' tastes : if users enjoying a also usually enjoy b , b will be recommended from a **and vice versa**. This symmetry translates into edge locality in G_R^h , captured by e.g., clustering (as appearing through clusters on Figure 1). Biased edges might not rely on such property, and therefore are less likely to produce locality. We also can argue that if bias actually results in the proposal of usual (i.e., local) items for a user, she is not likely to consider those recommendations as biased.

In practice, machine learning algorithms associate to each item a d -dimensional vector of features ; a recommender then for instance relies on k nearest neighbors (k NN) or cosine similarity on those vectors. Recommended items are thus often close-by in the d -dimensional feature space, while we expect biased edges to point to items that are relatively far in the feature space. Note that this observation of the effect of biased edges on the topology may relate to services providing *serendipitous* recommendations [4] (for bringing diversity to a user) ; this nevertheless arguably constitutes a form of bias w.r.t. user habits.

A small-world perspective We now propose to identify bias using a parallel to the Watts-Strogatz “small-world” model [5]. In that model, nodes have local connectivity in a given space between them (capturing for instance a geographical proximity), but also have so called long-range links (capturing for instance a familial relationship, loosely related to a geographical proximity). The consequences of these long edges are well known : they drastically impact average path length. We argue that biased links added to the recommender output have the same impact : provided they are *different enough* from the recommender edges, they will impact the graph structure. To capture this degree of difference, we propose the following model.

A biased recommendation model Our model considers two recommenders on a set of I items : one « official » R , and one used to issue « biased » recommendations, R' . To model the fact that biased recommendations may not be completely unrelated to normal recommendations, we use a tuning parameter $i_{RR'}$ presented hereafter. Let R the « official » recommender be a k NN recommendation system producing k_R items per query ($\forall i \in I, |R_i(u)| = k_R$), based on the $p_i \in \mathbb{R}^d$ feature vector : $\forall i, j \in I^2, s_R^i(u, j) = \|p_i p_j\|_2$. Let E_R be the set of produced edges : $G_R^h = (I, E_R)$.

In addition to its p_i vector, each item $i \in I$ is associated with another d' dimensional feature vector $b_i \in \mathbb{R}^{d'}$, representing « hidden » features (profitability, political support) leveraged to bias recommendation. Biased edge set $E_{R'}$ is produced by a k NN recommender with k_B output using b_i . The observable graph for a user is $G^h = (I, E_R \cup E_{R'})$ in which nodes have an out-degree of $k = k_R + k_B$.

Features are generated uniformly at random : $\forall i \in I : (p_i[c])_{1 \leq c \leq d} \sim U(0, 1)$ and $(b_i[c])_{1 \leq c \leq d'} \sim U(0, 1)$.

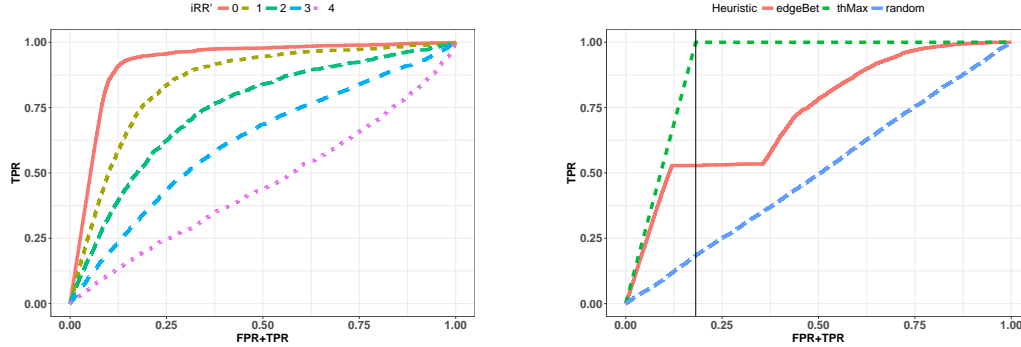


FIGURE 3: Tagging feasibility : ROC curves. Model (left) and Youtube crawl (right).

However, to vary the amount of bias, the dependency between p_i and b_i is as following : let $0 \leq i_{RR'} \leq \min(d, d')$. We set $\forall i \in I, \forall 1 \leq c \leq i_{RR'}, b_i[c] \leftarrow p_i[c]$. That is, if $d = d' = i_{RR'}$, both recommenders R and R' will produce exactly the same results (therefore $G^h = G_R^h$). On the other hand, if $i_{RR'} = 0$, p_i and b_i are independent, and so are the results of R and R' .

Experiments We re-use the Youtube crawl \mathcal{C} (Section 1), from an identified user u . In the set of Youtube recommendations at each page, some (around 20%) are tagged with the flag “Recommended for you” (whereas other simply display their number of views). For the experiment, we consider those recommendations as part of the biased set we seek to tag (ground truth). We parameter the model for comparable results : $|I| = 8753, k_R = 17, k_B = 2$. We set $d = d' = 5$.

First, we look at the small-world parallel with introduced bias under **Proposition 1** on Figure 2, where we plot the path length distribution of $G_R^4(u)$ and $G^4(u)$ (i.e., for Youtube, we use the full crawl, and then remove the “Recommended for you” edges). We note a clear change in the graph properties when biased edges are present : they shortcut many paths, as known in small-world graphs, in both experiments.

Second, **Proposition 2** is examined on Figure 3. For doing so, we run a symmetric edge-betweenness centrality algorithm on $G^4(u)$ graphs, for that metric is aimed at finding topologically important edges. We plot the ROC curves, representing the probability of biased edges actually ending-up in sorted top-result of the centrality metric. For the model, a bias based on few feature dependencies (e.g., $iRR'=1$ or 2) clearly allows for edge tagging. For Youtube, the heuristic is also way above a random tagging baseline, close to the maximum (thMax) for a while. For both experiments, we conclude that tagging algorithms have a clear room for providing accurate results : if one seeks the 4491 biased links of the dataset (recall@k, symbolized by the vertical line at $x = 0.19$), the betweenness on Youtube directly allows to identify 52% of the set.

Conclusion There are two main conclusions to this study. First, graphs of recommendations are an interesting abstraction for assessing recommender outputs : they clearly display the item locality a service provides users with (that we illustrate through the small-world parallel). Finally, if a service aims at biasing some recommendations, the effects might be witnessed on such user extracted graphs. We believe this brings both analytical and algorithmic interest in the scope of personalization transparency studies.

Future-work includes the proposal and comparison of heuristics for tagging biased edges, and the study of other applications of graphs of recommendations, possibly for usage by service providers themselves (e.g., Pagerank-like computation over *global* graphs of recommendations [2]).

Références

- [1] P. Covington, J. Adams, and E. Sargin. Deep neural networks for youtube recommendations. In *RecSys*, 2016.
- [2] E. Le Merrer and G. Trédan. The topological face of recommendation : models and application to bias detection. In *arXiv :1704.08991*, 2017.
- [3] E. Le Merrer and G. Trédan. Uncovering influence cookbooks : Reverse engineering the topological impact in peer ranking services. In *CSCW*, 2017.
- [4] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor. *Recommender Systems Handbook*. 1st edition, 2010.
- [5] D. J. Watts and S. H. Strogatz. Collective dynamics of ‘small-world’ networks. *Nature*, 393(6684) :409–10, 1998.