The Early-Adopter Graph and its Application to Web-Page Recommendation

Ida Mele*
Sapienza, University of Rome
Rome, Italy
mele@dis.uniroma1.it

Francesco Bonchi
Yahoo! Research
Barcelona, Spain
bonchi@yahoo-inc.com

Aristides Gionis
Yahoo! Research
Barcelona, Spain
gionis@yahoo-inc.com

ABSTRACT

In this paper we present a novel graph-based data abstraction for modeling the browsing behavior of web users. The objective is to identify users who discover interesting pages before others. We call these users *early adopters*. By tracking the browsing activity of early adopters we can identify new interesting pages early, and recommend these pages to similar users. We focus on news and blog pages, which are more dynamic in nature and more appropriate for recommendation.

Our proposed model is called *early-adopter graph*. In this graph, nodes represent users and a directed arc between users u and v expresses the fact that u and v visit similar pages and, in particular, that user u tends to visit those pages *before* user v. The weight of the edge is the degree to which the temporal rule "u visits a page before v" holds.

Based on the early-adopter graph, we build a recommendation system for news and blog pages, which outperforms other out-of-the-shelf recommendation systems based on collaborative filtering.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: [Information filtering]; H.2.8 [**H.2.8 Database applications**]: [Data Mining]

General Terms

Algorithms, Experimentation.

Keywords

user-browsing analysis, log mining, early-adopter graph, web-page recommendation.

1. INTRODUCTION

The digital revolution witnessed during the last decade has resulted in an explosion of available online content. An increasing number of people is now using the Internet on a daily basis to search for specific information, but also to stay informed by reading the news, or consuming user-generated content such as blogs.

Web-search engines are popular tools that allow people to search for information on the Internet. However, web search is effective only when the users have a clear idea of what they are looking. On the other hand, in many cases people have no specific information need, yet they are interested in discovering interesting and relevant content. Such are cases when people surf the Web to read news, funny stories, interesting blog posts, or check what their friends are posting in social-media platforms. Helping the users to find relevant content when they do not have a concrete information need is a problem of *information filtering*.

Recommender systems aim at producing relevant suggestions to the users of a system, and thus, they are essential tools in addressing the problem of information filtering. However, recommender systems offer effective mechanisms in static and relatively noise-free environments. For example, typical applications of recommender systems consist of recommending movies based on user-rating data, or recommending books based on purchase data, where a purchase is a clear indication of interest. On the other hand, designing recommender systems for web content, poses significant challenges due to high dynamicity: new pages appear continuously and old pages become obsolete very fast. Furthermore, a recommendation system based on user-browsing data should be able to deal with very high levels of noise, since visiting a web page is not as clear indication of interest as, say, renting a DVD to watch a movie or buying a book.

In this paper, we introduce a novel approach for making personalized web page recommendations, in particular recommending news articles and blog posts. Our idea is simple and intuitive: given a user-browsing log, we identify users who tend to discover interesting pages *before* others. We call these users *early adopters*, a term we borrow from social sciences, economics and marketing research, in which early adopters are people who embrace new technologies before others, buy new products soon after their release, and play an important role in influencing others to adopt innovations.

In contrast to previous approaches, we do not just identify clusters of similar users. Instead, we use the input log to build a directed and weighted graph among users. An edge between two users, u and v, expresses the fact that u and v

*This work was done while the author was an Intern at Yahoo! Research, Barcelona, Spain

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.

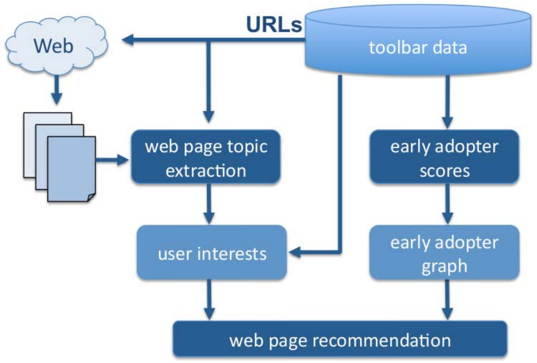


Figure 1: The workflow of our web-page recommendation framework.

visit similar pages and also that user u tends to visit those pages *before* user v. Thus, our model encodes the *latent temporal patterns* underlying user visits to web pages. As with technological innovations we assume that some users are better and faster in discovering interesting pages and other users "follow" them. By tracking the browsing activity of early adopters we can discover new interesting pages early on, and recommend these pages to users who "follow" the early adopters.

Our model is inspired by *information networks*, like `twitter`, in which users form a *social network* by following other users. In such a network, users influence each other, and information propagates by posts and re-posts of short messages. Much of social-network analysis research has been devoted to discovering *influential users* and quantifying the degree to which users influence each other [4, 5, 7, 10, 11, 13, 16, 17, 19]. Early adopters in our model correspond precisely to influential users in an information network, and link strength expresses the degree to which users are influenced by other users, as in social influence studies. The main difference is that the early-adopter graph is *implicit*, that is, users are unaware of the behavior of other users. However, as our results demonstrate, there is sufficient evidence that users exhibit common behavior and follow patterns of influence, which can be potentially explained by exogenous influences (events in the real world) and latent similarities among user interests.

Contributions. Our contributions can be summarized as follows: (i) we introduce the novel concept of early-adopter graph, a model we build based on user-browsing logs; (ii) we show how to use this model for recommending news articles and blog posts to users; (iii) we evaluate the recommendations provided by our system on a real dataset and we prove that they outperform state-of-the-art recommender systems.

2. MODEL OVERVIEW

In this section we provide an overview of the model. The outline of the overall workflow is given in Figure 1.

The input to our framework is a dataset $\mathcal{D}$ that records the browsing activity of a set of web users. Abstractly, we represent the dataset $\mathcal{D}$ as a set of triples (u, p, t), indicating that a user u visited the web page p at time t. For a user who visited the same page more than once, we keep in consideration only the first visit.

We create a dataset $\mathcal{D}$ by collecting browsing data from the *Yahoo! toolbar*. A toolbar is an application installed on top of a web browser that provides certain search functionalities, such as quick-links and other utilities.

The model we propose in this paper is an *attributed*, *weighted*, and *directed* graph $G(U, A, \Theta, \sigma, w)$, which we call the *early-adopter graph*, and which can be built from the dataset $\mathcal{D}$. The early-adopter graph is specified as follows:

- Each node in the early-adopter graph corresponds to a user $u \in U$. An arc $(u, v) \in A$ in the graph denotes the fact that there exists a page $p \in P$ such that $(u, p, t_u), (v, p, t_v) \in \mathcal{D}$ and $t_u < t_v$;
- We assume that we are given a set of topics $T = [1, K]$, and Θ is a user-topic matrix that associates a topic distribution $\vec{\theta}_u$ to each user $u \in U$. So, $\theta_{uz} = \Pr[Z = z \mid u]$ denotes the interest of user u in the topic $z \in T$ and $\sum_{z=1}^{K} \theta_{uz} = 1$ for each user $u \in U$;
- The function $\sigma : U \to \mathbb{R}$ is a score that represents the extent to which user u is an early adopter;
- Finally, the arc-weighting function $w : A \to \mathbb{R}$ represents the "strength" of the arc $(u, v) \in A$, or in other terms, the likelihood that a page visited by u will be then visited by v.

The idea underlying our approach is that whenever an early adopter u visits a page p for the first time, the information can be propagated along the edges of the early-adopter graph, and the page p can be recommended to other users $v \in U$. Ranking the page recommendations for each user will depend on various factors, such as the early-adopter score $\sigma(u)$ of the node u, the "influence" score $w(u, v)$ of the early adopter u to the user v, the topic of the page, and the interests of the users. We will combine all these factors by a ranking score $s(u, p)$ for each user u and a page p.

The problem we consider can be seen as a special case of the typical problem in recommender systems, however, there are a number of peculiarities. First, we do not have ratings but only visits to pages, which can be very noisy indications of interest. Second, we are dealing with a *cold-start problem*: we want to recommend new and interesting pages as soon as possible, even if we do not have sufficient information for those pages. And third, the pages we want to recommend are not given as input to the problem, as in the standard setting of recommender systems, but need to be *discovered*. We do this kind of discovery by exploiting the capability of early adopters to find interesting web pages before others.

3. THE EARLY-ADOPTER GRAPH

In this section we provide details on how to build the early-adopter graph and how to learn its parameters.

3.1 Dataset and graph construction

As we already explained, we start with a user-browsing log $\mathcal{D}$, consisting of triples (u, p, t), where (i) u is the anonymous id of the user, (ii) p is the url of visited page, and (iii) t is the timestamp of the visit of u at p. We collect such a dataset by sampling data from the *toolbar* log of *Yahoo!*. We restrict our analysis to five months of data, from January 2011 to May 2011, and we consider only urls of news pages and blog sites. To identify which urls correspond to news or blog sites we use a white list of known such sites. We also restrict our dataset to urls whose popularity (the number of distinct users that visited the page) is greater than 50.

For a pair of users u and v we define the *frequency* $\text{freq}(u, v)$ of the directed pair (u, v) to be the number of distinct pages $p \in P$ for which we observe in the data that the user u visited p before user v. In other words, $\text{freq}(u, v) = |P_{u;v}| = |\{p \in P \mid (u, p, t_u), (v, p, t_v) \in \mathcal{D} \text{ and } t_u < t_v\}|$.

In order to focus our analysis only to relevant arcs and reduce noise effects, we adopt a *minimum support threshold* $\theta \geq 1$. This means that we consider only arcs (u, v) such that $\text{freq}(u, v) \geq \theta$. In our experiments we use $\theta = 50$, which gives a pruned graph with 5 202 nodes and 335 091 arcs.

The average degree of 64.41 reflects that the graph is fairly dense. We also observed that users live on a "small world", as demonstrated by the average shortest-path length of 2.57. The number of communities that maximizes *modularity* is 54. Moreover, the graph has *strong* community structure at *microscopic level* and *weak* community structure at *macroscopic level*, as demonstrated by the very high clustering coefficient of 0.57 and the relatively low maximum value of modularity of 0.18, respectively.

3.2 Early-adopter score

For a page $p \in P$ we use $U_p \subseteq U$ to denote the set of users who visited p. Similarly, for a user $u \in U$ we denote with $P_u \subseteq P$ the set of web pages visited by u. Given a page $p_j \in P$ we can then organize the visits that p_j received by all users in a chronologically sorted access list $A(p_j) : \langle(u_1, t_{1j}), (u_2, t_{2j}), \ldots, (u_n, t_{nj})\rangle$, where t_{ij} is the timestamp of the first visit of user i to the page j. Naturally, early adopters tend to appear at the beginning of such a list.

In this paper we experiment with two different definitions of early-adopter score. Both definitions produce a score $\sigma(u) \in [0, 1]$ for all users $u \in U$. The more a user exhibits an early-adopter behavior, the higher is the score $\sigma(u)$. The score $\sigma(u)$ is computed as an average over all pages visited by u; more precisely, we define:

$$\sigma(u) = 1 - \frac{\sum_{p \in P_u} r(u, p)}{|P_u|},$$

where $r(u, p)$ is a measure of how early the user u appears in the access list $A(p)$. We adopt two definitions for $r(u, p)$, the first one considers only the relative position of the user u in the list $A(p)$, while the second one considers the relative time distance.

Relative position: we define the relative-position score to be $p_r(u, p) = \frac{|\text{pred}(u,p)|+1}{|A(p)|}$, where $\text{pred}(u, p)$ is the number of users who precede u in the list $A(p)$, and $|A(p)|$ is the length of the list.

Relative time distance: we define the relative-time distance to be $t_r(u, p) = \frac{t(u,p)-t_0(p)}{t_*(p)-t_0(p)}$, where $t(u, p)$ denotes the time that the user u visited page p, while $t_0(p)$ and $t_*(p)$ denote the time of the first and last visit to p, respectively.

Example 1 *Consider three pages, as shown in Figure 2, the early-adopter scores for user u_3, according to the two definitions provided above, are as follows:*

Relative position: $\sigma(u_3) = 1 - \frac{1}{3}(\frac{3}{6} + 1 + \frac{1}{4}) = 0.42$.

Relative time distance: $\sigma(u_3) = 1 - \frac{1}{3}(\frac{2}{5} + 1 + 0) = 0.54$.

The distributions of the relative-position and relative-time-distance scores for our dataset are shown in Figure 3(a).

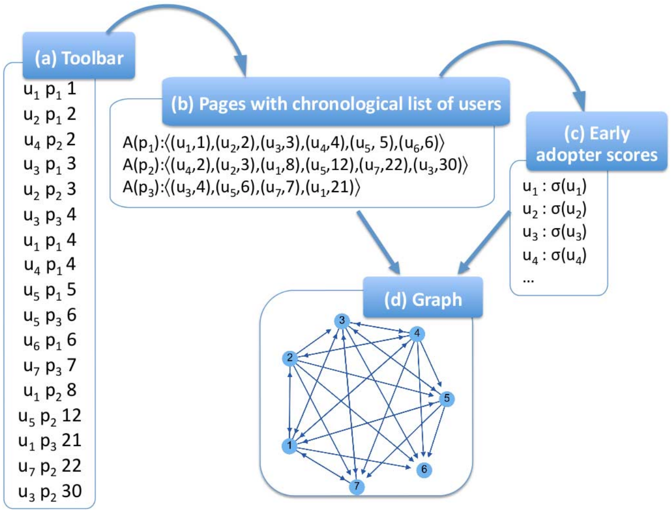
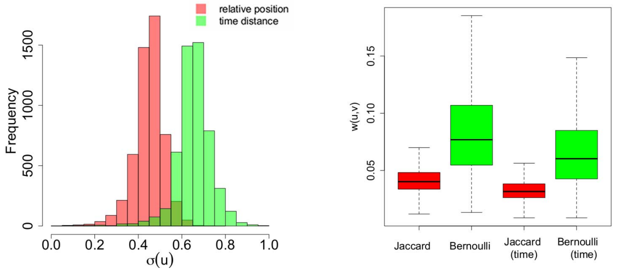


Figure 2: Building the early-adopter graph.



(a) Distribution of $\sigma(u)$ (b) Distribution of $w(u, v)$

Figure 3: The distributions of the early-adopter scores (left), and edge weights (right).

3.3 Arc strength

The strength of (u, v), expressed by a weight $w(u, v) \in [0, 1]$, represents the likelihood that a page visited by user u will be visited by user v.

Providing an estimate for the weight $w(u, v)$ resembles the problem of learning *influence probabilities* in social networks [11], which are used in applications such as *influence maximization* [13]. As we discussed in the introduction, the main difference between these social-networking studies and our early-adopter model is that in our application the graph is not explicit, instead it is reconstructed from user-browsing actions. Another difference is that we do not assume any underlying propagation model. Nevertheless, the problems are fairly similar, and thus, it is meaningful to try to estimate the strength of the arcs by using methods developed in the literature on social influence.

In this paper, we follow the work of Goyal et al. [11] and we adopt two basic ways of estimating the strength of an arc (u, v):

Bernoulli: $w(u, v) = \frac{|P_{u;v}|}{|P_u|}$.

Jaccard: $w(u, v) = \frac{|P_{u;v}|}{|P_u \cup P_v|}$.

The Bernoulli measure interprets each visit of u to a page as a hypothetical attempt of u to influence v in visiting the same page. The Jaccard measure considers also the pages visited by v and not by u: thus, it captures whether v follows *mostly* the actions of u and not many more.

One drawback of the two above definitions is that they consider only the ordering of the visits to the pages but not the time distance between such visits.

Based on the above consideration, to account for the time difference between page visits, we substitute the numerator $|P_{u;v}|$ of the definitions Bernoulli and Jaccard by a time-dependent term $\Delta^t_{u;v}$, defined as follows:

$$\Delta^t_{u;v} = \sum_{p \in P_{u;v}} \left(1 - \frac{t(v,p) - t(u,p)}{t_*(p) - t_0(p)} \right).$$

We then obtain two time-dependent versions for arc weights, in particular:

Bernoulli (time): $w(u,v) = \frac{\Delta^t_{u;v}}{|P_u|}$.

Jaccard (time): $w(u,v) = \frac{\Delta^t_{u;v}}{|P_u \cup P_v|}$.

The distributions of edge weights for our dataset are shown in Figure 3(b).

4. TOPIC MODELING

We now discuss how to extend our model by incorporating information regarding the topics of interest of the users.

First, we assume that each page belongs to one and only one topic. Then, we model each user u by a topic distribution $\vec{\theta}_u$, where the z-th coordinate $\theta_{uz} = \Pr[Z = z \mid u]$ denotes the interest of user u in the topic $z \in T$, and it is computed using the empirical frequency that a user visits pages of a certain topic.

For the classification, we consider 15 topics from the ODP directory, such as `entertainment`, `finance`, `politics`, `sports`, etc. Given a topic $z \in T$ we construct the *vocabulary* $V(z)$, which is a set of terms that it is typically associated with the topic. The vocabulary is composed of terms extracted from ODP categories and most discriminative words appearing in the web pages of the dataset.

Given a page $p \in P$ we create its bag-of-word representation $B(p)$, which is made out of terms appearing in the url, title, and content of the page. We normalize the terms by removing stop words, removing special characters, and converting to lower case.

In order to find the topic z for which the vocabulary $V(z)$ matches best with the bag representation $B(p)$, we apply a `tf.idf`-based measure. In particular, given a term $t \in V(z)$ we compute the $\texttt{tf.idf}(t,p)$ weight of the term t in the page p. The classifier assigns the page p to the topic $z(p) \in T$ such that $z(p) = \arg\max_{z \in T} \left\{ \sum_{t \in V(z)} \texttt{tf.idf}(t,p) \right\}$. If the maximum score is zero, p is assigned to the `notClassified`.

Using the approach described above we are able to assign a topic distribution to almost all the users in our dataset. We note that we have used this simple classification algorithm as a proof-of-concept demonstration, and using a better classifier has the potential to improve our results.

5. WEB-PAGE RECOMMENDATION

Our recommendation approach leverages the information found in the early-adopter influence graph G. Given an arc $(u,v) \in A$ we consider suggesting to user v pages that have been visited by user u. To improve the relevance of our recommendations, we rank recommendations by considering the early-adopter score $\sigma(u)$ of the user u from whom the recommendation originates, as well as the edge weight $w(u,v)$ that reflects the strength of the connection between u and v. Additionally, we use page topics to boost scores of pages whose topics match the interests of the user v. Overall, the recommendation score $s(v,p \mid u)$ of a page p recommended to v, given that the recommendation has originated by the early adopter u is:

$$s(v,p \mid u) = 1 - \left[(1 - \sigma(u))(1 - w(u,v))(1 - \theta_{vz^*}) \right],$$

where z^* is the topic of the recommended page p, and θ_{vz^*} is the preference of user v for that topic.

When a page is suggested to v by different early adopters, the final score $s(v,p)$ of p recommended to v is the sum of all the scores. Hence,

$$s(v,p) = \sum_{u \in N^-(v,p)} s(v,p \mid u) \qquad (1)$$

where $N^-(v,p)$ is the set of early adopters who have an arc to v and have visited the page p. The recommendation algorithm computes all these scores and then creates a ranked list of pages to recommend. Our empirical evaluation, described in the next section shows that our recommendation algorithm predicts user clicks with very good precision.

5.1 Empirical evaluation

We evaluate our recommendation algorithm on the dataset described in Section 3. We split our dataset, at the level of pages, in two portions: *training* and *test* sets. Referring to our notation in Section 3.2, we form the access lists $A(p)$ for all pages p and then we split the set of those lists at a ratio of 80-20 for the training vs. test sets. We also ensure that the two sets have a similar distribution in terms of the popularity of the pages.

The training subset is used to build the early-adopter graph G, learn the early-adopter scores $\sigma(u)$, the arc weights $w(u,v)$, and the topic distributions $\vec{\theta}_u$. Given a user v and the set of early-adopters $N^-(v)$, the algorithm recommends pages visited by $u \in N^-(v)$ to v. These pages are ranked by the score defined in Equation (1). The recommendation algorithm is evaluated by using `precision-at-k` (`p@k`) for $k = 1, 5, 10, 15$, which gives an indication of the percentage of recommended pages that are actually visited by v.

We compare our recommendation algorithm against collaborative-filtering approaches. We assume that a click on a page corresponds to a rating equal to 1 while a non-click corresponds to a rating equal to 0, and we compute user similarity with *Tanimoto* and *Log-Likelihood* coefficients.

For brevity we present the results achieved with the Bernoulli definition of the edge weights.[1] As we can see in Figure 4, our approach outperforms significantly algorithms based on collaborative filtering. In particular, the improvement is of 10% for `p@1` and of 20% for `p@15`.

We note that this is a difficult recommendation task, and a certain level of noise is present. Nevertheless, our methodology is completely automated, and we think that it is appropriate for comparing different algorithms.

[1] The results obtained with the other edge-weight definitions are qualitatively the same.

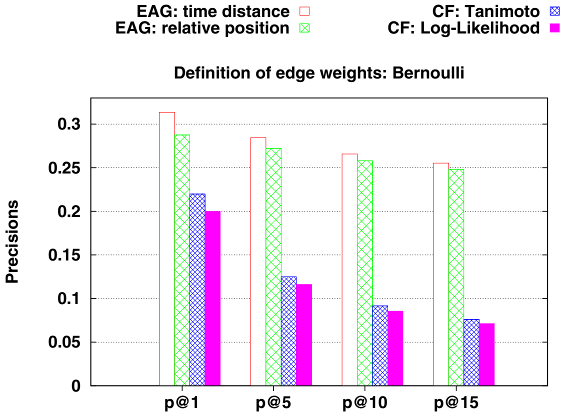


Figure 4: Precision results: early-adopter graph (EAG) vs. collaborative-filtering (CF).

6. RELATED WORK

Web-usage mining. Web logs represent a valuable source of information to study user behavior and to improve user web experience. Bilenko et al. [6] and White et al. [18] analyze web-activity logs to identify web sites frequently visited by users after a query. These "popular" destinations are then used to suggest authoritative websites for queries.

Recommender systems. Recommender systems allow to learn user preferences and make recommendations, based on user past behavior. They are used extensively for recommending products (e.g., books, movies, music, etc.) and helping people finding web content (e.g., news, photos, etc.).

Das et al. [9] propose a scalable content-agnostic approach based on collaborative filtering to recommend news. Resnick et al. [14] present a distributed system for gathering reader ratings of news.

Social influence and information propagation. Our work is also related to the large body of research on social influence and information propagation. The main computational problems in this area are: (*i*) distinguishing genuine social influence from "homophily" and other factors of correlation [3, 4, 8, 10]; (*ii*) measuring the strength of social influence over each social link [11, 16, 17, 19]; (*iii*) discovering a set of influential users [2, 13]. Besides, many researchers have focused on analyzing data (e.g., `twitter` data) to better understand the phenomenon of viral propagation of information in social networks and micro-blogging platforms [1, 5, 7, 12, 15].

7. CONCLUSIONS

In this paper we introduce a novel approach for recommending web pages. We exploit user-browsing behavior data to construct an *implicit* network of influence. We then identify users who discover interesting pages and we call them *early adopters*. The general idea of our framework is to monitor the activity of early adopters, and recommend pages discovered by them to users who "follow" the early adopters.

The early-adopter graph is a general model and its application to other domains deserves further investigation. As future work we also plan to investigate applying different influence models to learn edge weights, as well as applying a more sophisticated topic classifier.

8. REFERENCES

[1] E. Adar and L. A. Adamic. Tracking information epidemics in blogspace. In *Web Intelligence*, 2005.

[2] N. Agarwal, H. Liu, L. Tang, and P. S. Yu. Identifying the influential bloggers in a community. In *WSDM*, 2008.

[3] A. Anagnostopoulos, R. Kumar, and M. Mahdian. Influence and correlation in social networks. In *KDD*, 2008.

[4] S. Aral, L. Muchnik, and A. Sundararajan. Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. *Proc. of the National Academy of Sciences*, 106(51):21544–21549, 2009.

[5] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts. Everyone's an influencer: quantifying influence on twitter. In *WSDM*, 2011.

[6] M. Bilenko and R. W. White. Mining the search trails of surfing crowds: identifying relevant websites from user activity. In *WWW*, 2008.

[7] M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi. Measuring user influence in twitter: The million follower fallacy. In *ICWSM*, 2011.

[8] D. J. Crandall, D. Cosley, D. P. Huttenlocher, J. M. Kleinberg, and S. Suri. Feedback effects between similarity and social influence in online communities. In *KDD*, 2008.

[9] A. Das, M. Datar, A. Garg, and S. Rajaram. Google news personalization: scalable online collaborative filtering. In *WWW*, 2007.

[10] T. L. Fond and J. Neville. Randomization tests for distinguishing social influence and homophily effects. In *WWW*, 2010.

[11] A. Goyal, F. Bonchi, and L. V. S. Lakshmanan. Learning influence probabilities in social networks. In *WSDM*, 2010.

[12] D. Gruhl, R. V. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *WWW*, 2004.

[13] D. Kempe, J. M. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *KDD*, 2003.

[14] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *CSCW*, 1994.

[15] D. M. Romero, B. Meeder, and J. M. Kleinberg. Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter. In *WWW*, 2011.

[16] K. Saito, R. Nakano, and M. Kimura. Prediction of information diffusion probabilities for independent cascade model. In *KES*, 2008.

[17] J. Tang, J. Sun, C. Wang, and Z. Yang. Social influence analysis in large-scale networks. In *KDD*, 2009.

[18] R. W. White, M. Bilenko, and S. Cucerzan. Studying the use of popular destinations to enhance web search interaction. In *SIGIR*, 2007.

[19] R. Xiang, J. Neville, and M. Rogati. Modeling relationship strength in online social networks. In *WWW*, 2010.