DifRec: a Social-Diffusion-aware Recommender System

Hossein Vahabi
Yahoo Labs, Spain
puya@yahoo-inc.com

Iordanis Koutsopoulos
AUEB and CERTH, Greece
jordan@aueb.gr

Francesco Gullo
Yahoo Labs, Spain
gullo@yahoo-inc.com

Maria Halkidi
Univ. of Piraeus, Greece
mhalk@unipi.gr

ABSTRACT

Recommender systems used in current online social platforms make recommendations by only considering how relevant an item is to a specific user but they ignore the fact that, thanks to mechanisms like sharing or re-posting across the underlying social network, an item recommended to a user i propagates through the network and can reach another user j without needing to be explicitly recommended to j too. Overlooking this fact may lead to an inefficient use of the limited recommendation slots. These slots can instead be exploited more profitably by avoiding unnecessary duplicates and recommending other equally relevant items.

In this work we take a step towards rethinking recommender systems by exploiting the anticipated social-network information diffusion and withholding recommendation of items that are expected to reach a user through sharing/re-posting. We devise a novel recommender system, DifRec, by formulating the problem of maximizing the total user engagement as an allocation problem in a properly-defined neighborhoodness graph, i.e., a graph that models the conflicts of recommending an item to a user who will receive it anyway by social diffusion. We show that the problem is NP-hard and propose efficient heuristics to solve it.

We assess the performance of our DifRec by involving real data from Tumblr platform. We obtain substantial improvements in overall user engagement (130–190%) over the real recommender system embedded in Tumblr and over various existing recommender systems.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms, Optimization, Experimentation

Keywords

Social diffusion aware recommendation, User engagement

CIKM'15, October 19–23, 2015, Melbourne, VIC, Australia.
© 2015 ACM. ISBN 978-1-4503-3794-6/15/10 ...$15.00.
DOI: http://dx.doi.org/10.1145/2806416.2806559 .

1. INTRODUCTION

Online social platforms like Twitter, Tumblr, Facebook, Google+, Weibo, where users post multimedia content and create (virtual) social relationships have proliferated a lot in the last few years. The amount of posted information keeps increasing to unimaginable levels. Twitter, for instance, has 271 million active users who generate more than $6,000$ tweets per second. Tumblr has a user base of 50 millions and hosts about 200 million blogs and 90 billion posts.

These platforms have even created a new lifestyle and keeping users highly engaged is of utmost importance for their owners. Indeed, increased user engagement is strongly correlated to higher revenue, as the user is more likely to view or click on ads or other items that the platform owner monetizes on. Moreover, when users are engaged with the platform, they contribute in generating content that can be exploited for crucial tasks like profiling or personalization.

Recommender systems constitute a primary component for improving online user experience, and, as such, increase user engagement. Depending on the specific online platform, recommendations concern different types of items. For instance, Twitter uses a recommendation service for accounts or tweets, Facebook recommends friends and news, while Tumblr recommends posts. Regardless of the type of recommended item, the primary goal of a recommender system is to facilitate access to content that is of interest to users, without making them waste time in manually inspecting tons of irrelevant information.

A major feature of existing recommender systems is that they make recommendations by focusing only on the specific user to whom the recommendation is directed [22, 15]: an item is recommended to a specific user if it has been recognized as "relevant" to that user according to a certain measure of relevance. However, in such platforms a user is not an isolated entity, rather she is part of a social network that abstracts relationships among users. The social network underlying these online platforms can be represented as a directed graph where the semantics of each edge (i, j) is that user j "follows" user i, that is, user j has visibility on all content generated or shared/re-posted by user i.

The availability of a social network constitutes a unique opportunity to empower the quality of a recommender system and get higher overall user engagement. Indeed, when an item (e.g., a news, a post, a URL, a multimedia file) is recommended to user i, it directly contributes to an increase in the user i engagement with the platform. However, if this item is spread further by mechanisms like sharing or re-posting, it will also indirectly contribute to the engagement of users that are connected to i through the underlying social network. If in turn these users share/re-post the item, other users connected to them will also receive it, and so on.

As a more concrete example, consider an item that has been recognized as relevant to user i of a certain social platform, and, as such, it has been recommended to her. Assume also that the same item is relevant to i's followers and, as such, has been recommended to them too. If user i re-posts the item, their followers will receive it twice, both via re-posting and from the recommender system. Since the maximum number of items to be recommended to a user is limited to a few at a time (to avoid overloading the user with too much information), the scenario above leads to an undesirable situation where a significant amount of slots available for recommendation is wasted for recommending duplicate items that are expected to reach some users (i.e., i's followers) via re-posting. Instead, a more sophisticated recommender system could take into account anticipated reposting and recommend other equally relevant items to the followers of i, thus further increasing their engagement.

In this work we aim at incorporating social-network information diffusion in the recommendation strategy of recommender systems underlying online social platforms. To the best of our knowledge, this aspect is overlooked by all existing recommender systems. Even the so-called *social recommender systems* [14, 25, 7] do not consider it. Indeed, they aim at exploiting the social network only for better assessing how relevant an item is to a user, that is, for measuring the relevance of an item to a user by incorporating social contextual information such as the relevance of an item to the social neighborhood of that user.

1.1 Motivating example

Consider the simple example reported in Figure 1, where $N = 5$ users are organized in the social network depicted in the figure. Let $\mathcal{S} = \{t1, t2, t3, t4, t5\}$ be the set of possible items to be recommended to each user, and let the relevance scores of items in $\mathcal{S}$ to a user i be defined by the following vectors: $\vec{r}_1 = [0.9, 0.1, 0.3, 0.7, 0.4]$ (user 1), $\vec{r}_2 = [0.8, 0.6, 0.4, 0.3, 0.5]$ (user 2), $\vec{r}_3 = [0.5, 0.7, 0.2, 0.1, 0.9]$ (user 3), $\vec{r}_4 = [0.4, 0.8, 0.7, 0.1, 0.3]$ (user 4), $[0.7, 0.1, 0.5, 0.7, 0.4]$ (user 5). Moreover, assume that each user can be recommended only $K = 1$ items, and that re-posting always happens, i.e., every item recommended to user i is always re-posted and becomes available to all i's followers.[1]

A traditional recommender system would recommend items to a user based only on the relevance scores to that specific user, thus leading to the recommendations illustrated in Figure 1(a), where each user receives the item that is most relevant to her. Hence, users 1,2,3,4,5 are recommended items $t1$, $t1$, $t5$, $t2$, $t1$, respectively, and the total relevance of the items viewed by users, which include both the recommended items and the ones reaching a user via re-posting, is computed as follows. For user 1, the total relevance is 0.9, due to $t1$ only (she is not follower of anyone). For user 2, the total relevance is 0.8: user 2 receives $t1$ through both recommendation *and* re-posting (from user 1), but it clearly counts only once, as duplicates do not contribute to increase relevance. User 3 receives $t5$ through recommendation and $t1$ through re-posting from user 1, therefore her total relevance is 1.4. Similarly, the total relevance of users 4 and 5 are equal to 1.2 and 0.7, respectively. Summing all these individual total relevance scores leads to a relevance of the overall recommendation equal to 5.

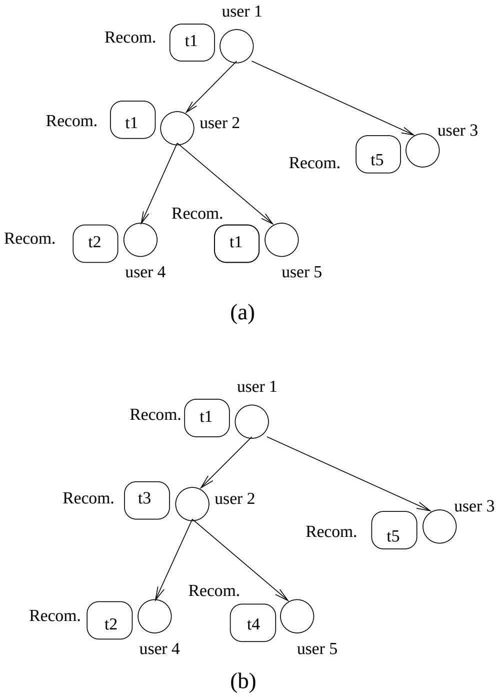

Figure 1: Example showing how more accurate a social-diffusion-aware recommender system can be with respect to traditional recommender systems. (*a*) Recommendations provided by traditional recommender systems lead to overall relevance of 5. *(b)* Social-diffusion-aware recommendations lead to overall relevance equal to 7.3.

Consider now a recommender system that takes into account social-network information diffusion, i.e., it recommends items not only based on the relevance scores but also considering the relevance of the item to the users who will receive that item via re-posting (Figure 1(b)). As shown later on, finding the optimal recommendations according to such a rule is an NP-hard problem. However, even solving the problem heuristically may lead to much higher overall relevance than in the case where social-network diffusion is discarded. Indeed, assume that we process the users in sequence, starting from user 1. Any item recommended to user 1 will reach every other user in the network, thus the item to be recommended to user 1 should correspond to the item that maximizes the sum of the relevance scores among all users; this is item $t1$, which leads to a (maximum) total relevance of $0.9 + 0.8 + 0.5 + 0.4 + 0.7 = 3.3$. For user 2, we exclude $t1$ that will be received anyway via re-posting from user 1, and we pick the item that maximizes the total relevance of nodes $2, 4, 5$, which are the nodes that are reached by re-posting originated in user 2; this corresponds to item $t3$, which leads to total relevance of $0.4 + 0.7 + 0.5 = 1.6$. With similar arguments, one can observe that the items recommended to users 3, 4, and 5 are $t5$, $t2$, and $t4$, respectively, and the relevance of the overall recommendation by summing all individual total relevance scores is 7.3, which is 46% more than the previous case.

[1] In the proposed recommendation method we actually include a probability of re-posting so as to maintain as much generality as possible. In this example we assume probability equal to one for simplicity.

1.2 Contributions

In this work we aim at rethinking recommender systems underlying online social platforms by incorporating the anticipated social-network information diffusion in the recommendation strategy. The main goal is to withhold recommendations of items that will anyway reach a user via sharing/re-posting mechanisms and utilize the limited slots available for recommendations in a more profitable way. To the best of our knowledge, this dimension is hitherto unexplored in the current recommender-systems literature. Specifically, the contributions of our work are as follows.

- We devise a model of social-network diffusion that captures diffusion and user behavior aspects, such as reposting probability, number of re-posts, and probability that information propagates from one user to another. The proposed diffusion model is exploited to define a neighborhoodness graph that models the redundancy of recommending the same item to two users if this will be received anyway through social-network diffusion.
- We formulate the problem of maximizing total user engagement through social-diffusion-aware recommendations as an allocation problem in the neighborhoodness graph.
- We show that the problem is NP-hard, and propose efficient heuristics that stem from the Max-Weight Independent Set (MWIS) problem.
- We incorporate all these ingredients (i.e., social diffusion models, neighboorhoodness graph, heuristics to solve the social-diffusion-aware recommendation problem) in a novel recommender system that we call DifRec. A major feature of the proposed DifRec is that, thanks to its very general design principles, it can be used in every existing online social platform.
- We assess the performance of the proposed social-diffusion-aware DifRec recommender system in a real-world scenario involving *real* users and posts coming from the Tumblr platform. Our evaluation shows that our method achieves substantial improvement—in most cases 130–190%—in user engagement with respect to the real recommender system currently embedded in that platform and over various existing recommender systems.

The rest of the paper is organized as follows. Section 2 discusses related work. In Section 3 we report the main concepts and notations used in the paper. In Section 4 we describe the proposed DifRec recommender system in detail. Section 5 presents an experimental evaluation on a dataset coming from a real online social platform, while Section 6 concludes the paper.

2. RELATED WORK

To the best of our knowledge, the proposed DifRec recommender systems has unique features that do not appear in any other existing recommender system defined in the literature. Even the so-called *social recommender systems* (that we discuss next) do not consider it. Indeed, they aim at exploiting the social network only for better assessing how relevant an item is to a user, that is, for measuring the relevance of an item to a user by incorporating social contextual information such as the relevance of an item to the social neighborhood of that user.

In the following we however provide an overview of the most prominent works in some neighborhood areas.

Information propagation. Kempe *et al.* [16] formulate the problem information propagation (or influence maximization) as follows: given a graph and a weight w_{ij} for edge (i, j) denoting influence probability, find a set S of seed users such that the largest number of users in the graph are influenced. Kempe *et al.* show that the problem is NP-hard and propose a greedy algorithm with constant-factor approximation guarantees.

Subsequent works on the same problem attempt to improve performance and reduce complexity of seed selection [18], or study different ways of quantifying influence spread that are reasonably close to that of the greedy algorithm [6, 17]. The same problem is also tackled from different perspectives, such as a data-driven approach that learns how influence flows in the network [4], a statistical-physics approach [3], and a mechanism-design approach that elicits influence probabilities [20]. The problem of maximizing diffusion speed under strategic users that make selfish adoption decisions is studied in [21]. There also exists a body of research which focuses on extracting graph and propagation features (e.g., propagation speed) out of observed cascade data through convex maximum-likelihood problem formulations [9, 10, 24], or applying diffusion models in the context of Twitter, e.g., to predict diffusion of URLs, given a training set of URL mentions [8].

In [11] the authors consider the problem of campaign design under an opinion-formation model in social networks. The recent work [19] studies the relation between user (re)posting and social graph evolution in Twitter. The works [12, 13] present models for predicting information diffusion. The former model combines social, semantic and temporal node features to make predictions about information propagation and takes into account local behaviour of users. The latter work presents a survey on information diffusion models and methods for social networks.

Recommender systems for social networks. The work [14] uses Bayesian inference on a social network to measure rating similarity between friends and calculates the most probable recommendation through a Bayesian network inference framework. Pennacchiotti *et al.* [25] introduce the problem of tweet recommendation for one user with the objective to maximize overall interestingness of recommended tweets. Two metrics use textual content of tweets to quantify interestingness of a tweet to a user. Chen *et al.* [7] propose an approach for recommending tweets based on a collaborative ranking model. The method relies on latent factors to unveil common user interests over tweet content and user social relations. Also [2] presents an approach for analyzing social media streams to provide personalized topic recommendation to users using matrix factorization to optimize the personalized ranking of topics.

Another line of works are implicitly connected to recommender systems. In [26], the authors introduce tweet entity linking to model user interests. They identify and link named entities that users mention in their tweets and construct a graph that shows interdependence between named entities for each user. A user interest propagation algorithm similar to PageRank is proposed to propagate interest scores.

In [27], the authors present the use of Co-Factorization Machines to build predictive models in Twitter.

In [28], an online version of collaborative-filtering-based recommender system is presented, that caters for new recommendation requests and item ratings. The work [29] demonstrates an online tweet recommender system based on matrix factorization to recommend topics (hashtags) according to real-time user interests. In [30], the authors propose a tweet summarization approach to select tweets that best represent user interests. The time dimension is integrated with an iterative optimization algorithm to select tweets based on novelty, coverage and diversity. All the works above on recommender systems for social networks deal with the tweet or post recommendation problem without consideration of diffusion of information in the network.

Recommender systems and social-network diffusion. In a recent work Pan *et al.* [31] integrate collaborative filtering and information diffusion to design recommender systems so as to deal with social-update information overload. The diffusion subgraph is used so as to identify similar users to a certain user. However in our work, the diffusion of reposts is used in a different way that aims at increasing the number of tweets that reach the user through recommendation and re-posts. We *model the anticipated diffusion* of a recommended item and use it to guide recommendation by recommending posts other than the ones that are likely to reach users via re-posts of others.

Another recommendation approach for increasing diffusion of a tweet, that of mentioning users in tweet, is presented in [32]. Users that are recommended (mentioned) in a tweet receive notification about the tweet and may re-tweet it, thus contributing to its diffusion. The problem of selecting users to mention is formulated as a ranking problem. Contrary to that, we exploit the anticipated social-network diffusion so as to optimize the recommendation of posts to users and thus increase user engagement.

Finally it is worth mentioning that avoiding duplicate (or highly similar) items has been a long-studied problem in information retrieval [1, 5, 23]. Our diffusion-aware recommender system can interestingly be viewed as an alternative way to augment the diversity of items seen by users, where the diversity is considered from a new angle, i.e., from a social-network-diffusion perspective.

3. PRELIMINARIES

We consider a set $\mathcal{U}$ of N users that are organized in a directed social graph $G = (V, E)$, where $V = \mathcal{U}$ and E is the set of edges. When a directed edge $(i, j) \in E$ exists between users i and j we say that user j "follows" i, with the implicit meaning that j has visibility on all the content generated/shared/re-posted by i. We also denote by $\mathcal{F}_i$ the set of users who follow user i, i.e., $\mathcal{F}_i = \{j \in V \mid (i, j) \in E\}$, and by $\mathcal{G}_i$ the set of users who are followed by i, i.e., $\mathcal{G}_i = \{j \in V \mid (j, i) \in E\}$.

In the remainder, we use the generic term "post" to refer to any piece of information (e.g., a news, a multimedia file, a URL, a tweet) that circulates through the social network. Thus, the term "post" may refer to either content generated by a user or an item provided to a user as a recommendation. We also use the term "re-post" to denote the action of sharing a piece of information from a user to all her followers. A user i may generate new posts, as well as re-post other posts that she has generated by herself, or have come to her by other users or through a recommender system.

Table 1: List of symbols used in the paper.

$\mathcal{S}$	Set of posts
$\mathcal{U}$	Set of users
N	Number of users
t	Post index
i, j, k	User indexes
$G = (V, E)$	Social-network graph with node (user) set V and edge set E
$G = (V_n, E_n)$	Neighborhoodness graph with node set V_n and edge set E_n
$\mathcal{L}_i$	Set of posts recommended to user i
K	Max number of posts recommended to each user
$\mathcal{F}_i$	Set of users who follow user i
$\mathcal{G}_i$	Set of users who user i follows
r_{it}	Relevance factor of post t to user i
e_{it}	Engagement factor of post t for user i
w_i	Re-post probability of user i
d_i	Out-degree of node i
$\mathcal{I}_t, W(\mathcal{I}_t)$	Independent set corresponding to post t as output of Algorithm 2
$W(\mathcal{I}_t)$	cost of independent set $\mathcal{I}_t$
E_{it}	Global engagement potential for post t when recommended to user i according to the proposed diffusion model
G_i	Diffusion subgraph with source i
$\mathcal{R}_i$	Reliable node set of user i
η	Threshold to compute reliable node sets
$\mathcal{D}$	Set of deterministic graphs sampled to generate reliable node sets
x_{it}	Control variable, equal to 1 if post t is recommended to user i, and 0 otherwise

A user i may therefore receive posts in two different ways: (i) via re-posting from users followed by her, (ii) through recommendations provided by some recommender system. In this work we focus on the impact of the recommended posts on the overall user engagement, i.e., the engagement of all the users in the network. Particularly, we fix the attention to a specific time point and study a static scenario where a set $\mathcal{S}$ of posts is available for recommendation, and the system has to recommend to each user $i \in \mathcal{U}$ a set $\mathcal{L}_i \subset \mathcal{S}$ of up to $K \ll |\mathcal{S}|$ posts from $\mathcal{S}$. K denotes the maximum number of recommendation slots available for each user ($|\mathcal{L}_i| \leq K, \forall i \in \mathcal{U}$). In real online platforms, in order to prevent users from information overload, K is typically in the order of a few tens (in some cases, it can even be equal to one). The dynamic scenario can easily be addressed by periodically updating G and $\mathcal{S}$ and running the recommendation method on the updated data. At the same time, however, a dynamic scenario clearly comes with additional problems and offers further research directions (e.g., incremental computation of the recommendations, changes in the relevance of a post, and so on), which are beyond the scope of this paper and are deferred to future work.

For each post $t \in \mathcal{S}$ and user $i \in \mathcal{U}$, we denote by r_{it} the *relevance* factor (or simply *relevance*) of t to i. Factor r_{it} quantifies how interesting post t is to user i, and, in general, it is estimated from the history of posts that user i has generated, viewed, or re-posted. Here we assume the relevance factors are computed somehow (many methods exist in the literature [25]); we provide further details on the specific method used in this work in Section 5.

The main symbols and notations used in this paper are summarized in Table 1.

4. THE DifRec RECOMMENDER SYSTEM

In this section we describe the proposed social-diffusion-aware DifRec recommender system. A major feature of our DifRec is that its general design principles make it suitable to work on any existing online social platform.

We start by introducing the diffusion model that is used to define the set of users who receive posts re-posted by a user (Section 4.1). Based on this model, we define the concept of *neighborhoodness graph* and formulate the diffusion-aware recommendation problem as an allocation problem on this neighborhoodness graph (Section 4.2). We show that the problem is NP-hard and propose a heuristic aimed at returning suboptimal solutions efficiently (Section 4.3).

4.1 Diffusion Model

We assume each user i is assigned a probability of re-posting w_i. This probability depends on how often a user tends to re-post and can be estimated from past interaction with the social platform. In Section 5 we describe how we specifically compute these probabilities for the platform considered in our experiments.

As said above, when a post is recommended to user i, it directly contributes to the engagement of that user, but it also indirectly contributes to the engagement of users $j \in \mathcal{F}_i$ that follow i. This is because if a post is of interest to user i, she may re-post it, and thus the post will be viewed by her followers as well. In turn, the followers j may re-post that post so that it will reach users that follow them, and so on. It can be seen that the initial direct recommendation of a certain post t to user i results in an *information diffusion process* in the social graph.

In the following we describe the model of diffusion considered in this work. The ultimate goal of using a diffusion model is to define the set of users who are reached by any re-posting initiated by user i. This is exploited later on (Sections 4.2–4.3) to prevent the proposed recommender system to recommend a post that will be received anyway via re-posting. We point out that our overall recommendation method is however independent of the specific diffusion model used. Any other model can be easily plugged in.

Diffusion model: Reliable node set of source i. The proposed diffusion model exploits re-posting probabilities assigned to each user to create a probabilistic graph $\widetilde{G}$ that has the same structure as the input graph G and whose edges are assigned an existence probability. More specifically, the probabilistic graph $\widetilde{G}$ is defined as a triple (V, E, p), where V and E are the node set and edge set of the input graph G, respectively, and $p : E \rightarrow (0, 1]$ is a function that assigns a probability of existence to each edge in such a way that $p(i, j) = w_i$, for all $(i, j) \in E$.

Given the probabilistic graph $\widetilde{G}$, the goal is to identify, for each user $i \in V$, a set of nodes that are reachable from i with high probability. This corresponds to finding the so-called *reliable node set* of user i [36]: given a probabilistic graph $\widetilde{G}$, a source node i, and a probability threshold $\eta \in [0, 1]$, find the set $\mathcal{R}_i$ of all nodes that are reachable from i with probability larger that η.

Computing the exact reliable node set is a hard problem, as it intrinsically relies on the problem known as *two-terminal reliability*, which consists in computing the probability that a node is reachable from another in a probabilistic graph and is known to be #P-complete [33, 34]. However, the reliable set $\mathcal{R}_i$ of node i can be accurately yet efficiently estimated via Monte Carlo sampling. The idea is to sample a set $\mathcal{D}$ of deterministic graphs from the probabilistic graph $\widetilde{G}$ according to edge probabilities and, for each of these deterministic graphs, computes the nodes reachable from i. The reliable set $\mathcal{R}_i$ is eventually computed as the set of all nodes that are reachable from i in the majority of the sampled graphs. A pseudocode of the method is reported as Algorithm 1. As the method basically performs a visit of each deterministic graph in $\mathcal{D}$, its time complexity is $\mathcal{O}(|\mathcal{D}| \times |E|)$. As a result, computing the reliable node set of all nodes in the input graph takes $\mathcal{O}(|\mathcal{D}| \times |V| \times |E|))$. Even though this time complexity may seem prohibitive for large graphs, this is not really the case as (*i*) due to the sampling strategy that relies on edge probabilities, deterministic graphs in $\mathcal{D}$ have in practice size much smaller than the input probabilistic graph $\widetilde{G}$, and (*ii*) the overall computation can easily be parallelized since reliable node sets are independent from a node to another.

Algorithm 1 Algorithm *ReliableSet*: Estimate via Monte Carlo sampling the set $\mathcal{R}_i$ of all nodes reachable from node i with probability larger than η in a probabilistic graph.

```
input: Probabilistic graph $\widetilde{G} = (\widetilde{V}, \widetilde{E}, p)$, node $i \in \widetilde{V}$,
    probability threshold $\eta \in [0, 1]$, number of samples $T \in \mathbb{N}$.
output: A set of nodes $\mathcal{R}_i \subseteq \widetilde{V}$.

$c(j) \leftarrow 0, \forall j \in V$
for $h = 1, \ldots, T$ do
    $E_h \leftarrow \emptyset$
    for all $e \in \widetilde{E}$ do
        $rnd \leftarrow$ a random number $\in [0, 1]$
        if $rnd \leq p(e)$ then
            $E_h \leftarrow E_h \cup \{e\}$
        end if
    end for
    $\mathcal{R}_i^{(h)} \leftarrow$ all nodes reachable from $i$ in the graph $G_h = (V, E_h)$
    $c(j) \leftarrow c(j) + 1, \forall j \in \mathcal{R}_i^{(h)}$
end for
$\mathcal{R}_i \leftarrow \{j \in V \mid c(j) \geq T/2\}$
```

The reliable node set $\mathcal{R}_i$ of a node i identifies the set of users that will receive (with high probability) any post re-posted by user i. Thus, it can be safely exploited to compute the *global engagement potential* for post t when recommended to user i. This is defined as the sum of of relevance factors r_{jt} over all nodes j belonging to $\mathcal{R}_i$:

$$E_{it} = \sum_{j \in \mathcal{R}_i} r_{jt} \, . \tag{1}$$

4.2 The diffusion-aware recommendation problem

The ultimate goal of the proposed DifRec recommender system is to optimize the targeted recommendation list of posts $\mathcal{L}_i$ for each user i through exploiting the *anticipated diffusion* of posts through the social-network graph, where the social diffusion is assumed to take place according to the diffusion model presented in the previous subsection. In the following we devise an optimization problem that captures this aspect and whose solution can give the desired diffusion-aware recommendations.

Neighborhoodness graph. Given a social-network graph $G = (V, E)$ and the reliable node sets $\mathcal{R}_i$ for every node $i \in V$, we define the (directed) *neighborhoodness graph* $G_n = (V_n, E_n)$ as follows. The node set of G_n corresponds to the node set of G, i.e., $V_n = V$, while a directed edge $(i, j) \in V_n$ exists between users $i, j \in V$ if and only if node j belongs to the reliable set $\mathcal{R}_i$ of node i.

The neighborhoodness graph is aimed at modeling the conflicts of assigning a post to a user if she will receive it anyway via re-posting according to the diffusion model at hand. In other words, if an edge (i, j) exists in the neighborhoodness graph, then all posts generated/re-posted by i will reach j via re-posting, thus meaning that a post t recommended to i should not be recommended to j too as it will get to j anyway. As it will become clear in the sequel, this is exactly the rule that our DifRec follows to make recommendations: the same post is never recommended to two nodes that are adjacent in the neighborhoodness graph.

Problem statement. The recommendation of post t to user i is denoted by a variable x_{it} that is defined as follows:

$$x_{it} = \begin{cases} 1, & \text{if post } t \text{ is recommended to user } i \\ 0, & \text{otherwise}\,. \end{cases}$$

Let $\mathbf{x} = (x_{it} : i \in \mathcal{U}, t \in \mathcal{S})$ denote the $(N \times |\mathcal{S}|)$-dimensional vector of zeros and ones denoting the assignment of posts to users. Under the assumptions above, the problem of maximizing total user engagement is formulated as follows.

Problem 1. *Given a social-network graph $G = (V, E)$, a neighborhoodness graph $G_n = (V_n, E_n)$, and a set of posts $\mathcal{S}$, find*

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} \sum_{i=1}^{N} \sum_{t \in S} E_{it} x_{it} \qquad (2)$$

subject to:

$$\sum_{t \in S} x_{it} = K\,, \qquad (3)$$

$$x_{it} + x_{jt} \leq 1, \forall\,(i, j) \in E_n, \forall t\,. \qquad (4)$$

Constraint (3) captures the recommendation space limit of K posts to be assigned to each user. Constraint (4) ensures that the same post is not assigned to any two nodes that are linked with an edge in the neighborhoodness graph.

We next prove that the problem is NP-hard.

Theorem 1. *Problem 1 is NP-hard.*

Proof. Given a neighborhoodness graph G_n, an independent set $\mathcal{I}$ on G_n is a subset of vertices so that no two vertices in $\mathcal{I}$ are adjacent in G_n. If nodes are assigned weights (for a given post t, Problem 1 assigns weight E_{it} to each vertex $i \in V_n$), a *Maximum-Weighted Independent Set $\mathcal{I}_t$* is an independent set exhibiting maximum sum of node weights, and no pair of vertices i, j in $\mathcal{I}_t$ are adjacent in G_n.

Consider Problem 1 for just one post t. The objective is to find the subset $\mathcal{I}_t \subseteq V_n$ of users i in the neighborhoodness graph G_n to assign the post to (i.e., $x_{it} = 1$), so that the total weight $\sum_{i \in \mathcal{I}_t} E_{it} x_{it}$ is maximized, and no pair of nodes in $\mathcal{I}_t$ are linked with an edge. This exactly corresponds to the Maximum-Weighted Independent Set problem (MWIS). The MWIS problem is an extension of the Maximum-Independent-Set (MIS) problem in the sense of relaxing the assumption that all edge weights are equal to one. The MIS problem in a graph G is NP-Hard through its equivalence with the Maximum-Clique problem in the complementary graph of G [35]. As a result, Problem 1 is NP-Hard as well. □

Algorithm 2 Algorithm $IS(G_n, t)$: Heuristic for generating a weighted Independent Set of users from neighborhoodness graph $G_n = (V_n, E_n)$ for a post t.

```
input: Neighborhoodness graph $G_n$, post $t$, engagement weight $E_{it}$ $\forall i \in V_n$.
output: An Independent Set $\mathcal{I}_t$ for post $t$ of weight $W(\mathcal{I}_t)$. Post $t$ is recommended to users $i \in \mathcal{I}_t$.

Initialization: $\mathcal{I}_t \leftarrow \emptyset$ ; $W(\mathcal{I}_t) = 0$ ; $V'_n \leftarrow V_n$ ; $E'_n \leftarrow E_n$.
Sort nodes $i$ in decreasing order of $E_{it}/(1 + d_i)$.

while $V'_n \neq \emptyset$ do
    Examine nodes $i \in V'_n$ in the order they are sorted above.
    if $\mathcal{I}_t \cup \{i\}$ is an independent set for graph $G_n(V'_n, E'_n)$ then
        $\mathcal{I}_t \leftarrow \mathcal{I}_t \cup \{i\}$
        $W(\mathcal{I}_t) \leftarrow W(\mathcal{I}_t) + E_{it}$
        $V'_n \leftarrow V'_n \setminus \{i\} \setminus \mathcal{F}_i$
        $E'_n \leftarrow E'_n \setminus \{(i, j) \in E_n : j \in \mathcal{F}_i\}$
    end if
end while
```

4.3 Heuristic algorithm

Given that Problem 1 is NP-hard, we aim at finding a suboptimum by designing efficient yet accurate heuristics. In particular, we propose a two-step algorithm that works as follows. We consider posts to be recommended one at a time and, for each post t, we first employ a classic greedy algorithm (Algorithm 2) so as to find an independent set of users in the neighborhoodness graph. This set represents the users to whom post t can be recommended without violating constraint 4 in Problem 1. In the second step, a further greedy algorithm (Algorithm 3) takes the output of Algorithm 2 and fill the recommendation lists $\mathcal{L}_i$ of the various users accordingly.

The algorithm we employ to find an independent set $IS(G_n, t)$ in the neighborhoodness graph G_n for a given post t is a classic greedy algorithm that works as follows. Nodes i are ordered in decreasing order of $E_{it}/(1+d_i)$, where d_i is the out-degree of node i, and they are included in the independent set in that order. When a node i is included in the independent set, we remove from the graph the outgoing links from node i as well as nodes in $\mathcal{F}_i$, i.e., i's out-neighbors. The rationale is that this way priority is given to: (i) nodes exhibiting large weight, that is, users for which recommendation of post t would lead to large diffusion effect, i.e., large total relevance for users that receive the post through re-post diffusion, and (ii) nodes with small out-degree, which do not incur a large set of conflicting neighbor nodes to which the

Algorithm 3 Algorithm *Recommend*: Heuristic algorithm for post recommendation to users.

```
input: Set $\mathcal{U}(= V_n)$ of users, set $\mathcal{S}$ of posts, neighborhoodness graph $G_n$, Engagement metric $E_{it}$ $\forall i \in V_n, \forall t \in \mathcal{S}$.
output: Recommendation of a set $\mathcal{L}_i$ of $K$ posts to each user $i$.

Initialization:
for $i \in V_n$ do
    $\mathcal{L}_i \leftarrow \emptyset$
end for
$V_n' \leftarrow V_n$ ; $\mathcal{S}' \leftarrow \mathcal{S}$
for $t \in \mathcal{S}$ do
    $IS(G_n, t)$
end for

Sort posts $t$ in decreasing order of IS weight $W(\mathcal{I}_t)$
while $V_n' \neq \emptyset$ and $\mathcal{S}' \neq \emptyset$ do
    Consider posts $t$ in the order they are sorted above
    for $i \in \mathcal{I}_t$ do
        $\mathcal{L}_i \leftarrow \mathcal{L}_i \cup \{t\}$
        if $|\mathcal{L}_i| = K$ then
            $V_n' \leftarrow V_n' \setminus \{i\}$
        end if
    end for
    $\mathcal{S}' \leftarrow \mathcal{S}' \setminus \{t\}$
end while
```

same post cannot be assigned. Nodes with small out-degree would therefore lead to high post reuse, i.e., recommendation of the same post in the graph. The algorithm above is referred to as Algorithm $IS(G_n, t)$ and its pseudocode is reported as Algorithm 2.

In the second stage, we need to find a set of K posts, $\mathcal{L}_i$, to be recommended to each user i. After running the $IS(G_n, t)$ algorithm we obtain an independent set $\mathcal{I}_t$ for each post $t \in \mathcal{S}$. Next, we sort posts in decreasing order of their weight $w(\mathcal{I}_t)$, and we assign posts to users in that order. Whenever for some user j we reach $|\mathcal{L}_j| = K$ assigned posts, we do not further consider this user for recommending posts. The procedure stops when each user has K recommended posts or when all posts (independent sets) are parsed. The outline of the overall recommendation algorithm is reported as Algorithm 3.

The overall time complexity is $\mathcal{O}(|\mathcal{S}|(|V_n|+|E_n|+\log|\mathcal{S}|))$, which is determined by, (i) running the IS procedure for each post $t \in \mathcal{S}$, which takes time linear in the size of the neighborhoodness graph, and (ii) sorting the posts in $\mathcal{S}$.

5. EXPERIMENTS

In this section we present the experiments we conducted in order to evaluate the performance of the proposed DifRec recommender system.

5.1 Dataset

We use real data coming from Tumblr, where users can generate content, can get connected to other users, and have the possibility to see the most recent posts of the users connected to them.[2]Particularly, the dataset consists of (i) a social-network graph that models the follower relationship among users, and (ii) posts and re-posts of the various users. Table 2 reports some public statistics.We refer the interested reader to [37] for more in deep statistics about the Tumblr social network, posts, the diffusion behavior and the similarity with other similar services such as Twitter or Facebook.

Table 2: Main characteristics of the dataset used in our evaluation.

Number of posts per user	26.17
Mean post length	421.01 Bytes
Median post length	84.28 Bytes
Number of unique users in induced graph	1.5 M
Mean indegree neighboorhoodness graph	8.70
Mean outdegree neighboorhoodness graph	10.82

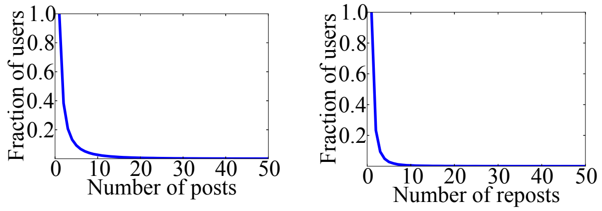

Figure 2: Post and re-post normalized distribution in our dataset.

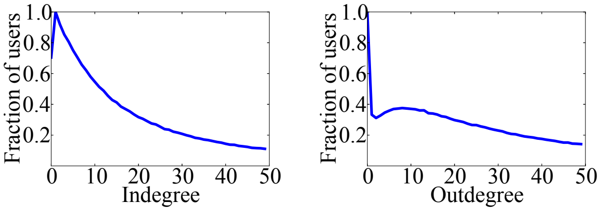

Figure 3: Normalized in-degree and out-degree distribution of our induced social graph.

We randomly sample a set of 1.5M users among users that have made at least one post/re-post during the time period from October 1st, 2013 until November 8th, 2013. As a social network, we therefore consider the subnetwork induced by this sample of users. For each user in our sample, we also extract all her public posts made during that time period. For each post, we focus on these fields: author, content, and reference to the original post (the latter clearly only if the post has been re-posted). We use the posts from October 2013 to build user profiles, posts from November 1–7 2013 to construct the neighborhoodness graph, and posts generated on November 8th for evaluation purpose (i.e., as posts to be recommended).

Figure 2 shows the normalized distribution of posts and re-posts activity in our dataset, while Figure 3 depicts the in-degree and out-degree distribution of the induced social network.

[2] The term "post" refers here to a generic piece of information.

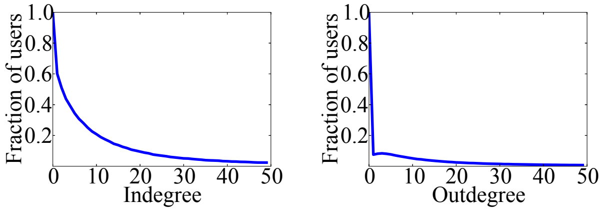

Figure 4: Normalized in-degree and out-degree distribution of our neighboorhoodness graph.

5.2 Setup

Relevance factor. We compute the relevance factor r_{it} of a post t to user i (Section 3) based on the well-established *set similarity measure* [25], which is define as a convex combination between a single-term-based score and a bi-gram-based score derived from the set of all posts that constitute the "profile" of a user. As shown in [25], the similarity set measure is strongly correlated to the actual interests of a specific user, thus making it an effective way of assessing how relevant a post is to that user.

Diffusion model. We experiment with the diffusion model described in Section 4.1. The probability w_i of re-posting of user i is computed as the percentage of re-posts of user i among all the posts she has seen in the time-window considered, that is, all the posts that have been posted or re-posted by any of the users followed by i. After analyzing the distribution of values of w_i, we set the threshold η that defines the reliable set $\mathcal{R}_i$ of user i to 0.02, as a good trade-off between stability of the model and size of the resulting neighborhoodness graph.

With this setting, the neighborhoodness graph that we get has 1.5M nodes (users), average in-degree 10.8, and average out-degree 8.7. Figure 4 depicts the normalized in-degree and out-degree distribution of the resulting neighborhoodness graph.

Baselines. We compare the proposed DifRec to the following baselines:

- *Real recommender system* (Real): this is the real recommender system embedded in the online social/micro-blogging platform we consider in our evaluation.[3]
- *Content-based recommender system* (Content): this approach is based on the estimation of the relevance factor of a post t to user i by considering only the profile of i [25]. In particular, as suggested in [25], for each user we define her profile by selecting the posts that are most relevant to her interests.
- *Social-aware recommender system* (Social): this is an extension of the Content recommender system where also the interests of i's friends are considered to construct her profile [25].

Metrics of performance assessment. We evaluate the performance of our method and the competitors in terms of the average *overall user engagement*, which is defined based on the relevance factor discussed in Section 3 and the reliable sets $\mathcal{R}_i$ used in the diffusion model described in Section 4.1. As explained in the previous sections, the overall user engagement metric gives direct evidence of the quality of the recommendation.

The idea is to compute, for each user i, the sum of the relevance factors r_{it} for all the posts t that are explicitly recommended to i by the recommendation method at hand, plus the sum of the relevance factors $r_{it'}$ for all the posts t' that reach user i through diffusion according to the aforementioned diffusion model. As the number of posts ultimately recommended to any user can vary from a user to another and from a recommendation method to another (K may be only an upper bound to the number of recommended posts), we average these two sums over the number of posts recommended. The ultimate metric we consider is the average of such a quantity over all users in the network.

Formally, for a user i, let $\mathbf{R}_i$ denote the set of users whose corresponding reliable set contains user i (excluding i herself), i.e., $\mathbf{R}_i = \bigcup_{j\in V} \mathrm{I}[i \in \mathcal{R}_j] \setminus \{i\}$ (where $\mathrm{I}[\cdot]$ denotes the indicator function, which is equal to one if the argument is true, zero otherwise). The (average) user engagement $E(\mathbf{L}, i)$ of a user i given a recommendation $\mathbf{L} = \{\mathcal{L}_i\}_{i\in V}$ is defined as the sum of two terms: the relevance factor of the posts explicitly recommended to her (i.e., the posts in her recommendation list $\mathcal{L}_i$), and the relevance factor of the posts coming to her through diffusion (i.e., the posts present in the recommendation lists of all the users in $\mathbf{R}_i$); the sum of these two terms is then averaged by the size of the i's recommendation list, so as to take into account that recommendation lists may in general have variable size. Formally:

$$E(\mathbf{L}, i) = \frac{1}{|\mathcal{L}_i|}\left(\sum_{t\in\mathcal{L}_i} r_{it} \quad + \sum_{t'\in\bigcup_{j\in\mathbf{R}_i}\mathcal{L}_j} r_{it'}\right),$$

The (average) overall user engagement of a recommendation $\mathbf{L}$ is eventually computed as:

$$E(\mathbf{L}) = \frac{1}{|V|}\sum_{i\in V} E(\mathbf{L}, i). \tag{5}$$

5.3 Results

Here we present the results of our evaluation. We provide two different evaluations:

- First, we test our method along with the aforementioned baselines, on the posts that have been recommended to the selected users by the *real recommender system*. This gives us the opportunity to verify that, by reassigning those posts according to our method, we are able to increase overall user engagement with respect to recommendations provided in reality. Specifically, we select the set of posts for this experiment as follows. Among all posts recommended to the selected users by the real recommender system, we take the ones that have been re-posted by the user to which the post has been recommended; this way, we are sure that the recommendation was really effective for that user. Among all these posts, we finally select a random sample of 5K posts.
- In order to avoid being biased by the set of posts selected by the real recommender system, we also repeat the evaluation on a different set of posts, i.e. a set of 5K posts randomly selected from the most recent ones (re-)posted by the selected users. In this case, we clearly cannot compare to the real recommender

[3] The real recommender system combines techniques from content-based and social-based recommender systems. The exact implementation details constitute sensitive information that cannot be disclosed.

Table 3: Comparison to the reality: (average) overall user engagement results of our method, the real recommender system, and the baselines on the set of posts recommended by the real recommender system, with varying the number K of posts recommended to each user. Engagement values are shown averaged by number of posts recommended and by the number of users who received a recommendation. The last line shows the % gain of our method with respect to the best one of the other competing methods.

K	1	5	10	15	20	25
Content	2 444	2 368	2 298	2 242	2 196	2 156
Social	4 395	4 117	3 912	3 753	3 619	3 498
Real	3 385	3 395	3 404	3 399	3 398	3 398
DifRec	10 080	11 414	10 831	10 504	10 421	10 185
	+129%	+177%	+177%	+180%	+188%	+191%

Table 4: Comparison on general posts: (average) overall user engagement results of our method and the baselines on the set of most recent posts, with varying the number K of posts recommended to each user. Engagement values are shown averaged by number of posts recommended and by the number of users who received a recommendation. The last line shows the % gain of our method with respect to the best one of the other competing methods.

K	1	5	10	15	20	25
Content	13 532	11 356	10 569	10 112	9 777	9 520
Social	11 012	9 077	8 267	7 732	7 310	6 971
DifRec	18 711	20 947	23 328	23 436	23 771	24 643
	+38%	+84%	+121%	+132%	+143%	+159%

system (as the set of posts is different from the ones selected by that system); we therefore only compare with the other baselines Content and Social.

We report the details of both the evaluations next.

Table 3 reports the average overall user engagement of our DifRec compared to the real recommender system and the two other baseline recommender system described above. The set of posts considered correspond to the posts recommended by the real recommender system.

Our method turns out to outperform all competitors at least by a factor that ranges from 129% to 191% for different values of K. This implies that DifRec recommends K posts that are most relevant and interesting to the user on average. These K recommended posts are most often different from the ones that will anyway reach the user through re-posts diffusion. Another observation is that the average overall user engagement for all methods is increasing as K gets larger, but only for small/moderate values of K, i.e., up to $K = 5$ or $K = 10$. For larger values of K ($K = 10, 15, 20, 25$), the average engagement decreases. This can be explained by the fact that, as K increases, the relevance factors of recommended posts decrease (cf. Equation (5)).

The average overall user engagement of our method increases up to $K = 5$ and slightly decreases by $5 - 10\%$ for larger values of K. However, despite that slight decrease in average engagement, the percentage benefit of DifRec compared to other methods increases from 129% or 177% up to 191% as K increases. This implies that our method tends to efficiently exploit the recommendation space better by recommending posts other than the ones that will reach the user through repost diffusion.

Similar performance benefits can be observed when comparing with two baselines over a more general, randomly drawn dataset in Table 4.

More details about both comparisons are shown in Tables 5 and 6, where we report cumulative distributions about percentage of users having (average) overall user engagement larger than various values. It can be observed that the distribution exhibited by the proposed DifRec is above the ones of the other competitors: for every value of user engagement, the proposed DifRec exhibits percentage values larger than all the competitors, thus further attesting its superiority.

6. CONCLUSIONS

In this work we propose DifRec, a novel recommender system that aims at increasing the overall user engagement in online social platforms by taking into account the social-network diffusion process. In particular, we take into account the fact that, when a user shares or re-posts a recommended item, her followers indirectly receive the post. Considering this aspect, that, to the best of our knowledge, has never been considered by any existing recommender systems, may allow for better selecting the items to be recommended to a user, by avoiding for instance to recommend to her an item that would have reached her anyway via re-posting.

We evaluate the proposed DifRec recommender systems by using real data coming from the Tumblr platform. We show that our DifRec improves upon the overall user engagement of the real recommender system embedded in that platform and various other existing recommender systems up to a factor of 130–190%.

Acknowledgement

This work was supported by the LEADS project (ICT-318809) funded by the European Community.

7. REFERENCES

[1] R. Agrawal, S. Gollapudi, A. Halverson and S. Ieong "Diversifying Search Results". In *WSDM*, 2009.

[2] E. Diaz-Aviles, L. Drumond, L. Schmidt-Thieme and W. Nejdl "Real-Time Top-N Recommendation in Social Streams". In *RecSys*, 2012.

[3] F. Altarelli, A. Braunsten, L. Dall' Asta and R. Zecchina, "Optimizing spread dynamics on graphs by message passing". In *Journal of Statistical Mechanics: Theory and Experiment*, 2013.

[4] A. Goyal, F. Bonchi, L. V. S. Lakshmanan, "A Data-Based Approach to Social Influence Maximization". In *PVLDB*, 2011.

Table 5: Percentage of users having (average) overall user engagement larger than a certain value: results of our method, the real recommender system, and the baselines, on the set of posts recommended by the real recommender system ($K = 10$ and $K = 20$).

engagement	$K = 10$				$K = 20$			
(Equation (5))	Content	Social	Real	DifRec	Content	Social	Real	DifRec
> 4000	69.80%	78.27%	69.61%	**78.56%**	69.54%	78.11%	69.73%	**78.67%**
> 6000	67.49%	76.06%	66.77%	**76.47%**	67.22%	75.89%	66.88%	**76.57%**
> 8000	65.95%	74.60%	65.00%	**74.98%**	65.68%	74.42%	65.10%	**75.07%**
> 10000	64.84%	73.51%	63.78%	**73.89%**	64.56%	73.33%	63.85%	**73.97%**

Table 6: Percentage of users having (average) overall user engagement larger than a certain value: results of our method, and the baselines, on the set of most recent posts ($K = 10$ and $K = 20$).

engagement	$K = 10$			$K = 20$		
(Equation (5))	Content	Social	DifRec	Content	Social	DifRec
> 4000	64.23%	72.87%	**73.58%**	63.96%	72.75%	**73.39%**
> 6000	59.92%	68.77%	**69.73%**	59.65%	68.65%	**69.54%**
> 8000	56.91%	65.91%	**67.02%**	56.65%	65.78%	**66.86%**
> 10000	54.66%	63.75%	**64.99%**	54.41%	63.62%	**64.85%**

[5] G. Capannini, F. M. Nardini, R. Perego and F. Silvestri "Efficient diversification of web search results". In *PVLDB*, 2011.

[6] W. Chen, C. Wang, and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks". In *KDD*, 2010.

[7] K. Chen, T. Chen, G. Zheng, Ou Jin, E. Yao, and Y. Yu, "Collaborative personalized tweet recommendation". In *SIGIR*, 2012.

[8] W. Galuba, K. Aberer, D. Chakraborty, Z. Despotovic, W. Kellerer, "Outtweeting the twitterers-predicting information cascades in microblogs". In *WOSN*, 2010.

[9] M. Gomez-Rodriguez, D. Balduzzi, B. Schoelkopf, "Uncovering the temporal dynamics of diffusion networks". In *ICML*, 2011.

[10] M. Gomez-Rodriguez, J. Leskovec, and A. Krause, "Inferring networks of diffusion and influence". In *KDD*, 2010.

[11] A. Gionis, E. Terzi and P. Tsaparas "Opinion maximization in social networks". In *SDM*, 2013.

[12] A. Guille, H. Hacid, "A predictive model for the temporal dynamics of information diffusion in online social networks". In *WWW Companion*, 2012.

[13] A. Guille, H. Hacid, C. Favre, D. Zighed, "Information Diffusion in Online Social Networks: A Survey". In *SIGMOD Record*, 2013.

[14] X. Yang, Y. Guo and Y. Liu, "Bayesian-inference based recommendation in online social networks". In *IEEE Trans. on Parallel and Distr. Systems*, 2013.

[15] Y. Koren, R. Bell and C. Volinsky, "Matrix factorization techniques for recommender systems". In *IEEE Computer*, 2009.

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

[17] M. Kimura and K. Saito, "Tractable models for information diffusion in social networks". In *PKDD*, 2006.

[18] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks". In *KDD*, 2007.

[19] S. Myers and J. Leskovec, "The Bursty Dynamics of the Twitter Information Network". In *WWW*, 2014.

[20] M. Mohite and Y. Narahari "Incentive Compatible Influence Maximization in Social Networks and Application to Viral Marketing". In *AAMAS*, 2011.

[21] J. Ok, Y. Jin, J. Shin and Y. Yi, "On Maximizing diffusion speed in social networks: Impact of random seeding and clustering". In *Sigmetrics*, 2014.

[22] A. Rajaraman and A.D. Ullman, "Mining of Massive Datasets". *Cambridge University Press, 2012.*

[23] Y. Shia, X. Zhaob, J. Wangb, M. Larsona and A. Hanjalic "Adaptive Diversification of Recommendation Results via Latent Factor Portfolio". In *SIGIR*, 2012.

[24] K. Saito, K. Ohara, Y. Yamagishi, M. Kimura, H. Motoda, "Learning diffusion probability based on node attributes in social networks". In *ISMIS*, 2011.

[25] M. Pennacchiotti, F. Silvestri, H. Vahabi, R. Venturini, "Making your interests follow you on Twitter". In *CIKM*, 2012.

[26] W. Shen, J. Wang, P. Luo, M. Wang, "Linking Named Entities in Tweets with Knowledge Base via User Interest Modeling". In *KDD*, 2013.

[27] L. Hong, A. Doumith, B. Davison, "Co-Factorization Machines: Modeling User Interests and Predictiong Individual Decisions in Twitter". In *WSDM*, 2013.

[28] B. Chandramouli, J.J. Levandovski, A. Elwady and M.F. Mokbel, "StreamRec: A real-time recommender system". In *SIGMOD*, 2011.

[29] C. Chen, H. Yin, J. Yao, B. Cui, "TeRec: A Temporal Recommender System Over Tweet Stream", In *Proceedings of the VLDB Endowment*, 2013.

[30] Z. Ren, S. Liang, E. Meij, M. de Rijke, "Personalized Time-Aware Tweets Summarization". In *SIGIR*,2013.

[31] Y. Pan, F. Cong, K. Chen, Y. Yu, "Diffusion-aware Personalized Social Update Recommendation". In *RecSys*, 2013.

[32] B. Wang, C. Wang, J. Bu, C. Chen, W. V. Azhang, D. Cai, X. He, "Whom to Mention: Expand the Diffusin of Tweets by @ recommendation on Micro-blogging Systems". In *WWW*, 2013.

[33] C. J. Colbourn, The combinatorics of network reliability. *Oxford University Press*, 1987.

[34] J.S. Provan and M.O. Ball, "The complexity of counting cuts and of computing the probability that a graph is connected". In *SIAM J. Comput.*, 1983.

[35] M.R. Garey and D.S. Johnson, Computers and intractability: A guide to the theory of NP-Completeness. *Freeman, New York*, 1979.

[36] A. Khan, F. Bonchi, A. Gionis, F. Gullo, "Fast Reliability Search in Uncertain Graphs". In *EDBT*, 2014.

[37] Y. Chang, L. Tang, Y. Inagaki, Y. Liu, "What is Tumblr: a statistical overview and comparison.", In *SIGKDD Explor. Newsl.*, 2014.