Patterns of Influence in a Recommendation Network

Jurij Leskovec[1], Ajit Singh[1], and Jon Kleinberg[2]

[1] Center for Automated Learning and Discovery
Carnegie Mellon University
{jure, ajit}@cs.cmu.edu
[2] Department of Computer Science
Cornell University
kleinber@cs.cornell.edu

Abstract. Information cascades are phenomena whereby individuals adopt a new action or idea due to influence by others. As such a process spreads through an underlying social network, it can result in widespread adoption overall. We consider information cascades in the context of recommendations, and in particular study the patterns of cascading recommendations that arise in large social networks. We investigate a large person-to-person recommendation network, consisting of four million people who made sixteen million recommendations on half a million products. Such a dataset allows to pose a number of fundamental questions: What cascades arise frequently in real life? What features distinguish them? We enumerate and count cascade subgraphs on large directed graphs; as one component of this, we develop a novel efficient heuristic based on graph isomorphism testing that scales to large datasets. We discover novel patterns: the distribution of cascade sizes and depths follows a power law. Generally, cascades tend to be shallow, but occasional large bursts of propagation can occur. Cascade subgraphs are mainly tree-like, but we observe variability in connectivity and branching across recommendations for different types of products.

1 Introduction

The social network of interactions between a group of individuals plays a fundamental role in the spread of information, ideas, innovation, and influence among its members. The network effect has been observed in many cases, where an idea or action gains sudden widespread popularity through word of mouth or viral

Work partially supported by the National Science Foundation under Grants No. IIS-0209107, SENSOR-0329549, IIS-0326322, CNS-0433540, CCF-0325453, IIS-0329064, CNS-0403340, CCR-0122581, a David and Lucile Packard Foundation Fellowship, and also by the Pennsylvania Infrastructure Technology Alliance (PITA). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, or other funding parties.

marketing. For example, some movies become widely popular through word-of-mouth advertising. Google's Gmail service captured a significant market share in spite of the fact that up to recently the *only* way to obtain a free email account is through a referral. One can also find many examples in weblogs (blogs), where a story or piece of information gets widely referred to by the blogger community and is eventually picked up by the mass media.

Information cascades are phenomena where an action or idea becomes widely adopted due to influence by others, as opposed to individual reasoning in isolation [4]. Cascades are also known as "fads" or "resonance." There has been significant work done in modeling the spread and adoption of ideas and influence through a social network. Models of node influence have been proposed [7, 8] and algorithms for choosing influential nodes have been developed [6, 11, 16].

The formalism for cascades is activation of nodes in a graph where nodes represent individuals, edges relationships, and a binary node state shows whether a person is part of the cascade. The chance that a node is activated is influenced by the state of its neighbors. A related formalism is a graph where the nodes are agents and a directed edge (i, j, t) indicates that i influenced j at time t.

Cascades have been studied for many years by sociologists concerned with the *diffusion of innovation* [17]; more recently, researchers in several fields have investigated cascades for the purpose of selecting trendsetters for viral marketing [16, 6], finding inoculation targets in epidemiology [15], and explaining trends in blogspace [2, 3, 9, 12]. To our knowledge, however, the difficulty in obtaining data has limited the extent of analysis on large-scale, complete datasets representing cascades. Here we look at the patterns of influence in a large-scale, real recommendation network and examine the topological structure of cascades.

Here we ask the question: What cascades arise frequently in real life? Are they like trees, stars, or something else? We describe a large person-to-person recommendation network, consisting of 4 million people who made 16 million recommendations on half a million products in section 3. To analyze the data, we first create graphs where incoming edges influenced the creation of outgoing edges. We remove edges that violate the temporal requirement of a cascade (*i.e.*, influence must be exerted before the effect). Then, we enumerate and count all possible cascade subgraphs using an algorithm developed in section 4. Therein, we propose a heuristic for graph isomorphism involving the degree distribution and the eigenvalues of the adjacency matrix that scales to large datasets. We apply the algorithm to the recommendation dataset, and analyze it in section 5.

We find novel patters and the analysis of the results gives us insight into the cascade formation process. We find that distribution of sizes and depths of cascades follows a heavy-tailed distribution. Generally cascades are shallow but occasional large bursts also occur. The cascade sub-patterns reveal mostly small tree-like subgraphs; however we observe differences in connectivity and the shape of cascades across product groups. We find common cases when people who do not link to each other nevertheless recommend to the same set of friends; and cases where recommendation propagates but returns to target the same people.

2 Related work

To our knowledge, this is the first large-scale study of cascades in a real recommendation network. We believe the lack of prior studies is due to the difficulty in acquiring large networks without link ambiguity from a real-world scenario.

Most work on extracting cascades has been done in the blog domain [1, 3, 9, 12]. The authors in this domain note that, while information propagates between blogs, examples of genuine cascading behavior appeared relatively rare. This is possibly due to bias in the web-crawling and text analysis techniques used to collect pages and infer relationships. In our dataset, all the recommendations are stored as database transactions, and we know no records are missing. Associated with each recommendation is the product involved, and the time the recommendation was made. Studies of blogspace either spend a lot of effort mining topics from posts [2, 3, 9] or consider only the properties of blogspace as a graph of unlabeled URLs [1, 12]. Temporally evolving graphs are explored in [5].

The theoretical behavior of cascades on random graphs is studied in [19]. Unlike this paper, the underlying network is known. An alternative to explicit thresholding are probabilistic models of node activation [6, 11, 16].

In our work we need to efficiently enumerate and count cascade subgraphs. The problem of graph isomorphism plays a key role in frequent subgraph mining. Much work has dealt with frequent subgraph mining [10, 13, 20, 18, 21]; however this body of work mostly assumes that graphs are richly labeled and undirected. This is suitable for mining chemical compound and richly labeled bio-informatics datasets. Our problem here differs in that we have directed and unlabeled graphs, where we are interested in purely topological structure of the subgraphs. So the clever pruning of search space using node and edge labels can not be applied in our case. We also have additional temporal constraints on cascades, so blind enumeration of frequent subgraphs will not work. For these reasons enumerating all possible sub-graphs may be computationally infeasible and nonexistence of labels means too many expensive graph isomorphism tests. So we take advantage of the specific problem domain and develop efficient algorithms for graph subpattern generation and approximate isomorphism testing.

3 The recommendation network

We study a recommendation network dataset from a large on-line retailer. At this retailer, each time a person purchases a book, music, DVD, or video tape he or she is given the option of sending an email recommending the item to his friends. The first recipient to purchase the same item receives a discount, at which time the sender receives a referral credit. Note that a person can make recommendations on a product only after she actually purchased it. Since each sender has an incentive for making effective referrals, it is natural to hypothesize that this dataset is a good source of cascades.

Each recommendation is annotated with the time it was sent, whether it resulted in a purchase, the date of purchase (if applicable), and what product

Group	p	n	e	e_u	b_t	b_r
Book	103,161	2,863,977	5,741,611	2,097,809	2,859,096	83,113
DVD	19,829	805,285	8,180,393	962,341	837,300	75,421
Music	393,598	794,148	1,443,847	585,738	712,673	10,576
Video	26,131	239,583	280,270	160,683	165,109	1,376
Full network	542,719	3,943,084	15,646,121	3,153,676	4,574,178	170,486

Table 1. Product group recommendation network statistics: p: number of products, n: number of nodes, e: total number of edges (recommendations), e_u: number of unique edges, b_t: total number of purchases, b_r: purchases made through recommendations.

was recommended. Customer information is anonymized, no demographic or uniquely identifying information is available.

We represent this relational dataset as a directed multigraph: nodes represent customers, and a directed edge (i, j, p, t) means that node i recommended product p to customer j at time t. The typical edge generation process is as follows: a node (person) i first buys product p at time t, and then recommends it to nodes $\{j_1, \ldots, j_n\}$. The j nodes can then buy the product (with the option to recommend it to others). Note that even if all nodes j buy the product, only the first purchaser will get the discount, which is marked by a purchase flag (*buy-bit*). We cannot directly use the buy-bit to determine whether a recommendation caused a purchase. In addition to the buy-bit, we also record the number of of customers who recommended the product (since they had to buy the product to recommend it).

The recommendation network consists of 15,646,121 recommendations made among 3,943,084 distinct users from June 2001 to May 2003 (711 days). Network is over 1 Gb in size. A total of 542,719 different products belonging to four product categories (Books, DVDs, Music and Videos) were recommended.

We extract per-group recommendation networks by taking the edge-induced subgraph formed by all the products of a given category. Table 1 describes the four per-group networks. The DVD network contains the most recommendations; but the book network involves more customers. On average a node in the DVD network made more than 10 recommendations; on average a book or music node made about two recommendations. For video nodes this drops to about one recommendation per node. While music recommendations involved almost as many customers as DVD recommendations, far fewer music recommendations were required to receive about the same coverage of the nodes (people).

There can be multiple recommendations between the nodes, so by counting only unique edges (e_u), we see that only DVDs have more edges than nodes. This means that all networks are very sparsely linked and that existing users heavily participated in the program (exchanged multiple recommendations), while the exploration of the social network was rather poor. At the end of the two year period, the largest connected component contained fewer than 2.5% of the nodes.

The last two columns of table 1 show the total number of purchases (b_t) and the purchases that resulted from a recommendation (b_r). Observe that for DVDs

9% of purchases could be attributed to recommendations, for books 3%, music 1.5% and video less than 1%. Comparing nodes to purchases, we see there is about one purchase per node.

All this indicates that people like to recommend and purchase DVDs. On the other hand, book recommendations seem to be more effective. On average there were two recommendations to each purchase in the book network, but almost 10 recommendations to each purchase in the DVD network. While book recommendations appear very influential, most readers do not appear to make many of them. Moreover, the DVD network is much denser than the book network.

4 Proposed method

In this section we present the algorithms and techniques developed to efficiently enumerate and count frequent recommendation patterns in a large graph, including a heuristic for subgraph isomorphism.

Ideally one would expect cascades to be trees or near-trees. We soon found out that recommendations create arbitrary graphs: there are multiple recommendations on the same product or multiple products between the nodes, there are multiple purchases of the same product, and one finds many cycles.

To find cascades one first needs to identify cases when incoming recommendations could cause purchases and further outgoing recommendations. Recommendations into node u that precede a purchase can be posited to have influenced the purchase. There are two ways to establish this. If an edge is marked by a purchase flag, we assume the recommendation influenced the purchase. Alternately, the existence of two directed edges (i, j, p, t) and (j, k, p, t') for $t' > t$ suggests cascade behavior. That is, node j receives a recommendation for product p at time t and then makes recommendation for the same product at a later time t'.

First we create a separate graph of recommendations for each product. To find cascades we propose the following two-step procedure:

Delete late recommendations: To keep only recommendations that influenced the purchase we *delete late recommendations*: given a single product recommendation network, for every node we delete all incoming recommendations (edges) that happened after the first purchase of a product. This procedure removes all recommendations of the product a person received after the first purchase. This guarantees that for every node the time of all incoming edges is strictly smaller than the time of all outgoing edges.

Delete no-purchase nodes: Preliminary data analysis showed that the majority of recommendations do not produce cascades. We also observed many star-like patterns where the center node recommends to a large number of people, none of whom purchase the product. This occurs frequently in DVD subgraphs. To prevent this type of large but shallow pattern, we delete all nodes that did not purchase the product.

After deleting late recommendations each connected component corresponds to a cascade. All paths in the component are time-increasing (*i.e.*, a cascade subgraph contains only directed paths with strictly increasing edge times). Deleting no-purchase nodes ensures that we detect only true cascade patterns.

Cascade enumeration: Next we enumerate all possible cascades. In preliminary experiments we first enumerated the maximal cascades, which after the steps described above reduces down to enumerating all connected components of the network. This approach works well and is very fast, but suffers from the fact that the counts are small. Here we take a different approach. Since we are interested in purely topological properties of the cascades, rather than enumerating all possible connected subgraphs, we enumerate all *local cascades.* This means that for every node we explore the cascade in the neighborhood around the node. For every node n, we create a graph induced on nodes up to H hops away from n, where H ranges from 1 up to the distance to the farthest node. One can think of this as exploring node n's neighborhood 1, 2, 3,... steps away. This way for every node we capture the local structure of the cascade around it at various distances.

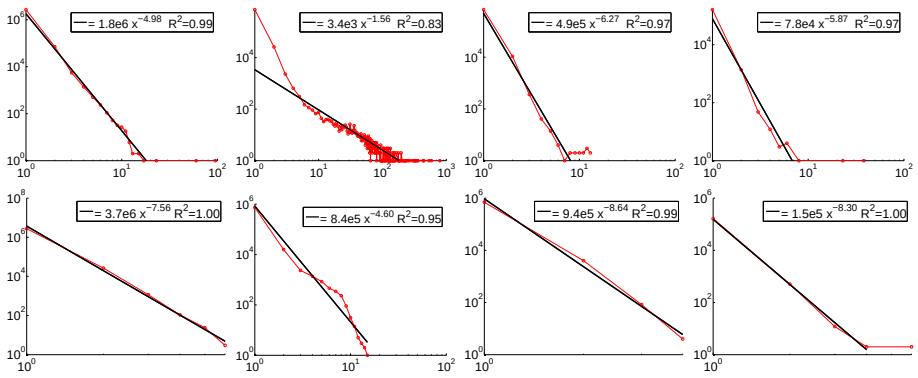
Approximate graph isomorphism: An essential step in counting cascades is determining whether a new cascade is isomorphic to a previously discovered graph. No polynomial-time algorithm is known for the graph isomorphism problem, and so we resort to an approximate, heuristic solution. For each graph we create a *signature*. A good signature is one where isomorphic graphs have the same signature, but where few non-isomorphic graphs share the same signature.

We propose a multi-level approach where the computational complexity (and accuracy) of the graph isomorphism resolution depends on the size of the graph. For smaller graphs we perform an exact isomorphism test; as the size of the graph increases this becomes prohibitively expensive so we use gradually simpler but faster techniques which give only approximate solutions. Another trick is that for each graph we create an efficiently computable signature, use hashing, and then use more expensive isomorphism tests only on graphs with the same signature.

For every graph we create a signature which is composed of the number of nodes, the number of edges, and the sorted in- and out-degree sequence. For graphs with fewer than 500 nodes, we also include the singular values of the adjacency matrix (via singular value decomposition).

We then hash the graphs using the signatures. Additionally, for graphs with fewer than 9 nodes we perform exact isomorphism checking. When the isomorphism check is used, we keep a list of all variants of graphs that collided. Since we first hash, and then check for isomorphism, the number of true isomorphism checks is small.

Note that a small minority of cascades are larger than 9 nodes, so for most of the subgraphs we get the exact solution; as the cascade size increases the number of occurrences decreases, and this is where we make use of an approximate solution.



(a) Book (b) DVD (c) Music (d) Video

Fig. 1. Size and depth distribution of the cascades for the four product groups. Top row shows the size distribution of the cascades (log size of cascade vs. log count). Bottom row shows the distribution of the depths of the cascades (log depth of the cascade vs. log count). Bold line presents a power-fit.

We performed a small set of experiments to evaluate the proposed approximate graph isomorphism algorithm. Given a graph with 8 nodes and 12 edges 100,000 brute-force evaluations of graph-isomorphism took under 40 seconds on a standard desktop. In the second experiment we generated 100,000 random graphs (Erdős-Rényi model), each of them with a randomly chosen number of nodes between 4 to 20 and twice as many edges (average degree of 2). The counting took 50 seconds. In this experiment we observed at most 53 non-isomorphic graphs (5 nodes, 10 edges) with the same signature. At the end the random generation created a total of 6,194 5-node graphs, of which 1,601 were non-isomorphic.

This preliminary analysis shows that we are able to efficiently find and enumerate cascades even in a large recommendation network. The graph isomorphism checking is fast and scalable to serve our purpose.

5 Patterns of recommendation

5.1 Size and depth distribution of cascades

We measure the size of the cascade in terms of the number of nodes and the depth, which is the length of the longest directed path in the cascade. As in all experiments we create per-product recommendation networks, delete late recommendations and no-purchase nodes, and then perform the analysis.

Figure 1 shows the distribution of cascade sizes (top row) and depths (bottom row) for the four product groups. The size of cascades follows a power-law. For books the largest cascade has 95 nodes and 231 edges. For DVDs the largest cascade is eight times larger ($n = 791, e = 5544$). The cascades involving music or videos are much smaller, the largest cascades are $n = 13, e = 56$ and $n = 37, e = 169$ respectively.

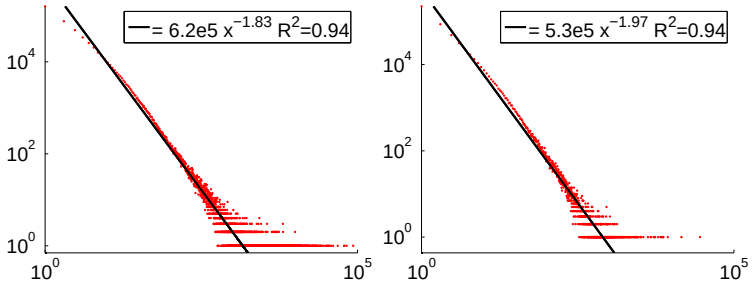


Fig. 2. Distribution of recommendations and purchases over the products: number of recommendations of the product vs. count (left); number of purchases vs. count (right).

The slopes of the power-fits (top row of figure 1) reveal that DVDs had the highest proportion of large cascades, as its power coefficient is the largest. For music the fraction of large cascades is much smaller. While the first part of the size distribution for DVDs (figure 1(b)) has slope -4.5, which is close to the other three product groups, the curve then flattens to -1.5.

The depth distribution, figure 1, shows that cascades are generally shallow except for DVDs. The maximum depth of a cascade is 6 for books, 15 for DVDs, 4 for music, and 6 for videos. So DVDs have the strongest evidence for cascades.

One might posit that cascades are branching processes. However it is known that for a particular run of a branching process, the distribution of depths, conditioned on the size being finite, is exponential. In other words, if cascades were purely branching processes, then the depths should be exponentially distributed. Figure 1 shows that the depth distribution follows a power-law; that is, we are observing more of deep cascades than expected under a branching process.

There are a number of possible explanations for this phenomena: cascades can collide, increasing the probability of success in some part of the social network [14]. Cascade sizes also reflect an underlying power-law in sales frequencies, as shown in figure 2. The number of purchases decays faster than the recommendations. In [14], a stochastic cascade generation process is proposed where the cascade size distribution follows a power law with the exponent -1.

5.2 Frequent cascade subgraphs

What cascades arise frequently in real life? Are they like trees, stars, long chains, or something else? We now explore the building blocks of the cascades, by performing the following procedure. For each product recommendation graph, we first identify cascades (delete late recommendations and no-purchase nodes). Then for each node we create a subgraph on nodes at distance at most h hops, where h varies from 1 up to the value where all nodes in the cascade are reached. We then count the graphs using the approximate graph isomorphism technique described in section 4.

General observations: For books we identified a total of 122,657 cascades, of which 959 are topologically different. There are 213 cascades that occur at least ten times. For DVDs we identified 289,055 cascades, of which 87,614 are topologically different. There are 3,015 cascades that occur at least ten times. For music we identified 13,330 cascades, of which 158 were topologically different. Only 23 cascades occurred at least ten times. Videos contained the least evidence for cascades, with 1,928 subgraphs containing 109 unique patterns. Only 12 subgraphs occurred more than ten times.

The number of cascades concur with observations made from figure 1 and table 1, where DVDs had the largest and richest set of cascades. Since DVDs contain the deepest cascade, there is more opportunity for topological variety than on the other products types. Even though the music network is three times larger than the video network, it does not exhibit much larger topological variety.

Analysis of frequent cascade patterns: Table 2 shows ranks R and frequencies F of 22 cascades for the 4 product groups. Cascades are ordered by size. The table also includes all sub-cascades with at most four nodes and four edges. Interesting, 14 cascade patterns can be observed in all the product groups. Table 2 shows ten of them.

The most common cascade, G_1, represents a single recommendation. This pattern accounts for 70% of all book cascades, 86.4% of all music cascades, 74% of all video cascades, but just 12.8% of DVD cascades. The chain of three nodes (G_3) is the most common depth two cascade, accounting for 4.1% of book cascades, about 3% of video and music cascades, but only 1.8% of DVD cascades. DVD cascades tend to be most densely linked.

Comparing G_2 and G_4 shows that simple splits are more frequent than collisions. For books there are 6.6 times more splits than collisions; for DVDs this factor drops to 1.3; and it is 4.2 and 8.25 for music and videos respectively. Very similar observations hold for splits and collisions on 4 nodes (G_6 and G_{13}); however notice that for DVDs the collision of 3 nodes (G_{13}) is slightly more frequent than the split (G_6). Another such example of reversed graphs are G_7, G_{11} and G_8, G_{12}. Again, the split pattern is more frequent than the collision. The ratio is more unbalanced for books (1 collision per 7 splits) than for DVDs (1 to 2).

Graphs from G_{14} to G_{19} all have a triangle, with one additional node attached. Again, except for DVDs, splits of recommendations (G_{14} and G_{15}) are more frequent than collisions (G_{18}, G_{19}). For DVDs the most frequent sub-graph of the set is G_{18} (a collision), followed by G_{14} and G_{15}.

A common observation is that simpler graphs, like chains and trees, tend to be more frequent in book recommendation networks, while for DVDs we observe richer and more diverse graphs all with relatively high counts.

Figure 3 shows larger graph patterns. Various types of collisions are becoming more frequent. For book cascades G_{27} is very frequent, while a version with reversed edges can only be found in DVDs. Graphs G_{34} and G_{35} are the two largest that can be found in recommendation networks from all 4 product groups. Larger DVD cascades tend to be frequent – G_{35} ranks 18 among DVD cascades.

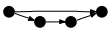
Id	Graph	Nodes	Edges	Book		DVD		Music		Video	
				R	F	R	F	R	F	R	F
G_1		2	1	1	86,430	1	36,863	1	11,518	1	1,425
G_2		3	2	2	10,573	4	3,238	2	492	5	33
G_3		3	2	3	5,089	2	5,147	3	389	3	61
G_4		3	2	6	1,593	5	2419	5	115	22	4
G_5		3	3	4	3,115	3	4746	4	201	2	63
G_6		4	3	5	2,769	15	505	6	55	20	5
G_7		4	3	8	726	25	416	7	30	27	4
G_8		4	3	10	598	7	909	8	25	0	0
G_9		4	3	12	398	33	312	13	12	0	0
G_{10}		4	3	13	362	22	424	9	18	26	4
G_{11}		4	3	18	156	37	276	53	4	0	0
G_{12}		4	3	29	82	24	418	28	8	0	0
G_{13}		4	3	92	21	12	549	54	4	0	0
G_{14}		4	4	9	625	11	552	31	7	13	8
G_{15}		4	4	22	112	16	495	10	15	0	0
G_{16}		4	4	23	111	20	435	57	3	0	0
G_{17}		4	4	26	85	17	485	83	2	0	0
G_{18}		4	4	30	79	9	706	32	7	29	3
G_{19}		4	4	37	64	38	273	24	9	0	0
G_{20}		4	4	47	51	955	28	0	0	0	0
G_{21}		4	4	90	21	857	31	0	0	0	0
G_{22}		4	4	91	21	1368	20	0	0	0	0

Table 2. Frequent cascades for the 4 product groups. We show all graphs up to 4 nodes and 4 edges. Ordered by size. For each graph we show rank (R) and frequency (F).

Last, figure 3 shows typical classes of cascades. Graphs G_{36} and G_{40} show the case when two people have the same set of friends but do not recommend to each other. A similar case is represented by cascades G_{37} and G_{39}, where the top node recommends to a set of people, and then one of the people in this set purchases and recommends to the same set of people. Flat cascades are also found (G_{38}, G_{41}, G_{42}) – a person recommends, a number of people respond (and purchase a product), but the cascade does not propagate. Graph G_{43} shows cascade that is quite intricate, but which nonetheless occurred 12 times for DVDs.

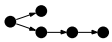
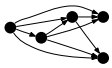
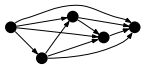
Id	Graph	Nodes	Edges	Book		DVD		Music		Video	
				R	F	R	F	R	F	R	F
G_{23}		4	5	14	274	23	422	0	0	0	0
G_{24}		4	5	34	77	75	171	38	5	28	3
G_{25}		4	5	84	23	52	216	0	0	109	1
G_{26}		4	6	24	105	6	1299	27	8	6	29
G_{27}		5	4	7	1024	74	174	20	10	0	0
G_{28}		5	4	16	211	332	62	47	5	0	0
G_{29}		5	4	50	47	333	62	64	3	0	0
G_{30}		5	4	53	41	282	69	48	5	0	0
G_{31}		5	4	60	31	1045	26	158	1	0	0
G_{32}		5	4	72	27	822	32	21	10	0	0
G_{34}		5	9	137	14	131	119	55	3	15	7
G_{35}		5	10	125	15	18	452	155	1	10	16

Table 3. Some larger frequent cascades for 4 product groups. Ordered by size. For each graph we show rank (R) and frequency (F).

6 Conclusion

The premise behind the study of social networks is that interaction leads to complex collective behavior. Cascades are a form of collective behavior that has been studied theoretically, but for which the study of complete, large-scale datasets has been limited. We have shown that cascades exist in a large real-world recommendation dataset, and investigated some of their structural features.

We developed a practical algorithm and set of techniques to illustrate the existence of cascades, and to measure their frequency. On a large real-life dataset we found novel patterns and our experiments showed that most cascades are small, but large bursts can occur. The cascade sizes and depths follow a power-law. Cascade behavior varies a lot among different product types. Topologically, most products (books, music, videos) tend to exhibit small and shallow tree-like cascades, while some (DVDs) can exhibit larger, more complex, and farther-reaching patterns of influence with collisions and expansion across communities.

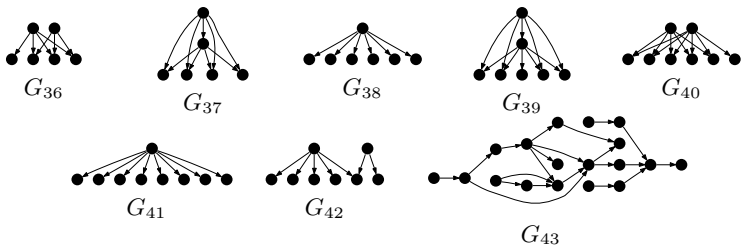


Fig. 3. Typical classes of cascades. G_{36}, G_{40}: nodes recommending to the same set of people, but not each other. G_{38}, G_{41}: a flat cascade. G_{37}, G_{39}: nodes recommending to same community. G_{43} is an example of a large cascade.

References

1. L. Adamic and N. Glance. The political blogosphere and the 2004 US election: Divided they blog. Report, March 2005.
2. E. Adar and L. A. Adamic. Tracking information epidemics in blogspace. 2005.
3. E. Adar, L. Zhang, L. A. Adamic, and R. M. Lukose. Implicit structure and the dynamics of blogspace. In *Workshop on the Weblogging Ecosystem*, 2004.
4. S. Bikhchandani, D. Hirshleifer, and I. Welch. A theory of fads, fashion, custom, and cultural change as informational cascades. *J. of Political Economy*, (5), 1992.
5. P. Desikan and J. Srivastava. Mining temporally evolving graphs. In *WebKDD*, 2004.
6. P. Domingos and M. Richardson. Mining the network value of customers. In *KDD '01*, 2001.
7. J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, 12, 2001.
8. M. Granovetter. Threshold models of collective behavior. *AJS*, 1978.
9. D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. *SIGKDD Explorations*, 6(2):43–52, Dec 2004.
10. A. Inokuchi, T. Washio, and H. Motoda. An apriori-based algorithm for mining frequent substructures from graph data. In *PKDD '00*, pages 13–23, 2000.
11. D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD '03*, 2003.
12. R. Kumar, J. Novak, P. Raghavan, and A. Tomkins. On the bursty evolution of blogspace. In *WWW '03*, pages 568–576. ACM Press, 2003.
13. M. Kuramochi and G. Karypis. An efficient algorithm for discovering frequent subgraphs. *IEEE Trans. on Knowledge and Data Engineering*, 16(9), 2004.
14. J. Leskovec, L. Adamic, and B. Huberman. The dynamics of viral marketing. 2005.
15. M. Newman. The spread of epidemic disease on networks. *Phys. Rev. E*, 66, 2002.
16. M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD '02*, 2002.
17. E. Rogers. Diffusion of innovations (4th ed.). Free Press, 1995.
18. C. Wang, W. Wang, J. Pei, Y. Zhu, and B. Shi. Scalable mining of large disk-based graph databases. In *KDD '04*, 2004.
19. D. Watts. A simple model of global cascades on random networks. *PNAS*, 2002.
20. X. Yan and J. Han. gspan: Graph-based substructure pattern mining. In *ICDM '02*, pages 721–724, 2002.
21. M. J. Zaki. Efficiently mining frequent trees in a forest. In *KDD '02*, 2002.