Patterns of Influence in a Recommendation Network

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

[1] School of Computer Science
Carnegie Mellon University
{jure, ajit}@cs.cmu.edu
[2] Department of Computer Science
Cornell University
kleinber@cs.cornell.edu

Abstract. Information cascades are phenomena in which 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 us to pose a number of fundamental questions: What kinds of 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 is approximately heavy-tailed; cascades tend to be shallow, but occasional large bursts of propagation can occur. The relative abundance of different cascade subgraphs suggests subtle properties of the underlying social network and recommendation process.

1 Introduction

The social network of interactions among a group of individuals plays a fundamental role in the spread of information, ideas, and influence. Such effects have been observed in many cases, when an idea or action gains sudden widespread

Work partially supported by the National Science Foundation under Grants No. IIS-0209107 IIS-0205224 INT-0318547 SENSOR-0329549 EF-0331657IIS-0326322 CCF-0325453, IIS-0329064, CNS-0403340, CCR-0122581, a David and Lucile Packard Foundation Fellowship, and also by the Pennsylvania Infrastructure Technology Alliance (PITA). This publication only reflects the authors' views.
The work of the third author was performed in part while on sabbatical leave at Carnegie Mellon University.

popularity through word-of-mouth or "viral marketing" effects. To take a recent example from the technology domain, free e-mail services such as Microsoft's Hotmail and later Google's Gmail achieved wide usage largely through referrals, rather than direct advertising. (Gmail achieved wide usage at a time when the *only* way to obtain an account was through a referral.) One also finds many examples in weblogs (blogs), where a piece of information spreads rapidly from one blogger to another before eventually being picked up by the mass media.

Information cascades are phenomena in which an action or idea becomes widely adopted due to influence by others [3, 5, 6]. Cascades are also known as "fads" or "resonance." Cascades have been studied for many years by sociologists concerned with the *diffusion of innovation* [15]; more recently, researchers in several fields have investigated cascades for the purpose of selecting trendsetters for viral marketing [9, 14], finding inoculation targets in epidemiology [13], and explaining trends in blogspace [2, 7, 10]. Despite much empirical work in the social sciences on datasets of moderate size, the difficulty in obtaining data has limited the extent of analysis on very 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.

We address a set of related questions: What kinds of cascades arise frequently in real life? Are they like trees, stars, or something else? And how do they reflect properties of their underlying network environment? We describe (in Section 3) a large person-to-person recommendation network, consisting of 4 million people who made 16 million recommendations on half a million products. To analyze the data, we first create graphs where incoming edges influence the creation of outgoing edges. Then, we enumerate and count all possible cascade subgraphs, using an algorithm developed in Section 4. There, we propose an approximate 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 patterns, and the analysis of the results gives us insight into the cascade formation process. We find that the distribution of cascade sizes can be approximated by 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, density, and the shape of cascades across product types. Indeed, the frequency of different cascade subgraphs is not a simple consequence of differences in size or density; rather, we find instances where denser subgraphs are more frequent than sparser ones, in a manner suggestive of properties in the underlying social network and recomendation process.

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 in large part to

the difficulty in acquiring large recommendation network datasets without link ambiguity from a real-world setting.

Most work on extracting cascades from large-scale on-line data has been done in the blog domain [1, 7, 10]. The authors in this domain note that, while information propagates between blogs, examples of genuine cascading behavior appeared relatively rare. This may, however, be due in part to the Web-crawling and text analysis techniques used to infer relationships among pages. In our dataset, all the recommendations are stored as database transactions, and we know that 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, 7] or consider only the properties of blogspace as a graph of unlabeled URLs [1, 10]. Temporally evolving graphs are explored in [4]. Theoretical analysis of cascades on random graphs is provided in [16], using a threshold model. Analysis based on thresholding as well as alternative probabilistic models of node activation is considered in [9, 14]. Note that this analytical work posits a known network; in the present paper, we are able to observe the cascades but not the underlying social network.

In our work we need to efficiently enumerate and count cascade subgraphs. This is an instance of the general issue of *frequent subgraph mining* [8, 11, 17]; however, most of the prior work in this area is focused on graphs that are richly labeled and undirected, often motivated by applications to chemical compound and bioinformatics datasets. While our data has labels as well, we are specifically interested in enumerating subgraphs based purely on their structures, so heuristics for pruning the search space using node and edge labels cannot be applied. Another crucial difference is that we have additional temporal constraints on cascades. We take advantage of the specific problem domain and develop efficient algorithms for extracting "temporally consistent" subgraphs, as well as heuristics for approximate graph isomorphism testing.

3 The recommendation network

We study a recommendation network dataset from a large on-line retailer. During the period covered by the dataset, each time a person purchased a book, DVD, video, or music product, he or she was given the option of sending an e-mail message recommending the item to friends. The first recipient to purchase the item received a discount, and the sender received a referral credit with monetary value. A person could make recommendations on a product only after purchasing it. Since each sender had 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 product recommended, the time the recommendation was sent, whether it resulted in a purchase, and the date of purchase (if applicable). Customer information is anonymized; no demographic or uniquely identifying information is available.

We represent this relational dataset as a labeled directed multigraph: nodes represent customers, and a directed edge (v, w) with label (p, t) means that node

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.

v recommended product p to customer w at time t. (For convenience, we will sometimes denote this edge by (v, w, p, t).) The typical edge generation process is as follows: a node (person) v buys product p at time t, and then recommends it to nodes $\{w_1, \ldots, w_n\}$. These nodes w_i can then buy the product (with the option to recommend it to others). Note that even if all nodes w_i buy the product, only the first purchaser will get the discount, which is marked by a purchase flag (*buy-bit*). However, we can infer purchases by others of the w_i by seeing if they generated subsequent recommendations for it. (Recall that one had to buy the product in order to be allowed 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). A total of 542,719 different products belonging to four product categories (Books, DVDs, Music, and Videos) were recommended. For a detailed analysis of the customer recommendation behavior in this dataset, see [12].

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 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.

There can be multiple recommendations between the nodes, and by counting only unique edges (e_u), we find that only DVDs have more edges than nodes. In summary, all networks are very sparsely linked, but those pairs of users who exchanged recommendations often did so several times. Moreover, 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 occurred in response to a recommendation (b_r). Observe that for DVDs 9% of purchases are associated with a recommendation, for books 3%, music 1.5% and video less than 1%.

Overall, then, while book recommendations appear quite influential, most readers do not appear to make many of them. The DVD network, while smaller, is significantly denser and can be viewed as having a qualitatively richer structure.

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 an approximate heuristic for subgraph isomorphism.

One might imagine cascades to be trees or near-trees. In fact, we find that recommendations create essentially arbitrary graphs: there can be multiple recommendations on the same product or multiple product recommendations between the same pair of nodes; there are multiple purchases of the same product by the same individual (this is natural given that many items are purchased as gifts); and one also finds many cycles.

To find cascades we need to identify cases when incoming recommendations cause purchases and further outgoing recommendations. Recommendations into a node u that precede a purchase can be posited to have potentially 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 (u, v, p, t) and (v, w, p, t') for $t' > t$ suggests cascade behavior. That is, node v receives a recommendation for product p at time t and then makes recommendation for the same product at a *later* time t'. (Recall that a node makes recommendations at the time of purchase.)

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

Delete late recommendations: given a single product recommendation network, for every node we delete all incoming recommendations (edges) that happen after the first purchase of a product. This removes all recommendations of the product a person received after the first purchase, i.e. keeps only recommendations that potentially influenced the purchase. Now for every node the time of all incoming edges is strictly smaller than the time of all outgoing edges.

Delete no-purchase nodes: Preliminary analysis showed that the majority of recommendations do not produce cascades. We observed many star-like patterns where the center node recommends to a large number of people, none of whom purchase the product. 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 in the undirected version of the graph can be viewed as a cascade, since all directed 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 include only nodes whose behavior was potentially affected by the cascade (as evidenced by the fact that they made a purchase).

Cascade enumeration: Next we enumerate all possible cascades. By the discussion in the previous paragraph, the undirected components correspond to maximal cascades, but simply enumerating components makes it difficult to reason about the smaller building blocks of the cascades. Thus, we instead focus on enumerating all *local cascades*: For every node we explore the cascade in the (undirected) neighborhood of the node. Thus, for every node v, we create the

subgraphs induced on nodes reachable by up to h steps forward or backward from v (stopping at h that includes all reachable nodes). One can think of this as capturing the local structure of the cascade at increasing distances around v.

Approximate graph isomorphism: An essential step in counting cascades is determining whether a new cascade is isomorphic to a previously discovered one. 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 technique employed is to create an efficiently computable signature for each graph, use hashing on this signature value, 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 sequences. 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 exact isomorphism check is used, we keep a list of all variants of graphs that collided (have the same signature). Since we first hash, we perform the isomorphism check only on graphs with the same signatures, and so the number of true isomorphism checks is small.

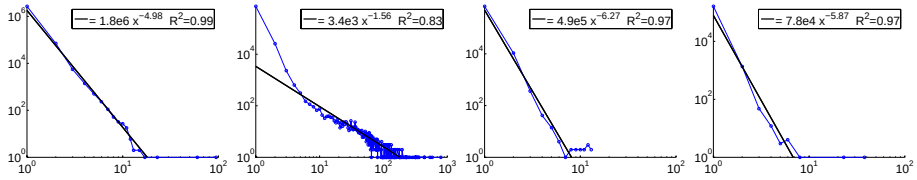
Note that a small minority of cascades are larger than 9 nodes, so for most of the subgraphs we get the exact solution; as the cascade size increases the number of occurrences decreases, and this is where we use an approximate solution.

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 machine. In the second experiment we generated 100,000 random graphs (using the Erdős-Rényi model), each of them with a randomly chosen number of nodes between 4 to 20 and twice as many edges. The counting took 50 seconds. In this experiment we observed at most 53 non-isomorphic graphs (5 nodes, 10 edges) with the same signature.

5 Patterns of recommendation

5.1 Size distribution of cascades

First, we discuss results on the size of the cascades, measured by the number of nodes. 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 for the four product types.



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

Fig. 1. Size distribution of the cascades for the four product types (log size of cascade vs. log count). Superimposed line presents a power-fit. R^2 is the coefficient of determination.

The size of cascades follows a heavy-tailed distribution. 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.

DVDs had the highest proportion of large cascades, and the plot for DVDs in Figure 1(b) has an interesting transition in its behavior. For smaller cascade sizes, in the size range consistent with most of the book, music, and video cascades, the DVD distribution has a power-law fit with slope -4.5, comparable to the other three product types. For larger cascades, which are observed in abundance only for DVDs, the distribution flattens to a slope of -1.5.

The cascade size distributions suggest that the simplest branching process models will not suffice to explain the underlying cascade process; a family of richer models is proposed in [12], in which the success probability increases when collisions occur among cascades, and cascade sizes follow a power law with exponent -1. We have also found that the cascade size distribution follows a heavy-tailed distribution in sales frequencies [12], with the number of purchases decaying as a function of rank faster than the number of recommendations does.

5.2 Frequent cascade subgraphs

What kinds of 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 described procedure: for each product recommendation graph, we first identify cascades by deleting late recommendations and no-purchase nodes. Then, for each node we create a subgraph on nodes at distance up to h hops, where h varies from 1 up to the value where all nodes are reached. We count the graphs using the approximate graph isomorphism technique from Section 4.

General observations: For books we identified 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, 87,614 are topologically different, and 3,015 cascades occur at least ten times. For music we identified 13,330 cascades, 158 were topologically different, and only 23 cascades occurred at least ten times. Videos were the least rich, with 1,928 subgraphs containing 109 unique patterns, and only 12 subgraphs occurring at least ten times.

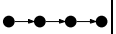
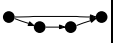
Id	Graph	Nodes	Edges	Book		DVD		Music		Video	
				R	F	R	F	R	F	R	F
G_1		2	1	1	86,430	1	36,863	1	11,518	1	1,425
G_2		3	2	2	10,573	4	3,238	2	492	5	33
G_3		3	2	3	5,089	2	5,147	3	389	3	61
G_4		3	2	6	1,593	5	2,419	5	115	22	4
G_5		3	3	4	3,115	3	4,746	4	201	2	63
G_6		4	3	5	2,769	15	505	6	55	20	5
G_7		4	3	8	726	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 types. We show all graphs up to 4 nodes and 4 edges. Ordered by size. For each graph we show rank (R) and frequency (F).

The frequency of different cascades concurs with observations made in connection with Figure 1 and Table 1, where DVDs had the largest and richest set of cascades. Also, 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 types, including all subgraphs with at most four nodes and four edges. Cascades are ordered by size. 14 cascade patterns can be observed in all four product types; Table 2 includes 10 of them.

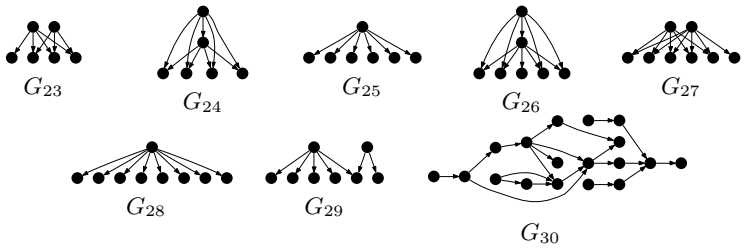


Fig. 2. Typical classes of cascades. G_{23}, G_{27}: nodes recommending to the same set of people, but not each other. G_{24}, G_{26}: one node recommends to another, and both recommend to the same community. G_{25}, G_{28}, G_{29}: a flat cascade. G_{30} is an example of a large cascade.

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) accounts 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} (involving a collision), followed by G_{14} and G_{15}.

Finally, Figure 2 shows typical classes of cascades. Graphs G_{23} and G_{27} show the case when two people have the same set of friends but do not recommend to each other. Similarly, in graphs G_{24} and G_{26}, the top node recommends to a set of people, and then one purchases and recommends to the same set. Flat cascades are also found (G_{25}, G_{28}, G_{29}) – a person recommends, a number of people respond (and purchase a product), but the cascade does not propagate. Graph G_{30} shows an illustrative example of a cascade that is quite intricate.

A concluding, general observation is that the frequency of cascade subgraphs does not simply decrease monotonically in the number of nodes and edges; for example, G_5 is more frequent than either of its subgraphs G_2 and G_4 in DVDs and videos (and more frequent than G_4 in books and music). Thus, frequency appears to reflect properties of the underlying social network (the clustering of people who know each other), as well as properties of the ways in which recommendations typically get made (e.g. splits are more common than collisions).

6 Conclusion

A basic 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 analyzed both empirically and 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 have investigated some of their structural features.

We developed a scalable algorithm and set of techniques to illustrate the existence of cascades, and to measure their frequencies. From our experiments, we found that most cascades are small, but large bursts can occur; that cascade sizes approximately follow a heavy-tailed distribution; that the frequency of different cascade subgraphs depends on the product type; and that these frequencies do not simply decrease monotonically for denser subgraphs, but rather reflect more subtle features of the domain in which the recommendations are operating.

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. 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.
4. P. Desikan and J. Srivastava. Mining temporally evolving graphs. In *WebKDD*, 2004.
5. 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.
6. M. Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, 83(6):1420–1443, 1978.
7. D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. *SIGKDD Explorations*, 6(2):43–52, Dec 2004.
8. 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.
9. D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD '03*, 2003.
10. R. Kumar, J. Novak, P. Raghavan, and A. Tomkins. On the bursty evolution of blogspace. In *WWW '03*, pages 568–576. ACM Press, 2003.
11. M. Kuramochi and G. Karypis. An efficient algorithm for discovering frequent subgraphs. *IEEE Trans. on Knowledge and Data Engineering*, 16(9), 2004.
12. J. Leskovec, L. Adamic, and B. Huberman. The dynamics of viral marketing. 2005.
13. M. Newman. The spread of epidemic disease on networks. *Phys. Rev. E*, 66, 2002.
14. M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD '02*, 2002.
15. E. Rogers. Diffusion of innovations (4th ed.). Free Press, 1995.
16. D. Watts. A simple model of global cascades on random networks. *PNAS*, 2002.
17. X. Yan and J. Han. gspan: Graph-based substructure pattern mining. In *ICDM '02*, pages 721–724, 2002.