Research Statement

1 Overview

My research interests lie in the use of algorithmic analysis tools to improve the scientific understanding of social, economic and information networks that exist on the World Wide Web. The astounding growth of social networks such as Facebook and Twitter in the past decade signifies an exciting time for scientists studying sociological phenomenon: the online (and in many cases, public) nature of these networks allow scientists access to a dataset of unprecedented scale, with visible social connections and an archive of the users' online activity. Given the growing popularity of social networks, economic activity such as marketing has also begun to proliferate on these systems [5]. Theoretical computer scientists such as myself are in a unique position to contribute to the study of social and economic networks as:

1. Traditional socio-economic analysis tools are ill-equipped to deal with the constraints that network structure places on social and economic interaction in online social networks. In contrast, graph-theoretic tools regularly employed by theoretical computer scientists are particularly well-suited for analytical study of networks (for example, tools from random graph theory have a ubiquitous presence in this literature).
2. The massive scale of these networks poses new and fundamental computational challenges, and algorithmic analysis is central to finding principled answers to these questions (analogous to the importance of routing algorithms in communication networks).

The contributions of algorithmic analysis to the study of social and economic networks is reflected in my own research: in the following two sections, I highlight a marketing policy for social networks that leverages the networked nature of the world, and a caching problem that is a computational bottleneck for efficient performance of real-life social networks. In the last section, I briefly recount three problems which use algorithmic analysis to propose solutions for problems motivated by Web applications.

2 Network Effects: Marketing on Social Networks

Social networks such as Facebook, Twitter, Orkut and MySpace are highly valued for both their user retention capabilities and the detailed user profile and network data that they collect. Despite the widespread belief that this information could be a gold mine for targeted advertising and other online businesses, much of this potential still remains untapped today. Facebook, for example, was valued at $15 billion by Microsoft in 2007 [19], but its estimated revenue in 2008 was only $300 million [24]. Thus, any monetization technology that can help bridge this gap is of paramount interest.

Of particular interest are large-scale monetization technologies that can effectively leverage the networked nature of the online social networks, and move away from the currently used paradigm of contextual advertising borrowed from sponsored search [7, 23, 10].

In a recent work [1], we propose a monetization approach that effectively leverages the network structure to market products on the social network through the spread of influence. The idea behind the approach is that users can often be convinced to purchase a product if many of their friends are already using it, even if these same users would be hard to convince through direct advertising. This is often a result of personal recommendations – a friend's opinion can carry far more weight than an impersonal advertisement. Furthermore, in some cases adoption among friends is important for even more direct reasons. For example, instant messenger users and cell phone users will want a product that allows them to talk easily and cheaply with their friends. This *network influence* effect is well-known, and has garnered increasing attention from the computer science community in the past decade [13, 14].

In fact, many sellers already do try to utilize strategies that are based on these tendencies. *Viral marketing* is an example of such a strategy, where a seller attempts to market the product through word-of-mouth

recommendations among potential customers [17, 18, 22]. A more powerful but riskier technique has been in use much longer: the seller gives out free samples or coupons to a limited set of people, hoping to convince these people to try out the product and then recommend it to their friends. To implement such a strategy, the seller needs to make the following decisions:

> How many free samples does the seller need to give out? Which people should the seller give out the free samples to? What incentives can the seller afford to give to people for recommending the product without jeopardizing his overall profit too much?

In our work [1], we find systematic answers to these questions. Towards this goal, we model the spread of a product as a cascading process on a social network. In the network, a node represents a single person, and an edge represents a friendship. Initially, one or more nodes is "active", meaning that person already has the product. This could either be a large set of nodes representing an established customer base, or it could be just one node – the seller – whose neighbors consist of people who independently trust the seller, or who are otherwise likely to be interested in early adoption.

At this point, the seller can encourage the spread of influences in two ways: 1) Offer cashback rewards to individuals in order to incentivize them to recommend the product to their friends (this is often seen in practice with "referral bonuses" – if a referral from an existing customer results in an new customer for the firm, the recommending customer receives a cash reward). 2) Offer discounts to specific people in order to encourage them to buy the product, above and beyond any recommendations they receive. It is important to choose a good discount from the beginning here. If the price is not acceptable when a prospective buyer first receives recommendations, they might not bother to reconsider even if the price is lowered later.

After receiving the discount offers and some set of recommendations, it is up to the prospective buyers to decide whether to go through with a purchase. We model their decision using a random utility function drawn from a known distribution that is influenced by two parameters: the value of the discount and the set of recommendations they have received. This distribution is a parameter of the model as it is determined by external factors, for instance, the quality of the product and various exogenous market conditions. One could interpret the utility according to a number of different models that have been proposed in the literature (for instance, the Independent Cascade and Linear Threshold models [14]), and hence it is desirable for the seller to be able to come up with a strategy that is applicable to a wide variety of models.

Note that the seller can choose the price for each person/node either in advance, called a *non-adaptive* strategy, or after observing the (random) decisions in each step, called an *adaptive* strategy.

> Our main theoretical contribution is a very efficient non-adaptive seller strategy whose expected revenue is within a *constant factor* of the optimal revenue from an *adaptive* strategy.

This guarantee holds for a wide variety of utility distributions, including natural extensions of both the Linear Threshold and Independent Cascade models[1]. We also show that the problem of finding an optimal non-adaptive strategy is NP-hard, which implies that an efficient approximation algorithm is the best theoretical result that one could hope for.

Intuitively, the seller strategy we propose is based on an *influence-and-exploit* idea, and it consists of categorizing each potential buyer as either an *influencer* or a *revenue source*. The influencers are offered the product for free and the revenue sources are offered the product at a pre-determined price, chosen based on the exact probability model. Briefly, the categorization is done by finding a spanning tree of the social network with as many leaves as possible, and then marking the leaves as revenue sources and the internal

[1] More precisely, the strategy achieves a constant-factor approximation for any *fixed* model, independent of the social network. If one changes the model, the approximation factor does vary, as is made precise in the paper [1].

nodes as influencers. Cashback amounts are chosen to be a fixed fraction of the total revenue expected from this process.

In summary, monetizing social networks is an important practical problem where exploiting the network structure seems to be the crucial aspect of the problem. The network aspect necessitates analysis distinct from traditional economics, and our research in an example of this distinction.

2.1 Ongoing Research and Future Directions

Several well-studied socio-economic problems that arise on social networks require reformulation given the network externality effect. In this section, we point out a natural extension of the model in the previous section. Note that in that model, users were assumed to be myopic and hence they did not strategize. In ongoing work [15], we are studying the equilibria that arise when both the sellers and buyers are allowed to strategize.

We consider the problem of choosing a sequence of prices for a monopolist seller who is trying to sell a good to users in a repeated interaction game. In particular, the monopolist posts a *single public price* in each period (i.e., all users are give take-it or leave-it offers at this price), and users decide whether to buy the product in this round or not. Each user's utility for the good consists of an intrinsic value, and a positive externality they observe from their friends on the network that purchase the good. Since both the seller and the buyers are strategic, this results in a game whose equilibrium behavior is a prediction for prices that would be observed.

The monopolist pricing problem with network effects has been studied extensively in economics but the literature has focused on the case where the underlying social network is the complete graph, i.e. the "network effect" is really a *size* effect [6]. We consider the multi-period pricing problem for general graphs, which is a more realistic model for interaction on online social networks. This problem has also attracted recent attention from the economics community for simple graph topologies such as the star and the circle [21]. Several computational questions arise for this problem:

> Note that the equilibrium for the game constitutes a valid prediction only if the players can compute it in polynomial time. Further, the seller might want to influence the equilibrium to one that is more favorable to him through targeted marketing. In particular, given an investment budget of the seller, say k nodes, which k nodes should the seller target?

There are several other such problems where the explicit modeling of the underlying social network results in a behavior and prediction distinct from one made by merely modeling the size effect. A relevant example is the problem of exploiting the network structure for contextual advertising on social platforms.

3 Algorithms for Social Networks: Caching Strategies for Social Networks

The massive scale of online social networks leads to several algorithmic challenges. We consider one such problem in an ongoing work [16], where the objective is to design caching strategies that social networks such as Facebook and Twitter can employ for efficiently serving the customized home pages of its users. As before, we model the social network as a graph, where users are nodes and edges indicate friendships. Recall that each time a user opens the social network website (called a query), he is presented with a customized home page that the social network constructs by pulling in information from all of the friends of the user.

> Thus, to answer each query in an efficient manner, the social network needs to have the querying node and all of its friends in the cache. A *cache miss* is said to occur each time one of these nodes is not in the cache. The objective of the caching strategy is to minimize the number (or more generally, a cost function) of cache misses.

Formally, we've been studying [16] a generalization of this problem, which can be termed *set caching*.

Given a set $V = \{1, 2, \ldots, n\}$, a series of requests are generated online where the request at a time t is a set $S_t \subseteq 2^V$. Any algorithm has available to it a cache C of size k (assume $k \geq \max_t |S_t|$), and can successfully answer a request if all of S_t are present in the cache at time t, i.e. $S_t \subseteq C$. If not, then the algorithm is allowed to swap in the required nodes $S_t \setminus C$ for a cost c_t. There are several ways one can count the cache misses, and a natural cost function c_t is as follows: the cost of i cache misses in one step have cost $1 + f(i)$, where $f(i) : [k] \to [0, 1]$ is a monotone function of the cache misses. This cost function is motivated by a fixed cost for accessing the disk and an additional cost of accessing each individual node.

Since the set requests S_t are generated online, the goal is to design an algorithm that has the optimum competitive ratio. Note that the above problem is clearly a generalization of the graph case, as the set S_t could correspond to a node and its neighborhood.[2] The problem is also a generalization of the weighted paging problem that is in itself quite well-studied [3, 4].

We have obtained upper and lower bounds for special cases of the cost function and are currently extending our results for more general cost functions, using primal-dual type techniques that have been developed for the weighted paging problem [3, 4].

3.1 Future Directions

Improved algorithms and tight analysis of the caching problem for the following special cases would be of immediate practical interest: detailed cost models that model the actual caching system closely, query distributions that are observed in practice and graph topologies that capture the structure of the actual network. One can also go one step further, and devise offline pre-processing strategies that could speed-up the runtime service. For instance, the social network might want to partition the graph such that different pieces of the partition are stored on different servers, and each piece can be constrained to be small enough so that it can be stored in the cache of a server. Suppose several such (different) partitions are allowed to be stored. The goal is to be able to answer any user query quickly, i.e. for any node, we need to ensure that there is a piece (of some partition) in which the user and its neighbors are in cache (each query can then be resolved by an index mapping nodes to pieces, and served from the cache). Note that the social network might be willing to store multiple copies of the graph, but this redundancy obviates the need for network access to answer node queries. The key computational question is: what is the minimum number of partitions needed, and how should the partitions be constructed?

4 Algorithms for the Web

I have also worked on several other algorithmic problems that find applications on the Web, and will briefly mention three problems here.

4.1 Diversification of Search Results

Search is one of the primary user activities on the Web, and despite the unequivocal success of search engines, search engines do not always understand *user intent* behind a query. Without any explicit knowledge of user intent, search engines want to diversify results to improve user satisfaction. In such a setting, the search engine would like to trade-off relevance for dissimilar results. The exact form of this trade-off has been quantified in several different ways, resulting in several interpretations of diversification. In joint work with Sreenivas Gollapudi at Microsoft Research [8], we attempt to unify the plethora of diversification systems using an axiomatic characterization of diversification under a general framework. In the context of web search, given a query, our diversification framework for web pages consists of three separate parts: a relevance function for the web pages given the query, a pairwise similarity function between web pages and

[2] Note that in this case, we would be assuming an oracle that gives us the neighborhood of a node at no cost. But this oracle can be absorbed in the cost function by adding an update cost proportional to the size of the cache (for updating the adjacency lists), and hence we will ignore this issue for now.

a diversification function which produces a diverse ranking given the relevance and distance function. In the paper:

> We provide a mathematical characterization of diversification systems in terms of a maximal (as demonstrated by an impossibility result) set of natural axioms, present new combinatorial algorithms for diversification with provable approximation guarantees, and conduct an experimental evaluation based on a data set derived from the disambiguation pages listed in Wikipedia.

4.2 Nearest Neighbor Search

Nearest neighbor search is a fundamental data structure query that finds applications in a variety of Web applications. The approximate Nearest Neighbor ($\mathcal{NN}$) search problem asks to pre-process a given set of points P in such a way that, given any query point q, one can retrieve a point in P that is *approximately* closest to q. Of particular interest is the case of points lying in high dimensions, which has seen rapid developments since the introduction of the Locality-Sensitive Hashing (LSH) data structure by [11]. Combined with a space decomposition by [9], the LSH data structure can answer approximate $\mathcal{NN}$ queries in sub-linear time using polynomial (in both d and n) space. Unfortunately, it is not known whether the space decomposition in [9] can be maintained efficiently under point insertions and deletions, so the above solution only works in a static setting. In our paper [2],

> We present a variant of Har-Peled's [9] decomposition, based on random semi-regular grids, which can achieve the same query time with the added advantage that it can be maintained efficiently even under adversarial point insertions and deletions. The outcome is a new data structure to answer approximate $\mathcal{NN}$ queries efficiently in dynamic settings. Our results also include the first sub-linear time algorithm for the related Reverse Nearest Neighbor ($\mathcal{RNN}$) problem (which seeks to find the *influence set* of a given query point q, i.e. the subset of points of P that have q as their nearest neighbor).

4.3 Learning Click-through Rates for Sponsored Search

Sponsored search auctions are a major revenue source for search engines, and not surprisingly, this problem has received a lot of attention in recent years. One of the primary factors that affect revenue from the auctions is the quality of ad click-through rate (CTR) estimates, as the CTR value is used both for ranking the ads and deciding what to charge the advertiser. In recent work [12], we studied algorithms for learning CTRs in the setting of sponsored search. Specifically, we study bandit algorithms for online learning by considering the separable model in which the probability that an ad will be clicked at a position is a product of the position bias (the probability that the user observes the ad in the position) and the ad quality (the probability that the ad is useful).

We extend previous results [20] and present an algorithm that can *learn CTRs and position biases simultaneously*. We show that the algorithm is efficient in terms of revenue by proving that the revenue of the algorithm can be lower from the revenue of the optimum algorithm by at most a logarithmic (in time) additive term. In the terminology of online algorithms, the algorithm achieves *logarithmic regret*, which is optimal up to constant factors. We further show that the algorithm outperforms previously known algorithms through simulations with historical sponsored search data.

References

[1] David Arthur, Rajeev Motwani, Aneesh Sharma, and Ying Xu. Pricing strategies for viral marketing on social networks. *Proceedings of the 5th Workshop on Internet and Network Economics*, pages 101–112, 2009. URL `http://arxiv.org/abs/0902.3485`. 1, 2

[2] David Arthur, Steve Oudot, and Aneesh Sharma. Finding Friends and Followers in Sub-linear Time, under submission. URL http://hal.archives-ouvertes.fr/inria-00429459/en/. 5

[3] Nikhil Bansal, Niv Buchbinder, and Joseph (Seffi) Naor. Randomized Competitive Algorithms for Generalized Caching. *Proceedings of the 40th Annual ACM Symposium on Theory of computing*, pages 235–244, 2008. URL http://research.microsoft.com/en-us/um/people/nivbuchb/papers/web-caching/web-caching.pdf. 4

[4] Nikhil Bansal, Niv Buchbinder, and Joseph (Seffi) Naor. Towards the Randomized k-Server Conjecture: A Primal-Dual Approach. *To appear in the Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2010. URL http://domino.research.ibm.com/comm/research_people.nsf/pages/nikhil.pubs.html. 4

[5] Amazon Associates Blog. Amazon affiliate marketing on twitter. http://affiliate-blog.amazon.com/2009/11/we-are-excited-to-announce-the-launch-of-a-new-feature-called-share-on-twitter-you-can-access-share-on-twitter-from-the-site.html, 2009. 1

[6] David Easley and Jon Kleinberg. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010. URL http://www.cs.cornell.edu/home/kleinber/networks-book/. 3

[7] Stephen Foley. Google searches and finds myspace deal. http://www.independent.co.uk/news/business/news/google-searches-and-finds-myspace-deal-411027.html, 2006. 1

[8] Sreenivas Gollapudi and Aneesh Sharma. An Axiomatic Approach for Result Diversification. *Proceedings of the 18th International Conference on World Wide Web*, pages 381–390, 2009. URL http://www2009.org/proceedings/pdf/p381.pdf. 4

[9] Sariel Har-Peled. A Replacement for Voronoi Diagrams of Near Linear Size. *Annual Symposium on Foundations of Computer Science*, 42:94–105, 2001. URL http://valis.cs.uiuc.edu/~sariel/research/papers/01/avoronoi/avoronoi.pdf. 5

[10] J. Hartline, V. Mirrokni, and M. Sundararajan. Optimal Marketing Strategies over Social Networks. *Proceedings of the 17th international conference on World Wide Web*, 2008. 1

[11] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613, 1998. 5

[12] Satyen Kale, Mohammad Mahdian, Kunal Punera, Tamas Sarlos, and Aneesh Sharma. Learning Click-through Rates for Sponsored Search Advertisements, under submission. 5

[13] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146, 2003. 1

[14] Jon Kleinberg. Cascading Behavior in Networks: Algorithmic and Economic Issues. In N. Nisan, T. Roughgarden, E. Tardos, and V.V. Vazirani, editors, *Algorithmic Game Theory*. Cambridge University Press New York, NY, USA, 2007. 1, 2

[15] Jon Kleinberg and Aneesh Sharma. Pricing Network Goods for Strategic Users, in preparation. 3

[16] Aleksandra Korolova, Rajeev Motwani, and Aneesh Sharma. Caching Strategies for Social Networks, in preparation. 3

[17] J. Leskovec, A. Singh, and J. Kleinberg. Patterns of influence in a recommendation network. *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, 2006. 2

[18] Jure Leskovec, Lada A. Adamic, and Bernardo A. Huberman. The dynamics of viral marketing. *ACM Trans. Web*, 1(1):5, 2007. ISSN 1559-1131. doi: http://doi.acm.org/10.1145/1232722.1232727. 2

[19] BBC News. Facebook valued at $15 billion. http://news.bbc.co.uk/2/hi/business/7061042.stm, 2007. 1

[20] S. Pandey and C. Olston. Handling advertisements of unknown quality in search advertising. *Advances in Neural Information Processing Systems*, 19:1065, 2007. 5

[21] Pekka Sääskilahti. Monopoly Pricing of Social Goods. *MPRA Paper*, 3526, 2007. URL http://mpra.ub.uni-muenchen.de/3526/1/MPRA_paper_3526.pdf. 3

[22] Erick Schonfeld. Amiando makes tickets go viral and widgetizes event management. http://www.techcrunch.com/2008/07/17/amiando-makes-tickets-go-viral-and-widgetizes-event-management-200-discount-for-techcrunch-readers/, 2008. 2

[23] Katharine Q. Seelye. Microsoft to provide and sell ads on facebook. http://www.nytimes.com/2006/08/23/technology/23soft.html, 2006. 1

[24] Wikipedia. Facebook revenue in 2008. http://en.wikipedia.org/wiki/Facebook, 2008. 1