Research Statement

Michael Ekstrand

I research ways to help people find, filter, track, remember, and make better use of information. Each day, the modern Internet user engages with countless pieces of information. They likely have personally stored additional data and there is much more available with a quick trip to Google. I want to build tools to help users find and keep their way in this sea.

Within this broad interest, I have devoted most of my research energies to date to studying recommender systems. My research activities include system building, offline experimentation with public data sets, and user studies, all bent towards one goal: improving recommendation (broadly defined) as a tool for helping users find the things they want or need.

Recommender Applications

My earliest recommender work [@Ekstrand:ReadingLists] was on recommending reading lists of research papers. I combined collaborative filtering with PageRank-style importance algorithms to find authoritative papers to recommend that the user read. At the same time, I wrote a survey of recommender research, particularly focusing on collaborative filtering [@Ekstrand:CFSurvey].

I then took an internship with Autodesk Research, where worked on another recommender application: context-sensitive software help search, mining a user's interactions with a software program for information that can help refine their search for help resources. The resulting paper, published in UIST 2011 [@Ekstrand:HelpSearch], also includes a survey and taxonomy of the aspects of context that are likely to be useful when searching for software help and how that context can be modeled and used computationally.

Towards Recommender Engineering

I think that discovering and organizing the knowledge needed to engineer recommenders from well-known principles is an important long-term goal for the field. Under this paradigm, a recommender developer will identify the important properties of their domain, the tasks they need to support, and the goals of their users. They will be able to use this information to find an algorithm or two that meets their requirements and then use user testing to fine-tune the result.

Much current recommender research is not yet adequate to get us to this state. Many experiments examine individual pairings of algorithms and tasks. New algorithms are tested for their ability to do better at some particular recommendation problem; multiple problems may be considered to show generality. In industrial settings, A/B testing has been very valuable to develop and test recommenders. Pairwise approaches suffer, however, from difficulties in producing generalizable results, particularly about mediating factors: what made algorithm A better for task B, and how can that knowledge be applied to engineer a new recommender application?

Advancing recommender research to enable engineering requires a lot of work. We need to develop extensive knowledge of the characteristics, strengths, and weaknesses of various recommender algorithms and understand how they relate to the requirements of

different domains, tasks, and user goals. These results also need to be robustly validated so they can be trusted in a wide array of situations.

I have focused my thesis work around moving the field in this direction, building tools for reproducible recommender research and using them to work towards recommender engineering and explore differences between recommender algorithms.

The first major component of this work is LensKit [@Ekstrand:LensKit], an open source toolkit for recommender systems research, prototyping, and education. LensKit has been the underlying technology for several GroupLens research projects and is the engine that powers MovieLens[1]. I also intend it to advance the state of recommender research by providing a common, reusable platform with well-defined evaluations and high-quality implementations of well-known algorithms.

LensKit has presented a number of technical challenges. For one, we have sought to make LensKit's algorithms modular and reconfigurable, with many reusable components; such object graphs are, unfortunately, difficult to configure manually. To meet LensKit's particular recommender configuration needs, I worked with Michael Ludwig to develop the Grapht dependency injector. Grapht enables LensKit to statically analyze the object graph needed to configure a recommender, providing automatic support for deploying recommender algorithms in web applications, and allows rich, context-sensitive configurations of arbitrarily composable components.

We have also worked on automatically tuning recommender parameters. Many algorithms have various parameters (neighborhood sizes, latent feature counts, etc.) that drastically affect their performance, but the relevant literature does not devote much effort to efficient means of tuning these parameters. Since thorough grid search is prohibitively expensive for large data sets, I worked with one of the undergraduate research assistants in our lab to develop efficient strategies for tuning the parameters of item-item collaborative filtering and attempt to do so for FunkSVD.

The second major component of my thesis work is an experiment in MovieLens asking users to identify differences between the lists produced by different recommender algorithms. I am trying to determine what characteristics the algorithms have (e.g. does one produce more diverse lists, but another more unexpected recommendations?) and how those characteristics contribute to users preferring one algorithm's recommendations to another's. This experiment will hopefully result in a deeper understanding of the differences between three well-known recommender algorithms. Understanding how they are *different*, not just which one is *better* by some observable metric, will help us understand how each algorithm might perform in other contexts.

I am collaborating on this experiment with Martijn Willemsen, a decision psychologist at the Technical University of Eindhoven. In both this project and a previous collaboration led by some of the younger students in our research lab [@Nguyen:RatingInterfaces], he has provided a valuable perspective in analyzing users' interactions with and opinions of a computing system.

[1] http://movielens.org

Future Work

In my future research, I hope to continue my blend of data-driven and user-based experimentation, along with interdisciplinary collaboration and open source software publication. I believe that embodying research results in open-source software is a powerful method for making the outcomes of computer science research accessible and usable.

My recent research has focused on entertainment recommendation (movies and music), due to the availability of data and experimentation platforms; I would like to turn back to recommendation and retrieval in more 'serious' domains such as research literature, news, and social media entities. I am particularly interested in domains with explicit relationships between items, such as research paper citations, co-authorship, or social network links. I think such relationships are a rich but underutilized source of information to improve recommendation, and that there is a lot of unrealized potential for recommendation technology to help users navigate and make sense of large, complex, graph-structured information spaces.

I am also interested in applying my skills in user experience and recommender research to supporting decentralized, censorship- and surveillance-resistant communication. There are many reasons why decentralized social networks have failed to gain traction; besides the economic difficulties, I think user experience and discovery are important concerns. I would like to study how to support discovery and recommendation on distributed, decentralized social computing systems without compromising those systems' important security and privacy properties.