Amir H. Hajizamani
St. John's College
ahh29

CST Part II Dissertation Project Progress Report

Recommender Systems for Social Networks

February 6, 2011

Project Supervisor: Cecilia Mascolo

Director of Studies: Robert Mullins

Overseers: Ann Copestake and Robert Harle

This report is a summary of the work done towards the aims of my project at its halfway point, the challenges faced and how the outcome of these compares against the plan I set out at the beginning.

In the context of the project proposal, a recommender system for a social network can be viewed as a predictor of future activity in the social graph, that is, given a dataset of users with social links, the output of the system is a set of predictions of further links that may appear at a later point in time. This could also work on the content interaction graph – recommending items for consumption. With the aim of building a recommender system that does this well, I have identified a number of high-level tasks that need to be accomplished: obtaining the dataset to work with; ensuring the dataset is in a consistent state ("scrubbing"); exploring the data in order to be familiar with its qualities and properties; choosing a model to describe the dataset with; and finally using the information from the previous stages to make predictions.[1]

Obtaining the dataset is the task that has taken the most amount of time so far by the nature of the fact that I had to fetch it through Mixcloud's API over HTTP. I have written a sophisticated wrapper around the API in Python which takes care of a lot of its particular intricacies, including some bugs which took some time to identify. Equipped with this, I then wrote a crawler to fetch the data. As the API allows access to users' data only by their unique usernames, I use a depth-first search to gather as much of the connected graph as possible, starting with some seed users. I store the JSON output of the API in the NoSQL database MongoDB and use the semi-persistence features of Redis (in-memory cache, similar to memcached, with periodic writes to disk) to ensure continuity over disruptions to the run, such as network outages. I eventually also added the capability to use proxies to access the API when the current one hits the API's rate limit, which had the effect of speeding up the scans by a factor equal to the number of proxies used. By now, I have obtained 3 snapshots of the dataset, with a reassuringly steady growth in their sizes.

Then I had to scrub each snapshot. The only issue worth mentioning here is the fact that due to the period of few days between the start and end of each scan there were inconsistencies in the data about the links between users and content. For example, if a link had been formed between two users towards the end of the scan, it would only be picked up with respect to only one of the users. Fortunately this was fairly straightforward to fix by, essentially, rolling back asymmetric links until everything matched a state of the network at some point near the start of the scan. Other minor clean-ups were also performed.

Exploring the dataset is an ongoing process, but in short, I collected basic statistics (including distribution of social activity and content interaction, correlations between different features of a user) on the data which at the very least confirmed that it was structured as expected—features such as neighbour count and upload count roughly following a power law—but also informed some decisions later on as to metrics to use to measure similarities, how to optimise performance and so on.

As for modelling and interpretting—in this case making predictions on—the data, that is currently where I am trying to make advances. For the time being I am focusing on the dataset's social structure and treating it as a social network: people with many common social connections are more likely to form their own direct connection than those with fewer. So I have now got a simple recommender system which, given a user to make recommendations for, will calculate social similarity values between that user and people in its near vicinity,

[1] `http://www.dataists.com/2010/09/a-taxonomy-of-data-science/`

for example friends of friends, and outputs the top most similar users as recommended new social connections. The similarity metrics I have tried have been variations on the Jaccard index, which in its pure form is the size of the intersection of two users' social links divided by the size of the union of their social links. To evaluate the performance of this simple recommender, I am initially generating test data from a given snapshot by hiding some of social links in the data, making predictions and looking at the recall and precision rates in the output given against the original data. The results have been fairly acceptable, with recall rates of above 50% but with low precision. The next step would be to incorporate more of data available on each user, such as favorited content, to at least increase the precision and perhaps recall, too.

In summary, I believe my project is reasonably on track. With respect to the timetabling in my original proposal, I do not think my recommender system is as sophisticated as I expected it to be at this stage, and there has been less preparation for the dissertation write-up than I planned. On the other hand, I am now in possession of several snapshots of the dataset that represent the problem I am trying to address. This means that the opportunity to do temporal evaluation of the system's performance is present. So I would say that so far there have been few difficulties that weren't promptly resolved and I am confident that I am making reasonable progress.

Amir H. Hajizamani
February 2011