**COS 424 Project Proposal** 4/6/2012

*Members*: Stephen Beard, Joey Dodds, Soumyadeep Ghosh, Matt Zoufaly

*Project Name*:Nightout

**Problem**

Exploring where you are isn’t as easy as it sounds. Word of mouth combined with reviews from places like Google and Yelp help to find good places to eat and fun things to do. However, it would be nice to have a simpler approach: create an application that finds things for you to do based on your interests and tastes.

We propose a rating system for activities in your area. Essentially, an activity can be anything ranging from a place to grab food, a place to go shopping, or a trip to a park. To put it simply, we’re going to combine a rating system from sites like Netflix with rating applications like Yelp.

Fundamentally, people enjoy different types of things. So recommending activities based on what “most people” like (such as Yelp) does not necessarily lead searching users to activities they’ll enjoy. Specific interests / parameters can drastically change the recommendation. Thus we plan to make recommendations based on ratings of similar types of activities.

**Data**

Ideally, we would be able to populate our databases with real user data. To this end, we’re creating a website that is slowly being populated. In order to evaluate our model, we can leverage existing ratings and information in internet databases (including Yelp, Google and FourSquare) to “seed” our databases with ratings. In order to do this, we’re going to create a web-crawler that will scrape businesses and user ratings from existing internet database. For each internet database, we will translate their rating system to ours (we intend to use a like / dislike approach, but an X star rating system is also something we might explore) and insert the data as if we had collected it ourselves.[[1]](#footnote-1) This will be customized for each database, but since there are only a couple of possible rating systems (and most use a 5-star approach), this is fairly simple. We can evaluate on, essentially, an arbitrary number of users / businesses. To start, we’ll evaluate on several dozen users / businesses, refine the model, and then continually expand until we feel the results are indicative of a realistic workload (we estimate this will be around several thousand users and businesses).

**Methods**

The method we will use is the Pearson correlation. This determines a relationship between the user we are trying to recommend to (we will call this the user) and other raters that have ranked similar things to that user. The approximate steps of applying this in the most naïve way possible are: 1) Determine the relationship between each rater and the user. 2) Select all users above a certain relatedness threshold. 3) Collect the average rating that similar raters gave to businesses the user hasn’t rated yet. The recommendations indicate “users with similar tastes to you liked this business ‘N’ percent of the time” where n is the average calculated in step 3. As described, the algorithm is inefficient, especially for a large number of users. One of our goals in the project will be to make this algorithm run quickly so that new recommendations can be calculated in a reasonable amount of time. We will look into steps like only selecting a subset of the data that might still give us good results.

There are also modifications that we will need to make to get the most accurate results. The threshold in step 2 will have a great effect on the accuracy of the results. If it is too high, we might not get a good sample size very often. If it is too low, we might not use users that are related enough to the querying user. We also need to decide what do to with users that have a high negative relationship to the querying user. Will these users still be reliable for making predictions? Will they be reliable at the same threshold as positive relationship users will?

Overall, we will explore the Pearson correlation during the project, and refine it so that the best recommendations end up on the front-page for the user. The refinements will not only be in ensuring the model becomes more accurate, but that the calculations are feasible (e.g. do we favor a slightly less accurate model if the more accurate one is much more computationally intensive). In our final writeup, we will discuss our tradeoffs, as well as any successes or failures we faced in applying Pearson.

**Evaluation**

To evaluate our project, we will employ the methods discussed in class. More concretely, we plan to keep 20% of the dataset as a testing set. The model will then be fit using five fold cross validation on the remaining 80% of the data.

The natural measure of success is how well we can predict the held out users ratings of the various places and activities in the data set.

**Contingency Plan**

If scraping the various sites proves to be too difficult, we have a few options for gathering the data.

* We can generate data by creating classes of people with various levels of ratings for different business types, then generate a dataset by randomly selecting a user class and filling in ratings by adding a Gaussian term to the base rating level.
* Generate the data using real people (friends, families, victims) or potentially something like Amazon’s Mechanical Turk

1. Note, this is *only* for testing the validity of the model. There might be legal ramifications in a live web application [↑](#footnote-ref-1)