Are.na Recommendation System

scra.per

Goal

Are.na is an online social networking community where users can save content, create collections, and connect ideas by following and collaborating on channels. Since users tend to only be interested in certain topics, this project is designed to recommend channels to users who are likely to contribute to them. The use case for this project is that the recommender system can help users find related channels of interest quickly with less effort. Our goal is to generate recommendations by predicting which Are.na channels users are likely to contribute to, based on current data for user collaboration on channels.

Data

We used Are.na's developer API to fetch data and process it into a csv containing every channel id mapped to a list of user ids which have collaborated on the channel, where each list has a length of at least 2. A collaborator of a page either owns the page, or has added content to the page after its creation. For this reason, it was safe to filter out channels with less than 2 collaborators, which would indicate a channel which no user other than the owner has added to.

Of 309,535 total channels, 10,123 channels have 2 or more collaborators (including owners). The distribution of collaborators per channel is visualized in the long tail plot included on our poster. The distribution of collaborators is strongly skewed; less than 10% of channels contain over 50% of total collaborators.

1012

Model+Evaluation Setup

We performed truncated singular value decomposition (SVD) on our data to construct an adjacency matrix with each row corresponding to a user and each column corresponding to a channel. For each entry of the adjacency matrix M, there is a 1 if that particular user contributed to the channel and a 0 otherwise. To test how good our model is at predicting which users will collaborate on which channels, we created a test matrix by removing 10% of all adjacencies equivalent to 1's from the original channels-collaborators matrix. Our model then predicted the missing values by reconstructing estimations of the original matrix by multiplying together a truncated U and V matrix achieved by SVD.

As our primary accuracy evaluation metric, we considered the proportion of values in the predictions that are greater than a certain threshold to the actual values. We also calculated the Root Mean Squared Error (RMSE) to measure the average deviation of the predictions from the actual values in the original channels-collaborators matrix to measure the accuracy of our recommender system. To evaluate the usefulness of our recommendations, we looked at the R-score and a Long Tail Plot. The R-score postulates that items recommended later in a list are exponentially less likely to be consumed by a user. The plot explores popularity patterns in the data, which is helpful because recommendations of the already popular channels marked by the "head" are not likely to be relevant to most users.

Results and Analysis

We hypothesized that after the model is trained using T as an input matrix, the resulting $\hat{T}$ should have a high prediction value on the test set (where $M \neq T$), as the test set contains known adjacencies that were masked during training. $\hat{T}$ is normalized by the largest value for each user (such that each user has at least one prediction score of 1), and the recommendations derived from $\hat{T}$ are defined as: $rec[i][j] = \{1 \; if \; \hat{T}[i][j] > t, \; 0 \; else\}$, where t is a test threshold.

Claim #1: The recommender system trained using 90% of channels-collaborators adjacencies outperforms the baseline models by a significant margin.
Support: The accuracy is measured as the proportion of values in $\hat{T}$ that are greater than a threshold t to the actual values. The accuracy with $\hat{T}$ normalized by the largest prediction value per user and $t = 0.5$ is .137 for entries that are labeled as "1" values in the M matrix. On the other hand, if we select indices from random entries that are not shown to correspond to "1" entries on the M matrix, we get an accuracy of 0.001.

Claim #2: Our most confident recommendations are much more "useful" than our least confident recommendations, in terms of the probability of recommended channels actually being contributed to.
Support: We used the expected utility score, or "R-score," to evaluate the success of our recommendations. R-score estimates the utility of a sorted list of recommendations by accounting for a user's "patience", or the half-life of a recommended item's relevance to the user. In other words, the formula postulates that items recommended later in a list are exponentially less likely to be consumed by a user. While this metric doesn't necessarily speak to the accuracy of our model or provide us with a statistic that is easily interpretable, we found that the lowest R-score above 0 for a single user is 0.15, and the highest is 0.19. By the definition of the metric, this means that our most confident recommendations are 26.7% more "useful" than our least confident recommendations in terms of the probability of recommended channels actually being contributed to.