Are.na / Recommender System

for creative content collaboration

Michael Coppolino, Spencer Small, Xavi Loinaz, Suhye Park

Scra.per

Motivation

Are.na is a visual content organization platform allowing users to build simple collections of content by adding links and files to channels. Channels can be collaborative and interconnected, in that one can function as a block of content to another, creating a dense web of connectivity.

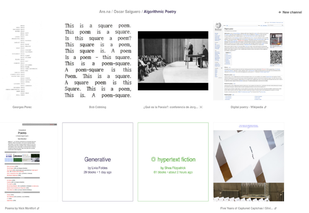

Figure 1: *Algorithmic Poetry*, an example of a multimedia channel containing links, images, text, and other channels.

Are.na currently lacks an "explore" feature, which would radically strengthen and grow the network of connections that already exists on the site. Thus, we originally wanted to create a recommender system that suggests new channels to users, but we were faced with a privacy-related restriction in doing so. Instead, **our tool now recommends Are.na channels to users who are likely to contribute to them** (i.e. add their own content to a channel, based on the channels they have contributed to before). This tool would help grow both popular and undiscovered channels by suggesting them to relevant contributors who may never find them otherwise.

Data

We collected our data on over 300,000 channels directly from Are.na's API using the API calls for channels and collaborators and found 10,123 with two or more collaborators. For each channel marked by its unique ID, we collected its collaborators' user IDs in a list. We also gathered data on each of the channels' owner and generated two separate CSV files named "collaborators.csv" and "collaborators_with_owners.csv." We then created dictionaries mapping the order that channels and users were read in (indices) to their IDs. Finally, we set up a matrix with dimensions given by the number of channels with more than two collaborators, and the total number of users who collaborate.

For each entry of the matrix corresponding to a certain user and a certain channel, there is a 1 if a user contributed to that channel, and a 0 otherwise. **With the huge number of channels each having relatively few collaborators, our dataset is very sparse.**

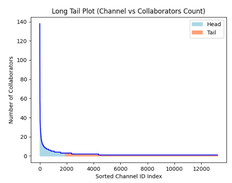

Figure 2: Distribution of channels by number of contibutors. This plot explains the variability in average prediction values as number of collaborators per channel increases. Over 80% of channels have less than 5 collaborators.

Results

The recommender system trained using 90% of channels-collaborators adjacencies outperforms the baseline models by a significant margin. The accuracy is measured as the proportion of values in T that are greater than a threshold t to the actual values: ``

$$Accuracy = \sum \frac{\hat{T}[i][j] > t}{M[i][j]} \; \forall \; i, j \text{ s.t } M[i][j] \neq T[i][j]$$

Due to the sparsity of our data and the nature of SVD, most predictions are below 0.5 (50% confidence), so it was necessary to include a threshold value after normalizing to ensure a quantity of recommendations.

The accuracy with T normalized by the largest prediction value per user with $t = 0.5$ is 0.137. For comparison, the model that recommends the most popular channels has an accuracy of 0.011, and the model that randomly recommends channels has an accuracy of 0.001. This first value is exceptionally low due to the niche nature of Are.na, where even the most popular channels are really not that "popular". Thus, while this metric would indicate success for our model, a model with no recommendations (our original dataset) would trivially have an accuracy of 1.0, so additional assesments of accuracy are considered.

Our most confident recommendations are much more "useful" than our least confident recommendations, based on the R-score from certain users. 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 (Shani, Gunawardana 23). In other words, the formula postulates that items recommended later in a list are exponentially less likely to be consumed by a user.

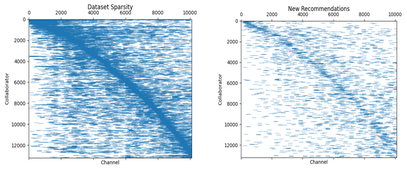

Figure 3, 4: The graph on the left represents our 2D channel-collaborator matrix with an increased marker size to show the relative sparsity of our entire dataset (the M matrix), when compared to the graph of new recommendations generated by the model with the same marker size (right). There is a dense diagonal region in the graphs, likely because collaborators who join the site create and collaborate on channels that are new as well.

We found that the lowest R-score above 0 for a single user is 0.15, the median is 0.16 the highest is 0.19. By the definition of the metric, this means that our most confident recommendations are significantly more useful than both our least confident and median recommendations.

Clustering channels based on embeddings produced sensible and novel groupings. Eight distinct clusters formed from running t-SNE and K-Means on channel embeddings, which reveal general topics on Are.na, as observed in the following table:

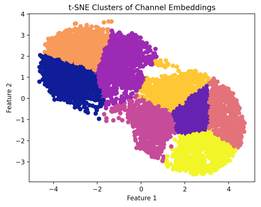

Figure 5: Clusters of channels that were dimensionally reduced using t-SNE. As shown in Table 1, we can observe channels from each cluster and determine whether or not they seemed related by our qualitative judgement to ensure our recommendation system has the data to recommend related channels to collaborators properly.

Cluster #	Channel Title	# Collaborators
0	Non-human Phenomenology	4
0	Alm data sources	3
0	Defining Love	2
1	3D Compositions	2
1	tile fabrication	2
1	Lido Deck Furniture	2
2	transcendental punk (WA)	3
2	No Parking	5
2	Missing / Lost / Found	3
3	Nemesis Protocol	3
3	Ethics in AI	3
3	RISD Museum Raid Inspo	3
4	UI	3
4	Layout Images	4
4	Editorial	4
5	Misc. Okimoto Images	3
5	landscapes in virtual space	16
5	Mixed Media Art	2
6	Avatar Inspo	2
6	Books - children vs death	2
6	VR & Narrative	2
7	nice places	3
7	Motels Hotels	2
7	SCC1 – Dissolving Hierarchies!?	3

Table 1: This table shows three channels for each cluster as a sanity check, along with the collaborator count for reference. While quirky, the given channels for a cluster seem to be related, meaning our model

Methodology

We used truncated singular value decomposition (SVD) to predict which users will contribute to certain channels by generating predictions for 'missing' adjacencies in M, and t-Distributed Stochastic Neighbor Embedding (t-SNE) to later analyze the embeddings of channels and users found in the truncated U and V matrices, respectively. To accomplish this, we first performed SVD on our M matrix to derive recommendations for potential channel-collaborator pairings. After M is decomposed into U, D, and V, U and V are truncated to reduce the dimensionality of the feature space, which yields an approximation of M (M_hat) when these truncated U and V matrices are multiplied back together. The resulting M_hat is no longer binary {0,1} and instead is filled with floats on [0,1] which can be interpreted as probabilities that a 1 should exist in the original M, i.e. that a user should collaborate on a channel. **More specifically, M_hat[i][j] can be interpreted as the probability that M[i][j] should be a 1.** The U and V matrices that are returned by SVD on M were truncated to a size determined by the singular values contained in D.

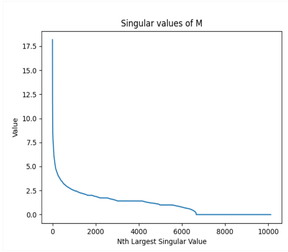

Figure 6: The distribution of singular values for SVD for our original dataset (M matrix). Singular values are ordered in descending fashion. The vast majority of singular values are below 3, but can go up to 18.

The values of D indicate the influence that each dimension has on M, and by removing the lower weighted dimensions and keeping the dimensions with higher weights, we can ensure that no important information is lost when encoding M_hat. **The truncated length was determined by locating the elbow point** of a line plot of the values of D (sorted by descending value by default) via inspection.

To test the results of the model, we created a test matrix T, which is a copy of M with 10% of adjacencies (M[i][j] = 1) removed. **We hypothesize that after the model is trained using T as an input matrix, the resulting T_hat should have a high prediction value on the test set** (where M != T), as the test set contains known adjacencies that were masked during training. T_hat 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 T_hat are defined as: rec[i][j] = {1 if T_hat[i][j] > t, 0 else}, where t is a test threshold.

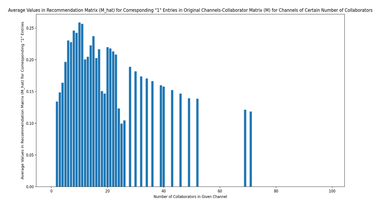

Figure 7: This plot demonstrates that the entries from M with value 1 retain high values after Truncated SVD, indicating that the M_hat matrix retains its general essence from the original data. Additionally, it shows that channels with around 10 collaborators tend to be the strongest predicted channels for