Recommending Subreddits to Reddit Users

Jessamyn Liu
jessamyn.liu@gmail.com

Quanquan Liu
quanquan@mit.edu

Abstract

In this paper we discuss the challenges of designing a recommenders system for Reddit. We implement both collaborative filtering and content based approaches, [evaluating the effects of various normalization techniques and similarity metrics].

1 Introduction

Reddit is a prominent social media platform which uses a bulletin board style website for users to share content with other users. In order to help users organize into communities of interest, Reddit hosts over 850,000 subreddits, with hundreds more being created each day. A recommender system which finds related subreddits would help new users quickly explore similar subreddits and may help existing users find unexpected subreddits related to their interests. Alternatively, marketers may be interested in finding all subreddits related to their niche area to subscribe to or make posts in to promote their product. A robust recommender system may even help uncover communities participating in illegal activity which would otherwise seek to remain hidden, assisting authorities who are seeking to shut down subreddits that perpetuate hate speech or enable crime (such as child pornography, theft of digital media, or trafficking of illicit substances).

Recommender systems are broadly grouped into two strategies. Content based systems focus on measuring similarity in the features of the item itself. In contrast, collaborative filtering uses user feedback, either explicit (e.g. star ratings on Netflix) or implicit (e.g. a user's viewing history), to find similarities between like-minded users. The use of collaborative filtering methods based on implicit user feedback has grown in popularity with the increased passive tracking of user behavioral data, but can present challenges such as sparse and/or noisy data, ambiguous data interpretation (no negative ratings, mapping of behavior to preference may be unclear), and the "cold start" problem (unable to evaluate new users or items).

2 Data

2.1 Data Collection

In this paper, we focus on memory-based, collaborative filtering methods using implicit user feedback, although we attempt a limited implementation of content based, text similarity recommendations for a smaller subset of subreddits. We use Reddit post and comment data from August 2016 made publicly available on Google BigQuery (7,591,689 posts and 69,654,819 comments by 3,698,088 unique users on 100,279 subreddits).[1] From this data set, we derived the following features:

Number of Posts There are two ways in which users can contribute content to Reddit, either by making a top-level post or by commenting on a post. The number of top-level posts a user makes to a subreddit can be a proxy for the degree to which that user is interested in that subreddit.

[1] https://bigquery.cloud.google.com/table/fh-bigquery:reddit_comments.2016_08, https://bigquery.cloud.google.com/table/fh-bigquery:reddit_posts.2016_08

6.867 Final Project (Fall 2016), Cambridge, MA.

Number of Comments Similarly, number of comments can be used as an implicit indicator of user interest. Comments are much more common than posts (by almost 10 to 1) and some users who never originate posts are nonetheless active commenters, so comments may provide more complete information on user preferences/behavior.

Length of Comments A raw count of posts or comments does not give any sense of the post/comment quality . Length of comments could be one way to gain additional insight into the "value" of a comment, with the assumption that longer comments may indicate a higher degree of interest. However, shorter comments do not necessarily indicate a lesser degree of interest, as brevity on Reddit is common and some subreddit cultures tend toward a short comment style. Since post titles have a character limit of 300, the length of post titles would not be an indicator of interest.

Post Scores All posts and comments have a score (or number of points) based on the number of upvotes or downvotes it receives. Reddit uses this score to determine which posts rise to the top of a subreddit home page or the top of the comments listed underneath each post. The total number of points a user has received from all his or her posts on a particular subreddit can be a proxy for the "quality" or value of his or her contribution to that subreddit (in the opinion of other members of that subreddit community). We infer that a user makes higher quality contributions to subreddits which he or she is more interested in.

Shared Users We infer than subreddits which share a greater number of users (based on posting and commenting history) are more related, and for each subreddit pair we calculate the percent of users who posted or commented on both subreddits in August. We use these results to conduct item-based collaborative, expecting that items will be easier to characterize than users (there are likely more similarities between two basketball subreddits than two basketball fans). Since items are typically less dynamic and fewer in number than users, item-based collaborative filtering can also help to address scalability issues.

Text Similarity In this content based approach, we consider the titles of posts in each subreddit and use term frequency-inverse document frequency (TF-IDF) to estimate similarity (see Section 2.3); similar methods could be applied to subreddit comments. Using the content of subreddits to generate recommendations would help to address the "cold start" problem, allowing users who do not have a post or comment history to find similar subreddits. Although a new subreddit would not be evaluated until it has built up a post/comment history, this limitation is not necessarily negative, as it is not meaningful to recommend a subreddit which lacks any content.

2.2 Data Preprocessing

The Reddit data set is both large and inherently noisy, challenges common to many recommender systems which rely on implicit feedback. In order to reduce data set to a more manageable size, we first removed known bots[2] and default subreddits[3].

After reducing the data set, the next critical step was to normalize the data in order to account for differences in user behavior. For example, in the August 2016 Reddit data set, users made anywhere from 1 to over 10,000 comments in one month. In order to compare across users and subreddits, we experimented with a number of different normalization, weighting, and segmentation techniques.

Gaussian Normalization In order to account for differences in a user's average behavior (some users may contribute a large number of comments whereas others may only make a few comments) and variance within user behavior (some users may post about the same amount on all subreddits whereas others may disproportionately post to a few subredits), we apply a Gaussian normalization,

[2]Bots were identified by manually checking all those users with >3000 comments in one month and also by scraping /r/botWatcher for the names of all bots mentioned in the 500 most recent posts. There are inevitably bots that remain unidentified in our data set, however we identified 955 bots (190 active) which accounted for 9.2% of total posts and comments in August 2016.

[3]There are 49 default subreddits which all users are automatically subscribed to when they sign-up for a new account. These subreddits are also linked at the top of every Reddit page and accounted for 17.6% of all posts and comments in August 2016.

subtracting each feature by the user's mean for that feature and dividing by the standard deviation:

$$\hat{F}_b(a) = \frac{F_b(a) - \bar{F}_b}{\sqrt{\sum_a (F_b(a) - \bar{F}_b)^2}}$$

where $\hat{F}_b(a)$ is the feedback for subreddit a by user b and $\bar{F}_b$ is the average feedback for user b.

Log Normalization We also experiment with log normalization ($\hat{F}_b(a) = 1 + log(F_b(a))$) with the intuition that as a user's activity on a subreddit increases, the additional information gained by that increased activity diminishes. For example, if user B makes 60% of his posts to subreddit A, he is likely more interested in subreddit A than user C, who makes 30% of his posts there, but not twice as interested.

"TF-IDF-like" Weighting In order to account for the fact that user activity on an extremely popular subreddit is not as informative as activity on a more niche subreddit in characterizing a user's preferences, we experimented with weighting our features with an "IDF-like" term which is given by $log(\frac{N}{n_s})$ where N is the number of users who posted or commented to any subreddit and n_s is the number of users who posted or commented to the subreddit s. TF-IDF is discussed in greater detail in the next section, but this term should have the effect of reducing the significance of a user's posts/comments on subreddits which have a large number of other users posting on it.

Segmentation Finally, we segmented our data set by the number of subreddits which each user posted or commented to (for example users who post to 2-4 subreddits, 4-15 subreddits, etc). Users who are active on only a few subreddits may lead to abnormally high ratings and skew the similarity measures (a user who posts the same amount to two subreddits would give each subreddit his maximum rating). By segmenting our data, we were able to select the best recommender for each category of user. We believe that users who comment or post to approximately the same number of subreddits are more similar to each other than to users who post in significantly different number of subreddits. Furthermore, we would like to see whether different similarity measures and parameters will result in the best recommendations for different segments. If this is the case, then, one could potentially first categorize an user based on the number of different subreddits she has posted to and then use that segment's recommender to recommend similar subreddits.

2.3 Text processing

Preprocessing text and generating text-based numeric features required additional considerations. In order to show proof of concept, we created a smaller data set from the post titles of the 5,000 most posted to subreddits that had less than 100,000 posts in August 2016 and were not default subreddits.[4] We then used a combination of regular expressions and packages from the Natural Language Toolkit (NLTK)[5] to create a "bag of words" for each subreddit by removing all non-alphanumeric characters and capitalization, filtering for stop words, tokenizing those words that were greater than two characters, and applying a Porter stemmer.

We use the TfidfVectorizer package from sklearn[6] to transform our bag of words to a matrix of TF-IDF features. TF-IDF is a well-known information retrieval weighting scheme which represents how significant a term is to a document in relation to how common that term is in the document and how rare that word is in the larger corpus of documents. In our case, a document is the collection of all cleaned and tokenized titles posted on a single subreddit in August 2016 and the corpus of documents is the entire set of titles for all subreddits.

We apply log normalization to the term frequency ($tf_{t,d}$) to account for our belief that multiple occurrences of a term have less significance as the number of occurrences increases. The log

[4]This data set resulted in subreddits with between 163 and 73,577 posts in August 2016. We were interested in active subreddits, but did not want to be overwhelmed by subreddits with an unusually high number of posts. We selected post titles because each post title is limited in length and therefore easier to compare across subreddits and more computationally manageable.

[5]http://www.nltk.org/

[6]http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

normalized term frequency ($\hat{tf}_{t,d}$), inverse document frequency ($idf_{t,N}$), and TF-IDF ($tfidf_{t,d,N}$) value for term t in document d are given by

$$\hat{tf}_{t,d} = \begin{cases} 1 + log(tf_{t,d}), & \text{if } tf_{t,d} > 0 \\ 0, & \text{otherwise} \end{cases} ; \; idf_{t,N} = log(\frac{N}{df_t})$$

$$tfidf_{t,d,N} = \hat{tf}_{t,d} \times idf_{t,N}$$

where N is the number of documents in the corpus and df_t is the number of documents in which the term t appears.

We further apply $L2$ (Euclidean) normalization to the TF-IDF values such that $\hat{v} = \frac{v}{\|v\|_2}$. Using this matrix of TF-IDF features, we compute pairwise cosine similarity by applying a linear kernel to the already $L2$-normalized data.

3 Recommender Architecture

To build our recommenders, we use the Java package Mahout[7]. The code used to test the various algorithms in our package and to implement our subreddit recommender can be found on our Github repo page[8]. The specific architectures that are used are the Mahout collaborative filtering algorithms. We use the following set of collaborative filtering similarity algorithms: Pearson Correlation, Euclidean Distance, Log Likelihood, Tanimoto Coefficient, and Spearman Correlation. Once these similarities are calculated, we further use a neighborhood algorithm to determine the nearest neighbors. This algorithm either finds the nearest n neighbors or uses a threshold similarity where if the similarity of a specific user is greater than a score, then we include the items liked by these users. We briefly explain the pitfalls and advantages of each of these algorithms below. Furthermore, we evaluate the accuracy of these algorithms by performing a series of hold-out experiments both by using the evaluator code provided by the Mahout package and by implementing our own testing suite.

3.1 Similarity Measures

Pearson Correlation [9]

The Pearson Correlation similarity measure looks at how similar data variation curves are from each other. Specifically, given two items, we compute the following as the Pearson Correlation:

$$r_{X,Y} = \frac{\sum_{x_i, y_i \in X,Y}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}}$$

where $x_i \in X$ and $y_i \in Y$ are data samples of X and Y respectively, n is the total number of samples, and $\overline{x}$ and $\overline{y}$ are the means of all samples in X and Y. In terms of users and item preference ratings, X is the set of item preferences of user X and Y is the set of item preferences of user Y.

One advantage of this similarity measure is that the data is normalized to look at proportionality trends between pairs of user preference vectors (essentially looking at the trends and the tendency of the values to move together proportionally) rather than the actual values. For instance, points on two parallel lines have Pearson correlation 1. Therefore, this similarity measure is robust against user preferences that have different means and variances among different users. For instance, for users who posted to more than 25 subreddits, the Pearson correlation similarity measure performed much better than all other similarity measures.

The major disadvantage of this similarity measure is that users who only rated one item or rated all items the same rating will have covariance 0 which prevents any Mahout collaborative filtering algorithm based on this measure from making a recommendation. Other similarity measures mentioned below will be better able to handle such sparsity of data. We see this effect in our results for data that includes users who have only made comments in $2 \leq n \leq 3$ subreddits (see Section 4.4.1). Using

[7] https://mahout.apache.org

[8] https://github.com/qqliu/subreddit-recommendations

[9] https://archive.cloudera.com/cdh4/cdh/4/.../similarity/PearsonCorrelationSimilarity.html

Pearson correlation, the predicted values matched poorly with the actual values. Furthermore, by removing the top ranked item, the data becomes too sparse for the recommendation algorithm to retrieve any meaningful recommendations (at least by the precision and recall measures).

Euclidean Distance [10]

This measure assumes items are dimensions and each user is a point in R^n where n is the total number of ratable items. Then, for any two users, we compute the Euclidean distance between the two users. The similarity r between two users, X and Y, is calculated by the following equation:

$$r_{X,Y} = \frac{\sqrt{n}}{1 + \sqrt{\sum_{x_i, y_i \in X,Y} (x_i - y_i)^2}}$$

where the $\sqrt{n}$ is meant to help correct against users with more ratings automatically receiving lower scores. (In fact, more similar ratings should indicate *greater* similarity between users.) Unlike the Pearson correlation, the Euclidean distance metric does not have a problem with sparse data sets, although this similarity measure performs better when all points have approximately the same number of dimensions.

Spearman Correlation [11]

This measure is the same implementation as the Pearson correlation similarity except all preferences are ranked and the rankings are used to compute the correlation value.

Log Likelihood [12]

This similarity measure computes the log likelihood of two ratings appearing together using the following equation: given two items, i and j, we can compute the number of users that have ratings for i and j, $k_{i,j}$, the number of users who rated i but not j, $k_{i,\bar{j}}$, the number of users who have rated j but not i, $k_{\bar{i},j}$, and the number of users who have rated neither i or j, $k_{\bar{i},\bar{j}}$ (i.e. the values are in a coocurrence matrix, K). The log likelihood of i and j occurring together is then:

$$r_{i,j} = 2 \left(\sum_{l \in i,\bar{i}, m \in j,\bar{j}} k_{l,m} \right) (H(K) - H(K_i) - H(K_j))$$

where $N = \sum_{l \in i,\bar{i}, m \in j,\bar{j}} k_{l,m}$, $H = \sum_{l \in i,\bar{i}, m \in j,\bar{j}} (k/N \log(k/N))$, and K_i is a vector of row sums of K and K_j is a vector of column sums.

The main advantage of this measure is that it allows for similarity measures based on boolean preferences or user interactions with items that do not have clear preferences. In fact, this similarity computes how *unlikely* it is for two probability distributions to be independent (i.e. how likely they are dependent), so higher values indicate greater dependence and greater similarity.

Tanimoto Coefficient [13]

The Tanimoto coefficient is another similarity measure meant for boolean data. Intuitively, it computes the fraction of items that are rated by both users over the total number of items that are rated by either or both users. If two users interacted with more of the same items, then they are more likely to be similar in tastes. This measure computes the fraction that preference user X and user Y share a relation with the same items:

$$r_{X,Y} = \frac{\sum_i X_i \wedge Y_i}{\sum_i X_i \vee Y_i}$$

This is a simpler similarity than the log likelihood measure and returns a value in $[0, 1]$.

[10] http://archive-primary.cloudera.com/cdh4/cdh/4/mahout-0.7-cdh4.3.2/mahout-core/org/apache/mahout/.../similarity/EuclideanDistanceSimilarity.html

[11] https://archive.cloudera.com/cdh4/cdh/4/mahout-0.7-cdh4.1.5/mahout-core/org/.../similarity/SpearmanCorrelationSimilarity.html

[12] https://archive.cloudera.com/cdh4/cdh/4/mahout-0.7-cdh4.5.0/mahout-core/org/.../similarity/LogLikelihoodSimilarity.html

[13] https://archive.cloudera.com/cdh4/cdh/4/mahout-0.7-cdh4.5.0/mahout-core/org/.../similarity/TanimotoCoefficientSimilarity.html

Uncentered Cosine If we imagine a user list of preferences for items is a vector in space, then the uncentered cosine similarity computes the cosine of the angle between these vectors. If the data is normalized, then this similarity measure calculates the same number as the Pearson correlation measure. Therefore, we only present a few results in Section 4 related to this measure.

3.2 User Neighborhood Algorithms

Nearest N Neighbors This returns the nearest N users (i.e. users with the highest similarity to the current user) and bases recommendations on the nearest N users. In our evaluations, we vary the neighborhood size to determine the optimum neighborhood size.

Threshold-based neighborhood The threshold-based neighborhood algorithm picks all users who have similarity greater than threshold k. All of these user's preferences are then factored into the computation of the recommendations.

4 Results

We use two forms of quantitative evaluations to determine the effectiveness of each of the methods we test. The two quantitative methods we use are two types of holdout experiments. In the first instance, we hold a random portion of the training data set as the testing set. Then, we compute the predicted preference values for each of the item-user pairs that was "held-out" and compute the error between the predicted preference values and the expected preference values (i.e. the actual preference values). We also compute the precision and recall of each evaluation based on how many of the returned values were held-out. Let p be the precision value and r be the recall value:

$$p = \frac{|\{\text{relevant items}\} \cap \{\text{retrieved items}\}|}{|\{\text{retrieved items}\}|}$$

$$r = \frac{|\{\text{relevant items}\} \cap \{\text{retrieved items}\}|}{|\{\text{relevant items}\}|}$$

where $\{\text{relevant items}\}$ is the set items that were "held-out" by the hold-out experiment and $\{\text{retrieved items}\}$ is the set of items returned as recommendations to users by the algorithm.

Here and throughout, we will use the following terms and notations: N is the number of users used in the Nearest N-Users neighborhood algorithm, T is the threshold in the Threshold algorithm for determining user neighborhoods, and H is the top H-th ratings held-out or removed from the training dataset and placed in the testing dataset to determine precision and recall. All file abbreviations can be found in Table 1.

4.1 Preference Values and Feature Mappings

Here, we explain our test results for the various feature mappings we used in Section 4.2. For each type of feature mapping, we test all algorithms mentioned in Section 3 by varying user similarity measures and nearest neighbor parameters and threshold values. For these evaluations, we perform both types of quantitative evaluations. For determining the error between predicted preferences and the actual preferences based on the feature mappings, we use the root-mean-square error calculation and the average absolute value difference. We use the RMS error calculations to compare similarity measures with each other to determine the best similarity measure to use for each feature mapping dataset. Then, we use the precision and recall values to compare effectiveness of different feature mappings and normalizations with each other. The statistics of each dataset is shown in Table 1.

We compare our results to a randomly generated file with $2,000,000$ lines. For the random file, we choose random user and item pairs such that the expected number of ratings per user is 10. We created a set of $20,000$ items where the items for each user is chosen uniformly at random from the set of total items. Furthermore, we compute a rounded two-decimal rating from a Gaussian distribution with mean 0.7 and standard deviation 0.17, similar to the mean and standard deviation for the features derived from the number of posts. Using this file of random user ratings and item pairs, none of the models are able to train, meaning that all the files we provide have some form of correlation/are able to train a model whereas the random file cannot.

File Type	Mean of User Means	Standard Deviation of User Means	Feature Value − User Mean	RMS of Feature Val − User Mean
Num Comments (NC; 7632154)	0.587	0.260	0.209	0.279
Num Posts (NP; 1603887)	0.713	0.189	0.137	0.213
Num Posts TF-IDF (NP-TFIDF; 1497880)	0.771	0.161	0.176	0.220
Num Comments TF-IDF (NC-TFIDF; 4715952)	0.713	0.189	0.216	0.262
Avg Comment Length (ACL; 7632044)	0.594	0.187	0.229	0.283
Posts Score (PS; 1434089)	0.638	0.238	0.275	0.336

Table 1: Details about each file. We compute the mean and standard deviation of the mean of each user vector, the mean difference between each user feature value and the mean of that user's feature values, and the root mean square of the different between each user feature value and the mean of the user's feature values. Each file is normalized by dividing by the maximum feature value for each user.

4.2 Feature Mapping and Recommendation Quantitative Results

We considered the 6 different feature mappings in this section. For each of these feature mapping procedures, we test the 5 similarity measures, Pearson Correlation, Euclidean Distance, Log Likelihood, Spearman Correlation, and Tanimoto Coefficient to determine the root-mean-square error between the predicted preference of each user for a given subreddit that was removed from the input file of ratings by the hold-out test. The lower the error score, the greater the chance that preferences calculated by using the similarity measure matches preferences expressed by the user *within a specific feature map*. We use the precision and recall to determine how well the recommender predicted subreddits the user *already interacts with* using this particular feature mapping. Note that since we use 0.9 of the data for training and 0.01 of the data for testing RMS error and 0.0001 of the users for testing precision and recall, there could be variations in the number of users that are tested depending on how many users and how many interaction data there is. The number of users and amount of item (subreddit) data used for testing can be found in Table 2.

We found through our tests that the feature map that resulted in the greatest precision and recall was the number of posts made to a particular subreddit. As can be seen in Fig. 1, for instance, for 50 neighbors and using Pearson correlation, the precision is 0.5 and the recall is 0.1 which outperforms all other feature maps when using the Nearest N-Neighbors algorithm with a neighborhood of 50 and any of the other similarity measures. For both the NC-TFIDF and NP-TFIDF feature maps, we see a performance that is worse than the original NC and NP datasets in terms of both precision and recall. We believe that one explanation for this observation is that the TFIDF algorithm might not be adding additional information to the dataset beyond what the number of posts within a subreddit tells us. In fact, it could be masking some of this information by weighing factors that may not be as important as user posting frequency. Fig. 1 shows an example testing trial. Table 2 shows more testing results that follows the aforementioned trends described above as well as the noted observations.

We see the results of the best similarity measures for each feature map in Fig. 1. To choose the similarity measure depicted, we use the RMS error to determine the best similarity measure for each dataset and the precision and recall to compare performance across datasets.

We believe that the number of posts performs the best out of all datasets because posting in a subreddit shows greater preference for the subreddit than commenting on a subreddit. Furthermore, users make fewer posts to fewer subreddits resulting in less noise than the comment dataset.

Furthermore, we analyze the effect of different parameters (such as neighborhood size and threshold value) on the precision and recall of recommender algorithms using the Tanimoto coefficient and the

Pearson correlation measure on the number of posts dataset. Here, H represents the top H-ratings removed from 0.0001 fraction which is ≈ 400 users of the training feature map set to be used as the testing set. Despite a few anomalies, we see as a general trend that both precision and recall decreases as the user neighborhood increases beyond a certain point. It appears that $N = 50$ and $t = 0.25$ are parameters that are most favorable in terms of *both precision and recall*.

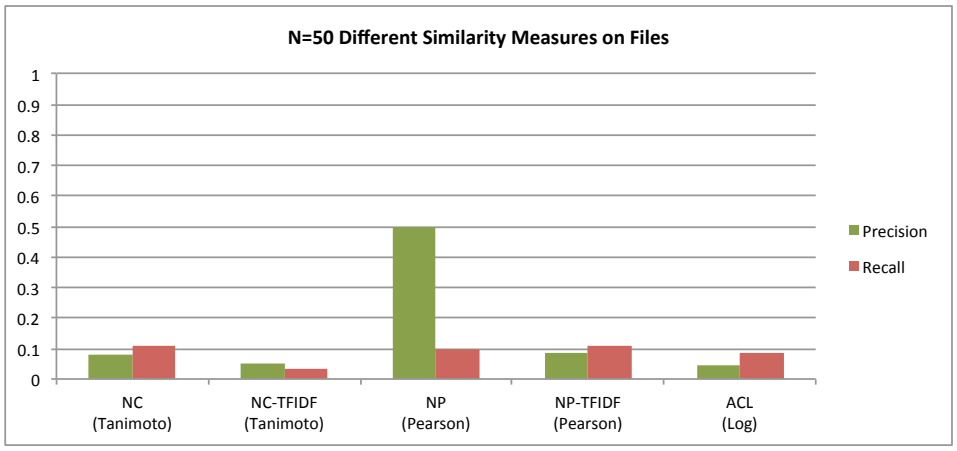

Figure 1: Precision and recall due to different similarity measures that resulted in smallest root-mean-squared error on different feature maps. We see that the Pearson correlation similarity measure on the number of posts data file performs better than other similarity measures for other files in terms of precision and performs not much worse in terms of recall.

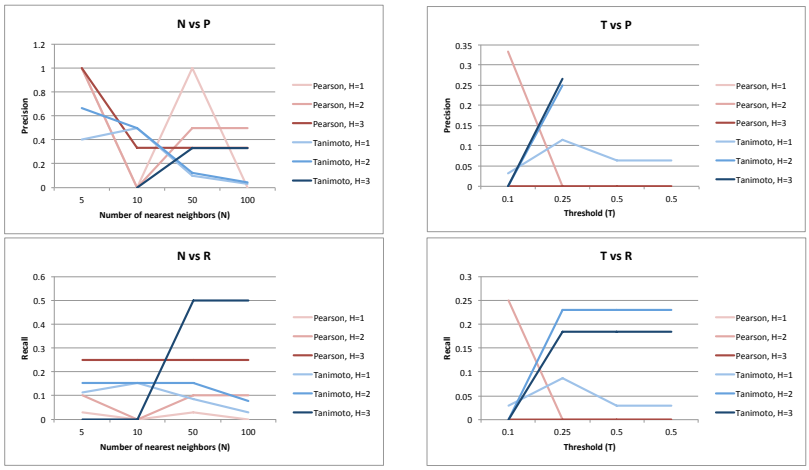

Figure 2: Precision and recall due to different similarity measures: Pearson correlation and Tanimoto coefficient on the file mapping user preferences to the number of posts made in a subreddit. We vary the number of nearest neighbors and the threshold similarity value when computing the neighborhood of similar users.

4.3 Different Normalization Results

Using the normalization methods mentioned in Section 2.2, we see that both NC-TFIDF and NP-TFIDF perform worse than NC and NP in terms of precision and recall. We can see these results in Fig. 3 where we see the precision and recall of NP-TFIDF and NP given user neighborhoods $N = 10, 50$ and varying H. However, we see that the logarithm normalization performs (somewhat surprisingly) worse on most of the similarity measures in terms of precision and recall than the original dataset as also can be seen In Fig. 3. On the other hand, the Gaussian normalization performs as well as the original and better in terms of both precision and recall for measures that do not

normalize the input data–such as Euclidean distance (as expected– since similarity measures as the Pearson correlation measure and other similarity measures that do not normalize as well–such as Euclidean distance–would be expected to do better).

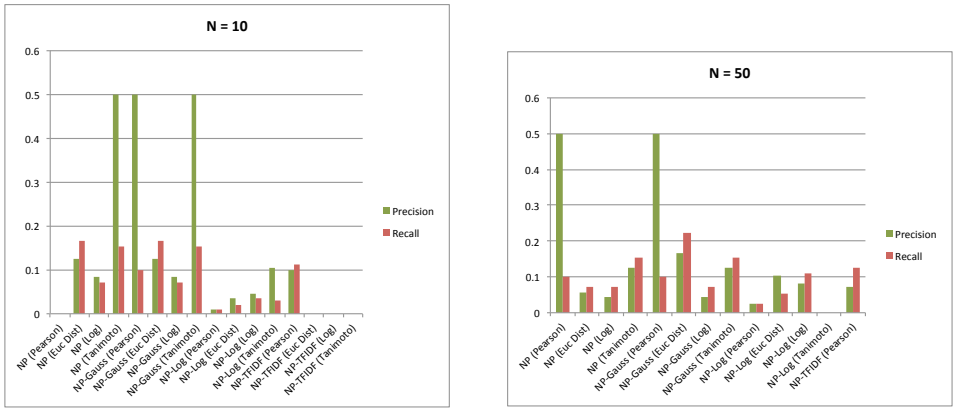

Figure 3: Precision and recall due to different normalization methods on the datasets: NP and NC (number of posts and number of comments).

4.4 Segmentation of Comments and Posts

We find that segmenting the comments and posts into smaller groups and applying a different similarity measure and collaborative filtering algorithm for each segment performed better than feeding a large data set *that is not segmented* into a recommender. For this procedure, we test against the NC dataset since it contains the most number of ratings and the greatest variance in terms of the number of subreddits that users contributed to. We segment the ratings into the following segments: ratings from users who contribute to [2, 3] subreddits, [3, 6] subreddits, [6, 15] subreddits, [15, 25] subreddits, [25, 50] subreddits, and, finally, [50-1000] subreddits. We see that segmenting the dataset resulted in greater precision and recall than the original NC dataset for dataset segments in the range [2, 50]. For users that exhibit bot-like behavior (i.e. post to lots of different subreddits), the results show that the precision and recall rates are comparable to that of using similarity measures on the entire NC file.

Furthermore, different recommender algorithm parameters performed better in each segment as we discuss below.

4.4.1 Users who commented or posted to 2 or 3 subreddits

We only display the results for the Nearest N-Neighbors algorithm in Fig. 4 since the threshold-based algorithm performed much worse (by an order of magnitude) in both precision and recall.

Here, we see that the Tanimoto coefficient performs better than the Pearson coefficient (also better than any other similarity measures) for $N = 10$ and $H = 1$. The Pearson coefficient performs slightly better than the Tanimoto coefficient when $N = 100$ and $H = 2$ in terms of precision. However, the difference is not great enough to warrant choosing Pearson coefficient rather than Tanimoto. Furthermore, the Pearson correlation measure may be slightly misleading here since removing 1 subreddit from a user reduces many users to only have one other subreddit that they interacted with. This means that the Pearson correlation will not return a similarity value for these users since there is not enough data left to compute a correlation. This could potentially explain the low recall returned by Pearson correlation when $H = 1$.

4.4.2 Users who commented or posted to 3 to 6 subreddits

Here, again as seen in Fig. 4, the Tanimoto coefficient performs better in terms of both precision and recall when $N = 10$ and performs slightly worse than Pearson when $N = 100$. The graphs for the case in Section 4.4.1 and here are very similar so we do not offer more insight in the trends of running the tests on this dataset than we offered before.

4.4.3 Users who commented or posted to more than 25 subreddits

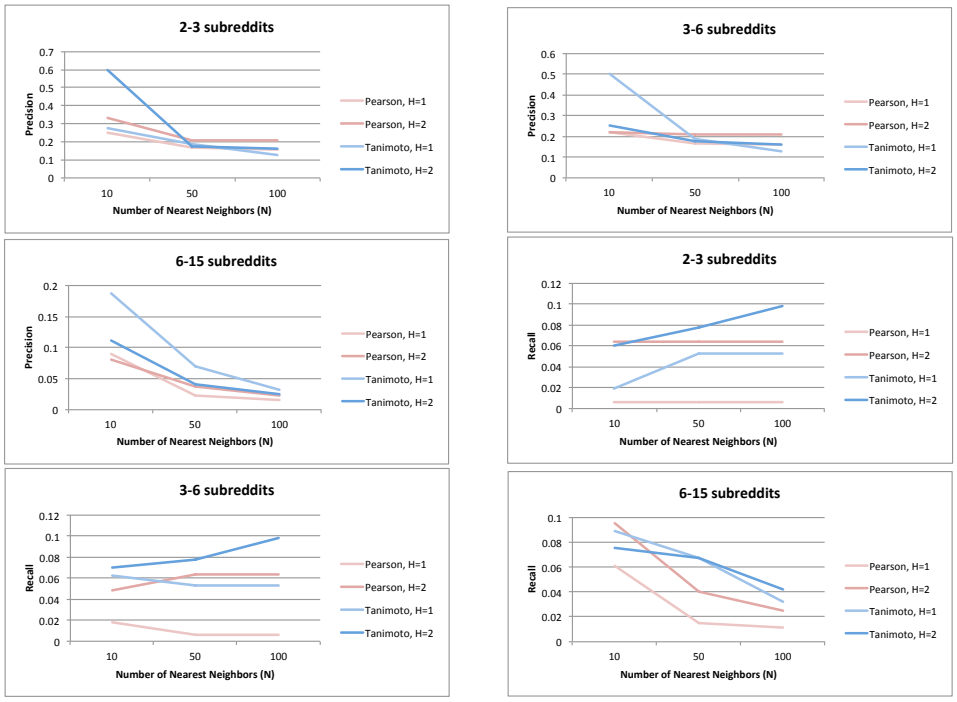

Figure 4: Precision and recall due to different similarity measures and different segmentation.

4.4.4 Users who commented or posted to 6 to 15 subreddits

Here, we see that the Tanimoto coefficient (especially when $H = 2$) performs solidly better in terms of precision and recall than the Pearson correlation. Furthermore, we see that for $N = 10$, it performs much better than using the Tanimoto coefficient on the total number of comments file (see Table 2).

4.4.5 Users who commented or posted to 15 to 1000 subreddits

We do not mention these results explicitly since the values are small (less than 0.1) for precision and do not offer us more insight than the above results on smaller number of subreddits. To see more the actual data please visit our Github repository[14].

5 Next Steps

For our results, we performed on the orders of hundreds of quantitative tests based mainly on precision and recall. However, these measures are very restricting in the sense that we do not test whether actual users prefer such items. In the future, if we continue this project, we retrieved sets of recommendations for users and compare to sets of randomly generated recommenders for to evaluate the similarity between items in the set qualitatively by real people. We can obtain such a population through Amazon Turk or by setting up a web domain and having real users receive recommendations for subreddits based on their user account activity.

Appendix

File	Sim Measure	# Training	# Testing	N	T	H	RMS Err	Precision	Recall

[14] https://github.com/qqliu/subreddit-recommendations

NC	Pearson	4244356	47159	50	N/A	N/A	0.398	N/A	N/A
NC	Euc Dist	4244356	47159	50	N/A	N/A	0.403	N/A	N/A
NC	Log	4244356	47159	N/A	50	N/A	0.399	N/A	N/A
NC	Spearman	4244356	47159	50	N/A	0.404	N/A	N/A	N/A
NC	Tanimoto	4244356	47159	50	N/A	N/A	0.391	N/A	N/A
NC	Log	4244356	47159	50	N/A	N/A	0.301 (Avg)	N/A	N/A
NC	Tanimoto	4244356	4719	50	N/A	N/A	0.316 (Avg)	N/A	N/A
NC	Pearson	4244356	471	50	N/A	2	N/A	0.029	0.0357
NC	Euc Dist	4244356	471	50	N/A	2	N/A	0.016	0.032
NC	Log	4244356	471	50	N/A	2	N/A	0.063	0.041
NC	Spearman	4244356	471	50	N/A	2	N/A	0.046	0.070
NC	Tanimoto	4244356	471	50	N/A	2	N/A	0.078	0.106
NC-TFIDF	Pearson	4244356	47159	10	N/A	N/A	0.394	N/A	N/A
NC-TFIDF	Euc Dist	4244356	47159	10	N/A	N/A	0.342	N/A	N/A
NC-TFIDF	Log	4244356	47159	N/A	10	N/A	0.353	N/A	N/A
NC-TFIDF	Spearman	4244356	47159	10	N/A	0.489	N/A	N/A	N/A
NC-TFIDF	Tanimoto	4244356	47159	10	N/A	N/A	0.350	N/A	N/A
NC-TFIDF	Log	4244356	47159	10	N/A	N/A	0.279 (Avg)	N/A	N/A
NC-TFIDF	Tanimoto	4244356	4719	10	N/A	N/A	0.289 (Avg)	N/A	N/A
NC-TFIDF	Pearson	4244356	471	10	N/A	2	N/A	0.0	0.0
NC-TFIDF	Euc Dist	4244356	471	10	N/A	2	N/A	0.230	0.291
NC-TFIDF	Log	4244356	471	10	N/A	2	N/A	0.063	0.041
NC-TFIDF	Spearman	4244356	471	10	N/A	2	N/A	0.037	0.065
NC-TFIDF	Tanimoto	4244356	471	10	N/A	2	N/A	0.0	0.0
NC-TFIDF	Pearson	4244356	47159	50	N/A	N/A	521	N/A	N/A
NC-TFIDF	Euc Dist	4244356	47159	50	N/A	N/A	0.343	N/A	N/A
NC-TFIDF	Log	4244356	47159	50	N/A	N/A	0.356	N/A	N/A
NC-TFIDF	Spearman	4244356	47159	50	N/A	N/A	399	N/A	N/A
NC-TFIDF	Tanimoto	4244356	47159	50	N/A	N/A	0.362	N/A	N/A
NC-TFIDF	Log	4244356	47159	50	N/A	N/A	0.300 (Avg)	N/A	N/A
NC-TFIDF	Tanimoto	4244356	47159	50	N/A	N/A	0.315 (Avg)	N/A	N/A
NC-TFIDF	Pearson	4244356	471	50	N/A	2	N/A	0.017	0.031
NC-TFIDF	Euc Dist	4244356	471	50	N/A	2	N/A	0.012	0.012
NC-TFIDF	Log	4244356	471	50	N/A	2	N/A	0.015	0.014
NC-TFIDF	Spearman	4244356	471	50	N/A	2	N/A	0.0	0.0
NC-TFIDF	Tanimoto	4244356	471	50	N/A	2	N/A	0.048	0.031
NC-TFIDF	Pearson	4244356	47159	N/A	0.1	N/A	0.278	N/A	N/A
NC-TFIDF	Euc Dist	4244356	47159	N/A	0.1	N/A	0.336	N/A	N/A
NC-TFIDF	Log	4244356	47159	N/A	N/A	0.1	0.341	N/A	N/A
NC-TFIDF	Spearman	4244356	47159	N/A	0.1	N/A	0.268	N/A	N/A
NC-TFIDF	Tanimoto	4244356	47159	N/A	0.1	N/A	0.344	N/A	N/A
NC-TFIDF	Log	4244356	47159	N/A	0.1	N/A	0.272 (Avg)	N/A	N/A
NC-TFIDF	Tanimoto	4244356	47159	N/A	0.1	N/A	0.294 (Avg)	N/A	N/A
NC-TFIDF	Pearson	4244356	471	N/A	0.1	2	N/A	0.0	0.0
NC-TFIDF	Euc Dist	4244356	471	N/A	0.1	2	N/A	0.0	0.0
NC-TFIDF	Log	4244356	471	N/A	0.1	2	N/A	0.0	0.0
NC-TFIDF	Spearman	4244356	471	N/A	0.1	2	N/A	0.023	0.043
NC-TFIDF	Tanimoto	4244356	471	N/A	0.1	2	N/A	0.0	0.0

NC-TFIDF	Pearson	4244356	47159	N/A	0.5	N/A	0.272	N/A	N/A
NC-TFIDF	Euc Dist	4244356	47159	N/A	0.5	N/A	0.323	N/A	N/A
NC-TFIDF	Log	4244356	47159	N/A	N/A	0.5	0.333	N/A	N/A
NC-TFIDF	Spearman	4244356	47159	N/A	0.5	N/A	0.317	N/A	N/A
NC-TFIDF	Tanimoto	4244356	47159	N/A	0.5	N/A	0.327	N/A	N/A
NC-TFIDF	Log	4244356	47159	N/A	0.5	N/A	0.282 (Avg)	N/A	N/A
NC-TFIDF	Tanimoto	4244356	47159	N/A	0.5	N/A	0.266 (Avg)	N/A	N/A
NC-TFIDF	Pearson	4244356	471	N/A	0.5	2	N/A	0.0	0.0
NC-TFIDF	Euc Dist	4244356	471	N/A	0.5	2	N/A	0.0	0.0
NC-TFIDF	Log	4244356	471	N/A	0.5	2	N/A	0.0	0.0
NC-TFIDF	Spearman	4244356	471	N/A	0.5	2	N/A	0.023	0.043
NC-TFIDF	Tanimoto	4244356	471	N/A	0.5	2	N/A	0.0	0.0
NP	Pearson	6868938	76321	5	N/A	N/A	0.999	N/A	N/A
NP	Euc Dist	6868938	76321	5	N/A	N/A	0.418	N/A	N/A
NP	Log	6868938	76321	5	N/A	N/A	0.405	N/A	N/A
NP	Spearman	6868938	76321	5	N/A	N/A	0.167	N/A	N/A
NP	Tanimoto	6868938	76321	5	N/A	N/A	0.269	N/A	N/A
NP	Log	6868938	76321	5	N/A	N/A	0.203 (Avg)	N/A	N/A
NP	Tanimoto	6868938	76321	5	N/A	N/A	0.254 (Avg)	N/A	N/A
NP	Pearson	6868938	763	5	N/A	1	N/A	1.0	0.031
NP	Euc Dist	6868938	763	5	N/A	1	N/A	0.231	0.071
NP	Log	6868938	763	5	N/A	1	N/A	0.2	0.029
NP	Spearman	6868938	763	5	N/A	1	N/A	0.0	0.0
NP	Tanimoto	6868938	763	5	N/A	1	N/A	0.4	0.114
NP	Pearson	6868938	763	5	N/A	2	N/A	1.0	0.1
NP	Euc Dist	6868938	763	5	N/A	2	N/A	0.3	0.167
NP	Log	6868938	763	5	N/A	2	N/A	0.333	0.0714
NP	Spearman	6868938	763	5	N/A	2	N/A	0.0	0.0
NP	Tanimoto	6868938	763	5	N/A	2	N/A	0.667	0.153
NP	Pearson	6868938	763	5	N/A	3	N/A	1.0	0.25
NP	Euc Dist	6868938	763	5	N/A	3	N/A	0.0	0.0
NP	Log	6868938	763	5	N/A	3	N/A	NaN	0.0
NP	Spearman	6868938	763	5	N/A	3	N/A	0.0	0.0
NP	Tanimoto	6868938	763	5	N/A	3	N/A	NaN	0.0
NP	Pearson	6868938	76321	10	N/A	N/A	0.999	N/A	N/A
NP	Euc Dist	6868938	76321	10	N/A	N/A	0.416	N/A	N/A
NP	Log	6868938	76321	10	N/A	N/A	0.374	N/A	N/A
NP	Spearman	6868938	76321	10	N/A	N/A	0.118	N/A	N/A
NP	Tanimoto	6868938	76321	10	N/A	N/A	0.295	N/A	N/A
NP	Log	6868938	76321	10	N/A	N/A	0.199 (Avg)	N/A	N/A
NP	Tanimoto	6868938	76321	10	N/A	N/A	0.281 (Avg)	N/A	N/A
NP	Pearson	6868938	763	10	N/A	1	N/A	0.0	0.0
NP	Euc Dist	6868938	763	10	N/A	1	N/A	0.097	0.071
NP	Log	6868938	763	10	N/A	1	N/A	0.214	0.088
NP	Spearman	6868938	763	10	N/A	1	N/A	0.0	0.0
NP	Tanimoto	6868938	763	10	N/A	1	N/A	0.5	0.154
NP	Pearson	6868938	763	10	N/A	2	N/A	0.0	0.0
NP	Euc Dist	6868938	763	10	N/A	2	N/A	0.125	0.167
NP	Log	6868938	763	10	N/A	2	N/A	0.083	0.071
NP	Spearman	6868938	763	10	N/A	2	N/A	0.0	0.0

NP	Tanimoto	6868938	763	10	N/A	2	N/A	0.5	0.154
NP	Pearson	6868938	763	10	N/A	3	N/A	0.333	0.25
NP	Euc Dist	6868938	763	10	N/A	3	N/A	0.333	0.111
NP	Log	6868938	763	10	N/A	3	N/A	0.0	0.0
NP	Spearman	6868938	763	10	N/A	3	N/A	0.0	0.0
NP	Tanimoto	6868938	763	10	N/A	3	N/A	0.0	0.0
NP	Spearman	6868938	763	10	N/A	4	N/A	0.083	0.167
NP	Pearson	6868938	76321	50	N/A	N/A	0.233	N/A	N/A
NP	Euc Dist	6868938	76321	50	N/A	N/A	0.405	N/A	N/A
NP	Log	6868938	76321	50	N/A	N/A	0.387	N/A	N/A
NP	Spearman	6868938	76321	50	N/A	N/A	0.999	N/A	N/A
NP	Tanimoto	6868938	76321	50	N/A	N/A	0.373	N/A	N/A
NP	Log	6868938	76321	50	N/A	N/A	0.345 (Avg)	N/A	N/A
NP	Tanimoto	6868938	76321	50	N/A	N/A	0.261 (Avg)	N/A	N/A
NP	Pearson	6868938	763	50	N/A	1	N/A	1.0	0.031
NP	Euc Dist	6868938	763	50	N/A	1	N/A	0.024	0.027
NP	Log	6868938	763	50	N/A	1	N/A	0.103	0.088
NP	Spearman	6868938	763	50	N/A	1	N/A	0.083	0.024
NP	Tanimoto	6868938	763	50	N/A	1	N/A	0.097	0.086
NP	Pearson	6868938	763	50	N/A	2	N/A	0.5	0.1
NP	Euc Dist	6868938	763	50	N/A	2	N/A	0.056	0.071
NP	Log	6868938	763	50	N/A	2	N/A	0.045	0.071
NP	Spearman	6868938	763	50	N/A	2	N/A	0.056	0.038
NP	Tanimoto	6868938	763	50	N/A	2	N/A	0.125	0.154
NP	Pearson	6868938	763	50	N/A	3	N/A	0.333	0.25
NP	Euc Dist	6868938	763	50	N/A	3	N/A	0.0	0.0
NP	Log	6868938	763	50	N/A	3	N/A	0.0	0.0
NP	Spearman	6868938	763	50	N/A	3	N/A	0.056	0.0625
NP	Tanimoto	6868938	763	50	N/A	3	N/A	0.333	0.5
NP	Spearman	6868938	763	50	N/A	4	N/A	0.083	0.167
NP	Pearson	6868938	76321	100	N/A	N/A	0.390	N/A	N/A
NP	Euc Dist	6868938	76321	100	N/A	N/A	0.379	N/A	N/A
NP	Log	6868938	76321	100	N/A	N/A	0.396	N/A	N/A
NP	Spearman	6868938	76321	100	N/A	N/A	0.553	N/A	N/A
NP	Tanimoto	6868938	76321	100	N/A	N/A	0.390	N/A	N/A
NP	Log	6868938	76321	100	N/A	N/A	0.344 (Avg)	N/A	N/A
NP	Tanimoto	6868938	76321	100	N/A	N/A	0.341 (Avg)	N/A	N/A
NP	Pearson	6868938	763	50	N/A	1	N/A	0.0	0.0
NP	Euc Dist	6868938	763	50	N/A	1	N/A	0.024	0.024
NP	Log	6868938	763	50	N/A	1	N/A	0.032	0.029
NP	Spearman	6868938	763	50	N/A	1	N/A	0.0	0.0
NP	Tanimoto	6868938	763	50	N/A	1	N/A	0.032	0.029
NP	Pearson	6868938	763	50	N/A	2	N/A	0.5	0.1
NP	Euc Dist	6868938	763	50	N/A	2	N/A	0.0	0.0
NP	Log	6868938	763	50	N/A	2	N/A	0.0	0.0
NP	Spearman	6868938	763	50	N/A	2	N/A	0.0	0.0
NP	Tanimoto	6868938	763	50	N/A	2	N/A	0.042	0.077
NP	Pearson	6868938	763	50	N/A	3	N/A	0.333	0.25
NP	Euc Dist	6868938	763	50	N/A	3	N/A	0.0	0.0
NP	Log	6868938	763	50	N/A	3	N/A	0.0	0.0
NP	Spearman	6868938	763	50	N/A	3	N/A	0.0	0.0
NP	Tanimoto	6868938	763	50	N/A	3	N/A	0.333	0.5

NP	Spearman	6868938	763	50	N/A	4	N/A	0.083	0.0167
NP	Pearson	6868938	76321	N/A	0.1	N/A	0.370	N/A	N/A
NP	Euc Dist	6868938	76321	N/A	0.1	N/A	0.374	N/A	N/A
NP	Log	6868938	76321	N/A	0.1	N/A	0.382	N/A	N/A
NP	Spearman	6868938	76321	N/A	0.1	N/A	N/A	N/A	N/A
NP	Tanimoto	6868938	76321	N/A	0.1	N/A	0.382	N/A	N/A
NP	Log	6868938	76321	N/A	0.1	N/A	0.332 (Avg)	N/A	N/A
NP	Tanimoto	6868938	76321	N/A	0.1	N/A	0.309 (Avg)	N/A	N/A
NP	Pearson	6868938	763	N/A	0.1	1	N/A	0.0	0.0
NP	Euc Dist	6868938	763	N/A	0.1	1	N/A	0.0	0.0
NP	Log	6868938	763	N/A	0.1	1	N/A	0.0	0.0
NP	Spearman	6868938	763	N/A	0.1	1	N/A	0.0	0.0
NP	Tanimoto	6868938	763	N/A	0.1	1	N/A	0.032	0.029
NP	Pearson	6868938	763	N/A	0.1	2	N/A	0.333	0.25
NP	Euc Dist	6868938	763	N/A	0.1	2	N/A	0.0	0.0
NP	Log	6868938	763	N/A	0.1	2	N/A	0.0	0.0
NP	Spearman	6868938	763	N/A	0.1	2	N/A	0.0	0.0
NP	Tanimoto	6868938	763	N/A	0.1	2	N/A	0.0	0.0
NP	Pearson	6868938	763	N/A	0.1	3	N/A	0.0	0.0
NP	Euc Dist	6868938	763	N/A	0.1	3	N/A	0.0	0.0
NP	Log	6868938	763	N/A	0.1	3	N/A	0.0	0.0
NP	Spearman	6868938	763	N/A	0.1	3	N/A	0.0	0.0
NP	Tanimoto	6868938	763	N/A	0.1	3	N/A	0.0	0.0
NP	Tanimoto	6868938	763	N/A	0.1	4	N/A	0.083	0.5
NP	Tanimoto	6868938	763	N/A	0.1	5	N/A	0.143	1.0
NP	Pearson	6868938	76321	N/A	0.25	N/A	0.370	N/A	N/A
NP	Euc Dist	6868938	76321	N/A	0.25	N/A	0.374	N/A	N/A
NP	Log	6868938	76321	N/A	0.25	N/A	0.382	N/A	N/A
NP	Spearman	6868938	76321	N/A	0.25	N/A	N/A	N/A	N/A
NP	Tanimoto	6868938	76321	N/A	0.25	N/A	0.382	N/A	N/A
NP	Log	6868938	76321	N/A	0.25	N/A	0.332 (Avg)	N/A	N/A
NP	Tanimoto	6868938	76321	N/A	0.25	N/A	0.309 (Avg)	N/A	N/A
NP	Pearson	6868938	763	N/A	0.25	1	N/A	0.0	0.0
NP	Euc Dist	6868938	763	N/A	0.25	1	N/A	0.024	0.024
NP	Log	6868938	763	N/A	0.25	1	N/A	0.0	0.0
NP	Spearman	6868938	763	N/A	0.25	1	N/A	0.0	0.0
NP	Tanimoto	6868938	763	N/A	0.25	1	N/A	0.115	0.086
NP	Pearson	6868938	763	N/A	0.25	2	N/A	0.0	0.0
NP	Euc Dist	6868938	763	N/A	0.25	2	N/A	0.0	0.0
NP	Log	6868938	763	N/A	0.25	2	N/A	0.0	0.0
NP	Spearman	6868938	763	N/A	0.25	2	N/A	0.0	0.0
NP	Tanimoto	6868938	763	N/A	0.25	2	N/A	0.25	0.23
NP	Pearson	6868938	763	N/A	0.25	3	N/A	0.0	0.0
NP	Euc Dist	6868938	763	N/A	0.25	3	N/A	0.0	0.0
NP	Log	6868938	763	N/A	0.25	3	N/A	0.0	0.0
NP	Spearman	6868938	763	N/A	0.25	3	N/A	0.0	0.0
NP	Tanimoto	6868938	763	N/A	0.25	3	N/A	0.267	0.185
NP	Tanimoto	6868938	763	N/A	0.25	4	N/A	0.625	0.417
NP	Tanimoto	6868938	763	N/A	0.25	5	N/A	1.0	0.222
NP	Pearson	6868938	76321	N/A	0.5	N/A	N/A	N/A	N/A
NP	Euc Dist	6868938	76321	N/A	0.5	N/A	0.374	N/A	N/A
NP	Log	6868938	76321	N/A	0.5	N/A	0.344	N/A	N/A

NP	Spearman	6868938	76321	N/A	0.5	N/A	N/A	N/A	N/A
NP	Tanimoto	6868938	76321	N/A	0.5	N/A	0.381	N/A	N/A
NP	Log	6868938	76321	N/A	0.5	N/A	0.334 (Avg)	N/A	N/A
NP	Tanimoto	6868938	76321	N/A	0.5	N/A	0.327 (Avg)	N/A	N/A
NP	Pearson	6868938	763	N/A	0.5	1	N/A	0.0	0.0
NP	Euc Dist	6868938	763	N/A	0.5	1	N/A	0.024	0.024
NP	Log	6868938	763	N/A	0.5	1	N/A	0.0	0.0
NP	Spearman	6868938	763	N/A	0.5	1	N/A	0.0	0.0
NP	Tanimoto	6868938	763	N/A	0.5	1	N/A	0.063	0.029
NP	Pearson	6868938	763	N/A	0.5	2	N/A	0.0	0.0
NP	Euc Dist	6868938	763	N/A	0.5	2	N/A	0.0	0.0
NP	Log	6868938	763	N/A	0.5	2	N/A	0.0	0.0
NP	Spearman	6868938	763	N/A	0.5	2	N/A	0.0	0.0
NP	Tanimoto	6868938	763	N/A	0.5	2	N/A	NaN	0.23
NP	Pearson	6868938	763	N/A	0.5	3	N/A	0.0	0.0
NP	Euc Dist	6868938	763	N/A	0.5	3	N/A	0.0	0.0
NP	Log	6868938	763	N/A	0.5	3	N/A	0.0	0.0
NP	Spearman	6868938	763	N/A	0.5	3	N/A	0.0	0.0
NP	Tanimoto	6868938	763	N/A	0.5	3	N/A	NaN	0.185
NP	Pearson	6868938	76321	N/A	0.5	N/A	N/A	N/A	N/A
NP	Euc Dist	6868938	76321	N/A	0.5	N/A	0.374	N/A	N/A
NP	Log	6868938	76321	N/A	0.5	N/A	0.344	N/A	N/A
NP	Spearman	6868938	76321	N/A	0.5	N/A	N/A	N/A	N/A
NP	Tanimoto	6868938	76321	N/A	0.5	N/A	0.381	N/A	N/A
NP	Log	6868938	76321	N/A	0.5	N/A	0.334 (Avg)	N/A	N/A
NP	Tanimoto	6868938	76321	N/A	0.5	N/A	0.327 (Avg)	N/A	N/A
NP	Pearson	6868938	763	N/A	0.5	1	N/A	0.0	0.0
NP	Euc Dist	6868938	763	N/A	0.5	1	N/A	0.024	0.024
NP	Log	6868938	763	N/A	0.5	1	N/A	0.0	0.0
NP	Spearman	6868938	763	N/A	0.5	1	N/A	0.0	0.0
NP	Tanimoto	6868938	763	N/A	0.5	1	N/A	0.063	0.029
NP	Pearson	6868938	763	N/A	0.5	2	N/A	0.0	0.0
NP	Euc Dist	6868938	763	N/A	0.5	2	N/A	0.0	0.0
NP	Log	6868938	763	N/A	0.5	2	N/A	0.0	0.0
NP	Spearman	6868938	763	N/A	0.5	2	N/A	0.0	0.0
NP	Tanimoto	6868938	763	N/A	0.5	2	N/A	NaN	0.23
NP	Pearson	6868938	763	N/A	0.5	3	N/A	0.0	0.0
NP	Euc Dist	6868938	763	N/A	0.5	3	N/A	0.0	0.0
NP	Log	6868938	763	N/A	0.5	3	N/A	0.0	0.0
NP	Spearman	6868938	763	N/A	0.5	3	N/A	0.0	0.0
NP	Tanimoto	6868938	763	N/A	0.5	3	N/A	NaN	0.185
NP-TFIDF	Pearson	1348092	14978	10	N/A	N/A	0.118	N/A	N/A
NP-TFIDF	Euc Dist	1348092	14978	10	N/A	N/A	0.346	N/A	N/A
NP-TFIDF	Log	1348092	14978	10	N/A	N/A	0.339	N/A	N/A
NP-TFIDF	Spearman	1348092	14978	10	N/A	N/A	0.098	N/A	N/A
NP-TFIDF	Tanimoto	1348092	14978	10	N/A	N/A	0.345	N/A	N/A
NP-TFIDF	Log	1348092	14978	10	N/A	N/A	0.267 (Avg)	N/A	N/A
NP-TFIDF	Tanimoto	1348092	14978	10	N/A	N/A	0.292 (Avg)	N/A	N/A
NP-TFIDF	Pearson	1348092	150	10	N/A	2	N/A	0.1	0.111
NP-TFIDF	Euc Dist	1348092	150	10	N/A	2	N/A	0.0	0.0

NP-TFIDF	Log	1348092	150	10	N/A	2	N/A	0.0	0.0
NP-TFIDF	Spearman	1348092	150	10	N/A	2	N/A	0.0	0.0
NP-TFIDF	Tanimoto	1348092	150	10	N/A	2	N/A	0.0	0.0
NP-TFIDF	Pearson	1348092	14978	50	N/A	N/A	N/A	N/A	N/A
NP-TFIDF	Euc Dist	1348092	14978	50	N/A	N/A	0.329	N/A	N/A
NP-TFIDF	Log	1348092	14978	50	N/A	N/A	0.326	N/A	N/A
NP-TFIDF	Spearman	1348092	14978	50	N/A	N/A	N/A	N/A	N/A
NP-TFIDF	Tanimoto	1348092	14978	50	N/A	N/A	0.323	N/A	N/A
NP-TFIDF	Log	1348092	14978	50	N/A	N/A	0.281 (Avg)	N/A	N/A
NP-TFIDF	Tanimoto	1348092	14978	50	N/A	N/A	0.264 (Avg)	N/A	N/A
NP-TFIDF	Pearson	1348092	150	50	N/A	2	N/A	0.083	0.111
NP-TFIDF	Euc Dist	1348092	150	50	N/A	2	N/A	0.0	0.0
NP-TFIDF	Log	1348092	150	50	N/A	2	N/A	0.071	0.125
NP-TFIDF	Spearman	1348092	150	50	N/A	2	N/A	0.0	0.0
NP-TFIDF	Tanimoto	1348092	150	50	N/A	2	N/A	0.0	0.0
NP-TFIDF	Pearson	1348092	14978	N/A	0.1	N/A	NaN	N/A	N/A
NP-TFIDF	Euc Dist	1348092	14978	N/A	0.1	N/A	0.325	N/A	N/A
NP-TFIDF	Log	1348092	14978	N/A	N/A	0.1	0.311	N/A	N/A
NP-TFIDF	Spearman	1348092	14978	N/A	0.1	N/A	NaN	N/A	N/A
NP-TFIDF	Tanimoto	1348092	14978	N/A	0.1	N/A	0.307	N/A	N/A
NP-TFIDF	Log	1348092	14978	N/A	0.1	N/A	0.255 (Avg)	N/A	N/A
NP-TFIDF	Tanimoto	1348092	14978	N/A	0.1	N/A	0.315 (Avg)	N/A	N/A
NP-TFIDF	Pearson	1348092	150	N/A	0.1	2	N/A	0.0	0.0
NP-TFIDF	Euc Dist	1348092	150	N/A	0.1	2	N/A	0.0	0.0
NP-TFIDF	Log	1348092	150	N/A	0.1	2	N/A	0.0	0.0
NP-TFIDF	Spearman	1348092	150	N/A	0.1	2	N/A	0.0	0.0
NP-TFIDF	Tanimoto	1348092	150	N/A	0.1	2	N/A	NaN	0.0
NP-TFIDF	Pearson	1348092	14978	N/A	0.5	N/A	NaN	N/A	N/A
NP-TFIDF	Euc Dist	1348092	14978	N/A	0.5	N/A	0.325	N/A	N/A
NP-TFIDF	Log	1348092	14978	N/A	N/A	0.5	0.340	N/A	N/A
NP-TFIDF	Spearman	1348092	14978	N/A	0.5	N/A	0.268	N/A	N/A
NP-TFIDF	Tanimoto	1348092	14978	N/A	0.5	N/A	0.337	N/A	N/A
NP-TFIDF	Log	1348092	14978	N/A	0.5	N/A	0.262 (Avg)	N/A	N/A
NP-TFIDF	Tanimoto	1348092	14978	N/A	0.5	N/A	0.156 (Avg)	N/A	N/A
NP-TFIDF	Pearson	1348092	150	N/A	0.5	2	N/A	0.0	0.0
NP-TFIDF	Euc Dist	1348092	150	N/A	0.5	2	N/A	0.0	0.0
NP-TFIDF	Log	1348092	150	N/A	0.5	2	N/A	0.0	0.0
NP-TFIDF	Spearman	1348092	150	N/A	0.5	2	N/A	0.0	0.0
NP-TFIDF	Tanimoto	1348092	150	N/A	0.5	2	N/A	NaN	0.0
PS	Pearson	1290680	14340	10	N/A	N/A	0.166	N/A	N/A
PS	Euc Dist	1290680	14340	10	N/A	N/A	0.167	N/A	N/A
PS	Log	1290680	14340	N/A	10	N/A	0.280	N/A	N/A
PS	Spearman	1290680	14340	10	N/A	0.226	N/A	N/A	N/A
PS	Tanimoto	1290680	14340	10	N/A	N/A	0.266	N/A	N/A
PS	Log	1290680	14340	10	N/A	N/A	0.149 (Avg)	N/A	N/A
PS	Tanimoto	1290680	14340	10	N/A	N/A	0.174 (Avg)	N/A	N/A
PS	Pearson	1290680	14340	50	N/A	N/A	0.369	N/A	N/A
PS	Euc Dist	1290680	14340	50	N/A	N/A	0.342	N/A	N/A

PS	Log	1290680	14340	N/A	50	N/A	0.381	N/A	N/A
PS	Spearman	1290680	14340	50	N/A	N/A	0.660	N/A	N/A
PS	Tanimoto	1290680	14340	50	N/A	N/A	0.307	N/A	N/A
PS	Log	1290680	14340	50	N/A	N/A	0.209 (Avg)	N/A	N/A
PS	Tanimoto	1290680	14340	50	N/A	N/A	0.173 (Avg)	N/A	N/A
PS	Pearson	1290680	14340	N/A	0.1	N/A	0.397	N/A	N/A
PS	Euc Dist	1290680	14340	N/A	0.1	N/A	0.365	N/A	N/A
PS	Log	1290680	14340	N/A	N/A	0.1	0.245	N/A	N/A
PS	Spearman	1290680	14340	N/A	0.1	N/A	0.327	N/A	N/A
PS	Tanimoto	1290680	14340	N/A	0.1	N/A	0.300	N/A	N/A
PS	Log	1290680	14340	N/A	0.1	N/A	0.243 (Avg)	N/A	N/A
PS	Tanimoto	1290680	14340	N/A	0.1	N/A	0.333 (Avg)	N/A	N/A
PS	Pearson	1290680	14340	N/A	0.5	N/A	0.241	N/A	N/A
PS	Euc Dist	1290680	14340	N/A	0.5	N/A	0.391	N/A	N/A
PS	Log	1290680	14340	N/A	N/A	0.5	0.388	N/A	N/A
PS	Spearman	1290680	14340	N/A	0.5	N/A	0.478	N/A	N/A
PS	Tanimoto	1290680	14340	N/A	0.5	N/A	NaN	N/A	N/A
PS	Log	1290680	14340	N/A	0.5	N/A	0.242 (Avg)	N/A	N/A
PS	Tanimoto	1290680	14340	N/A	0.5	N/A	NaN	N/A	N/A
NC-TFIDF (Log)	Pearson	4244356	47159	10	N/A	N/A	0.392	N/A	N/A
NC-TFIDF (Log)	Euc Dist	4244356	47159	10	N/A	N/A	0.313	N/A	N/A
NC-TFIDF (Log)	Log	4244356	47159	N/A	10	N/A	0.325	N/A	N/A
NC-TFIDF (Log)	Spearman	4244356	47159	10	N/A	0.470	N/A	N/A	N/A
NC-TFIDF (Log)	Tanimoto	4244356	47159	10	N/A	N/A	0.323	N/A	N/A
NC-TFIDF (Log)	Log	4244356	47159	10	N/A	N/A	0.253 (Avg)	N/A	N/A
NC-TFIDF (Log)	Tanimoto	4244356	47159	10	N/A	N/A	0.262 (Avg)	N/A	N/A
NC-TFIDF (Log)	Pearson	4244356	471	10	N/A	2	N/A	0.0	0.0
NC-TFIDF (Log)	Euc Dist	4244356	471	10	N/A	2	N/A	0.236	0.291
NC-TFIDF (Log)	Log	4244356	471	10	N/A	2	N/A	0.067	0.042
NC-TFIDF (Log)	Spearman	4244356	471	10	N/A	2	N/A	0.049	0.076
NC-TFIDF (Log)	Tanimoto	4244356	471	10	N/A	2	N/A	0.0	0.0
NC-TFIDF (Log)	Pearson	4244356	471	10	N/A	5	N/A	0.0	0.0
NC-TFIDF (Log)	Euc Dist	4244356	471	10	N/A	5	N/A	0.278	0.208
NC-TFIDF (Log)	Log	4244356	471	10	N/A	5	N/A	0.0	0.0
NC-TFIDF (Log)	Spearman	4244356	471	10	N/A	5	N/A	0.076	0.119

NC-TFIDF (Log)	Tanimoto	4244356	471	10	N/A	5	N/A	0.0	0.0
NC-TFIDF (Log)	Pearson	4244356	47159	50	N/A	N/A	0.484	N/A	N/A
NC-TFIDF (Log)	Euc Dist	4244356	47159	50	N/A	N/A	0.312	N/A	N/A
NC-TFIDF (Log)	Log	4244356	47159	N/A	50	N/A	0.323	N/A	N/A
NC-TFIDF (Log)	Spearman	4244356	47159	50	N/A	0.378	N/A	N/A	N/A
NC-TFIDF (Log)	Tanimoto	4244356	47159	50	N/A	N/A	0.329	N/A	N/A
NC-TFIDF (Log)	Log	4244356	47159	50	N/A	N/A	0.275 (Avg)	N/A	N/A
NC-TFIDF (Log)	Tanimoto	4244356	47159	50	N/A	N/A	0.285 (Avg)	N/A	N/A
NC-TFIDF (Log)	Pearson	4244356	471	50	N/A	2	N/A	0.017	0.031
NC-TFIDF (Log)	Euc Dist	4244356	471	50	N/A	2	N/A	0.012	0.012
NC-TFIDF (Log)	Log	4244356	471	50	N/A	2	N/A	0.016	0.014
NC-TFIDF (Log)	Spearman	4244356	471	50	N/A	2	N/A	0.0	0.0
NC-TFIDF (Log)	Tanimoto	4244356	471	50	N/A	2	N/A	0.048	0.031
NC-TFIDF (Log)	Pearson	4244356	471	50	N/A	5	N/A	0.0	0.0
NC-TFIDF (Log)	Euc Dist	4244356	471	50	N/A	5	N/A	0.1	0.208
NC-TFIDF (Log)	Log	4244356	471	50	N/A	5	N/A	0.0	0.0
NC-TFIDF (Log)	Spearman	4244356	471	50	N/A	5	N/A	0.029	0.071
NC-TFIDF (Log)	Tanimoto	4244356	471	50	N/A	5	N/A	0.0	0.0
NC-TFIDF (Log)	Pearson	4244356	47159	N/A	0.1	N/A	0.264	N/A	N/A
NC-TFIDF (Log)	Euc Dist	4244356	47159	N/A	0.1	N/A	0.307	N/A	N/A
NC-TFIDF (Log)	Log	4244356	47159	N/A	N/A	0.1	0.312	N/A	N/A
NC-TFIDF (Log)	Spearman	4244356	47159	N/A	0.1	N/A	0.251	N/A	N/A
NC-TFIDF (Log)	Tanimoto	4244356	47159	N/A	0.1	N/A	0.312	N/A	N/A
NC-TFIDF (Log)	Log	4244356	47159	N/A	0.1	N/A	0.250 (Avg)	N/A	N/A
NC-TFIDF (Log)	Tanimoto	4244356	47159	N/A	0.1	N/A	0.264 (Avg)	N/A	N/A
NC-TFIDF (Log)	Pearson	4244356	471	50	N/A	2	N/A	0.0	0.0
NC-TFIDF (Log)	Euc Dist	4244356	471	50	N/A	2	N/A	0.0	0.0
NC-TFIDF (Log)	Log	4244356	471	50	N/A	2	N/A	0.071	0.125

NC-TFIDF (Log)	Spearman	4244356	471	50	N/A	2	N/A	0.023	0.043
NC-TFIDF (Log)	Tanimoto	4244356	471	50	N/A	2	N/A	0.0	0.0
ACL	Pearson	4244356	47159	10	N/A	N/A	0.376	N/A	N/A
ACL	Euc Dist	4244356	47159	10	N/A	N/A	0.399	N/A	N/A
ACL	Log	4244356	47159	N/A	10	N/A	0.361	N/A	N/A
ACL	Spearman	4244356	47159	10	N/A	0.401	N/A	N/A	N/A
ACL	Tanimoto	4244356	47159	10	N/A	N/A	0.347	N/A	N/A
ACL	Log	4244356	47159	10	N/A	N/A	0.301 (Avg)	N/A	N/A
ACL	Tanimoto	4244356	47159	10	N/A	N/A	0.289 (Avg)	N/A	N/A
ACL	Pearson	4244356	471	10	N/A	2	N/A	0.014	0.007
ACL	Euc Dist	4244356	471	10	N/A	2	N/A	0.236	0.291
ACL	Log	4244356	471	10	N/A	2	N/A	0.045	0.056
ACL	Spearman	4244356	471	10	N/A	2	N/A	0.017	0.025
ACL	Tanimoto	4244356	471	10	N/A	2	N/A	0.087	0.036
ACL	Pearson	4244356	47159	50	N/A	N/A	0.380	N/A	N/A
ACL	Euc Dist	4244356	47159	50	N/A	N/A	0.364	N/A	N/A
ACL	Log	4244356	47159	N/A	50	N/A	0.333	N/A	N/A
ACL	Spearman	4244356	47159	50	N/A	0.376	N/A	N/A	N/A
ACL	Tanimoto	4244356	47159	50	N/A	N/A	0.348	N/A	N/A
ACL	Log	4244356	47159	50	N/A	N/A	0.281 (Avg)	N/A	N/A
ACL	Tanimoto	4244356	47159	50	N/A	N/A	0.286 (Avg)	N/A	N/A
ACL	Pearson	4244356	471	50	N/A	2	N/A	0.0	0.0
ACL	Euc Dist	4244356	471	50	N/A	2	N/A	0.008	0.017
ACL	Log	4244356	471	50	N/A	2	N/A	0.044	0.083
ACL	Spearman	4244356	471	50	N/A	2	N/A	0.0	0.0
ACL	Tanimoto	4244356	471	50	N/A	2	N/A	0.021	0.027

Table 2: Partial table of training data including all data for the various feature mapping techniques. Data for segmentation and Gaussian normalization is not included in this table because entering such data into the table manually would take approximately 2 hours. Generalized trends from the data are shown in the sections in the body of the paper. Training Percentage is 0.9 and 0.01 is tested. N is the number of nearest neighbors. T is the threshold used in the Threshold algorithms. H is the H-th highest rating items taken out from the set when calculating precision and recall. RMS Err is the root-mean-square error. Precision and recall are as defined above.