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]. [UPDATE]

**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 subreddits, which are user created and moderated. Reddit currently has over 850,000 subreddits, with hundreds more being created each day.[[1]](#footnote-1) 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**

Recommender systems are broadly grouped into two strategies. Content based systems focus on the features of the item itself; similarity between items is determined by measuring similarity in these features. In contrast, collaborative filtering uses user behavior to find similarities between like-minded users.[[2]](#footnote-2) A major appeal of collaborative filtering approaches is that is does not require the development of features to describe items or users; the only required information is the past behavior of users.[[3]](#footnote-3) However, collaborative filtering has a number of challenges as well, such as sparse/missing data, noisy data and a lack of negative ratings (in the case of implicit feedback), and the “cold start” problem (see data collection section). In this paper, we focus on collaborative filtering, although we attempt limited implementation of content based text similarity on a small subset of subreddits.

**Data Collection** [STILL WORKING]

Collaborative filtering relies on user feedback, either explicit (e.g. star ratings on Netflix) or implicit (e.g. a user’s purchase history). Reddit does not have explicit user feedback for subreddits, but numerous observable user behaviors can serve as implicit feedback.

* 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 how much that user is interested in that subreddit.
* 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 make posts are nonetheless active commenters, so comments may provide more complete information on user preferences/behavior.
* Length of comments: While number of posts and number of comments provide the frequency of user activity on a subreddit, a raw count does not give any sense of the quality of that activity. Length of comments may 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. Shorter comments do not always mean lesser degree of interest, as brevity 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 is not an indicator of interest
* [I have some stuff written about comment/post scores, but took it out to save space – if you did analysis on those features, I will clean it up and put it back in].

For this study, we used Reddit post and comment data from August 2016 made publically available on Google BigQuery.[[4]](#footnote-4) This data set includes 7,591,689 posts and 69,654,819 comments by 3,698,088 unique users on 100,279 subreddits.

We also considered two item-similarity metrics. Item-similarity is typically easier to estimate than user-similarity because items are more stable and easier to categorize than humans (for example, sports subreddits are easier to characterize than sports fans). In addition, providing item-based similarity metrics help to address the “cold start” problem, allowing users who do not have a post or comment history to find similar subreddits. Both of these features rely on subreddit history, however, so they do not resolve the item cold-start problem (new subreddit without any user activity would not be evaluated until it has built up a post/comment history).

* 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.
* 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 text processing section); similar methods could be applied to the comments of a subreddit.
* Cross-posts??

**Data Preprocessing**

The Reddit data set is both large and inherently noisy, challenges common to many recommender systems which rely on implicit ratings. In order to reduce data set to a more manageable size, we first removed known bots[[5]](#footnote-5) and default subreddits[[6]](#footnote-6). [UPDATE - We further reduced the data set by only selecting those remaining users who posted to at least 10 subreddits or more.[[7]](#footnote-7) Unless otherwise noted, this data set was used in all the following evaluations.]

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 and weighting techniques.

* dividing by the max number of comments on any subreddit (thereby putting all ratings on 0 – 1 scale with 1.0 being any subreddit equal to the max number of comments by that user on any subreddit that month)
* Log normalization: 1 + log (x) [I sort of end up describing this a bit in the TF-IDF section, if you want to either reference that section or move it up here.]
* Gaussian
* “TF-IDF-like” weighting scheme: In order to account for the fact that posting or commenting behavior on an extremely popular subreddit (such as /r/AskReddit) is not as informative as posting or commenting on a more niche subreddit in characterizing a user’s preferences, we experiment with weighting our features with a “IDF-like” term which is given by where N is the number of users who posted or commented to any subreddit and ns 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 users’ posts/comments on subreddits which have a large number of other users posting on it.

Used cross-validation to select best normalization technique or combination of techniques...?

**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.[[8]](#footnote-8) We then used a combination of regular expressions and packages from the Natural Language Toolkit (NLTK) to create a “bag of words” for each subreddit by removing all non-alphanumeric characters and capitalization, filtering for stop words (using a set of stop words provided by NLTK), tokenizing those words that were greater than two characters, and applying a Porter stemmer (as implemented by NLTK).[[9]](#footnote-9)

We use the TfidfVectorizer package from sklearn to transform our bag of words to a matrix of TF-IDF features.[[10]](#footnote-10) 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 subreddit in August 2016 and the corpus of documents is the entire set of subreddits.

We apply log or sublinear normalization to the term frequency (*tf*), to account for the fact that multiple occurrences of a term have less significance as the number of occurrences increases. The log normalized term frequency (*tf­norm*), inverse document frequency (*idf*), and TF-IDF value for term *t* in document *d* are given by

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

We further apply L2 (Euclidean) normalization to the TF-IDF values, dividing each term by , where *x* is thevector of TF-IDF features. Using this matrix of TF-IDF features, we compute pairwise cosine similarity by applying a linear kernel to the already L2-normalized data.

**Recommender Algorithms**

**Evaluation**

EVERYTHING AFTER THIS LINE IS JUST SCRAPS / NOTES

* Total points from posts: 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 assume that a user makes higher quality contributions to subreddits which he or she is more interested in.
* Total points from comments: Similarly for comments...

There are short-comings to each of these implicit ratings, which we will discuss further in the results section...(?)

For content-based filtering, we use two data sources. The first a set of cross-posts... TF-IDF content similarity based on posts/comments.

Challenges with the data/things we did to clean up the data

* Remove bots – manually checked all those users with >3000 comments in one month. Scraped /r/botWatcher for the names of bots mentioned. Some bots inevitably are left, but XXX active bots identified and the account for over X million comments.
* Remove users who only posted or commented to one subreddit – this does not provide any information
* Remove users who posted/commented < 1(3,10, whatever) and > 3000?
* Normalization...

**Preprocessing the Data**

One challenge to using implicit ratings is the need to normalize to account for differences in user behavior (this is also true of explicit ratings...). For example, in the August 2016 Reddit data set, users made anywhere from 1 to over 10,000 comments in one month. The wide range of behaviors is typical across all five metrics/rating proxies. In order to normalize user data to compare across users, we tried

* dividing by the max number of comments on any subreddit (thereby putting all ratings on 0 – 1 scale with 1.0 being any subreddit equal to the max number of comments by that user on any subreddit that month)
* TF-IDF type normalization

Used cross-validation to select best performance

Introduction

* Why reddit
* Why recommender system for Reddit – new users, marketers, discover dark networks
* What we plan to do

Discussion of recommender systems

* two main approaches
  + Collaborative filtering:
    - User: find like-minded users who have rated things similarly
    - Item:
  + Content-based – create a profile for each user or product to characterize its nature, challenge is how do you develop those features?
* Combine techniques to achieve better performance?
* We will start with CF (there is so much raw user data), but will also try content-based (TF-IDF)

Collaborative filtering

* Challenges
  + Cold start – cannot address products new to the system (new subreddit – how do we rate it before any one comments to it?)
* Need to get rating on user behavior
  + Explicit feedback – Netflix star ratings
    - Subscribe to subreddit? binary
    - Moderate subreddit? binary
  + Implicit – user behavior (purchase history, browsing history, amount of time spent on a movie/song, etc)
  + Clean data
    - Identify and remove bots; represents X% of comments, X% of posts
    - Identify and remove default subreddits?
  + Raw data from BigQuery (X comments, X posts, X users, X subreddits – Aug 2016)
    - # comments/#posts per user per sub (or combination of posts and comments):
      * comments are part of a post thread, they are the most common type of behavior on Reddit
      * posts are the top level links, could be considered more significant
      * we assume more comments/posts means a higher rating; this is a bit problematic ...
        + Trump voter trolling a Hillary subreddit, but in this case we consider that a different type of “like”
        + some subreddits have a frequent posting behavior – like /r/counting, which may cause it to be skewed; others have high bar to posting (original fiction subs)
    - length of comments
      * length of posts is not as meaningful because they are typically titles/often meant to be brief
      * length of comments is a way to sepearate those “cheap” comments (/r/counting) for those more “expensive” comments (fiction sub)
      * not always true – internet/reddit culture is sometimes just to be brief, longer isn’t necessarily better, probably some threshold – longer than this means something, but how much longer doesn’t tell us as much
    - sum score comments/posts
      * this is the total number of points received for comments/posts by a user on a subreddit
      * indication more of what other users think of that user, less what the user thinks of that subreddit
      * however, does give a sense indirectly of whether that user made serious/good comments (in the opinion of that subreddit community)
    - Challenges to implicit ratings (Hu et al, 2009)
      * No negative feedback – see what users chose to post/comment as like, hard to infer what they did not like
      * Inherently noisy
      * Numerical value of implicit feedback indicates confidence instead of preference
      * The numerical value of explicit feedback indicates preference, whereas the numerical value of implicit feedback indicates confidence. Systems based on explicit feedback let the user express their level of preference, e.g. a star rating between 1 (“totally dislike”) and 5 (“really like”). On the other hand, numerical values of implicit feedback describe the frequency of actions, e.g., how much time the user watched a certain show, how frequently a user is buying a certain item, etc. A larger value is not indicating a higher preference. For example, the most loved show may be a movie that the user will watch only once, while there is a series that the user quite likes and thus is watching every week. However, the numerical value of the feedback is definitely useful, as it tells us about the confidence that we have in a certain observation. A one time event might be caused by various reasons that have nothing to do with user preferences. However, a recurring event is more likely to reflect the user opinion. (Hu, 2009)
      * Difficult to interpret implicit rating measures

Evaluation of implicit-feedback recommender requires

appropriatemeasures. In the traditional setting where a

user is specifying a numeric score, there are clear metrics

such as mean squared error to measure success in

prediction. However with implicit models we have to

take into account availability of the item, competition

for the item with other items, and repeat feedback. For

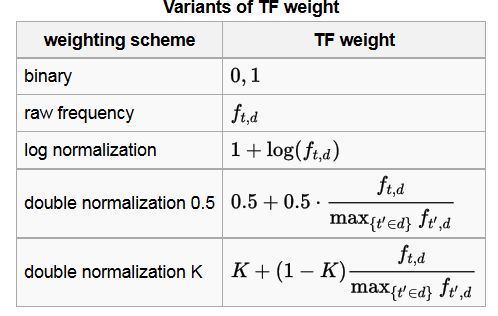
example, if we gather data on television viewing, it is

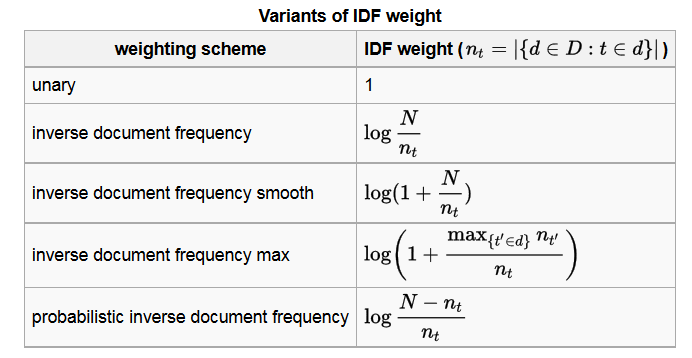
unclear how to evaluate a show that has been watched

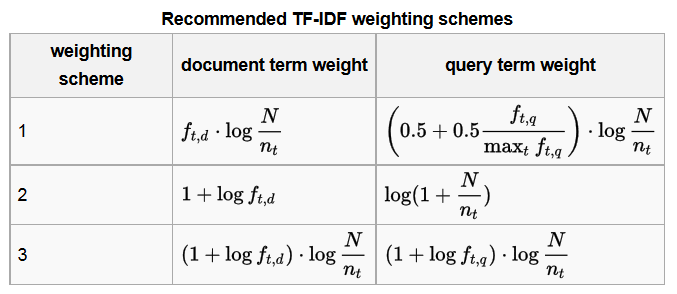
more than once, or how to compare two shows that are

on at the same time, and hence cannot both be watched

by the user.







Reddit is organized by both formally, by the features in how the website's software is designed

to operate, and informally, through a set of rules and etiquette, known as Reddiquette, by

both the platform's owners and users. Reddit is made up of many individual communities,

also known as subreddits, with each community having its own unique web page, subject

matter, users, and moderators. Users, known as redditors, can make their own subreddits or

join existing ones. Within a subreddit, users contribute, or post, stories, links, and media to

the community that they believe add value to that subreddit's prede\_ned subject matter, and

other users can then vote and comment on those posts. Through voting, known as upvoting

or downvoting, users can determine which posts rise to the top of subreddit community

pages, as well as, the main home page of Reddit, which pulls in popular content from all

active subreddits. Comments allow users to provide an editorial narrative regarding posts

and engage in community-oriented discussion, and can also be voted on in the same way

as posts. Users gain a reputation on the site when their posts or comments get upvoted,

quanti\_ed by a metric denoted as karma, which is intended to be a measure of how much a

user has contributed to the Reddit community. You can see how much karma a user has by

looking at their individual pro\_le page.

The main goals are to make clear what your findings are, why you think they came out the

way they did, and why that might be important and to be precise enough to allow someone to

replicate your experiments (or verify your proofs).

Communicate results and process

Finally, “collaborative filtering systems are either memory based or model based. Memory-based systems work directly with user data. Given the selections of a given user, a memory-based system identifies similar users and makes recommendations based on the items selected by these users. Model-based systems compress such user data into a predictive model” .[[11]](#footnote-11) In this paper, we focus on memory-based systems, using Apache Mahout...

Although Reddit users do not give explicit ratings on subreddits, we experimented with using a number of different behavioral metrics as implicit user ratings for a subreddit.

1. http://expandedramblings.com/index.php/reddit-stats/; http://redditmetrics.com/ [↑](#footnote-ref-1)
2. Stanford ML recsys, pg 312. [↑](#footnote-ref-2)
3. Collaborative Filtering with Decoupled Models for Preferences and Ratings, Rong Jin , Luo Si , ChengXiang Zhai and Jamie Callan; Collaborative Filtering for Implicit Feedback Datasets Hu, Koren, Volinsky [↑](#footnote-ref-3)
4. Cite BigQuery; data sets are also available for other months [↑](#footnote-ref-4)
5. 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. 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. [↑](#footnote-ref-5)
6. 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 account for 17.6% of all posts and comments in August 2016. [↑](#footnote-ref-6)
7. Document the size of that data set? Further rationale – important to post to at least 10 subreddits in order to have meaningful similarity metrics?? [↑](#footnote-ref-7)
8. 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 also more manageable in terms of the number of words to be processed. Bag of words are typically high [↑](#footnote-ref-8)
9. http://www.nltk.org/, also cite code [↑](#footnote-ref-9)
10. http://scikit-learn.org/stable/modules/generated/sklearn.feature\_extraction.text.TfidfVectorizer.html. Parameters: set the minimum document frequency (min\_df) to 0, normalize using l2; sklearn\_tfidf = TfidfVectorizer(norm='l2',min\_df=0, use\_idf=True, smooth\_idf=False, sublinear\_tf=True)#, stop\_words = 'english') [↑](#footnote-ref-10)
11. MDP-Recommender System, 1269; Jin, et al; Normalization methods; Hu, et al. [↑](#footnote-ref-11)