Two novel techniques for categorizing Reddit comments

Neil Fitzgerald, Jesika Haria, Kevin Wu

December 5, 2013

Abstract

We consider the problem of classifying Reddit comments into their originating subreddits given only the comment text. To do so, we use two novel methods based on clustering and Latent Dirichlet allocation. We were able to successfully classify 21% of comments randomly chosen from the default subreddits. After analysis, we conclude that we are generally able to classify content-based subreddits but run into difficulty with format-based subreddits.

Contents

1 Introduction

We present two novel systems for categorizing Reddit comments into their originating subreddits. The first method we call *outlier-based clustering*. Outlier-based clustering is an approach to clustering that focuses on points diverging from the main pack. The second method we call *multi-alpha LDA*. Multi-alpha LDA uses Latent Dirichlet allocation with one alpha per subreddit and a common beta to predict topics, which are then used to calculate the probability of various comments belonging to each subreddit.

Considering the difficulty of the problem, the generic nature of many comments, and our limited time scope, we have acheived a fair degree of success, correctly classifying 21% of comments between a representative set of all default subreddits.

2 Background

Before beginning the analysis of our methods, we stop to more carefully posit the framework of our problem, previous statistical work done on Reddit comments, and the dataset we have chosen to work with.

2.1 What is Reddit?

Reddit is an online platform for sharing various kinds of content. Users submit either links or text posts, which are then voted on by users. High-scoring posts appear near the top of the page, meaning the most popular content becomes the most visible. Users can comment on each post, and comments are voted on similarly to posts. Again, the highest-scoring comments are the most visible. Users can also comment on comments to create so-called comment threads.

Because users post a wide variety of content, the site is divided into subreddits, each accessible at www.reddit.com/r/[subredditname]. Many of the subreddits are content-based subreddits formed around interest groups, like /r/pokemon for fans of the Pokemon franchise, or /r/atheism for atheists and agnostics. These interest groups vary in scope, from very broad subreddits like /r/gaming (all video gamers), to narrower subreddits like /r/GrandTheftAutoV (players of GTA V). Other subreddits are format-based: /r/IAmA allows users with unique experiences or famous users to post about themselves and answer questions from the comments. /r/funny is for any content that users find humorous.

Some of these subreddits, known as default subreddits, provide the content that appears on the front page of Reddit to a new user. As of this writing, and at the time our dataset was collected, there are 22 default subreddits.[6]

2.2 Project Goal

Our goal is to predict what subreddit a given comment belongs to, based solely on the text of the comment. Such a prediction tool could be used to provide suggestions for where to post a given comment for a new user who is unfamiliar with the site's intricate system of subreddits. Additionally, the prediction tool could provide potentially interesting insights into the unique community on Reddit: e.g., which subreddits' comments are the most ”predictable”.

2.3 Previous Work

Following the problem of discovery of smaller, quality subreddits, a calculation of distances between subreddits based on common user participation was performed by /u/chicken_bridges

[5]. Previous attempts at clustering subreddits based on posts have used Support Vector Machines and Naive Bayes classifiers [3] as well as using common links from posts [7]. Attempts at predicting subreddits have taken advantage of the Prediction API, although with modest results [4]. Prediction of a specific post's properties from its title, such as its NSFW-ness has also been attempted [2].

Finally, towards the continued goal of improved personalized content discovery, A. Taalimanesh et al have provided approaches for recommending posts based on user voting history [1], with visualizations by A. Verster [9]. To the best of the authors' knowledge though, no previous work has been done specfiically on predicting subreddits from comments. Comments are a richer and less gamed dataset than post titles, and are also more natural expression of users than the actual post contents.

2.4 Our Data Set

Our data set is a representative sample of 660,646 Reddit comments collected across all subreddits over the period of a week in November 2013.[8] The comments are distributed across 9,842 subreddits, from /r/AskReddit (73,423 comments) to /r/zoology (1 comment). Only 86 subreddits had 1,000 or more comments.

3 Technical Approach

For the purposes of this paper, we considered comments using the so-called "bag-of-words model". That is, we regarded each comment as a set of words without trying to parse grammar or multi-word phrases. For both methods, we stripped punctuation and capitalization from the comments in the data set and reduced them to a list of lowercase strings. We then removed "stop words": common words such as "and" or "the" that individually signify little or nothing about the meaning of the sentence.[1]

We extracted 10% of the comments from each subreddit for test data, and used as much of the remainder as we could feasibly process for training. From this point, we used two different methods of our own invention to train a learner and use it to predict a test set. These were largely inspired by popular data clustering algorithms, k-means and LDA.

3.1 Outlier-Based Clustering

k-means is an example of Hard EM for a mixture of Gaussians and is one of the most popular clustering algorithms. In our implementation, a string of comma-separated words in a comment formed a tag. Each tag was tokenized, and then vectorized. Under CountVectorizer, each tag is convered into a vector whose features are all the words present in the cumulative dataset, with position in each dimension determined by the count. With the TfidfVectorizer, this plain count is replaced by a weighted count proportional to the inverse of the word frequency in the corpus. The initialization method chosen was k-means++, for faster convergence and to avoid getting stuck in local optima.

[1] Some common words that weren't in our list, particularly "just" and "like", popped up very regularly, but we found these words to be acceptably few and decided against trying to stamp out all common words.

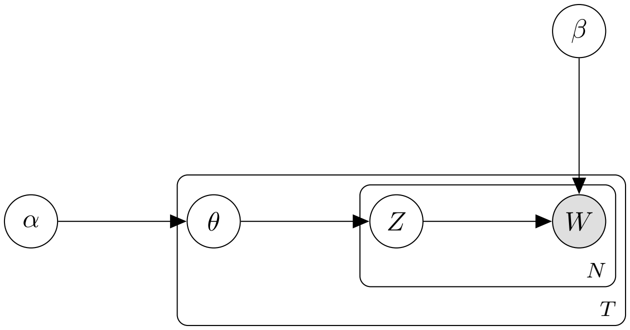

Figure 1: Original LDA model

3.2 Multi-Alpha LDA

In latent Dirichlet allocation (also known as LDA), collections of observed words, or *documents*, are modeled as mixtures of a set of underlying topics, where each word in a document was generated by one of the document's topics. These documents were in turn generated from some Dirichlet distribution of topic distributions. As seen in the plate diagram in Figure 1 , this generative system is essentially defined by latent parameters β, the distribution of words for each topic, and α, the Dirichlet prior on the distrubition of topics for each document. From α, we can generate a topic distribution θ_t for each document t, which in turn generates topics z_i for the N_t words in document t. This is combined with appropriate word per topic distributions β_{z_i} to yield the observed words, $w_{t,i}$.

In our case, we can then treat each comment as a document about some variable mixture of topics. A comment in the /r/gaming subreddit, for instance, could be about computers and graphics cards, or it could be about football in a sports game. However, we note that many topics are much more likely to come up in comments belonging to particular subreddits. For example, The topic of religion is much more likely to appear in a comment in /r/atheism than in /r/pokemon. This observation leads us to believe that each subreddit has its own distribution of topic distributions. In other words, each subreddit, r, has its own α_r, resulting in the plate diagram in Figure 2.

We note that the model is very similar to before, except now each subreddit r has its own set of T_r documents with per document topic distributions $\theta_{r,t}$ and so on, independent of the other subreddits.

For this multi-alpha LDA, we use an approximate EM algorithm similar to the one discussed in lecture. For our R subreddits, we use Gibbs sampling to obtain topic assignments $(z_1^{r,t}, ..., z_{N_{r,t}}^{r,t})$ for each document $d^{r,t} = (w_1^{r,t}, ..., w_{N_{r,t}}^{r,t})$, $r = 1, ..., R$, $t = 1, ..., T_r$. The complete log-likelihood that will be maximized then has form

$$\log P(d^{r,t}, z_1^{r,t}, ..., z_{N_{r,t}}^{r,t} \forall r, t; \alpha_1, ..., \alpha_R, \beta)$$

Due to the independence of word and topic choice between the subreddits and between

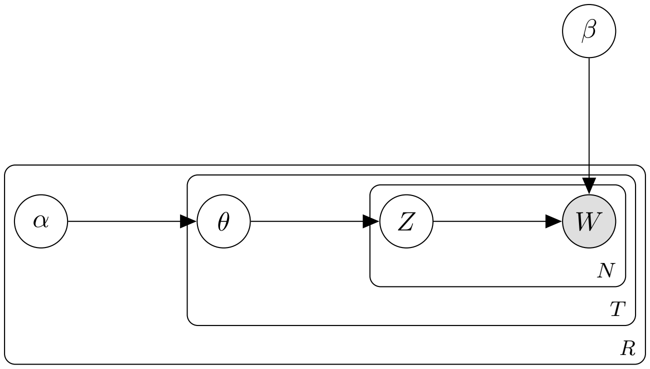

Figure 2: Multi-alpha LDA model

documents, this becomes

$$\sum_{r=1}^{R}\sum_{t=1}^{T_r}\log P(d^{r,t}, z_1^{r,t}, ..., z_{N_{r,t}}^{r,t}; \alpha_r, \beta)$$

By approximating this expected log-likelihood as in HW8, we find maximizing values in the m^{th} M-step by satisfying

$$\beta_{w|z}^{[m+1]} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T_r} n^{r,t}(z,w)}{\sum_{r=1}^{R}\sum_{t=1}^{T_r} n^{r,t}(z)}$$

$$\psi(\alpha_{r,z}^{[m+1]}) + \frac{1}{T}\psi(\sum_{k=1}^{K} a_{r,k}) = \frac{1}{T}\psi(\sum_{t=1}^{T} n_{\theta}^{r,t}(z))$$

where statistics $n^{r,t}(z)$, $n^{r,t}(z,w)$, $n_{\theta}^{r,t}(z)$ are calculated in the E-step after Gibbs sampling and are analogously defined as in HW8.

Using these estimated values for latent variables $\alpha_1, ..., \alpha_R$ and β, we can attempt to estimate probability of a new comment with n words having subreddit r using Bayesian inference:

$$P(subreddit = r | commentd = w_1, ..., w_n) = \frac{P(d|\alpha_r, \beta) * P(subreddit = r)}{\sum_{i=1}^{R} P(d|\alpha_i, \beta) * P(subreddit = i)}$$

where $P(subreddit = i)$ is estimated by relative frequency in training data and we estimate

$$P(d|\alpha_r, \beta) = P(w_1, ..., w_n|\alpha_r, \beta) = \prod_{i=1}^{n} P(w_i|\alpha_r, \beta)$$

$$P(w_i|\theta_r, \beta) = \sum_{k=1}^{K} P(w_i|z_k, \beta)P(z_k|\alpha_r, \beta) = \sum_{k=1}^{K} \beta_{w|k} * \alpha_{r,k}$$

3.3 Measurement

We measure the *absolute* success of our methods by calculating what percentage of the test set is correctly categorized. We believe this is representative of how useful such methods would be in real applications.

We can learn more about our results by looking at *relative* success: that is, what subreddits we seem to handle well, and which ones we don't. For this more subtle quality, we use the F_1 score, which is a harmonic mean of two other metrics: precision and recall. *Precision* is the fraction of comments classified to a particular subreddit that actually belong in it. It is a measure of the number of false positives. *Recall* is the fraction of comments actually from a particular subreddit that are correctly classified to it. It is a measure of the number of false negatives.

4 Analysis

Ultimately, we found that multi-alpha LDA yielded the best results, correctly classifying 21% of comments from a representative set. However, outlier-based clustering provided some valuable information about the nature of the data, which is worth examining before proceeding to our more in-depth analysis of multi-alpha LDA.

4.1 Accuracy of Outlier-Based Clustering

One of the initial problems found with this approach was due to generality of the majority of the comnents, leading to vastly imprecise and volatile clustering that was sensitive to even the minutest outliers. Our hypothesis was that this general class of comments could then belong to a separate cluster that was non-specific to the actual subreddit but would form the bulk of the comments. Setting k as the number of subreddits plus one (for the general category), we obtained that most of the comments did fall into a general category, but there was no evidence to prove that the other categories were not also smaller general categories. We tested this by observing the final distribution of the points from a single subreddit. Irrespective of the subreddit, the distribution remained similar. This observation indicated that subreddits weren't the deciding factor for clustering, and hence led to the observation that topic modeling could be better acheived with EM in the form of LDA.

4.2 Accuracy of Multi-Alpha LDA

Our LDA methods fared somewhat better. In a test set of five subreddits, we achieved a correct classification rate of 50%. On a test set culled from all of the default subreddits, we correctly classified 21%. This seems like a rather unfortunate figure, but by examining the F_1 scores of the various subreddits, we can see where our LDA classifier was successful and where it was struggling.

/r/AskReddit tops the list, mostly because it was the largest subreddit by an order of magnitude, and therefore indeterminate comments were more likely to be classified to /r/AskReddit. Because of the dominance of /r/AskReddit comments, that turned out to be correct for a very large fraction of the data set.

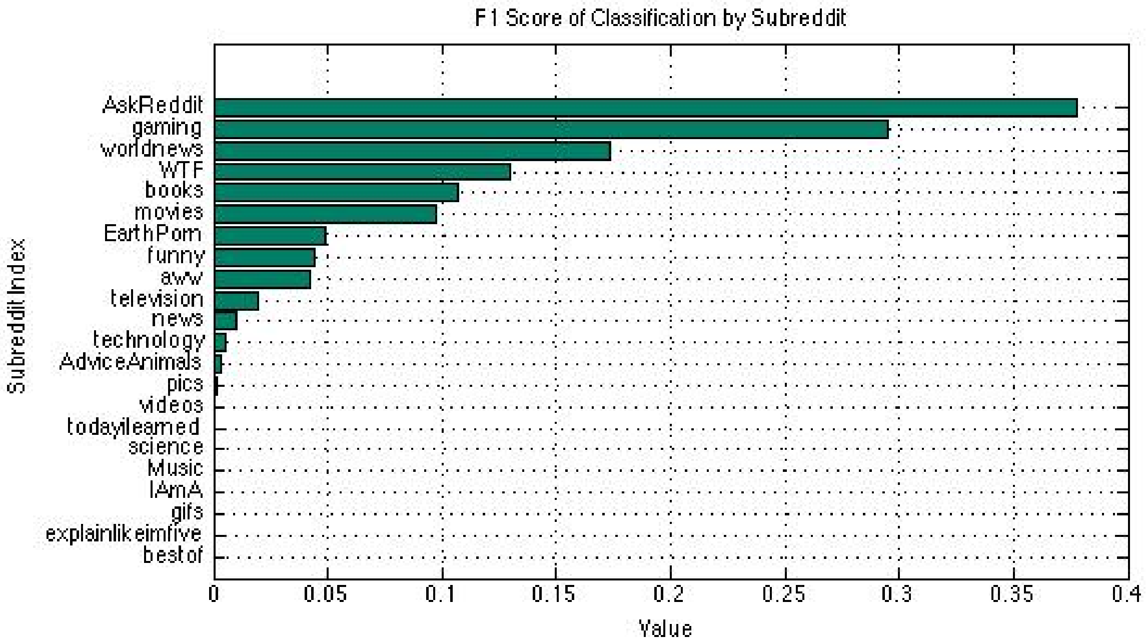

Figure 3: F_1 scores of classification for the default subreddits, sorted in descending order

Looking at the components of F_1, we can see that /r/AskReddit has only the third-highest recall, behind /r/gaming and /r/movies. In the face of the dominance of /r/AskReddit and their relatively small training sets, /r/worldnews, /r/WTF, /r/books, and /r/EarthPorn (the last with less than 300 comments total) also managed to post reasonable F_1 scores. Note that these subreddits are all content-based, i.e., intended to host discussions on fairly specific topics.

Finally, we can see a large number of subreddits which were actually never guessed, and thus have undefined F_1 scores. As we might expect, most of these subreddits are format-based rather than content-based. Subreddits such as /r/videos, /r/todayilearned, /r/IAmA, and /r/gifs could contain discussion on essentially any theme, making it difficult for our LDA learner to pin down distinctive topics for them.

5 Future Work

Although we've made a strong start, there are potentially many other (possibly more accurate) methods that fell outside the scope of our research. Most prominently, the bag-of-words model may oversimplify comments, and an approach that uses bigrams/trigrams to handle phrases, or even (more ambitiously) attempts to semantically parse comments, could possibly extract more meaning.

Additionally, researchers who simply have access to more computational resources than us might be able to improve classification through brute force. To make our data set manageable, we had to reduce it to a fraction of comments from a handful of subreddits. More computing power would allow us to analyze much more data, and, in the case of LDA, run

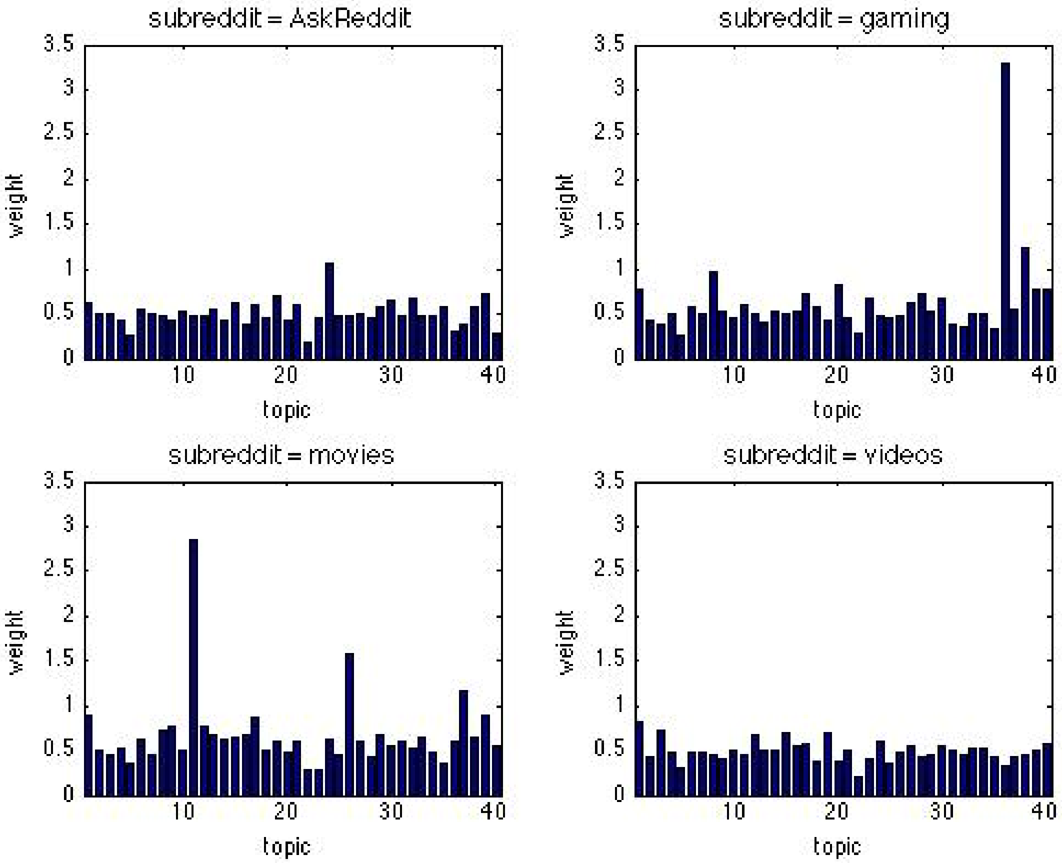

Figure 4: Alphas for various subreddits. Other than /r/AskReddit, most of the ”successful” subreddits exhibit patterns like /r/gaming or /r/movies, with large spikes at one or two topics. Less successful subreddits, like /r/videos, tend to be uniform.

EM for more iterations.

6 Conclusion

We proposed two novel machine-learning methods for categorizing Reddit comments into subreddits. The first method, outlier-based clustering, ran into difficulties with the generic nature of the majority of the comments. The second method, however, multi-alpha LDA, acheived modest success in classifying comments, particularly from focused-content subreddits such as /r/gaming and /r/worldnews. On the whole, we feel that we have taken a strong first stab at a difficult problem.

References

[1] A. TaalimaneshandM. Aleagha. Recommendationsfor reddit users. http://cs229.stanford.edu/proj2012/TaalimaneshAleagha-RecommendationsForRedditUsers.pdf, 2012.

[2] Anonymous Authors. Title classication of reddit posts: predicting not safe for work content. http://www.cs.ubc.ca/~nando/540-2013/projects/p52.pdf, 2013.

[3] @joelsemar. reddit_classifier. https://github.com/joelsemar/reddit_classifier, 2013.

[4] Nick Johnson. Guessing subreddits with the prediction api. http://blog.notdot.net/2010/06/Trying-out-the-new-Prediction-API, 2010.

[5] /u/chicken_bridges. Hierarchical clustering of subreddits based on user participation. http://www.reddit.com/r/TheoryOfReddit/comments/1ndbjd/hierarchical_clustering_of_subreddits_based_on/, 2012.

[6] /u/cupcake1713. New default subreddits? omgomgomg. http://blog.reddit.com/2013/07/new-default-subreddits-omgomgomg.html, 2013.

[7] /u/sharkbait784. A graph of reddit, linking subs based on internal posts. http://www.reddit.com/r/TheoryOfReddit/comments/1e596w/a_graph_of_reddit_linking_subs_based_on_internal/, 2013.

[8] /u/twentythree_nineteen. I downloaded 600,000 reddit comments over a week. http://www.reddit.com/r/datasets/comments/1r76wp/i_downloaded_600000_reddit_comments_over_a_week/, 2013.

[9] Adrien Verster. A subreddit interaction map. http://ajverster.github.io/blog/2013/04/01/redditinteractionmap/, 2013.