Exploring Tweet Engagement in the RecSys 2014 Data Challenge

Jacek Wasilewski
The Insight Centre for Data Analytics
School of Computer Science and Informatics
University College Dublin
jacek.wasilewski@insight-centre.org

Neil Hurley
The Insight Centre for Data Analytics
School of Computer Science and Informatics
University College Dublin
neil.hurley@insight-centre.org

ABSTRACT

While much recommender system research has been driven by the rating prediction task, there is an emphasis in recent research on exploring new methods to evaluate the effectiveness of a recommendation. The Recommender Systems Challenge 2014 takes up this theme by challenging researchers to explore engagement as an evaluation criterion. In this paper we discuss how predicting engagement differs from the traditional rating prediction task and motivate the rationale behind our approach to the challenge. We show that standard matrix factorization recommender algorithms do not perform well on the task. Our solution depends on clustering items according to their time-dependent profile to distinguish topical movies from other movies. Our prediction engine also exploits the observation that extreme ratings are more likely to attract engagement.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Evaluation

1. INTRODUCTION

In this paper we discuss the Recommender System Challenge 2014 [3], the motivations behind the methods that we explored to address the challenge and the results that we obtained. Given a set of tweets generated automatically from the IMDb app, the challenge is to predict which of these tweets will attract *engagement* in the form of retweets or favorites. The challenge provides 389 days of data, including user id, movie id, rating, time-stamp and full tweet metadata, split into three subsets: training (315), test (34) and evaluation (40). Given the evaluation subset of data, with favorite and retweet information omitted, the goal is to order each user's tweets in decreasing order of engagement, defined as the sum of favorites and retweets. The tweets have an identical, automatically generated structure, consisting of just the movie and the user's rating. Engagement must therefore be determined from the context in which the tweet is issued. We might expect popular movies to attract more engagement than unpopular ones and such side information can be obtained from various external sources. However, in our approach to the challenge, we decided to exploit only that data that is made directly available in the challenge, to see how much could be learned from the dataset alone.

RecSysChallenge'14, October 10, 2014, Foster City, CA, USA.

http://dx.doi.org/10.1145/2668067.2668075.

2. PREVIOUS RESEARCH

Accurately predicting the user's interests, as expressed by explicit or implicit ratings, has been the main drive of the recommender systems field up until recently. A wider perspective towards recommendation utility, including but beyond prediction accuracy, has been expressed in the last decade or so, and recently more and more papers are appearing that examine different aspects of utility, such as diversity, novelty or serendipity [7, 5]. This has led to new recommendation algorithms and new evaluation methodologies. The motivation of the Recommender System Challenge 2014 is to explore such 'beyond accuracy' aspects of recommendation. However, the requirements of this challenge are quite different to those that have been explored in previous recommender system research. We may think of the tweets that form the input to this challenge as general recommendations to a wide audience consisting of everyone who happens to read the tweet. If they are personalized recommendations, it is only in the sense that tweeters may expect their followers to have similar tastes to them, or at least to be influenced by their tastes. However, the challenge does not measure satisfaction with the recommender, but rather reaction to the recommendation. While a favorite might indicate general agreement with the tweet, retweeting may indicate that this is a surprising or interesting tweet. Collaborative filtering algorithms have been motivated by the intuition that similar users like similar items – but does this intuition follow through on this challenge? Some work has been carried out on engagement in Twitter, e.g. [2], uses word analysis to predict different retweet behaviors on Twitter but this requires access to a corpus of tweets from each user. In [6], a classifier to predict whether a user will retweet uses social-based, content-based, tweet-based and author-based features. Of these, only the tweet-based features can be applied directly to our dataset and even these are not very relevant. For example, the classifier uses the number of hashtags in a tweet. While it is possible that some users may have added extra

hashtags to the automatically generated IMDb tweet, we find that this does not correlate with engagement.

3. THE EVALUATION CRITERION

The challenge uses the nDCG@10 to evaluate the rankings of tweets by engagement. Before proceeding with ranking algorithms, it is worth exploring the characteristics of this measure. Firstly, it is noteworthy that the vast majority of tweets obtain no engagement and, in fact, for many users, none of their tweets obtain engagement . As engagement is the relevance measure in the nDCG@10 score, the ordering of tweets for these users can be arbitrary, as it will not affect the score. Randomly ordering the tweets in the test set results in an nDCG@10 of 0.749. Another useful operation point is the solution to the binary classification problem: i.e. if a binary predictor correctly predicts which tweets in the test set do and do not obtain positive engagement, then the resulting ordering obtains an nDCG@10 of 0.9862. Therefore solving the binary classification problem might be a useful step in solving the ranking problem. Moreover, we can examine the relevance of any particular feature to the ranking problem by computing the nDCG@10 score for a ranking based on this feature. A relevant feature should obtain a score greater than the random score.

4. EXPLORING THE DATASET

We explored the dataset in order to obtain some intuitions about the sort of tweets that attract positive engagement. From the outset, it is clear that there are a number of different types of features that might determine the attractiveness of a tweet, namely:

- **User-based** features: characteristics of the people who tweet the rating;
- **Item-based** features: characteristics of the movies that are the subject of the tweet;
- **Content-based** features: characteristics of the tweet itself;
- **Context-based** features: the context in which the tweet is issued, in particular the time at which it is tweeted.

As the challenge is to rank the tweets of each particular user in correct order of engagement, we initially felt that user-based features may be less important than other features. For example, a user with many friends is likely to attract more engagement than one with few friends; however, the issue is not to determine the overall amount of engagement that the user attracts, but rather to rank his tweets by engagement. We could use the user's general activity level to determine the importance of getting this ranking correct, but this general activity level would not per-se help to rank the user's tweets.

Item-based features seemed more important in determining the ranking – people may be more inclined to comment on a popular movie than an obscure one. Following from the usual reasoning that motivates recommender systems, we might expect that users fall into communities that have their own niche tastes, that movie popularity within each community might differ and hence that different communities may be more interested in discussing particular movies than others. While a movie's popularity is available from many online sources, restricting ourselves to the challenge dataset, we can estimate popular movies as those that have obtained some engagement in the past (i.e. in the training set).

The automatically generated tweets generated by IMDb have almost no distinguishing features – they simply report the movie and its rating. Nevertheless, the rating itself provides an indication of the newsworthiness of the tweet. If the tweet is not saying anything interesting, then it is unlikely to be retweeted. Hence, we might expect *opinionated* tweets i.e. tweets that rate movies on the extremes of the rating scale to be more likely to be tweeted than tweets that rate at the middle of the rating scale.

Movies are events in time – there is a window around their release in which they are topical. Beyond this window, movies obtain some longer-term status. Renowned movies, such as Godfather, may become icons of a particular genre and, as such, will always generate a certain amount of commentary, particularly when people dare to challenge the general consensus. We might think of the engagement that a movie attracts as decomposing into these two time frames – a short-term engagement around the time that the movie is topical and a longer-term background engagement that reflects the standing that the movie has in popular culture.

From these musings, our intuition suggests that:

- Regardless of user or movie, engagement will be correlated with the tweet rating, with rating extremes more likely to generate engagement.
- Engagement will be reflected in movie popularity with user communities having different movie popularity profiles.
- Determining whether a movie is in its background or topical phase may be critical in predicting its likely engagement. This may be achieved through an analysis of the engagement profile of the movie over time.

In these paper we will focus on a solution that follows our intuition, similar solution and deep data analysis can be found in [1].

4.1 Dataset correlations

In Figure 1, we plot the engagement level of tweets against the rating provided for the movie in the tweet. From this plot, we can see that extreme ratings are indeed more likely to obtain engagement. Also, we note that high ratings attract more engagement than lower ratings. This simple observation leads to a simple ranking algorithm of the following form, which pushes tweets with middle ratings to the bottom of the ranking:

if (rating $<$ min_threshold or rating $>$ max_threshold) **then**
    engagement = rating;
else
    engagement = 0
end if

We find that when `min_threshold`=2 and `max_threshold`=6, we obtain an average score of 0.8121 for the nDCG@10, over random orderings of equal ratings.

A second simple observation is that movies that attracted engagement in the past are somewhat more likely to attract

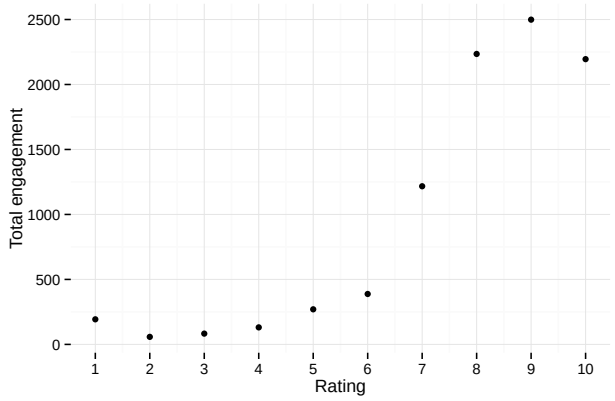

Figure 1: Total sum of engagement received for each rating.

future engagement. Sorting tweets according to the sum of engagements obtained by tweets of the movie in the training set, we find a small positive effect, which, by itself, obtains a nDCG@10 score of 0.7524. Combining with the rating predictor, by sorting items of equal rating according to their previous engagement level results in a very small improvement over the average rating of 0.8155.

Another plausible intuition is that popular movies that are most often rated in the training dataset may also be more likely to attract engagement. Sorting by popularity obtains an nDCG@10 score of 0.7509, but produces no positive effect when combined with ratings, obtaining an nDCG@10 score of 0.805. Similarly, we examine the activity levels of users, as the number of times that they have tweeted in the dataset. Using this to predict engagement obtains a score of 0.74924. Combined with the rating predictor, we obtain nDCG@10 of 0.8133.

For each tweet, besides information about rating and movie, we can explore additional information such as hashtags, or user mentions. By default, the IMDb application inserts one hashtag which is the same for all tweets but there are some tweets where users added their own hashtags, but the correlation between number of hashtags and engagement does not exist. However, we do find a weak correlation between number of user mentions and engagement. Our intuition is that mentioned users are notified of the tweet and this may increase the chances of getting engagement from these mentioned users. Using only this feature as a predictor gives an nDCG@10 score of 0.7583.

4.2 Logistic Classification

We trained a logistic classifier on binary labels of whether engagement occurs or not. Using the above analysis, we created a 4-component feature vector representation of a tweet. The first three components describe the rating, $f_i = r \times I_i, i = 1, \ldots, 3$, where I_i are indicator variables, that split the ratings into the three regions as described above. The fourth captures the activity level of the item that's being tweeted. We measure this as the ratio of the total sum of the engagements received by tweets about the item divided by the total number of tweets about the item that receive some engagement. A large number suggests that the item tends to attract attention when tweeted. We scaled this

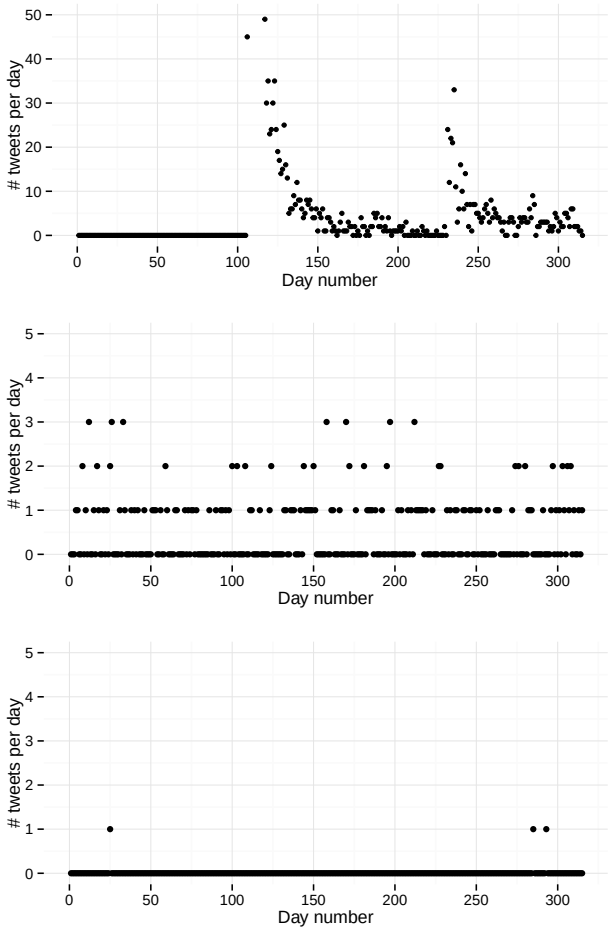

Figure 2: Total number of tweets per day for three different items.

feature by taking a logarithm. Note that this value is set to 0 if the item does not appear in the training set.

Sorting tweets by their probability of engagement, as output of the classifier, gave an nDCG@10 score of 0.8237. Other simple measures of item popularity or engagement level were not successful in improving classifier performance over this level at this point.

As a simple measure of item *topicality*, we counted the number of engagements that an item received in the previous n days prior to the tweet creation date. However, this failed to improve the classifier's performance.

5. ITEM CLUSTERING

To improve classification performance, we looked for methods to cluster the items into meaningful clusters according to the levels of engagement that they can expect to get. Initially, we split the item popularity range into three intervals (using k-means), creating three item clusters. However, the addition of this cluster label to our classifier failed to improve its performance. The reason may be the fact that this simple clustering groups items that had got similar number of tweets over the full time period but it does not give us any information of whether these tweets were issued over e.g. 5 days or 100 days. However, topical movies are more likely to

k value	Silhouette score	nDCG@10 score
2	0.9818	0.8312
3	0.9557	0.8317
4	0.9385	0.8296
5	0.9252	0.8283
6	0.7946	0.8302
7	0.7732	0.8286

Table 1: Silhouette and nDCG@10 scores of k-means results for different k values.

have a bursty time profile, we get a high number of tweets issued over a short period of time. In Figure 2, we plot number of tweets that three different items receive per day. We can observe that each of them has different characteristics when it comes to how often and how many tweets about them occur, that leads us to hypothesis that we might have different types of items. Our solution is to cluster the items according to their full time series profile, in order to distinguish different item types. Specifically, we generate a time series representing the daily numbers of tweets about the movie over all days of the training period. As a result for each item we get vector of 389 numbers representing the number of tweets per day that movie obtains. Then we transform the series into the frequency domain using fast Fourier transformation and cluster the resulting vectors using k-means clustering. To pick k value for k-means algorithm we decided first to check the clusters itself by calculating Silhouette scores for different k - scores can be found in Table 1. At the same time, for k values that got the highest score ($k = 2, 3, 4, 5$) we run the solution to check which value would get the highest nDCG@10 score on our data - as a result we decided to pick $k = 3$ which as well fits our intuition about different types of items. The clusters correspond to different levels of overall activity. We label the clusters $1, \ldots, k$ in ascending order of total activity in the cluster. Adding the value of activity of the cluster, we get a small improvement to the nDCG@10 score.

6. FINAL SOLUTION

As our final solution, to predict and rank the engagement we decided to use probabilities produced by logistic regression model, based on combined 7 features that we identified before:

- $rating_1(i)$ - rating's feature for the rating value less then lower threshold,
- $rating_2(i)$ - rating's feature for the rating values between lower and upper threshold,
- $rating_3(i)$ - rating's feature for the rating value higher then upper threshold,
- $pop(i)$ - item's feature representing item popularity; scaled by logarithm,
- $eng\_lvl(i)$ - item's feature representing the engagement level for items that got engagement; scaled by logarithm,
- $cluster\_eng\_lvl(c)$ - item's cluster feature representing the engagement level for all items within the cluster; scaled by logarithm,

Feature	Coefficient
$rating_1(i)$	-0.25530677
$rating_2(i)$	-0.45833852
$rating_3(i)$	-0.88634991
$pop(i)$	-0.03864514
$eng\_lvl(i)$	-0.21829483
$cluster\_eng\_lvl(c)$	-0.02983127
$mentioned(t)$	-0.48994379
Intercept	3.31797053

Table 2: Coefficients of logistic regression model.

Feature	Coefficient
$rating_1(i)$	-0.23635914
$rating_2(i)$	-0.43695132
$rating_3(i)$	-0.86541020
$pop(i)$	-0.03666516
$eng\_lvl(i)$	-0.17489093
$cluster\_eng\_lvl(c)$	0.03019511
$mentioned(t)$	-0.20440849
$is\_retweet(t)$	-1.21393216
$has\_retweets(t)$	-1.1957637
Intercept	3.36787059

Table 3: Coefficients of logistic regression model enriched with retweets information.

- $mentioned(t)$ - binary tweet's feature indicating that tweet mentions some user,

where t represents tweet, i represents item, c represents item's cluster. Using these features we trained our model, resulting in 0.8317 nDCG@10 score on the test dataset. Learnt coefficients for all features can be found in Table 2.

6.1 Retweets

While the number of retweets each tweet receives in the dataset has been removed, the JSON metadata still retains the information that any particular tweet is a retweet of another. In fact there are around 330 retweets in the test dataset. It seems unfair to directly use this information – instead perhaps these test cases should have been removed from the dataset.

Adding one additional binary feature that indicates whether tweet is a retweet or not, improves the nDCG@10 score to 0.8343.

Inspired by the article [1] we have added another feature indicating if tweet has retweets in the dataset and that improves the score to 0.8726 - features and coefficients can be found in Table 3. Using these features is controversial in the sense of the challenge task but also it puts some boundaries on when it can be used, especially using *has_retweets* feature which boosts the result but also acts like an prophetic feature. If the task is to give the ranking of tweets, at some point of time using a snapshot of data without exact information about the engagement, then using *has_retweet* feature is not such a infraction. Although if we would like to predict the tweet's engagement the same moment as tweet occurs and then just to rank them later, such a feature can not be used.

7. OTHER STRATEGIES

We experimented with a number of other strategies, including user clustering and user activity levels, to no avail.

7.1 Recommender Algorithms

One hypothesis is that engagement is simply a form of implicit rating. This suggests that if we organize the training data into a standard recommender system input form, by computing for each user and movie, the total number of engagements obtained over all tweets involving this pair in the dataset, then we can train a standard recommender system algorithm, to predict future engagements. Note that in employing this strategy, we are ignoring the values that correlate most strongly with engagement, namely the ratings.

There is a wide variation in engagement values – while the majority of tweets receive no engagement, there are some with over 100 engagements. To simplify evaluation, we focus on the binary problem – where we set the implicit rating value as 1 or -1, depending on whether or not the user-movie pair receive any engagement in the training period. Training a standard[1] matrix factorization algorithm, to minimize the root mean squared prediction error on the training set, results in a RMSE of 0.80 on the test set. However, this translates into only a small gain over random ranking, giving 0.76 on the nDCG@10 score. Similarly, when the ranking objective of [4] is used instead, we get a similar performance. Hence, the collaborative filtering heuristic that similar users will obtain engagement for similar items holds only weakly in this dataset.

8. CONCLUSIONS

Performance of several approaches have been checked in order to find the best solution on the task of raking based on predicted engagement. We find that the rating by itself is by far the strongest indicator of likely engagement, all other correlations that we have tried to exploit have resulted in only incremental gains at best, with the final nDCG@10 score of 0.8317. In question is the use of retweet metadata that remains in tweets after removing information about the actual number of favorites and retweets, although it is another strong indicator of engagement - using this information gives the result of 0.8726.

9. ACKNOWLEDGMENTS

The Insight Centre for Data Analytics is supported by Science Foundation Ireland under Grant Number SFI/12/RC/2289

10. REFERENCES

[1] D. Loiacono, A. Lommatzsch, and R. Turrin. Recsys challenge 2014: Learning to rank. 2014.

[2] J. Mahmud, J. Chen, and J. Nichols. Why are you more engaged? predicting social engagement from word use. *arXiv preprint arXiv:1402.6690*, Feb 2014.

[3] A. Said, S. Dooms, B. Loni, and D. Tikk. Recommender systems challenge 2014. In *Proceedings of the eighth ACM conference on Recommender systems*, RecSys '14, New York, NY, USA, 2014. ACM.

[4] G. Takács and D. Tikk. Alternating least squares for personalized ranking. In *Proceedings of the sixth ACM conference on Recommender systems*, RecSys '12, pages 83–90, New York, NY, USA, 2012. ACM.

[5] S. Vargas and P. Castells. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems*, RecSys '11, pages 109–116, New York, NY, USA, 2011. ACM.

[6] Z. Xu and Q. Yang. Analyzing user retweet behavious on twitter. In *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. IEEE, 2012.

[7] M. Zhang and N. Hurley. Niche product retrieval in top-n recommendation. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2010, Toronto, Canada, August 31 - September 3, 2010, Main Conference Proceedings*, pages 74–81. IEEE, 2010.

[1] We minimize $\sum_{u,i}(\mathbf{p}_u^T\mathbf{q}_i - r_{ui})^2$ i.e. we do not include bias terms, nor take account of time-dependency