Predicting IMDB Movie Ratings Using Social Media

Andrei Oghina, Mathias Breuss, Manos Tsagkias, and Maarten de Rijke

ISLA, University of Amsterdam, Science Park 904, 1098 XH Amsterdam, The Netherlands
aoghina@gmail.com,mathiasbreuss@gmx.at,{e.tsagkias,derijke}@uva.nl

Abstract. We predict IMDb movie ratings and consider two sets of features: surface and textual features. For the latter, we assume that no social media signal is isolated and use data from multiple channels that are linked to a particular movie, such as tweets from Twitter and comments from YouTube. We extract textual features from each channel to use in our prediction model and we explore whether data from either of these channels can help to extract a better set of textual feature for prediction. Our best performing model is able to rate movies very close to the observed values.

1 Introduction

Information Retrieval (IR) is expanding from its core business of ranking documents to encompass a range of applications where objects in general need to be ranked. Modern IR systems use multiple signals from multiple sources for ranking these objects effectively. Such tasks can be viewed as a *cross-channel prediction task*: observe signals in a list of channels to predict a signal in another channel. We focus on one such task: predicting movie ratings from multiple social media signals. Movie ratings are influenced by demographics and by personal factors that are hard to model explicitly. We hypothesize that correlations can be found between ratings provided by the Internet Movie Database (IMDb) and activity indicators around the same artefacts (i.e., movie titles) in other channels, such as social media. As an example, the movie "Inception" generated 167K tweets and rates 8.9/10 vs. the movie "Justin Bieber: Never Say Never" which generated 25K tweets and rates 1.1/10. Most prediction work on movies focuses on forecasting revenues, not ratings. Some use the publication rate of tweets relevant to a movie title for forecasting box-office revenues [1]. Joshi et al. [4] combine surface features with text features extracted from reviews. Tsagkias et al. [6, 7] predict the volume of comments on online news, using a set of surface and textual features.

2 Feature Engineering

To address the task of predicting movie ratings from social media, we identify activity indicators that can be translated into extractable features. We group indicators, and the derived features, into two classes for understanding their effect in prediction performance. *Quantitative indicators* aim at capturing the amount of activity around a movie title (e.g., how many people talk about a movie) and result in *surface features*. *Qualitative indicators* aim at capturing the meaning of the activity (i.e., what people say about a movie) and result in *textual features*. Surface features strongly depend on the underlying social media platform as each platform has its own activity indicators (e.g., number of diggs in Digg, number of views in YouTube). Our choice of Twitter and YouTube

as content providers, resulted in the following surface features: *views*, *number of comments*, *number of favorites*, *number of likes* and the *number of dislikes*, the *fraction of likes over dislikes* for each trailer clip on YouTube, and the *number of tweets* on Twitter. Each feature is represented by the natural logarithm of its frequency.[1] Textual features are extracted by comparing the log-likelihood of a term in two corpora [5]. These corpora consist of tweets and YouTube comments associated with the top- and bottom-N movies, based on their IMDb ratings; $N = 5$ shows the optimal performance. Examples of extracted positive textual features include the stems *amaz*, *perfect*, *awesom*; negative ones include *stupid*, *worst*, *terribl*.

3 Experimental Setup

For predicting movie ratings using signals from social media, we conduct regression experiments. The data set used consists of 70 movies, and their ratings as reported on IMDb on April 4, 2011. From this set, 10 movies were kept aside for extracting textual features, leaving 60 titles for testing. The data set was complemented with Twitter and YouTube data. Our Twitter data consists of 1,6M tweets published between March 4, 2011 and April 4, 2011 that mention a movie title. Our YouTube data consists of metadata and 55K comments of movie trailers. We use the linear regression implementation in the WEKA toolkit [3]. All regression results reported are calculated using ten-fold cross validation on a set of 60 movies. We report on Spearman's ρ and on standard regression measures: mean absolute error (MAE), root squared mean error (RMSE). For ρ higher is better, for MAE and RMSE lower is better. Bold face indicates best performing feature per metric. Significance is tested using a two-tailed paired t-test and is marked as $^{\blacktriangle}$ (or $^{\blacktriangledown}$) for significant differences for $\alpha = .01$, or $^{\triangle}$ (and $^{\triangledown}$) for $\alpha = .05$.

4 Results and Analysis

The first movie rating prediction experiment we carry out uses individual surface features, and their combination, to rate movies from 1 to 10. We set our baseline to the number of tweets that mention a particular movie as it has proven a strong predictor in [1]. Table 1 shows that the strongest feature is the fraction of likes over dislikes (likes-per-dislikes) peaking the correlation coefficient at 0.4831. Although likes and dislikes as individual features show poor correlation, their fraction yields the strongest surface feature. Next, we combine all surface features, and consider subsets of features that optimize performance using the CfsSubset attribute selection method. The combination of all surface features does not outperform likes-per-dislikes, and attribute selection results in similar performance as likes-per-dislikes (see last two columns in Table 1). Our second experiment is aimed at finding out whether the textual content generated on different social media platforms has equal predictive power. We compare the performance of a linear regression model using textual features extracted from tweets, from YouTube comments, and from their combination using both all textual features and an optimal subset of textual features. Table 2 (columns 1–8) shows that the best performance is achieved when using all textual features from Twitter. Textual features from YouTube show poor performance compared to likes-per-dislikes, which almost doubles when

[1] We conducted experiments where features were represented by their raw, normalized, and log frequencies, and we found that using log helped performance.

Table 1: Regression performance using surface features. Statistical significance tested against *number of tweets*. ("Comm." abbreviates "Comments.")

Metric	**Random**	**Tweets**	**Views**	**Comm.**	**Favorites**	**Likes**	**Dislikes**	$\frac{likes}{dislikes}$	All	Att.Sel.
ρ	-0.3392	0.1128	0.1638	-0.0756	0.1061	0.1209	-0.2880	**0.4831**	0.4261	0.4805
MAE	0.7838	0.7924	0.7945	0.7923	0.7912	0.8020	0.7982	0.6993	0.7089	**0.6912**
RMSE	0.9963	0.9838	0.9744	0.9968	0.9843	0.9817	1.0194	**0.8620**$^{\triangle}$	0.9039	0.8649$^{\triangle}$

Table 2: Regression performance using textual features extracted from Twitter (T), YouTube (YT) comments, and their combination (T+YT). Significance tested against *likes-per-dislikes* and *T* without attribute selection. Last col.: combination experiment.

		Without att. sel.			With att. sel.			
Metric	$\frac{likes}{dislikes}$	**T**	**YT**	**T+YT**	**T**	**YT**	**T+YT**	$\frac{likes}{dislikes}$**+T**
ρ	0.4831	**0.7870**	0.2768	0.6625	0.4739	0.5029	0.6420	**0.8539**
MAE	0.6993	**0.5051**$^{\blacktriangle}$	0.9090$^{\triangledown}$	0.5828	0.7201	0.6971	0.6045	**0.4203**$^{\blacktriangle}$
RMSE	0.8620	**0.6084**$^{\blacktriangle}$	1.1140$^{\triangledown}$	0.7395	0.8826	0.8739	0.7675	**0.5227**$^{\blacktriangle}$

we use only an optimal subset of the features (chosen from CfsSubset). The combination T+YT outperforms the baseline but not the Twitter model. In our third experiment we study the effect of using the best surface and textual model, namely, the likes-per-dislikes, and textual features from Twitter; see last column in Table 2. The combination proves beneficial, and significantly outperforms both individual models. Experiments that involved adding YouTube features, were unable to match the baseline.

The performance differences in Twitter and YouTube textual features ask for further investigation. First, we look at merging data from both datasets before extracting textual features. The resulting set of features is unable to outperform textual features from Twitter, possibly due to topical drift [6] ($\rho = 0.1775$, 0.9624 MAE, 1.2046 RMSE). Second, we investigate whether manual curation of the resulting textual features helps prediction performance. We manually curated the set of textual features from the best performing platform (Twitter), discarding features that do not correspond to proper words, or features that carry neutral meaning; this left us with 102 features. Performance now reaches $\rho = 0.8460$, 0.4208 MAE, and 0.5276 RMSE, an improvement of 20% over the *Twitter* model without attribute selection, almost matching the best run performance (*likes-per-dislikes+Twitter*). The combination of curated textual features with the likes-per-dislikes feature achieves the best performance at $\rho = 0.8915$, 0.3525 MAE, 0.4459 RMSE, a significant improvement at 0.01 level over the best run in Table 2.

We tested the normality of the residuals by plotting the quantiles of their empirical distributions versus the quantiles of the theoretical (normal) distributions [2] in Fig. 1. The residuals show a reasonable match with normal distributions, with the residuals of the combined model (*likes-per-dislikes+Twitter*) showing the least deviation from the straight line that denotes perfect correlation with the normal residuals. Possible explanations of strong deviations are that textual features refer to other movies than the one at hand, e.g., the original Tron movie, the demographics of the users who rate the movie may differ from those who share comments, and finally, the movie release date may play a role, as the ratings may oscillate substantially shortly after the release.

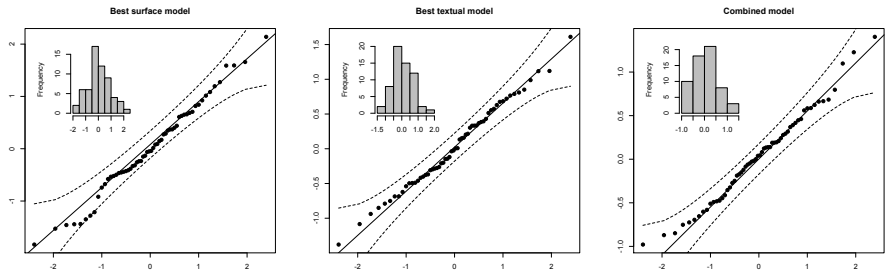

Fig. 1: The quantile-quantile plot of the residuals of the linear regression models: *likes-per-dislikes+views* (Left), *Twitter* (Middle), and *Combined* (Right). X-axis: Normal quantiles. Y-axis: Residual quantiles. The inset shows the distribution of the residuals.

5 Conclusions and Outlook

We addressed the task of predicting movie ratings using data from social media. We identified qualitative and quantitative activity indicators for a movie in social media, and extracted two sets of surface and textual features. The fraction of the number of likes and dislikes on YouTube, combined with textual features from Twitter lead to the best performing model, with strong agreement with the observed ratings and high predictive performance. We plan to consider more social media channels, look at the effect of user demographics, and develop methods that predict ratings far ahead in time.

Acknowledgments. This research was supported by the European Union's ICT Policy Support Programme as part of the Competitiveness and Innovation Framework Programme, CIP ICT-PSP under grant agreement nr 250430, the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreements nr 258191 (PROMISE Network of Excellence) and 288024 (LiMoSINe project), the Netherlands Organisation for Scientific Research (NWO) under project nrs 612.061.814, 612.061.-815, 640.004.802, 380-70-011, 727.011.005, the Center for Creation, Content and Technology (CCCT), the Hyperlocal Service Platform project funded by the Service Innovation & ICT program, the WAHSP project funded by the CLARIN-nl program, under COMMIT project Infiniti and by the ESF Research Network Program ELIAS.

6 References

[1] S. Asur and B. A. Huberman. Predicting the future with social media. *CoRR*, abs/1003.5699, 2010.

[2] J. Fox. *Applied Regression Analysis, Linear Models, and Related Methods*. SAGE Publications, February 1997.

[3] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: an update. *SIGKDD*, 11:10–18, November 2009.

[4] M. Joshi, D. Das, K. Gimpel, and N. A. Smith. Movie reviews and revenues: An experiment in text regression. In *Proceedings of NAACL-HLT*, 2010.

[5] G. Mishne and M. de Rijke. Capturing global mood levels using blog posts. In *AAAI-CAAW '06*, pages 145–152, 2006.

[6] E. Tsagkias, M. de Rijke, and W. Weerkamp. Predicting the volume of comments on online news stories. In *CIKM 2009*, pages 1765–1768, Hong Kong, 2009. ACM.

[7] E. Tsagkias, W. Weerkamp, and M. de Rijke. News comments: Exploring, modeling, and online predicting. In *ECIR 2010*, pages 191–203. Springer, 2010.