Noname manuscript No.
(will be inserted by the editor)

Modeling and Predicting the Popularity of Online News based on Temporal and Content-Related Features

Steven Van Canneyt · Philip Leroux · Bart Dhoedt · Thomas Demeester

Received: date / Accepted: date

Abstract As the market of globally available online news is large and still growing, there is a strong competition between online publishers in order to reach the largest possible audience. Therefore an intelligent online publishing strategy is of the highest importance to publishers. A prerequisite for being able to optimize any online strategy, is to have trustworthy predictions of how popular new online content may become. This paper presents a novel methodology to model and predict the popularity of online news. We first introduce a new strategy and mathematical model to capture view patterns of online news. After a thorough analysis of such view patterns, we show that well-chosen base functions lead to suitable models, and show how the influence of day versus night on the total view patterns can be taken into account to further increase the accuracy, without leading to more complex models. Second, we turn to the prediction of future popularity, given recently published content. By means of a new real-world dataset, we show that the combination of features related to content, meta-data, and the temporal behavior leads to significantly improved predictions, compared to existing approaches which only consider features based on the historical popularity of the considered articles. Whereas traditionally linear regression is used for the application under study, we show that the more expressive gradient tree boosting method proves beneficial for predicting news popularity.

Keywords online news · popularity modeling · popularity prediction · regression · feature engineering

Steven Van Canneyt, Philip Leroux, Bart Dhoedt, Thomas Demeester
Department of Information Technology
Ghent University - iMinds
Ghent, Belgium
E-mail: {steven.vancanneyt,philip.leroux,bart.dhoedt,thomas.demeester}@ugent.be

1 Introduction

The online consumption of news content, a large and still growing market with respect to the traditional printed media, is undergoing major changes. The original paradigm of users consuming content that was pre-selected by news agents, shifts towards a setting where users themselves decide on which content is relevant to them and their circles, and whom they share it with over social media. As there is a strong competition between online publishers in order to reach the highest possible audience, it is becoming very important to decide which articles to promote on the front page of a news website, and which articles to publish on different social media platforms such as Twitter and Facebook. Therefore, in this paper, we propose a novel methodology to model and predict the popularity of online news articles. These popularity models and predictions can then be used by news agents to optimize their online publishing strategy (which falls outside the scope of the current work).

We first conduct a thorough study to identify the distributions which underlie the view patterns of articles, i.e. the number of visits articles receive over time. This is important in order to understand how the popularity changes over time. This study is performed on the articles published by the Belgian Buzzfeed-like website newsmonkey[1] between April and September 2015. We observe that a view pattern in general consists of several components. The contribution that we refer to as the *direct views*, becomes visible as soon as the article is published on the news publisher's website. However, when the article is additionally published on social media channels, clear additional components in the view patterns start to appear. In this paper, besides the direct views, we will focus on the *Facebook views* and the *Twitter views*. We introduce a model that closely fits these components and demonstrate that this model performs better than baseline log-normal fits [10,19,18]. Additionally, we take the influence of the diurnal cycle on the view patterns into account to further increase the accuracy, without obtaining more complex models.

The proposed approach can be considered one example of a more general idea for dealing with the heterogeneous character of item views. We show that a flexible and complex model for the total number of views can be obtained by separately modeling the most important components, for example those originating from specific social media channels, especially as a response to known events, such as pushing the item on those channels. Our experimental results show that the same, very simple, elementary model can be used for quite different contributions, in this case the direct views, Facebook views, and Twitter views.

As a second contribution, we propose a novel methodology to predict the final popularity of online news articles. As the total number of views consists of easily identifiable components related to the origin of the views (i.e., direct views, Facebook views, Twitter views), we train different regressors to respectively predict the behavior for each of these components. Existing ap-

[1] http://newsmonkey.be

proaches train linear regressors using features based on historical popularity values of the articles [17,15,5,9]. We investigate three ways to improve upon these baseline methods: (a) We explicitly make use of our proposed temporal model underlying the historical view pattern of the considered article, and use its parameters as additional features for the regressors. (b) In addition to using the historical popularity of the articles, we show that a variety of content-based and meta-data related features (such as author, category, emotion...) significantly contribute to improving the popularity predictions. (c) Finally, we show that more complex regression algorithms, as compared to the standard linear regression approach, can further improve the prediction effectiveness.

The remainder of this paper is structured as follows. We start with a review of related work in Section 2. In Section 3 we describe the data acquisition process. Subsequently, in Section 4, we investigate the dynamics of the views received by articles, and propose a simple and effective model to model the view patterns. This model is evaluated in Section 4.5. Our methodology to better predict the final popularity of articles using novel features and advanced regression algorithms is described in Section 5. The experiments and comparisons with existing approaches are described in Section 5.3. Finally, we conclude our work in Section 6.

2 Related Work

The prediction of the popularity of online content has recently attracted a considerable amount of research. Some authors tackled the problem of predicting the popularity of an item before its publication [19,2,1]. Pre-publication predictions are particularly useful for web content characterized by a short lifespan such as online news articles. The researchers in [19,2,1] built classifiers to classify news articles into different classes, such as 'low popularity', 'medium popularity', and 'high popularity'. As quantitative indicators of popularity, they considered the number of comments on an article, the number of associated tweets, and the number of views. However, the researchers in [19] and [1] concluded that it is hard to accurately estimate an article's popularity without incorporating any early-stage popularity information. We did similar pre-publication experiments on our dataset which led to the same conclusion. Therefore, we will focus on post-publication predictions in this paper.

Post-publication prediction methods predict an item's popularity based on the users' attention received early after publication. Kaltenbrunner et al. [10] analyzed the popularity of news articles, and found that the long term target popularity of online content is strongly correlated with its early reference popularity. Based on that observation, they proposed a linear popularity prediction model with the early popularity and a constant multiplication factor as input. The multiplication factor was set to the average growth in the training set. The authors of [17] improved that prediction model by optimizing the multiplication factor specifically for the considered performance metric. Their method showed good predictive performance on several data sets: votes on

Digg stories [17], views of Youtube videos [17], views of blog posts [11], and comments on articles published on a French [18] and Dutch news platform [20, 18].

While the model of Szabo and Huberman [17] seems reasonably accurate, especially given its simplicity, it does have shortcomings. In particular, different pieces of content may display a very similar popularity at an early stage, yet exhibit a diverse popularity behavior afterwards. In other words, despite the observations in [10], online content may experience very different popularity evolution patterns [15,5]. Therefore, the authors of [15,12] investigated whether the use of the historical popularity values of online content between the publication time and an early reference time leads to more accurate predictions of the total popularity at a future target time. Pinto et al. [15] divided the time between publication of the article and the reference time into different intervals, and trained a linear regression model using the number of observed views the articles received during each time interval. This model was further improved by incorporating features constructed from the similarities between the considered view pattern and the training instances. The model proposed in [12] used the retweet pattern of a tweet during its first hour to predict the number of retweets three days after publication. The authors partitioned the first hour into five equally sized time intervals, and then recorded the number of retweets during each time interval. This information was used to describe each tweet by a set of features (such as retweet time series, retweet acceleration, and author). These features were used to determine the most similar tweets in the training set of the given tweet. The predicted popularity was then set to the weighted average of the number of retweets among these similar tweets.

The last category of post-publication prediction methods uses data from one domain (e.g. social media) and transforms it into knowledge to predict content popularity in another domain (e.g. the site where the content was published). Oghina et al. [14] trained a linear regression model based on several textual features extracted from Twitter, as well as various statistics from Youtube, to predict movie ratings on IMDb. The authors of [5] proposed a second-order multiple linear regression model to predict the number of views of online news articles after 7 days. For a given reference time, the model used the total number of views, Facebook shares and Twitter posts of the article, in addition to Twitter statistics such as the average number of followers of people sharing on Twitter and the entropy of the tweets.

The objective of the ECML/PKDD 2014 Predictive Analytics challenge[2] was to predict the number of views, Facebook shares, and Twitter posts of web pages after their first 48 hours online. As input, the popularity trends during the first hour were given. The winner [9] of the challenge combined different ideas of the models proposed in [5] and [15]. Similar to [5], they used second-order multiple linear regression models based on several popularity metrics to predict the number of views. For the given reference time (i.e., one hour after publication), the model considered the number of views, Facebook shares and

[2] https://sites.google.com/site/predictivechallenge2014/

Twitter posts per time interval (i.e. 5 minutes), starting from the publication time until the reference time. Additional features were formed using the publication weekday and hour of the article. The authors of [9] further improved their model by using the ideas presented in [15]. In particular, they also used the similarity of the view pattern to canonical patterns extracted from the training set, in order to improve the model performance. These canonical patterns were constructed by normalizing and clustering all view patterns in the training set.

Our proposed method differs from these post-publication approaches in multiple aspects. We explicitly model the temporal behavior underlying the historical popularity of the articles, and use the resulting parameters as additional features for the regressors. We also consider features related to the content and meta-data of the articles. Finally, we propose the use of a more advanced regression algorithm.

3 Experimental Data

In this study we use data from the newsmonkey[3] online news platform. Newsmonkey is a Buzzfeed-like[4] news website, currently focusing on the Belgian market. Similar to Buzzfeed, newsmonkey combines breaking news with highly shareable stories. Our dataset consists of 2614 articles and the detailed associated click data, which we collected between April 27, 2015 and September 10, 2015. The first three quarters (until July 25, 2015) are used as a training set K, and the last quarter is considered as the test set U (with content from August 6, 2015 onwards, in order to limit the immediate correlation between both). An article's final popularity is measured as the number of views 120 hours (5 days) after publication on the website. Figure 1 shows a boxplot of the total number of views received per day after publication. We observe that most articles have a lifetime considerably shorter than this 5 day period, such that the number of views becomes stable well before 5 days after their initial publication. The average number of views per article is about 2000, of which 62% directly come from Facebook and only 3% from Twitter. This is in line with the strategy of newsmonkey, mainly focusing on optimizing their popularity on Facebook.

Few articles reach a very high number of views, whereas the majority of articles only get a low reach. As an illustration, Figure 2 shows the number of views for all articles as a function of the rank of each article, when sorted by decreasing number of views. Except for the least popular articles, the observed behavior is approximately linear on logarithmic axes. This Zipfian behavior [13,21–24] means that the number of views per article follows a power law.

Figure 3 shows the normalized popularity for the articles in the training set as the fraction of views for each hour of the day, considered separately for

[3] http://newsmonkey.be

[4] Buzzfeed is a popular American news platform, and one of the first who focuses on highly shareable breaking news, original reporting, entertainment, and video.

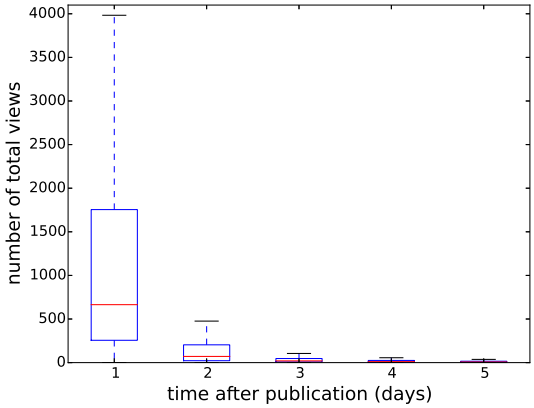


Fig. 1: Boxplot of number of total views received per day after publication, for all articles in the dataset.

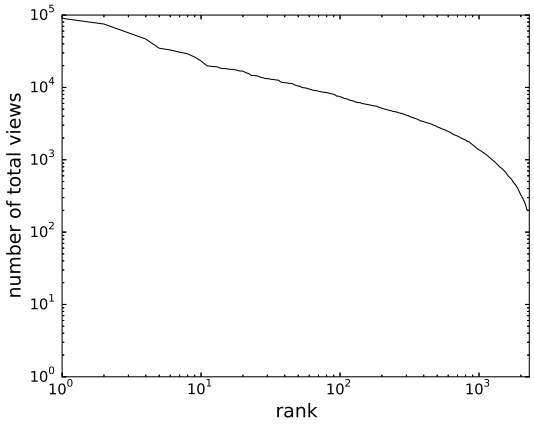


Fig. 2: Zipfian distribution of the number of views for all articles in the dataset, ranked in decreasing order.

the views originating from Facebook (Facebook views), from Twitter (Twitter views), and all remaining views (direct views). We notice that the users are much less active at night than during the day. Also, there is some difference in behavior between the three considered types of views.

4 Popularity Pattern Modeling

A good understanding of how the popularity changes over time and which external elements have the largest impact, is essential in order to create a suitable model, or to design appropriate features for popularity predictions. Therefore

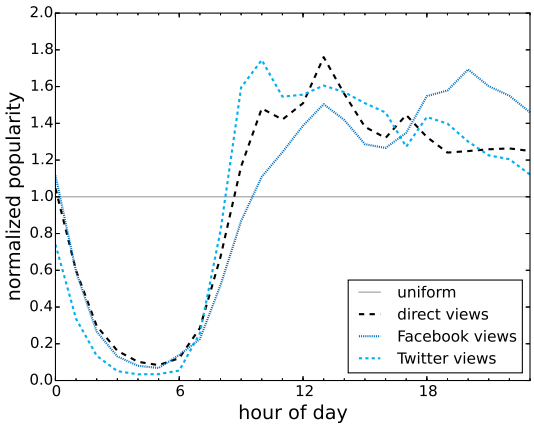


Fig. 3: Normalized number of direct views, Facebook views, and Twitter views for each hour of the day, for the articles in the training set.

in this section we propose a new temporal popularity model. Despite its simplicity, we show that this model is able to accurately capture the temporal behavior of a particular popularity measure for a given article, and compare it with a number of existing models. In this paper, we consider the evolution of the total number of views of each article, measured at hourly intervals. However, the methodology can be easily extended towards other popularity metrics (e.g. Facebook shares) and more fine-grained time intervals. As an illustration throughout this section, we will use the total number of views of a typical article, shown in Figure 4. This particular article was published on Twitter immediately after its publication online, and on Facebook 25 hours later. In Section 5, a prediction model is introduced that makes use of the insights obtained from the proposed temporal model and explicitly uses its parameters as features.

4.1 Log-normal Baseline

In previous work, the popularity of online news articles is often modeled using a log-normal distribution [10, 19, 20, 18]. In particular, its cumulative distribution can be used to model the total number of views at a particular time:

$$v_t^i \approx s^i \cdot \text{clogn}(t; \mu^i, \sigma^i) \tag{1}$$

with v_t^i the observed total number of views of article i at time t, s the scale factor that corresponds to the number of views at infinity, and clogn the cumulative log-normal distribution given by

$$\text{clogn}(t; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \int_0^t e^{-\frac{(\ln(\xi)-\mu)^2}{2\sigma^2}} d\xi \tag{2}$$

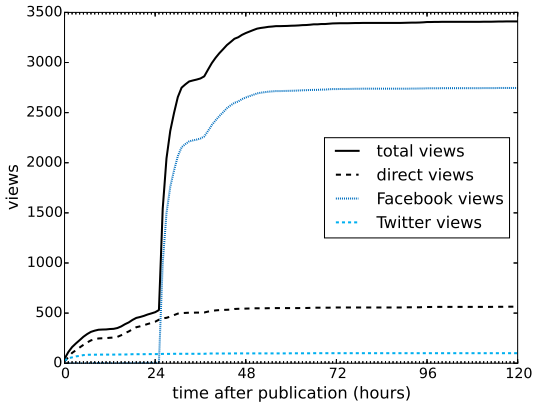
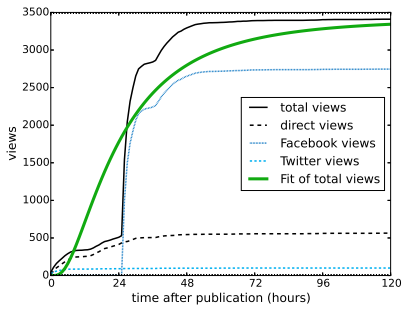
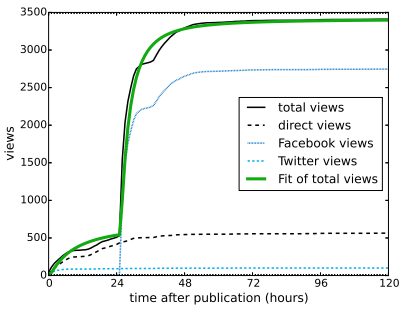


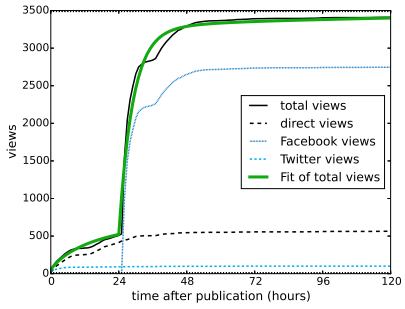
Fig. 4: Number of views of an example article as a function of time.



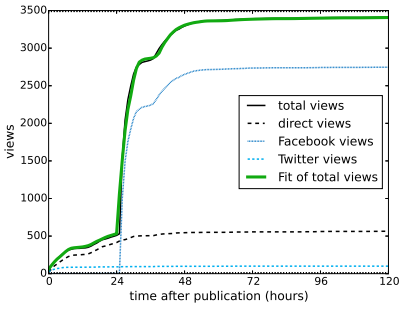
(a) Log-normal fit of total views, Eq. 1. Total views RRSE=0.342.



(b) Log-normal fit per component, Eq. 3. Total views RRSE=0.077.



(c) LinExp fit per component, Eq. 6. Total views RRSE=0.067.



(d) Transformed-time fit per component. Total views RRSE=0.045.

Fig. 5: Example curve fit with different models, for the example in Fig. 4, with indication of the root relative squared error (RRSE) of the total views fit.

with μ and σ the parameters of the distribution. The log-normal fit of the view pattern of Figure 4 is shown in Figure 5a.

The authors of [10,18] used user comments as popularity metric. However, the number of view patterns may be more complex. As can be seen in Figure 4, the curve of the total number of views consists of multiple components which do not necessarily start at the publication time. As could be anticipated, it appears to be a good approximation to assume that the direct views start to arrive at the moment the article is published on the website ($t = 0$), the Facebook views at the moment the article is published on Facebook, and the Twitter views at the moment it is posted on Twitter. Since we can measure which views originate from Facebook, Twitter, or from elsewhere, we can explicitly model these different components. A better model for the total number of views therefore consists of the sum of the separate log-normal fits of the different components:

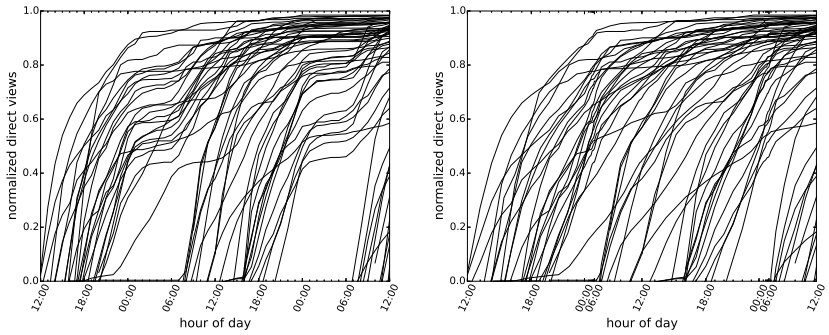
$$\begin{aligned} v_t^i \approx\ & s_d^i \cdot \text{clogn}_d(t; \mu_d^i, \sigma_d^i) \\ & + b_F \cdot s_F^i \cdot \text{clogn}_F(t - t_F; \mu_F^i, \sigma_F^i) \\ & + b_T \cdot s_T^i \cdot \text{clogn}_T(t - t_T; \mu_T^i, \sigma_T^i) \end{aligned} \tag{3}$$

with $\text{clogn}_d(\cdot)$, $\text{clogn}_F(\cdot)$ and $\text{clogn}_T(\cdot)$ the log-normal distribution associated with respectively the direct views, Facebook views, and Twitter views as defined in Equation 1. Parameter b_F (resp. b_T) is a known binary parameter that indicates whether the article is published on Facebook (resp. Twitter). Parameter t_F is the number of time units after the original publication ($t = 0$) that the article is posted on Facebook, and similarly for t_T on Twitter. Strictly speaking, $\text{clogn}(t; \mu, \sigma)$ is not defined for $t < 0$, but in Equation 3 we simple assume the contributions from the Facebook and Twitter components to be zero before their respective publication moments. The fit of the example article of Figure 4 according to this strategy can been seen in Figure 5b.

4.2 Linear-Exponential Popularity Model (LinExp)

In this section, we investigate alternative models, accurately capturing the observed behavior, preferably having parameters that are intuitively interpretable. When inspecting the data, measured by the hour, we rarely observed the typical log-normal behavior of an initial slow uptake, which increases and then again slows down towards the asymptotic value. We noticed that most often there simply is an initial uptake speed, that immediately starts to relax in a gradual way. Also, sometimes we noticed a small and constant uptake, independent from the large initial uptake directly after publication. A simple model for the uptake speed ν (or the number of views per time unit) corresponding to these observations and starting at time $t = 0$ is

$$\nu(t; c_1, c_2, T) = \frac{c_1}{T} e^{-\frac{t}{T}} + c_2 \tag{4}$$



(a) Uniform hours (b) Normalized hours

Fig. 6: Number of direct views in function of time for articles published between June 10, 2015 12:00 and June 12, 2015 12:00.

The first term of the right-hand side represents an exponential relaxation that reflects the gradual decrease of the uptake speed. The second term is the small constant uptake that sometimes becomes visible. It can be explained intuitively by assuming a small constant chance that a random user clicks the considered article, e.g. when browsing the news site, and which is independent of the article's publication time.

By integrating Equation 4 up to the current time t, the cumulative behavior becomes

$$V(t; c_1, c_2, T) = c_1(1 - e^{-\frac{t}{T}}) + c_2 t. \tag{5}$$

The total number of views v_t^i for article i at time t can thus be modeled by adding different components of this form, for direct views, Facebook views, and Twitter views.

$$\begin{aligned} v_t^i \approx\ & V_d(t; c_{1,d}^i, c_{2,d}^i, T_d^i) \\ & + b_F \cdot V_F(t - t_F; c_{1,F}^i, c_{2,F}^i, T_F^i) \\ & + b_T \cdot V_T(t - t_T; c_{1,T}^i, c_{2,T}^i, T_T^i) \end{aligned} \tag{6}$$

We will call this the *Linear-Exponential model*, abbreviated as the *LinExp model*. The fit of the example article of Figure 4 according to this proposed popularity model can be seen in Figure 5c.

4.3 Time Transformation

As can be seen in Figure 4, the number of visits retrieved between 7 and 14 hours and between 31 and 38 hours after publication is almost zero. This corresponds more or less to the period between 1 am and 8 am. As most people in the target audience sleep during that period, the articles do not retrieve a lot of additional visits and the view pattern also 'sleeps'. This is reflected by the

average number of views per hour of the day or night as shown in Figure 3 for direct views, Facebook views, and Twitter views. We would like to integrate this behavior into the model, without adding more degrees of freedom than necessary. In order to give a better qualitative idea of the problem, in Figure 6a we show all direct views for two randomly chosen consecutive days (June 10-12, 2015), as a function of time. The x-axis denotes the time starting on June 10, 2015 at noon, up to 2 days later. Along the y-axis, the direct views are shown, normalized for convenience by their stabilized value after 5 days. We clearly see that the night has a similar effect on most articles. This effect is stronger with later publication times. Also, during the second night that articles have been online, this effect is less pronounced but still present.

There are several ways to model this effect. Directly replacing the functions from Equation 5 by a more complex mathematical expression that depends on the publication time and models the observed behavior, would come with additional parameters and lead to a more complex model. This can be avoided by noticing that the day/night effect seems to be article-independent. We therefore propose the following heuristic: we replace each uniform time interval by the corresponding normalized value of the average number of reads during that hour, i.e. the values shown in Figure 3 for the respective components. As a result, nightly hours have a shorter normalized duration or effectively go faster, whereas during the day the effective time goes slower than on average. By applying this time transformation, the average number of reads per unit of normalized time would become uniform throughout the day. If indeed this time effect is completely article independent, we can expect that the day/night effect in the individual article view patterns disappears as well. Figure 6b shows the same view patterns as Figure 6a, but with the transformed time axis, and we can conclude qualitatively that the day/night effect is no longer clearly visible. Note that the time transformation needs to be applied for each of the components (direct, Facebook, Twitter) separately, as they are subject to a different reading behavior as already shown in Figure 3.

The proposed time transformation seems to be a suitable heuristic, and has an important advantage: we need to calculate the transformation only once per component type (direct views, Facebook, or Twitter), after which we can apply the original model of Equation 5 on the transformed time axis, without adding any model parameters. Even more, this transformation could be adapted to the day of week or weekend, or to the seasons, just by suitably averaging the number of views per hour. While evaluating the model, the inverse transformation needs to be made. For example, the predicted popularity at transformed time $\tilde{t}$ corresponds to the predicted popularity at the actual time t, in which $\tilde{t}$ was obtained by transforming t as described above. The transformed-time fit of the example article of Figure 4 can be seen in Figure 5d.

4.4 Parameter Estimation

For the log-normal baseline, the parameters s, μ, and σ are estimated using maximum likelihood estimation (MLE), as described in [7].

For the LinExp popularity model of Section 4.2, the parameters are also estimated with MLE. More in particular, with $\boldsymbol{c} := [c_1, c_2]^\top$ and $\boldsymbol{\phi}(t) := [(1 - e^{-\frac{t}{T}}), t]^\top$, we can write $V(t; \boldsymbol{c}, T) = \boldsymbol{c}^\top \boldsymbol{\phi}(t)$. Note that in line with Section 4.3, t denotes the transformed time with respect to the start of the considered component.

Minimizing the sum of squared errors, or equivalently, maximizing the likelihood under the assumption of additive Gaussian noise, leads to the following estimate $\hat{\boldsymbol{c}}$ for the coefficients:

$$\hat{\boldsymbol{c}} = \Big(\sum_t \boldsymbol{\phi}_t \boldsymbol{\phi}_t^\top \Big)^{-1} \Big(\sum_t v_t \boldsymbol{\phi}_t \Big) \tag{7}$$

in which v_t denotes the observed popularity value at time t, and we shortly write $\boldsymbol{\phi}_t := \boldsymbol{\phi}(t)$. A detailed treatment of this linear regression problem is given in [3].

The time constant T is also an unknown parameter. It can be determined by applying the expectation maximization (EM) algorithm, in which the expectation step, given by Equation 7, is followed iteratively by the maximization step

$$T = \text{argmax}_T \Big(- \sum_t (v_t - \hat{\boldsymbol{c}}^\top \boldsymbol{\phi}_t)^2 \Big). \tag{8}$$

4.5 Evaluation

To evaluate the quality of the curve fitting for article i, we use the root relative squared error (RRSE):

$$RRSE_i = \sqrt{\frac{\sum_t (\hat{v}_t^i - v_t^i)^2}{\sum_t (\overline{v}^i - v_t^i)^2}} \tag{9}$$

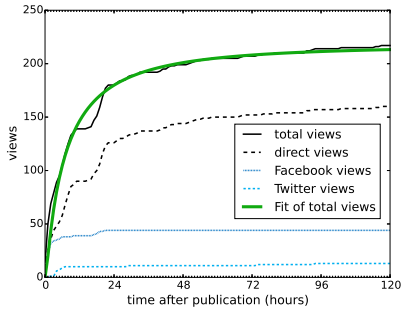
with v_t^i the observed number of total views at time t of article i, and $\hat{v}_t^i$ the value approximated by the model. We denote the average of the v_t^i observations for the considered article as $\overline{v}^i$. For our experiments we have an hourly observation of the total views, starting from the moment of publication, up to 5 days (120 hours) later, or $t = 0, \ldots, 120$. The RRSE is calculated over the articles in the training set K and its mean value (written MRRSE) is used to evaluate the different models:

$$MRRSE = \frac{1}{|K|} \sum_{i=1}^{|K|} RRSE_i \tag{10}$$

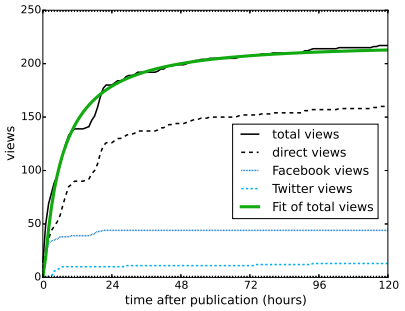
The MRRSE values for the considered temporal popularity models are shown in Table 1. We notice that the curve fitting performance is improved by

model	MRRSE
log-normal fit, Eq. 1	0.233
log-normal fit per component, Eq. 3	0.211
linexp fit per component, Eq. 6	0.151
transformed-time fit per component	0.124

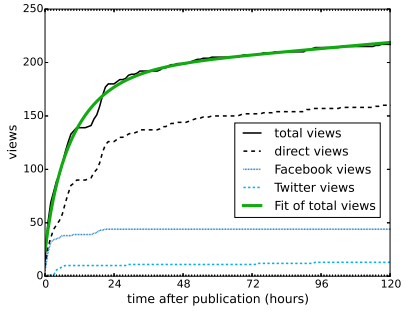
Table 1: MRRSE of the temporal popularity models. All differences between models are significant ($p < 0.001$).



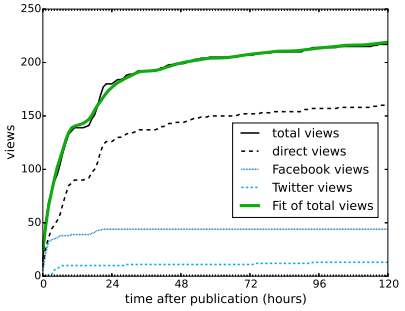
(a) Log-normal fit of total views, Eq. 1. Total views RRSE=0.160.



(b) Log-normal fit per component, Eq. 3. Total views RRSE=0.156.



(c) LinExp fit per component, Eq. 6. Total views RRSE=0.101.



(d) Transformed-time fit per component. Total views RRSE=0.065.

Fig. 7: Another example curve fit with different models, with indication of the root relative squared error (RRSE) of the total views fit.

explicitly modeling the different components (Equation 3) instead of directly fitting the total number of views (Equation 1). The proposed LinExp model as defined in Equation 6 leads to further improvements. We can hence conclude that the functions in Equation 5 better describe the separate components than the log-normal model. The time transformation leads to a further decrease in the average error. All mentioned improvements appeared significant up to the level $p = 0.001$, using a one-sided bootstrap significance test [16].

Figure 5 clearly shows the added value of modeling the direct views, Twitter views and Facebook views separately instead of directly fitting the total number of views (Figure 5a vs. 5b). The error further decreases when using our proposed model (Figure 5c). The error reduction is small because the popularity totally stagnates after two days, leading to a near-zero coefficient c_2 in the linear component $c_2 t$ (Equation 5). Finally, the use of the time transformation leads to further improvements (Figure 5d). Figure 7 provides another visual illustration. It shows the various model fits for a less popular article as compared to Figure 4, with indication of the RRSE. In this example, the article was published on Facebook and Twitter together with its initial online publication, such that modeling all three components separately does not contribute much with respect to directly modeling the total number of views (Figure 7a vs. 7b). However, we notice that while the Twitter and Facebook views become stable after one day, there is a noticeable linear increase of the direct views, which continues during the subsequent days. The linear component $c_2 t$ in Equation 5, which we introduced as a constant (i.e., publication time independent) rate of users browsing to the article, accurately models that behavior. This leads to a lower error, as seen from Figure 7b with the log-normal model vs. Figure 7c with the proposed LinExp popularity model. The time-transformed model as described in Section 4.3 further reduces the error, confirming the added value of taking into account the variation in popularity throughout the day.

5 Popularity Prediction

In this section, we show how the total popularity of news articles can be predicted. An important insight from the previous section, is the need to model the different components separately, which we follow for the prediction task as well. In particular, we train three different regressors to respectively predict the direct views, Facebook views, and Twitter views. The articles' final popularity is measured for each of these components, at target time τ after publication on the website, on Facebook, or on Twitter, respectively. The objective is thus to predict for each article at a particular *reference time* r its final popularity at a future point in time, which we will refer to as the *target time* τ (with $0 \leq r \leq \tau$). Note that we use the general term 'views' to indicate any of the previously introduced popularity metrics. It may refer to direct views, Facebook views, or Twitter views, but also to other popularity metrics like Facebook shares, which we do not explicitly treat in this paper.

In Section 5.1, we give an extensive overview of existing approaches, which we implemented as baseline methods. Our proposed prediction methodology is described in Section 5.2. Finally, we evaluate the baselines and the proposed methodology in Section 5.3.

5.1 Baselines

This section provides an overview of baseline methods based on linear regression models. Similar to the approaches described in [17,5,9] we log-transform the popularity values as there is a better correlation between the log-transformed popularities at reference time r and target time τ than between the untransformed popularities. The regression model thus takes the form

$$log(1 + \hat{\boldsymbol{v}}_\tau) = log(1 + \boldsymbol{X}_r)\boldsymbol{\beta} \tag{11}$$

in which we assume a component-wise logarithmic transformation of the vector $\hat{\boldsymbol{v}}_\tau$ of predicted views at target time τ for the considered articles, and of the article features in matrix $\boldsymbol{X}_r$ constructed at reference time r. Each row $\boldsymbol{x}_r^i$ in matrix $\boldsymbol{X}_r$ corresponds to the vector of feature values of article i and

$$log(1 + \hat{v}_\tau^i) = log(1 + \boldsymbol{x}_r^i)\boldsymbol{\beta} \tag{12}$$

with vector $\hat{v}_\tau^i$ the predicted number of views at target time τ of article i. The parameters $\boldsymbol{\beta}$ are estimated using ordinary least squares on the training set K:

$$\boldsymbol{\beta} = \text{argmin}_{\boldsymbol{\beta}} \, ||log(1 + \boldsymbol{v}_\tau) - log(1 + \boldsymbol{X}_r)\boldsymbol{\beta}||_2^2 \tag{13}$$

with $\boldsymbol{v}_\tau$ the observed views at target time τ and $||\cdot||_2^2$ the squared L_2-norm. The goal of this objective function is to minimize the sum of the squared errors on the log-transformed data. We consider several baseline methods that are based on this linear regression model and describe them in the following paragraphs.

Szabo and Huberman model (SH model) The simplest model, introduced by Szabo and Huberman [17], only considers the number of visits measured at reference time r,

$$\boldsymbol{x}_r^i = [v_r^i] \tag{14}$$

with v_r^i the number of visits for article i at reference time r.

Multivariate Linear model (ML model) Pinto et al. [15] extended the SH model by considering the whole history of the number of visits, or

$$\boldsymbol{x}_r^i = [v_1^i, v_2^i \ldots v_r^i] \tag{15}$$

with v_t^i the number of visits for article i, observed t time units after publication.

Radial Basis Functions model (RBF model) The authors of [15] extended their ML model by indirectly incorporating the different possible popularity patterns. In particular, they proposed to take into account the similarity in terms of early popularity between the article and n randomly selected examples from the training set, called subset S. Gaussian Radial Basis Functions (RBF) were used for measuring the similarity between articles i and $a \in S$:

$$RBF_a(i) = e^{-\frac{||\boldsymbol{x}_r^i - \boldsymbol{x}_r^a||_2^2}{2 \cdot \sigma^2}} \tag{16}$$

with $\boldsymbol{x}_r^i$ the ML feature vector as defined in Equation 15, and parameter $\sigma > 0$. Equation 12 can then be rewritten as

$$log(1 + \hat{v}_\tau^i) = log(1 + \boldsymbol{x}_r^i)\boldsymbol{\beta} + \sum_{a \in S} w_a \cdot RBF_a(i) \tag{17}$$

with $\boldsymbol{x}_r^i$ as defined in Equation 15. The ML model and RBF model were originally optimized and evaluated using the mean relative squared error, instead of the sum of the squared logarithmic errors as used in this paper.

First-Order Social Media model (FOSM model) The fourth baseline is based on the model introduced by Castillo et al. [5]. The authors proposed a multiple linear regression model which uses the number of visits at reference time, together with metrics retrieved from social media. The first-order model is given by Equation 12, whereby

$$\boldsymbol{x}_r^i = [v_r^i,\ v_{r,F}^i,\ v_{r,T}^i,\ m_{r,F}^i,\ m_{r,T}^i] \tag{18}$$

with $v_{r,F}^i$ the number of views originating from Facebook article i received at reference time r, $v_{r,T}^i$ the number of article i views originating from Twitter at reference time r, and $m_{r,F}^i$ and $m_{r,T}^i$ the number of respectively Facebook shares and tweets related to article i at time r. To be precise, the original model described in [5] does not consider Facebook or Twitter views. Instead, their model includes the number of visits from link referrals, direct traffic from e-mail, and some Twitter statistics such as the entropy of the tweets and number of unique tweets. However, since these features are not available in our dataset, we replace them by the features listed in Equation 18.

Second-Order Social Media model (SOSM model) The paper of [5] also describes a second-order variant of their first-order social media model. In addition to the first-order features described in Equation 18, they also include the second-order interactions of these features. These features are included to model the interdependency of the variables.

Mixed model Our last baselines are based on the models proposed by Figueiredo et al. [9], winner of the ECML/PKDD 2014 Predictive Analytics Challenge. Their models are based on the ideas of the RBF and SOSM model. The first model considers both the whole history of the popularity metric values and the metrics retrieved from social media. The vector representing the whole history of the number of visits is defined as

$$\boldsymbol{v}_r^i = [v_1^i, v_2^i \dots v_r^i] \tag{19}$$

Similarly, the history of Facebook views, Twitter views, Facebook shares and Twitter posts are represented by vectors $\boldsymbol{v}_{r,F}^i$, $\boldsymbol{v}_{r,T}^i$, $\boldsymbol{m}_{r,F}^i$, $\boldsymbol{m}_{r,T}^i$, respectively. The binary vector $\boldsymbol{d}^i$ is a one-hot feature vector to represent the week day, and similarly, $\boldsymbol{h}^i$ represents the publication hour of article i. The feature vector $\boldsymbol{x}_r^i$ representing article i in Equation 12 is then constructed by concatenating $\boldsymbol{d}^i$, $\boldsymbol{h}^i$, $\boldsymbol{v}_r^i$, $\boldsymbol{v}_{r,F}^i$, $\boldsymbol{v}_{r,T}^i$, $\boldsymbol{m}_{r,F}^i$, and $\boldsymbol{m}_{r,T}^i$, and all of their pairwise interactions, represented by the elementwise products. Again, the original model does not

consider the Facebook views and Twitter views. It considers the time series of the average time each user spends on the page, which we have to leave out as it is unavailable in our dataset.

Mixed-Trend model Similar to the RBF model, the authors of [9] extended their Mixed model by indirectly incorporating the different possible popularity patterns. In particular, they proposed to take into account the similarity in terms of early popularity between the article and k cluster centers. The early popularity of article i can be represented by the vector

$$\boldsymbol{p}_r^i = [log(1+\delta_1^i), log(1+\delta_2^i) \dots log(1+\delta_r^i)]^\top \tag{20}$$

with δ_t^i the number of visits gained in time interval t, i.e. $\delta_t^i = v_t^i - v_{t-1}^i$. The similarity between two articles a and i is then quantified using the euclidean distance:

$$dist_a(i) = ||\overline{\boldsymbol{p}}_r^i - \overline{\boldsymbol{p}}_r^a||_2 \tag{21}$$

with $\overline{\boldsymbol{p}}_r^i$ the z-normalized vector of $\boldsymbol{p}_r^i$. This distance function is used to determine k cluster centers (set C) using the k-means algorithm on the training set. Equation 12 can then be modified to

$$log(1+\hat{v}_\tau^i) = log(1+\boldsymbol{x}_r^i)\boldsymbol{\beta} + \sum_{a \in C} w_a \cdot dist_a(i). \tag{22}$$

5.2 Proposed Methodology and Features

We will evaluate the popularity predictions based on five different models, besides the baselines described above. These five proposed models differ in terms of the considered regression algorithm, and the different types of included features. The features, discussed below, are listed in Table 2. For the models, we distinguish between a linear regression model (similar to the baselines) and the gradient tree boosting (GTB) algorithm. The latter is often used in winning methodologies for Kaggle competitions[5], because it can handle non-linearities in the data and interactions between the features. We use the regression implementations available in the Python scikit-learn package[6]. The models can be characterized as follows

- **LM history**: linear regression model, based on the 'history' features described in Table 2,
- **LM history+curve**: linear regression model, based on the 'history' and 'curve' features described in Table 2,
- **RIDGE history+curve**: linear regression model with L2 regularization (ridge regression, $\alpha = 1.0$), based on the 'history' and 'curve' features,
- **GTB history+curve**: GTB regression, with the 'history' and 'curve' features,
- **GTB all**: GTB regression, with all described features.

domain	*name*	*description*
History	views	number of views for article a at reference time r, i.e. v^a_r
	viewsHistory	$\forall h \in [1,5]$ number of views for article a received between reference time r and h hours earlier, i.e. $v^a_r - v^a_{r-h}$
	directViews	number of direct views for article a at reference time r, i.e. $v^a_{r,d}$
	directViewsHistory	$\forall h \in [1,5]$ number of direct views for article a received between reference time r and h hours earlier, i.e. $v^a_{r,d} - v^a_{r-h,d}$
	facebookViews	number of Facebook views for article a at reference time r, i.e. $v^a_{r,F}$
	facebookViewsHistory	$\forall h \in [1,5]$ number of Facebook views for article a received between reference time r and h hours earlier, i.e. $v^a_{r,F} - v^a_{r-h,F}$
	twitterViews	number of Twitter views for article i at reference time r, i.e. $v^a_{r,T}$
	twitterViewsHistory	$\forall h \in [1,5]$ number of Twitter views for article a received between reference time r and h hours earlier, i.e. $v^a_{r,T} - v^a_{r-h,T}$
	facebookShares	number of Facebook shares for article i at reference time r, i.e. $m^a_{r,F}$
	facebookSharesHistory	$\forall h \in [1,5]$ number of Facebook shares for article a received between reference time r and h hours earlier, i.e. $m^a_{r,F} - m^a_{r-h,F}$
Curve Features	relaxation amplitude	parameter c_1 in Equation 4, after curve fitting of Equation 4 on the view pattern between hour 0 and reference time r
	linear amplitude	parameter c_2 in Equation 4, after curve fitting of Equation 4 on the view pattern between hour 0 and reference time r
	relaxation constant	parameter T in Equation 4, after curve fitting of Equation 4 on the view pattern between hour 0 and reference time r
Author	authorAverage	average popularity of articles in the training set published by the author of a
	authorStd	standard deviation of the popularity of the articles in the training set published by the author of a
	authorCount	number articles in the training set published by the author of a
	authorBinary	binary vector representing the author of article a
Category	categoryAverage	average popularity of articles in the training set with the same category as a
	categoryStd	standard deviation of the popularity of the articles in the training set with the same category as a
	categoryCount	number of articles in the training set with the same category as a
	categoryBinary	binary vector representing the category of article a
Publication Time and Date	hourOfDayAverage	average popularity of the articles in the training set published during the same hour of day as a
	hourOfDayStd	standard deviation of the popularity of the articles in the training set published during the same hour of day as a
	hourOfDayCount	number of the articles in the training set published during the same hour of day as a
	hourOfDayBinary	binary feature indicating the hour of day the article is published
	dayOfWeekAverage	average popularity of the articles in the training set published during the same day of week as a
	dayOfWeekStd	standard deviation of the popularity of the articles in the training set published during the same day of week as a
	dayOfWeekCount	number of the articles in the training set published during the same day of week as a
	dayOfWeekBinary	binary feature indicating the day of week the article is published
Title	numberInTitle	binary feature indicating if the title of a contains a number
	entityInTitle	binary vector indicating the type of the named entity in the title of a (if present)
Source Article	hasSourceArticle	binary feature indicating if a has a source article
	sourceArticleShares	number of Facebook shares the source article of a received at publication time of a
Virality	willGoViral	binary feature indicating if objective of the article is to go viral on Facebook
Target Audience	genderAverage	average popularity of articles in the training set with the same target gender as a
	genderStd	standard deviation of the popularity of the articles in the training set with the same target gender as a
	genderCount	number of articles in the training set with the same target gender as a
	genderBinary	binary vector representing the target gender of article a
	ageAverage	average popularity of articles in the training set with the same target age as a
	ageStd	standard deviation of the popularity of the articles in the training set with the same target age as a
	ageCount	number of articles in the training set with the same target age as a
	ageBinary	binary vector representing the target age of article a
Emotion	emotionAverage	average popularity of articles in the training set with the same target share emotion as a
	emotionStd	standard deviation of the popularity of the articles in the training set with the same target share emotion as a
	emotionCount	number of articles in the training set with the same target share emotion as a
	emotionBinary	binary vector representing the target share emotion of article a

Table 2: Features considered in this paper for training regressors to predict the popularity of article a.

We provide a short description for each of the ten groups of features listed in Table 2:

History These features capture the popularity pattern of the article. Similar to [5], we use the popularity expressed by other metrics (e.g., Facebook views and Twitter views) to better predict the considered popularity metric (e.g., direct views). In particular, the total views, direct views, Facebook views, Twitter views, and Facebook shares are considered. Similar to the prediction method described in Section 5.1, all popularity values are log-transformed.

Curve Features We incorporate our knowledge of the distributions which underlie the article popularity pattern, as discussed in Section 4. In particular, we estimate the parameters of Equation 5 for the known historical popularity values as described in Section 4.4. These parameters are then used as features for the regression model.

Author We include the average popularity of the articles in the training set published by the same author of the considered article, its standard deviation, and also the number of training articles by that author. In addition, one-hot feature vectors are used to indicate the specific author.

Category The journalists of newsmonkey manually labeled all articles with one or more category from a set of 16 categories (society, politics, tv, music, life and style, cyberspace, tech and gadgets, planet, travel, movies, starts, economy, body and soul, science, pets, and games). Similar to the author features, we represent the categories of the article by an average popularity, standard deviation, number of articles, and a binary vector to indicate the categories.

Publication Time and Date Similar to [9], we include the publication hour and week day as features.

Title We determine whether the title of the considered article contains a number (binary feature). Articles containing a number in their title are mostly articles containing lists, and their title often starts with a phrase like 'n reasons why...', with n a number. These 'list' articles are constructed with the main objective that they are very shareable on Facebook, which makes the described feature very informative. Additionally, we use named entity recognition [8] to extract the named entities and their type from the title. The possible entity types, for which binary features are introduced, are organization, location, person, or miscellaneous.

Source Article With a binary feature, we indicate whether or not the article refers to a source article, i.e., an article from another news website which is cited by the article (for example from Business Insider or Mashable). We also include a feature with the number of Facebook shares the source article already received at publication time of the considered article.

Virality The journalists of newsmonkey annotated some articles with labels reflecting their experience on which articles will go viral on Facebook and

[5] https://www.kaggle.com/
[6] http://scikit-learn.org

why. In particular, they manually labeled their articles as 'will go viral' if they estimated that the article would be very popular on Facebook.

Target Audience For the articles labeled as 'will go viral', the authors also indicated the target audience. The target audience is given by the target gender (female, male, or both) and target age range (18-24, 25-34, or 18-34 years old).

Emotion In addition to the target audience, the authors labeled 'will go viral' articles with an emotion label, which is included as a binary feature vector. The annotated emotion label of a particular article is not the direct emotion it is expected to provoke in the readers, as considered in previous research [1,4]. Instead, it is the emotion of why users are expected to *share* the article. The considered emotion labels are recognizability, identity, awe, humor, pride, malicious pleasure, altruism, taboo, outrage, nostalgia, and softening.

5.3 Evaluation

In this section, we evaluate the proposed prediction methodologies. The models are trained using training set K and evaluated on the articles in a separate test set U. The parameters of the models are optimized using 5-fold cross-validation on the training set. As evaluation metric indicating the performance of the predictions, we use the root mean squared log error (RMSLE):

$$RMSLE = \sqrt{\frac{1}{|U|}\sum_{i \in U}(log(\hat{v}_\tau^i + 1) - log(v_\tau^i + 1))^2} \tag{23}$$

with v_τ^i the observed number of views of article i at target time τ, and $\hat{v}_\tau^i$ the predicted number of views. In other words, the RMSLE indicates how well the popularity at target time τ is predicted for all articles in the test set. This evaluation metric is also used in the ECML/PKDD 2014 Predictive Analytics challenge[7]. To determine whether the difference in performance of two methods is statistical significant, we use the unpaired bootstrap hypothesis test [16]. We consider the predictions with a target time τ of 5 days (120 hours) after publication of the article, and reference time r one to 24 hours after publication, with time intervals of one hour. For each considered reference time and popularity metric (direct views, Facebook views and Twitter views), we train and evaluate a separate regressor. We focus on the first 24 hours after publication, because in order to adapt the publishing strategy it is most important to get good predictions at an early stage after publication on the website. For example, 90% of the Facebook publications of the articles in our dataset appear within 13 hours after they were published on the website. We first describe our evaluation on the predictions of the direct views, after which we discuss the Facebook and Twitter predictions.

[7] https://sites.google.com/site/predictivechallenge2014/

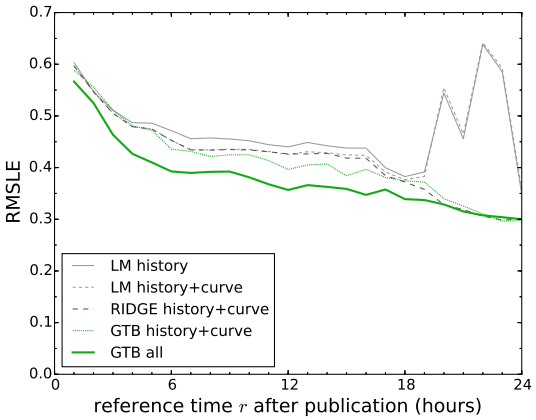


Fig. 8: Performance of the four different versions of our proposed methodology, considering direct views.

5.3.1 Direct views

We first consider the number of direct views five days after online publication as the popularity quantity to be predicted. The performance of the five methods we proposed in Section 5.2 is shown in Figure 8. The RMSLE is shown as a function of the reference time r. In other words, the RMSLE indicates for a particular reference time r how well the popularity at target time τ is predicted, given observations and features up to time r. The linear regression model trained on both the historical popularity and the curve features (LM history+curve) performs better than the linear model only trained on the historical popularity (LM history) before hour 20. However, the improvement is not statistically significant (bootstrap hypothesis test, $p > 0.2$). The error of the linear models (LM history and LM history+curve) decreases steadily up to 19 hours after publication, after which the error increases again and starts to fluctuate. This is mainly due to over-fitting of the linear model. When regularization is applied (RIDGE history+curve), we notice that the performance is similar for the first 19 hours after publication (bootstrap hypothesis test, $p > 0.2$). However, the error for the regularized model further decreases after hour 20, as over-fitting is avoided. The GTB regressors (GTB history+curve) are also robust to over-fitting, and lead to a slight improvement with respect to the ridge regressors (RIDGE history+curve) for reference hours 5 to 17 (bootstrap hypothesis test, $p > 0.2$). The last model, which applies GTB regression on all proposed features (GTB all), outperforms GTB history+curve for reference time r between hours 1 and 19. The improvement is statistically significant for $3 \leq r \leq 6$ and $10 \leq r \leq 14$ ($p < 0.05$). We conclude that adding content and meta-data related features on top of temporal features significantly improves the prediction effectiveness.

model	RMSLE
GTB all	0.381
GTB history+author	0.405
GTB history+target	0.409
GTB history+virality	0.410
GTB history+emotion	0.415
GTB history+category	0.438
GTB history+publication	0.442
GTB history+source	0.451
GTB history+title	0.453
GTB history	0.453

Table 3: Performance of the content and meta-data feature types at reference time 10, considering direct views.

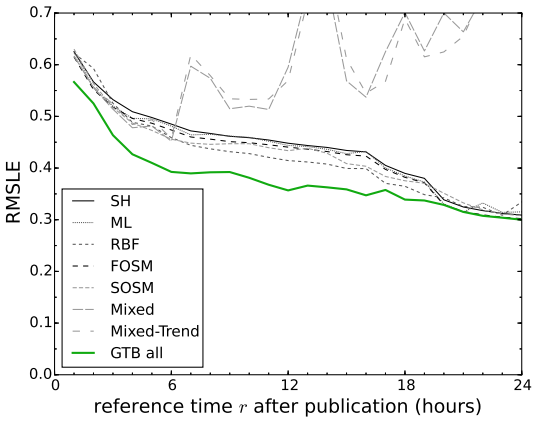


Fig. 9: Performance of the baselines and our proposed methodology, considering direct views.

To investigate the contribution of each content and meta-data related feature type described in Table 2, a GTB regressor is trained using the history features and the features of the considered type. The performances of these models at reference time 10 can be found in Table 3. We observe that the author features lead to the highest increase of performance (about 5% RMSLE reduction), closely followed by the manually annotated features (i.e. virality, target audience and emotion). The use of the publication or category features in addition to the history features also improves the performance (about 1% in RMSLE). On the other hand, the article source and title features hardly improve the GTB history model. Using all introduced content and meta-data features results in the best performance (decrease of about 7% in RMSLE).

We now compare the baselines introduced in Section 5.1 with our best model (GTB all), as shown in Figure 9. First of all, and most importantly, we see that our method outperforms all seven considered baselines significantly between one hour and 16 hours after publication ($p < 0.05$). Furthermore,

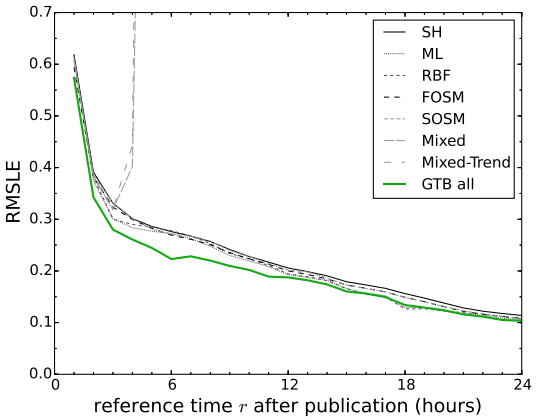


Fig. 10: Performance of the baselines and our proposed methodology, considering Facebook views.

between reference hour 3 and 16, our method improves on all baselines with more than 10% in RMSLE. As it is most important to get good predictions in an early stage after publication, our proposed methodology has a high added value compared to the baselines. For instance, the RMSLE for the method GTB all at reference hour 7 (0.387) is only achieved at reference hour 17 for the best baselines. In other words, the prediction performance of the baselines 17 hours after publication is already achieved by our method after only 7 hours. Starting from reference hour 20, all methods (except for Mixed and Mixed-Trend) have similar performance (bootstrap hypothesis test, $p > 0.2$). This is because the popularity of articles typically becomes stable after having been published that many hours. As a result, for $r \geq 20$, the added value of more complex regression algorithms and additional features on top of the historical popularities is no longer significant.

When we compare the baseline methods, we see that one of the most complex methods (Mixed) introduced by [9] performs on average as the best baseline model between hour one and six. This is in line with the observation made by [9], testing their model with a reference time of one hour after publishing and target time of 48 hours after publishing. However, starting from 7 hours after publication, the RMSLE for the methods Mixed and Mixed-Trend increases and starts to fluctuates. This is due to over-fitting of their linear regression model trained on a large set of features. This would be resolved by using regularization, but from the description in [9], it was unclear if and which sort of regularization was used. Between 7 hours and 19 hours after publication, the RBF model introduced by [15] has the best baseline performance. However, the improvement above the other baselines (except for Mixed and Mixed-Trend) is not statistical significant ($p > 0.2$).

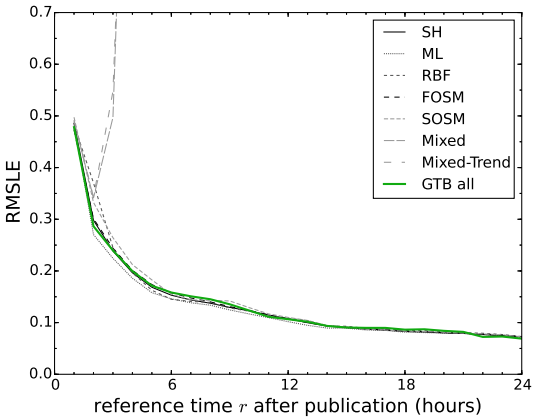


Fig. 11: Performance of the baselines and our proposed methodology, considering Twitter views.

5.3.2 Facebook views

We now consider the number of Facebook views five days after publishing the article on Facebook as the popularity to be predicted. We only consider the training articles which effectively got published on Facebook (1940 out of 2614 articles). The performance of the baselines and our proposed methodology is shown in Figure 10. Our model (GTB all) is the best performing model between hour 1 and 16 after publication on Facebook. The improvement with respect to the baselines is significant between reference hour 5 and 8 ($p < 0.05$). In particular, for $2 \leq r \leq 11$, the RMSLE decreases with more than 8% when using our method instead of the baseline. As an example, the RMSLE for our method at reference hour 6 (0.223) is only achieved after 10 hours for the baselines. Starting from 12 hours after publication on Facebook, all considered methods (except Mixed and Mixed-Trend) display a similar performance ($p > 0.2$). We notice that the Mixed and Mixed-Trend baselines start to over-fit at hour 4, which could again be avoided using regularization. The other baselines show similar prediction performances ($p > 0.2$).

5.3.3 Twitter views

We now evaluate the performance of the models in their ability to predict the number of Twitter views. We only consider the 1724 training articles effectively published on Twitter, and predict the number of Twitter views five days after publishing the article on Twitter. The performance of all models is shown in Figure 11. We see that their performance (except for Mixed en Mixed-Trend) is very similar ($p > 0.2$). The main reason is that the average number of Twitter views received for articles published on Twitter is very low (around 80 views), and becomes constant soon after publication. It is thus not

obvious to improve the prediction performance in terms of RMSLE by using more advanced features and algorithms. Note that this behavior is not representative for any news data, but in Belgium Twitter is not as widely adopted as in other countries [6].

We can conclude that our method outperforms all baselines during the first hours after publication, when direct views or Facebook views are considered. As mentioned before, the prediction of the direct views and Facebook views at these early hours is most relevant to optimize the publishing strategy of online articles. The significant improvement with respect to previously published methods therefore has a high added value for popularity predictions in practice.

6 Conclusion

In order to improve the online publishing strategy of news content, methods to model and predict the popularity of online news articles are required, which forms the main topic of this paper. We first identified the distributions which underlie the view patterns of online news articles. These consist of several distinct components. The first component becomes visible as soon as the article is published on the news publisher's website. The corresponding views are referred to as the direct views and originate from e.g. search and browsing. When the article is published on social media, clear additional components in the view patterns start to appear. In this paper, we focused on the views originating from Facebook and Twitter. We then introduced a model that allows to accurately model these view pattern components. This model captures the popularity behavior, is simple to fit to observed views, and has parameters that are intuitively interpretable. Based on real-world data from a young Belgian publisher that actively targets the distribution of its content over social media, we demonstrated that this model outperforms previously proposed log-normal fits. In addition, we took the influence of the day versus night on the view patterns into account to further increase the accuracy, without leading to a more complex model. By transforming each actual time interval into an equivalent time interval with an effective duration equal to the normalized total number of views for that time interval, the influence of the average hourly variations in number of views is largely canceled out, which allowed for a better fit of the view pattern to smooth base functions.

As a second contribution, we proposed a methodology to predict the final popularity for each component adding to the total popularity of an article (i.e. direct views, Facebook views, and Twitter views). We focused on articles which are at most one day old, as the predictions of those articles are most useful. Our primary model was based on existing methods, with linear regression algorithms and features based on the historical popularity of the articles. We then proposed models with improved prediction effectiveness, based on the following three ideas. First, we used the parameters of our proposed

popularity model as additional input features, leading to a small overall improvement during the first hours after publication, although not significant. Second, we showed that the use of a more advanced regression technique, i.e., Gradient Tree Boosting, gives more accurate predictions. Third, the prediction performance was significantly improved by considering features based on the content and meta-data of the articles. Our best model outperformed all discussed baselines during the first hours after publication, at least for the direct views or Facebook views. In particular, we considered seven baseline methods, with features mainly capturing the historical popularity of the considered articles. The performance of the Twitter view predictions appeared similar to the baseline predictions. However, the average number of Twitter views per article appeared very low in our experimental setup, which prevented further improvements by using a more complex method. As the prediction of the direct views and Facebook views at the early hours are most relevant in order to optimize the publishing strategy of online articles, the significant improvement with respect to previously published methods has a high added value for popularity predictions in practice.

In this paper, we proposed a method which predicts the final popularity of news articles. However, it is often useful to also predict the popularity dynamics between the current time stamp and its final popularity. This information can for instance be used to better decide which would be the most suited moment to publish and promote an article over social media. Therefore, in future work, we will propose a methodology that predicts the entire future popularity pattern. This will be achieved by combining the knowledge of the proposed popularity model, including the day-night behavior of the different components, with the prediction method with content and meta-data related features in a single time series prediction setup.

We focused our experimental research on the particular case of a Belgian news company, training and evaluating our models on the habits and response of the local audience. Although our trained models are as such not readily applicable to other cases, we believe our strategy is general enough, to be applied to different scenarios as well. We showed that a single mathematical model allows modeling view patterns with very distinct origins (e.g., direct views vs. Facebook views). Combining the resulting contributions from different homogeneous components (direct views, or distinct social media channels), allows modeling the more complex temporal patterns of total views. In other contexts, first an analysis of the main components (e.g., also characterized by their origin) needs to be carried out, after which the recorded views can be used, similarly to our work, for training predictors that take into account the different components. Important aspects such as information of events that initiate new components (e.g., social pushes) need to be monitored as well, obviously. Furthermore, we have discussed some highly informative qualitative (meta-data) features in our experiments (category, emotion, editorial intended popularity levels...), which could provide ideas to companies intending to design a popularity prediction system. Of course, different features would be available in other contexts. This makes it difficult to rely on the absolute ef-

fectiveness scores mentioned in our work, but we propose that the qualitative approach and mathematical description hold nevertheless.

Acknowledgements We thank Ke Zhou for useful suggestions on drafts of the manuscript. Steven Van Canneyt is funded by a Ph.D. grant of the Agency for Innovation by Science and Technology in Flanders (IWT). Part of the presented research was performed within the MIX-ICON project PROVIDENCE, facilitated by iMinds-Media and funded by the IWT.

References

1. Arapakis, I., Cambazoglu, B.B., Lalmas, M.: On the feasibility of predicting news popularity at cold start. In: Proceedings of the 6th Internation Conference on Social Informatics, pp. 290–299 (2014)
2. Bandari, R., Asur, S., Huberman, B.: The pulse of news in social media: Forecasting popularity. In: Proceedings of the 6th International Conference on Weblogs and Social Media, pp. 26–33 (2012)
3. Barber, B.: Bayesian reasoning and machine learning. Cambridge University Press (2012)
4. Berger, J., Milkman, K.L.: What makes online content viral? Journal of Marketing Research **49**(2), 192–205 (2012)
5. Castillo, C., El-Haddad, M., Stempeck, M., Jazeera, A., Pfeffer, J.: Characterizing the life cycle of online news stories using social media reactions. In: Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work and Social Computing, pp. 211–213 (2014)
6. Cheng, A., Evans, M., Singh, H.: Inside Twitter: An in-depth look inside the Twitter world. Tech. rep. (2014)
7. DeGroot, M.H., Schervish, M.J.: Probability and statistics (2010)
8. Deleu, J., Moor, A.D.: Named entity recognition on Flemish audio-visual and newspaper archives. In: Proceedings of the 12th Dutch-Belgian Information Retrieval Workshop, pp. 38–41 (2012)
9. Figueiredo, F., Gonçalves, M., Almeida, J.M.: Improving the effectiveness of content popularity prediction methods using time series trends. In: ECML/PKDD Discovery Challenge on Predictive Analytics, pp. 1–6 (2014)
10. Kaltenbrunner, A., Gómez, V., López, V.: Description and prediction of Slashdot activity. In: Proceedings of the Latin American Web Conference, pp. 57–66 (2007)
11. Kim, S.D., Kim, S.H., Cho, H.G.: Predicting the virtual temperature of web-blog articles as a measurement tool for online popularity. In: Proceedings of the 11th International Conference on Computer and Information Technology, pp. 449–454 (2011)
12. Kong, S.: Predicting future retweet counts in a microblog. Journal of Computational Information Systems **4**(10), 1393–1404 (2014)
13. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA (2008)
14. Oghina, A., Breuss, M., Tsagkias, M., De Rijke, M.: Predicting IMDB movie ratings using social media. In: Proceedings of the 34th European Conference on Advances in Information Retrieval, pp. 503–507 (2012)
15. Pinto, H., Almeida, J.M., Gonçalves, M.A.: Using early view patterns to predict the popularity of YouTube videos. In: Proceedings of the 6h ACM International Conference on Web Search and Data Mining, pp. 365–374 (2013)
16. Sakai, T.: Evaluating evaluation metrics based on the bootstrap. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 525–532 (2006)
17. Szabo, G., Huberman, B.: Predicting the popularity of online content. Communications of the ACM **53**, 80–88 (2008)
18. Tatar, A., Antoniadis, P., de Amorim, M.D., Fdida, S.: From popularity prediction to ranking online news. Social Network Analysis and Mining **4**(1), 174–186 (2014)

19. Tsagkias, M., Weerkamp, W., De Rijke, M.: Predicting the volume of comments on online news stories. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, pp. 1765–1768 (2009)
20. Tsagkias, M., Weerkamp, W., De Rijke, M.: News comments: Exploring, modeling, and online prediction. In: Proceedings of the 32nd European Conference on Advances in Information Retrieval, pp. 191–203 (2010)
21. Yan, C., Zhang, Y., Dai, F., Wang, X., Li, L., Dai, Q.: Parallel deblocking filter for HEVC on many-core processor. Electronics Letters **50**(5), 367–368 (2014)
22. Yan, C., Zhang, Y., Dai, F., Zhang, J., Li, L., Dai, Q.: Efficient parallel HEVC intra-prediction on many-core processor. Electronics Letters **50**(11), 805–806 (2014)
23. Yan, C., Zhang, Y., Xu, J., Dai, F., Li, L., Dai, Q., Wu, F.: A highly parallel framework for HEVC coding unit partitioning tree decision on many-core processors. IEEE Signal Processing Letters **21**(5), 573–576 (2014)
24. Yan, C., Zhang, Y., Xu, J., Dai, F., Zhang, J., Dai, Q., Wu, F.: Efficient parallel framework for HEVC motion estimation on many-core processors. IEEE Transactions on Circuits and Systems for Video Technology **24**(12), 2077–2089 (2014)