Is Your Story Going to Spread Like a Virus? Machine Learning Methods for News Popularity Prediction

Xuandong Lei
xuandong@stanford.edu

Xiaoti Hu
xiaotihu@stanford.edu

Hongsheng Fang
hsfang@stanford.edu

Abstract—Online website has been the majority source for news to spread. An interesting news is going to be shared thousands of times through the Internet. In this project, we tried different methods to predict the popularity before its publication. Using over 39000 sample articles from Mashable.com, we tried to address the problem both as a numerical and a multinomial classification problem. Specifically, we applied linear regression, polynomial regression, GAM with smoothing splines and Lasso to predict the exact shares. The best CV error result is 0.7649 acquired by GAM with smoothing splines. And then we used SVM, Random Forest and Bagging to predict popularity, which is divided into four categories, for each article. In this case, Random Forest gives us the best result, which achieves 50.4% prediction accuracy.

1 Introduction

Tons of news, stories and articles are published on website everyday. The author or editor would like their articles get shared and referred around the world as many times as it can be. But even the most skillful journalists can't be completely sure that their news directly hit people's tastes, no matter how well-organized or gorgeous it might be. There certainly exists a large amount of features contributing to an impressive online news or article. If one can know what kind of news people mostly like prior to the publication, creating an amazing article is just a matter of time and proper modification.

Our project aims to develop an effective learning algorithm to predict how popular an online article (especially news or short stories) would be before its publication by analyzing several statistic characteristics extracted from it. Measurement of popularity is the number of shares an article gets. We use real-world dataset from UCI Machine Learning Repository. Instead of inspecting each single word or phases of the contents, we used some derivations.

The input to our algorithm is a large list of features of articles which were published in Mashable: popularity of referenced articles; natural language features (e.g. global subjectivity and polarity); popularity of articles used the same keyword; number of digital media (e.g. images and videos) and published time (e.g. day of the week). We use these features to predict the popularity an article would be prior to its publication. We solve this problem in two ways. Firstly, we predicted the exact shares using different regression models (e.g. Regression, GAM, Lasso). After that, we split all articles into four categories (very unpopular/ unpopular/ popular/ very popular) and apply three classification methods (e.g. SVM, Random Forest and Bagging) to find the best classification for all test samples.

2 Related Work

Several popularity prediction methods on web content have been proposed and compared in recently years. Generally speaking, popularity prediction can be classified into four cases: single domain, cross domain, before publication and after publication. Previously, similar researches mainly dealt with predicting eventual online popularity based on early popularity. However, according to Tatar, Alexandru, et al. [1], predicting before publication is particularly useful for short lifespan web content like online news. To make it concrete, two different ways to indicate popularity: (i) Numerical prediction – predict the exact value of the popularity, (ii) Classification – predict the popularity range that an item is most likely to fall in.

Tsakias et al. [2] address the prediction task as a two-stage classification problem: binary classification to identify if news articles will receive comments and if they do, go through another binary classification to identify if the number of comments will be high or low. They used a random forest classifier based on a five groups of features and get a solid performance for the former task, while the performance for the latter degrades.

Moreover, a robust rolling windows evaluation of five state of the art models is demonstrated in [3]. They collected 39,000 articles from Mashable website and labeled those articles with a chosen threshold. The best result is given by random forest with a discrimination power of 73% for binary classification.

Bandari et al. [4], tried to address the prediction task both as a numerical and a classification problem and predict the number of tweets of news. The research demonstrated that although predicting the exact number of tweets may have a large error, it is possible to predict ranges of popularity on twitter with an overall 84% accuracy.

Another perspective on this problem is described in work [5], they used passive-aggressive algorithm to predict re-tweets number. Their study showed that the number of re-tweet is mainly determined by social features like number of followers and friends while tweet features like number of urls and tags could be a substantial boost.

3 Dataset and Features

The data set us obtained from UCI Machine learning repository. The original set contains 58 unique features with over 39,000 samples along with the actual number of shares for that article. Those key features can be grouped into seven categories, i.e. words, links, digital media, time, keywords, and features related to natural language processing (such as closeness to LDA topic, sentiment polarity etc.).

We start from manually filtering those features to decrease the number of features to 28 and pre-process the feature to normalize its mean and variance. The features we utilized in model is listed in Table together with their type.

Table 1 List of Features

Feature	Type(#)
Number of links	Number(1)
Number of links to other articles published by Mashable	Number(1)
Number of images	Number(1)
Number of videos	Number(1)
Number of keywords in the metadata	Number(1)
Data channel (Lifestyle, Entertainment, Business, Social Media, Tech or World)	Boolean(6)
Worst keyword (max, min, avg)	Number(3)
Best keyword (max, min, avg)	Number(3)
Average keyword (max, min, avg)	Number(3)
Shares of referenced articles in Mashable(max, min, avg)	Number(3)

The distribution of target value is shown in Fig 1.

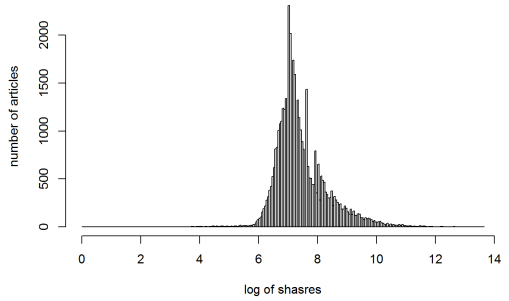

Figure 1 Histogram of Shares

4 Methods

After getting the dataset, we tried to solve the problem both in regression and classification methods to predict the popularity. Prior to that, we also use a feature selection method called best subset selection to choose the most significant features. To evaluate the trained algorithm, 10-fold cross validation is adopted to compute the CV error. Further, we then use error based on 0-1 loss to measure the result of classification models.

4.1 Best Subset Selection

In high dimensional problem, too many features always lead to a poor regression result. Hence it is very important to select a small amount of features with statistical significance. Best subset selection is one of the most popular methods used to select those features. For a feature set $F = \{1, 2, ..., p\}$, we consider every subset with i features and define $F^{(i)} = \{p_1^{(i)}, p_2^{(i)}, ..., p_i^{(i)}\}$ to be the subset of F and the one with minimum RSS (Residual Sum of Squares) among all F's subset with i features. Then we compare all $F^{(0)}, F^{(1)}, ..., F^{(p)}$ and find the one with minimum BIC (Bayesian Information Criterion), which is defined as:

$$BIC = \frac{1}{n}(RSS + \log(n) d \sigma^2),$$

where n is the number of observations, d is the number of features and σ is an estimate of the variance of the irreducible error ε.

4.2 Regression Methods

We first try regression methods to predict the shares of a given article. We run four classical regression models: linear regression, polynomial regression, GAM with smoothing splines and Lasso.

4.2.1 Linear Regression

Linear regression is the most basic regression model. Given a dataset $\{y_i, x_{i1}, \ldots, x_{ip}\}_{i=1}^n$. Assume that the relationship between target value and features is linear. Thus the model takes the form:

$$y = \beta_0 + \sum_{i=1}^{p} \beta_i x_i + \varepsilon = g(x) + \varepsilon$$

4.2.2 Polynomial Regression

Polynomial regression is an extension of linear regression, which replace each x_i with some polynomial functions of x_i. The model takes the form like:

$$y = \beta_0 + \sum_{i=1}^{p} \beta_i f_i(x_i) + \varepsilon,$$

where $f_i(\cdot)$ is a polynomial function with order p.

4.2.3 GAM (Generalized Additive Model)

GAM [6] gives us a more generalized form to extend linear regression model to non-linear cases. Instead of using the form of polynomial regression, GAM tries the form of:

$$y = \beta_0 + \sum_{i=1}^{p} \beta_i f_i(x_i) + \varepsilon,$$

where $f_i(\cdot)$ is the basis function. In this problem, we let $f_i(\cdot)$ to be the smoothing spline for quantitative features, which tries to minimize:

$$\sum_{i=1}^{n}(y_i - g(x_i))^2 + \lambda\int\left(g^{(m)}(t)\right)^m dt,$$

where m is the degree of freedom.

4.2.2 Lasso

Lasso is another regression method that tries to minimize:

$$\sum_{i=1}^{n}(y_i - g(x_i))^2 + \lambda\sum_{j=1}^{p}\left|\beta_j\right|.$$

The advantage of Lasso is that while gives out the regression fit as linear regression model, it can also show the statistical significance of features. And Lasso always obtains great results when only a small portion of features are of statistical significance.

4.3 Classification Methods

As we all know the shares of each article is quite variable and hard to predict. Besides, in the real world predicting the exact number of shares is not that interesting. Instead, people may just want to know whether the article is going to be popular or not. Therefore, we also try three classification methods called one-vs-one SVM, Random Forest and Bagging.

4.3.1 One-vs-one SVM

The **SVM** model is defined as:

$$\max W(\alpha) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} y^{(i)}y^{(j)}\alpha_i\alpha_j K(x^{(i)},x^{(j)})$$

$$s.t.\quad 0 \le \alpha_i \le C$$

$$\sum_{i=1}^{n}\alpha_i y^{(i)} = 0$$

where $K(x^{(i)},x^{(j)})$ is the kernel function. In our problem, we use both radial kernel and linear kernel. Radial kernel is defined as:

$$K(x^{(i)},x^{(j)}) = \exp\left(-\gamma\sum_{k=1}^{p}(x_k^{(i)} - x_k^{(j)})^2\right).$$

The linear kernel is defined as:

$$K(x^{(i)},x^{(j)}) = \sum_{k=1}^{p} x_k^{(i)}x_k^{(j)}.$$

One-vs-one SVM is an extended version of original SVM, which is used in a multiclass classification situation. Suppose we need to classify all data into K clusters. We construct C_K^2 SVMs. Each SVM is used to compare a pair of clusters. The final classification is determined by putting an observation to the most frequently assigned cluster.

4.3.2 Bagging (Bootstrap aggregation)

Bagging is a totally different classification method using decision tree. We first use bootstrap to generate B sets of training data from the original set of training data. For the b^{th} set of all B sets of training data, we fit a decision tree (classification tree) $f^b(x)$. Then we generate the final prediction by putting an observation to the most frequently assigned cluster.

4.3.3 Random Forest

Random Forest is very similar to bagging in general. However, there is some difference when fitting a decision tree for b^{th} bootstrap training data set. While making a split in the decision tree, we only take a random sample of m predictors into consideration. This method increases the diversity and hence can reduce the variance.

5 Results

5.1 Regression Result

To start with, let us take a look at the data distribution (Fig. 1). Similar to what is described in [1], the distribution can be thought as a log-normal distribution.

After that, the best subset feature selection suggests that a set of 8 features gives the minimum BIC and the BIC for each $F^{(i)}$ is shown in Fig. 2. As in Fig. 2 the minimum BIC is -700.7008 and the 8 top features are listed in table 2.

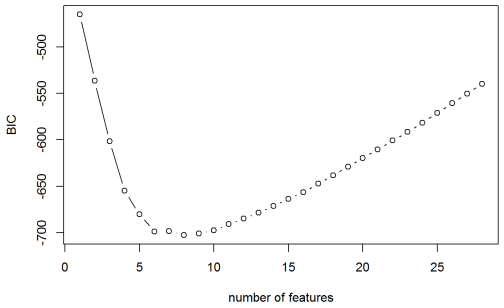

Figure 2 Best Subset Feature Selection

Table 2 Features with Statistical Significance

No	Features
1	Number of links to other articles
2	Article channel is entertainment
3	Min shares of worst keyword
4	Min shares of average keyword
5	Max shares of average keyword
6	Average shares of average keyword
7	Min shares of referenced articles
8	Closeness to top 3 LDA topic

Then we use four regression models described earlier.

For the linear regression model, we compute the test MSE while using 10-fold cross validation. The linear model gets a cross validation error of 0.789375.

For the polynomial regression, to prevent our models from a high order which usually over-fits the training data, we only consider the order of 2 to 5 for each predictor with quantitative values. However, since its meaningless to fit any polynomial functions on qualitative values, we simply keep them unchanged. The polynomial model has a minimum CV error of 0.7711951 when using cubic function (order 3).

For the GAM model, we fit smoothing splines with degree of freedom of 1,2,3,4. While using the smoothing splines with degree of freedom of 4, it gives the best CV error, which is 0.7649096.

The Lasso gives the CV error of 0.7869486 (Table 3). The left vertical dash line is the model with minimum MSE, and the right dash line indicates the model chosen by the one standard error rule. The above figures are the number of features chosen by the Lasso (namely the features with non-zero coefficients).

Table 3 Results of Regression

Regression Model	CV error
Linear	0.789375
Polynomial	0.7711951
GAM w Smoothing Spline	0.7649096
Lasso	0.7869486

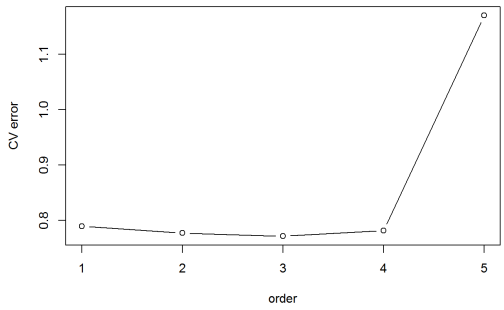

Figure 3 Linear and Polynomial Regression

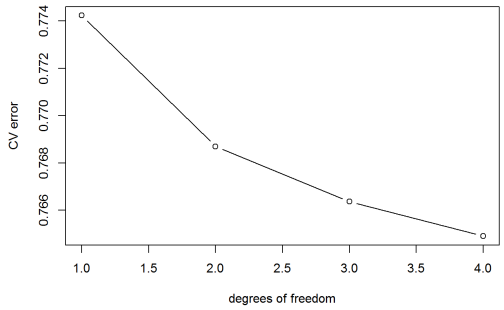

Figure 4 GAM w smoothing spline

As we can see from Fig.5, even by the one standard error rule, there are still 7 features that have a non-zero coefficients. Hence all those features have quite similar statistical significance under regression model.

Overall, the error range regression model gives us is generally in 0.76-0.79. The error itself, however, is unable to provide us with the intuition of how well the model works. Considering the fact that the exact number of shares does not make a big difference when the number of shares is quite high or low, we label the data sets into four categories to provide an intuitive result.

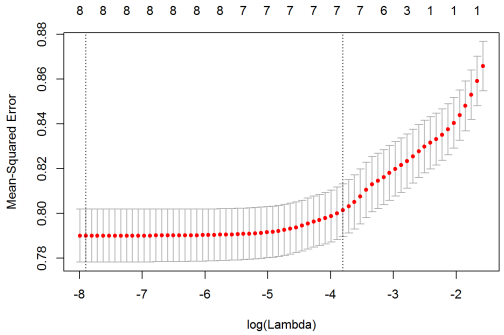

Figure 5 Lasso

5.2 Classification Result

In the classification section, since in real world, the extremely popular and unpopular articles are relatively rare and unpopular and popular ones are approximately even, we intuitively divide articles into 4 categories by their shares shown in Table 4.

Table 4 Popularity Categories

Interval of Log of Shares	Popularity	No.
[0, 6.5624)	Very unpopular	1
[6.5624, 7.2442)	Unpopular	2
[7.2442, 8.732]	Popular	3
[8.732, +∞)	Very popular	4

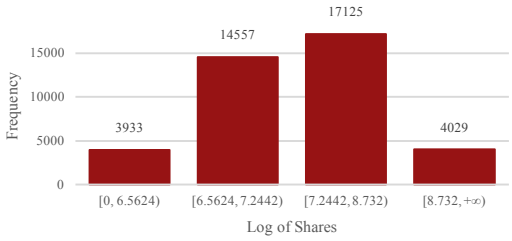

Figure 6 Popularity Categories

For the one-vs-one SVM with radial kernel, we try $C = 0.1, 1, 10, 100, 1000$ together with $\gamma = 0.5, 1, 2, 3, 4$ (25 combinations in total). It obtains a minimum error of 0.52 when the cost $C = 0.1$ and $\gamma = 0.5$. Then we use one-vs-one SVM with linear kernel and also try the same set of C. It obtains a minimum error of 0.55 when $C = 10$.

For bagging, we use all 28 features (namely the number of variables tried at each split) and build 500 decision trees

in total. In the case of bagging and random forest, we usually do not compute CV error anymore because it will be too time-consuming. Instead, we use out-of-bag error estimation (OOB). Notice that in random forest and bagging, we first use bootstrap to generate B sets of training data. To predict a single prediction for the i^{th} observation, we only need to take a majority vote for the predictions generated by training sets without $x^{(i)}$ as its training data. And the OOB for bagging is 50.03%.

For random forest, we try 5, 10, 15, 20 and 25 features (namely the number of variables tried at each split) and find that the minimum OOB, which is 49.64%, is obtained when $m = 10$ (use 10 features at each split).

Table 5 Confusion Matrix for Random Forest

	1	2	3	4	Error
1	179	2345	1389	20	0.95
2	164	7605	6717	71	0.47
3	98	4828	12047	152	0.30
4	15	815	3067	132	0.96

The confusion matrix shows that the class 2 and class 3 have quite good predictions. We can learn from the matrix that the there is good pattern for predicting popular and unpopular classes. However, the pattern for very popular and very unpopular classes are not that clear.

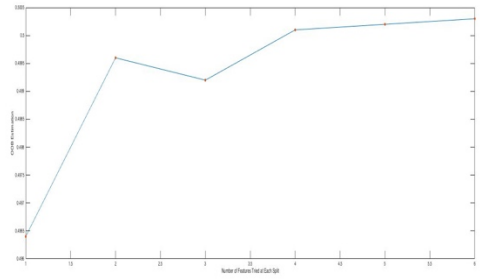

Figure 7 Random Forest and Bagging

Table 6 Classification Prediction Accuracy

Model	Accuracy
SVM	0.48
Random Forest	0.5036
Bagging	0.4997

6 Conclusion

In this project, various regression and classification models were trained for predicting the popularity of an online news article before its publication. We use shares to indicate popularity. The models were trained using Mashable data from UCI.

The GAM model performs best in regression problem. As we all know, Lasso and Linear regression are the model with lowest flexibility. Polynomial and GAM, on the other hand, are more flexible. More flexible models have more variance but less bias. The data set we have is actually a data with highly non-linear relationships between features. Therefore, highly non-linear models and models with high variance will give better result. That is exactly what we obtain in our regression model (GAM and Polynomial model give better result than Lasso and linear model). The Random Forest model performs best in Classification model. Random Forest model avoid the case that some strong features makes all trained decision tree to be highly correlated, which will definitely lead to a worse result. All in all, Random Forest is typically helpful when a large number of predictors are correlated, which is exactly what we face in our problem.

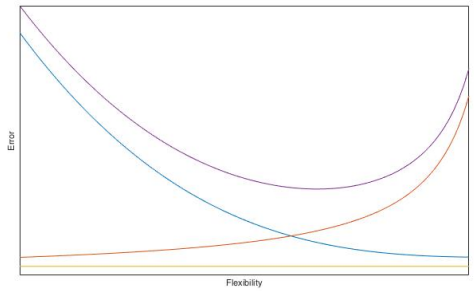

Figure 8 Bias-Variance Trade-off

We also demonstrated that the eight most significant features influencing the popularity of an online news are shown in Table 2.

Overall, the work done in this project was important in understanding the main reasons that effects popularity of online news from a statistical point, not from writing style and content. Author and editors can easily use the information we get to adjust their articles. In future work, we intend to do more exploration about the result we got. For example, we can find specific information on how important each feature is, so that the editor can focus on the most important ones and ignore some to save time.

References

[1] Tatar, Alexandru, et al. "A survey on predicting the popularity of web content." Journal of Internet Services and Applications 5.1 (2014): 1-20.

[2] Tsagkias, Manos, Wouter Weerkamp, and Maarten De Rijke. "Predicting the volume of comments on online news stories." Proceedings of the 18th ACM conference on Information and knowledge management. ACM, 2009.

[3] Fernandes, Kelwin, Pedro Vinagre, and Paulo Cortez. "A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News." Progress in Artificial Intelligence. Springer International Publishing, 2015. 535-546.

[4] Bandari, Roja, Sitaram Asur, and Bernardo A. Huberman. "The Pulse of News in Social Media: Forecasting Popularity." ICWSM. 2012.

[5] Petrovic, Sasa, Miles Osborne, and Victor Lavrenko. "RT to Win! Predicting Message Propagation in Twitter." ICWSM. 2011.

[6] James, Gareth, et al. An introduction to statistical learning. New York: springer, 2013.