XiaoA: A Robot Editor for Popularity Prediction of Online News Based on Ensemble Learning

Fei Long, Meixia Xu, Yulei Li, Zhihua Wu, Qiang Ling

XiaoA: A Robot Editor for Popularity Prediction of Online News Based on Ensemble Learning

Fei Long[1], Meixia Xu[1], Yulei Li[1], Zhihua Wu[1], and Qiang Ling[2]

[1]Chinaso Inc, Beijing, 100077
{longfei,xumeixia,liyulei,wuzhihua}@chinaso.com
http://www.chinaso.com
[2]Dept. of Automation, University of Science and Technology of China, Hefei 230027
{qling}@ustc.edu.cn

Abstract. In this paper, we propose a robot editor called XiaoA to predict the popularity of online news. A method for predicting the popularity of online news based on ensemble learning is proposed with the component learners such as support vector machine, random forest, and neural network. The page view (PV) of news article is selected as the surrogate of popularity. A document embedding method Doc2vec is used as the basic analysis tool and the topic of the news is modeled by Latent Dirichlet Allocation (LDA). Experimental results demonstrate that our robot outperforms the state of the art method on popularity prediction.

Keywords: robot, popularity prediction, ensemble learning, LDA

1 Introduction

Online news articles are attractive to a large amount of Internet users for the short length and rich content. However, the popularity of those articles are not evenly distributed. Only a small fraction of news attract the public attention successfully and become the so called hot news. For the popularity of online content is always related to the revenue, it is important to predict it beforehand.

The Blossom bot built by New York Times can solve this problem well. It is a chat bot within the messaging app Slack, which utilizes machine learning in its backend. It helps decide which story to post to social media, which got 380 percent more clicks than a typical post. In view of this, we develop a robot editor similar to Blossom named XiaoA to help editors improve there works. The main task of XiaoA is to predict the popularity of a large amount of news.

Predicting the popularity of online news is a great challenge for it is affected by several factors. According to the previous researches, quality of the content, article topic and the title influence the popularity of the article a lot, so these three factors are considered in our following prediction. In order to predict the popularity of the online news, the popularity itself needs to be quantified. Here, page view is selected as the only surrogate of popularity.

As mentioned in many former research papers, the lives of most online news articles are very short. Hence it is more valuable to predict the early popularity of a news. Fortunately, these data are available from kinds of news rankings. Moreover, the prediction of online content was modeled as regression or classification problem in preceding researches. So we present a method for popularity prediction of online news based on ensemble learning . The news ranking data of 163 (http://news.163.com/rank/) is used as our training and testing set.

The contributions of our paper are:

1) We propose a popularity prediction method for online news based on ensemble learning which outperforms the state of the art method.

2) We evaluate the performance of several classifiers on popularity prediction and get some meaningful conclusions.

3) We find the relationship between popularity and the news features.

2 Related work

Linear regression was used to predict the views of Digg and Youtube [11]. In [11], the long term popularity after 30 days can be predicted based on the early popularity within one hour. However, the popularity news needs to be predicted before promotion. Ye Zhang [15] presented a rationale augmented Convolutional Neural Network (CNN) model for text classification. The RA-CNN model outperforms some baseline CNN models. Although the popularity prediction of online news is also modeled as a classification problem, the unique features of online news make the application scenario a little different from this paper.

Several papers [7] [9] [12] [13] [2] [1] were published to predict the popularity of online news. Yaser etc. [7] defined the popularity of an article as page views within the first day. They casted popularity prediction as a regression problem. Various features were used in the prediction while some of which such as social media features are not available for news. [12] and [2] both used regression models to predict the popularity of news articles. However, those classifiers may not perform well among unpopular articles [1]. [13] addressed the prediction problem as a two stage classification. This kind of two stage classifier was demonstrated to perform below the average of other classifiers under the dataset in our former experiments. [9] used the number of votes to present the popularity of a story. However, it does not reflect the relationship between the article itself and its popularity. According to the works mentioned above, several classifiers are used comprehensively here to predict the popularity. We will show that our method can substantially improve the prediction accuracy of online news.

3 Preliminaries

3.1 Problem statement

We seek to predict whether a given news will be popular given its content and title. If a dataset of news and their corresponding PVs is available, the prediction

can be formulated as a classification problem. An article with its PVs above a certain threshold is deemed as popular and vice versa. The contents and titles of the articles are transformed into vectors, besides, the topics of the articles are also transformed into vectors through LDA. These three vectors of an article are combined as a comprehensive vector $x = [x_1, x_2, ..., x_n]$, as the input of the classifier. The output of the classifier is whether the article is popular or not.

3.2 Original dataset

Our dataset is crawled from 163 news. As shown in Figure 1, the category of the news is labelled in the top red box. Here the characters in the red box means 'news'. The red box in the second line means the 'PV rank within 24 hours'. The page views of a single news article is marked in the column of the right red box.

Fig. 1. Rank of news from 163.com.

The dataset contains 25733 pieces of news in 20 weeks. We use this dataset to train and test our popularity model.

3.3 Data preprocessing

As mentioned above, the article should be transformed into vector. The title, content and topic of an article are transformed into vectors respectively, and

combined as a comprehensive vector. Doc2Vec [8] is used to vectorize the title and content, and LDA [4] is used to represent the topic. Before vectorization, the stop words should be deleted. The stop words are some function words such as 'is', 'and', 'but' and 'or' etc., and some meaningless symbols. For we know the existing categories of 163 news is 10, the topic of LDA is chosen to 10 accordingly.

4 Proposed methodology

According to previous literatures, there is no single classifier that can overwhelm all the others. Therefore, we bring ensemble learning for popularity classification. Several classifiers are chosen as our component learners such as Random forest (RF), Neural network (NN), Support vector machine (SVM), Logistic regression (LR), Nearest centroid (NC) and Restricted Boltzmann machine (RBM). We will give a brief introduction of three main learners in this paper.

4.1 Component learners

Support vector machhine (SVM): As mentioned above, the content, title and topic of an article will be transformed into vectors and combined together as the input. If the dimension of the combined vector is n, the input of SVM is $x^{(i)}$=[$x_1^{(i)}$, $x_2^{(i)}$,..., $x_n^{(i)}$] . The key idea of SVMs is to find a maximum margin hyperplane that separates two sets of points in a higher dimensional space[5]. Each article in the training set is deemed as a point. The combined vector $x^{(i)}$=[$x_1^{(i)}$, $x_2^{(i)}$,..., $x_n^{(i)}$] is a n-dimensional vector represents the article i itself, the value $y^{(i)}$ is the PVs of article i. Hence the article can be deemed as a $n+1$-dimensional point $\mathbf{x}^{(i)} = (x_1^{(i)}, x_2^{(i)}, ..., x_n^{(i)}, y^{(i)})$. What we want is to find a 'line' as the boundary of the two sets of points that represents popular and unpopular points respectively. The boundary of the two sets is defined as:

$$\mathbf{y}(\mathbf{x}) = w^T\mathbf{x} + b \tag{1}$$

The problem can be represented as:

$$min_{w,b,\xi}\frac{1}{2}\|w\|^2 + \frac{\gamma}{2}\sum_{i=1}^{m}\xi_i^2 \tag{2}$$

subject to

$$\mathbf{y}_i(w\mathbf{x}_i + b) \geq 1 - \xi_i, i = 1, 2, ..., m$$
$$\xi_i \geq 0, i = 1, 2, ..., m$$

where γ is the penalty parameter, and ξ_i is the slack variable[3].

Neural network (NN): Artificial neural networks were proved to be good classifier. if the weight is w_{ij}, where i represents the start node and j represents the end node. Node i has s input nodes, the output of this node can be

represented as:

$$a_i = f(\sum_{j=1}^{s} w_{ij} + b_i) \tag{3}$$

where b_i is the bias of node i, and $f(\cdot)$ is the activation function. In this paper, the combined vector $x^{(i)}$=[$x_1^{(i)}$, $x_2^{(i)}$,..., $x_n^{(i)}$] of article i is the input of the neural network. If $y^{(i)}$ is the PVs of article i, $(x^{(i)}, y^{(i)})$ is a training point accordingly. The network can be solved by a back propagation (BP) algorithm [10].

Random forest (RF): Random forest is an ensemble learning method based on decision trees [6]. A news article can be represented as a vector x=[x_1, x_2,..., x_n], and y represents whether the article is popular, y={0,1}. We know that the training set $\{(x^{(1)}, y^{(1)})$,...,$(x^{(m)}, y^{(m)})\}$ can form a decision tree based on information gain. However, the decision tree may overfit the training set. Random forest utilizes a multitude of decision trees to make classification by voting of these decision trees. The training set of each tree is a subset of the whole training set, and is created by a bootstrap manner.

4.2 Ensemble learning

Ensemble learning accomplish learning task through constructing and combining various learners. Our component learners are already introduced above, ensemble learning integrate all the component learners to achieve a better performance. Before introducing the algorithm, we should answer some questions:

(1) How to quantify the influence of topic, content and title of the article on its popularity?

(2) How to determine the threshold τ of the popularity of these news articles?

(3) How to integrate the component learners to get a strong learner?

Title and content are coupled tightly. Although same content can have totally different titles, the title should be related to the content finally. On the other hand, the topic is not so tightly coupled with the content. The contents of articles under the same topic can be of a wide variety. If the topic vector o is p-dimensional, o=[o_1,o_2,...,o_p], the content vector c is q-dimensional, c=[c_1,c_2,...,c_q], and the title vector t is also q-dimensional, t=[t_1,t_2,...,t_q], the input vector x can be defined as x=[o,c,t]. In order to test the contributions of the content and title to the popularity, we use x=[o,c',t'] instead of [o,c,t], in which $c'=\alpha c$, $t'=(1-\alpha)t$, where α is the contribution weight.

The threshold τ can be estimated by the dataset and parameter α will be tested by experiments. We use voting policy to determine the classification result. Assume the whole set is: D={d_1,d_2,...,d_n}, where d_i is a news contains title and content. U={d_1,d_2,...,d_m} is the training set, $m < n$. Accordingly, the test set is V=D-U. The Ncomponent leaners can be deemed as functions. Each function F_i can be trained using the training set U by SGD or other methods given the parameters τ and α. The input of function F_i is the vector x of an article mentioned above, and the output is the classification result y, $y \in \{0, 1\}$. The algorithm of the ensemble learning is as follows.

Algorithm 1 Popularity prediction based on ensemble learning

Input: U, x, τ, α
Output: y

Transform each news article in D into vector x using LDA and Doc2Vec with certain α
for all $i \in [1, N]$ **do**
  Train the function F_i using training set U with certain threshold τ
end for
for all $F_i \in \{F_1, F_2, ..., F_N\}$ **do**
  if $F_i(x) == 1$ **then**
    $T++$
  else
    $F++$
  end if
end for
if $T > F$ **then**
  return $y = 1$
else
  return $y = 0$
end if

5 Experiments

We traced the 163 news rank for 20 weeks and got 25733 pieces of news with their PVs in 24 hours. Since the measurement of popularity differs in each field (for example, the average PVs of entertainment is apparently higher than other fields), the threshold of popularity should be considered comprehensively. According to our statistics, the percentage of the popular news is about 10% of the whole dataset, the threshold of all the fields is about 15000 PVs, that is, $\tau = 15000$. To balance the positive and negative samples, the popular news and unpopular news are chosen with a proportion of 1:1. Finally, 8000 articles are chosen as our training set with 4000 popular news and 4000 unpopular news, 1600 news are chosen as the test set similarly.

In order to find out the influence between title and content on the popularity of an article, we tune the parameter α to 0, 0.5 and 1 respectively. $\alpha = 0$ means that we only consider the influence of the title. $\alpha = 1$ means that we only consider the influence of the content. $\alpha = 0.5$ means that we consider both the title and the content equally. The dimension of the topic vector is 4, the dimension of the content and title vector both are 200, hence the dimension of vector x is 404. We evaluate the accuracy, precision, recall and F1 score of each base learner, the parameter α is tuned to 0, 0.5 and 1. Since random forest (RF) is already an ensemble learning method, the rest learners are combined to form the ensemble learning in our first experiment. The results are shown in the following tables.

We use F1 score to measure the performance of all the leaners mentioned in Table 1 to Table 3. Three conclusions can be drawn from the results:

Table 1. Popularity prediction results with $\alpha = 0.5$

Base learner	Accuracy	Precision	Recall	F1
RF	0.778	0.791	0.759	0.775
NN	0.772	0.739	0.843	0.788
SVM	0.784	0.938	0.610	0.740
LR	0.608	0.608	0.622	0.615
NC	0.648	0.674	0.582	0.625
RBM	0.545	0.528	0.924	0.672
Ensemble (without RF)	0.752	0.725	0.815	0.767

Table 2. Popularity prediction results with $\alpha = 0$

Base learner	Accuracy	Precision	Recall	F1
RF	0.756	0.764	0.743	0.754
NN	0.707	0.678	0.795	0.732
SVM	0.792	0.956	0.614	0.748
LR	0.519	0.512	0.964	0.669
NC	0.574	0.577	0.570	0.574
RBM	0.523	0.514	0.968	0.671
Ensemble (without RF)	0.636	0.593	0.880	0.709

(1) Some base learners perform obviously better than others on our dataset. Seen from the results, RF, NN, and SVM are much better than the other three in F1 score about 10% on average whenever α is 0, 0.5 or 1.

(2) The ensemble learning can not always perform better than each of the base learner especially when the performances of the base learners differ a lot. As shown in the results, the ensemble learning method is better than LR, NC and RBM in all situations. However, it can not always perform better than NN and SVM. The poor performances of LR, NC and RBM may degrade the performance of the ensemble learning.

(3) The content is more important than title for we find that the average performance of the base learners and ensemble learning increases when α increases. Although some clickbait title may attract many clicks at the beginning, the netizens will soon be familiar with this kind of title and ignore them finally.

Notice that the performance of RF, NN and SVM is similar and obviously better than others, we use them as the base learners in our second experiment. The ensemble learnings in the first and second experiment are marked as En1 and En2. The experiment results are shown in Table 4 to Table 6.

As shown in Table 4 to Table 6, the performance of En2 improves a lot. The increasing of F1 score of En2 when α is 0, 0.5 and 1 can also demonstrate conclusion (3). The good performance of En2 demonstrate conclusion (2) from the opposite side that the ensemble learning can improve the performance when the performances of the base learner are similar. The comparison between base

Table 3. Popularity prediction results with $\alpha = 1$

Base learner	Accuracy	Precision	Recall	F1
RF	0.784	0.771	0.811	0.791
NN	0.766	0.744	0.815	0.778
SVM	0.774	0.810	0.719	0.762
LR	0.667	0.679	0.639	0.658
NC	0.653	0.676	0.594	0.632
RBM	0.507	0.505	0.984	0.668
Ensemble (without RF)	0.766	0.724	0.863	0.788

Table 4. Improved popularity prediction results with $\alpha = 0.5$

Base learner	Accuracy	Precision	Recall	F1
RF	0.778	0.791	0.759	0.775
NN	0.772	0.739	0.843	0.788
SVM	0.784	0.938	0.610	0.740
En1	0.752	0.725	0.815	0.767
En2	0.800	0.861	0.719	0.783

learners and En1, En2 can be seen from Figure 2, which represents the trend that the content improving the performance with the increasing of weight α.

Our work is also compared with a state of the art news popularity prediction method [14]. Three different algorithms such as LPBoost, Random Forest and AdaBoost were implemented in this paper to predict the popularity of news articles. The dataset of 39797 news were collected from UCI machine learning repository. According to this paper, the best popularity prediction model was adaptive Boosting on the MCI dataset, which had achieved F1 score of 73% and accuracy of 69%. Our En2 method can achieve F1 score of 80.2% and accuracy of 80.4% when $\alpha = 1$ under a much less dataset of 8000 news articles.

6 Conclusion

This paper presents a robot editor called XiaoA to predict the popularity of online news based on ensemble learning. Our ensemble learning method outperforms the state of the art prediction method. Besides, we find that the content of an article plays the most important role in determining the popularity. We also find that learners with similar performance will have a better performance using ensemble learning. Although ensemble learning achieves good performance in popularity prediction, a lot of factors will also affect the popularity of online news, which will be discussed in out future researches.

Table 5. Improved popularity prediction results with $\alpha = 0$

Base learner	Accuracy	Precision	Recall	F1
RF	0.756	0.764	0.743	0.754
NN	0.707	0.678	0.795	0.732
SVM	0.792	0.956	0.614	0.748
En1	0.636	0.593	0.880	0.709
En2	0.776	0.835	0.691	0.756

Table 6. Improved popularity prediction results with $\alpha = 1$

Base learner	Accuracy	Precision	Recall	F1
RF	0.784	0.771	0.811	0.791
NN	0.766	0.744	0.815	0.778
SVM	0.774	0.810	0.719	0.762
En1	0.766	0.724	0.863	0.788
En2	0.804	0.814	0.791	0.802

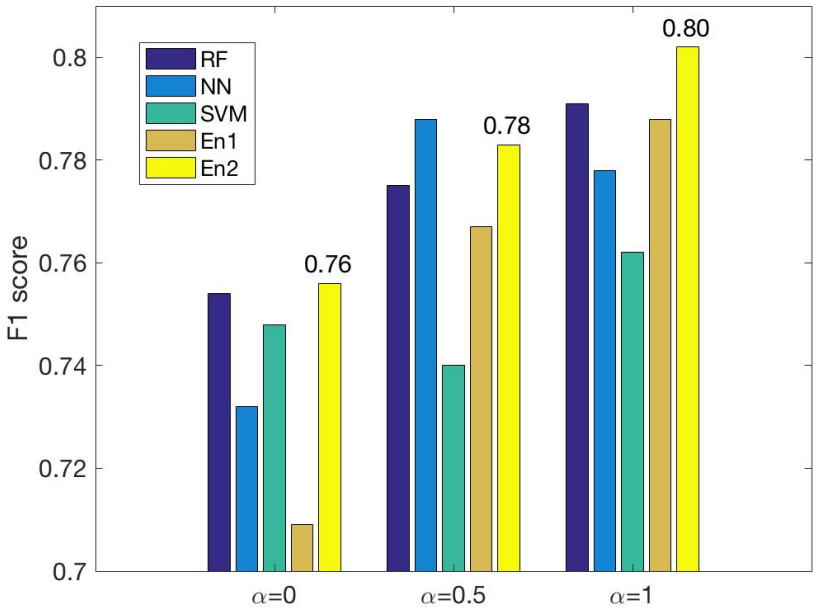

Fig. 2. F1 score of the learners

References

1. Arapakis, I., Cambazoglu, B.B., Lalmas, M.: On the Feasibility of Predicting News Popularity at Cold Start, pp. 290–299. Springer International Publishing, Cham

(2014), https://doi.org/10.1007/978-3-319-13734-6_21
2. Bandari, R., Asur, S., Huberman, B.A.: The pulse of news in social media: Forecasting popularity. In: Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media. pp. 26–33 (2012)
3. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York, Inc. (2006)
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. 3, 993–1022 (Mar 2003), http://dl.acm.org/citation.cfm?id=944919.944937
5. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20(3), 273–297 (Sep 1995), https://doi.org/10.1007/BF00994018
6. Ho, T.K.: Random decision forests. In: International Conference on Document Analysis and Recognition. pp. 278–283 (1995)
7. Keneshloo, Y., Wang, S., Han, E.H., Ramakrishnan, N.: Predicting the popularity of news articles. In: Siam International Conference on Data Mining. pp. 441–449 (2016)
8. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32. pp. II–1188–II–1196. ICML'14, JMLR.org (2014), http://dl.acm.org/citation.cfm?id=3044805.3045025
9. Lerman, K., Hogg, T.: Using a model of social dynamics to predict popularity of news. In: Proceedings of the 19th International Conference on World Wide Web. pp. 621–630. WWW '10, ACM, New York, NY, USA (2010), http://doi.acm.org/10.1145/1772690.1772754
10. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Neurocomputing: Foundations of research. chap. Learning Representations by Back-propagating Errors, pp. 696–699. MIT Press, Cambridge, MA, USA (1988), http://dl.acm.org/citation.cfm?id=65669.104451
11. Szabo, G., Huberman, B.A.: Predicting the popularity of online content. Communications of the Acm 53(8), 80–88 (2008)
12. Tatar, A., Antoniadis, P., Amorim, M.D.D., Fdida, S.: Ranking news articles based on popularity prediction. In: International Conference on Advances in Social Networks Analysis and Mining. pp. 106–110 (2012)
13. Tsagkias, M., Weerkamp, W., Rijke, M.D.: Predicting the volume of comments on online news stories. In: ACM Conference on Information and Knowledge Management. pp. 1765–1768 (2009)
14. Vilas, T., Dhanashree, D.: Analysis of online news popularity prediction and evaluation using machine intelligence. International Journal of Mathematical and Computational Methods 2(2), 120–131 (2017)
15. Zhang, Y., Marshall, I., Wallace, B.C.: Rationale-augmented convolutional neural networks for text classification. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 795–804 (2016)