Web advertisement detection using Naive Bayes

Xin Deng¹, Lunqing Hou², Fei Wang³

¹College of Computer Science and Software, Shenzhen University Shenzhen, 518060 P.R. China
²College of Computer Science and Software, Shenzhen University Shenzhen, 518060 P.R. China
³College of Computer Science and Software, Shenzhen University Shenzhen, 518060 P.R. China
¹1062388964@qq.com, ²lqhou@outlook.com, ³wangfei2017@email.szu.edu.cn

ABSTRACT Nowadays, with the development of the internet, more and more people find information on the websites. So the network promoters pretend to write advertising content on websites, advertising content will reduce the user’s experiment if the users always find the advertisement instead of the content they need. It will take web managers a lot of time to review whether the content is advertisement or useful content. Even worse, when we daily use message applications, there are always groups full of advertisement messages. It takes a lot of time to manage the users and the content, so many managers just leave it and the groups will be worthless. Basically, when we try to separate useful content from advertisement, we use the keywords. For example, if the content contains keywords like promotion champion, sale, we prefer to define it as advertisement. On the other hand, if the content has keywords like weekend, dinner, we prefer to define it as useful content. This processing can be automated as a text classification problem with the help of machine learning algorithms: k-NN Algorithm, Naive Bayes Algorithm, support vector machine, decision tree, neural network [1][2][3][4][5][6]. This paper applies Naive Bayes to advertisement detection in web content [3].

1. INTRODUCTION

This paper mainly tries to apply Naive Bayes into advertisement problem. Human classify content actually by keywords, when we see content contains keywords like sale, promotion, we prefer to think it is advertisement. This is the same schedule like Naive Bayes. Naive Bayes will count actually what is the probability, when one keyword appears. So, it is convenient for the web managers to delete the advertisement and give the users more useful information. Advertisement content is gathered from a website full of advertisement without any manual management, normal content is gathered from an authoritative news website with strict management.

2. Related work

There are several ways of web spam detection. Some researchers use link based techniques, in detail, they use links and ranking algorithms method like link farm detection to detect spam. They want to detect the web page that attracts search engine referrals [7]. Some researchers try to analysis reviews spam by combining review content and user behavior. They summarize view spam to 3 types: untruthful opinions, reviews on brands only and non-reviews [8][9]. Some researchers use machine
learning algorithm to classify the reviews to two classes spam and non-spam. They collect the data by selecting the duplicate review as spam content and other review as normal content [10]. This paper gathers spam data from a website advertisement category and normal data from a news website.

3. Web advertisement detection

3.1 Data source

100,000 advertisement articles from a website full of advertisement are labeled as negative dataset. 100,000 normal content from an authoritative news website are labeled as positive dataset. We use 80,000 positive and 80,000 negative data as training set; 20,000 positive and 20,000 negative data as test set.

3.2 Data processing

At first, the articles contain words that need to be transformed as features that are represented as vectors.

Define articles as $D_i$, define words as $w_i$, So:

$D_1 = \{w_{11}, w_{12}, ..., w_{1n}\}$
$D_2 = \{w_{21}, w_{22}, ..., w_{2n}\}$
...
$D_n = \{w_{n1}, w_{n2}, ..., w_{nn}\}$

Let V represents the indexed vocabulary.

$V = \{w_1, w_2, ..., w_k\}$

When the vocabulary is built, every word has a unique id, so the articles can be transferred to vectors.

Define article as $D_i$, define word’s id as $k_i$, so:

$D_1 = \{k_{11}, k_{12}, ..., k_{1n}\}$
$D_2 = \{k_{21}, k_{22}, ..., k_{2n}\}$
...
$D_n = \{k_{n1}, k_{n2}, ..., k_{nn}\}$

3.3 Feature selection

The data is transferred to vectors, and a vocabulary is built. If all the keywords are taken into account, there will be too many features. Not only too many features will make the model too complex with many parameters, but also they may contain noise to the model.

At first the word’s frequency need to count, so the word has an id and frequency. The words that have a very high frequency will be removed because they may appear in positive and negative dataset. If the word’s frequency is too low, it will be removed from the features, because it is some special content’s feature instead of the class’s feature.

3.4 Naive Bayes

Naive Bayes is an algorithm that depends on Bayes Rules, the articles $D = \{D_1, D_2, ..., D_n\}$ are represented as vectors $d$, the classes are represented as $C = \{c_1, c_2\}$ while $c_1$ is positive and $c_2$ is negative.

The origin Naive Bayes is:

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$$

The class that the content belongs to will be:

$$C_{MAP} = \arg \max_{c \in C} P(c|d), c \in C$$

$$C_{MAP} = \arg \max_{c \in C} \frac{P(d|c)P(c)}{P(d)}, c \in C$$

The denominator is the same, so drop the denominator:

$$C_{MAP} = \arg \max_{c \in C} P(d|c)P(c), c \in C$$
The document is represented as \((x_1, x_2, \ldots, x_n)\):

\[
C_{MAP} = \arg\max \, P(x_1, x_2, \ldots, x_n \mid c)P(c), \, c \in C
\]

4. Experiment

4.1 Share words trend
At first the top frequency words are counted in both classes. Since they share a same vocabulary of 10000 words. The figure shows that the classes only share a little more than 26% words in top 200, even in top 900 they only share a little more than 34% words. So the two classes will be easy to separate, because they are different enough to each other.

Fig. 1. Share words trend

4.2 Confusion matrix
According to the confusion matrix, the experiment has achieved expected result for spam detection. In 19906 positive samples, only 17 samples classified to negative class by mistake. In 20094 negative samples, only 172 samples classified to positive class by mistake.

Fig. 2. Confusion matrix
Fig. 3. Normalized confusion matrix

4.3 Analysis Detection percentage of different threshold

Normally, when the probability is bigger than 50% of a class or the probability of a class is the biggest, the model will predict to this class. To build a sensitive model to the negative content, we use a list of thresholds to show the improvement. It turned out that the thresholds are not useful, because the normal probability has achieved a very high accuracy.

4.4 Precision, Recall, F1 Score

Precision can be used to validate the classifier’s ability of not to label a negative sample as positive. Recall can be used to measure the classifier’s ability of find all the positive samples [11] [12] [13].

Define:

True Positives (TP) means that the positive article is classified as positive class.

True Negatives (TN) means that the negative article is classified as negative class.

False Positives (FP) means that the negative article is classified as positive class.

False Negatives (FN) means that the positive article is classified as negative class.

\[
\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}
\]
5. Conclusions

Advertisement detection or spam detection is critically important nowadays. We use message applications to communicate; websites to find information, shopping applications to buy products. These applications are full of advertisement content or spam. The advertisement content waste a lot of our time and reduce our using experience. The content managers also pay much attention to these content. This paper proposes to use Naive Bayes to solve the problem automatically. At first, we try to analysis the words distribution in the two classes, so we can make sure that they are different enough to separate. The Naive Bayes Algorithm shows good performance in classifying the content to normal and spam classes. We try to figure out whether the threshold can make a contribution to the advertisement detection. It turns out to be helpless, because the model has achieved a very high performance at threshold of 50%.

Advertisement or spam exists everywhere, their words distribution different from each other. In the future work, more advertisement resources will be added to the dataset, so the model can be trained with more data and can detect more type of advertisement or spam. What’s more, the advertisers are changing their strategy, so the model has to update with more data added to it.

REFERENCES

[1] Korde, V., & Mahender, C. N. (2012). Text classification and classifiers: A survey. International Journal of Artificial Intelligence & Applications, 3(2), 85.

[2] Bijalwan, V., Kumar, V., Kumari, P., & Pascual, J. (2014). KNN based machine learning approach for text and document mining. International Journal of Database Theory and Application, 7(1), 61-70.

[3] Aghila, G. (2010). A Survey of Naive Bayes Machine Learning approach in Text Document Classification. arXiv preprint arXiv:1003.1795.

[4] Tong, S., & Koller, D. (2001). Support vector machine active learning with applications to text classification. Journal of machine learning research, 2(Nov), 45-66.

[5] Johnson, D. E., Oles, F. J., Zhang, T., & Goetz, T. (2002). A decision-tree-based symbolic rule induction system for text categorization. IBM Systems Journal, 41(3), 428-437.

[6] Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. In Advances in neural information processing systems (pp. 649-657).

[7] Ghiam, S., & Pour, A. N. (2012). A survey on web spam detection methods: taxonomy. arXiv preprint arXiv:1210.3131.

[8] Jindal, N., & Liu, B. (2008, February). Opinion spam and analysis. In Proceedings of the 2008 international conference on web search and data mining (pp. 219-230). ACM.

[9] Farooq, S., & Khanday, H. A. Opinion Spam Detection: A Review.

[10] Jindal, N., & Liu, B. (2007, May). Review spam detection. In Proceedings of the 16th international conference on World Wide Web (pp. 1189-1190). ACM.
[11] Stehman, S. V. (1997). Selecting and interpreting measures of thematic classification accuracy. *Remote sensing of Environment, 62*(1), 77-89.

[12] Powers, D. M. (2011). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation.

[13] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *Journal of machine learning research, 12*(Oct), 2825-2830.