International Journal of Database Theory and Application
Vol.9, No.7 (2016), pp.207-214
http://dx.doi.org/10.14257/ijdta.2016.9.7.18

A Study on Naïve Bayes Classifier Based Document Classification Scheme with an Apriori Feature Extraction

Jong-Yeol Yoo, Min-Ho Lee, Grace Aloyce and Dong-Min Yang

Dept. of Information & Communications Engineering, Daejeon University, Daejeon, Korea
E-mail : gum10011@naver.com , biy006@naver.com, g.aloyce@gmail.com, dmyang@dju.ac.kr

Abstract

A document classifier is an essential tool for classifying the various types of documents being generated in the Big Data era. In recent years, the wide variety of information services available for use with smartphones and portable mobile devices (tablets) have provided a technique that efficiently classifies the quality of sorted data. A common type of document classification scheme is the naïve Bayes classifier. The Naïve Bayes scheme is based on performance classification, which varies widely depending on the method of extraction used in the document. In this paper, we propose a system model that offers feature extraction methods which combine frequency with associated words. This model is then applied to the Naïve Bayes classifier to precisely classify documents. This method is proposed as an alternative to using traditional classification techniques. In addition, experiments will be evaluated by the existing document classification techniques and the proposed techniques.

Keywords*: Document Classifier, Naïve Bayes, TF-IDF, Association Word, Apriori*

1. Introduction

In today's modern society, large-scale information is generated by Big Data. Due to the large and unforeseen amount of data necessary for sharing and storing information, the efficient classification of generated data is not easy. Some files featuring large-scale data generated in a short period of time include data characters, figures, images, and videos. Thus, a variety of data types must be classified based on character features, value features, extraction features, and inter-image features. The process required to extract these various data tends to be complex and time consuming. In this paper, we mainly focus on the classification and use of the character data types that are most representative of the majority of email documents.

The data found in email documents make use of datasets consisting of ham and spam email, both of which come from the Enron Email Dataset [1]. These datasets consist of characters that is, techniques for classifying a document comprising text [2-4] consisting of two types. The first process involves extracting characteristic points from a text document. This process involves finding a key that can be analyzed in order to distinguish between the characteristics of the document prior to and following document classification. The second process required for the classification of documents is to apply the article sorter via the extracted feature points. Completion of this process requires the use of the typical classifier made by the Naïve Bayes Company. In the present paper, we propose the use of two steps to effectively classify a document as an email.

In order to extract the characteristic point of an email document, the full text of the document must be analyzed and word sorting must be conducted during pre-processing. During pre-processing, the number of occurrences of the target word must be counted and a set of associated words must be generated. This makes it possible to examine the

ISSN: 2005-4270 IJDTA

relevance of the word at the same time. Once the feature points of the document have been extracted, any frequently-occurring words associated with the frequency of the words is given a weight value. This is done to classify the document and propose a scheme for increasing the probability classification of the document. Therefore the proposed technique is evaluated against the probability of the classification by calculating only the frequency of the existing words extensibly.

2. Related Works

2.1. A Term Frequency-Inverse Document Frequency (TF-IDF) [5-7]

Term Frequency–Inverse Document Frequency (TD–IDF) is one of the most common methods used to extract the feature points of words in a text document. This method is used when a document set consists of several documents; the words existing in all the documents are used as statistical values to indicate the importance within a particular document. Term frequency (TF) is a value that represents the frequency at which a word occurs in the document. The TF value increases the more often the word appears in the document. Inverse document frequency (IDF) is the inversed version of document frequency (DF), which is a value indicating the number of documents containing a specific word. The DF value is therefore greater throughout the entire document and appears frequently in the entire document. IDF is the reciprocal of DF; the higher the IDF value is, the higher the document discrimination will also be. The TF–IDF weight of the keyword expression is obtained with the following equation (1).

$$W = tf * idf \tag{1}$$

- W : Weight value of the particular word
- tf : Specific word frequency in the current document
- idf : Reverse frequency of document containing certain words

2.2. Naïve Bayes Classifier [8-9]

The naïve Bayes classifier is a probabilistic classifier model. It is applied between each character in the independent assumed Bayes theorem. In the Bayes theorem, when x, y parameters exist, category 1 $p_1(x,y)$ and category 2 $p_2(x,y)$ exist within those parameters. If $p_1(x,y) > p_2(x,y)$, this value belongs to the first category, and if $p_1(x,y) < p_2(x,y)$, this value belongs to the second category. The Bayes theorem can be used to separately calculate the probability of each category, and thus can be used to classify the larger probability.

The naïve Bayes classifier establishes the learning features of the document, then learns how to use these features to classify newly inputted document data into the correct category. This classifier is one of the most widely used algorithms for document classification because it is simple and effective. The naïve Bayes classifier defines different classes of categories according to (1) the features of the document category and (2) the calculated probability of a feature belonging to a category.

$$P(C \mid F) = \frac{P(F) \times P(F \mid C)}{P(C)} \tag{2}$$

C represents the class in which the category belongs and F represents a particular word. $P(C|F)$ represents the probability of a certain word F being included in a certain class C. According to the naïve Bayes classifier, the results $P(C|F)$ value cannot be calculated directly. The formula is therefore modified to $P(F)\cdot P(F|C)/P(C)$ to calculate the value. In $P(F)$, the value for all category classes is the same as the probability that a particular word appears in the entire document. $P(C)$ is the ratio of the total number of documents belonging to class C. This ratio is calculated by dividing the number of words belonging to class C by the number of words in the entire document. $P(F)$ is the probability that a

particular word *F* will appear in class *C*. This is calculated by dividing a certain number of words by the number of words in the entire document.

2.3. Apriori [10-12]

Apriori is an algorithm which analyzes the relation between groups of specific items by using the association rule. The procedure of Apriori consists of two steps. The first step is to find the frequent item set. Frequent item set is a collection of the items with the transaction approval rating than the minimum support (min-support). The second step is to generate the association rule. This step derives the association rules based on frequent item set. Approval rating (support) and reliability (confidence) for generating the association rules are explained in the equation (3).

$$Confidence = \frac{(X \cup Y) \text{ number of transactions included.}}{X \text{ number of transactions included.}} \quad (3)$$

Apriori operates as described in Figure 1. At first, in the TDB database items, in order to produce the table C_1, we obtain a unique set of times by removing duplicity among the items. In table C_1, because the support value of D, 1, is less than the minimum support (=2), it is removed and the table L_1 is generated. From table L_1, table C_2 with item sets of two elements is derived. The item sets are discard items sets below the minimum support from C_2 and table L_2 is generated.

From table L_1, table C_2 with item sets of two elements is derived. The item sets are discard items sets below the minimum support from C_2 and table L_2 is generated. From table L_2, table C_3 with items sets of three elements is derived. This process is repeatedly performed. Finally, table L_1 with the item set $\{B, C, E\}$ is derived and becomes associable.

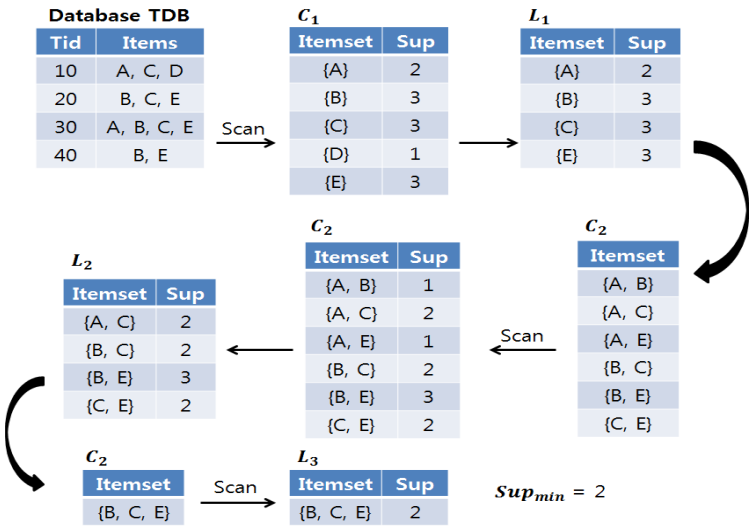

Figure 1. Procedure of Apriori Algorithm

3. Improving Feature Extraction [13]

In this paper, we used ham and spam email documents provided by the Enron Email Dataset [1] as datasets for the feature extraction. Before the feature points of the ham and spam email documents are extracted, the documents are pre-processed using morphological analysis. The morphological analysis is organized in Table 1. In this paper, words are taken from a generated ham and spam email document stored in the database; the frequency at which each word occurs is counted. For example, in the ham email document, the frequency of the word in the document is counted whether it occurs 1 time

or 1000 times. The frequency is then stored in the database. An operation is then performed to find the weight associated with a specific set of words. One way to find the specified set of words can be found in Table 1. This method involves assuming that the word selected from each document has a frequency of more than 50% and thus occurs frequently enough to create a related word set based on related words. An example of this method can be found in Table 1. In this example, according to the ham email, the words "go-game" and "Lee Se-do" appear 1–4 times, making the probability of their appearance in the document 75%. An association between these two words is then created. The associative set of words is then tied the original word. The two words appear in the document at a frequency of at least 50%. These words are grouped to the *n*-word as a set to generate an associative set of words.

Table 1. Example of Stemming Article Using the Ham Email

Document name	Word.
Ham-mail 1	Go-game, Alpha-go, Lee Se-do
Ham-mail 2	Go-game, Alpha-Go, Lee Se-do, Google,
Ham-mail 3	Go-game, Lee Se-do
Ham-mail 4	Lee Se-do, Lee Chang-Ho

The generated word's frequency and associated words set the extracted feature points. These can then be configured, as shown in Table 2.

Table 2. Example of Word Frequency and Word Association Weighting of the Row Email in this Study

Document name.	Go-game	Alpha-Go	Lee Se-do	Google	Lee Chang-Do	{Go-game, alpha-Go}	{Go-game, Lee Se-Do}	{Alpha-Go, Lee Se-Do}	{Go-Game, Alpha-game, Lee Se-Do}
Ham-mail 1	1	1	1	0	0	1	1	1	1
Ham-mail 2	1	1	1	1	0	1	1	1	1
Ham-mail 3	1	0	1	0	0	0	1	0	0
Ham-mail 4	0	0	1	0	1	0	0	0	0

4. System Model [13]

The system model of this paper is divided into two steps: learning the document and classifying the document's input. The first step, learning the document, involves sorting the ham and spam email document into the existing categories. Doing this generates the learning result of the document. This is done through the extracted feature points. The feature points are extracted by applying the previous techniques for extracting feature points in each document. This process is repeated to produce a learning document category for the ham and spam email documents. This category is used to classify the newly inputted document. The system model used to generate the learning document is shown in Figure 2.

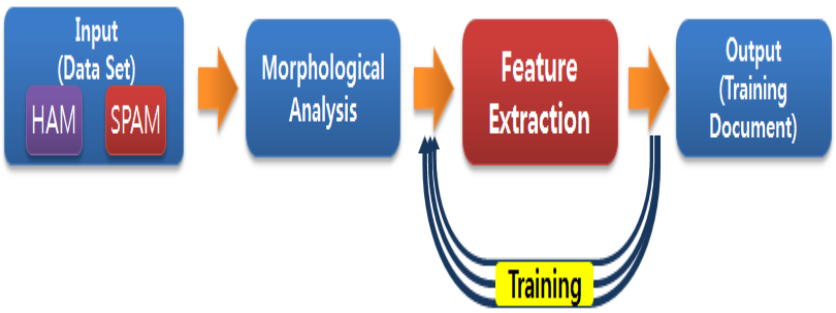

Figure 2. Document Learning Step of the System Model

While classifying the newly inputted document, the test email makes use of the learning document by using the ham email document and spam email document, both of which have already been learned by the system. The document input is classified, as the system utilizes the existing document learning of the ham and spam email documents. These documents are then compared and classified using the naïve Bayes classifier. The system model is used to classify the document, as shown in Figure 3.

Figure 3. The Newly Classified Document

4.1. Morphological Analysis and Feature Extraction

Prior to extraction of the feature points, the document is subjected to morphological analysis. The ham and spam email document classification occurs during pre-processing. The morphological process is performed using Python. This process divides the mail into morphological units and extracts feature points from the documents.

4.2. Document Classification

Document classification occurs one step before the morphological analysis that occurs during the pre-treatment process. This classification results in feature extraction. The naïve Bayes classifier is used for document classification. The classifier uses the Python programming language as well as Codebox [14]. This process was previously conducted for the purpose of classifying a document according to word frequency. However, in the present paper, the purpose of extracting the features of the relationships between the different words is to find a set of words associated with the frequency of a word. The performance of the classifier improves the calculation formula, as illustrated in equation 3.

$$P(C \mid F) = \frac{P(F) \times P(F \mid C) + P(AF)}{P(C)} \tag{3}$$

5. Experiments and Considerations

The second and following pages should begin 1.0 inch (2.54 cm) from the top edge. On all pages, the bottom margin should be 1-3/16 inches (2.86 cm) from the bottom edge of the page for 8.5 x 11-inch paper; for A4 paper, approximately 1-5/8 inches (4.13 cm) from the bottom edge of the page.

Table 3. Experimental Environment

Experimental environment	
OS	Window 7
CPU	Core(TM) I7-4790
RAM	8GB
Language	Python

Table 4. Experiment Results of TF-Based and Apriori-Based Schemes

Method / Classification.	Term frequency	Term frequency + Associated set of words
Normal classification Ham -> ham	**90**	**95**
Error classification Ham -> spam	**10**	**5**
Normal classification Spam -> spam	**93**	**97**
Error classification Spam -> ham	**7**	**3**
Success	**183**	**192**
Fail	**17**	**8**
Reliability	**91.5%**	**96%**

Out of 100 ham and 100 spam test mails, TF-based scheme succeeds to classify 183 mails correctly and fails to classify 17 mails. The reliability of TF-based scheme is 91.5 %. Whereas, our proposed scheme, Apriori-based scheme succeeds to classify 192 mails correctly and fails to classify 7 mails. The reliability of Apriori-based scheme is 96 %. Apriori-based scheme outperforms TF-based scheme by up to 5 %.

6. Conclusion

In this paper, feature extraction techniques and document classification techniques were proposed to increase the accuracy of the classification of ham and spam email documents using the naïve Bayes classifier. In previous studies, the simple method of classifying words as they appeared in the document was used to classify documents. In this paper, we proposed a more advanced technique in which feature points are first added to the related word set and given weights when a document is presented for classification. In the future, we will implement a direct system model that will differ from the existing document classifier. We will propose an improved method for identifying feature points during the extraction step. In addition to implementing a classifier to classify various documents using a support vector machine (SVM) [15], we will propose a document classifier to satisfy three goals in future research.

Acknowledgments

This work was supported by the Human Resource Training Program for Regional Innovation and Creativity through the Ministry of Education and National Research Foundation of Korea (2015H1C1A1035859).

References

[1] L. Ozgur, T. Gungor and F. Gurgen, "Spam Mail Detection Using Artificial Neural Network and Bayesian Filter", Intelligent Data Engineering and Automated Learning, vol. 3177, **(2004)**, pp. 505-510.
[2] I. Idris, "E-mail Spam Classification With Artificial Neural Network and Negative Selection Algorithm", International Journal of Computer Science & Communication Networks, vol. 1, no. 3, **(2014)**, pp. 227-231.
[3] E. S. You, G. H. Choi and S. H. Kim, "Study on Extraction of Keywords Using TF-IDF and Text Structure of Novels", Journal of The Korea Society of Computer and Information, vol. 20, no. 2, **(2015)**.
[4] S. J. Lee and H. J. Kim, "Keyword Extraction from News Corpus using Modified TF-IDF", The Journal of Society For e-Business Studies, vol. 14, no. 4, **(2009)**.
[5] W. S. Choi and S. B. Kim, "N-gram Feature Selection for Text Classification Based on Symmetrical Conditional Probability and TF-IDF", Journal of the Korean Institute of Industrial Engineers, vol. 41, no. 4, **(2015)**.
[6] H. J. Kim, J. J. Jung and G. S. Jo, "Spam-Mail Filtering System Using Weighted Baysian Calssifier", Journal of KIISE : Software and Applications, vol. 31, no. 8, **(2004)**.
[7] J. H. Yeon, J. H. Shim and S. G. Lee, "Modified Naïve Bayes Classifier for Categorizing Questions in Question-Answering Community", Journal of KIISE : Computing Practices and Letters, vol. 16, no. 1, **(2009)**.
[8] Y. Kim, "A Study on Design and Implementation of Personalized Information Recommendation System based on Apriori Algorithm", Journal of the Korean biblia, vol. 23, no. 4, **(2012)**.
[9] B. D. Gerardo and J. W. Lee, "Lossy Techniques Apriori Algorithm for Efficient Association Rule Mining", KSII Transactions on Internet and Information Systems, vol. 5, no. 1, **(2004)**.
[10] H. C. Kang, K. T. Yang, C. S. Kim, Y. J. Rhee and B. K. Lee, "A Time-based Apriori Algorithm", Journal of Electrical Engineering & Technology, vol. 59, no. 7, **(2010)**.
[11] L. Xing and S. Yueheng, "An Adaptive Spam Filter Based on Bayesian Model and Strong Features", World Automation Congress(WAC), **(2012)**.
[12] J. Y. Yoo, M. H. Lee, G. Aloyce and Do. M. Yang, "Creating a Naïve Bayes Document Classification Scheme Using an Apriori Algorithm", Advanced Science and Technology Letters (Current Research Trend of IT Convergence Technology IX), ASTL, vol. 133, **(2016)**.
[13] I. Pilaszy, "Text Categorization and Support Vector Machines", in Proceedings of the Sirth International Symposium of Hungarian Researchers on Computational Intelligence, **(2005)**.
[14] Enron Email Dataset, Available: http://www.aueb.gr/users/ion/data/enron-spa
[15] Codebox, Naïve Bayesian Classifier, Available: http://github.com/codebox/bayesian-classifier

Authors

Jong-Yeol Yoo, gum10011@naver.com, B. S. Information & Communication Engineering, DJU, 2015, M. S. Information & Communication Engineering, DJU, 2015~

Min-Ho Lee, biy006@naver.com, B. S. Information & Communication Engineering, DJU, 2015, M. S. Information & Communication Engineering, DJU, 2015~

Grace Aloyce, g.aloyce@gmail.com, B. S. in Computer Science & Engineering, St Joseph University in Tanzania, 2008, M. S. Information & Communication Engineering, DJU, 2015~

Dong-Min Yang, dmyang@dju.ac.kr, B. S. in Computer Science & Engineering, POSTECH, 2000, M. S. in Computer Science & Engineering, POSTECH, 2003, Ph.D. in Computer Science & Engineering, POSTECH, 2011

