ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/331988179

Feature Extraction based Text Classification using K-Nearest Neighbor Algorithm

Article · December 2018

CITATIONS
13

READS
1,838

4 authors:

Muhammad Azam
The Women University Multan
27 PUBLICATIONS **276** CITATIONS
SEE PROFILE

Tanvir Ahmed
American International University-Bangladesh
17 PUBLICATIONS **68** CITATIONS
SEE PROFILE

Fahad Sabah
Superior University
10 PUBLICATIONS **29** CITATIONS
SEE PROFILE

Muhammad Iftikhar Hussain
Beijing University of Technology
10 PUBLICATIONS **121** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Information Security View project

Multi-Cloud Security View project

All content following this page was uploaded by Muhammad Iftikhar Hussain on 26 March 2019.

The user has requested enhancement of the downloaded file.

Feature Extraction based Text Classification using K-Nearest Neighbor Algorithm

Muhammad Azam[1], Tanvir Ahmed[1], Fahad Sabah[1], Muhammad Iftikhar Hussain[2]

[1]Department of Computer Science and Information Technology, Superior University Lahore, Lahore, Pakistan
[2]Faculty of Information Technology & Beijing Engineering Research Center for IoT Software and Systems, Beijing University of Technology, Beijing 100124, China

Abstract
Scientific publications has been increasing enormously, with this increase classification of scientific publications is becoming challenging task. The core objective of this research is to analyze the performance of classification algorithms using Scopus dataset. In text classification, classification and feature extraction from the document using extracted features are the major issues for decreasing the performances in different algorithms. In this paper, performances of classification algorithms such as Naïve Bayes (NB) and K-Nearest Neighbor (K-NN) shown better improvement using Bayesian boost and bagging. The performance results were analyzed through selected classification algorithms over 10K documents from Scopus examined using F-measure and produced comparison matrices to estimate accuracy, precision and recall using NB and KNN classifier. Further, data preprocessing and cleaning steps are induced on the selected dataset and class imbalance issues are analyzed to increase the performance of text classification algorithms. Experimental results showed performances over 7% using K-NN and revealed better as compared to NB.

Key words:
K-NN, naïve bayes, text classification, rapid miner, feature extraction

1. Introduction

In past decades, number of scientific publications has been extremely increased and this growth requires an effective organization and categorization of these documents. According to the data taken from SCImago journals, number of publication become excessive (more than one million) right from the beginning as shown in Figure 1. It can be inferred form the Figure 1, number of publications have been increased every successive year and raised to 3.0 million for the year of 2015, which is large digit. However, scientists are engaged to muddle these exponentially increasing numbers of scholarly articles to find selected articles. Classification algorithms can help scientists with this task, wherein, various algorithms are incorporated in different scenarios for classification such as music, movies and products. Since, performance of these algorithms is varied on dataset containing scientific publications like Scopus. Additionally, library of medicine contains more than 25K vocabulary terms to represent research areas. Whereas, there are around 400 types and sub-types of music.

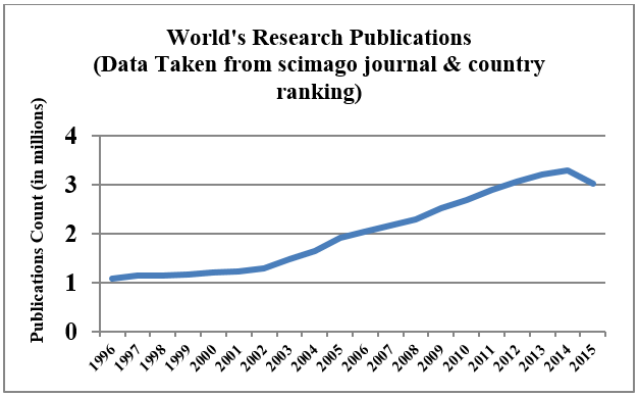

Fig. 1 World's Publications from 1996 to 2015

In this paper, we analyzed and concluded the performance of K-Nearest Neighbor (K-NN) and Naïve Bayes (NB) over 10K documents from Scopus. All labeled documents based on source journal category are used to cross validate the categories which are produced using K-NN & NB. Scopus Dataset contain following five categories; Medicine(all), Mathematics(all), Finance, Agricultural & Biological Sciences(all) and Engineering(all). While, training and testing the documents we also produced confusion matrices to measure and compare the accuracy, precision and recall of these algorithms.

Rest of the paper is organized as follows: Literature review is discussed in next section to elaborate the background and recently proposed algorithms of text classification. Section 3 provides description of dataset, clustering algorithms and evaluation parameters of clustering algorithms are briefly explained. Final section is devoted to discuss experimental results and comparison performances of various text classification algorithms.

Manuscript received December 5, 2018
Manuscript revised December 20, 2018

2. Literature Review

Traditional techniques concentrated on measuring the similarity between two documents based on the co-occurrences of words. Gongde Guo et al [1] presented a study of two extensively used techniques KNN & Rocchio classifier for text categorization using similarity based learning framework. To overcome the shortcomings, they developed a new approach known as K-NN model-based algorithm. Bin Othman et al [2] evaluated and explored Weka based five classifier on breast cancer data. Bayes network classifiers showed best accuracy of 89.71%. Ashmeet Singh and R Sathyaraj [3] proposed that small datasets showed more suitable and accurate results in case of NB. While, on large datasets Decision Tree (DT) showed more suitable and better values of accuracy, recall and precision in Rapid Miner. For achieving high accuracy, precision and recall values, the classification algorithm depends on the features using health care data [4]. S.L. Ting et al [5] stated that NB showed best values of accuracy and computational efficiency in document classification than DT and Support Vector Machine (SVM) classifiers. On the bases of simplicity and efficiency, K-NN was generally used test classifier. Moreover, some issues regarding inductive biases and model misfit also presented on distrusted trained data sets using K-NN [6]. Algorithms are performed on larger datasets required to test and measure the result of different classifiers in particular conditions for better performances [7]. R. E. S. Singer et al [8] presented an approach called BootTexter for text categorization. The results compared the efficiency of BoosTexter with other different classifiers on variety of tasks [8]. Recently some machine learning algorithms also used for text categorization. AdaBoost showed good results when applied on real text datasets. Most of the boosting algorithms used binary value for text classification. NB allows boosting techniques to use frequency values for the improvement of accuracy, the proposed method obtained significant improvement [9]. In K-NN classifier, features space selection used training dataset and value of k can enormously affect the classification accuracy, therefore, it is modifiable. Therefore, provided improved code for further enhancement of efficiency and accuracy [10]. Vaibhav C. Gandhi and Jignesh A. Prajapati [11] investigated the problem of classifying text documents automatically into categories, which relies on standard machine learning algorithms based on a set of training examples. It can learn a classification rule to categorize new text documents automatically. Comparative experiments of algorithms not fairly conducted. Since, algorithms such as NB or K-NN classifier produced better results than SVM [12]. Weka presented an empirical results on three text categorization approches (NB, SVM and C4.5) on two datasets (Diabetes and Calories) by training the dataset instances. The results compared based on the recall and precision values, where each of the algorithm was returning and presented percentage split of the dataset into training set and test set [11]. Muhammad Bilal et al [13] also used Weka based knowledge analysis on language sets rather English like Roman Urdu extracted from a blog. Further, they emphasized algorithms such as KNN, NB and dDT. Results shown that NB performed better against KNN & DT in metric like precision, accuracy, recall and F-measure. Arrhythmia signals coefficient extracted using Principal component analysis, linear discriminant analysis and weighted K-NN were applied to control the weighted signal and sensitivity relied on the size of K-NN to improve accuracy [14]. Feature based weighted K-NN applied on fifteen datasets of UCI machine. Granger causality and analytical hierarchy process (AHP) were applied as weighted feature to improve the performance and accuracy of K-NN classification using trained set, while, AHP applied as weight for different features [15]. Prototype selection (PS) algorithms used with K-NN in an experimental study framework, where PS conditions are tested in classical and realistic manner on given data set to handle the non-realistic and distributed nature using PS algorithm. PS algorithms includes techniques such as Considering Nearest Neighbor (CNN), Editing Nearest Neighbor, Repeated editing considering Nearest Neighbor (RCNN), Fast considering Nearest Neighbor(FCNN), Further Neighbor (FN) and Detrimental reduction optimization Procedure (DROP3) [16]. Weighted KNN were applied with the loss function on video data set categorized by Locally-Sensitive Discriminant Sparse Representation (LSDSR). Video semantic concept was adopted by the integration of error and representation of separated semantics. Results shown that proposed method enhance the accuracy and discrimination of video semantic concept [17]. A hybrid KNN and SVM model proposed to improve performance of similar letters of given number plate dataset. Results shown that the performance and accuracy of this hybrid K-NN-SVM model was improved by 3 percent [18]. Kernel K-NN algorithms were proposed to analyze road traffic statistics by using regional traffic attractors to achieve high accuracy. Results shown that multi-dimensional and multi-granularity local traffic merged into high-dimensional traffic based on kernel function, thereby, result achieved form Kernel KNN approach were more accurate and stable [19].

3. Methodology

Performance of classification algorithms on Scopus datasets are evaluated using rapid miner. Description of dataset and text preprocessing steps are as follows.

3.1 Dataset

The Scopus data used in this paper is deemed for bibliographic database comparison of abstracts and citations of academic journal articles. The dataset is owned and maintained by the Elsevier and is available online through subscription. It contains around 22k titles from more than 5k publishers, of 20k journals that are written and reviewed by different experts in the field of scientific, technical, medical, and social sciences. For analysis, we extracted the abstracts of 10K documents from Scopus with their journals categories. The preprocessing tasks involved such as preprocess of data, training the data and tested different classification algorithms using rapid miner. Words vectors are created from abstracts for analysis and considered categories as special/label attribute.

3.2 Rapid Miner

Various data mining tools exist, which helps in processing different types of datasets and produced the summarized results to take appropriate decisions. Rapid Miner is an open source tool it offers integrated working platform for Machine learning analysis with multiple extensions for data analysis.

3.3 Classification Algorithms

Different classification algorithms are taken under consideration for the performance comparison and induced in this proposed study are Naïve Bayesis (NB). NB is supervised learning algorithm and statistical method of classification. It is a probabilistic model, which allows solving analytical and foretelling problems.

$$P\left(\frac{h}{D}\right) \quad (1)$$

$$= P\left(\frac{D}{h}\right) P(h) P(D) \quad (2)$$

where, $P(h)$ is the prior probability of hypothesis h. $P(D)$ is prior probability of training data D. $P\left(\frac{h}{D}\right)$ is the probability of h given D. $P(D/h)$ is probability of D given h. In this relation, NB classifier considered as in the presence/absence of specific attribute as unrelated to any other feature. For example, a fruit may be an “apple” if it is red, round and has a diameter of 4 inches. If these features are present in some other fruits, the NB classifier consider all these as an “apple” depend on the specific predefined feature and attributes. The advantage of the NB classifier is to require small values of trained dataset to guess the means and alterations of required variables for classifications. Since, only the alternation of each variables for each label has to be determined where independent variables can be assumed, thus, no need to determine the entire covariance matrix. Labeled dataset is given as input to NB with particular features. NB classifier can be applied on unseen datasets for the predication of labels. On the other hand, K-NN algorithm compare the given test examples with similar training example known as learning by analogy. Training examples are denoted by “n” where each attribute presented by n-dimensions to store all training examples. When we test an unknown example then K-NN algorithms search K “nearest neighbor” with respect to given example. “Closeness” defined in terms of Euclidean distance metric. Basic K-NN algorithm is consist of two stages:

i. Find “k” training examples closet to the given set example.
ii. Take the most frequently matching features for k examples (or take average of these k values in case of regression)

3.4 Bagging

Bootstrap aggregating (bagging) is a machine learning ensemble meta-algorithm to improve classification and regression models in terms of stability and classification accuracy”. Variance and over-filtering also reduced by using this technique. Learner is need as a sub process to generate a model form given dataset. A better model can be produced by learners provided in some process by applying various operators.

3.5 **Boosting**

Boosting based on Bayes' theorem, a meta-algorithm to be used in conjunction with various learning algorithms to improve the performance and to train Boolean target attributes. In each iteration of trained ensemble training set is reweighted and prior trained sets are “sampled out”. The inner classifier is based on DT algorithms which can be applied as a series of steps and combine each model as a global model. The number of models depend on the trained iteration parameters.

3.6 Text Preprocessing

For text classification the preprocessing is very important. Text preprocessing takes most of the time in text classification it consists of following steps:

3.6.1 Vectors Creation

In this process word vectors are generated from a text. By word vectors we mean that document tokens are used to generate a vector which numerically denote the document. Usually the word vector is created by TF-IDF.
Different prune methods are also used in vectors creation which states that for the building of word list of specific frequency to frequent words should be ignored. Like, prune

below percent denotes that are less than percentage of all document are ignored and prune above as vice versa.

3.6.2 Filter Tokens

Different techniques are in use to further process the document for preparation to apply the model or classify text e.g. Filter Tokens (by Length, by Content, by Region) etc. Tokenization process divides documents text into tokens sequence. There are several options how to specify the splitting points. Either you may use all non-letter character. The resultant tokens contain one word which is suitable to build a final Word vector. In case if we want to build a complete windows of tokens then we must have to split at least one complete sentence. This is possible by setting the split mode to specify character and enter all splitting characters. You can define regular expression more elastic for some particular cases in third option where non-letter charterers are used as separators.

3.6.3 Stemming

The process terms are reduced to a base form using an external file with replacement rules. The file must contain a rule per line: target Expression: patter1 patter2 ... where target Expression is the term to which the input terms are reduced, if it matches any of the patterns. patterX is a simple string or a regular expression as described in appendix A. A simple example would be a mapping like: weekday: .*day Please keep in mind, that very short words are filtered out in the default setting of the TextInput operators.

3.6.4 Stop Words Elimination

English stopword list is created and if the value of tested stopword form a document is equal to the stopword provided in the list then its token will be removed. Please note that, for this operation to work properly, every token should represent a single English word only. To obtain a document with each token representing a single word, you may tokenize a document by applying the Tokenization beforehand.

3.7 Evaluation Measures

Accuracy, precision and recall are the three important measures, which are used to decide the quality performance of the classification algorithm. Correctly predicted values belong to precision class, actual predicted values related to class recall, while, overall predictions referred as accuracy. Average values of each precision and recall class are taken to generate overall classifier. Rapid Miner tool is used to calculate accuracies of the classifier by the factor like true positive, false positive, f-measures, precision and recall values.

3.7.1 Accuracy

Accuracy is calculated as number of instances predicted positively divided by Total number of instances. Percentage of the accurate predicted values among the all values. We take the values of accuracy from 0 to 100. In an expression accuracy can be denoted as.

$$\text{Accuracy} = ((\text{True Positive} + \text{True Negative}) / (P + N)) * 100 \quad (3)$$

3.7.2 Precision

Precision is a positively predicted value. It is an instance which has class x / total classified. Accurate result can be obtained from high precision values. In other words no of related chosen items.

$$\text{Precision} = (\text{True Positive} / (\text{True Positive} + \text{False Positive})) * 100 \quad (4)$$

3.7.3 Recall

Sensitivity of the problem can be determined by recall which present the quality and completeness of the product. In a simple way recall is the most related part of the given set which is relevant to the result of that particular query or the no of chosen related objects.

$$\text{Recall} = (\text{True Positive} / (\text{True Positive} + \text{False Negative})) * 100 \quad (5)$$

3.7.4 True Positive (TP)

Correctly labeled values by any classifier known as true positive. Module projection of positively and specified resulted can be calculated through true positive.

$$\text{True Positive rate} = (\text{True Positive} / (\text{True Positive} + \text{False Negative})) * 100 \quad (6)$$

3.7.5 False Positive (FP)

In correct values classified by class x / total class, except x. incorrectly predicted values compare to the original resultant.

$$\text{False Positive rate} = \big(\text{False Positive} / (\text{False Positive} + \text{True Negative})\big) * 100 \quad (7)$$

3.7.6 F-measure

F-measure is calculated as

$$\left(2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}\right) * 100 \quad (8)$$

4. Results and Discussions

Scopus dataset consist of an attribute, which was taken as label/category aspect used for classification of give sets. Using rapid miner, Scopus (10K instances) dataset was applied to classifiers NB and K-NN with bagging and boosting. The rapid miner was implied to classify the testing data using X-validation and introduced best classifier based on precision value. Tables 7 presented with results of both algorithms on the Scopus dataset with and without bagging and boosting. Figure 2 shows the main process diagram in which blocks/operators used in this process. Figure 3 represents the sub processes for the process documents operator. Operators used in Figure 2 and Figure 3 are elaborated in the section 3. The sub-process of validation operator contains the classifier with model operators and their results.

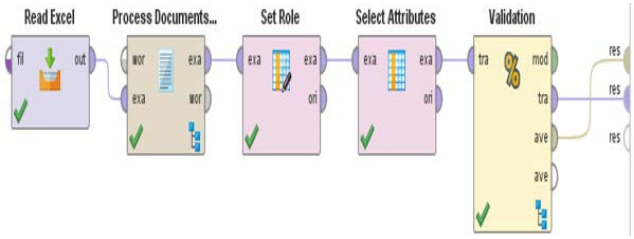

Fig. 2 Process diagram for training and validation of classifiers

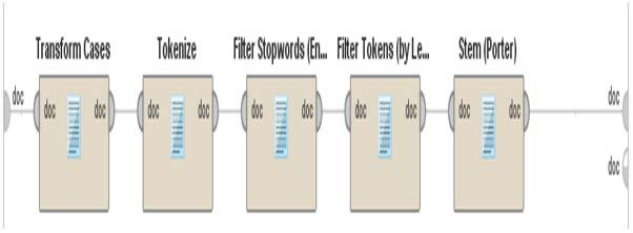

Fig. 3 Sub-processes/Nested Operations for Preprocessing Documents

Overall accuracy of Naïve Bayes is 71.11%. Table 1 shows the precision and recall using NB classifier for each selected category. It can be seen that Mathematics (all) conceived with the highest precision approximately 88%, whereas, Engineering (all) achieved the highest recall approximately 80%. However, Medicine (all) achieved with lowest precision and recall of 61.29% and 63.33%.

Table 1: Precision and Recall of Naïve Bayes

	class precision	class recall
Medicine(all)	61.29%	63.33%
Mathematics(all)	88.00%	73.33%
Finance	87.50%	70.00%
Agricultural & Biological Sciences (all)	61.54%	66.67%
Engineering (all)	78.57%	80.00%

Overall accuracy of k-NN (where k=5) is approximately 78.67%. Table 2 shows the precision and recall of k-NN classifier for each selected category. Engineering (all) achieved the highest precision of 85.19%, whereas, both the Mathematics (all) and Finance achieved the highest recall which is about 90%. Medicine (all) conceived lowest precision and recall approximately of 68.97% and 66.67%, respectively. Overall accuracy was achieved better using k-NN as comparison to Naïve Bayes classifier.

Table 2: Precision and Recall of k-NN for k=5

	class precision	class recall
Medicine(all)	68.97%	66.67%
Mathematics(all)	79.41%	90.00%
Finance	77.14%	90.00%
Agricultural & Biological Sciences (all)	84.00%	70.00%
Engineering (all)	85.19%	76.67%

Bayesian Boost (Naïve Bayes): Overall accuracy of Bayesian Boost with Naïve Bayes is approximately 76%. Table 3 shows the precision and recall for each selected category using nested classifier known as Bayesian Boost and Naïve Bayes. It can be inferred that Mathematics (all) achieved the 100% precision, whereas. both the Mathematics (all) and Agricultural & Biological Sciences (all) achieved the highest recall of approximately 80%. Medicine (all) with lowest precision and Engineering (all) achieved lowest recall are approximately 60.53% and 66.67%, respectively. Overall accuracy was increased using Bayesian Boost.

Table 3: Precision and Recall of Naïve Bayes with Bayesian Boost

	class precision	class recall
Medicine(all)	60.53%	76.67%
Mathematics(all)	100.00%	80.00%
Finance	88.46%	76.67%
Agricultural & Biological Sciences (all)	63.16%	80.00%
Engineering (all)	83.33%	66.67%

Bayesian Boost (k-NN): Overall accuracy of Bayesian Boost with k-NN is approximately 80%. Table 4 shows the precision and recall for each selected category using Bayesian Boost with k-NN as nested classifier. It can be

seen that Engineering (all) conceived 88.46% precision, whereas, Mathematics (all) achieved the highest recall which is approximately 93.33%. Medicine (all) achieved lowest precision and recall are approximately 68.97% and 66.67%, respectively. Overall accuracy was increased using Bayesian Boost.

Table 4. Precision and Recall of Naïve Bayes with Bayesian Boost

	class precision	class recall
Medicine(all)	68.97%	66.67%
Mathematics(all)	82.35%	93.33%
Finance	77.14%	90.00%
Agricultural & Biological Sciences (all)	84.62%	73.33%
Engineering (all)	88.46%	76.67%

Bagging (k-NN): Overall accuracy of bagging using k-NN is 78.67%. Table 5 shows the precision and recall for each selected category using Bagging with k-NN as nested classifier. It can be seen that Agricultural & Biological Sciences (all) conceived 84.62% precision, whereas, Finance achieved the highest recall of 93.33%. Medicine (all) achieved lowest precision and recall about 72% and 60%, respectively.

Table 5. Precision and Recall of k-NN with Bagging

	class precision	class recall
Medicine(all)	72.00%	60.00%
Mathematics(all)	83.87%	86.67%
Finance	75.68%	93.33%
Agricultural & Biological Sciences (all)	84.62%	73.33%
Engineering (all)	77.42%	80.00%

Bagging (Bayes): Overall accuracy of Bagging using Naïve Bayes is approximately 75.33%. Table 6 shows the precision and recall for each selected category using Bagging with Naïve Bayes as nested classifier. It can be seen that Mathematics (all) achieved the highest precision about 100%, whereas, Mathematics (all) and Agricultural & Biological Sciences (all) achieved the highest recall approximately 80%. Medicine (all) achieved lowest precision about 60.53% and Engineering (all) conceived the lowest recall approximately 63.33%. Table 7 shows the overall Accuracy and F-Score of the classifiers under consideration. It achieved accuracy using k-NN, which is better than Naïve Bayes on Scopus dataset. Accuracies with the inclusion of Boosting and Bagging.

Table 6. Precision and Recall of Naïve Bayes with Bagging

	class precision	class recall
Medicine(all)	60.53%	76.67%
Mathematics(all)	100.00%	80.00%
Finance	88.46%	76.67%
Agricultural & Biological Sciences (all)	61.54%	80.00%
Engineering (all)	82.61%	63.33%

Table 7. Overall Accuracy and F-Score of Classifiers

Classifier	Accuracy	F-Score
Naïve Bayes	71.11	72.95
k-NN	78.67	78.81
Bayesian Boost (Naïve Bayes)	76.00	77.52
Bayesian Boost (k-NN)	80.00	80.15
Bagging (Naïve Bayes)	76.95	76.95
Bagging (k-NN)	78.69	78.69

5. Conclusion

To our best knowledge, we evaluated and investigated the performance of selected data mining classifiers on a particular dataset (Scopus) with five different categories such as Medicine (all), Mathematics(all), Finance, Agricultural & Biological Sciences (all) and Engineering (all). In this paper, K-NN results produced better performance than Naïve Bayes classifier. Further, boosting and Bagging remarkably increased the overall accuracy using Naïve Bayes Classifier, whereas, performance of k-NN remained unchanged and better than Naïve Bayes classifier. Additionally, we shown that Bagging and Boosting brought an impact in the performances using k-NN and concluded that k-NN with Bagging and Boosting revealed better classifier for classifying the scientific publications.

References

[1] G. Guo, H. Wang, D. Bell, Y. Bi, K. Greer, Using kNN model for automatic text categorization, Soft Computing, 10 (2006) 423–430.

[2] M. -f. -b. Othman, T. -m. -s. Yau, Comparison of Different Classification Techniques Using WEKA for Breast Cancer, in: proc. 2006 3rd Kuala Lumpur International Conference on Biomedical Engineering, 2006.

[3] Singh, R. Sathyaraj, A Comparison Between Classification Algorithms on Different Datasets Methodologies using Rapidminer, International Journal of Advanced Research in Computer and Communication Engineering, 5-5 (2016).

[4] P. -r. Harper, A review and comparison of classification algorithms for medical decision making, Health Policy, 71-3 (2005) 315-331.

[5] S. –l. Ting, W. -h. Ip, A. -h. -c. Tsang, Is Naïve Bayes a Good Classifier for Document Classification?, International Journal of Software Engineering and Its Applications, 5-3 (2011) .

[6] S. Tan, An effective refinement strategy for KNN text classifier, Expert Systems with Applications, 30 (2006) 290–298.
[7] G. -k. -m, Nookala, B. -k. Pottumuthu, N. Orsu, S. -b. Mudunuri, Performance Analysis and Evaluation of Different Data Mining Algorithms used for Cancer Classification, International Journal of Advanced Research in Artificial Intelligence, 2-5 (2013).
[8] R. -e. SCHAPIRE, Y. SINGER, BoosTexter: A Boosting-based System for Text Categorization, Machine Learning, 39, 135–168 (2000).
[9] Y. -h. Kim, S. -y. Hahn, B. -t. Zhang, Text filtering by boosting naive Bayes classifiers, in: Pro. SIGIR '00 23rd annual international ACM SIGIR conference on Research and development in information retrieval, Athens, Greece - July 24 - 28 (2000) 168-175.
[10] A.-k. Nikhath, K.Subrahmanyam, R.Vasavi, Building a K-Nearest Neighbor Classifier for Text Categorization, International Journal of Computer Science and Information Technologies, 7-1 (2016) 254-256.
[11] M. Trivedi, S. Sharma, N. Soni, S. Nair, Comparison of Text Classification Algorithms, International Journal of Engineering Research and Technology, 4-2(2015).
[12] V. -c. Gandhi, J. -a. Prajapati, Review on Comparison between Text Classification Algorithms," International Journal of Emerging Trends & Technology in Computer Science, 1-3 (2012).
[13] M. Bilal, H. Israr, M. Shahid, A. Khan, Sentiment classification of Roman-Urdu opinions, Journal of King Saud University –Computer and Information Sciences, 28 (2016), 330–344.
[14] W.-h. Jung, S.-g. Lee, An Arrhythmia Classification Method in Utilizing the Weighted KNN and the Fitness Rule," IRBM Elsevier Masson France, 38 (2017) 138–148.
[15] G. Bhattacharya, K. Ghoshb, A. -s. Chowdhury, Granger Causality Driven AHP for Feature Weighted kNN, Pattern Recognition, 66 (2017) 425–436.
[16] J. -j. Valero-Mas, J.Calvo-Zaragoza, J. -r. Rico-Juan, "On the suitability of Proto type Selection methods for kNN classification with distributeddata, Neurocomputing, 203 (2016) 150–160.
[17] Y. Zhan, J. Liu, J. Gou, M. Wang,A video semantic detection method based on locality-sensitive discriminant sparse representation and weighted KNN, J. Vis. Commun. Image R. 41 (2016) 65–73.
[18] S. -s. Tabrizia, N. Cavusa, A hybrid KNN-SVM model for Iranian license plate recognition," in: Pro. Computer Science, 12th International Conference on Application of Fuzzy Systems and Soft Computing, ICAFS 2016, Vienna, Austria, 29-30 August (2016).
[19] X. Dong-wei, W. Yong-dong, J. Li-min, L. Hai-jian, Z. Gui-jun, Real-time road traffic states measurement based on Kernel-KNN matching of regional traffic attractors, Measurement, 94 (2016) 862–872.

View publication stats