AN IMPROVED NAIVE BAYES WITH KERNEL DENSITY ESTIMATION FOR OPINION MINING

S RAJA RAJESWARI
Department of Computer Applications,
Fatima College (Autonomous),
Madurai, India.
rraji_raji01@yahoo.co.in

A JOHN SANJEEV KUMAR
Department of Computer Applications,
Thiagarajar College of Engineering,
Madurai, India.
ajscse@tce.edu

Abstract

With the rapid growth of social media in recent years, opinion mining has gained much attention. The polarity classification is the task of opinion mining that is used to take the decision based on online customer reviews and survey responses. The function of polarity classification is to classify the text document into positive or negative, at the feature level. In this paper, we propose the Naive Bayes classifier with Kernel Density Estimation (NB-KDE) function for polarity classification. The NB-KDE is a non-parametric classifier that computes the probability density function based on the kernel estimator. The NB-KDE is a supervised classifier that considers the independent behavior of the feature to train the classifier. In the finite data sample, the proposed classifier performs data smoothing by the inference of the population. Stemming and stop word removal are the pre-processing techniques which are used to produce reliable results.

Key words: Opinion Mining, Naïve Bayes, Kernel Density Estimation

1 Introduction

For the past several decades, opinion mining has gained much attention. Opinion mining uses text analysis and Natural Language Processing to identify the customer's opinions by contextual polarity [2], [4], [8], [10]. Social media, forums and blogs are the main sources of opinion reviews which represent the opinion in the way of movie reviews [8], [16], product surveys [8], [4], [11], news [13], and health care data. The opinion mining helps to improve the business by taking good decisions in launching a new product, making product improvements [12], based on the popularity of the product among the customers.

The role of opinion mining is to classify the polarity of a given text document into positive or negative, in the feature level [1]. The polarity classification task can be carried out in two different categories. The first technique classifies the documents into positive or negative with predefined keywords called lexicons [21], [15]. In this technique, the corpora and dictionaries are used to classify the documents. Second, the machine learning technique is used to perform text mining with features of sentiment words [9], [17] – [22]. Currently, the text classification is carried out by parametric methods like Hybrid of random forest and Support Vector Machine (SVM) [2], and Ensemble methods. In this paper, the Naive Bayes with Kernel Density Estimation (NB-KDE) is proposed to classify the opinion in a non-parametric way. The NB-KDE does not require any assumption about the distribution of the population. The Naive Bayes classifier is a simple probabilistic classifier that assumes the independent behaviour of a feature in the given class variable. The non-parametric NB-KDE classifier reduces the complexity to train the classifier. The non-parametric classifier computes the conditional probability with the unknown density of the features. This property can help to obtain better accuracy and reduces the noise. The pre-processing techniques are used to improve the efficiency of the classifier. In the pre-processing stage, the stemmer is used to reduce the word to its root. The stop word removal technique is used to remove the substring of the document which does not give any information about the document. Tokenization is also the pre-processing technique which divides the strings into the number of tokens like words, symbols, and phrases.

2 **Literature Review**

This section deals with the description of previous work for opinion mining. Opinion mining has to identify the polarity by the predefined words that express the sentiment. The process of opinion mining has to be carried out in different levels like document level, sentence level and feature level [1], [15], [22]. In document-level, the whole document is considered to be a single unit to classify the entire document as positive or negative [5], [8], [10], [17]. In the sentence level, each sentence is used to identify the positive or negative opinion [4]. The opinion mining has to be carried out in three different categories; supervised [1], [10], [17], unsupervised [5], [6], [14] and semi-supervised [11], [19].In view of supervised classification, Muhammad et al. describe the sentiment classification of Roman language [20]. The pre-processing method is carried out by the StringToWordVector filter in WEKA environment to transform the strings into word tokens. The StringToWordVector filter includes the methods TF-IDF (Term Frequency-Inverse Document Frequency), Output Words Count, Tokenizer, Words to keep, Numeric to binary filter and so on. The classification task is carried out by three techniques namely Naïve Bayesian, Decision Tree, and K-Nearest Neighbor. Test results show that the Naïve Bayes outperformed in terms of accuracy, recall, and F-measure.

Meanwhile, Fersini et al. suggest the ensemble based Bayesian learning [8] to reduce the noise related to the language and provides results that are more accurate. This model estimates the uncertainty and reliability based on the ensemble of the Bayesian Model Averaging. The greedy approach is used to evaluate the performance of the classifier. Mangi et al. have proposed the Hidden Markov Model (TextHMM) [16] that uses the sequence of words for text classification. The co-occurrence of the words is the semantic clusters that represent the hidden variables. This method gained the advantage of implicit opinion. On the other hand, Jayashri and Mayura proposed the sentiment classification in a supervised way [10] by the Naïve Bayes, Maximum Entropy (ME) and Support Vector Machine (SVM) classifiers. The Naïve Bayes classifier calculates the conditional independence value of the feature vector. The Naïve Bayes classifier performed well with highly dependent features. The Maximum Entropy uses the normalization function over the feature vector, and it performed well with conditional independence function. It does not consider the assumption about the relationship between the features. In the case of SVM, it uses the maximum margin hyperplane to divide the features into different classes.

To improve the classifier's accuracy, Aritz et al. suggested the kernel based Bayesian network paradigm [3] for supervised classifications. This method estimates the true density of continuous variables by discretization of the variables using kernels. The kernel function is used to calculate the mutual information between the pair of continuous variables. Instead of using the Bayesian network, the kernel-based Naïve Bayes is used here for document classifications. Liu et al. address the issue of overfitting by kernel density estimation [13] method by training the auxiliary training samples. The kernel function is also used to calculate the joint distribution of the response and predictors with the product form of the unknown mixing measure that was proposed by Abhishek et al. In this system, the Bayesian method [1] computes the posterior value of the feature. To handle the opinion mining efficiently, the proposed system uses Naïve Bayes with kernel density to classify the feature sets. It works well in a large number of datasets.

3 **Proposed Model**

The proposed method has the following steps: The pre-processing takes part in the first step. To compute the conditional probability of the given text the Naïve Bayes with the Kernel Density Estimator (NB-KDE) is used as a density estimation technique. The taxonomy of the proposed system is given in figure 1.

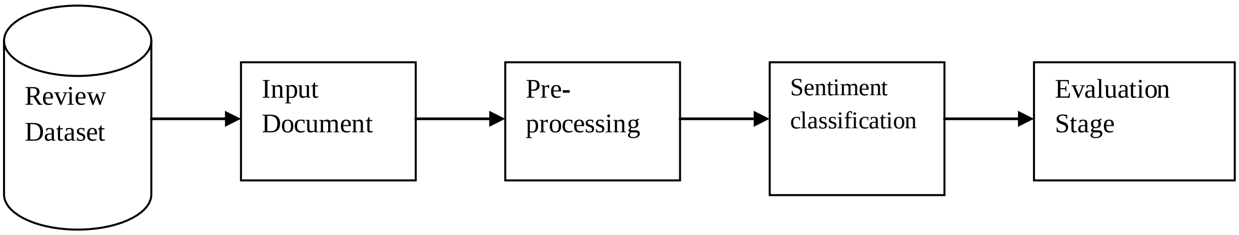


Figure 1 Taxonomy of the proposed system

3.1 Pre-processing techniques

The datasets are pre-processed by StringToWordVector [7] filter in WEKA. The filter converts the opinion (which is represented as documents) into a set of numeric attributes. The numeric attributes represent the word occurrence information. This filter creates a dictionary for each class to handle the tasks in a supervised manner.

3.1.1 Stop word handler

Stop words [7] are filtered out before processing the text. The stop words refer to the most common words that do not have any meaningful information. In English, there are nearly 500 stop words which are available (examples: of, as, a). In the proposed system, the Rainbow technique is used to remove the stop words.

3.1.2 Stemmer

Stemming [7] is the process of reducing the words into its root by dropping the unnecessary characters placed at the end of each word (example: occurrence – occur). The stemming process is a language-dependent task. In stemming, all words that have close root or stem are put together as one stem. For example, the words “fishing, fished, and fisher” have the stem “fish”. In the proposed system the snowball stemmer is used to truncate the words into their root.

3.1.3 Tokenizer

Tokenization [7] is the act of dividing the strings into a number of tokens like words, symbols, and phrases. It is the conventional method of text analysis to generate the basic unit of words (for example "isn't" to "is not"). Errors made in this stage lead to poor classifier accuracy. Therefore, the identification of basic units that cannot be decomposed into further unit is most important. In WEKA the WordTokenzier is used as a simple tokenizer which gets a stem of characters and gives the output as tokens. This tokenizer creates the tokens like [], !, &, * and numbers by reading the delimiters.

3.2 Classification

3.2.1Naive Bayes

Naïve Bayes [22] classifier assumes that the predictive attributes are conditionally independent in the given class. No attributes have hidden influence over other. It can assign the class label to a feature vector, which is drawn from a finite set. In view of this, let X be the vector of random 'n' variables which is represented by X= (x1,x2,......xn) and k be the possible outcomes of the class C. Then the conditional probability is computed by

$$p(C_k|X) = \frac{p(C_k)p(X|C_k)}{p(X)} \qquad (1)$$

In equation (1), the denominator cannot estimate the distribution directly without the help of the numerator. It does not depend on the class value C over the feature value X. Therefore, it can be considered as a constant. As the conditional independence of the attribute is taken into account, so the likelihood ($p(X|C_k)$) of the above equation can be written as

$$\begin{aligned} p(X|C_k) &= p(x_1|C_k)p(x_2|C_k)p(x_3|C_k)\ . . . \\ &= \prod_{i=1}^{n} p(x_i|C_k) \qquad (2) \end{aligned}$$

Using the above equations, Naïve Bayes computes the class conditional probability for the variables. Probability Density Function (PDF) is used to estimate the distribution of the variables. In general, the Naïve Bayes uses the Gaussian model to estimate the PDF but in the proposed model the KDE is used as the density function instead of the Gaussian method.

3.2.2 Kernel density estimation (KDE)

In KDE the density function is calculated in a non-parametric [6] way. Let x be the number of samples drawn from some distribution with an unknown density that is computed by the kernel density estimator. From equation (1) the KDE can be derived by using the Gaussian kernel as the density function,

$$p(X|C_k) = \frac{1}{n}\sum_i g\left(x, \mu_{i,} \sigma c\right) \qquad (3)$$

Where 'i' be the training set attribute of X in class C and $\mu_i = x_i$. The standard kernel density estimation equation be like

$$p(X|C_k) = \left(nh^{-1}\right)\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \qquad (4)$$

Therefore the equation (4) equals to the equation (3) where K= $g(x, 0,1)$ and h=σ is the kernel function, where h is a smoothing parameter called bandwidth. It must be non-negative and integrate to 1 (h→0, n→∞). The Gaussian kernel is used to find the kernel function k. The proposed algorithm is presented below

Algorithm 1 Naive Bayes with kernel density estimation (NB-KDE) algorithm

```
Input:  Unprocessed text review documents D={D};
Output: Classified text documents;
1.  for all text documents;
2.      Pre-process the document and get the feature vector as $x_{i;}$
3.      for all $x_i$ in class C;
4.         Calculate the Gaussian kernel function by equation (3); /* K=g(x,0,1) and h=σ, h→0,
                                                               n→∞.*/
5.         Substitute the value of the kernel function with the equation (1) to get the conditional
           probability;
6.  Using the conditional probability calculate the posterior value of the classifier with
    equation (1)
7.  Repeat step 4 for all $x_{i\,;}$  // To consider the conditional independence
8.       end for ;
9.  end for;
10.  According to the feature vector the documents were classified into positive or negative
```

3.2.3 Proposed Naive Bayes with kernel density estimation (NB-KDE) algorithm

In opinion mining, extracting the relevant attributes from the list of features is more important to get reliable results. In WEKA the StringToWordVector filter is used to pre-process the text documents D. In this filter, the tokenization is taking part to segment the words, symbols, numbers, and punctuations from the text review. Then the snowball stemmer is used to trim the words into its root word. The Rainbow stop word technique removes the meaningless words called stop words from the text reviews. After the pre-processing stage, the StringToWordVector filter produces the list of attributes that are used to train the classifier.

The classifier model computes the probability density of the attributes, which is produced by the StringToWordVector filter. The probability density is estimated by the kernel density estimation technique. In the proposed model, KDE uses the Gaussian technique to estimate the kernel function by equation (3). The kernel estimates the variance and mean value for each data point xi. Then the kernels are combined to estimate the density estimation by equation (4). In equation (4) h is the smoothing parameter which is used to remove the noise and retain the important pattern of the dataset. The KDE is computing the density of the entire observed attribute X in class C. Then the density is averaged for the large set of kernels. By using the probability density estimation, the Naive Bayes classifier computes the class conditional probability for each independent attribute. The attribute that has the maximum posterior value of the given class is assigned to the given class label. The NB-KDE is a simple and efficient model, which does not need prior assumption and it produces reliable results in categorical data.

4 Results and Discussion

This section describes the implementation and the comparison of the proposed model with other feature selection algorithms and classifiers.

4.1 Data set

Four discrete datasets such as IMDB movie reviews, home appliances [11], electronic items [11], and Book reviews [11] are used to evaluate the proposed model. The movie reviews are taken from the IMDB website. IMDB is the most authoritative source of internet movie database which consists of the information related to TV shows and movies. The movie reviews are represented in terms of rating or text reviews. The proposed system represents the reviews as documents and classifies it by non-parametric approach. These review documents are classified into positive and negative reviews. Some reviews express a person's opinion explicitly like "Good entertaining, time pass movie... slightly different than they the usual...worth watching for sure". In this review, the features or words represent the opinion explicitly, and it is classified as a positive review. In some cases, the reviews express the opinions implicitly. For an example "Who thought this would make a good film? None of the people working on the film ever had this thought? I had to walk out during the second half because of how absurd it was". In this review, the words indirectly represent the negative opinion for a movie. The proposed system takes the advantage of classifying the implicit opinion efficiently. In the non-parametric model, no prior assumptions are needed to classify the data. The kernel function estimates the probability density function from all data points. The kernel function takes bandwidth to smoothen the PDF. By using appropriate bandwidth, the kernel function produces accurate results.

Like movie review, the proposed system also efficiently classifies the home appliances product reviews, book reviews and the reviews about electronic items. These reviews are taken from the Amazon product review databases. To evaluate the proposed system in various aspects the number of reviews for classification varies from 100 to 2000. In 2000 review dataset, 1000 positive and 1000 negative reviews are taken to evaluate the model.

4.2 Result

The first step is to convert the 300 documents into a single .arff file by WEKA command line interface. Then the .arff file is loaded into WEKA explorer for pre-processing. The StringtoWordVector [7], [20] performs the pre-processing. This filter produces a list of feature vectors, which are represented by the vector matrix. These feature vectors are taken to train the classifier. The 10-fold cross-validation is used to validate the results of the classifier. In the given dataset, the batch size will be taken as 100.

4.2.1 Evaluation Measures

To build the classifier model one should prove that the classifier model is best among all the models. Evaluation metrics will help to prove the best of the classifier model. Based on the values of metrics, the necessary improvements are made with the classifier model until the desired accuracy had met. In the proposed system precision, recall, F-measure, accuracy [10] and ROC area are used to evaluate the classifier model. The accuracy is used to evaluate the classifier model by computing the ratio between the numbers of correct predictions with the total number of predictions. The accuracy is

$$Acc = \frac{t_{pos} + t_{neg}}{t_{pos} + t_{neg} + f_{pos} + f_{neg}} \qquad (5)$$

Where t_{pos}, is the true positive rate and t_{neg} is the true negative and f_{pos}, f_{neg} are the false positive and false negative. True positives are the ones that are predicted correctly as positive reviews. True negatives are the ones that are predicted correctly as negative reviews. The false positives represent the misclassified positive reviews and the false negatives also represent the misclassified negative reviews.

The precision value is used to get the relevant attributes. It denotes the ratio of the correctly predicted positive attributes from the total number of predicted positive attributes. It can be computed by

$$Pre = \frac{t_{pos}}{t_{pos} + f_{pos}} \qquad (6)$$

To measure the classifier completeness the recall value is computed as the ratio between the number of predicted positive attributes to the total number of positive class values. The recall is computed by

$$Recall = \frac{t_{pos}}{t_{pos} + f_{neg}} \qquad (7)$$

To compute the test accuracy of the data the F measure is used as the weighted harmonic mean of precision and recall. It tells us how many instances the model correctly classifies. The best value of F measure is at 1, and its worst value is at 0. To get results that are more reliable, the F measure is computed by equation (8)

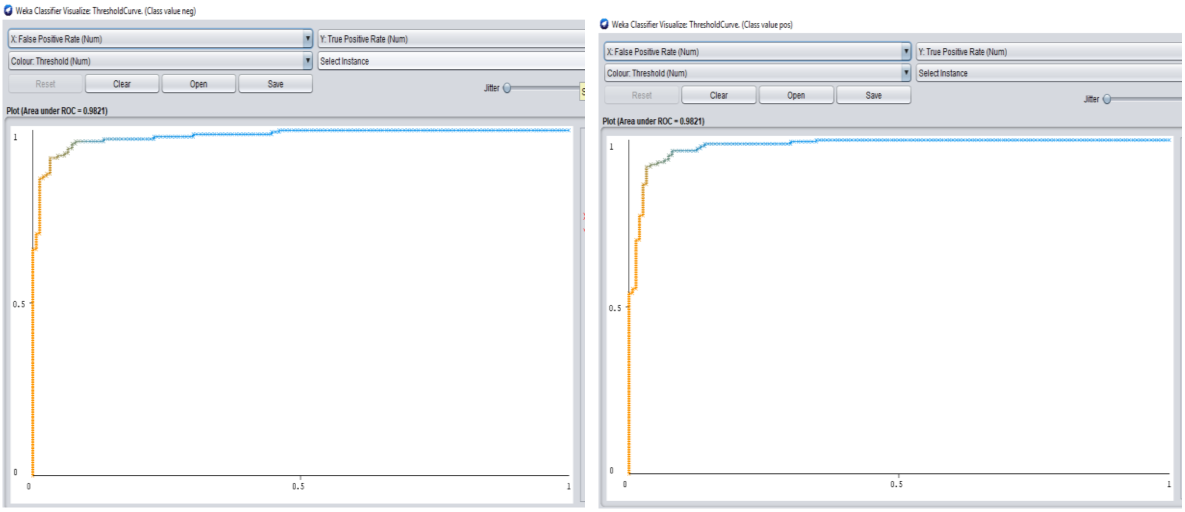
$$F_{measure} = 2 * \frac{Pre * Recall}{Pre + Recall} \qquad (8)$$

The ROC area computes the threshold value of the area under ROC (Receiver Operating Characteristic) curve for positive and negative reviews. The ROC curve is used to examine the performance of the classifier by visualizing the graph for the threshold value of the classifier. The ROC curve is used to select the optimal model, and it directly relates to the cost/benefit analysis of decision-making. The evaluation result for all the four datasets by NB-KDE is given in table 1.

Table 1 : Evaluation measures of NB-KDE classifier for various opinion reviews

Data set	Precision	Recall	F-Measure	Accuracy
Book	94.2	93.7	93.6	93.6
Home appliances	93.2	93	93	93
Electronics	89	88	87.9	88
Movie	94.4	94.3	94.3	94.3

Table 1 shows the evaluation result of the proposed model for 300 reviews in each category. Totally 281 instances are correctly classified in book review dataset and achieved 93.6% accuracy. Like book review, the classifier also secures 93% of accuracy for home appliances review dataset and 279 instances are correctly classified. Likewise, 281 instances are correctly classified in the review of the electronic item dataset and it achieves 88% accuracy. For movie review, 283 reviews are classified correctly and the accuracy is 94.3%. It shows that the proposed model can achieve a better result for the book, home appliances, electronic item reviews and movie reviews.


Weka Classifier Visualize: ThresholdCurve. (Class value neg)
X: False Positive Rate (Num)
Y: True Positive Rate (Num)
Colour: Threshold (Num)
Select Instance
Reset
Clear
Open
Save
Jitter
Plot (Area under ROC = 0.9821)
1
0.5
0
0
0.5
1
Weka Classifier Visualize: ThresholdCurve. (Class value pos)
X: False Positive Rate (Num)
Y: True Positive Rate (Num)
Colour: Threshold (Num)
Select Instance
Reset
Clear
Open
Save
Jitter
Plot (Area under ROC = 0.9821)
1
0.5
0
0
0.5
1

a) Negative review b) Positive review

Figure 2 ROC curve for training set for negative and positive review

Figure 2 visualize the performance evaluation of the classifier model for the movie review data set. The true positive and false positive rate is needed to draw the ROC curve. The best prediction of ROC curve is depicted in upper left corner. The ROC curve uses the threshold value of the given class to visualize the graph. It shows that 98.2% of area under ROC is achieved.

4.2 Discussion

The training and test data set is used to compute the efficiency of the classifier model. The test data set can be loaded into the classifier model by the "percentage split" option in WEKA environment. The amount of percentage split is 66% for the training data set and 34% for the test data set.

Table 2 : Accuracy of NB-KDE classifier for Book review training set

	Number of Instance	Percentage
Correctly classified instances	139	92.6667 %
Incorrectly classified instances	11	7.3333 %
Total number of instances	150	

Table 3 : Evaluation measures of NB-KDE classifier for Book review for training set

	Precision	Recall	F-Measure	ROC Area	
	1.000	0.853	0.921	0.994	neg
	0.872	1.000	0.932	0.994	pos
Weighted Avg	0.936	0.927	0.926	0.994	

Table 2 represents the summary of the evaluation result of the book review for 150 instances. This summary shows the NB-KDE which achieves 92.6% accuracy. To evaluate the result, the proposed model uses 10 cross-validations for the training book reviews.

Tables 3 represents the evaluation results of NB-KDE for the training dataset of book reviews. The results are evaluated in terms of precision, recall, F-measure and ROC area. The model achieves 100% precision value for negative reviews, and the weighted average value achieved is 93.6%.

Table 4 : Accuracy of NB-KDE classifier for Book review test set

	Number of Instance	Percentage
Correctly classified instances	50	98.0392 %
Incorrectly classified instances	1	1.9608 %
Total number of instances	51	

Table 5 : Evaluation measures of NB-KDE classifier for Book review for test set

	Precision	Recall	F-Measure	ROC Area	
	0.964	1.000	0.982	0.996	neg
	1.000	0.958	0.979	0.998	pos
Weighted Avg	0.981	0.980	0.980		

Table 4 shows the summary of the test set of book reviews. Comparing the tables 2 & 4, the percentage of correctly classified instances has been increased by 98% for the test dataset. The proposed model takes 66% of training data and 34% of test data to evaluate the model. Table 5 the represents the results for the test dataset of the book review. The weighted average of precision, recall, F-measure is also increased by 98%. This shows that the proposed NB-KDE produces higher accuracy for test dataset.

Table 6: Accuracy of NB-KDE classifier for Movie review training set

	Number of Instance	Percentage
Correctly classified instances	1686	84.3%
Incorrectly classified instances	314	15.7%
Total number of instances	2000	

Table 7: Evaluation measures of NB-KDE classifier for Movie review for training set

	Precision	Recall	F-Measure	ROC Area	
	0.824	0.873	0.848	0.924	neg
	0.865	0.813	0.838	0.924	pos
Weighted Avg	0.844	0.843	0.843	0.924	

To check the efficiency of the classifier model the number instances are increased by 2000 for movie review dataset. In table 6, the summary of the result for the training data set is presented. The proposed model correctly classifies 1686 instances out of 2000 instances. Table 7 represents evaluation measurement value for the training dataset. The weighted average is achieved by 84%. The proposed classifier model also achieved a better result in the huge dataset.

Table 8: Accuracy of NB-KDE classifier for Movie review test set

	Number of Instance	Percentage
Correctly classified instances	582	85.5882 %
Incorrectly classified instances	98	14.4118 %
Total number of instances	680	

Table 9 : Evaluation measures of NB-KDE classifier for Movie review for test set

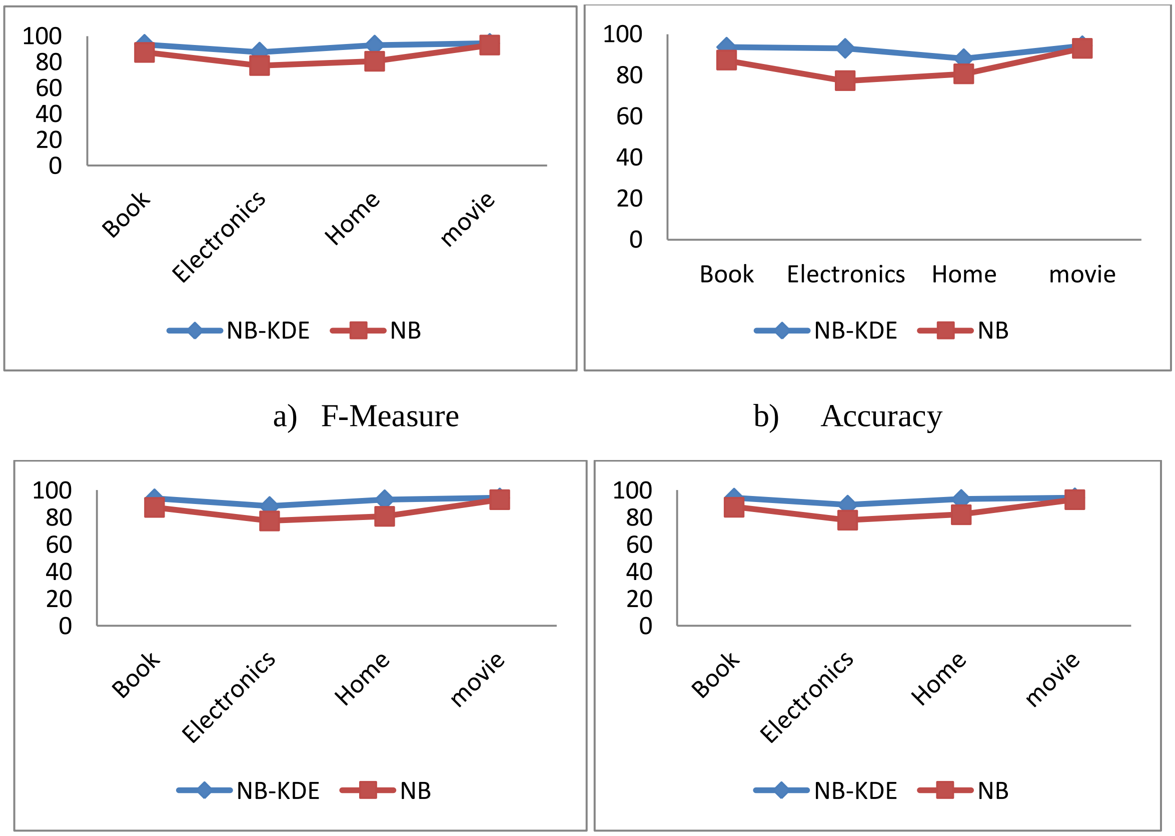
	Precision	Recall	F-Measure	ROC Area	
	0.853	0.863	0.858	0.926	neg
	0.859	0.849	0.854	0.926	pos
Weighted Avg	0.856	0.856	0.856	0.926	

Table 8 & 9 represent evaluation measures of test movie reviews. The accuracy of the classifier model increased from 84.3% to 85.5% for test dataset. In test dataset, 582 instances are correctly classified out of 680 instances.

Table 10: Comparison of results for NB-KDE with simple NB

Dataset	Classifier	Precision	Recall	F-Measure	Accuracy
Book	NB-KDE	**94.2**	**93.7**	**93.6**	**93.6**
	Simple NB	87.4	87.3	87.3	87.3
Home	NB-KDE	**93.2**	**93**	**93**	**93**
	Simple NB	77.9	77.3	77.2	77.3
Electronics	NB-KDE	**89**	**88**	**87.9**	**88**
	Simple NB	81.9	80.7	80.5	80.6
Movie	NB-KDE	**94.4**	**94.3**	**94.3**	**94.3**
	Simple NB	93	93	93	93

In table 10, the comparison of the proposed NB-KDE with simple Naïve Bayes (NB) is shown. It shows that NB-KDE can produce better results than simple NB for all the four dataset. To produce a significant result and to avoid bias, the KDE works in a non-parametric way. To reach accurate results, it estimates the density function for all the attributes instead of selective features.



a) F-Measure

b) Accuracy

c) Recall

d) Precision

Figure 3 Comparison of NB-KDE with simple NB for various data sets

Figure 2 shows the comparison results of NB-KDE with simple NB for F-measure, accuracy, recall and precision. Figure 3.a shows the comparison of simple NB with NB-KDE in terms of F-measure. It represents that the NB-KDE has achieved better results for all the four dataset compared with simple NB, and the figure shows that the NB-KDE is achieved best results for movie, home appliances and book review. The F-measure value for electronic item review for NB-KDE is lower than other datasets. Likewise, figures 3.b to 3.d show that the NB-KDE achieved better results than simple NB in all dataset.

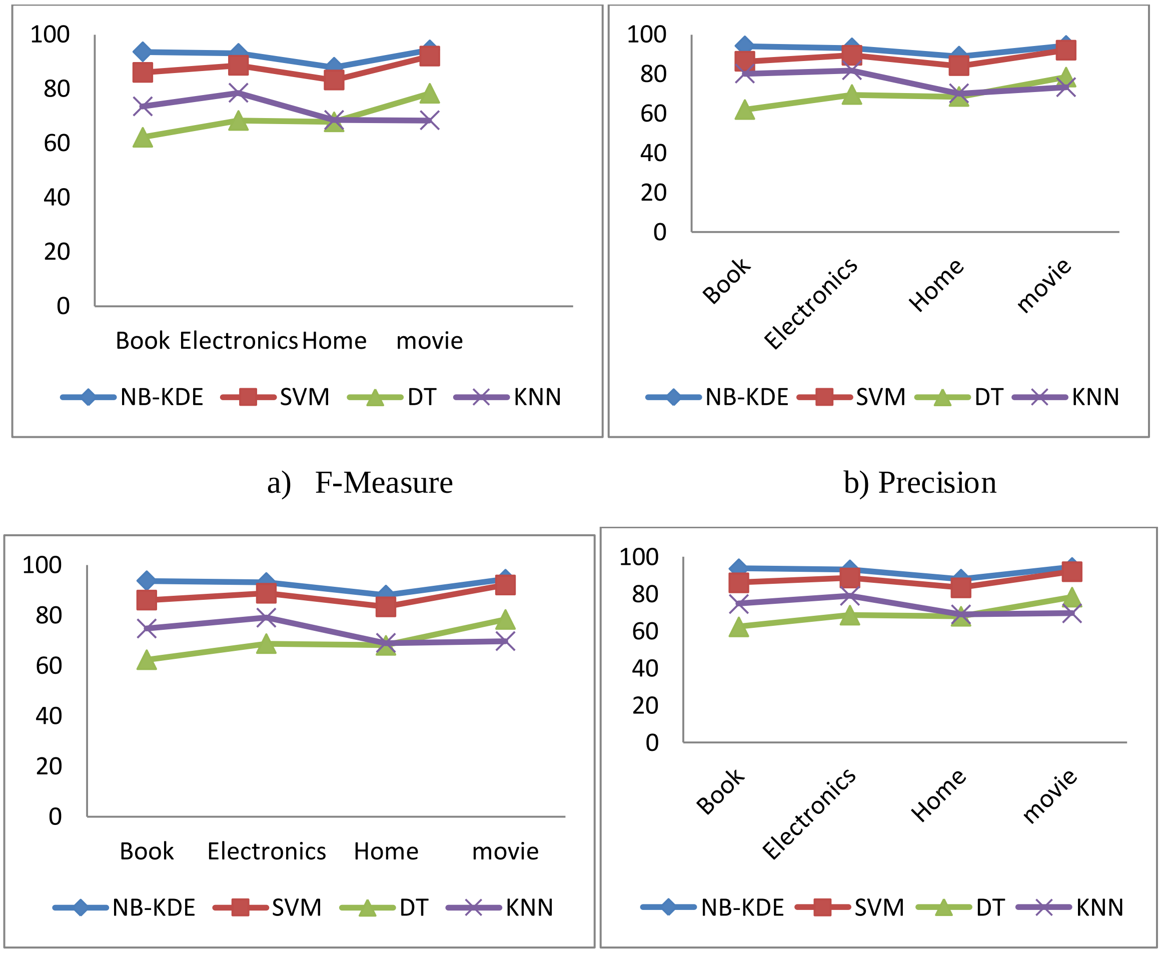
Table 11: Evaluation measure of NB-KDE classifier for various instances

Dataset	No of Instances	Precision	Recall	F-Measure	Accuracy
Book	100	91.8	91	91	91
	150	93.6	92.7	92.6	92.6
	300	94.2	93.7	93.6	93.6
Home appliances	100	95.5	95	95	95
	150	94.6	94	94	94
	300	93.2	93	93	93
Electronics	100	90.3	89	88.9	89
	150	92.8	92	92	92
	300	89	88	87.9	88
Movie	100	100	100	100	100
	150	100	100	100	100
	300	94.4	94.3	94.3	94.3

Table 12 : Evaluation measure of SVM classifier for various instances

Dataset	No of Instances	Precision	Recall	F-Measure	Accuracy
Book	100	58.3	58	57.6	58
	150	86.7	86.7	86.7	86.6
	300	86.4	86	86	86
Home appliances	100	77.9	73	71.8	73
	150	88.4	86.7	86.5	86.6
	300	89.4	88.7	88.6	88.6
Electronics	100	81	76	75	76
	150	84.9	81.3	80.8	81.3
	300	84.2	83.3	83.2	83.3
Movie	100	70	70	70	70
	150	94.1	94	94	94
	300	92	92	92	92

Tables 11 & 12 describe the evaluation measures of NB-KDE and SVM for various instances. Support Vector Machine (SVM) uses the multinomial logistic regression [9], [10] model with a ridge estimator as a calibrator. The poly kernel method is used as the kernel estimator. The SVM works well even in complex problems, but it has some time delay to estimate the kernel. The SVM classifier needs several parameters for handling high dimensional datasets to produce accurate results. The results should depend on the value of the parameter, and the most widely used parameter for SVM is the kernel type. The selection of the kernel parameter is a complex task in SVM. The NB-KDE is a non-parametric approach. Therefore, it does not need any parameter to estimate the kernel function, and the NB-KDE produces better result by calculating the density estimation for each attribute. Concluding from the results of the tables 11 & 12, NB-KDE produces better precision, recall, accuracy and F-measure values for book, movie, home appliances and electronic item reviews than SVM classifier.



a) F-Measure

b) Precision

c) Recall

d) accuracy

Figure 4: Comparison of various classifiers with Book, electronics , homeappliances and moview review dataset

Figure 4, shows the comparison results of various classifiers like Support Vector Machine (SVM) [9], [10] Decision tree (DT) [20] , K-Nearest Neighbor (KNN) [20], with NB-KDE. The KNN classifier is a linear classifier model. In KNN [20] classification, the documents are assigned to a class by majority votes by its neighbors. A positive integer value k represents the votes of its neighbors. The Euclidean distance function is a distance calculator function, which is used to compute the value of k. The accuracy of the classifier increases with the increase in k. It follows its neighbours to predict the label of the test dataset. It is the main drawback of KNN.

The Decision tree [20] is a powerful tool that has a minimum amount of pre-processing to clean the data. For high dimensional data, it becomes very complex to build the model. The selection of optimal split value needs more experience with the training data. Decision tree takes too much of time to build the tree comparing to other classifiers. Tree pruning is performed to reduce the number of nodes after building the model. When the data trained deeply, it suffers from the problem of overfitting.

Figure 4.a to 4.d shows that the proposed NB-KDE produces higher results compared to other classifiers. For 300 Book reviews, NB-KDE achieves 93.6% as the accuracy compared to other classifiers. The accuracy value for SVM is 86%, for DT is 62.3%, and for KNN the accuracy is 74.6%. Concluding the results, the proposed NB-KDE classifier produces the best result in terms of precision, recall, F-measure and accuracy for books, electronics, home appliances, and movie review. Therefore, NB-KDE is the best classifier compared to other well-known classifiers.

5 Conclusion & Future Enhancement

In this paper, we have proposed the Naive Bayes with Kernel Density Estimation to classify the given text reviews. Before building the classifier, the dataset undergoes some pre-processing techniques like stemming, stopword removal and tokenizing. To classify the given attributes, the Naive Bayes model is used to estimate the probability density function for each independent class over the given attributes. To compute the density of the given categorical variable the Naive Bayes uses the kernel density estimation by averaging the large set of kernels. To reduce the noise in the given set of attributes the kernel function includes the bandwidth to compute the probability density function. We argued that the proposed kernel method performs well in situations that violate the prior assumption about the attributes. Experiments showed that the proposed NB-KDE performed better than the simple Naive Bayes method. The reason behind the good performance of the classifier is that it selects the features with maximum discriminative power. The promising improvements of the proposed model are compared with other well known algorithms.

In future, the multi-class classification for opinion mining will be carried out. Based on the ratings, it can be classified as excellent, good, average and so on. To reduce the dimensionality and to improve the accuracy of the classifier, the feature selection task is carried out for opinion mining. It follows either wrapper or filter method for attribute selection. The extension also works with the imbalanced datasets.

References

[1] Abhishek Bhattacharya & David Dunsonb, “Nonparametric Bayes classification and hypothesis testing on manifolds,” in Journal of Multivariate Analysis: Elsevier, Vol.111, pages 1–19, 2012.

[2] Al Amrani, Mohamed Lazaar, Kamal Eddine El Kadiri, “Random Forest and Support Vector Machine based hybrid approach to sentiment analysis,” in Proceedings of the 1st International Conference on Intelligent Computing in Data Sciences: Elsevier, Vol. 127, pages 511–520, 2018.

[3] Aritz Perez, Pedro Larranaga & Inaki Inza, “Bayesian classifiers based on kernel density estimation: Flexible classifiers,” in International Journal of Approximate Reasoning: Elsevier, Vol.50, pages 341–362, 2009.

[4] Arun Meena & T.V. Prabhakar, “Sentence Level Sentiment Analysis in the Presence of Conjuncts Using Linguistic Analysis,” in European Conference on Information Retrieval: Springer, Vol. 4425, pages 573-580, 2007.

[5] Bo Pang and Lillian Lee, “Opinion mining and sentiment analysis,” in Foundations and Trends in Information Retrieval: Now publishers, Vol. 2, pages 1-135, 2008.

[6] B.W. Silverman, "Density Estimation for Statistics and Data Analysis,” in Monographs on Statistics and Applied Probability: Chapman and Hall Publishers, 1986.

[7] Dharmendra Sharma and Suresh Jain, “Evaluation of Stemming and Stop Word Techniques on Text Classification Problem,” in International Journal of Scientific Research in Computer Science and Engineering, Vol. 3, Pages 1-4, 2015.

[8] E. Fersini , E. Messina & F.A. Pozzi, “Sentiment analysis: Bayesian Ensemble Learning,” in Decision Support Systems: Elsevier, Vol.68, pages 26–38, 2014.

[9] Janardhana D R and Manjunath Mulimani, “Sentiment Analysis and Opinion Mining using Machine Learning,” in International Journal of Innovative Research in Computer and Communication Engineering, Vol. 3, pages 9321-9329, 2015.

[10] Jayashri Khairnar and Mayura Kinikar, “Machine Learning Algorithms for Opinion Mining and Sentiment Classification,” in International Journal of Scientific and Research Publications, Vol. 3, 2013.

[11] Jyoti S. Deshmukh, Amiya Kumar Tripathy, “Entropy based classifier for cross-domain opinion mining,” in Applied Computing and Informatics, Vol.14, pages 55–64, 2018.

[12] Kumar Ravi and Vadlamani Ravi, “A survey on opinion mining and sentiment analysis: Tasks, approaches and applications,” in Knowledge-Based Systems: Elsevier, Vol.89, pages 14-46, 2015.

[13] Liu Yang, Han Liguo & Cai Xuesen, “A kernel density estimation based text classification algorithm,” in Advanced Science and Technology Letters, SERSC: Science & Engineering Research Support society, Vol. 78, pages 49-54, 2014.

[14] Liu, B, Synthesis lectures on human language technologies, “Sentiment Analysis and Opinion Mining,” in Morgan & Claypool Publishers. 2012.

[15] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll and Manfred Stede, “Lexicon - Based Methods for Sentiment Analysis,” in Computational Linguistics, Vol. 37, pages 267-307, 2011.

[16] Mangi Kang, Jaelim Ahn and Kichun Lee, “Opinion mining using ensemble text hidden Markov models for text classification,” in Expert Systems with Applications: Elsevier, Vol. 94, Pages 218–227. 2018.

[17] Mans Hulden, Miikka Silfverberg and Jerid Francom, “Kernel Density Estimation for Text-Based Geolocation,” in Association for the Advancement of Artificial Intelligence, Pages, 145-150, 2015.

[18] Michael P. Holmes, Alexander G. Gray and Charles Lee Isbell, “Fast Nonparametric Conditional Density Estimation,” in Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence, Pages, 175-182, 2012.

[19] Min Xiao and Yuhong Guo, “Semi-Supervised Kernel Matching for Domain Adaptation,” in Proceedings of the 26th AAAI Conference on Artificial Intelligence, Pages 1183-1189, 2012.

[20] Muhammad Bilal, Huma Israr, Muhammad Shahid and Amin Khan, “Sentiment classification of Roman-Urdu opinions using Naı¨ve Bayesian, Decision Tree and KNN classification techniques,” Journal of King Saud University – Computer and Information Sciences: Elsevier, Vol. 28, pages 330–344, 2016.

[21] Wei Jin & Hung Hay Ho, “A Novel Lexicalized HMM-based Learning Framework for Web Opinion Mining,” in Proceedings of the 26th International Conference on Machine Learning, Pages 465-472, 2009.

[22] Yoichi Murakami and Kenji Mizuguchi, “Applying the Naïve Bayes classifier with kernel density estimation to the prediction of protein–protein interaction sites,” in Bio informatics, Vol. 26, pages 1841–1848, 2010.