IDENTIFYING USERS ON SOCIAL NETWORKS USING PATTERN RECOGNITION
IN MESSAGES

By

Justin K. Joshuva

Dr. Farah Kandah
Professor of Computer Science
(Chair)

Dr. Craig Tanis
Professor of Computer Science
(Committee Member)

Dr. Sumith Gunesakara
Professor of Mathematics
(Committee Member)
IDENTIFYING USERS ON SOCIAL NETWORKS USING PATTERN RECOGNITION IN MESSAGES

By

Justin Kizhakkyil Joshuva

A Thesis Submitted to the Faculty of the University of Tennesse at Chattanooga in Partial Fulfillment of the Requirements of the Degree of Master of Science: Computer Science

The University of Tennessee at Chattanooga
Chattanooga, Tennessee

December 2017
Copyright © 2017

By Justin Kizhakkayil Joshuva

All Rights Reserved
ABSTRACT

Online social network, such as Facebook, have become a huge part of many people’s lives, often as their main means of communication to others. The blind belief of falsely believing the identity of a person sometimes results in security threats due to the passing of private or confidential information to the wrong user. This may lead to the malicious use of a user’s private information. This work proposes a mathematical model for identifying security threats using pattern recognition with the aid of an extension of the Naïve Bayes method called the Friendship Naïve Bayes. Since specific patterns could be observed by examining the communication history between users, the proposed scheme uses these patterns to authenticate that the new message was written by the same person from the history. The scheme then calculates the probability of identifying the person as either the correct user.
ACKNOWLEDGMENTS

I have received support from many people throughout my years as a Master’s student at the University of Tennessee at Chattanooga (UTC). First, I sincerely thank my advisor, Professor Farah Kandah, for his continuous support. He has diligently prepared me not only for my success in the Master’s program but also for success in my future research work. Professor Kandah has inspired and encouraged me throughout my graduate program at UTC. I would also like to express my gratitude to Professor Tanis, and Professor Gunesakara for agreeing to be part of my committee. I had the wonderful opportunity of being in several of the Statistics classes of Dr. Gunesakara. I also like to thank all my friends in the Computer Science Department. I also would like to thank the faculty and staff of the Computer Science Department.
DEDICATION

This work is dedicated to my family, without their support I would not have been able to finish this work.
# TABLE OF CONTENTS

ABSTRACT .................................................................................. iv

ACKNOWLEDGMENTS .................................................................. v

DEDICATION ................................................................................. vi

LIST OF FIGURES ......................................................................... ix

LIST OF SYMBOLS ......................................................................... x

LIST OF ABBREVIATIONS ............................................................. xii

1 INTRODUCTION ........................................................................ 1

1.1 Background ........................................................................... 1
   1.1.1 Naïve Bayes Algorithm .................................................. 2
   1.1.2 Support Vector Machine ............................................... 4
   1.1.3 Social Trust .................................................................. 4
   1.1.4 Modeling and Identifying relations ................................. 5
   1.2 Questions and Limitations ................................................ 5

2 RELATED WORKS ...................................................................... 7

3 METHODOLOGY ......................................................................... 9

3.1 Confusion Matrix .................................................................... 10
3.2 On Algorithms ....................................................................... 11
   3.2.1 Friendship Naïve Bayes Classifier (FNB) ....................... 12
   3.2.2 Word Tokenization and Prediction ................................. 13
   3.2.3 Verification Algorithm ................................................ 14
3.3 Implementation of the FNB algorithm .................................. 14
3.4 Word Tokenization ............................................................. 17
3.5 Verification Algorithm .......................................................... 17
# LIST OF FIGURES

3.1 An Example of Undirected Graph .................................................. 9
3.2 Confusion Matrix ......................................................................... 11
3.3 Algorithm Steps ........................................................................ 15
3.4 Wordcloud of Most Common Words ............................................. 16
4.1 Reading Data ............................................................................. 20
4.2 Corpus Data .............................................................................. 20
4.3 Cleaning the Corpus ................................................................. 21
4.4 Cleaning the Corpus .................................................................. 21
4.5 Training Data ............................................................................ 21
4.6 Testing Data .............................................................................. 22
4.7 Frequent Words ......................................................................... 22
4.8 Checking the Function ............................................................... 22
4.9 Classifying the Data Using FNB ................................................ 22
4.10 Confusion Matrix ..................................................................... 23
4.11 Table Format of the confusion Matrix ....................................... 23
4.12 Accuracy Comparison ............................................................... 24
4.13 Comparison of FNB and SVM Using Sensitivity and Specificity ............................................................................... 25
4.14 Comparison of FNB and SVM Using PPV and NPV ................... 26
LIST OF SYMBOLS

$P(A|B)$, conditional probability of the event A given that event B has already occurred

$P(A)$, the probability of the event A.

$P(B)$, the probability of the event B.

$P(B|A)$, conditional probability of the event B given that event A has already occurred

$X$, Random variable X

$x$, value of x

$Y$, Random variable Y

$y$, value of y

$X_i$, X at ith location

$y_k$, y at the kth location

$\sigma$, standard deviation

$\mu_{ik}$, Mean of the values.

$\pi$, The PI constant with the value of 3.1415…

$\sqrt{}$, The square root

$p(X_i = x|Y = y_k)$, The conditional probability of the event X=x, given that event Y=y already occurred.

$C_k$, The class at k

$!$, the factorial symbol

$\sum$, single summation

$\prod$, single Product

$p(x|C_k)$, the conditional probability of the event x given that event C has already occurred

$P_{i}^x$, Probabilities of the x

$1 - P_{ki}$, complement of the probability

$\omega$, vector of coefficients

$\omega^T$, transpose of the $\omega$

$C$, Capacity Constraint
\( b \), constant of the error function
\( \zeta \), parameters for the non-separable data
\( \phi \), Kernel function used to transform the input to the feature space
\( M_{ij} \), Messages between users
\( \cap \), symbol for intersection
\( L_{a,b} \), Friendship link between user 'a' and user 'b'
\( W_{a,b} \), weight of the messages between user 'a' and user 'b'
LIST OF ABBREVIATIONS

K-NN, K-Nearest Neighbor
NB, Naïve Bayes
MNB, Multinomial Naïve Bayes
BNB, Bernouli Naïve Bayes
SVM, Support Vector Machine
JGAAP, Java Graphical Author Attribution Program
PMSVM, Power Mean Support Vector Machine
W-SVM, Weibull based Support Vector Machine
SCAP, Source - Code author profile
FNB, Friendship Naïve Bayes
MLE, Maximum Likelihood Estimation
TPR, True Positive Rate
FPR, False Positive Rate
TNR, True Negative Rate
FNR, False Negative Rate
PPV, Positive Predictive Value
NPV, Negative Predictive Value
CNN, Convoluted Neural Networks
BoW, Bag of Words
CHAPTER 1
INTRODUCTION

1.1 Background
Since the start of civilization, humans have developed different ways to communicate with each other: from cave paintings, cuneiforms and hieroglyphics, to scrolls, letters and electronic communications like e-mails, text messages, and mobile communication services. Communication that once took many days now takes a few seconds to reach someone on the other side of the world. The introduction of online social networks like Facebook, Twitter, MySpace, LinkedIn and many other varieties of social media has exponentially accelerated this type of enhanced communicative efficiency among humans.

However, exponential progress has also introduced exponential security risks. Of the many examples of this, one that concerns the scope of this study is the apparent weakness of some privacy algorithms to adequately analyze linguistic characteristics in order to correctly identify the profile users.

Pattern recognition is the automated identification of shapes, forms or patterns of speech [1]. This is also a branch of machine learning that emphasizes the identification of data patterns or data regularities [2]. Pattern recognition in machine learning is divided into two classification methods: supervised and unsupervised. The unsupervised classification method predicts the output from unlabeled data. Supervised classification, on the other hand, and contains classifiers like Naïve Bayes, K-nearest neighbor, Support Vector Machines and neural networks [1].

Another important related concept is Natural Language Processing, a field that focuses on the interactions between human language and computers. Since digital communications use
the human languages transmitted through computers, one can apply the idea of natural lan-
guage processing to pattern recognition. The combination of pattern recognition and natural
language processing is also used in stylometry, the statistical analysis of variations in literary
style between one writer or genre and another. In addition, the algorithms used in Natural
Language Processing are also used in pattern recognition. Algorithms like Naïve Bayes and
Support Vector Machine algorithm are used for document classification in Natural Language
Processing [2].

1.1.1 Naïve Bayes Algorithm

Naïve Bayes is a classification technique based on Bayes’ Theorem wherein the predictors are
assumed to be independent. Bayes Theorem is a theory in statistics where the calculation of
the probability of an event is based on prior knowledge of conditions that might be related
to the event. The equation for Bayes’ Theorem is

\[ P(A|B) = \frac{P(B|A)P(A)}{P(B)} \]  \hspace{1cm} (1.1)

where A and B are events and \( P(B) \neq 0 \), \( P(A) \) and \( P(B) \) are the probabilities of observing
A and B without regard to each other, \( P(A|B) \) is the conditional probability that equals
the probability of event A given that event B has occurred, and \( P(B|A) \) is the conditional
probability of event B given that event A has occurred [3].

In Naïve Bayes classifiers, there are three popular distributions: Gaussian Naïve Bayes the
multinomial Naïve Bayes, and the Bernoulli Naïve Bayes.

The Gaussian Naïve Bayes (GNB): The Gaussian Naïve Bayes is used for data gener-
ated through a Gaussian process which assumes a normal distribution. The GNB equation is

\[ p(X_i = x|Y = y_k) = \frac{1}{\sqrt{2\pi \sigma_{ik}^2}} e^{\frac{1}{2} \left( \frac{x - \mu_{ik}}{\sigma_{ik}} \right)^2} \]  \hspace{1cm} (1.2)

where \( p(X_i = x|Y = y_k) \) is the probability of the x value at given Y, \( \frac{1}{\sqrt{2\pi \sigma_{ik}^2}} e^{\frac{1}{2} \left( \frac{x - \mu_{ik}}{\sigma_{ik}} \right)^2} \) is the
The function for normal distribution, where $\sigma$ is the standard deviation, and $\mu_{ik}$ is the mean of the distribution. Using the normal distribution above the Naïve Bayes is trained for each value of $y_k$ and the new X value is classified [4].

The Multinomial Naïve Bayes (MNB): In the MNB, the features (word frequencies) are generated by a multinomial distribution, where $p_i$ is the probability that an event $i$ occurs. The MNB is typically used for document classification, with events representing the occurrence of a word in a single document. The likelihood of observing the feature vector is given by equation:

$$p(x|C_k) = \frac{\left(\sum_i x_i\right)!}{\prod_i x_i!} \prod_i p_{ki}^{x_i}$$

(1.3)

$p(x|C_x)$ is the probability of x at the given class of C, $\prod_i x_i!$ is the product of all factorial from x, $(\sum_i x_i)$ is the sum of all the x factorials, $\prod_i p_{ki}$ is the product of all the probabilities to the given x level.

Frequency-based probabilities will estimate to zero, which will result in an error due to multiplications by zero. To correct this, a pseudo-count is used to regularize NB. This regularization is called Laplace smoothing [4].

The Bernoulli Naïve Bayes (BNB): In the Bernoulli event model, all features are independent and describe inputs. The Bernoulli model can be used for document classification, where the features are used to identify a word from a predetermined vocabulary. The likelihood of this document occurs in class $C_x$ is given in equation:

$$p(x|C_x) = \prod_{i=1}^n p_{ki}^n (1 - p_{ki})^{(1-x_i)}$$

(1.4)

where $p_{ki}$ is the probability of class $C_x$ for identifying the term $x_i$ [4], $p(x|C_x)$ is the probability of x at the given class of C, $\prod_{i=1}^n p_{ki}^n (1 - p_{ki})^{(1-x_i)}$ is the product of all successes and failures at the given x.

The combination of Bernoulli and Multinomial Naïve Bayes’ models can be used to create very accurate classifiers for document classification.
1.1.2 Support Vector Machine

Support Vector Machine (SVM) is a supervised classification algorithm used for classifying documents or data. The idea behind SVM is that the inputs and outputs are separated by a hyperplane, a line known as the decision boundary. A Support Vector Machine will maximize the margins from both categories to create the best hyperplane. To classify the document, the vectors are classified in multidimensional space and the word frequencies are used. The equation of the SVM:

\[ f(w) = \frac{||w||^2}{2} + C(\sum_{i=1}^{N} \zeta_i) \]  

where the \( w \) represents the coefficient of weights of the messages and \( C \) is the penalty for misidentifying the data.

Using the combination of NB algorithms mentioned above and the Support Vector Machine algorithm, one can classify documents with fairly good accuracy. In this research, the NB theorem was extended to FNB and compared to the Support Vector Machine algorithm in this research to accurately identify the user.

1.1.3 Social Trust

Trust is very important in dealing with people, so the question is: what is trust and how does it apply to social networking? The term trust is defined as a measure of confidence that an entity or entities will behave in an expected manner [5]. Thus, social trust, a term used in social networking, can be defined as a measure of trustworthiness within a social network or group. Social trust implies that members of a social group act according to the expectation that other members of the group are also trustworthy [5].

However, some people within social networks do not adhere to the norms of social trust, raising concerns about the privacy of data in social networking sites like Facebook, Twitter, MySpace, or LinkedIn. Privacy concerns are classified into two categories. One is privacy after the data release, which has to do with the identification of specific individuals in a data set after its release to the general public or to paying customers for specific usage [6].
The second category is privacy information leakage, which is related to details about an individual that are not explicitly stated but rather inferred through other released details and relationships to individuals who may express that relation [7].

1.1.4 Modeling and Identifying relations

Social network relations can be modeled according to different methods. Some of the models used include matrices to represent social relations, statistical models for analysis, and using graphs to represent social relations [7]. Models which use statistical models and graphs are preferable because the former can be used to analyze the relationship among the entities of the model, and the latter can be used to better visualize the relations. People identify different languages completely differently based on lexical and sublexical patterns. Most bilingual people, for example, activate both languages non-selectively even in monolingual settings [8]. That is, they will hear one language and then translate to their native language in order to better understand the second language. Chong et al., did a study of semantics on Twitter, where they identified three steps in semantic classification: subjectivity classification, semantic association, and polarity classification [9]. They classified the tweets as either subjective or objective. The subjective tweets were associated with semantics and used to find the sentiment lexicons. Thus, this technique can be modified and used to identify the lexical analysis of each individual’s typing style and perform identification.

1.2 Questions and Limitations

The main questions behind this research were:

- Is it possible to use pattern recognition to identify users based on the way they write messages?

- How well do the algorithms perform compared to other existing tools?

I have answered the above questions by implementing a new algorithm called Friendship Naïve Bayes (FNB), which is an extension of the Naïve Bayes (NB) algorithm.
The limitations of this research were:

- Accessing the messaging data on Facebook.
- The fact that messages are from one person to another person.
There are various existing authorship attribution techniques. One of the techniques is the Java Graphical Author Attribution Program (JGAAP), which is an authorship program that can identify who wrote a document. JGAAP can use multiple algorithms like the Naïve Bayes, K-NN, Linear SVM, LDA, and Markov Chain. JGAAP inputs a sample of authors and their documents and, given a document from an unknown author, it will show the possible author of the document from the samples [10]. While this is a great tool to find the authors of books and other literary works, this is not a great tool to find the authors of messages in text messages or private messages on social networks. This is because books and literary works are patterned in a very specific manner, while Facebook messages and the like tend not to have a set pattern.

Another way to identify authors is by using Support Vector Machine (SVM), especially the Power Mean SVM (PMSVM) and W-SVM (Weibull - Support Vector Machine) for Open Set Attribution. PMSVM is designed for large-scale image classification and can be used for text classification. The Power SVM uses a coordinate descent method with a gradient approximation and will avoid the over-fitting of the data. W-SVM, meanwhile, is a Weibull-based formulation of the Support Vector Machine that combines binary SVM and Class-1 SVM. The Weibull probability is coupled with the radial basis of the SVM to give a better probability for rejection of the false rejection zone [11].

Boutwell described the authorship attribution using multimodal features. This paper used the NB classifier along with analyzing the life analysis patterns like Twitter login data, common friends, and the signal collection from the mobiles. Combining the NB classifier
and the analysis of the users daily pattern, Boutwell was able to identify the user with high accuracy. This method gives a high efficiency for short messages like tweets and text messages. However, it would not provide useful accuracy in longer messages like Facebook messages [12].

Shreshta et al. proposed a method to identify authors using Convolutional Neural Networks (CNN). This technique takes character N-grams and applies the Convolutional Neural Network model to determine the author. While CNN gives better author attribution of tweets, this, similar to Boutwell's method, will not work for messages like those in the Facebook message system due to only focusing on small sections of text [13].

Authorship Attribution can be applied to texts in other languages other than English, like Chinese, Dutch, and Arabic. Altheneyan and Menai proposed a Naïve Bayes classifier to identify authors of Arabic languages. This work analyzed the usual features like characters, syntactic, semantic, and structural and language-specific features like inflection, diacritics, and elongation. This work gave a high-performance rate of Naive Bayes classifier of Arabic texts [14].

Other methods that can be used to identify users are profile search, content search, self-mention search, and Network Search. The methods for identity matching are Syntactic matching or image matching [15]. While the above methods will identify the users across multiple social networks, they do not help to identify the users from previously logged conversations. The only method that is somewhat similar is Syntactic matching, which compares multiple posts on different social networks.
CHAPTER 3  
METHODOLOGY

Social Networks can be visualized with an undirected weighted graph $G = (V, E)$ (Refer to Figure 3.1). Each vertex ($V$) is a person, and each edge ($E$) is the friendship between two people. The weights on the edges of the graph stand for the message history between the vertices.

![Figure 3.1: An Example of Undirected Graph](image)

This study was focused on messages between a pair of nodes or people. Two of the assumptions made in the study were as follows: the pairs studied were friends on a social network, and a message history existed between these two people. The messages were categorized into different types of communication such as greetings, goodbyes, statements, and questions. The classifier looked at the message history and compared it to the written message. However, the NB is not a single algorithm for training classifiers, but a family of
algorithms based on a common principle. The two types of the NB model that are beneficial for this problem are the Bernoulli and the Multinomial models. As previously discussed, the Bernoulli model is a binomial model that checks whether or not a word is on the list. The Multinomial model, meanwhile, takes the Bernoulli model one step further by not only checking if the word is included but also by checking how often the word is mentioned.

### 3.1 Confusion Matrix

In statistics, a confusion matrix shows the “visualization” of the data. The confusion matrix is a table with two rows and two columns that show the false positives, false negatives, true positives and true negatives. This table also can show the sensitivity, specificity, Positive Predictive Value (PPV), and Negative Predictive Value (NPV).

The values of sensitivity and specificity are used to measure the performance of the classification test. The sensitivity is also known as the true positive rate (TPR), and the specificity is known as the true negative rate (TNR).

The Sensitivity measures the proportion of positives, and sensitivity values avoid the false negatives. Sensitivity can be written mathematically as

\[
sensitivity = \frac{TP}{(TP + FN)} \tag{3.1}
\]

where TP is the number of true positives and FN is the number of false negatives.

The Specificity and Sensitivity are the “opposite sides of the same coin.” The specificity measures the proportion of the negatives and, like sensitivity avoids the false positives. Specificity can be written mathematically as

\[
specificity = \frac{TN}{(TN + FP)} \tag{3.2}
\]

where TN is the number of true negatives and FP is the number of false positives.

The positive and negative predictive values are the proportions of positive and negative results. The positive predictive values are derived from Bayes’ theorem and can be calculated
as

\[ PPV = \frac{TruePositive}{(TruePositive + FalsePositive)} \]  \hspace{1cm} (3.3)

One way to interpret the PPV is that it gives the overall probability that author A wrote the document. For example, if the PPV of a document is 0.2312, then it can be concluded that there is a 23.12% probability that the author wrote the document.

The negative predictive values are the proportion of the negative values. The negative predictive value is calculated as

\[ NPV = \frac{TrueNegative}{(TrueNegative + FalseNegative)} \]  \hspace{1cm} (3.4)

Similar to PPV, the NPV can be interpreted as the overall probability of all negative probabilities.

![Confusion Matrix](image)

**Figure 3.2: Confusion Matrix**

### 3.2 On Algorithms

In this study, a variety of algorithms were proposed: the Friendship Naïve Bayes Classifier (FNB), the Word Tokenization using an N-gram model, and the verification algorithm. The FNB is an extension of the Naïve Bayes algorithm. The word Tokenization algorithm breaks the sentences into particular words by using the “N-gram modeling.” The final algorithm used, verification, checked the accuracy of a class if multiple classes had the same result by checking for unique words.
3.2.1 Friendship Naïve Bayes Classifier (FNB)

In this work, there are multiple words (features), say \( M_1, M_2, \ldots, M_i, \ldots, M_n \) for \( 1 \leq i \leq n \) and authors (classes), say \( C_1, C_2, \ldots, C_j, \ldots, C_k \) for \( 1 \leq j \leq k \), where \( C_j \) is the class of authors, and \( M_i \) is the words that belong to all classes. Since NB algorithm learns the probability of authorship with certain features belonging to a particular class or group, the conditional probability of classifying that person with a given feature belonging to a particular class \( C_j \) is given by the equation:

\[
P(C_j|M_1, M_2, \ldots, M_n) \propto \left( \prod_{j=1}^{n} p(m_i|C_j) \right) \cdot (p(C_i)) \quad \text{for} \quad 1 \leq j \leq k \quad (3.5)
\]

The classes of the Friendship Naïve Bayes Classification algorithm determine how the Bayes Theorem will determine the probabilistic model. For this study, there were only two classes: Good/1st person and bad/2nd person. The decision rule is:

Classify author as Good if

\[
P(C_j|M_1, M_2, M_3, \ldots, M_n) \geq P(C_j|M_1, M_2, M_3, \ldots, M_n)
\]

To keep a high accuracy, we assume that 80% will be the cut-off probability used to measure the probability of the classes.

For example: If \( j = 1 \) then the class is designated as Good (1st person) and the probability will meet or exceed 80%. If \( j = 2 \), then the class is designated as Bad (second person), and the probability will fall below 80%.

Given a vertex (node), with \( M_{ij} \) messages and two classes to choose from \( C_1, C_2 \), the NB determines which class \( C_j \) is more likely under the assumption that messages are independent. The FNB equation is shown in the equation:

\[
\arg\max_x \left[ P(C_x|M_{ij1}, M_{ij2}, \ldots, M_{ijN}) \right] = \arg\max_x \left[ \frac{P(C_x \cap M_{ij1} \cap M_{ij2} \cap \ldots \cap M_{ijN})}{P(M_{ij1}, M_{ij2}, \ldots, M_{ijN})} \right]
\]

(3.6)

The FNB Algorithm is:
Algorithm 1 FNB

1: Collect the Messages
2: if Message in Readable format then
3:  goto step 7
4: else
5:  Convert the messages to CSV format
6: end if
7: Create the Corpus of data
8: Clean the Document
9: Create the Bag of words (sentences to words)
10: Split the data to a testing data and training data
11: Find the most frequent words
12: Create a function that count if the word is present or not
13: Create a testing Matrix and training Matrix
14: Classify the documents by applying FNB + Laplace Smoothing to the matrices
15: Predict the testing matrix(comparing with training)
16: Create a confusion matrix from the prediction data
17: Check values of the confusion Matrix

3.2.2 Word Tokenization and Prediction

The words of sentences are broken down into individual words. Then, the next word a user type can be predicted using an N-gram model, which uses the probability score of individual words. In addition, one can find the entire word sequence of a sentence using the chain rule of probability, which is given by the equation:

\[ P(w_n^1) = \prod_{k=1}^{n} P(w_k|w_{k-1}^{k-1}) \]  \hspace{1cm} \text{(3.7)}

where \( w_n^1 \) is the current word, and \( w_k \) is the word at kth instance. The Maximum Likelihood Estimation (MLE) can be used to estimate the probabilities. The count of words can be gained from the sentences, and it can be normalized so that the counts can be between 0 and 1. The probability of a particular word, given the previous word, is computed by counting the N-grams, and normalizing the sum of all N-grams that share the first word is computed by equation:

\[ P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{\sum_{w} C(w_{n-1})} \]  \hspace{1cm} \text{(3.8)}
which can be written in a general form by

\[ P(w_n|w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1}w_n)}{C(w_{n-N+1}^{n-1})} \]  \hspace{1cm} (3.9)

Algorithm 2  Word Tokenization
\begin{enumerate}
\item Split the messages by user
\item Clean the message
\item Create a Bag of words model
\item Find the unique words of each dataset
\item Save the unique words of authors into separate dictionary files.
\end{enumerate}

3.2.3 Verification Algorithm

The steps of the verification algorithm are

Algorithm 3  Verification Algorithm
\begin{enumerate}
\item Load all the unique wordlists
\item Create a smaller unique wordlist by comparing all wordlists
\item if word is in the small unique list then
\item The probability of that words is added to the classes
\item continue to check the other words
\item end if
\end{enumerate}

3.3 Implementation of the FNB algorithm

The steps of the algorithm can be seen in the flowchart. (Refer to Figure 3.3):
In the algorithm 1, the messages are collected first, and then the messages are checked if they are in the correct format of comma-separated file, if the messages are not in the correct format, then it needs to be converted to a comma-separated file format (line 1 - line 5). Since the messages are in the form of dialogue, the classes (authors) need to be added as a column to the message and a corpus of the data is created (line 7). A corpus is a collection of documents that contains metadata of the document in the form of tag-value pairs. The corpus is cleaned by removing numbers, punctuation and stop words (line 8). The words are then tokenized using the Bag of Words model (line 9). Bag of Words (BoW) is a way of extracting information from text data for use in a machine learning algorithms. In BoW model, the structure of the words is discarded [19]. The data is then split into a training set and testing set (line 10). The training set contains the top 75% of the data, and the testing set contains the remaining 25%. Using the bag of words model and with the aid of the R packages such as tm, RTextTools, naivebayes, dplyr and the caret, the words can be found
and converted to individual words, after which the most common words can be found (line 11). The most frequent words can be visualized in a word cloud form as in Figure 3.4.

Figure 3.4: Wordcloud of Most Common Words

Figure 3.4 shows that the most common words are “will”, “lol”, “go” and “good”. When we use the verification algorithm, the most common words can be discarded to leave us to work with very unique words. Once the most frequent words are found, those words are checked using a function to determine whether those words are in the testing data or training data (lines 12). If the words are found, then they marked as “yes,” otherwise “no.” The testing and training matrix will contain all the words in the training or test dataset and the FNB with Laplace smoothing is applied to the matrix (lines 13 - 14). The testing matrix data is compared to the training set and is used to predict the data (line 15). Once the prediction is complete, a confusion matrix is created. The confusion matrix (Refer to Figure 3.2) contains the True Positive Rate (TPR), False Positive Rate (FPR), True Negative Rate
(TNR) and False Negative Rate (FNR). The TPR is the probability that prediction is true for the 1st class, and FPR is the probability that the data is predicted as true when it is false. Meanwhile, the TNR is the probability that the prediction of the second class, while FNR contains the probability that the data is predicted as negative when it is positive (line 16).

### 3.4 Word Tokenization

If the TPR is low and there are multiple conversations with different authors, then a verification algorithm is used to find the average probability of all conversations. The verification algorithm is divided into two parts, the first part will create the unique datasets (refer to algorithm 2) and the second part will create a smaller unique dataset and compare the probabilities for each word (refer to algorithm 3).

The word tokenization algorithm 2, will create dictionary files for the individual words of each author. We split the messages by the author (line 1). Once the messages are split, then the messages are cleaned by removing all non-alphanumeric characters and removing the stopwords in English (line 2). Line 3 - 4 creates the BoW model by breaking down the sentences to individual words and remove the common words to identify the words that are mentioned only once. After the unique words are found for author, the words are saved to a dictionary file in a comma-separated format (line 5).

### 3.5 Verification Algorithm

The verification algorithm uses the wordlists made using the algorithm 2. To create a smaller unique wordlist, all the dictionary files created in algorithm 2 are loaded (line 1). All the wordlists are compared to each other, and a smaller unique list for each author is made (line 2). The words in the smaller list are compared to the BoW model words from algorithm 1 and if the word is in the list, then the probability of that word is added to the probability of the classes and continue to check other words in the list (line 3 - 5). By creating a smaller
unique wordlist, only the very small unique words will be in the wordlist. By comparing that small list to another list, then it can be concluded if the probability of an author can be added to the overall probability.
The FNB model was evaluated using datasets of the number of messages exchanged in various conversations at various times among twelve people on Facebook. Each conversation had a single common friend, and each conversation was from the beginning of their friendship on Facebook. The length of conversations was important because the FNB model looks at the length as an indicator of the closeness of the subjects. The greater the quantified closeness between the subjects, the better the prediction is.

4.1 Example

Let us consider a dataset as a number of messages at various times. In each dataset, by applying the proposed FNB algorithm, a particular author was compared to the counterpart based on the conversation.

In the steps of the FNB algorithm applied to the example, given in algorithm 1, the first step is to read the data, (Refer to Figure 4.1 and algorithm 1 line 1).
Once the corpus data is created (refer to Figure 4.2 and line 5 of algorithm 1), the corpus is cleaned by removing the punctuation and numbers, converting to lower case and removing the English stop words. (Refer to Figure 4.3 and line 7 of algorithm 1). Once the data is cleaned, the sentences are tokenized into words using the N-gram model. (Refer to Figure 4.4 and algorithm 1 line 8).
Once the words are tokenized, then the data is split into training and testing data (Refer to Figure 4.5 and Figure 4.6).
The most used words in the data are checked (line 11 of algorithm 1), and then a count function is used to check if the word is in the testing or training set (Refer to Figure 4.7 and Figure 4.8).

The classifier is created and it will display the probability for each word for both authors. (Refer to Figure 4.9 and line 14 of algorithm 1).
From the classifier, the author is predicted and displayed in a confusion matrix. (Refer to Figure 4.10 and Figure 4.11 and line 15 - 16 of algorithm 1).

```
Confusion Matrix and Statistics

Reference
Prediction Author A Author B
Author A  403     222
Author B  68      71

Accuracy : 0.6204
95% CI : (0.5849, 0.655)
No Information Rate : 0.6265
P-Value [Acc > NIR] : 0.4273

Kappa : 0.1087
Mc Nemar’s Test P-value : <2e-16
Sensitivity : 0.8556
Specificity : 0.2423
Pos Pred Value : 0.6448
Neg Pred Value : 0.5108
Prevalence : 0.6165
Detection Rate : 0.5275
Detection Prevalence : 0.8181
Balanced Accuracy : 0.5490

'Positive' class : Author A
```

Figure 4.10: Confusion Matrix

```
Total observations in Table:  764

<table>
<thead>
<tr>
<th>actual predicted</th>
<th>Author A</th>
<th>Author B</th>
<th>Row Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>403</td>
<td>222</td>
<td></td>
<td>625</td>
</tr>
<tr>
<td>68</td>
<td>71</td>
<td></td>
<td>139</td>
</tr>
<tr>
<td>Author B</td>
<td>0.144</td>
<td>0.242</td>
<td></td>
</tr>
<tr>
<td>0.758</td>
<td>0.293</td>
<td></td>
<td>764</td>
</tr>
<tr>
<td>Column Total</td>
<td>471</td>
<td>293</td>
<td></td>
</tr>
<tr>
<td>0.616</td>
<td>0.384</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Figure 4.11: Table Format of the confusion Matrix

4.2 Results

The FNB algorithm was applied to pairwise conversations between twelve authors, and the results of the FNB implementation was compared to the SVM.
From the Figure 4.12, it is clear that FNB performed better than or the same as SVM in almost all cases except for Authors K and L. In the cases of both authors K and L, the data sizes were small enough not to get a good prediction using the FNB. Since the accuracy of the model does not give a good idea of the model, other measures must be applied for both Friendship Naïve Bayes and Support Vector Machines. The other measures we will look at the sensitivity and specificity, and the positive predictive values (PPV) and the negative predictive values (NPV).

The charts for sensitivity and specificity show the comparison between Support Vector Machines and Friendship Naïve Bayes algorithm. When these values are compared with the accuracy, it is very clear to see that the Friendship Naïve Bayes model is much better than the Support Vector Machine model. The chart 4.13 supports the idea of FNB performing better than the SVM.
Figure 4.13: Comparison of FNB and SVM Using Sensitivity and Specificity
Comparing the PPV, NPV, sensitivity, and specificity, it is clear that the Friendship Naïve Bayes performed better than Support Vector Machine model. Looking at the accuracy alone, the difference between the two models is not too apparent because the accuracy is either very close or is about the same. However, since sensitivity and specificity contains all information about the predictions, this is the better value for comparison.

When comparing two algorithms like the FNB and SVM, it is important to keep in mind that just checking the accuracy alone would not work. The example of this is given in Chart 4.12, where, for both authors K and L, the SVM performed much better than FNB. Both author K and L had extremely small data sets. Thus, in order to see if FNB performs better than SVM, sensitivity, specificity, PPV, and NPV can be checked (Refer to Figure 4.13, and Figure 4.14).
The current growth of personal communication via social networks is high; and it will not slow down. Thus, the ability to trust people online is very important in today’s society, because people spend much of their time online, especially talking to friends and families on social networks. In addition, given the sensitivity of one’s personal data, the question of “is the person I am conversing with who they claim themselves to be?” becomes a very important one. Using pattern recognition to identify users based on the words they use is one way to identify the user. Every person is unique and, therefore, in everyday conversations, people will use unique words. These unique words and the style they use can be very helpful identifying the users.

While there are different techniques that can identify authors, none of them can be used to effectively identify authors of chat messages or messages on social networks. Books, news articles and magazines all follow a certain literary pattern; social media messages do not have a set pattern. Since these patterns are hard to identify, classification algorithms like Naïve Bayes and Support Vector Machine algorithms can only be used with some modification. The Naïve Bayes was extended to Friendship Naïve Bayes (FNB) to detect and accurately predict the patterns in messages.

The FNB takes in the data, normalizes the data by removing all non-alphabetic characters and English stop words. The normalized data is split into training or testing data. The data from both testing and training is put in the form of a matrix, then compared to each other for prediction. The prediction results in a confusion matrix that can be used to show the positive predictive values, negative predictive values, the sensitivity and the specificity. In
addition, these values are compared to the values of the Support Vector Machine algorithm to understand the accuracy of the FNB algorithm better.

Running the experiments on this model shows that this algorithm will perform better than other algorithms in moderate to large datasets. The only drawback to this algorithm is the size of the datasets. When the data is smaller this will not perform better than other classification algorithms.

The accuracy of the FNB algorithm shows that it can be used as an accurate model to predict the user. FNB can be used to identify the user in any social network provided there is enough data to create a training data and testing data. This algorithm can be applied to different fields of computer science, sociology, and psychology.
REFERENCES


APPENDIX
APPENDIX A
Equations

A.1 Bayes Theorem

The equation for Bayes theorem is:

\[ P(A|B) = \frac{P(B|A)P(A)}{P(B)} \]  \hspace{1cm} (A.1)

where A and B are events and \( P(B) \neq 0 \).

\( P(A) \) and \( P(B) \) are the probabilities of observing A and B without regard to each other.

\( P(A|B) \) is the conditional probability that equals the probability of event A given that event B has occurred.

\( P(B|A) \) is the conditional probability that equals the probability of event B given that event A has occurred [3].

A.2 Gaussian Naïve Bayes Theorem

\[ p(X_i = x|Y = y_k) = \frac{1}{\sqrt{2\pi\sigma_{ik}^2}} e^{\frac{1}{2} \left( \frac{x - \mu_{ik}}{\sigma_{ik}} \right)^2} \]  \hspace{1cm} (A.2)

\( p(X_i = x|Y = y_k) \) is the probability of the \( x \) value at given \( Y \).

\[ \frac{1}{\sqrt{2\pi\sigma_{ik}^2}} e^{\frac{1}{2} \left( \frac{x - \mu_{ik}}{\sigma_{ik}} \right)^2} \] is the function for normal distribution, where \( \sigma \) is the standard deviation, and \( \mu_{ik} \) is the mean of the distribution. Using the Normal distribution above the naïve Bayes is trained for each value of \( y_k \) and new X value is classified [4].

A.3 Multinomial Naïve Bayes Theorem

\[ p(x|C_k) = \frac{(\sum_i x_i)!}{\prod_i x_i!} \prod_i p_{ki}^{x_i} \]  \hspace{1cm} (A.3)

\( p(x|C_k) \) is the probability of \( x \) at the given class of \( C \).

\( \prod_i x_i! \) is the product of all factorial from \( x \). \( \sum_i x_i \) is the sum of all the \( x \) factorials.

\( \prod_i p_{ki}^{x_i} \) is the product of all the probabilities to the given \( x \) level.

A.4 Bernoulli Naïve Bayes Theorem

\[ p(x|C_x) = \prod_{i=1}^{n} p_{ki}^{x_i}(1 - p_{ki})^{(1-x_i)} \]  \hspace{1cm} (A.4)

where \( p_{ki} \) is the probability of class \( C_x \) for identifying the term \( x_i \) [4].

\( p(x|C_x) \) is the probability of \( x \) at the given class of \( C \).
\[ \prod_{i=1}^{n} p_{k_i}^{m_i} (1 - p_{k_i})^{1-x_i} \] is the product of all successes and failures at the given \( x \).

### A.5 Support Vector Machine

Since SVM will divide the inputs and outputs by using a decision boundary, this works great in cases where the input and output are easily separable. In the case of where authors and messages cannot be separated easily, the \( \zeta \) is used a relaxation parameter similar to the Laplace smoothing technique of the NB. The SVM equations becomes

\[ f(w) = \frac{||w||^2}{2} + C \left( \sum_{i=1}^{N} \zeta_i \right) \] (A.5)

where the \( w \) are the coefficient of weights of the messages and \( C \) is the penalty for misidentifying the data.

### A.6 Friendship Naïve Bayes Classifier (FNB)

The equation is developed by

\[ P(C_i | M_1, M_2, \ldots, M_n) = \frac{P(M_1, M_2, \ldots, M_n | C_i)(1)}{P(M_1, M_2, \ldots, M_n)} \text{ for } 1 \leq i \leq k \] (A.6)

The numerator of the fraction on the right-hand side of the equation above is \( P(M_1, M_2, \ldots, M_n | C_i) = P(M_1, M_2, \ldots, M_n, C_i) \) can be expanded to

\[ P(M_1, M_2, \ldots, M_n, C_i) = P(M_1 | M_2, \ldots, M_n, C_i)(P(M_2, \ldots, M_n, C_i) \]

\[ = P(M_1 | M_2, \ldots, M_n, C_i)P(M_2 | M_3, \ldots, M_n, C_i)P(M_3, \ldots, M_n, C_i) \]

\[ = \ldots \]

\[ = P(M_1 | M_2, \ldots, M_n, C_i)P(M_2 | M_3, \ldots, M_n, C_i) \ldots P(M_{n-1} | M_n, C_i)P(M_n | C_i) \] (A.10)

The conditional probability term \( P(M_j | M_{j+1}, \ldots, M_n, C_i) \) becomes \( P(M_j | C_i) \) because of the assumption that features are independent or Naïve. The expression \( P(M_1, M_2, \ldots) \) is constant for all the cases, we can simply say that

\[ P(C_j | M_1, M_2, \ldots, M_n) \propto \left( \prod_{j=1}^{n} p(m_i | C_j) \right) (p(C_i)) \text{ for } 1 \leq j \leq k \] (A.11)

Therefore, the estimator for FNB Classifier is given by \( \hat{C}_j = argmax_{c_j} P(C_j) \prod_{j=1}^{n} P(M_i | C_j) \) which can be also written as

\[ \argmax_{x} [P(C_x | M_{ij1}, M_{ij2}, \ldots, M_{ijN})] = \argmax_{x} \left[ \frac{P[C_x \cap M_{ij1} \cap M_{ij2} \cap \ldots \cap M_{ijN}]}{P[M_{ij1}, M_{ij2}, \ldots, M_{ijN}]} \right] \] (A.12)

Because \( P(M_{ij1}, M_{ij2}, \ldots, M_{ijN}) \) is a positive constant over all possible classes for any user and this becomes irrelevant when all probabilities are compared. This will reduce the original
problem to \( P(M_{aj1}|C) \) for all \( 0 \leq a \leq N \).

### A.7 Word Tokenization

\[
P(w_1^n) = \prod_{k=1}^{n} P(w_k|w_1^{k-1}) \tag{A.13}
\]

Markov assumption is the assumption that the probability of a word depends on the previous word. The generalization of N-gram model can be written as

\[
P(w_n|w_1^{n-1}) \approx P(w_n|w_1^{n-N+1}) \tag{A.14}
\]

\[
P(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})} \tag{A.15}
\]

which can be written in a general form by

\[
P(w_n|w_1^{n-N+1}) = \frac{C(w_1^{n-N+1}w_n)}{C(w_1^{n-N+1})} \tag{A.16}
\]
Justin Joshuva was born in Kerala, India to the parents of Yossuva Kuruvilla and Saramma Kizhakkayil. He is the older of two children with one younger sister, Annie. He and his family moved to New York in 1996. He has attended Museum Jr. High school and then onto Gorton High School. He has finished his high school at Tyner High School in Chattanooga, TN. Justin has enrolled at Chattanooga State Community College as a transfer student then has attended the University of Tennessee at Chattanooga where he studied Applied Mathematics with the concentration in General Mathematics and also minored in Computer Science. Justin is working at Chattanooga State Community College as an adjunct instructor while continuing his education in Computer Science: Information Security Assurance by pursuing a Masters of Science degree at the University of Tennessee at Chattanooga. He plans to graduate in December 2017 and pursue Ph.D. in Computer Science.